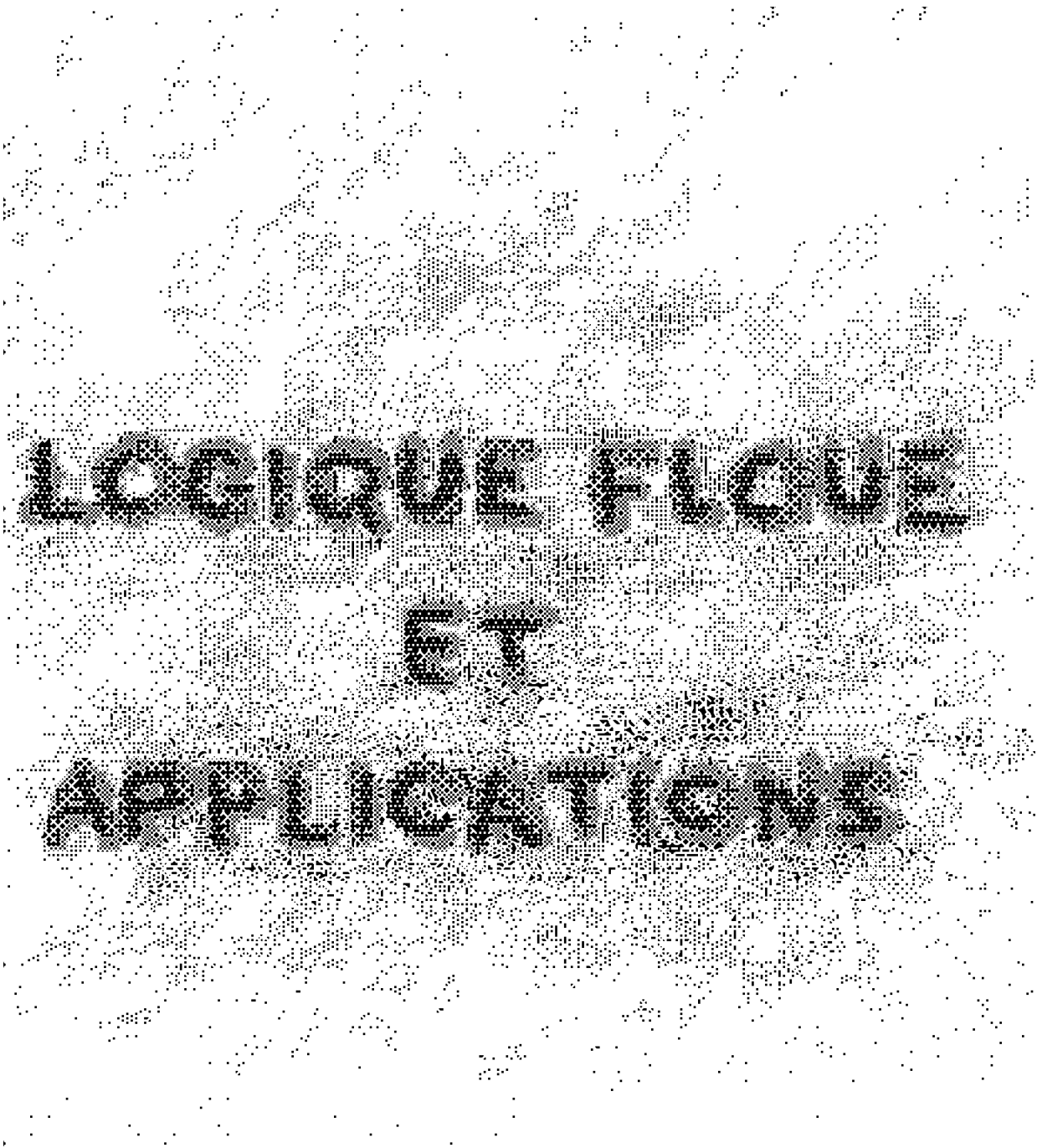


COMMUNAUTE EUROPEENNE
REPUBLIQUE FRANCAISE
MINISTERE DE L'EDUCATION NATIONALE
CONSERVATOIRE NATIONAL DES ARTS ET METIERS
INSTITUT D'INFORMATIQUE D'ENTREPRISE D'EVRY



LOGIQUE FLOUE
ET
APPLICATIONS

Novembre 2003

L.Gacône



Nom Prénom
Classe
Année

À l'heure actuelle, la commande floue est la partie médiatiquement émergée d'un iceberg scientifique.(D. Dubois)

Sommaire

Introduction	4
Chapitre 1 La théorie des sous-ensembles flous	5
1.1 Ensembles flous, nombres flous	
1.2 Relations floues	
1.3 Prédicats vagues	
1.4 Application aux requêtes floues dans une base de données	
Chapitre 2 Mesures de confiances	20
2.1 Mesures de confiance et théorie des possibilités	
2.2 La théorie de Dempster-Shafer	
Chapitre 3 Operateurs logiques	30
3.1 Les connecteurs logiques	
3.2 Le problème de l'agrégation	
3.3 Différentes formes de l'implication	
3.4 Le modus ponens généralisé	
Chapitre 4 Le contrôle flou	43
4.1 Algorithme du contrôleur flou	
4.2 Applications	
4.3 Réalisations en matériel	
Chapitre 5 Problèmes d'apprentissage	56
5.1 Classement et classification	
5.2 Les arbres de décision flous	
5.3 Réseaux de neurones et neuro-flou	
5.4 Algorithmes génétiques et stratégies d'évolution	
Conclusion	78
Annexe 1 Rappels de logique classique	79
A.1.1 Langage des propositions	
A.1.2 Langage des prédicats	
A.1.3 Arithmétique	
A.1.4 Théorie des ensembles et ensembles flous	
A.1.5 Récursivité	
Annexe 2 Logiques non classiques	85
A.2.1 L'intuitionisme	
A.2.2 La logique modale	
A.2.4 La logique des défauts	
A.2.5 Les logiques temporelles	
Annexe 3 Langage Fril	91
A.3.1 Rappel sur le langage Prolog	
A.3.1 Langage de programmation Fril	
Annexe 4 Le langage Mvl	98
A.4.1 Rappels sur les treillis	
A.4.2 Langage de programmation Mvl	
Annexe 5 Les réseaux bayésiens	103
A.5.1 Rappel de probabilités	
A.5.2 Définition d'un réseau bayésien	
Annexe 6 Aperçu d'automatique	108
Annexe 7 Quelques procédures	109
Problèmes	116
Bibliographie	124

Introduction

La logique floue, ou plus généralement le traitement des incertitudes, a pour objet d'étude la représentation des connaissances imprécises et le raisonnement approché. On peut donc la situer à côté des heuristiques de résolutions de problèmes, des systèmes experts, de l'apprentissage, de l'intelligence artificielle distribuée et même du traitement de la langue naturelle, domaines qui composent les techniques d'intelligence artificielle au sein des sciences cognitives. C'est de cet aspect «intelligence artificielle», où s'établissent des modèles du comportement intellectuel humain, que traite ce livre à travers les applications concrètes qui y sont détaillées et du lien avec des techniques d'apprentissages telles que les réseaux de neurones et les algorithmes génétiques (chapitre 5).

Cependant la logique floue peut être intégrée, à côté d'autres extensions, à la logique, qui, en toute généralité peut être vue comme la grammaire des mathématiques (la logique floue a d'ailleurs donné lieu à toute une généralisation des mathématiques classiques fondée sur le concept très simple de sous-ensemble flou). C'est pourquoi on trouvera en annexe d'autres points de vue sur la logique et les fondements du raisonnement.

Enfin, par ses nombreuses applications industrielles en commande, la logique floue est aussi associée à l'automatique. Ces thèmes n'étant pas le propos du livre, figurent néanmoins en annexe.

Dans les problèmes de prise de décision, d'aide au diagnostic et plus généralement dans tous les systèmes à base de connaissances, on souhaite, à partir d'observations, parvenir à une conclusion qui peut être la détermination d'un objet ou une action à prendre. Or lors du fonctionnement de ces systèmes, interviennent des connaissances mal définies, mal décrites et imparfaitement connues, puis au niveau des règles d'inférence, intervient un traitement imparfait et incomplet du déroulement de la déduction, enfin survient le problème du traitement des contradictions et de la fusion (agrégation) de données voisines.

Tous les problèmes concrets sont, en fait, confrontés aux notions d'incertitude et d'imprécision. Ces deux notions sont habituellement mêlées et c'est essentiellement l'observation statistique qui induisait, jusqu'à présent dans la pratique, la mesure probabiliste des incertitudes.

Mais la théorie des probabilités reste assez rigide et il existent d'autres types d'incertitudes liées à la difficulté des observations, aux imprécisions linguistiques, à la fiabilité tant des observateurs humains que des capteurs et instruments de mesure, à l'utilisation de connaissances empiriques et à l'imprécision du raisonnement humain. Toutes ces questions de l'utilisation de catégories linguistiques habituelles et du raisonnement humain vont nous amener à distinguer certains concepts.

Imprécision L'imprécision, c'est ce qui est relatif au contenu d'une proposition «mesurer environ 1,75m», «avoir entre 20 et 25 ans», on ne donne pas de valeur précise mais un intervalle, une «fourchette». Ainsi, ce qui était couramment appelé incertitude probabiliste ou intervalle de confiance, (cas par exemple d'un poids donné à 10g près) est une «imprécision».

Sous-ensembles flous Il se peut, de plus, que ces ensembles où les variables peuvent prendre leurs valeurs, aient des frontières mal définies, c'est bien évidemment le cas de l'exemple «grand», la notion de sous-ensemble flou va nous permettre de prendre en compte cette nouvelle notion (chapitre 1).

Incertain L'incertain est un coefficient apporté au fait qu'une proposition soit vraie ou fausse. Dans «il est possible qu'il soit grand», le prédicat «grand» est considéré comme vrai ou faux sans aucune nuance, mais c'est la proposition qui est douteuse. Nous voyons ici, pour une proposition qui reste booléenne (soit vrai soit fausse) la notion de possibilité et nécessité (chapitre 2). Par ailleurs, donner un degré de vérité situé entre 0 pour le «faux» et 1 pour le «vrai», constitue une incertitude probabiliste et a été, plus généralement, envisagée par les logiques multivaluées (chapitre 3).

Prédicats vagues Dans «être jeune», «il a environ 1,75m», le prédicat est quantitatif mais mal défini et dans «l'ambiance est bonne» ou «l'italien est proche du français», il est qualitatif ou difficilement quantifiable. En fait l'opposition classique entre quantitatif et qualitatif n'est pas elle-même bien définie. Ces propositions ne sont plus booléennes (chapitre 1).

Comparaisons floues Lorsqu'on exprime «x est bien plus petit que y», on peut se ramener à des prédicats vagues unaires par des définitions telles que x, y voisins si et seulement si le rapport x/y est environ égal à 1 (chapitre 1).

Modificateur de prédicat En prenant par exemple «il est presque grand» il s'agit d'une proposition certaine dans laquelle «presque» peut être regardé comme un modificateur de prédicat «presque grand» ou «très grand» sont alors d'autres prédicats obtenus à partir de «grand».

Connecteur flou Dans les propositions «il est presque grand» ou «il est grand est presque vrai», nous avons deux concepts complètement distincts. Le premier modifie un prédicat, en demeurant éventuellement dans la logique booléenne, le second modifie la valeur de vérité de la proposition, ce pourrait être un connecteur au même titre que la négation (chapitre 3).

Quantificateurs flous Ils sont utilisés fréquemment dans le raisonnement dit «de sens commun» avec des expressions telles que «la plupart des», «presque tous», «peu de», ces quantificateurs peuvent être numériques dans le cas «60% des présents», ou flous dans «environ la moitié des participants» (chapitre 1).

Raisonnement approximatif Naturellement les règles sont entâchées d'incertitude (chapitre 3) : «il est probable que s'il pleut, la cérémonie sera reportée», «les mammifères ne pondent pas d'oeufs» (invalide dans certains cas), «le vent d'est est signe de beau temps» (certitude 0,8), «les trains de banlieue arrivent à l'heure avec une probabilité 0,6» (connaissance statistique), «si un homme est âgé, ses cheveux sont blancs» (faits décrits avec imperfection), «un élève qui réussit bien en mathématiques est généralement bon en physique» (certitude non absolue).

Règles graduelles Le traitement du raisonnement humain est encore confronté au plus vaste problème des règles graduelles quantifiables : «plus un diamant est gros, plus il est cher», ou qualitatif : «plus une tomate est rouge, plus elle est mûre», ou encore hybride (mêlant imprécision et incertitude) «plus un individu est jeune, plus il y a de chance pour qu'il soit célibataire».

Le raisonnement approximatif et la prise en compte de faits imprécis et incertains sont tout à fait naturels à l'homme. Leur formalisation en vue de traitement automatique est encore rudimentaire, mais connaît un développement rapide, et les applications bien que nombreuses en sont encore à des systèmes partiels et surtout les systèmes de décision véritablement opérationnels sont rares hormis le contrôle flou, mais c'est une direction de recherche en plein essor. Les domaines d'applications de ces outils formels sont bien sûr tous les problèmes courants mal spécifiés ou bien qui sont difficilement modélisables, ainsi en optimisation, la recherche de consensus, en classification et recherche documentaire, dans les domaines médical et économique, et sur le plan industriel, l'automatisation des tâches de surveillance et la commande floue (chapitre 4). Ce dernier domaine n'utilise que très peu de concepts, mais son succès est un fort stimulant pour la recherche.

Certaines remarques marginales, les applications ainsi que les exercices sont rédigés en plus petits caractères, ils peuvent être sautés en première lecture.

Je remercie A.Autin, □A.Cornuejols, P.Facon, B.Grau, M.Jouve, M.Mallem et R.Mijoule pour leur relecture du manuscrit.
1997, l'auteur.

Chapitre 1

La théorie des sous-ensembles flous

Nous présentons dans ce chapitre, le concept de base de sous-ensemble flou. C'est à partir de cette idée extrêmement simple que peuvent être généralisées les relations unaires telles que « x est grand» ou « x est très grand» et des relations binaires telles que « x est voisin de y » ou « x est nettement plus petit que y », puis les quantificateurs flous. Ce concept est adapté à la description des situations intermédiaires, telles que «la plupart», «peu de», «presque tous» ...

1.1. Ensembles flous, nombres flous

Dans un ensemble de référence E , depuis [Zadeh 65] et [Kaufmann 72], un sous-ensemble flou de ce référentiel E est caractérisé par une fonction d'appartenance μ de E dans l'intervalle des nombres réels $[0, 1]$ (degré d'appartenance qui est l'extension de la fonction caractéristique d'un sous-ensemble classique). En fait un sous-ensemble flou (nous dirons plus brièvement un ensemble flou) est formellement défini par l'application μ , mais pour se ramener au langage des mathématiques classiques, nous parlerons d'un ensemble flou A , et noterons μ_A sa fonction d'appartenance.

NOYAU, SUPPORT ET OPÉRATIONS SUR LES ENSEMBLES FLOUS

Pour un sous-ensemble flou A d'un référentiel E on donne les définitions suivantes :

Noyau $N(A) = \{x / \mu_A(x) = 1\}$ Les éléments «vraiment» dans A .
Support $S(A) = \{x / \mu_A(x) \neq 0\}$ Ceux qui y sont à des degrés divers.

Pour un ensemble classique A , noyau et support sont confondus avec A , et sa fonction caractéristique μ n'admet que 0 ou 1 pour valeurs.

EXEMPLE D'ENSEMBLES FLOUS

L'intervalle flou couramment utilisé dans \mathbb{R} est décrit par sa fonction d'appartenance. Le plus simple type pour ce qu'il est convenu d'appeler un «intervalle flou» est une représentation trapézoïdale :

$$\text{On pose } \mu_A(x) = \begin{cases} 0 & \text{si } x < a - \alpha \text{ ou } b + \beta < x, \text{ (} x \text{ hors du support de } A\text{)} \\ 1 & \text{si } a < x < b, \text{ (} x \text{ dans le noyau de } A\text{)} \\ 1 + (x - a) / \alpha & \text{si } a - \alpha < x < a, \\ 1 - (b - x) / \beta & \text{si } b < x < b + \beta \end{cases}$$

La notation (a, b, α, β) , souvent utilisée dans les applications informatiques, est alors très simple pour ces «intervalles flous».

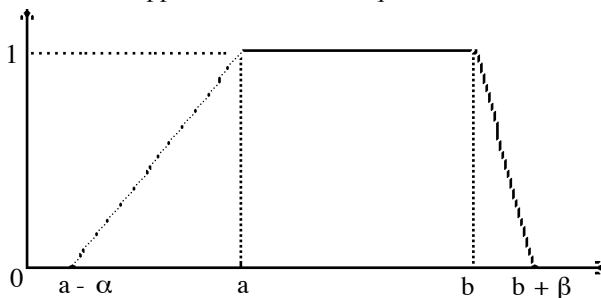


Figure 1.1 Ensemble flou trapézoïdal

Une représentation à côtés paraboliques (figure 1.2) est parfois employée avec :

$$\mu_A(x) = \begin{cases} 0 & \text{si } x < a - \alpha \text{ ou } b + \beta < x, \text{ (x hors du support de A)} \\ 1 & \text{si } a < x < b, \text{ (x dans le noyau de A)} \\ 2(x - a + \alpha)^2 / \alpha^2 & \text{si } a - \alpha < x < a - \alpha/2 \text{ alors} \\ 1 + 2(x - a)^2 / \alpha^2 & \text{si } a - \alpha/2 < x < a \text{ alors} \\ 1 - 2(b - x)^2 / \beta^2 & \text{si } b < x < b + \beta/2 \text{ alors} \\ 2(x - b + \beta)^2 / \beta^2 & \text{si } a + \beta/2 < x < b \text{ alors} \end{cases}$$

Dans les deux cas $[a, b]$ est le noyau, $[a - \alpha, b + \beta]$ est le support.

On utilise aussi des fonctions gaussiennes $\mu(x) = \exp[-(x - m)^2 / 2\sigma^2]$ atteignant 1 uniquement pour la valeur modale m .

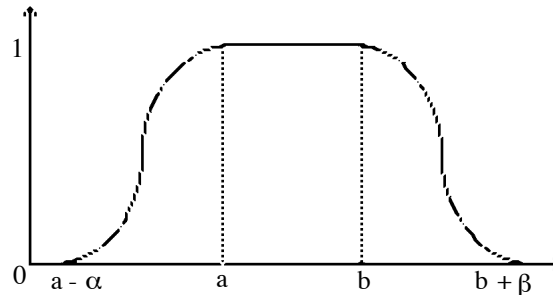


Figure 1.2 Exemple d'ensemble flou convexe

LES NOMBRES FLOUS TRIANGULAIRES

Ils sont définis par $\mu(x) = 1$ pour $x = m$ (le mode), par 0 si $|x - m| > \sigma$, et enfin par le fait que μ est continue et affine par morceaux. Plus généralement par deux fonctions L, R semi-continues inférieurement, de valeur 1 en 0, décroissantes au sens large et de limite 0 à l'infini (ci-dessous, figure gauche). La fonction d'appartenance est alors définie par $\mu(x) = \text{si } x < m \text{ alors } L(m - x) \text{ sinon } R(x - m)$. On pourra voir plus loin qu'en définissant les opérations arithmétiques sur des ensembles flous réels, la somme de nombres triangulaires reste un nombre triangulaire.

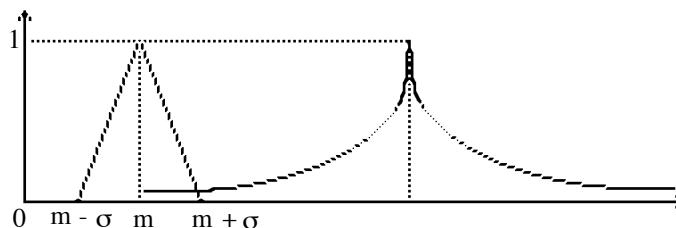


Figure 1.3 Nombres flous triangulaire et quelconque

Soit par exemple (m, a, b) une notation commode pour le nombre flou de mode m , dont le noyau est l'intervalle $[m - a, m + b]$ et à «pentes droites», alors on peut vérifier des relations simples $(m, a, b) + (m', a', b') = (m + m', a + a', b + b')$ et pour la différence $(m, a, b) - (m', a', b') = (m - m', a - a', b - b')$.

Mais ce n'est plus vrai pour le produit et le quotient, ce qui oblige tous les systèmes à base de connaissances adoptant cette représentation simple à faire des approximations quant aux résultats opératoires afin de conserver une représentation homogène.

HAUTEUR D'UN SOUS-ENSEMBLE FLOU

C'est la borne supérieure $ht(A) = \sup_{(x \in X)} \mu_A(x)$ de la fonction d'appartenance, aussi n'est-elle pas nécessairement atteinte. Par exemple, si G comme «grand» est l'ensemble flou défini par $th(x)$ sur \mathbb{R}^+ , $th(x)$ croît de la valeur 0 en 0 vers la limite 1 quand x tend vers l'infini, mais cette limite 1 n'est jamais atteinte (figure 1.4 ci-dessous). Le noyau de G est l'ensemble vide, son support est \mathbb{R}^{+*} et sa hauteur 1.

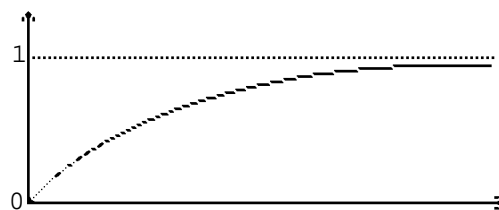


Figure 1.4 Ensemble flou de hauteur 1 non atteinte

ENSEMBLE FLOU NORMALISÉ

Un ensemble flou est dit normalisé s'il est de hauteur 1.

ENSEMBLES FLOUS DISCRETS

Si le support d'un ensemble flou est fini, il est possible de donner la fonction d'appartenance par l'énumération de ses valeurs, ainsi par exemple dans le référentiel $E = \{a, b, c, d\}$, on note habituellement $A = \{a / 0.1, b / 0.5, c / 0.3, d / 0\}$ (ou en couples, en indices ou encore avec le «deux points») le sous-ensemble flou formellement défini par les degrés d'appartenance :

$$\mu(a) = 0.1, \mu(b) = 0.5, \mu(c) = 0.3 \text{ et } \mu(d) = 0.$$

INCLUSION

Par définition, l'inclusion est étendue grâce à : $A \subseteq B \Leftrightarrow \mu_A \leq \mu_B$

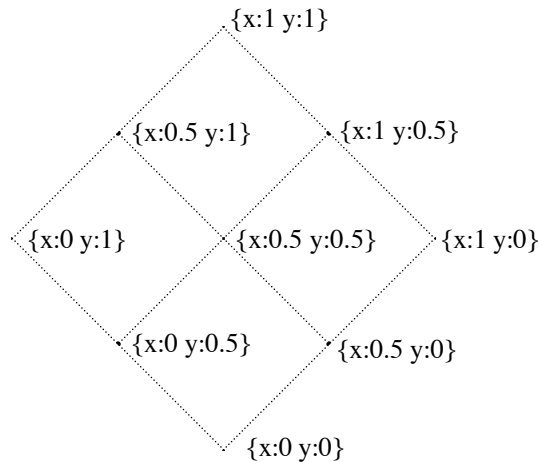


Figure 1.5 Exemple des parties floues d'un ensemble fini $E = \{x, y\}$ en se limitant aux seuls degrés d'appartenance 0, 0.5 et 1. Le treillis de tous les sous-ensembles flous avec ces seules valeurs, est formé des 9 parties ci-dessous dont 4 sont exactes (sans la valeur 0.5) et les 5 autres floues.

Cette définition d'inclusion est une simple relation de domination entre les fonctions d'appartenance, on peut facilement vérifier qu'elle généralise l'inclusion des ensembles classiques. Par exemple, on pourrait définir $A = \text{«trentaine»}$ et $B = \text{«adulte»}$ par des ensemble flous avec A inclus dans B .

Exercice 1.1

Représenter le treillis des parties floues de $E = \{x, y\}$ avec les valeurs d'appartenance 0, 1/4, 1/2, 3/4. Puis en se limitant aux valeurs 0, 1/2, 1, mais pour $E = \{x, y, z\}$

Exercice 1.2

Combien y a-t-il de parties floues dans un ensemble de n éléments, si on se limite à p valeurs d'appartenance ? (Réponse p^n).

COMPLÉMENT D'UN SOUS-ENSEMBLE FLOU

Le complémentaire d'un ensemble flou A dans un ensemble de référence E est naturellement défini par la relation (nous utiliserons le symbole de négation \neg) :

$$\mu_{\neg A} = 1 - \mu_A$$

Cette opération est involutive, c'est à dire $\neg\neg A = A$, on a d'autre part les propriétés $\neg\emptyset = E$ et $\neg E = \emptyset$. Ce n'est cependant pas un complément au sens des treillis car en général : $A \cap \neg A \neq \emptyset$ et $A \cup \neg A \neq E$ mais c'est la seule opération n possible sur les degrés d'appartenance, qui soit une fonction continue strictement décroissante, renversant les valeurs de 0 et 1 et vérifiant $n(x) + n(1 - x) = 1$, elle est alors involutive (n composée avec elle-même donne l'identité $n(n(x)) = x$).

On peut remarquer que pour la négation, support et noyau d'un ensemble flou sont des notions duales (si E est le référentiel) :

$$\text{supp}(\neg A) = E - \text{noy}(A) \quad \text{et} \quad \text{noy}(\neg A) = E - \text{supp}(A)$$

Exercice 1.3

Trouver une fonction continue, strictement décroissante, involutive sur $[0, 1]$ qui ne soit pas le complément à 1 ordinaire $f(x) = 1 - x$. Montrer qu'il en existe une infinité.

UNION, INTERSECTION

On définit l'union et l'intersection de deux ensembles flous A et B , comme respectivement le plus petit ensemble flou contenant A et B , et le plus grand ensemble flou contenu dans A et dans B d'autre part. En d'autres termes :

$$\mu_{A \cup B} = \max(\mu_A, \mu_B) \quad \mu_{A \cap B} = \min(\mu_A, \mu_B)$$

Toutes les propriétés de treillis distributif et les relations de Morgan demeurent, ainsi l'idempotence

$$A \cap A = A \quad A \cup A = A$$

la commutativité $A \cap B = B \cap A$ $A \cup B = B \cup A$

l'associativité :

$$A \cap (B \cap C) = (A \cap B) \cap C \quad A \cup (B \cup C) = (A \cup B) \cup C$$

les distributivités mutuelles

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$

les relations de Morgan

$$\neg(A \cap B) = (\neg A) \cup (\neg B)$$

$$\neg(A \cup B) = (\neg A) \cap (\neg B)$$

et les lois d'absorption

$$A \cup (A \cap B) = A \cap (A \cup B) = A$$

La démonstration de ces propriétés est issue des propriétés analogues pour min, max et le complément à 1.

Par contre $A \cap \neg A = \emptyset$ et $A \cup \neg A = E$ ne sont plus vrais.

Il faut remarquer que tout ensemble «vraiment flou» c'est à dire non exact, et défini par une fonction d'appartenance continue, coupera toujours son complémentaire à la hauteur 1/2.

Ceci, signifiant qu'un fait et son contraire peuvent coexister, constitue un point commun avec les logiques multivaluée ou intuitionniste qui remettent en cause le principe du «tiers-exclu» (annexe 1).

On justifie les choix de min et de max par le fait que ce sont les seules opérations possédant les propriétés de commutativité, idempotence, associativité, distributivités mutuelles, continues, mais vérifiant également :

$$\mu_{A \cap B} \leq \mu_A, \mu_B \leq \mu_{A \cup B} \text{ ainsi que } \min(1, 1) = 1 \text{ et } \max(0, 0) = 0$$

L'union et l'intersection sont illustrées par les figures ci-dessous (cependant d'autres définitions peuvent être prises comme on le verra au chapitre 3, en gras dans la figure droite, par exemple le simple produit des deux fonctions d'appartenances) :

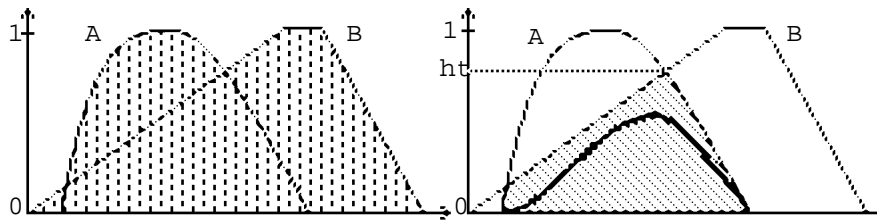


Figure 1.6 Union et intersection de deux ensembles flous

CARDINAL D'UN ENSEMBLE FLOU

On peut définir le nombre d'éléments d'un ensemble flou A par $\text{card}(A) = \sum \mu_A(x)$ dans le cas fini, et par l'intégrale de la fonction d'appartenance μ_A si A est continu. On retrouve bien le nombre d'éléments dans le cas où A est exact.

PRODUIT CARTÉSIEN

Le produit cartésien est défini par $\mu_{A*B}(x, y) = \min[\mu_A(x), \mu_B(y)]$.

Si R est un ensemble flou de $E*F$ (une relation floue) sa projection sur E sera définie par :

$$\mu_{\pi(R, E)} = \sup\{\mu_R(x, y) / y \in F\} \text{ conformément au principe d'extension énoncé plus loin.}$$

DISTANCE DE HAMMING

La notion de distance entre ensembles flous peut être utile pour définir des relations telles que «à peu près égal» ou «très supérieur à», il s'agit :

soit de la distance de Hamming : $d(A, B) = \sum_{(x \in E)} |\mu_A(x) - \mu_B(x)|$ qui est l'intégrale $\int_{\mathbb{R}} |\mu_A(x) - \mu_B(x)| dx$ sur l'ensemble des réels R, soit de la distance euclidienne $e^2(A, B) = \sum_{(x \in E)} (\mu_A(x) - \mu_B(x))^2$.

En outre, si le référentiel est fini, il peut être intéressant de définir une distance relative par le rapport

$$\delta(A, B) = d(A, B) / \text{card}(E) \text{ qui sera maximal égal à 1 entre les parties vide et pleine.}$$

Exercice 1.4

En mesurant les distances de Hamming entre les sous-ensembles flous figurés sur le treillis de la page précédente, montrer que chaque arc correspond à la même distance et que ces distances s'additionnent quel que soit le chemin montant ou descendant. Etudier cette distance sur les treillis de l'exercice 1.1.

ALPHA-COUPES D'UN ENSEMBLE FLOU A ET INDICES

Ce sont les ensembles exacts définis par $A_\alpha = \{x \in X / \mu_A(x) \geq \alpha\}$, on a les propriétés :

$$(A \cup B)_\alpha = A_\alpha \cup B_\alpha \quad (A \cap B)_\alpha = A_\alpha \cap B_\alpha \text{ Mais en général, et sauf pour } 1/2, \text{ on a } (\neg A)_{1-\alpha} \neq \neg(A_\alpha).$$

SOUS-ENSEMBLE VULGAIRE ASSOCIÉ À UN ENSEMBLE FLOU

$\underline{A} = \{x / \mu_A(x) \geq 0.5\}$ est l'ensemble des x que l'on peut davantage considérer dans A que dans son complémentaire. \underline{A} minimise la distance euclidienne entre les deux. On peut établir les relations $\underline{A \cup B} = \underline{A} \cup \underline{B}$, $\underline{A \cap B} = \underline{A} \cap \underline{B}$ et $\underline{\neg A} = \neg \underline{A}$

Exercice 1.5

Soit (G, .) un groupe c'est à dire un ensemble muni d'une loi binaire notée par le point, qui est associative, qui possède un élément neutre noté 1, et telle que pour tout x de G, il existe un inverse x' vérifiant $x.x' = x'.x = 1$. Un sous-groupe est une partie de G stable pour l'inverse et l'opération binaire. Montrer que si A est un sous-ensemble flou de G, on a les équivalences :

$$\begin{aligned} \forall x \in G \quad \mu_A(x') &\geq \mu_A(x) \\ \Leftrightarrow \forall x, y \in G \quad \mu_A(x, y') &\geq \min(\mu_A(x), \mu_A(y)) \\ \Leftrightarrow \forall \alpha \in [0, 1] \quad A_\alpha &\text{ sous-groupe de } G. \end{aligned}$$

On dira alors que A est un groupe flou dans G.

INDICE DE FLOU [Kaufmann 72]

Plusieurs indices ont été proposés afin de caractériser l'aspect plus ou moins flou d'un sous ensemble. Celui de Kaufmann, dans le cas où E est fini est :

$$v(A) = (2/n) * (\sum_{x \in E} \mu_{A \cap \neg A}(x)) = 1 - (2/n) \sum |0.5 - \mu_A(x)| = (2/n) \sum |\mu_{\underline{A}}(x) - \mu_A(x)| = (2/n) * d(A, \underline{A})$$

C'est un élément de [0, 1] pouvant caractériser le «flou» avec les propriétés :

$$v(\neg A) = v(A) \quad \text{et} \quad v(A \cup B) + v(A \cap B) = v(A) + v(B)$$

Exemple : pour un ensemble flou représenté par un « λ -trapèze» sur l'univers [U, V] (λ représente le niveau d'incertitude qu'il y a à être dans le support):

$$\begin{aligned} v(A) &= 2 * \int_{[U, V]} |\mu_A(x) - \mu_{\underline{A}}(x)| dx / (V - U) \\ &= 2 * [\lambda * (V - U + a - b) + (\alpha + \beta) * (2 * \lambda^2 - 4 * \lambda + 1) / 4 * (1 - \lambda)] / (V - U) \quad \text{si } \lambda \leq 1/2 \\ &= 2 * (1 - \lambda) * [V - U + a - b - (\alpha + \beta) / 2] / (V - U) \quad \text{si } \lambda \geq 1/2 \end{aligned}$$

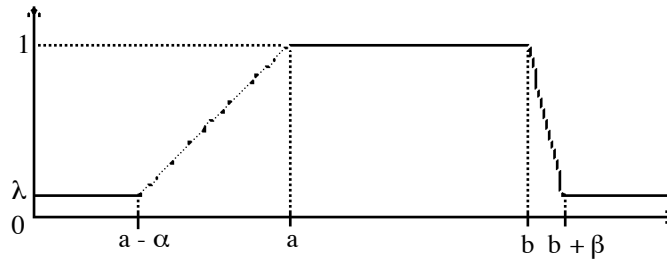


Figure 7 Exemple de lambda-trapèze

Si $\lambda = 0$ (cas de la certitude) $v(A) = (\alpha + \beta) / 2 * (V - U)$ est donc directement proportionnel à l'amplitude des «pentes» du trapèze, il est au maximum égal à 1/2 pour $\alpha + \beta = V - U$ donc $a = b$.

Si $\lambda = 0$ et $\alpha + \beta = 0$ (certitude et précision) alors l'indice de flou est nul.

Si l'ensemble trapézoïdal vérifie $\alpha + \beta = 0$ (non flou, mais incertain) alors dans le cas $\lambda < 1/2$, $v(A) = 2\lambda(1 - (b-a)/(V-U))$, sinon, $v(A) = 2(1 - \lambda)(1 - (b-a)/(V-U))$

Le cas de $v(A) = 1$ est réalisé par l'ensemble (le plus flou possible) sans noyau de fonction d'appartenance constante $\lambda = 0.5$.

Un autre indice a été proposé par $v(A) = 1 - (\sum_{x \in E} |\mu_A(x) - \mu_{\neg A}(x)|) / n$

DEGRÉ DE FLOU [Deluca, Termini 72]

$$d(A) = - \sum_{x \in E} [\mu_A(x) * \log \mu_A(x) + (1 - \mu_A(x)) * \log (1 - \mu_A(x))]$$

Par analogie avec l'entropie (chapitre 5) avec la convention $0 \log 0 = 0$, ce degré mesure bien l'aspect flou car il est minimal et égal à zéro pour les ensembles non flous.

Exercice 1.6

Vérifier que $d(A)$ est minimal de valeur 0 si et seulement si A est un ensemble exact, et maximal égal à 1 pour l'ensemble flou dont la fonction d'appartenance est constante égale à 1/2.

SPÉCIFICITÉ D'UN ENSEMBLE FLOU [Yager 93]

Cette notion a été établie afin de donner un indice de la précision en posant :

$$S(A) = \int_0^{\alpha_{\max}} \frac{d\alpha}{\text{card}(A_\alpha)}$$

Exemple, si $A = \{a / 1, b / 0.7, c / 0.3, d / 0.1\}$, alors :

$$S(A) = 0.1 / 4 + (0.3 - 0.1) / 3 + (0.7 - 0.3) / 2 + (1 - 0.7) / 1 = 0.592$$

On a en général $0 \leq S(A) \leq 1$, si A exact, alors $S(A) = 1 / \text{card}(A)$ et donc :

$S(A) = 1 \Leftrightarrow A$ est un singleton non flou, c'est l'ensemble le plus spécifique possible contenant exactement un élément.

Décroissance de S : $A \supseteq B \Rightarrow S(A) \geq S(B)$, on peut donc convenir que $S(\emptyset)$ est 1.

Exercice 1.7

Pour un ensemble flou trapézoïdal de R défini par le noyau [a, b] et [a - alpha, b + beta] pour support, montrer que sa spécificité est donnée par $\ln [1 + (\alpha + \beta) / (a + b)] / (\alpha + \beta)$.

ENSEMBLES FLOUS CONVEXES

Un ensemble flou est convexe $\Leftrightarrow \forall a, b \forall x \in [a, b] \mu(x) \geq \min(\mu(a), \mu(b))$

La fonction d'appartenance est «quasi-concave» ce qui fait que toutes ses α -coupes sont des intervalles. (Attention ce n'est pas la définition usuelle des fonctions convexes, il n'y a pas de minimum relatif pour μ). Ainsi A et B dans la figure plus haut illustrant l'union et l'intersection, sont convexes mais non leur union.

Exercice 1.8

Montrer que la définition d'ensemble flou convexe sur R^n correspond au fait que pour tout α , l'ensemble A_α est un ensemble convexe de R^n (c'est à dire contient tous ses segments $\forall x, y \in A_\alpha \forall \lambda \in [0, 1]$ on a $\lambda x + (1 - \lambda)y \in A_\alpha$).

INDICE DE CHEVAUCHEMENT

En définissant $ht(A, B) = \sup(\min(\mu_A, \mu_B))$ ce sera la notion très importante de «possibilité que A soit B» ou «compatibilité entre A et B» (chapitre 2) utilisée dans la plupart des systèmes qui doivent confronter une donnée à une référence afin d'apprécier à quel degré la donnée observée A vérifie la donnée de référence B, mais on peut aussi définir le chevauchement par $\text{card}(A \cap B) / \text{card}(A \cup B)$.

INDICE D'INCLUSION

Plusieurs définitions ont été proposées pour le degré d'inclusion de A dans B, par exemple $I(A, B) = \text{card}(A \cap B) / \text{card}(A) = \sum \mu_A \cap B / \sum \mu_B$ (degré d'inclusion de Sanchez) ou plutôt :

$1 - ht(A \cap \neg B) = \inf \max(1 - \mu_A, \mu_B)$ qui est un indice plus fort car égal à 1 uniquement si le support de A est inclu dans le noyau de B. Ce sera la «nécessité que A soit B» qui, cette fois-ci, n'est pas une définition symétrique.

INDICE D'ÉGALITÉ

Si I est un indice d'inclusion, $\min[I(A \subseteq B), I(B \subseteq A)]$ est la traduction floue de «A inclus dans B et réciproquement».

INDICE D'INTERSECTION

$J(A, B) = \text{card}(A \cap B) / \text{card}(A \cup B)$ est l'indice de Jacquart, mais on trouve encore $R(A, B) = ht(A \cap B) / ht(A \cup B)$

Exercice 1.9

Vérifier que pour ces indices, $A = B \Rightarrow J = R = 1$ et que si les supports sont disjoints, $J = R = 0$

PRINCIPE D'EXTENSION DE ZADEH

Ce principe énoncé pour toute relation exacte ϕ entre deux ensembles E et F, permet la généralisation au flou d'un certain nombre de concepts :

Si A est un sous ensemble flou de E, son image B par ϕ sera définie par :
$$\mu_B(y) = \sup\{\min(\mu_\phi(x, y), \mu_A(x)) / x \in E\}$$

En particulier, si ϕ est une application de E dans F, on peut définir un sous-ensemble flou B de F qui sera l'image par ϕ d'un sous-ensemble flou A de E par $\mu_B(\phi(x)) = \mu_A(x)$ dans le cas où ϕ est injective, et si plusieurs éléments de E admettent la même image alors $\mu_B(y) = \sup\{\mu_A(x) / \phi(x) = y\}$, ce qui signifie que chaque fois que y est obtenu, son degré d'appartenance au résultat est le meilleur degré de toutes les façons de l'obtenir et égal à 0 sinon.

Ce principe d'extension est une sorte de «sup-min convolution», il sera d'ailleurs à l'origine de bien des définitions, en particulier du modus-ponens généralisé.

Exercice 1.10

Dans R, on donne l'intervalle flou trapézoïdal A de noyau [1, 2] ayant pour support [0, 5] et $\phi(x) = x^2$, montrer que l'image B de A par ϕ est de noyau [0, 1] et décrit par deux arcs de paraboles entre 1 et 9.

Exercice 1.11

Si $\phi(x) = \sin x$ et A est le nombre flou centré sur 0 de support [-p, p], quel est l'ensemble flou image B? Le représenter pour $p = \pi$.

ARITHMÉTIQUE SUR LES INTERVALLES FLOUS

Sur R, on définit conformément au principe d'extension une somme par :

$$\mu_{A+B}(z) = \max\{\min(\mu_A(x), \mu_B(y)) / x + y = z\}$$

Chaque fois que z est obtenu comme une somme, son degré d'appartenance est le meilleur parmi les décompositions.

La multiplication par $\mu_{A \cdot B}(z) = \max\{\min(\mu_A(x), \mu_B(y)) / xy = z\}$

Comme cas particuliers se pose alors le problème de donner des approximations trapézoïdales à A + B et A.B car bien sûr rien n'indique que A + B l'est.

L'addition de nombres flous sera donnée par :

$(a, b, \alpha, \beta) + (a', b', \alpha', \beta') = (a + a', b + b', \alpha + \alpha', \beta + \beta')$
 Multiplication par un scalaire positif $\lambda : \lambda (a, b, \alpha, \beta) = (\lambda a, \lambda b, \lambda \alpha, \lambda \beta)$
 Par contre il est beaucoup plus difficile de définir la multiplication et la division, voir [Dubois, Prade 79].

Exercice 1.12

$A_1 = \{1/0,6 \ 2/0,8 \ 3/1 \ 4/0,6\}$ et $A_2 = \{0/0,5 \ 1/0,7 \ 2/0,9 \ 3/1 \ 4/0,4\}$.
 Trouver $A_1 + A_2$ et $A_1 * A_2$.

Exercice 1.13

La méthode de définition des opérations arithmétiques par les α -coupes est la suivante, si A et B sont des intervalles flous (donc des sous ensembles flous convexes), et si on note $A_\alpha = [a_\alpha^-, a_\alpha^+]$, alors $A + B$, $A - B$, $A \cdot B$, A/B sont respectivement définis par leurs α -coupes $(A + B)_\alpha = [a_\alpha^- + b_\alpha^-, a_\alpha^+ + b_\alpha^+]$,
 $(A - B)_\alpha = [\min(a_\alpha^- - b_\alpha^-, a_\alpha^+ - b_\alpha^-), \max(a_\alpha^- - b_\alpha^-, a_\alpha^+ - b_\alpha^-)]$,
 $(A \cdot B)_\alpha = [\min(a_\alpha^- b_\alpha^-, a_\alpha^- b_\alpha^+, a_\alpha^+ b_\alpha^-, a_\alpha^+ b_\alpha^+), \max(a_\alpha^- b_\alpha^-, a_\alpha^- b_\alpha^+, a_\alpha^+ b_\alpha^-, a_\alpha^+ b_\alpha^+)]$,
 et $(A / B)_\alpha = [\min(a_\alpha^- / b_\alpha^-, a_\alpha^- / b_\alpha^+, a_\alpha^+ / b_\alpha^-, a_\alpha^+ / b_\alpha^+), \max(a_\alpha^- / b_\alpha^-, a_\alpha^- / b_\alpha^+, a_\alpha^+ / b_\alpha^-, a_\alpha^+ / b_\alpha^+)]$. Montrer pour l'addition, que cette définition coïncide avec celle donnée par le principe d'extension.

LES ENSEMBLES BIFLOUS

[Krassimir Atanassov 86] a étendu la notion d'ensemble A «flou intuitioniste» ou «biflou» dans un ensemble E par la donnée de deux fonctions : l'appartenance μ_A et la non-appartenance ν_A toutes deux dans [0, 1] avec la condition supplémentaire :

$$0 \leq \mu_A + \nu_A \leq 1.$$

On définit alors le complémentaire par $\mu_{\neg A} = \nu_A$ et $\nu_{\neg A} = \mu_A$, l'inclusion grâce à la relation $A \subseteq B \Leftrightarrow \mu_A \leq \mu_B$ et $\nu_B \leq \nu_A$, l'intersection par $\mu_{A \cap B} = \min(\mu_A, \mu_B)$ et enfin $\nu_{A \cap B} = \max(\nu_A, \nu_B)$.

On montre que les propriétés de commutativité, associativité, idempotence, distributivités mutuelles et lois de Morgan sont conservées. Des opérateurs modaux peuvent être définis pour tout ensemble biflou A :

La nécessité $\circ A$ par μ_A et $1 - \mu_A$ et la possibilité $\diamond A$ par $1 - \nu_A$ et ν_A de sorte que : $\circ A \subseteq A \subseteq \diamond A$. Ce sont deux ensembles flous non biflous qui encadrent l'ensemble biflou A.

On peut alors montrer que ce sont des connecteurs idempotents duaux :

$$\circ A = \neg(\diamond \neg A) \text{ et } \diamond A = \neg(\circ \neg A), \text{ établissant des morphismes :}$$

$$\circ(A \cap B) = \circ A \cap \circ B \text{ et } \diamond(A \cup B) = \diamond A \cup \diamond B \text{ et vérifiant la prédominance du nécessaire } \circ \diamond A = \diamond \circ A = \circ A.$$

En définissant la corrélation entre deux ensembles biflous par :

$$c(A, B) = \sum (\mu_A \mu_B + \nu_A \nu_B) \text{ sur } E \text{ supposé fini, on peut définir une «variance intuitioniste» } \text{var}(A) = \sum_{x \in E} (\mu_A^2(x) + \nu_A^2(x))$$

alors, au cas où A est flou, on a bien sûr $\nu_A = 1 - \mu_A$ et alors $\text{var}(A) = \sum_{x \in E} e(\mu(x))$, avec $e(x) = 2x^2 - 2x + 1$ qui est notée l'acuité [O.Onicescu] c'est une «accentuation» valant 1 pour 0 ou 1 et 0 pour 0.5.

On a les propriétés : $\text{var}(A) = \text{var}(\neg A)$, et $\text{cov}(A, B) = 0 \Leftrightarrow [A, B \text{ sont des ensembles exacts}]$, et enfin la «corrélacion» $\text{cov}(A, B) / \sqrt{(\text{var}(A) \cdot \text{var}(B))}$ est égale à 1 $\Leftrightarrow A = B$

Une autre extension des ensembles flous a été faite avec les ensembles «ultra-flous» définis par une fonction d'appartenance dont les valeurs ne sont pas dans [0, 1] mais sont eux-même des sous ensembles flous de [0, 1], [Tong Tong 95].

LES ENSEMBLES BRUTS «ROUGH SETS»

L'idée de ne connaître un ensemble qu'imprécisément entre deux bornes ensemblistes est due à [Pawlak 82]. Dans un univers U, si R est une relation d'équivalence, on rappelle que la classe d'équivalence de x notée R(x) contient tous les y liés à x par R. Les classes d'équivalences forment une partition et on donne les définitions d'approximations basse et haute de X :

$$\text{Si } X \subseteq U \quad R^-(X) = \{x \in U / R(x) \subseteq X\} \text{ et } R^+(X) = \{x \in U / R(x) \cap X \neq \emptyset\}$$

L'ensemble grossier X n'est défini que par le couple $(R^-(X), R^+(X))$. Ce sont respectivement les ensembles d'éléments pouvant certainement (possiblement) être classés dans X. On verra l'analogie avec les probabilités basse et haute (chapitre 2) et avec la logique modale [Farinhas del Cerro, Orłowska 85].

On dit que X est R-définissable si X est l'union de classes d'équivalence. Dans ce cas $X = R^-(X) = R^+(X)$, on définit également deux autres relations d'appartenances faible et forte :

$$x \in^- X \Leftrightarrow x \in R^-(X) \text{ et } x \in^+ X \Leftrightarrow x \in R^+(X)$$

En ce cas, son complémentaire noté $\neg X$ est aussi R-définissable.

Un indice d'imprécision dans [0, 1] est défini dans la cas fini par :

$$\rho_R(X) = 1 - \text{card}(R^-) / \text{card}(R^+).$$

Deux ensembles X, Y sont dits grossièrement égaux vis à vis de R, s'ils ont les mêmes approximations basse et haute. Cela définit une équivalence au sein des ensembles bruts.

Des applications ont été effectuées, en classification médicale (avoir un même attribut définit une relation d'équivalence, des exemples peuvent ainsi être groupés par classes d'équivalence et une observation correspond à un ensemble brut dont on détermine deux bornes, ce qui est certain et ce qui est possible) [Pawlak 90].

Exercice 1.14

Montrer les propriétés suivantes :

$$\begin{array}{lll} R^-(X) \subseteq X \subseteq R^+(X) & R^-(\neg X) = \neg R^+(X) & R^+(\neg X) = \neg R^-(X) \\ R^-(X \cap Y) = R^-(X) \cap R^-(Y) & R^+(X \cup Y) = R^+(X) \cup R^+(Y) & R^-(X) \cup R^-(Y) \subseteq R^-(X \cup Y) \\ R^+(X \cap Y) \subseteq R^+(X) \cap R^+(Y) & R^-(R^-(X)) = R^+(R^-(X)) = R^-(X) & R^+(R^+(X)) = R^-(R^+(X)) = R^+(X) \end{array}$$

R^- et R^+ sont croissantes pour l'inclusion.

On souhaite obtenir une image floue (des niveaux de gris) mais non bruitée à partir d'une image floue. Donnons en premier lieu les définitions élémentaires de la morphologie mathématique non floue [Schmitt, Mattioli 94] :

Etant donné un ensemble E dans \mathbb{R}^2 , et B un «élément structurant» qui peut être un voisinage fermé et connexe de 0 , on peut considérer par exemple $B_0 = \{x / |x| \leq \varepsilon\}$ et $B_x = x + B_0$, le plus simple est de prendre la norme sup pour laquelle B est un carré dans le plan, on définit :

$$\text{Dilatation de } E \text{ par rapport à } B : \text{Dil}_B(E) = \{x / E \cap B_x \neq \emptyset\}$$

$$\text{Erosion de } E : \text{Ero}_B(E) = \{x / B_x \subset E\}$$

Ces deux définitions correspondent à une sorte de vision discrète de l'adhérence et l'intérieur topologiques d'un ensemble. Ce sont d'ailleurs des notions duales car $\text{Ero}_B(E) = \neg \text{Dil}_B(\neg E)$ où « \neg » désigne le complémentaire.

Ouverture de E : $\text{Dil}(\text{Ero}(E))$ qui donne de bons résultats en traitement d'images, le but étant de rectifier des formes et d'obtenir des images lissées où les bruits sont éliminés.

La fermeture est le contraire $\text{Cl}(E) = \text{Ero}(\text{Dil}(E))$, il est aussi possible de les enchaîner empiriquement jusqu'à un «lissage» satisfaisant.

On a toujours $\text{Ouv}(E) \subset E \subset \text{Cl}(E)$ et ces deux opérateurs sont idempotents; un filtre est une opération $(\text{Cl} \circ \text{Ouv}) \circ (\text{Cl} \circ \text{Ouv}) \circ \dots(E)$. Le «chapeau haut de forme» est l'ensemble $E - \text{Ouv}(E)$, quant au squelette de E c'est l'ensemble des centres des boules ouvertes maximales incluses dans E , on peut montrer que si $B(r)$ désigne la boule ouverte de rayon r , c'est :

$$\text{Sq}(E) = \bigcup_{r \geq 0} \bigcap_{\varepsilon \geq 0} [\text{Ero}_{B(r)}(E) - \text{Ouv}_{B(\varepsilon)}\text{Ero}_{B(r)}(E)]$$

Ces définitions peuvent s'étendre au flou [Bloch, Maître 93], si E , maintenant flou, est caractérisé par sa fonction d'appartenance μ , et f est un élément structurant (une définition floue de voisinage) défini par exemple par $f(t) = \min(1, \max(0, 1 - t/a))$ (un cône) ou bien paraboliquement par $f(t) = 1 - t^2/a^2$ où a est une distance à régler en fonction du problème (de l'ordre de 1 à 5 pixels).

$$\text{On pose : } \text{Dil}_\mu(x) = \sup_{y \in E} \{T(\mu_E(y), f(d(x, y)))\}$$

où T est une t -norme et S sa conorme associée (chapitre 3). Voir [Bloch, Maître 95] pour une étude comparative suivant les t -normes et l'élément structurant et [Bloch, Pellot 96] pour une application à 3 dimensions.

De même, l'érosion est étendue au flou :

$$\text{Ero}_\mu(x) = \inf_{y \in E} \{S(\mu_E(y), 1 - f(d(x, y)))\}$$

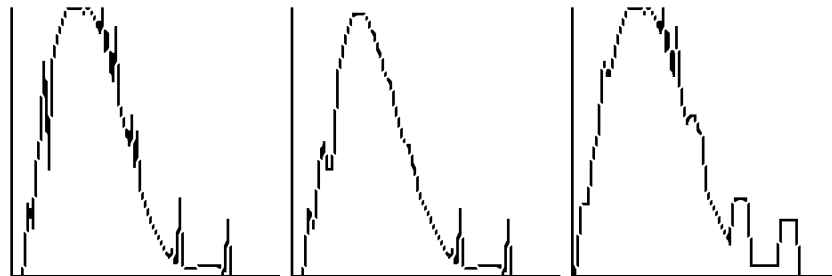


Figure 1.8 Soit la fonction d'appartenance sur $[0, 1]$ donnée dans $[0, 1]$ par la troncature de $\mu(x) = \sin(\pi x) + \sin(2\pi x)/2 + \sin(3\pi x)/3 - 0.4$ sur laquelle on a opéré une vingtaine de modifications aléatoires (première figure à gauche).

Pour l'élément structurant $f(t) = \min(1, \max(0, 1 - t/a))$ avec $a = 0.05$ on obtient successivement l'érosion au milieu et la dilatation à droite.

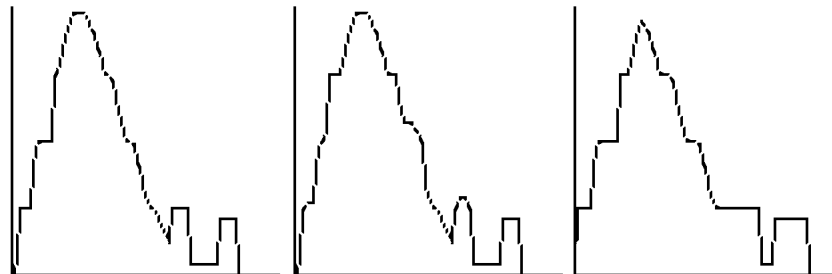


Figure 1.9 Ouverture, fermeture et l'ouverture de l'ouverture sur le même exemple.

1.2 Relations floues

PRODUIT CARTÉSIEN ET RELATIONS FLOUES

Une relation floue n'est autre qu'un sous-ensemble flou d'un produit cartésien, elle est donc caractérisée par une fonction $E^2 \rightarrow [0,1]$.

Rappelons que le produit est défini par : $\mu_{A*B}(x, y) = \min[\mu_A(x), \mu_B(y)]$

Si R est un ensemble flou de E*F (une relation floue) sa projection sur E est définie par $\mu_{\pi(R, E)} = \sup\{\mu_R(x, y) / y \in F\}$ il est clair que si $A = \pi(R, E)$ et $B = \pi(R, F)$, alors $\mu_R \leq \mu_{A*B}$.

Plus généralement si X_1, X_2, \dots sont des ensembles ayant des parties floues A_1, A_2, A_3, \dots on définit le produit cartésien flou $A = A_1 * A_2 * A_3 * \dots$ par la fonction d'appartenance $\mu_A(x_1, x_2, x_3, \dots) = \inf(\mu_{A_1}(x_1), \mu_{A_2}(x_2), \dots)$, ceci peut être une première façon de définir une relation entre les X_1, X_2, \dots

Prenons par exemple $X_1 = \{\text{jaune, bleu}\}$ et $X_2 = \{\text{rond, long}\}$, on pose les ensembles flous :

$A_1 = \{j / 0.8, b / 0.2\}$ et $A_2 = \{r / 0.4, l / 0.6\}$ alors, le produit cartésien :

$A_1 * A_2 = \{(j, r) / 0.4, (j, l) / 0.6, (b, r) / 0.2, (b, l) / 0.2\}$ peut définir une relation floue entre la couleur et la forme dans laquelle «jaune» et «long» sont les attributs les plus liés.

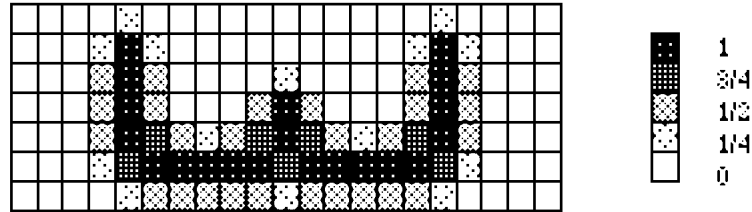


Figure 1.15 Dans le cas discret, une relation binaire peut être définie par un simple schéma.

Exemple de relation binaire définie directement par une formule :

$$\begin{aligned} \mu_{<}(x, y) &= 0 \text{ si } y - x \leq 2 \\ &= (y - x) / 2 - 1 \text{ si } 2 \leq y - x \leq 4 \\ &= 1 \text{ si } y - x \geq 4 \end{aligned}$$

C'est une définition trapézoïdale de «y plutôt supérieur à x».

Exercice 1.16

Montrer que la relation suivante est de noyau vide, quel est son support, sa hauteur ?

$$\mu_{<<}(x, y) = \begin{cases} \text{si } y \leq x \text{ alors } 0 \\ \text{sinon } \frac{1}{1 + \frac{1}{(x-y)^2}} \end{cases}$$

C'est une définition continuellement dérivable de «x très inférieur à y». En regardant la définition de la transitivité ci-dessous, montrer que « est transitive. Soit maintenant la relation binaire définie par la formule $\mu_{\approx}(x, y) = 1 / (1 + |x - y|)$ donnant une version de «approximativement égal». Quel est son noyau et son support ?

COMPOSITIONS DE RELATIONS FLOUES

Il existe plusieurs façons de définir la composition de deux relations floues, la plus employée est la définition «max-min» conforme au principe d'extension :

$$\mu_{S \circ R}(x, z) = \sup_y (\min(\mu_R(x, y), \mu_S(y, z))) \text{ elle est associative.}$$

En particulier si A est une relation unaire et R est binaire on peut définir de la même façon : $\mu_{R \circ A}(v) = \max_u (\min(\mu_A(u), \mu_R(u, v)))$

Les définitions relatives aux relations binaires exactes peuvent se généraliser aisément au flou :

Symétrie : $\forall x \forall y \mu_R(x, y) = \mu_R(y, x)$

Antisymétrie : $\forall x \forall y \mu_R(x, y) > 0 \Rightarrow \mu_R(y, x) = 0$

Réflexivité : $\forall x \mu_R(x, x) = 1$

Antiréflexivité : $\forall x \mu_R(x, x) = 0$

Transitivité : $\forall x \forall y \forall z \mu_R(x, z) \geq \max_y (\min(\mu_R(x, y), \mu_R(y, z)))$ soit :

$$\mu_R \geq \mu_R \circ R \text{ ce qui permet parfois de vérifier plus rapidement la transitivité.}$$

On peut montrer les propriétés suivantes :

R symétrique $\Leftrightarrow \forall \alpha \in [0, 1]$ La α -coupe de R est symétrique

R transitive $\Leftrightarrow \forall \alpha \in [0, 1]$ La α -coupe de R est transitive

FERMETURE TRANSITIVE D'UNE RELATION

C'est la relation $R^* = R \cup R^2 \cup R^3 \cup \dots$ (les exposants étant pris au sens de la composition).

L'algorithme de Floyd étendu à la construction de R^* est le suivant :

Sur $E = \{x_1, \dots, x_n\}$ le graphe de R^* est initialisé par celui de R et représenté par la matrice formée par tous les degrés de R, c'est à dire la matrice des coefficients $(\mu_R(x_i, x_j))_{ij}$.

Pour k = 1 à n

 Pour i = 1 à n

 Pour j = 1 à n faire

$$R^*(x_i, x_j) \leftarrow \max(R^*(x_i, x_j), \min(R^*(x_i, x_k), R^*(x_k, x_j))) \quad (\text{Attention à l'ordre des boucles})$$

Exercice 1.17

Y a-t-il transitivité pour la relation définie dans {a, b, c, d} ?

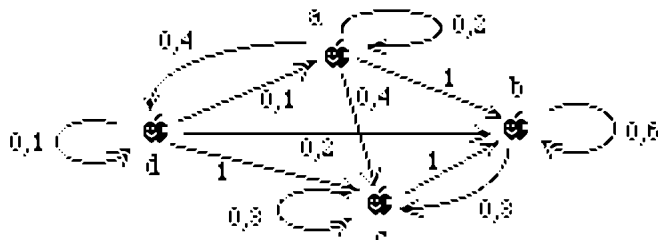


Figure 1.11 Exemple de relation binaire floue pour un ensemble de 4 éléments.

Exercice 1.18

«x transmet facilement un message à y» est une relation définie par les données suivantes : $\mu(x_1, x_1) = 0.2$ $\mu(x_1, x_2) = \mu(x_3, x_2) = 1$ $\mu(x_2, x_1) = \mu(x_3, x_1) = 0$ $\mu(x_2, x_2) = 0.6$ $\mu(x_2, x_3) = \mu(x_3, x_3) = 0.3$ $\mu(x_1, x_3) = 0.4$ faire le graphe et montrer que la relation est transitive.

Construire la matrice M de la relation, effectuer le produit matriciel M.M avec les opérations min et max et vérifier que $M \geq M^2$.

Réponse :
$$\begin{pmatrix} 0.2 & 1 & 0.4 \\ 0 & 0.6 & 0.3 \\ 0 & 1 & 0.3 \end{pmatrix} \geq \begin{pmatrix} 0.2 & 0.6 & 0.3 \\ 0 & 0.6 & 0.3 \\ 0 & 0.6 & 0.3 \end{pmatrix}$$

Exercice 1.19

Montrer que la relation $\mu_{\approx}(x, y) = \exp(-k*(x-y)^2)$ est réflexive et symétrique mais non transitive, son support est \mathbb{R}^2 et son noyau $\{y = x\}$.

RESSEMBLANCES

Les relations $\mu_{\approx 1}(x, y) = \exp(-k*(x-y)^2)$ et $\mu_{\approx 2}(x, y) = 1 / (1 + |x - y|)$ (d'analogie) vues plus haut sont réflexives et symétriques. Toute relation binaire réflexive et symétrique est dite relation de ressemblance. On dit alors que $\mu(x, y)$ est le degré de ressemblance entre x et y.

ORDRES FLOUS

Ce sont les relations floues réflexives, transitives et antisymétriques. Par exemple $p(A > B) = \sup (x \geq y) \min (\mu_A(x), \mu_B(y))$, pour deux nombres flous triangulaires, est le complément de valeur maximale de $A \cap B$.

Exemple : $\mu(x, y) =$ si $y < x$ alors 0 sinon $\exp(-x)$, est une relation transitive dans les réels positifs \mathbb{R}^+ , exprimant que x est petit indépendamment de y.

Exercice 1.20

Montrer que la relation binaire R définie sur {a, b, c, d} par le tableau ci-dessous est un ordre total. En tracer le graphe.

R +	a	b	c	d
a	0.7	0.6	0.8	0.8
b	0	1	0	0.2
c	0	0.6	0	0.4
d	0	0	0	0.7

SIMILARITÉ OU SIMILITUDE

La définition suivante élargit celle de relation d'équivalence exacte :

$$\begin{aligned} &R \text{ est une relation binaire de similarité} \\ \Leftrightarrow &R \text{ relation réflexive, symétrique et transitive} \\ \Leftrightarrow &\forall \alpha \in [0, 1] \text{ la } \alpha\text{-coupe de } R \text{ est une équivalence.} \end{aligned}$$

Si R est une relation de similarité ou la fermeture transitive d'une ressemblance, en posant : $d(x, y) = 1 - \mu_R(x, y)$ on peut vérifier que $d(x, y) \leq \max (d(x, z), d(y, z))$ donc a fortiori l'inégalité triangulaire. Comme d est symétrique, pour que d définisse une distance (bornée), il suffit donc que $\mu_R(x, y) = 1$ n'ait lieu que pour $x = y$.

CLASSE DE SIMILARITÉ

Pour x, la «classe» C(x) de x, est un ensemble flou défini par sa fonction d'appartenance $\mu_{C_x}(y) = \mu_R(x, y)$ et on peut remarquer que :

$$\begin{aligned} &\text{Si } \mu_R(x, y) = 1 \text{ alors } C(x) = C(y) \\ &\text{Si } \mu_R(x, y) = 0 \text{ alors } C(x) \cap C(y) = \emptyset \end{aligned}$$

Exercice 1.21

Si R est réflexive, symétrique et $\mu_R(a, b) = 1, \mu_R(a, c) = \mu_R(b, c) = 0.8, \mu_R(d, e) = 0.6, \mu_R(a, d) = \mu(a, e) = \mu_R(b, d) = \mu_R(b, e) = \mu_R(c, d) = \mu_R(c, e) = 0.5$, les autres étant 0.2, établir les classes d'équivalence des coupes $R_1, R_{0.8}, R_{0.6}, R_{0.5}$ et $R_{0.2}$ placés en arbre.

Exercice 1.22

Montrer que la relation binaire réflexive et symétrique R définie sur {a, b, c, d} par $\mu(a, b) = \mu(b, c) = \mu(b, d) = 0.3, \mu(a, d) = \mu(c, d) = 0.5$ et $\mu(a, c) = 0.9$ est transitive. Quelles sont les classes de similarité pour les coupes de R aux niveaux d'appartenance 1 puis 0.9, puis 0.5 puis 0.3 ? Construire l'arbre de ces différentes partitions.

DISSIMILITUDE

Une dissimilitude est une relation contraire d'une similarité $\mu_R(x,y) = 1 - \mu_{\neg R}(x,y)$ où $\neg R$ est réflexive, symétrique et transitive, alors R est antiréflexive, symétrique et vérifie $\mu_R(x, z) \leq \min_y (\max(\mu_R(x, y), \mu_R(y, z)))$

Exercice 1.23

Pour quelles valeurs de k, la relation binaire suivante μ est-elle une dissimilitude ?

$$\begin{aligned} \mu(x, y) &= 0 \text{ si } x = y \\ &= 1 - e^{-k \cdot (y+1)} \text{ si } y < x \\ &= 1 - e^{-k \cdot (x+1)} \text{ si } y > x \end{aligned}$$

Exercice 1.24

Montrer que dans {A, B, C, D, E}, la première relation est une similitude, et la seconde, une dissimilitude.

	A	B	C	D	E
A	1	0.8	0.7	1	0.9
B	0.8	1	0.7	0.8	0.8
C	0.7	0.7	1	0.7	0.7
D	1	0.8	0.7	1	0.9
E	0.9	0.8	0.7	0.9	1

	A	B	C	D	E
A	0	0.2	0.3	0	0.1
B	0.2	0	0.3	0.2	0.2
C	0.3	0.3	0	0.3	0.3
D	0	0.2	0.3	0	0.1
E	0.1	0.2	0.3	0.1	0

PROJECTION ET EXTENSION D'UNE RELATION FLOUE

Si R est une relation binaire entre E et F, c'est-à-dire une partie floue de $E \times F$, sa projection sur E est définie conformément au principe d'extension par $\mu_{\pi(R/E)}(x) = \sup \{ \mu_R(x, y) \mid y \in F \}$.

L'extension (cylindrique) d'un ensemble flou A de E dans un ensemble produit $E \times F$ est la partie floue binaire $\mu_{\text{ext}(A)}(x, y) = \mu_A(x)$ de la sorte on a donc toujours $\text{proj}(\text{ext}(A)) = A$ mais bien évidemment $\text{ext}(\text{proj}(A))$ est différent de A en général.

1.3. Prédicats vagues

FAMILLES DE PRÉDICATS

Un prédicat unaire issu d'une appréciation linguistique courante comme «jeune», «grand», ... est traditionnellement réalisé par un ensemble flou.

Une famille de prédicats est une famille de sous-ensembles flous d'un intervalle de R, censée traduire une hiérarchie comme très petit, petit, moyen, grand, très grand...

Elle est dite normale si chaque prédicat est normalisé (admet 1 pour sommet), elle est dite binaire si pour tout x il existe au plus deux prédicats tels que $\mu(x) \neq 0$, ainsi x pourra être à des degrés divers à la fois petit et moyen, mais ne pourra pas être en même temps trois qualificatifs. Plus généralement :

La famille (A, B, ...) de prédicats est α -binaire
 $\Leftrightarrow \forall x$ il existe au plus deux prédicats A, B tels que $\mu_A(x) > \alpha$ et $\mu_B(x) > \alpha$

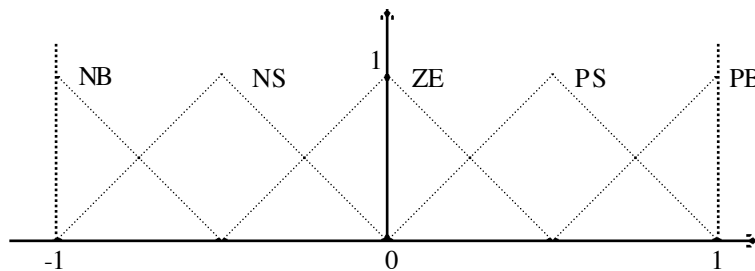


Figure 1.12 Exemple de famille régulière couramment utilisée ramenée à [-1, 1] (NB = grand-négatif, NM = moyen-négatif, NS = petit-négatif, ZE = zéro, PS = petit-positif, moyen-positif, PB = grand-positif)

Il est en tous cas utile de se ramener à l'intervalle $[-1, 1]$ car par des facteurs d'échelle, on peut toujours le faire, et on peut ramener l'étude de prédicats binaires à des prédicats unaires tels que ceux-ci.

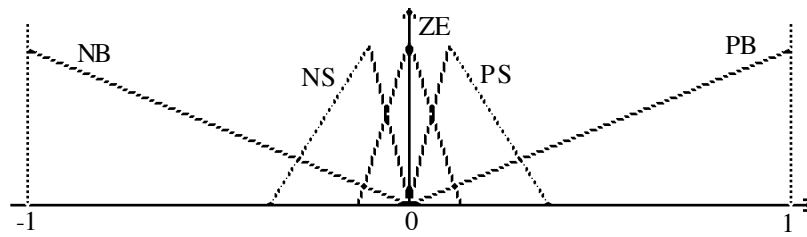


Figure 1.13 Exemple de famille (utilisée dans le suivi d'une ligne) non binaire.

PARTITION FLOUE

Pour un ensemble E , on dit que les sous ensemble flous $(E_i)_{1 \leq i \leq n}$ de E constituent une partition floue si pour tout x , $\sum \mu_{E_i}(x) = 1$ [Bezdek 81](voir chapitre 5).

Exercice 1.25

Définir une famille de prédicats binaires avec des triangles non isocèles tels que la suite des écarts entre les sommets soit arithmétique.

Réponse, on peut prendre $2k + 1$ prédicats avec $x_p = p(p + 1)/k(k + 1)$ pour $0 \leq p \leq k$.

MODIFICATEURS DE PRÉDICATS

Deux concepts sont à distinguer : une modification de degré de vérité dans « X est A » est très vrai, peu vrai ... et une modification de la relation « X est très A » etc ... Quoiqu'on peut parfois les confondre [Akdag, Pacholczyk 90], c'est de cette dernière dont il est question ici, par exemple une concentration avec $\mu_{\text{très vrai}}(x) = \mu^2(x)$ et une dilatation avec $\mu_{\pm \text{vrai}}(x) = \sqrt{\mu(x)}$.

Exercice 1.26

En prenant par exemple $\mu_{\text{vieux}} = \text{si } x \leq 40 \text{ alors } 0 \text{ sinon } 1 / (1 + 25 / (x - 40)^2)$ [Zadeh 78], représenter ces deux transformations.

Si A est un prédicat défini par une fonction d'appartenance μ_A , le prédicat mA s'en déduit [Bouchon-Meunier 88, 90] [Bouchon-Meunier, Yao 92] où t est une application de $[0, 1]$ dans $[0, 1]$ par $\mu_{mA} = t(\mu_A)$ avec :

Modification restrictive $\Leftrightarrow \mu_{mA} \leq \mu_A$ (très, fortement, vraiment ...) c'est à dire un renforcement de A , par exemple en prenant $t(x) = x^2$

Modification expansive $\Leftrightarrow \mu_A \leq \mu_{mA}$ (peu, plutôt, plus ou moins ...) c'est à dire un affaiblissement de A , par exemple $t(x) = \sqrt{x}$

Si t est linéaire on prendra en fait $t(\mu(x)) = \min(1, \max(0, a\mu(x) + b))$, trois exemples de modifications :

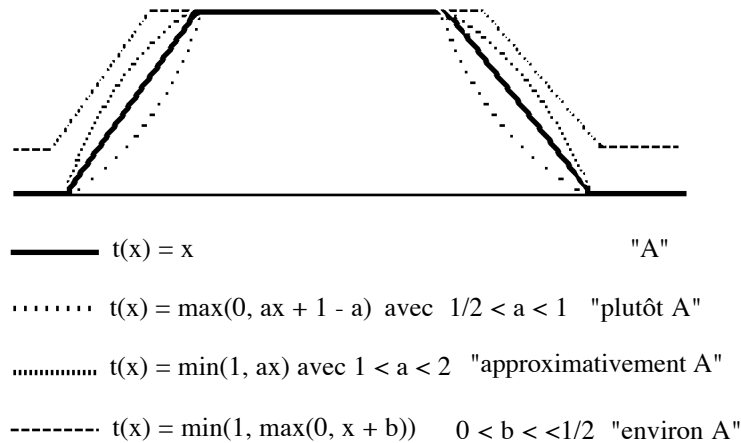


Figure 1.14 Exemple de famille (utilisée dans le suivi d'une ligne) non binaire.

QUANTIFICATEURS FLOUS

Mostowski en 1957 décrit pour la première fois cette notion. Les quantificateurs flous peuvent, le plus aisément, se définir de façon relative par un ensemble flou sur l'intervalle $[0, 1]$. Le quantificateur noté Q , associé à la fonction d'appartenance μ_Q , sera utilisé dans la proposition $Qx A(x)$ afin de lui donner la valeur de vérité $\mu_Q(\sum \mu_A(x) / \sum x)$.

L'APPROCHE DES QUANTIFICATEURS FLOUS PAR DES INTERVALLES DE CONFIANCE

Plusieurs auteurs ont construit des systèmes de déduction fondés sur des appréciations linguistiques correspondant à des intervalles.

Ainsi pour [Dubois, Prade, Godo, Lopez de Mantaras 93] où des quantificateurs flous sont définis grâce à «Aucun» = 0, «Presque aucun» =]0, a], «Peu» = [a, b], «Presque à moitié» = [b, 1 - b], «La plupart» = [1 - b, 1 - a], «Presque tous» = [1 - a, 1[, «Tous» = 1. On peut prendre a = 0,2 et b = 0,4.

Deux ordres partiels sont alors définis sur les intervalles avec [Aucun, Tous] comme minimum de l'ordre spécifique et Tous comme maximum à la fois de l'ordre spécifique et de l'ordre de la certitude. Ce système a été appliqué au raisonnement qualitatif, ainsi il ne peut rien être conclu de : tous les A sont B, la plupart des B sont A, tous les C sont B, à peu près la moitié des B sont C.

EXEMPLES

Si 60 parmi 100 jetons sont de couleur verte (prédicat exact), et si «la plupart» est défini par un trapèze de support [0.5, 1] et de noyau [0.8, 1], «la plupart des jetons sont verts» obtiendra la valeur 1/3. A 70%, cette valeur serait 2/3, et à 80%, elle serait 1. Bien entendu, se pose le problème de la définition du quantificateur, devant traduire une expression linguistique particulièrement vague.

DÉFINITIONS DE QUANTIFICATEURS FLOUS

«Il existe au moins un» se traduit par la fonction μ_{\exists} nulle pour 0, et valant 1 partout ailleurs.

«Pour tous» est la fonction μ_{\forall} nulle sur]0, 1[, et valant 1 en 1.

Plus généralement la quantificateur exact p% peut être rendu par la fonction nulle sauf en p/100.

Le quantificateur de fonction constante 1 «n'importe quelle proportion» nommé «any».

Le quantificateur de fonction constante nulle peut être appelé «aucun».

Les quantificateurs «peu», «moitié», «most», (la plupart) et «beaucoup» sont définis à titre d'exemples par les trapèzes des figures ci-dessous.

COMBINAISON DE QUANTIFICATEURS FLOUS

[Ralescu, Bouchon-Meunier 95] définissent pour deux quantificateurs Q_1 et Q_2 la combinaison de Q_1 par Q_2 sur [0, 1] en convenant de 0/0 = 0 :

$$\mu_{Q_2 * Q_1}(x) = \sup_{y \in [x, 1]} \min [\mu_{Q_2}(x/y), \mu_{Q_1}(y)]$$

Cette opération est illustrée par les figures :

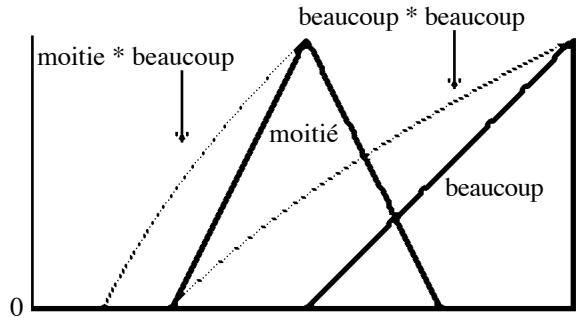


Figure 1.15 Combinaisons de deux quantificateurs relatifs.

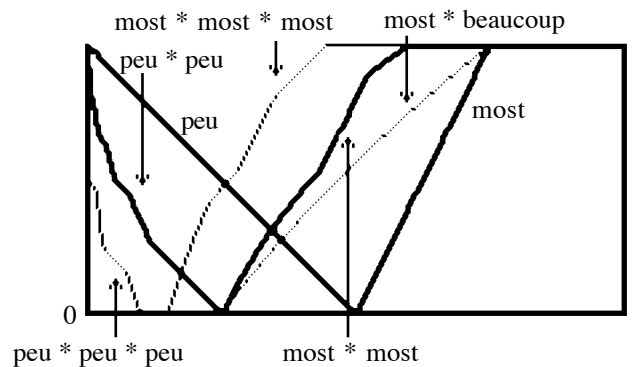


Figure 1.16 Diverses combinaisons avec «peu», «most» et «beaucoup».

Exercice 1.27

Montrer que «any» est neutre à droite $\mu_{Q * any} = \mu_Q$, mais qu'au contraire $\mu_{any * Q}$ est la plus petite fonction décroissante supérieure à μ , et que «tout» est neutre à gauche et à droite.

Montrer que «aucun» de fonction nulle, est absorbant à gauche et à droite :

$$\mu_{aucun * Q} = \mu_Q * \mu_{aucun} = \mu_{aucun}$$

Montrer que $\mu_{p\% * Q}(x) = \mu_Q * \mu_{p\%}(x) = \mu_Q(100x/p)$.

Exercice 1.28

Montrer que la combinaison des quantificateurs est associative.

Exercice 1.29

Si Q est un quantificateur relatif défini par une fonction d'appartenance convexe, il est toujours possible de poser son alpha-coupe $Q_\alpha = [a_\alpha, b_\alpha]$. Si $c \in [0, 1]$, on peut définir la contraction Q^c de Q par la fonction d'appartenance

$$\mu_{Q^c}(x) = \begin{cases} \mu_Q(x) & \text{si } x > c \\ 0 & \text{sinon} \end{cases} \text{ (x/c)}$$

Montrer que $\inf(Q^*Q')_\alpha = \inf(Q^{\inf(Q'\alpha)})_\alpha$ et $\sup(Q^*Q')_\alpha = \sup(Q^{\sup(Q'\alpha)})_\alpha$.

Application, si Q est défini par un trapèze de support $[a, d]$ et de noyau $[b, c]$ et Q' de la même façon, alors le support de Q^*Q' sera $[aa', dd']$ et son noyau $[bb', cc']$ et ses «pentes» seront des portions de paraboles d'axes parallèle à l'axe des abscisses.

1.4. Application aux requêtes floues dans une base de données

Le but de l'interrogation floue de base de données précises est tout à fait clair, on aimerait pouvoir demander par exemple la liste des articles vérifiant des critères vagues tels que un prix modéré, une bonne qualité ...

Il est évidemment beaucoup trop restrictif de demander les articles inférieurs à 100 F, car on risque d'écarter celui qui vaut 105 F mais qui serait très intéressant pour d'autres critères. On voit donc que les problèmes posés sont encore une fois la définition des prédicats tels que "cher", «grande taille», «bonne qualité» ... celle de quantificateurs vagues tels que «la plupart», «presque tous», «peu de»... et surtout la conduite à adopter pour l'agrégation des critères recherchés.

Si le premier problème est résolu par la définition de prédicats flous trapézoïdaux, le second peut également l'être par des ensemble flous dans $[0, 1]$.

Le projet de P.Bosc d'étendre SQL aux requêtes floues en un SQLf [Pivert 91], [Bosc, Pivert 95, 96], doit suivre un certain nombre de points, il doit en effet être le plus conforme possible au SQL qui est le plus répandu des langages d'interrogation de bases de données, et toutes les extensions proposées doivent avoir une sémantique qui correspond aux notions exactes de sélection, jointure, division relationnelle ...

REQUÊTE FONDAMENTALE

En SQL, la requête fondamentale est :

select <objet> from <base relationnelle> where <condition>

qui permet d'obtenir la liste des élément satisfaisant une condition plus ou moins structurée. La question peut être demandée suivant un classement (une partition) par

select <objet> from <base> where <condition 1> group by <objet> having <condition 2>

La requête *select A from R where P* dans laquelle P est une proposition logique où \neg, \wedge, \vee sont interprétés comme en logique floue, devra donc fournir un ensemble flou Q tel que :

$$\mu_Q(a) = \sup_{x \in A} \mu_R(x) = a \quad \min(\mu_R(x), \mu_P(x))$$

Exemple : au lieu d'une requête exacte telle que :

select #emp from Emp where age < 35 and #dep in (select #dep from Dep where budget > 100000)

pour obtenir les jeunes d'un ensemble d'employés dont le département a un budget élevé, on définira les ensembles flous 'jeune' et 'élevé' et la requête :

select #emp from Emp where age = 'jeune' and Emp.#dep = Dep.#dep and budget = 'élevé'

donnera la fonction d'appartenance : $\mu(e) = \min(\mu_{\text{jeune}}(e.\text{age}), \sup\{\min(\mu_{\text{élevé}}(\text{budget}), (\mu_{\text{=}}(d.\text{#dep}, e.\text{#dep}) / d \in \text{Dep})\}$

SÉLECTION

Si R est une relation unaire sur X et ϕ une fonction booléenne définie sur X (c'est à dire la même chose que la définition d'un prédicat unaire), on définit avec «deux points» $R : \phi = \{x / R(x) \text{ et } \phi(x)\}$ qui s'élargit par la définition $\mu_R : \phi = \min(\mu_R, \phi)$ naturellement au flou.

PROJECTION

Si R est une relation binaire entre les ensembles X et Y , on pose (mieux notée par R/A mais conforme à celle donnée plus haut) $R[A] = \{x \in A / \exists y xRy\}$ donc $\mu_{R[A]}(x) = \sup\{\mu_R(x, y) / y \in Y\}$ est la restriction à A du domaine de R .

JOINTURE

Pour deux relations unaires P et Q respectivement sur R et S , deux autres relations unaires (les attributs) A et B respectivement sur R et S , et un opérateur θ (la conjonction * par exemple), on définit la jointure : $P [A \theta B] Q = \{(x, y) / P(x) \text{ et } Q(y) \text{ et } (A(x) \theta B(y))\}$.

Cette opération est facilement étendue au flou par : $\mu_{P [A \theta B] Q}(x, y) = \min(\mu_P(x), \mu_Q(y), (\mu_A(x) \theta \mu_B(y)))$

Exemple, en prenant la notation informatique $x.A$ pour la valeur de l'attribut A sur l'objet x , si θ est un opérateur :

select R.A, S.B from R, S where P(R) and Q(S) and R.C \theta S.D

va délivrer une relation floue telle que : $\mu(a, b) = \sup\{\min(\mu_P(x), \mu_Q(y), \mu_\theta(x.C, y.D)) / x \in R \text{ et } y \in S \text{ et } x.A = a \text{ et } y.B = b\}$

EXTENSION DU «IN» :

«a est dans E» peut être étendu par «a est voisin d'un élément de E» avec :

$$\mu_{\text{inf}}(a, E) = \sup\{\min(\mu_{\approx}(a, b), \mu_E(b)) / b \in \text{supp}(E)\}$$

On tombe alors dans le problème du choix de l'approximation, on peut proposer par exemple, pour des valeurs positives non nulles : $\mu_{\approx}(x, y) = \min(x, y) / \max(x, y)$, ce qui permet d'avoir les couples proportionnels au même niveau.

Exemple : *select #emp from Emp where age < 35 and #dep inf (select #dep from Dep where budget > 100000)*

pour obtenir les jeunes d'un ensemble d'employés dont le département a un budget élevé.

inf est choisi de façon à avoir l'équivalence entre

select R. from R, S where P(R) and Q(S) and e.A \approx e.B*

et la question $select * from R where P and A inf (select B from S where Q)$

GENERALISATION DU «COUNT» DE SQL

On prend naturellement $count(E) = \sum_{x \in \text{supp}(E)} \mu_E(x)$

GENERALISATION DU «EXISTS» DE SQL

On pourra définir $\mu_{\text{exists}}(E) = \sup\{\mu_E(x) / x \in \text{supp}(E)\}$ si E est un ensemble flou normalisé.

Exemple : $select * from R where A inf (select B from S)$

et : $select * from R where exists (select * from S where B \approx R.A)$

vont tous deux produire l'ensemble flou de R défini par : $\mu(x) = \sup\{\mu_{\approx}(x.A, y.B) / y \in S\}$

PREDICATS GENERALISES

En SQL, «a < all E» signifie «a inférieur à tous les éléments de E, et «a = any E» signifie a est égal à au moins un élément de E. Ces relations peuvent s'étendre à une relation binaire floue θ et un ensemble flou E, de la façon suivante :

$$\mu_{\theta \text{ any}}(a, E) = \sup \{ \min (\mu_E(x), \mu_{\theta}(a, x)) / x \in \text{supp}(E) \}$$

$$\mu_{\theta \text{ all}}(a, E) = \inf \{ \max (1 - \mu_E(x), \mu_{\theta}(a, x)) / x \in \text{supp}(E) \}$$

Exemple : les employés dont le salaire est nettement supérieur à presque tous ceux des autres employés du même département, serait obtenu par la question :

$select * from Emp E where sal >> almost all (select sal from Emp where age = 'jeune' and \#dep = E.\#dep and \#emp \neq E.\#emp)$

où μ_{almost} et $\mu_{>>}$ sont définis préalablement par l'utilisateur.

On a l'équivalence entre les questions

$select * from R where A \theta all (select B from S where P)$

et : $select * from R where not exists (select B from S where P and not (A \theta B)).$

LA DIVISION RELATIONNELLE

Une relation exacte dans une base de donnée, est un ensemble de n-uplets, soit par exemple une relation binaire R entre X et Y. Citons l'exemple donné dans [Bosc, Liétard 95] où R est la relation commerciale où une succursale est en relation avec un produit si et seulement elle dispose de ce produit. Soit une deuxième relation S entre Y et Z, par exemple la relation donnant simplement la description (ou le prix) des produits de Y.

Nous notons $R \div S$ l'ensemble des éléments x de X qui sont en relation avec tous les y de Y pour lesquels S met un élément z de Z en relation, formellement :

$$x \in (R \div S) \iff \forall y \forall z (y S z \implies (x R y)) \iff \{y / \exists z (y, z) \in S\} \subseteq \{y / (x, y) \in R\}$$

Pour l'exemple donné, c'est donc simplement l'ensemble des succursales possédant tous les produits ayant une description.

Ce calcul de la division relationnelle est bien sûr long à évaluer, il faut donc si possible réduire la taille de la relation R. Cela se fait par exemple si on demande tous les magasins de Bretagne possédant tous les produits décrits, on évaluera l'expression $(R : b) \div S$ plutôt que $(R \div S) : b$ qui est identique mais plus complexe à évaluer.

DIVISION RELATIONNELLE IMPRECISE

On voudrait maintenant pouvoir avoir la liste des succursales ayant un chiffre d'affaire «important» et distribuant «presque tous» les articles de «bonne qualité».

La généralisation à des relations floues peut se faire en traduisant naturellement les relations par des relations floues. On peut donner une interprétation plus probabiliste que possibiliste, en disant que $\mu_R(x, y) = 0.5$ signifie la plus grande incertitude quant au lien entre x et y.

La généralisation de la division relationnelle passe évidemment par la définition d'un degré d'inclusion entre sous-ensembles flous, soit celle d'une implication (chapitre 3) $\mu_{R \div S}(x) = \inf \{ \mu_S(y, z) \rightarrow \mu_R(x, y) / y \in Y, z \in Z \}$

En prenant l'implication de Kleene-Dienes $(x \rightarrow y) = \max(1 - x, y)$, on montre la cohérence de cette extension avec la division relationnelle exacte.

EXTENSION DE L'INCLUSION «contains_f» pouvant être défini par $\mu_{\text{inc}}(F, G) = \inf \max(1 - \mu_F, \mu_G)$ qui n'est autre que la traduction de l'implication de Kleene-Dienes (chapitre 3).

L'expression de la division relationnelle peut être faite avec imbrication de blocs par :

$select distinct X from R, R1 where (select A from R where X = R1.X contains_f (select B from S)).$

SEMANTIQUE DES QUANTIFICATEURS FLOUS

Des quantificateurs flous (relatifs) Q tels que «la plupart», «peu de», «au moins la moitié» peuvent être définis par des fonctions de [0, 1] dans [0, 1] de sorte que trouver les éléments x tels que «Qy vérifiant S sont R(x, y)» est interprété par l'ensemble flou :

$$\mu(x) = \sup \{ \min(Q(h), \min \{ \mu_S(u) \rightarrow \mu_R(x, u) \}) / C \subseteq \text{supp}(S) \}$$

On utilise l'implication de Kleene $(x \rightarrow y) = \max(1 - x, y)$, et h est la proportion d'éléments vérifiant S dans C (en rappelant que $\text{card}(A) = \sum \mu_A(u)$: $h = \sum_{u \in C} \mu_S(u) / \sum \mu_S(u) = \text{card}(S \cap C) / \text{card}(S)$).

Exemple la question de trouver les départements dont la plupart des employés sont jeunes se fera par :

$select \#dep from Emp group by \#dep having most are age = 'jeune'.$

Une extension aux opérateurs OWA (chapitre 2) est même prévue afin de pouvoir demander par exemple «les éléments possédant au moins la moitié des critères possédés par a».

Chapitre 2

Mesures de confiance

Dans ce chapitre, nous présentons différentes approches élaborées en vue de mesurer la confiance que l'on peut accorder à un fait. La théorie des probabilités, s'inscrit en effet maintenant au sein d'un éventail de théories dont la plupart sont moins restrictives. Par ailleurs les probabilités (issues dans la pratique de mesures statistiques) peuvent rendre compte d'un degré de vérité (un degré de prédiction) mais pas de l'incertitude. La théorie de Dempster-Shafer, issue d'une autre approche, rentre également dans ce cadre.

2.1. Mesures de confiance et théorie des possibilités

MESURE DE CONFIANCE

Afin de donner un cadre général aux différentes approches, la première définition de confiance dans un événement, encore appelée mesure non additive ou mesure floue, ne comporte que trois axiomes. C'est la notion la plus générale définie sur un ensemble Ω d'éventualités. En général, nous parlerons d'ensembles finis auquel cas les «événements» sont toutes les parties de Ω , mais au cas où Ω n'est pas fini, les événements doivent former une tribu de parties, c'est à dire un ensemble de parties comprenant la partie pleine (Ω en totalité), stable par complémentation, intersections et unions finies ou dénombrables. Nous verrons plus loin une extension de cette définition à des sous-ensembles flous de l'univers Ω . On peut alors nommer mesure de confiance, toute fonction c d'une tribu de Ω vers $[0, 1]$ ayant (au minimum) les propriétés $c(\emptyset) = 0$ et $c(\Omega) = 1$ et vérifiant :

$$A \subseteq B \Rightarrow c(A) \leq c(B) \text{ et pour toute suite croissante ou décroissante :} \\ A_1 \subseteq A_2 \subseteq A_3 \subseteq \dots \subseteq A_n \dots \Rightarrow \lim c(A_n) = c(\lim(A_n)) \text{ (continuité)}$$

En conséquence on aura :

$$c(A \cup B) \geq \max(c(A), c(B)) \text{ et } c(A \cap B) \leq \min(c(A), c(B))$$

INVERSE DE MÖBIUS D'UNE MESURE

Si c est une mesure sur un univers fini Ω pour chaque partie A de Ω , en désignant par $|A|$ le cardinal de A , on pose m , l'inverse ou «masse» de Möbius de c , la fonction définie par $m(A) = \sum_{B \subseteq A} (-1)^{|A-B|} c(B)$

On aura alors $c(A) = \sum_{B \subseteq A} m(B)$, ce qui signifie que la confiance en A est la somme de toutes les masses de ses parties disjointes ou non.

Démonstration : pour une partie quelconque A , on a :

$$\sum_{B \subseteq A} m(B) = \sum_{B \subseteq A} \sum_{C \subseteq B} (-1)^{|B-C|} c(C) = \sum_{(C \subseteq B \subseteq A)} (-1)^{|B-C|} c(C)$$

La mesure $c(C)$ est comptée positivement dans cette somme chaque fois qu'on peut trouver un sur-ensemble de C (dans un B lui même inclu dans A) formé en rajoutant un nombre pair d'éléments. Or, si la différence $A - C$ comporte k éléments, le nombre de parties de $A - C$ est 2^k dont la moitié sont des parties paires et l'autre moitié des parties impaires, il y a donc annulation du terme en $c(C)$ dans la dernière expression sauf au cas où $C = A$ où alors le seul B que l'on puisse intercaler entre C et A , est $B = A$, et donc $\sum_{B \subseteq A} m(B) = (-1)^0 c(A) = c(A)$.

A l'origine, le résultat de Möbius fut démontré pour une suite entière c qui s'exprime à partir de la suite m par $c(n) = \sum_{d|n} m(d)$, où $|$ désigne la relation de divisibilité.

En posant $\mu(k) = [si (k = 1) \text{ alors } 1 \text{ sinon si } k \text{ se décompose en } p \text{ nombres premiers distincts alors } (-1)^p \text{ sinon } 0]$, m s'exprime à son tour par $m(n) = \sum_{d|n} \mu(n/d) \cdot c(d)$.

MESURE DE POSSIBILITÉ

C'est la solution la plus pessimiste ou la plus prudente [Zadeh 78], pour une confiance, à savoir que l'on prend comme définition supplémentaire :

$$ps(A \cup B) = \max(ps(A), ps(B))$$

donc en notant $\neg A$ l'événement contraire, c'est à dire le complémentaire, et on a en particulier :

$$\max(ps(A), ps(\neg A)) = 1$$

MESURE DE NÉCESSITÉ

C'est une mesure de confiance vérifiant $nc(A \cap B) = \min(nc(a), nc(B))$ sans qu'il soit question d'aucune notion d'indépendance. Chaque fois qu'une mesure de possibilité ps est définie, alors la nouvelle mesure définie par : $nc(A) = 1 - ps(\neg A)$

est une nécessité, et réciproquement si nc est une mesure de nécessité, alors ps est définie par $ps(A) = 1 - nc(\neg A)$ qui est à son tour une possibilité. Ces deux notions sont donc duales.

Ayant l'une ou l'autre, on a donc $\max(ps(A), ps(\neg A)) = 1$, ce qui entraîne les deux propriétés :

$$ps(A) < 1 \Rightarrow nc(A) = 0 \quad 0 < nc(A) \Rightarrow ps(A) = 1$$

Ces deux propriétés sont très importantes, car elles signifient que pour le couple (nc, ps) situé dans $[0, 1]$, l'un est toujours à une extrémité de l'intervalle (0 ou 1) avec bien sûr $nc \leq ps$, et que l'on n'a absolument pas tous les couples possibles de cet intervalle comme cela est le cas pour les mesures suivantes. La théorie des possibilités (et donc nécessités) a été initialement prévue pour tenir compte des informations incomplètes à propos de propositions booléennes c'est à dire soit vraie, soit fausse. Le degré de possibilité n'est donc en rien un degré de vérité mais plutôt un degré de préférence alors que la nécessité mesurera plutôt une priorité.

MESURE DE CRÉDIBILITÉ (FONCTION DE CROYANCE)

C'est une mesure de confiance qui a la propriété supplémentaire :

$$g(A_1 \cup \dots \cup A_n) \geq \sum g(A_i) - \sum_{i < j} g(A_i \cap A_j) + \dots + (-1)^{n+1} g(A_1 \cap \dots \cap A_n)$$

(sur-additivité générale ou encore appelée ∞ -monotonie).

Alors si $A_1 = A$ et $A_2 = \neg A$ on a en particulier : $g(A) + g(\neg A) \leq g(\Omega) \leq 1$

Voir à ce sujet [Smets, Mamdani, Dubois, Prade 88]. Une crédibilité est par ailleurs une mesure de confiance dont la masse de Möbius est non-négative [Chateauneuf, Jaffray 89].

PLAUSIBILITÉ

C'est la notion duale, si cr est une crédibilité, alors $pl(A) = 1 - cr(\neg A)$ est une «plausibilité», mesure sous-additive. Nous verrons comment à partir d'une distribution de masse m , on peut définir pour tout $A \neq \emptyset$ la «croyance» en A (c'est une crédibilité) et la plausibilité par :

$$cr(A) = \sum_{B \subseteq A} m(B) \text{ et } pl(A) = \sum_{A \cap B \neq \emptyset} m(B)$$

On obtient $pl(A) + cr(\neg A) = 1 - m(\emptyset)$, donc 1 si m est normale.

MESURE DE PROBABILITÉ

On retrouve les probabilités comme mesure de confiance pr ayant l'axiome supplémentaire

$$A \cap B = \emptyset \Rightarrow pr(A \cup B) = pr(A) + pr(B)$$

C'est alors une crédibilité avec l'égalité au lieu d'une inégalité, soit la formule de Poincaré :

$$g(\cup_{1 \leq i \leq n} A_i) = \sum_{I \subseteq \{1, \dots, n\} \quad I \neq \emptyset} (-1)^{|I|+1} g(\cap_{j \in I} A_j)$$

Les probabilités, en accordant 0 à \emptyset , ne distinguent pas entre l'ignorance et la contradiction, qui cependant, dans le raisonnement de sens commun, ne recèlent pas la même quantité de savoir.

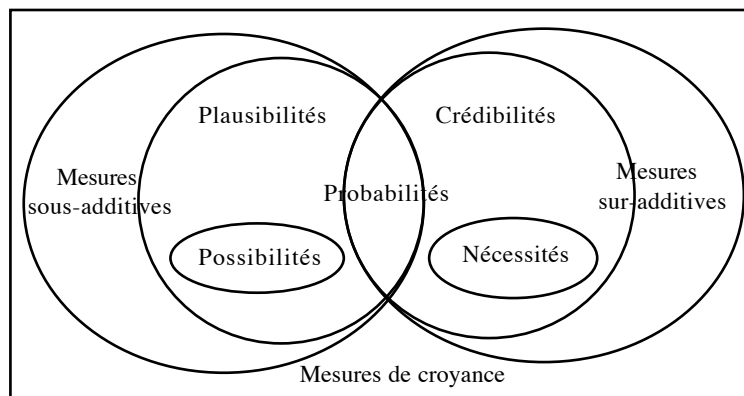


Figure 2.1 Illustration de la hiérarchie des types de mesures.

MESURES DE CROYANCE ET D'INCRÉDULITÉ, L'EXEMPLE DE MYCIN [BUCHANAN 84]

Ces mesures sont introduites empiriquement : MB (belief) qui est une possibilité et $MD(A) = MB(\neg A)$ (disbelief) qui est le complément à 1 d'une possibilité. Toute deux vérifiant respectivement \max et \min pour les disjonctions et conjonctions.

On a $0 < MD(A) \Rightarrow MB(A) = 1$ et $MB(A) < 1 \Rightarrow MD(A) = 1$

Le coefficient de certitude d'un fait est défini par : $cf(A) = MB(A) - MD(A)$ qui est donc dans $[-1, 1]$ alors $cf(A) + cf(\neg A) = 0$, cf est une sorte de degré d'ignorance qui fait correspondre en particulier le vrai à 1, le faux à -1 et l'incertain à 0.

Si $cf = 1$ alors $MB = 1$ et $MD = 0$, et si $cf = -1$ c'est le contraire.

Il revient au même de parler de $v = (1 + cf)/2$ comme un degré de vérité, ou du coefficient $\lambda = pr(\text{évidence} / A) / pr(\text{évidence} / \neg A)$ avec $cf(A) = (\lambda - 1) / (\lambda + 1)$, bijection de $[0, +\infty[$ vers $[-1, 1[$.

Inférences : pour une règle le «cf» de l'ensemble des prémisses est le \min des cf s'ils sont tous supérieurs à 0.2, le \max s'ils sont tous inférieurs à -0.2 et 0 sinon, et le cf de la conclusion est le produit du cf des prémisses par le cf de la règle. Cette option min-produit est la plus courante dans les systèmes experts.

Agrégation : si deux règles aboutissent à la même conclusion avec les coefficients x et y dans $[-1, 1]$, alors :

$$cf(\text{conclusion}) = \begin{cases} \text{si } x, y > 0 \text{ alors } x + y - xy & (\text{on reconnaît la disjonction probabiliste}) \\ \text{si } x, y < 0 \text{ alors } x + y + xy \\ \text{sinon } (x+y) / (1 - \min(|x|, |y|)) \end{cases}$$

DANS LE SYSTÈME PROSPECTOR, connaissant les probabilités conditionnelles, on définit pour $P \Rightarrow Q$, un degré de «vraisemblance ou suffisance» $ds = \text{pr}(P / Q) / \text{pr}(P / \neg Q)$, un «degré de nécessité» $dn = \text{pr}(\neg P / Q) / \text{pr}(\neg P / \neg Q)$, et aussi un degré de «prédiction» $v(Q / P) = \text{pr}(Q / P) / \text{pr}(\neg Q / P)$. Enfin les «probabilités» a priori ne respectent pas la théorie des probabilités car Prospector utilisent le min pour la conjonction et le max pour la disjonction, et il définit en outre une «priorité» ou «cote» ou encore «degré de diagnostic» $\alpha(Q) = \text{pr}(Q) / \text{pr}(\neg Q)$. Le degré de prédiction est infini si $P \Rightarrow Q$, égal à 1 s'il y a une indifférence supposée, et nul si $P \Rightarrow \neg Q$.

DISTRIBUTION DE POSSIBILITÉ, MESURE DE POSSIBILITÉ ET ENSEMBLE FLOU

Une distribution de possibilité sur un univers U est une application π de U dans l'intervalle $[0, 1]$ qui est normalisée, c'est à dire vérifiant $\text{sup}(\pi) = 1$.

Le rapport évident avec les possibilités sur U , est que toute distribution de possibilité permet de définir une mesure de possibilité pour les parties de U simplement par $\text{ps}(A) = \text{sup}\{\pi(x) / x \in A\}$, de plus $\text{nc}(A) = \text{inf}\{1 - \pi(x) / x \notin A\}$ définit la nécessité duale.

Réciproquement, une mesure de possibilité ps sur U permet de définir une distribution de possibilité $\pi(x) = \text{ps}(\{x\})$ pour $x \in U$. Une distribution de possibilité équivaut donc à la donnée d'un ensemble flou normalisé sur U .

REMARQUE : une distribution de possibilité n'a rien à voir avec une distribution de probabilité car la somme de π (somme discrète si U est fini, intégrale de π sur U sinon) n'est pas 1 en général.

Exemple, sur $U = [0, 100]$ on pourrait définir «trentaine» par un intervalle flou A de noyau $[30, 38]$ et de support $[28, 42]$. En ce cas, si X est une variable (linguistique telle que «âge») prenant ses valeurs dans U , « X est A », par exemple «âge est trentaine» est une proposition atomique.

On peut définir la restriction associée à X notée $\text{ps}(X = x / A) = \mu_A(x)$.

Alors si $X = 29$, $\text{ps}(X \text{ est } A) = 0.5$ et si $X = 39$, $\text{ps}(X \text{ est } A) = 0.75$.

Si A est normalisé, et c'est le cas de notre exemple, cela induit une mesure de possibilité conditionnelle pour toute partie B de U par :

$\text{ps}(B) = \text{ps}(X \in B / A) = \text{sup}\{\mu_A(x) / x \in B\}$ qu'il faut lire comme la possibilité de l'événement B , connaissant A .

Concrètement, les systèmes à base de connaissances sont confrontés au problème suivant, étant donné un ensemble flou (normalisé) A (ou donc, ce qui revient au même, à une distribution de possibilité π), et une variable X dont la valeur est elle-même imprécise, mal connue (connue seulement par un ensemble flou A'), comment mesurer la confiance dans le fait « X est A » ? On peut voir par exemple A comme un prédicat «grand» et X assigné à une valeur floue A' (ou une distribution de possibilité π'). On posera la définition de [Zadeh 78] :

$\text{ps}(X \text{ est } A) = \text{ps}(A / A') = \text{sup}\{\min(\pi'(x), \mu_A(x)) / x \in U\}$ noté $\text{sup min}(\mu_{A'}, \mu_A)$ borne supérieure d'une fonction.

On lira «la possibilité d'avoir A sachant que X est A' ».

Cette définition a pour conséquences (pour le même X) que $A = B \Rightarrow \text{ps}(A) \leq \text{ps}(B)$ et (max et min étant des lois binaires mutuellement distributives) que la possibilité de la disjonction est $\text{ps}(A \cup B) = \max(\text{ps}(A), \text{ps}(B))$.

Ce «ps» conditionné par le fait que X est assigné à A' est donc bien une mesure de possibilité sur U .

En résumé, si on a des représentations $\mu_{A'}$ pour X et un prédicat A , la possibilité que X soit A est donnée par la hauteur d'une intersection d'ensemble flous. On peut voir ps comme un degré d'intersection de A' avec A . Pour la nécessité que X soit A , on va utiliser le complément de la possibilité du contraire, soit :

$$\text{nc}(X \text{ est } A) = 1 - \text{ps}(X \text{ est } \neg A) = 1 - \text{sup min}(\mu_{A'}, 1 - \mu_A) = \text{inf max}(1 - \mu_{A'}, \mu_A)$$

Cette nécessité est aussi notée de façon conditionnelle $\text{nc}(A / A')$. C'est un degré d'inclusion de X (plus précisément de sa valeur floue A') dans A car :

$$\text{«}X \text{ satisfait entièrement } A\text{»} \Leftrightarrow \text{nc}(X \text{ est } A) = 1 \Leftrightarrow \text{supp}(A') \subseteq \text{supp}(A)$$

$\text{ps}(X \text{ est } A) = \text{sup min}(\mu_{A'}, \mu_A) = \text{ps}(A, A')$ $\text{nc}(X \text{ est } A) = \text{inf max}(1 - \mu_{A'}, \mu_A)$

REMARQUE

[Dubois, Prade 85] et d'autres auteurs définissent la nécessité $\text{nc}(A / A')$ au contraire par : $1 - \text{ps}(\neg A' \text{ est } A) = 1 - \text{sup min}(1 - \mu_{A'}, \mu_A) = \text{inf max}(\mu_{A'}, 1 - \mu_A)$

C'est alors un degré d'inclusion de A dans A' .

L'une ou l'autre de ces deux options sont constamment utilisées dans les travaux réalisés sur des domaines concrets et sont connues sous le nom de filtrage flou.

Dans les deux cas, dès que X est assigné à un A' non exact, on sort de la théorie des possibilités car on peut avoir $0 < \text{nc} < \text{ps} < 1$ pour le fait « X est A » et non l'inverse comme ci-dessus.

Par contre, si X est exactement connu par une valeur précise, la seconde option vérifie bien les deux axiomes $0 < \text{nc} \Rightarrow \text{ps} = 1$ et $\text{ps} < 1 \Rightarrow \text{nc} = 0$ et donc cette deuxième définition est plus conforme à la théorie des possibilités. C'est cependant, la plupart du temps, la première option qui est adoptée dans les systèmes concrets, c'est pourquoi nous l'avons présenté en premier lieu.

Pour cette première définition, si X a une valeur précise, on vérifiera que $\text{ps} = \text{nc}$ redonne un degré de vérité.

Cette remarque montre qu'il est plus général d'envisager des systèmes où la confiance accordée à un fait est un couple (x, y) pouvant prendre toutes les valeurs en accord avec $0 \leq x \leq y \leq 1$, c'est le cas des crédibilités-plausibilités.

Une autre notion est la compatibilité définie elle-même comme un ensemble flou :

$$\mu_{\text{cp}}(X, A)(y) = \text{si } \mu_X^{-1}(y) = 0 \text{ alors } 0 \text{ sinon } \text{sup}\{\mu_A(x) / y = \mu_X(x)\} \text{ [Zadeh 79].}$$

Une «véritable logique floue» n'aurait pas seulement la présence de prédicats vagues, mais les valeurs de vérité seraient elles mêmes des ensembles flous dans $[0, 1]$. Pour un degré de vérité perçu comme un ensemble flou, voir aussi [Godo 89].

En logique floue les propositions sont structurées à partir des propositions atomiques du type «X est A» où la sémantique leur donne la valeur $\mu_A(x)$ pour chaque valeur précise de la variable X. B.Bouchon a introduit la notion de variable linguistique comme un triplet (X, E, P) où X est une variable prenant ses valeurs dans l'univers U, et P une famille de prédicats flous sur U (petit, grand, ...). Les système de contrôle flou du chapitre 4 mettent en oeuvre par exemple trois variables linguistiques s'ils ont deux entrées et une sortie. La variable linguistique X en elle-même est un attribut tel que âge, température, etc recevant des valeurs : jeune, vieux, chaud, ... définies par des sous-ensembles flous.

EXTENSION DE LA NOTION DE MESURE FLOUE

Soit $F(\Omega)$ l'ensemble de toutes les parties floues de Ω . Une mesure (de confiance) floue peut être définie comme une application c de $F(\Omega)$ vers $[0, 1]$, croissante au regard de l'inclusion et vérifiant $c(\emptyset) = 0$. Il est toujours possible de se ramener à $c(\Omega) = 1$ si c est bornée. Par exemple, la hauteur, le cardinal, l'intégrale ou même le cardinal du support peuvent constituer des mesures floues définies pour chaque sous-ensemble flou de Ω .

Dans le but de comparer des sous-ensembles flous [Rifqi 95] introduit :

Différence sur $F(\Omega)$: une opération de différence doit vérifier pour toutes parties floues A et B :

$$A \subset B \Rightarrow A - B = \emptyset, \quad A - B = A - (A \cap B), \quad (B \subset B') \Rightarrow (B - A \subset B' - A)$$

La différence entre ensembles exacts est généralisée, et par exemple les deux définitions $\mu_{A-B} = \max(0, \mu_A - \mu_B)$ et $\mu_{A-B} = [\text{si } (\mu_B = 0) \text{ alors } \mu_A \text{ sinon } 0]$, peuvent convenir, mais pas celle de $\min(\mu_A, 1 - \mu_B)$.

Mesure de comparaison s entre ensembles flous : une mesure de comparaison s est définie pour toutes parties floues A, B par $s(A, B) = f(c(A \cap B), c(B - A), c(A - B))$ où f est une fonction de $[0, 1]^3$ dans $[0, 1]$ croissante au sens large pour son premier argument. Nous avons un cadre général pour définir des similitudes, et la définition usuelle de «possibilité» $\sup \min(\mu_A, \mu_B)$ en est un exemple.

Mais ce qui est intéressant, est que l'on peut définir une «satisfiabilité» (A satisfait B) dès lors que f ne dépend pas de son troisième argument et que de plus $f(0, v, w) = 0$, et $f(u, 0, w) = 1$ pour $u \neq 0$.

On peut également définir une mesure d'inclusion dès lors que f est indépendante de son second argument et que $f(0, v, w) = 0$ et $f(1, v, 0) = 1$ de sorte que $s(A, A) = 1$.

Ainsi par exemple $\text{inclu}(A, B) = |A \cap B| / |A|$ convient à ces contraintes ou bien : $\text{inclu}(A, B) = \inf(\min(1 - \mu_A + \mu_B, 1))$.

A noter encore l'exemple de $s(A, B) = (2/\pi) \text{Arctan}[(c(A \cap B) / c(A - B))]$ qui est décroissante en $c(A - B)$ et qui ne vaut 1 que si $A = B$.

Une dissimilarité sera au contraire une mesure déterminée par f indépendante de son premier argument, croissante au sens large vis-à-vis de ses autres arguments et vérifiant $f(u, 0, 0) = 0$, par exemple $ds(A, B) = \sup |\mu_A - \mu_B|$.

2.2. La théorie de Dempster-Shafer

Cette théorie [Shafer 76] se fonde sur un partage de l'univers de référence Ω dans lequel on considère une famille F de parties (un corps d'évidence normal, des événements «focaux») dont on est sûr d'une «évidence» m (ce n'est pas une mesure de confiance définie plus haut).

DISTRIBUTION DE MASSE

C'est une fonction m de $P(\Omega)$ dans $[0, 1]$ telle que $[m(A) > 0 \Leftrightarrow A \in F]$ et telle que l'on ait la somme $\sum_{A \subset \Omega} m(A) = 1$

Il faut bien noter que m ne vérifie pas la croissance vis à vis de l'inclusion.

L'ignorance totale serait traduite par une masse vérifiant $m(\Omega) = 1$ et $m(A) = 0$ pour toute partie A non pleine de Ω . Si par exemple, on ne considère que deux éventualités $\Omega = \{a, b\}$ avec $m(\{a\})$ et $m(\{b\})$, alors $m(\{a, b\})$ symbolise l'indifférence à a ou b, et $m(\emptyset)$ symbolise la part de ce qui pourrait être autre chose que a ou b. On ne requiert donc pas $m(\emptyset) = 0$, mais si c'est le cas, m est dite normale ou complète, et alors, on pourra montrer qu'une telle masse définit une famille de probabilités.

Exemple, en termes d'événements, avec deux événements A et B, si les seuls éléments focaux sont B de masse q et $\neg B$ de masse 1 - q, cette masse représente toutes les probabilités vérifiant $\text{pr}(A \text{ et } B) = x$, $\text{pr}(A \text{ et } \neg B) = y$, $\text{pr}(\neg A \text{ et } B) = z$, $\text{pr}(\neg A \text{ et } \neg B) = t$ avec $x + z = q$ et $y + t = 1 - q$ et donc $x + y + z + t = 1$.

COMPLÉMENT D'UNE MASSE Si m est définie sur E par $m(A_i) = m_i$, les éléments focaux du complément m' sont les compléments des A_i et $m'(\neg A_i) = m_i$.

RELATION DE RESTRICTION ENTRE DEUX MASSES

On dit que m' est une restriction de m si, sur un élément focal F de m, il existe deux parties distinctes A, B de F, focales pour m et telles que $m'(A) = m(A) + x$, $m'(B) = m(B) + y$, $m'(F) = m(F) - x - y$, m' étant confondue avec m par ailleurs.

En définissant $m' \leq m$ comme la clôture transitive de cette relation, on peut montrer que cela signifie que la famille des probabilités issues de m' est incluse dans celle issue de m [Baldwin, Martin 94].

Deux masses m_1 et m_2 sont dites orthogonales (incomparables et noté $m_1 \perp m_2$) si, ni m_1 n'est restriction de m_2 , ni l'inverse.

MASSE ET PROBABILITÉS ASSOCIÉES À UN ENSEMBLE FLOU

Etant donné un ensemble flou discret défini par les valeurs $\mu_i = \mu(x_i)$ que l'on va supposer ordonnées de façon décroissante :

$1 \geq \mu_1 \geq \mu_2 \geq \mu_3 \geq \dots \geq \mu_n$ en posant alors $m(\emptyset) = 1 - \mu_1$ et $m(\{x_1, \dots, x_i\}) = \mu_i - \mu_{i+1}$ convenant que $\mu_{n+1} = 0$, on définit une masse sur l'ensemble des parties décrites.

Si l'ensemble flou est normalisé, alors la masse est normale, sinon une normalisation est toujours possible en remplaçant μ par $\mu^* = \mu / \mu_1$.

Exemple, si on dispose de l'ensemble flou défini par : $\mu(a) = 1, \mu(b) = 0.7, \mu(c) = 0.5, \mu(d) = 0.1$, on aura $m\{a\} = 0.3, m\{a, b\} = 0.2, m\{a, b, c\} = 0.4$ et $m\{a, b, c, d\} = 0.1$.

Toutes les probabilités pr associées à cette masse vérifieront $0.3 \leq pr(a) \leq 1, 0 \leq pr(b) \leq 0.7, 0 \leq pr(c) \leq 0.5, 0 \leq pr(d) \leq 0.1$ car $pr(a) + pr(b) + pr(c) + pr(d) = 1$, enfin on a $pr(b) + pr(c) + pr(d) \leq 0.7, pr(c) + pr(d) \leq 0.5$ et $pr(d) \leq 0.1$.

BORNE INFÉRIEURE DE DEUX MASSES

Pour deux masses m_1 et m_2 , la borne inférieure $m_1 \wedge m_2$ est la masse associée à l'intersection des deux familles de probabilités. Si les éléments focaux de m_1 sont les M_i et ceux de m_2 les N_j pour un nombre fini d'indices i et j , alors les éléments focaux de $m_1 \wedge m_2$ sont toutes les intersections $L_{ij} = M_i \cap N_j$ avec les masses l_{ij} de telle sorte que $\sum_j l_{ij} = m_1(M_i)$ et $\sum_i l_{ij} = m_2(N_j)$. En ce cas si L est un élément focal de $m_1 \wedge m_2$, il peut avoir été obtenu de plusieurs façons et :

$$m_1 \wedge m_2(L) = \sum_{L_{ij}=L} m(L_{ij}).$$

Exemple 1 : soient deux masses m et m' issues des ensembles flous μ défini par $\{a/1, b/0.8, c/0.3\}$ et μ' défini par $\{a/1, b/0.5, c/0.4\}$, les deux masses m et m' ont ainsi leurs valeurs 0.2, 0.5, 0.3 et 0.5, 0.1, 0.4 respectivement sur $\{a\}, \{a, b\}, \{a, b, c\}$ comme indiqués à l'extérieur du tableau de gauche. Pour calculer la distribution de masse de $m_1 \wedge m_2$ on commence par l'élément focal $\{a, b, c\}$ qui ne peut recevoir que la valeur 0.3 (dernière case en bas à droite). Puis, pour $\{a, b\}$, ce sont les 3 cases encadrant la précédente. En essayant de maximiser le nombre de zéros, à partir des sommes marginales, on peut remplir le tableau et donc donner 0.2 pour $\{a, b\}$ et enfin 0.5 pour les 5 cases restantes correspondant à $\{a\}$.

De $m_1 \wedge m_2$, on peut déduire ici un ensemble flou $\{a/1, b/0.5, c/0.3\}$ qui n'est autre que $\mu \cap \mu'$.

	$\{a\}$	$\{a, b\}$	$\{a, b, c\}$	
$\{a\}$	0.2	0	0	0.2
$\{a, b\}$	0.3	0.1	0.1	0.5
$\{a, b, c\}$	0	0	0.3	0.3
	0.5	0.1	0.4	
		m'		

	$\{a\}$	$\{a, c\}$	$\{a, b, c\}$	
$\{a\}$	0.2	0	0	0.2
$\{a, b\}$	0.1	0.2	0.2	0.5
$\{a, b, c\}$	0	0.3	0	0.3
	0.3	0.5	0.2	
		m'		

Exemple 2 : Soient les deux ensembles flous μ et μ' définis sur $\{a, b, c\}$ par $\{a/1, b/0.8, c/0.3\}$ et $\{a/1, b/0.2, c/0.7\}$.

Cette fois les masses m et m' qui en sont déduites et dont les valeurs figurent comme sommes marginales du tableau de droite sont $m\{a\} = 0.2, m\{a, b\} = 0.5, m\{a, b, c\} = 0.3$, et $m'\{a\} = 0.3, m'\{a, c\} = 0.5, m'\{a, b, c\} = 0.2$.

Les valeurs de $m_1 \wedge m_2$ sont calculées à partir du tableau et sont $\{a, b, c\} 0, \{a, b\} 0.2, \{a, c\} 0.3$, et sur $\{a\}$ les 6 autres cases soit 0.5, mais cette masse ne correspond à aucun ensemble flou. En revanche la solution $m_0\{a\} = 0.7, m_0\{a, c\} = 0.1, m_0\{a, b, c\} = 0.2$ convient aux sommes et correspond à l'ensemble flou $\{a/1, c/0.3, b/0.2\}$, qui est $\mu \cap \mu'$, mais $m_0 < m_1 \wedge m_2$.

CONFIANCE ACCORDÉE AU FAIT «X EST A»

Lorsque A est un ensemble flou (un prédicat tel que «grand» ...) et X une variable (telle que «untel») assignée à un ensemble flou A' (une donnée imprécise sur sa taille par exemple), une solution étudiée dans [Baldwin, Martin 94], [Baldwin, Lawry, Martin 96] est de calculer la crédibilité et la plausibilité de «X est A» à partir d'une masse définie sur l'ensemble $L = \{f, i, t\}$ des trois valeurs de vérité faux, indéterminé et vrai.

Supposons X à valeurs dans l'univers U où A et A' conduisent aux distributions de masse m et m' dont les éléments focaux sont respectivement notés M_i et N_j . On construit la matrice à valeurs dans $\{t, i, f\}$ en donnant la valeur vrai si $N_j \subseteq M_i$, faux si $N_j \cap M_i = \emptyset$ et indéterminé dans les autres cas.

	$\{b\}$	$\{b, c\}$	$\{a, b, c\}$	$\{a, b, c, d\}$	
$\{a\}$	f 0.09	f 0.15	i 0.03	i 0.03	0.3
$\{a, b\}$	t 0.15	i 0.25	i 0.05	i 0.05	0.5
$\{a, b, c\}$	t 0.06	t 0.1	t 0.02	i 0.02	0.2
	0.3	0.5	0.1	0.1	
		valeurs de m' pour A'			

m pour A

La crédibilité de «X est A» lorsque X est assigné à A' , sera la masse totale attribuée à l'élément t, soit ici 0.33, la plausibilité de «X est A» sera $1 - cr(X \text{ est } \neg A) = 1 - \text{masse}(f) = 1 - 0.24 = 0.76$. La masse 0.43 de i est ce qui reste, elle mesure l'écart entre la plausibilité 0.76 et la crédibilité 0.33. Si A n'est pas normalisé, le calcul est conduit de la même façon, mais avec des $M_i \neq \emptyset$.

Remarque, en prenant les définitions «possibilistes» vues plus haut, pour le même exemple on obtient : $ps(X \text{ est } A) = \sup \min(\mu_A, \mu_{A'}) = 0.7$ (valeur voisine de 0.76), et $nc(X \text{ est } A) = 1 - ps(X \text{ est } \neg A) = 0.3$ qui est aussi voisine de 0.33, mais quel sens donner à ce couple ?

LIEN AVEC LES PROBABILITÉS

Pour la même proposition «X est A» où X est connu par un ensemble flou A' qu'il faut donc confronter à l'ensemble flou A , en posant: $pr(X \text{ est } A) = pr(A / A') = \sum m_i.n_j / |M_i \cap N_j| / |N_j|$

La fraction figurant dans chaque terme est donc égale à 1 pour les $N_j \subseteq M_i$, donc $cr \leq pr$, elle est nulle pour ceux qui satisfont $N_j \cap M_i = \emptyset$, pour les autres il s'agit d'une fraction de la masse de l'élément i , et on a bien $pr \leq pl$, ce calcul donnant bien une probabilité compatible avec la crédibilité.

Dans le cas de l'exemple ci-dessus $pr(X \text{ est } A) = 0 + 0 + 0.03 / 3 + 0.03 / 4 + 0.15 + 0.25 / 2 + 0.05 (2 / 3) + 0.5 / 2 + 0.06 + 0.1 + 0.02 + 0.02 (3 / 4) = 0.53908$

Pour tout $A \neq \emptyset$, on définit la «croyance» en A (c'est une crédibilité) par :

$$cr(A) = \sum_{B \subseteq A} m(B) \text{ c'est naturellement la part d'évidence que contient A, c'est une crédibilité. En termes de logique ce serait } cr(q) = \sum_{p \Rightarrow q} m(p)$$

On définit également : $pl(A) = \sum_{A \cap B \neq \emptyset} m(B)$ donne l'ensemble des pièces d'évidence rendant une occurrence de A possible, on peut l'exprimer. $pl(q) = \sum \{m(p) / p \text{ n'entraîne pas } \neg q\}$ et c'est une plausibilité.

$$cr(A) = \sum_{B \subseteq A} m(B) \text{ et } pl(A) = \sum_{A \cap B \neq \emptyset} m(B)$$

On obtient $pl(A) + cr(\neg A) = 1 - m(\emptyset)$, donc 1 si m est normale.

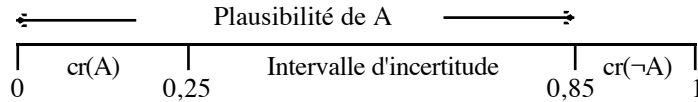


Figure 2.2 Illustration d'un exemple où $cr(A) = 0.25$ et $cr(\neg A) = 1.15$.

Exercice 2.1

Montrer la sur-additivité $cr(A \text{ ou } B) \geq cr(A) + cr(B) - cr(A \text{ et } B)$ (faire un dessin)

et la sub-additivité $pl(A \text{ et } B) \leq pl(A) + pl(B) - pl(A \text{ ou } B)$.

Ainsi, il est possible d'avoir $cr(A) = cr(\neg A) = 0$ et $pl(A) = pl(\neg A) = 1$, de plus on a clairement les inégalités : $0 \leq cr(A) \leq pl(A) \leq 1$ L'intervalle $[cr(A), pl(A)]$ peut être vu comme l'encadrement d'une probabilité mal connue, c'est pourquoi ces bornes sont souvent appelées probabilités basse et haute.

Chaque fait focal $m(A) \neq 0$, est incompatible avec tout fait qu'il n'entraîne pas.

Inversement, si cr est une crédibilité, et si on pose $m(A) = \sum_{B \subseteq A} (-1)^{\text{card}(A-B)} cr(B)$, on montre que cr est bien la crédibilité définie à partir de cette masse m.

Théorème Si $cr = pl$ alors cette mesure commune est une probabilité

En effet si B est focal et coupe A, alors $B \subseteq A$ d'après les définitions de cr et pl. On suppose alors que les événements ne sont pas toutes les parties de E mais une tribu quelconque, la sur-additivité et la sous-additivité entraînent la propriété supplémentaire des probabilités. On peut même montrer que ceci est équivalent au fait que tous les faits focaux sont «premiers» c'est à dire entraînés par rien d'autre qu'eux-mêmes et le faux.

Théorème Une crédibilité sur un ensemble fini, est une probabilité si et seulement si les éléments focaux sont seulement des singletons.

Si cr est une probabilité, alors $cr(\{x_1, x_2, \dots, x_n\}) = \sum cr(\{x_i\}) = \sum m(x_i)$ et d'après la définition de cr, toute autre partie qu'un singleton, incluse dans A ne peut avoir qu'une masse nulle.

Inversement $\sum m(x) = 1$, alors si A, B parties disjointes, $cr(A) + cr(B) = \sum_{x \in A} m(x) + \sum_{x \in B} m(x) = \sum_{x \in A \cup B} m(x) = cr(A \cup B)$ C'est une probabilité.

Autre cas particulier avec m normale :

Théorème Si les propositions focales sont totalement ordonnées par l'implication ou, ce qui revient au même, que les événements focaux sont totalement ordonnés par l'inclusion, alors cr est une mesure de nécessité, et pl une possibilité.

Si F est le plus grand élément focal de A et G celui de B, on peut supposer $F \subseteq G$ et donc $F \subseteq A \cap B$. L'intersection $A \cap B$ n'en a donc pas de plus grand et :

$$cr(A \cap B) = cr(A) = \sum_{H \subseteq F} m(H) = cr(A) = \min(cr(A), cr(B))$$

Réciproquement supposons que pour toutes parties A et B, on ait $cr(A \cap B) = \min(cr(A), cr(B))$. Alors, pour deux éléments focaux F et G quelconques la propriété vaut, F et G ne peuvent donc être disjoints car m est normale. Si F et G se coupent donc, en supposant $cr(F) \leq cr(G)$, $cr(F \cap G) = cr(F) = \sum_{H \subseteq F \cap G} m(H) = \sum_{H \subseteq F} m(H)$ signifie que les éléments focaux H de F sont exactement les mêmes que ceux contenus à la fois dans F et dans G, en particulier pour $H = F$, on a $F \subseteq G$.

PROBABILITÉ PIGNISTIQUE

Si m est une masse sur Ω fini et si |F| désigne le cardinal de F, elle est définie comme la distribution $pp(x) = \sum_{x \in F \text{ focal}} m(F) \cdot |F|$, [Smets 82, Smets 90] on vérifie que la somme totale égale à 1.

On déduit donc pour une partie quelconque A de Ω que $pp(A) = \sum_{F \cap A \neq \emptyset} m(F) \cdot |F \cap A| / |F| \leq pl(A)$

On peut montrer qu'il s'agit de la moyenne de toutes les probabilités qui sont inférieures à cette plausibilité pl définie par m.

EXTENSION DE LA NOTION DE MASSE À DES ÉLÉMENTS FOCaux FLOUS [Yager 95]

Si m est définie comme un poids non nul accordé à un nombre fini de sous-ensembles flous normalisés (dits focaux) de Ω , et si la masse totale est 1, il est possible de poser $cr(A) = \sum_{B \text{ focal}} m(B) \cdot \text{Inclu}(B, A)$ et $pl(A) = \sum_{B \text{ focal}} m(B) \cdot \text{Inter}(B, A)$ où les relations

floues d'inclusion et d'intersection sont définies comme d'habitude par $\text{Inter}(B, A) = \sup(\min(\mu_A, \mu_B))$ et (en notant $\neg A$ le complémentaire flou de A), et $\text{Inclu}(B, A) = 1 - \text{Inter}(B, \neg A) = \inf(\max(\mu_A, 1 - \mu_B))$.

RÈGLE DE DEMPSTER

Si plusieurs sources d'informations sont disponibles pour une proposition P, par des mesures de crédibilité et plausibilité, chaque source fournit une pondération probabiliste, soit deux «masses» m_1 et m_2 . La règle de Dempster (1968) encore appelée somme orthogonale que nous noterons par le symbole \oplus , généralise Bayes (1763) en donnant une combinaison commutative et associative de m_1 et m_2 :

$$m_1 \oplus m_2 (P) = \frac{\sum \{m_1(Q) * m_2(R) / Q \wedge R = P\}}{1 - \sum \{m_1(Q) * m_2(R) / Q \wedge R = \emptyset\}}$$

Le principe de cette opération est de considérer, pour P, la somme des masses pour les couples Q et R dont la conjonction est P, c'est à dire : $\alpha \sum \{m_1(Q)m_2(R) / Q \wedge R = P\}$ où $\alpha = 1 / (1 - c)$ est un coefficient normalisateur mesurant de 1 à l'infini le conflit. $c = \sum \{m_1(Q)m_2(R) / Q, R \text{ incompatibles}\}$ entre 0 et 1, est appelé le degré de conflit entre m_1 et m_2 . Lorsque c devient proche de 1, la normalisation opérée en divisant par 1-c devient évidemment discutable, lorsqu'il est égal à 1, la combinaison n'est pas définie.

Exercice 2.2

En posant $(Q(m))(A) = \sum_{A \subseteq C} m(C)$ appelée «commonalité» de m, montrer un résultat inverse donnant la crédibilité $cr(A) = \sum_{A \cap B = \emptyset} (-1)^{|B|} Q(B)$ ainsi que : $Q(m_1 \oplus m_2) = \alpha Q(m_1).Q(m_2)$.

Pour un ensemble de propositions $\{0, P, \neg P, 1\}$, si $m(P) = L$ et $m(\neg P) = 1-U$, une partie de la masse n'est pas attribuée, c'est $1 - \sum m(q) = 1 - [L + (1-U)] = U - L$

L'application de la règle à partir de deux couples (L_1, U_1) et (L_2, U_2) c'est à dire $m_1(P) = L_1$ et $m_2(P) = L_2$, $m_1(\neg P) = 1-U_1$, $m_2(\neg P) = 1-U_2$ (L pour «low» et U pour «up») peut recevoir plusieurs légitimations :

En fait le point de vue probabiliste est le suivant, pour chaque source d'information nous avons trois événements, en premier lieu P connu avec au moins la probabilité L, puis $\neg P$ avec au moins 1-U, et enfin «l'ignorance sur P» que nous notons ?P dont la probabilité est le complément à 1 soit U-L. En confrontant deux sources nous avons le tableau ci-dessous tel que la somme des lignes et des colonnes donne ces trois valeurs pour P1 et P2.

On considère alors que si P1 et P2 sont les deux conclusions pour P relatives aux sources m_1, m_2 , P est obtenu avec dans le numérateur de la formule, P1 et P2, P1 et Ignorance sur P2, et enfin, Ignorance sur P1 et P2.

Les situations de conflit sont les cases grisées dont la probabilité est degré de conflit

$$c = L_1(1-U_2) + L_2(1-U_1)$$

		Preuve 1		Preuve 2	
		P	$\neg P$	P	$\neg P$
		L1	1 - U1	L2	1 - U2
P	L2	$L_1 * L_2$	$L_2(1 - U_2)$	$L_2(U_1 - L_1)$	
$\neg P$	1 - U2	$L_1.(1 - U_2)$	$(1 - U_1).(1 - U_2)$	$(1 - U_2).(U_1 - L_1)$	
?P	U2 - L2	$L_1.(U_2 - L_2)$	$(1 - U_1).(U_2 - L_2)$	$(U_1 - L_1).(U_2 - L_2)$	

Figure 2.3 Confrontation de deux preuves de P établies avec les supports respectifs (L_1, U_1) et (L_2, U_2)

La probabilité basse de P est obtenue par l'événement «au moins une fois P» (cases entourées en gras) qui est donc :

$$L' = [L_1 * L_2 + L_1 * (U_2 - L_2) + (U_1 - L_1) * L_2] = L_1 * U_2 + L_2 * U_1 - L_1 * L_2$$

$$= L_1 + L_2 - L_1 * L_2 - c \text{ mais en divisant par } 1-c \text{ (les situations acceptables), on fait une normalisation soit :}$$

$$L = (L_1 + L_2 - L_1 L_2 - c) / (1 - c)$$

De même la probabilité basse de $\neg P$ est (zones blanches) :

$$L(\neg P) = [(1-U_1)(1-U_2) + (1-U_1)(U_2-L_2) + (U_1-L_1)(1-U_2)] = 1 - c - U_1 * U_2 \text{ puis, en normalisant (en divisant par } 1-c) \text{ et en prenant le complément à 1, on a}$$

$$U = U_1 U_2 / (1 - c)$$

On vérifiera que la zone d'incertitude (pointillée) $(U_1 - L_1)(U_2 - L_2)$ normalisée par $(1 - c)$ est bien quantifiée par $U - L$.

En définitive, cette opération d'agrégation dans $S = \{(x, y) / 0 \leq x \leq y \leq 1\}$ s'exprime par :

$$(x, y) \oplus (x', y') = \left(\frac{x + x' - xx' - c}{1 - c}, \frac{yy'}{1 - c} \right) = \left(\frac{xy' + x'y - xx'}{1 - x - x' + xy' + x'y}, \frac{yy'}{1 - x - x' + xy' + x'y} \right)$$

Cette opération est commutative, associative, mais non idempotente.

Le vrai (1, 1) et le faux (0, 0) sont absorbants, l'incertain (0, 1) est neutre, et elle est partout définie sauf pour le couple de supports le plus antagoniste (0, 0) avec (1, 1).

Plus c est grand, plus cette règle devient comme on l'a déjà dit, discutable. Ces formules d'agrégation ont aussi été employées dans Mycin (voir aussi l'annexe langage FRIL).

Pour un point de vue algébrique, voir [Gacogne 93] et surtout [Daniel 94]. En cherchant les fonctions # de [0, 1] dans lui-même respectant les trois contraintes : deux points fixes $f(1) = 1$ et $f(0) = 0$, (1/2, 1/2) centre de symétrie (on souhaite l'auto-dualité $\# \neg x = \neg \# x$) et $f'(x) = 0$ pour $x = 0$ ou 1 (T et F sont atteints plus vite par #P que par P), la fonction polynômiale de plus petit degré est $\#(x) = x^2(3 - 2x)$ et la fonction rationnelle de plus petits degrés étant $\#x = x^2 / (2x^2 - 2x + 1)$.

C'est cette dernière qui, étant choisie comme «connecteur de renforcement», se généralise naturellement à $([0, 1]^2)^2$ sous l'opération de Dempster. D'autres renforcements sont possibles [Pearl 90], ainsi $\#x = 1 - (1 - x^a)^a$, mais ils ne sont pas auto-duaux.

EXEMPLE [Klir 88]

Face à un tableau, on pose 3 questions R = peinture de Raphaël, D = oeuvre de l'un de ses disciples, C = contrefaçon, deux experts donnent leurs «poids» à 7 événements. On calcule $c = 0,03$ et m est le résultat de l'agrégation de m_1, m_2 .

	m1	cr1	m2	cr2	m	cr
R	0.05	0.05	0.15	0.15	0.21	0.21
D	0	0	0	0	0.01	0.01
C	0.05	0.05	0.05	0.05	0.09	0.09
R ou D	0.15	0.20	0.05	0.20	0.12	0.34
R ou C	0.1	1	0.5	1	0.31	1
R ou D ou C	0.6	1	0.5	1	0.31	1

APPLICATION EN RECHERCHE PÉTROLIÈRE [Aminzadeh 94]

Dans ce système expert, plusieurs connaissances d'ordre géochimiques ou géologiques sont formalisées en vue d'arriver à déterminer un type de saturation dans une couche géologique. Les règles utilisent des prédicats flous comme «plutôt supérieur à 2» ou «plutôt inférieur à 1». L'intérêt de cette application réside dans le fait que plusieurs conclusions s'excluent mutuellement (autres qu'une connaissance et son contraire) bien qu'il soit difficile de séparer certaines conclusions (eau ou pétrole...).

Par exemple, si les règles de la géologie concluent à «gaz ou pétrole» avec une probabilité minimale de 0.4, une première agrégation avec les conclusions de la géochimie va se présenter :

Géol. \ Géochimie	Eau 0.5	Pétrole 0.3	Indétermination 0.2
Gaz ou Pétrole 0.4	Conflit 0.2	Pétrole 0.12	Gaz ou Pétr. 0.08
Indét. 0.6	Eau 0.3	Pétrole 0.18	Indéterminé 0.12

La normalisation s'opère en multipliant par $1 / (1 - c) = 1.25$ ce qui donne la première colonne du tableau ci-dessous (par exemple le pétrole obtient $1.25 * (0.12 + 0.18)$). A présent une agrégation est effectuée avec les conclusions d'une autre série de règles issues de la géophysique (première ligne) :

Précéd. \ Géoph.	Gaz 0.03	Eau ou Pétrole 0.4	Indéterminé 0.3
Gaz ou Pétrole 0.1	Gaz 0.03	Pétrole 0.04	Gaz ou Pétrole 0.03
Eau 0.375	Conflit 0.1125	Eau 0.15	Eau 0.1125
Pétrole 0.375	Conflit 0.1125	Pétrole 0.15	Pétrole 0.1125
Indéterminé 0.15	Gaz 0.045	Eau ou Pétrole 0.06	Indéterminé 0.045

La normalisation s'effectue en multipliant par $1 / (1 - 0.1125 - 0.1125) = 1.29$, ce qui donne les crédibilités 0.097 pour le Gaz, 0.039 pour Gaz ou Pétrole, 0.339 pour l'eau, 0.077 pour Eau ou Pétrole, 0.39 pour le pétrole seul et enfin 0.058 pour l'indétermination. Pour les trois conclusions intéressantes, on a donc :

$$\begin{aligned}
 pl(\text{gaz}) &= 1 - cr(\neg\text{gaz}) = 1 - cr(\text{eau}) - cr(\text{eau ou pétrole}) - cr(\text{pétrole}) \\
 &= 1 - 0.339 - 0.077 - 0.39 = 0.194 \text{ et } pl(\text{pétrole}) = 1 - cr(\text{gaz}) - cr(\text{eau}) \\
 &= 1 - 0.097 - 0.339 = 0.564
 \end{aligned}$$

D'où les supports $(L - U)$ pour le gaz $(0.097 - 0.194)$, pour l'eau $(0.339 - 0.474)$ et le pétrole $(0.39 - 0.564)$ qui est donc le plus probable.

UN POINT DE VUE UNIFICATEUR ENTRE 3 TYPES DE THÉORIE

On peut représenter la logique booléenne par les deux points T(vrai) et F (faux), les logiques multivaluées par le segment FT, la théorie des possibilité par l'ensemble des couples des segments FI et IT, (en notant I l'indéterminé) et enfin la théorie des probabilités haute et basse (notées ici obligation et éventualité) par la zone hachurée. La négation étant dans tous les cas la symétrie par rapport à IM.

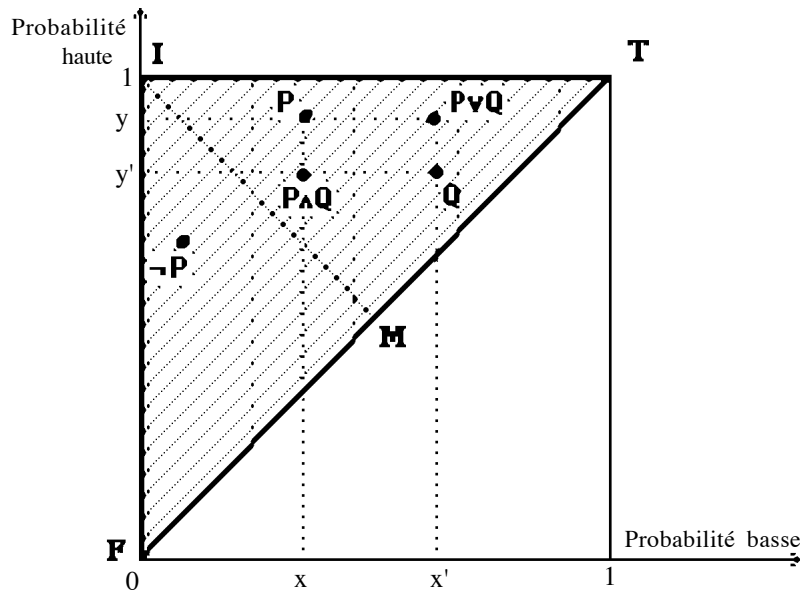


Figure 2.4 L'ensemble $\{(x, y) / 0 \leq x \leq y \leq 1\}$ des supports.

Ajoutons que l'axe MI est celui des incertitudes croissantes et IM celui, au contraire, de la connaissance croissante, son maximum en (1, 0) serait la «contradiction» notée TF.

Voir l'annexe sur le système MVL pour le système d'axes formés par (FT) et (IM).

Un autre système d'axes formé IT et IF où les coordonnées deviennent I (0, 0), T (1, 0), F (0, 1), M (1/2, 1/2), c'est à dire par $a = x$ et $b = 1 - y$, peut être envisagé, la négation est alors $\neg(a, b) = (b, a)$ le symétrique par rapport à (IM) et la règle de Dempster s'écrit :

$$(a, b) \otimes (c, d) = [1 - (1 - a)(1 - c) / (1 - (ad + bc)), 1 - (1 - b)(1 - d) / (1 - (ad + bc))]$$

Pour tout élément (a, b), son image par la fonction définie comme $h(a, b) = (a, b) \otimes M = ((1 - b) / (2 - a - b), (1 - a) / (2 - a - b))$ est le point de [F, T], donc sans incertitude, intersection avec le segment joignant (a, b) et le sommet (1, 1) qui représente la «contradiction». Pour une telle structure (un dempsteroïde) [Daniel 94, 96] montre que [F, T] est un groupe pour \otimes et que tout automorphisme préservant I et M est entièrement déterminé comme extension d'un couple d'automorphismes sur FT et sur FI \cup IT.

REMARQUES

En observant les itérés par la règle de Dempster (non idempotente) d'un point dans l'ensemble des supports (x, y), on s'aperçoit que suivant sa position, le résultat consiste en une «accentuation» vers le faux (0, 0) ou bien vers le vrai (1, 1). Les exceptions sont le cas de l'incertain (0, 1) dont le carré pour cette coopération est lui-même et les autres éléments de IM dont on peut montrer que les itérés se rapprochent de M. L'opération de Dempster est donc bien un renforcement de la certitude.

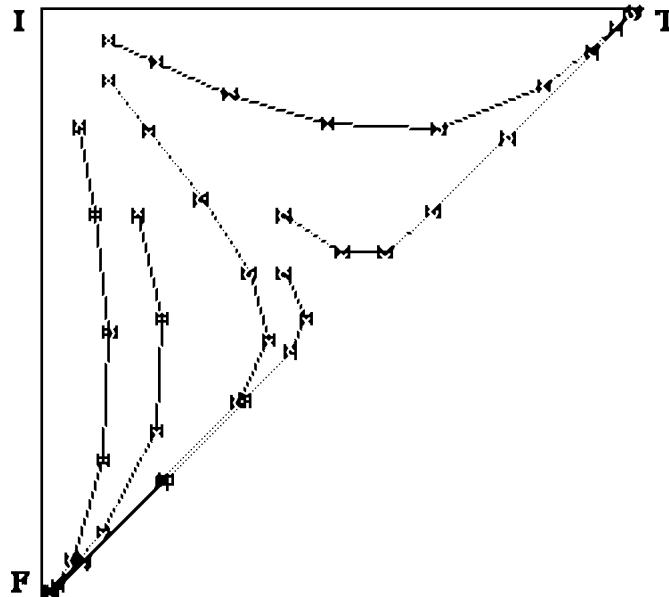


Figure 2.5 Six exemples d'itération par Dempster convergent vers T ou F.

En disposant régulièrement des points suivant un quadrillage, les carrés au sens de l'opération de Dempster montrent un mouvement vers moins d'incertitude en se rapprochant du segment FT, tout en se rapprochant également des deux valeurs booléennes F et T.

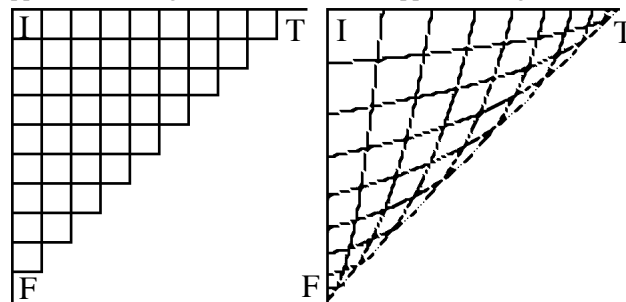


Figure 2.6 Déformation d'un quadrillage par Dempster

APPLICATION À LA FUSION DE DONNÉES MULTI-CAPTEURS

Le problème général est de déterminer une hypothèse (la meilleure) parmi n objets ou n classes, lorsque p sources d'information (des capteurs par exemple) envoient leurs «avis». Ce problème se présente notamment pour le traitement des images prises par satellites. On appelle «cadre de discernement» un ensemble de n hypothèses (une classification en fait) non nécessairement exclusives, par exemple $E = \{\text{ami, ennemi, neutre}\}$ ou encore $E = \{\text{avion-de-chasse, transporteur, bombardier, hélicoptère-léger, hélicoptère-lourd}\}$ avec $n = 5$. On dispose de m capteurs ayant des sensibilités différentes d'une cible à l'autre et le problème est de fusionner leurs conclusions.

On nomme matrice de confusion, la matrice (à trois dimensions) formée par les :

$$p(i, k, j) = \text{proba}(\text{«le capteur } j \text{ émette l'avis } k \text{ en présence de l'objet } i\text{»})$$

Si H_i est une hypothèse et D_j une déclaration (de capteur) :

$$\text{Dans l'approche bayésienne, } p(H_i) = \frac{\prod_{j=1}^p p(D_j / H_i) p(H_i)}{p(D_j)}$$

En fait, le capteur j renvoie simplement une «grandeur de discrimination» $e(k, j)$ mesurant la vraisemblance de l'hypothèse k et $p(i, k, j)$ est la probabilité que ce niveau $e(k, j)$ soit le plus grand de tous les $e(h, j)$ pour toutes les hypothèses h , sachant que la cible présentée est i .

Dans le cadre restreint $E = \{H_i, \neg H_i\}$, seuls $H_i \cap H_i = H_i$ et $H_i \cap E = H_i$, alors la règle de Demspster donne $m_i(H_i) = \sum m_{i1}(A_1).m_{i2}(A_2)...m_{im}(A_m) / (1 - c)$ d'où : $(1-c)m_i(H_i) = \prod_{1 \leq j \leq m} (m_{ij}(H_i) + m_{ij}(E)) - \prod m_{ij}(E)$

On en déduit pour ce même E une plausibilité pl_i que l'on va confondre avec celle relative au cadre de discernement $E = \{H_1, H_2, \dots, H_n\}$ donc $pl(H_i) = pl_i(H_i) = m_i(H_i) + m_i(E) = \{ \prod_{1 \leq j \leq m} m_{ij}(H_i) \} / (1-c)$

On souhaite maintenant choisir l'hypothèse H_i telle que $C_{i1} = pl(\neg H_i) = 1 - cr(H_i)$ soit minimum et que $C_{i2} = pl(H_i)$ soit maximum. La conservation des variations relatives exige le système différentiel $\partial C_i / C_i = -\partial C_{i1} / C_{i1}$ et $\partial C_i / C_i = \partial C_{i2} / C_{i2}$. On peut donc choisir de maximiser la solution $C_i = C_{i2} / C_{i1}$ soit :

$$C_i = \frac{pl(H_i)}{1 - cr(H_i)} = \frac{\prod_{j=1}^m m_{ij}(H_i)}{\prod_{j=1}^m (1 - m_{ij}(H_i))} = \prod_{j=1}^m \frac{e_{ij}^{\max}}{1 - e_{ij}^{\min}}$$

EXPÉRIENCE, si $n = m = 3$ et $p_i = \text{proba}$ (déclaration i / hypothèse i) en supposant que les trois capteurs fonctionnent de la même façon. La matrice de confusion est :

Hypothèses :	H1	H2	H3
D1	p_1	$(1 - p_2) / 2$	$(1 - p_3) / 2$
Déclarations : D2	$(1 - p_1) / 2$	p_2	$(1 - p_3) / 2$
D3	$(1 - p_1) / 2$	$(1 - p_2) / 2$	p_3

A partir des p_i , sont calculés les autres éléments de la matrice par $p_{ij} = (1-p_i) / (n-1)$, la cible choisie i sera celle qui va maximiser le critère C_i défini par :

Solutions de la théorie de l'évidence : $C_i = \prod_{1 \leq k \leq n} \prod_{1 \leq j \leq m} (1 - p(i, k, j) * (1 - e(k, j)))$.

puis $C_i = \prod_{1 \leq k \leq n} \prod_{1 \leq j \leq m} [(1 - p(i, k, j) * (1 - e(k, j))) / (1 - p(i, k, j) * e(k, j))]$ (maximisations de la plausibilité et de la crédibilité, courbe «ev»)

De façon analogue dans [Appriuri 88] d'autres méthodes sont comparées :

Solutions Bayésienne $C_i = \prod_{1 \leq j \leq m} e(i, j)$ pour «prob1» et $C_i = \prod_{1 \leq j \leq m} p(i, k, j)$ où $p(i, k, j)$ est le coefficient de la matrice de confusion du capteur j pour la cible i réalisant le plus grand $e(h, j)$ pour toutes les hypothèses h (courbes «prob2»).

Solution possibiliste : \max des $C_i = \min_{1 \leq k \leq n, 1 \leq j \leq m} (1 - p(i, k, j) * (1 - e(k, j)))$ (courbe «poss»).

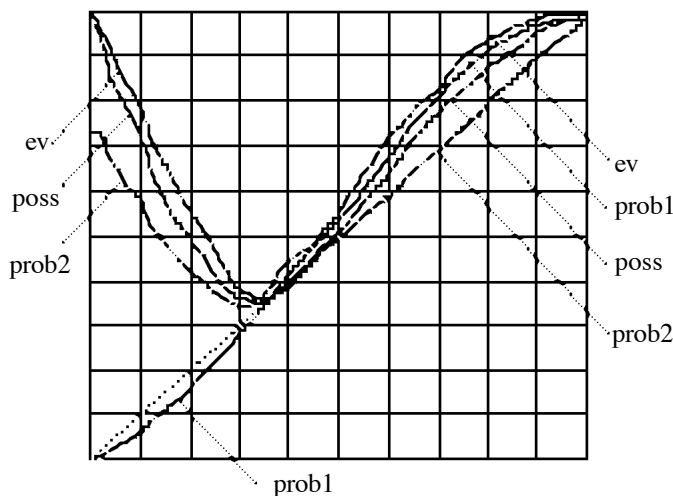


Figure 2.7 Taux de reconnaissance en fonction de a.

En augmentant la taille n du discernement, les performances augmentent en convergeant vers une courbe proche de celle de $n = 10$.

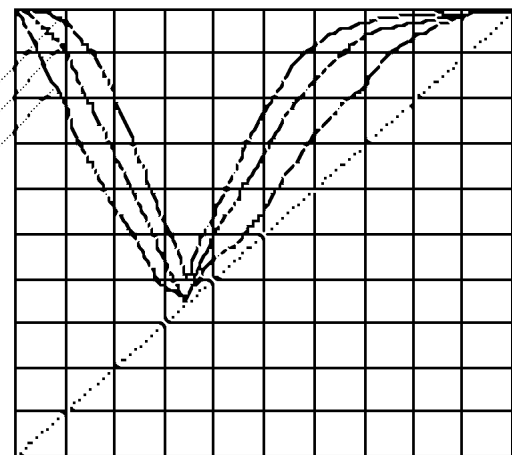


Figure 2.8 (a varie de 0 à 1 en abscisse et C_i de 0 à 1 en ordonnée)

On effectue des tirages aléatoires des coefficients $e(k,j)$ un grand nombre de fois (plus de 5000), et trace les courbes du taux de reconnaissance par fusion : (nombre de succès) / (nombre d'essais) en fonction de la valeur commune $a = p_i$.

Si $a < 1/3$ (très mauvais capteurs) les résultats sont meilleurs à ceux d'un capteur unique, mais cela signifie que les capteurs ont plus de chance de proposer une cible différente de celle qui est présentée. Un capteur complètement aléatoire signifie $a = 1/3$, en ce cas la fusion de plusieurs de ces capteurs doit également fournir un taux de reconnaissance $1/3$, ce qui est le cas. Les cas intéressants se réalisent pour $a > 1/3$, alors les résultats de ev sont meilleurs que ceux d'un capteur unique, et supérieurs à ceux de poss.

En faisant varier le nombre de capteurs (second schéma), l'expérience montre que l'on peut classer prob1, prob2, pos, ev dans l'ordre croissant, mais qu'à partir de 8 capteurs les performances de «poss» sont comparables à celles de «ev».

Chapitre 3

Opérateurs logiques

Afin de généraliser la logique booléenne où on ne compte que deux valeurs de vérité, plusieurs systèmes de logiques trivaluées ont été proposés à partir de 1920, considérant le vrai (1 ou T), le faux (0, ⊥ ou F), et une troisième valeur généralement interprétée comme «l'indéterminé» (1/2 ou I) ou le «possible», par Kleene, Lukasiewicz et Przymusiński. Puis apparurent des systèmes avec un nombre fini de valeurs $\{0, 1/n, 2/n, \dots, (n-1)/n, 1\}$ et enfin des systèmes à valeurs continues dans $[0, 1]$. Nous étudions dans ce chapitre, différentes généralisations des opérateurs logiques booléens, ainsi que le problème de la déduction approchée et de l'agrégation avec ces formulations.

3.1. Les connecteurs logiques

LES PREMIERS SYSTÈMES DE LOGIQUE TRIVALUÉE

Pour $\{F, I, T\}$, désignant les valeurs logiques «faux», «indéterminé» et «vrai», également notées $\{0, 1/2, 1\}$ plusieurs généralisations de la logique binaire ont été proposées. Dans tous les cas la négation échange T et F pour laisser fixe I. La logique L3 de Lukasiewicz (1920)

q	p ∧ q		
1	0	1/2	1
1/2	0	1/2	1/2
0	0	0	0
	0	1/2	1

q	p ∨ q		
1	1	1	1
1/2	1/2	1/2	1
0	0	1/2	1
	0	1/2	1

q	p → q		
1	1	1	1
1/2	1	1	1/2
0	1	1/2	0
	0	1/2	1

Cette logique, comme les suivantes, vérifie les lois de Morgan et définit l'équivalence par : $(p \leftrightarrow q) = (p \rightarrow q) \wedge (q \rightarrow p)$

Elle ne vérifie pas $(p \rightarrow q) = \neg p \vee q$. L'implication peut être rendue par la formule :

$(p \rightarrow q) = [\text{si } p \leq q \text{ alors } 1 \text{ sinon } 1 - p + q]$, qui permet la généralisation à un nombre fini de valeurs de vérité ou à toutes les valeurs de $[0, 1]$. La logique B3 de Bochvar (1939)

q	p ∧ q		
1	0	1/2	1
1/2	1/2	1/2	1/2
0	0	1/2	0
	0	1/2	1

q	p ∨ q		
1	1	1/2	1
1/2	1/2	1/2	1/2
0	0	1/2	1
	0	1/2	1

q	p → q		
1	1	1/2	1
1/2	1/2	1/2	1/2
0	1	1/2	0
	0	1/2	1

On peut définir :

$p \wedge q = [\text{si au moins un des deux vaut } 1/2 \text{ alors } 1/2, \text{ si } p = q = 1 \text{ alors } 1 \text{ sinon } 0]$.

En ce cas $(1/2) \wedge 0 = 1/2$ ce qui semble peu intuitif.

Quant à la disjonction : $p \vee q = [\text{si au moins un est } 1/2 \text{ alors } 1/2, \text{ si } p = q = 0 \text{ alors } 0 \text{ sinon } 1]$.

L'implication vérifie $(p \rightarrow q) = \neg p \vee q$, elle peut être définie par :

$(p \rightarrow q) = [\text{si au moins un est } 1/2 \text{ alors } 1/2, \text{ si } p = 1 \text{ et } q = 0 \text{ alors } 0, \text{ sinon } 1]$.

La logique S3 de Sobocinski (1952). L'implication vérifie $(p \rightarrow q) = \neg p \vee q$.

q	p ∧ q		
1	0	1	1
1/2	0	1/2	1
0	0	0	0
	0	1/2	1

q	p ∨ q		
1	1	1	1
1/2	0	1/2	1
0	0	0	1
	0	1/2	1

q	p → q		
1	1	1	1
1/2	1	1/2	0
0	1	0	0
	0	1/2	1

Pour les logiques K3 de Kleene (1938) et H3 de Heyting (1956), $p \wedge q$ et $p \vee q$ sont définis comme dans L3.

q	K3			p→q	q	H3			p→q	q	p→q		
1	1	1	1	1	1	1	1	1	1	1	1	1	1
1/2	1	1/2	1/2	1/2	1/2	1	1	1/2	1/2	1/2	1	0	0
0	1	1/2	0	0	0	1	0	0	0	0	1	0	0
	0	1/2	1	p	0	1/2	1	p		0	1/2	1	

L'implication de Kleene vérifie $(p \rightarrow q) = \neg p \vee q$, celle de Heyting, n'ayant pas de symétrie par rapport à la seconde diagonale, ne le vérifie pas non plus que la contraposition $(\neg q \rightarrow \neg p) = (p \rightarrow q)$.

La définition de Gödel donnée par $(p \rightarrow q) = [si\ p \leq q\ alors\ 1\ sinon\ q]$, généralise cette dernière implication.

A noter le dernier tableau, qui constitue une implication obtenue à partir de :

$p \wedge q = si\ l'un\ est\ faux\ alors\ 0\ sinon\ 1$, qui n'a que des résultats binaires.

Gödel a prouvé qu'aucune logique multivaluée finie ne pouvait engendrer une théorie (un calcul propositionnel) identique à celle du calcul propositionnel intuitionniste (annexe 2).

Par contre, ce dernier augmenté de l'axiome $(A \rightarrow B) \vee (B \rightarrow A)$ engendre la même théorie que la logique de Gödel ayant un nombre dénombrable de valeurs de vérité. En 1936, Jaskowski montre une logique multivaluée infinie engendrant la même théorie que le calcul propositionnel intuitionniste, enfin Tarski a montré que H3 était équivalent à ce dernier augmenté de l'axiome : $(\neg A \rightarrow B) \rightarrow ((B \rightarrow A) \rightarrow B) \rightarrow B$. Les logiques trivaluées ou avec un nombre supérieur, mais fini de valeurs, ont été appliquées à des problèmes concrets, y compris grâce à des seuils, pour des circuits électroniques [Kameyana, Higushi 82].

Cependant l'intérêt pour ces logiques a chuté depuis les années 70 et 80. En logique «floue», on choisit habituellement : $p \wedge q = \min(p, q)$ et $p \vee q = \max(p, q)$ qui sont les généralisations de L3.

NÉGATION

Dans $[0, 1]$ la négation utilisée est toujours le complément à 1 grâce à la simple formule $N(a) = 1 - a$, cependant toute application continue involutive et strictement décroissante telle que $N(0) = 1$ (négation forte) peut convenir comme définition.

Exercice 3.1

Montrer que les applications $N(x) = (1 - x^2)^{1/2}$ ou $N(x) = (1 - \sqrt{x})^2$ sont des négations. Le complément de Sugeno $N_\lambda(x) = (1 - x) / (1 + \lambda x)$ en est-il une ?

DÉFINITIONS DES T-NORMES ET T-CONORMES

Les t-normes [Menger 42] étendent la conjonction ordinaire. Les t-normes T sont les applications commutatives, associatives, croissantes de $[0, 1]^2$ dans $[0, 1]$ (telles que $a \leq c$ et $b \leq d$ entraînent $T(a, b) \leq T(c, d)$) vérifiant $T(0, 0) = 0$ et où 1 est neutre, pour tout a $T(a, 1) = a$ (la stricte monotonie n'est pas exigée par la définition).

Une t-norme est dite diagonale si et seulement si $T(x, x) \leq x$ pour $x < 1$. Les plus connues sont :

$T_0(a, b) = si\ \max(a, b) = 1\ alors\ \min(a, b)\ sinon\ 0$	(produit drastique de Weber)
$T_1(a, b) = \max(0, a + b - 1)$	(Lukasiewicz)
$T_{1,5}(a, b) = ab / (2 - a - b + ab)$	(Einstein)
$T_2(a, b) = ab$	(algébrique ou probabiliste)
$T_{2,5}(a, b) = ab / (a + b - ab)$	(Hamacher)
$T_3(a, b) = \min(a, b)$	(Zadeh)

Exercice 3.2

Vérifier $T_0 \leq T_1 \leq T_{1,5} \leq T_2 \leq T_{2,5} \leq T_3$ et que toute autre T-norme est comprise entre T_0 et T_3 . Montrer que T_1 possède des diviseurs de 0 ainsi $T(1/2, 1/2) = 0$.

Exercice 3.3

On appelle générateur additif f une fonction numérique strictement décroissante telle que $f(0) = +\infty$ et vérifiant $f(1) = 0$.

Montrer que $T(a, b) = f^{-1}(f(a) + f(b))$ ramené à $[0, 1]$ est une t-norme et l'exprimer pour les quatre fonctions $f(x) = 1 - x$, $-\ln(x)$, $1/x - 1$, $\cotan(\pi x/2)$. Quelles sont ces t-normes dans les trois premiers cas ? (réponses Lukasiewicz, probabiliste, Hamacher).

Montrer qu'en général, T est continue et strictement monotone. Soit $f(1) = 0$ et $f(x) = 2 - x$ pour $0 \leq x < 1$, prouver que T est la norme de Weber, en donnant cette fois pour pseudo-inverse $f^{-1}(y) = \sup\{t \in [0, 1] / f(t) > y\}$.

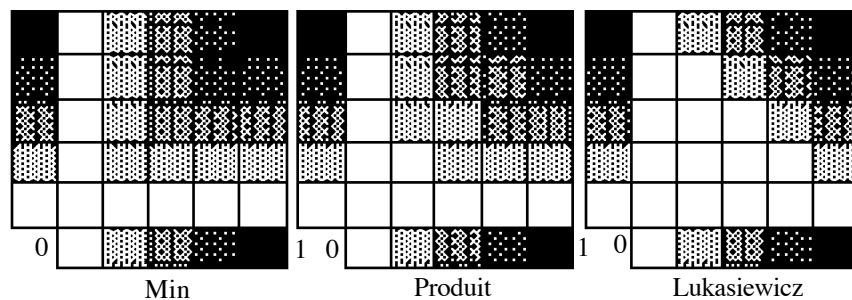


Figure 3.1 Illustration de trois conjonctions avec 5 valeurs, de 0 à 1 du blanc au noir.

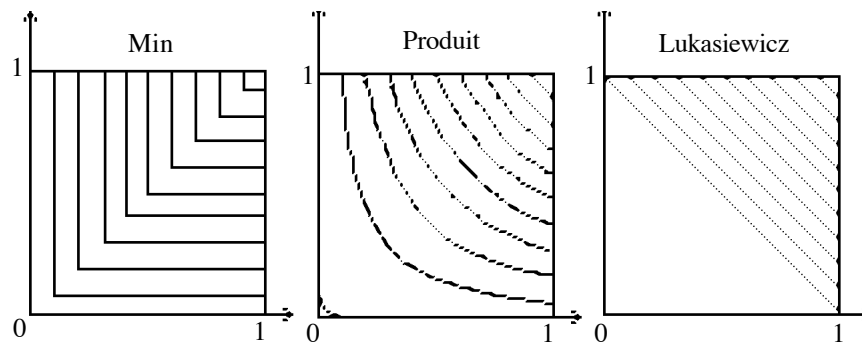


Figure 3.2 Plus finement dans $[0, 1]^2$, on peut tracer (ici 11) lignes de niveaux.

Exercice 3.4

Si f est une bijection continue croissante de $[0, 1]$, montrer que $N(x) = f^{-1}(1 - f(x))$ est une négation et que l'on obtient une t-norme en définissant : $T(x, y) = [\text{si } f(x) + f(y) > 1 \text{ alors } f^{-1}(f(x) + f(y)) \text{ sinon } 0]$.

Les définitions de Sugeno $n(x) = (1 - x) / (1 + px)$ et de Yager $n(x) = (1 - x^p)^{1/p}$ peuvent-elles être des négations ?

Exercice 3.5

Si f est une bijection continue croissante de $[0, 1]$, montrer que :

$T(x, y) = f^{-1}(f(x).f(y))$ est une t-norme.

Exprimer T pour $f(x) = (a^x - 1) / (x - 1)$, pour $4 \arctan(x) / \pi$ et pour $x^2 / (2x^2 - 2x + 1)$.

Exercice 3.6

Une t-norme T est archimédienne si pour tous x, y dans $]0, 1[$, il existe une puissance n de x vis à vis de T , pour laquelle $x^{(n)} = T(x, \dots, x) < y$. Montrer qu'alors pour tout x différent de 1 la limite de $x^{(n)}$ est 0 quand n tend vers l'infini. Montrer la réciproque.

T est régulière si $\forall x, y, z \ T(x, y) = T(x, z) \text{ et } x \neq 0 \Rightarrow y = z$.

Montrer que T est régulière si et seulement si elle est strictement monotone.

T est « diagonale » si $\forall x < 1 \ T(x, x) < x$

Montrer que « min » n'est ni archimédienne, ni régulière, ni diagonale, mais continue.

Montrer que la norme de Weber est archimédienne et diagonale mais ni continue ni régulière.

Montrer que si T est continue, la régularité entraîne que T est diagonale et archimédienne.

$T(x, y) = \text{si } x, y \leq 1/2 \text{ alors } 0, \text{ sinon si } 1/2 < x, y \text{ alors } 2xy - x - y + 1 \text{ sinon } \min(x, y)$ est la norme de Schweizer et Sklar. Montrer qu'elle est diagonale mais ni continue, ni archimédienne, ni régulière.

LES T-CONORMES

Ce sont les applications S commutatives associatives croissantes depuis $[0, 1]^2$ dans $[0, 1]$ (telles que $a \leq c$ et $b \leq d$ entraînent $S(a, b) \leq S(c, d)$), vérifiant $S(1, 1) = 1$ et telles que pour tout a , on ait $S(a, 0) = a$ (0 pour neutre). En notant $N(a) = 1 - a$ la négation, les deux notions sont évidemment duales ce qui signifie : $S(a, b) = N(T(N(a), N(b)))$. On a donc les t-conormes associées :

$S_0(a, b) = \text{si } \min(a, b) = 0 \text{ alors } \max(a, b) \text{ sinon } 1$	(Weber)
$S_1(a, b) = \min(1, a + b)$	(Lukasiewicz)
$S_{1,5}(a, b) = (a + b) / (1 + ab)$	(Einstein)
$S_2(a, b) = a + b - ab$	(Probabiliste)
$S_{2,5}(a, b) = (a + b - 2ab) / (1 - ab)$	(Hamacher)
$S_3(a, b) = \max(a, b)$	(Zadeh)

On peut alors vérifier $S_3 \leq S_{2,5} \leq S_2 \leq S_{1,5} \leq S_1 \leq S_0$ et que toute autre t-conorme est comprise entre S_3 et S_0 . Regardons par exemple simplement le cas où $a = 0$ et $0 < b$, alors $b \leq S(0, b) \leq b$ donc $S_3 \leq S \leq S_0$, par ailleurs si $0 < a \leq b$, $b = S(0, b) \leq S(a, b) \leq 1$ donc c'est encore vrai.

Voir des études systématiques des t-normes et t-conormes $[0, 1]^2 \rightarrow [0, 1]$ dans [Yager 80], [Dubois Prade 85], [Bonissone 87], [Mizumoto Masaharu 89].

ROBUSTESSE [Nguyen H.T. 96]

Pour une fonction f de $[0, 1]^n$ dans $[0, 1]$, on peut définir le risque en ∂ de f par :

$\Delta_f(\partial) = \sup \{ \|f(x) - f(y)\| / |x - y| < \partial \}$, et on montre que parmi les t-normes (resp. t-conormes), la moins sensible est min (resp. max). Par contre, en se restreignant aux t-normes différentiables, c'est la t-norme probabiliste qui est la moins sensible.

Exercice 3.7

On dit qu'une t-conorme est diagonale si $x < S(x, x)$.

On dit que la fonction f est un générateur pour la t-conorme S , si f est continue et strictement croissante de $[0, 1]$ vers $[0, +\infty]$ et que $S(x, y) = f^{-1}(\min(f(x), f(y)))$.

Montrer que si $f(1)$ est fini, alors S est diagonale et T nilpotente c'est à dire qu'il existe une puissance (un nombre d'occurrence de x) telle que $T(x, x, \dots, x) = 0$

Exercice 3.8

On définit l'opération de Yager par $S_n(a, b) = \min(1, (a^n + b^n)^{1/n})$ pour $n > 0$, montrer que la limite de $S_n(a, b)$ lorsque n tend vers l'infini est $\max(a, b)$.

On prendra $\ln(S_n(a, b))$ avec $a < b$ et le $\min \neq 1$.

Exercice 3.9

Que donne la famille de t-normes de Hamacher $T(x, y) = xy / [p + (1-p)(x + y - xy)]$ avec $p \geq 0$ si $p = 0$, si $p = 1$?, si p tend vers l'infini ? (réponses : Hamacher, produit, Weber). Exprimer la t-conorme associée S .

Exercice 3.10

Montrer que l'on a des t-normes pour les familles :

- a) Dombi : $T(x, y) = 1/[1 + [(1/x - 1)^p + (1/y - 1)^p]^{1/p}]$ cas de $p = 1$?
- b) Schweizer : $T(x, y) = [\max(0, x^p + y^p - 1)]^{1/p}$. Montrer que les limites en $-\infty, 0$ et $+\infty$ donnent respectivement les normes min, probabiliste et drastique. Pour $p = 1$?
- c) Franck : si $p > 0$ et $p \neq 1$ $T(x, y) = \log_p[1 + (p^x - 1)(p^y - 1) / (p - 1)]$
 Quelles sont les limites de T lorsque p tend vers 0, 1, et $+\infty$?
 (Réponses : Zadeh, produit, Lukasiewicz)
 Exprimer S et vérifier que pour tous x, y : $T(x, y) + S(x, y) = x + y$
- d) Weber : si $p \geq -1$ $T(x, y) = \max[0, (1 + p)(x + y - 1) - pxy]$.
 Regarder les cas de $p = -1$ et $p = 0$.
- e) Dubois : si $0 < p \leq 1$, $T(x, y) = xy / \max(x, y, p)$

REMARQUE, LIEN ENTRE T-NORME ET NORME

Le produit d'espaces normés est défini habituellement par $n(x, y) = \sup(n_1(x), n_2(y))$, cependant il est possible de généraliser par des définitions topologiquement équivalentes en dimension finie (théorème de Riesz).

Soient (E_1, n_1) et (E_2, n_2) deux espaces normés et S une application de R^{+2} dans R^+ , en définissant n sur le produit $E_1 * E_2$ par $n(x, y) = S(n_1(x), n_2(y))$ on peut montrer certains liens entre les propriétés de n et celle de S .

- a) S est sous-additive pour l'addition des couples $\Leftrightarrow n$ vérifie l'inégalité triangulaire
- b) Si S est croissante $[a < c$ et $b < d \Rightarrow S(a, b) \leq S(c, d)]$ alors :
 $S(1, 1) = 1 \Leftrightarrow$ Le produit des boules unités de E_1 et E_2 est la boule unité de $E_1 * E_2$
- c) La norme n se projette suivant n_1 sur E_1 , n_2 sur $E_2 \Leftrightarrow [\forall a, S(a, 0) = S(0, a) = a]$
- d) $[n(x, y) = 0 \Rightarrow x = y = 0] \Leftrightarrow [\forall a, b, S(a, b) = 0 \Leftrightarrow a = 0$ et $b = 0]$

S se comporte alors comme une disjonction.

Cependant le fait que n soit une norme n'est pas équivalent au fait que $S/[0, 1]^2$ soit une conorme triangulaire. [Menger 42, Schweitzer 83]

En effet, si nous prenons par exemple dans R^2 la norme définie par :

$n(x, y) = |x| + |x + y|$, nous avons alors $S(1, 1) = 2$ et S non associative. En prenant $n(x, y) = |x| + 2|y|$, n est également une norme telle que S n'est pas commutative et $S(1, 1) = 3$. Enfin, réciproquement si S est la t-conorme algébrique définie par contre par $S(a, b) = a + b - ab$, alors n ne vérifie pas la propriété $[\forall \lambda, X, n(\lambda X) = |\lambda| X]$, mais vérifie l'inégalité triangulaire.

Cependant, si nous considérons les trois normes les plus utilisées pour le produit d'espaces normés, nous retrouvons comme applications S associées :

- Conorme de Zadeh $S(a, b) = \max(a, b)$ pour la norme «sup»
- Conorme de Lukasiewicz $S(a, b) = \min(1, a + b)$; elle correspond à la norme de Hamming où N_1 est définie par $N_1(x, y) = n_1(x) + n_2(y)$
- Conorme de Yager $S(a, b) = (a^2 + b^2)^{1/2}$ correspond à la norme euclidienne.

La question du choix d'une norme triangulaire pour la conjonction des prémisses d'une règle floue, correspond donc partiellement à celle du choix de la norme dans R^n à propos de la définition des voisinages flous qui va suivre.

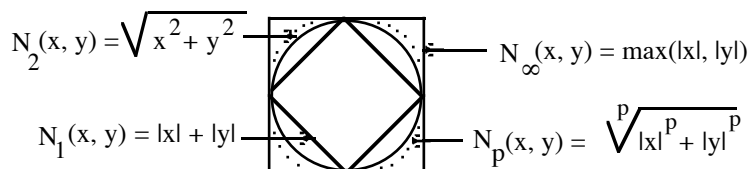


Figure 3.3 En dimension 2, la boule-unité pour ces trois normes plus la norme «p», on montre facilement que celle-ci tend vers la norme «sup» quand p tend vers l'infini.

Plaçons-nous dans $[-1, 1]^n$, et notons $B(a, r)$ la «boule floue» ou «hypercône» centrée en $a \in [-1, 1]^n$ et de rayon $r \in R^+$ le sous-ensemble flou définie par la fonction d'appartenance : $\mu_{B(a, r)}(x) = \max(0, 1 - N(a - x) / r)$

Si T est une t-norme et N la norme associée (un des trois cas précédent), on aura donc $\mu_{B(a, r)}(x) = T(\mu_{B(a_1, r)}(x_1), \dots, \mu_{B(a_n, r)}(x_n))$ où $a = (a_1, \dots, a_n)$ et $x = (x_1, \dots, x_n)$.

Voyons à présent, en dimension deux l'image d'une boule floue : soit ZE (comme zéro), le prédicat défini sur $[-1, 1]$ par la fonction d'appartenance $\mu(x) = \max(0, 1 - |x|)$, il est figuré comme coupes suivant $x = 0$ ou bien $y = 0$ de chacun des quatre figures suivantes. On définit, c'est le cas des systèmes de règles floues, la valeur de « x est ZE» et « y est ZE» par $u = T(\mu(x), \mu(y))$, ce qui donne plusieurs définitions possibles pour ZE^2 suivant les différentes t-normes employées.

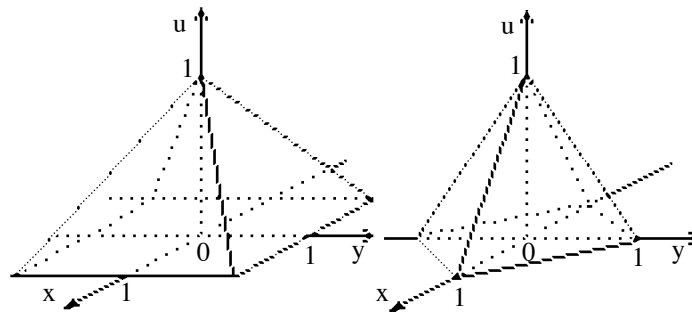


Figure 3.4 Boule-unité floue $T_{Zad}(x, y) = \min(x, y)$, $T_{Luk}(x, y) = \max(0, x + y - 1)$.

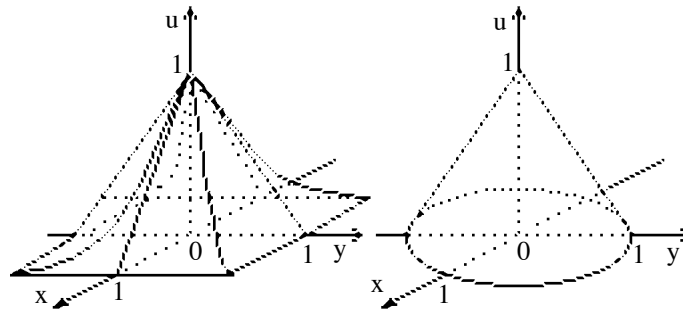


Figure 3.5 $T_{Yag}(x, y) = 1 - [(1-x)^2 + (1-y)^2]^{1/2}$, $T_{Proba}(x, y) = xy$.

3.2. Le problème de l'agrégation

Un opérateur d'agrégation en général, est une fonction H de R^n dans R croissante au sens large pour chacun de ses arguments, commutative et idempotente ce qui doit signifier pour tout nombre a , $H(a, a, a, \dots, a) = a$. Cette définition valable pour des valeurs réelles exactes, doit être étendue à des ensembles flous, car dans la pratique le problème sera la plupart du temps de prendre une décision (c'est à dire déterminer une valeur exacte) rendant compte de plusieurs valeurs précises ou non, certaines ou non, bref, de réaliser un «consensus» entre valeurs floues.

Le problème est, soit de donner une valeur de vérité à un fait lorsqu'on en a démontré plusieurs, soit, plus généralement, de donner une assignation floue à une variable lorsqu'on en connaît plusieurs. Ce problème n'est résolu qu'empiriquement dans les applications, par divers opérateurs, mais aucune option générale ne peut convenir. Néanmoins, on trouve dans toutes les approches destinées à résoudre ce problème, l'idée de renforcement, c'est à dire une accentuation de la valeur la mieux représentée, comme on l'a vu pour l'opérateur de Dempster (chapitre 2).

Dans ce qui suit, si x_1, \dots, x_n sont des valeurs pondérées par des poids w_1, \dots, w_n dont la somme est $\sum w_i = 1$, on peut définir :

LES MOYENNES GÉNÉRALISÉES

$$H(x_1, \dots, x_n) = f^{-1}(\sum w_i f(x_i))$$

Rappelons que la moyenne arithmétique r de n valeurs x_1, \dots, x_n est obtenue en prenant la fonction $f = I$, la moyenne géométrique g si $f = \ln$ (logarithme népérien), la moyenne harmonique h si $f = \text{«inverse»}$, quadratique q si $f = \text{«carré»}$, et que pour deux réels positifs a et b on a toujours les inégalités :

$$\text{Avec } \frac{2}{h} = \frac{1}{a} + \frac{1}{b}, \text{ on a : } h \leq g = \sqrt{ab} \leq r = \frac{a+b}{2} \leq q = \sqrt{\frac{a^2 + b^2}{2}}$$

Prendre l'opération de moyenne géométrique si on veut maximiser, favorise les couples moyens au détriment des couples de plus forte étendue. Par exemple pour des candidats ayant la même moyenne arithmétique 0.5 sur deux notes, celui qui possède les notes 0.5 et 0.5 aura la moyenne géométrique 0.5 alors que celui qui a obtenu 0.4 et 0.6 aura 0.49, de plus le couple de notes 0.3 et 0.7 aura 0.46, et le couple 0.1 et 0.9 obtiendra 0.3. On voit donc que tous ces couples de notes de même moyenne arithmétique, ont une moyenne géométrique d'autant plus basse que leur écart est important.

MODÉLISATION ET AGRÉGATION DE PRÉFÉRENCES FLOUES [PERNY 92]

Dans cette thèse sont étudiées différentes méthodes d'agrégation pour exprimer une décision parmi des classes $\{C_1, \dots, C_m\}$ suivant des critères notés v_1, \dots, v_n pondérés par des poids, on y trouve aussi l'idée d'accentuer les notes v_i par une fonction sigmoïde sur $[0, 1]$. A partir d'une relation binaire S sont définies des relations de :

Préférence $>_S(a, b) = \text{si } S(a, b) > S(b, a) \text{ alors } \min(S(a, b), 1 - S(b, a)) \text{ sinon } 0$

Indifférence $I_S(a, b) = \min(S(a, b), S(b, a))$

Incomparabilité $R_S(a, b) = \min(1 - S(a, b), 1 - S(b, a))$
 Non-préférence $\sim_S(a, b) = \max(I_S(a, b), R_S(a, b))$

L'OPÉRATEUR Γ DE ZIMMERMANN-ZYSNO

$Z_\gamma(x, y) = (xy)^{1-\gamma} (x + y - xy)^\gamma$, réalise, pour $0 \leq \gamma \leq 1$, un intermédiaire entre $Z_0 = T_{\text{prob}}$ et $Z_1 = S_{\text{prob}}$, et il se généralise en une opération malheureusement non associative : $H(x_1, \dots, x_n) = (\prod x_i^{w_i})^{1-\gamma} (1 - \prod(1 - x_i^{w_i}))^\gamma$ pour plusieurs variables. Néanmoins pour 1, on a, si on veut minimiser, une analogie avec la moyenne géométrique.

EXEMPLE DU PROBLÈME DE L'AGRÉGATION DANS LE DOMAINE DE L'ORDONNANCEMENT DES TÂCHES [Felix 91]

Supposons par exemple que la qualité d'une tâche est l'agrégation de trois critères MDD (date exigée), PT (temps de travail) et RO (occupation des ressources) avec comme ordres de priorités 1, 0.5, 0.7. Chacun de ces trois critères est le Z_γ avec $\gamma = 0.7$ (opérateur de Zimmermann) de deux ou trois buts.

Toute action a est évaluée par $\sum_{z \text{ fils de } a} \text{priorité}(z) (\mu_{Uz}(a) - \mu_{Bz}(a))$ et on décide de l'action ayant l'évaluation maximale. Chaque but étant associé à deux ensembles flous, son support Uz , et son ensemble d'empêchement Bz , s'il n'y a que quelques actions à choisir, chaque ensemble est discret.

Les différents buts sont en outre reliés par 14 relations comme par exemple :

$$\mu_{\text{indifférent}}(z_1, z_2) = \min[\mu_{\text{-inclu}}(U_1, U_2), \mu_{\text{-inclu}}(U_2, B_2), \mu_{\text{-inclu}}(U_1, B_2), \mu_{\text{-inclu}}(U_1, B_2)]$$

$$\mu_{\text{presque coopérant}}(z_1, z_2) = \min(\mu_{\text{-inclu}}(U_1, B_2), \mu_{\text{-inclu}}(U_2, B_1))$$

$$\mu_{\text{entrave}}(z_1, z_2) = \min(\mu_{\text{-inclu}}(U_1, U_2), \mu_{\text{inclu}}(U_1, B_2), \mu_{\text{inclu}}(U_2, B_1))$$

μ_{\pm} étant défini ici par $\text{card}(A \cap B) / \text{card}(A)$

Dans ce système, seules les relations supérieures à 0.7 sont retenues.

Le problème de l'ordonnancement des tâches n'est pas facile à étendre pour des durées incertaines et des échéances imprécises.

Ordinairement, si la tâche i possède la durée C_i avec la période T_i , le critère de charge à respecter est $\sum C_i / T_i \leq 1$.

LES OWA (ordered weighted averaging)

Si $W(w_1, \dots, w_n)$ est un vecteur de poids de somme 1 et si y_i désigne le i-ième plus grand parmi les x_i rangés dans l'ordre décroissant, [Yager 88] définit $OWA(x_1, \dots, x_n) = \sum w_i y_i$. Si chaque poids vaut $1/n$ on retrouve bien sûr la moyenne arithmétique. On a, si $W = (1, 0, \dots, 0)$, $OWA = \max$, et si $W = (0, \dots, 0, 1)$ alors $OWA = \min$. D'autre part, pour toute t-norme T et sa t-conorme associée S : $T \leq OWA \leq S$

Les OWA cherchent à considérer les meilleures valeurs parmi plusieurs. Ils formalisent plus précisément la notion de «la plupart des critères sont vérifiés», ces opérateurs peuvent donc servir à réaliser un consensus (suivant les poids donnés) plutôt des valeurs voisines, en éliminant les valeurs extrêmes.

Cependant, l'inconvénient, est qu'il faut donner un vecteur de poids, ceci peut être réalisé presque automatiquement par la méthode de [O'Hagan 90] où seul un coefficient «d'optimisme» α est demandé (plus il est proche de 1, plus les grandes valeurs seront privilégiées). La méthode consiste à maximiser $-\sum w_i \log(w_i)$ sous les contraintes $0 \leq w_i \leq 1$, $\sum w_i = 1$, $\sum w_i(n-i)/(n-1) = \alpha$.

LES INTÉGRALES FLOUES DE CHOQUET OU SUGENO

Si c est une mesure sur l'ensemble des critères ou attributs $\{A_1, \dots, A_n\}$ et si f est la fonction discrète définie par les degrés de certitudes (x_1, \dots, x_n) dans l'ordre croissant sur les A_i , alors :

$$H(x_1, \dots, x_n) = \int_{\text{Sug}} f \circ c = \max_{1 \leq i \leq n} \min(f(x_i), c\{x_i, x_{i+1}, \dots, x_n\})$$

On a donc une généralisation au cas continu :

$$\int_{\text{Sug}} f \circ c = \sup_{\alpha \in [0, 1]} \min(\alpha, c\{x / f(x) > \alpha\})$$

De plus on peut accorder des poids w_i aux critères A_i suivant lesquels on veut décider.

INTÉGRALE FLOUE DE SUGENO [Sugeno 74]

Si f est une application de X un espace mesuré avec la mesure c , dans $[0, 1]$ où c est une «mesure floue» de confiance, on définit l'intégrale de Sugeno de f comme :

$$\int_{\text{Sug}} f \circ c = \sup_{\alpha \in [0, 1]} \min(\alpha, c\{x / f(x) > \alpha\})$$

Si X est discret et si f est croissante, l'intégrale de Sugeno est alors calculée par :

$\max_{1 < i < n} \min(f(x_i), c(A_i))$ avec $A_i = \{x_i, x_{i+1}, \dots, x_n\}$. Au cas où f est l'identité et c uniforme ($c(I)$ est la longueur d'un intervalle I), alors l'intégrale donne le terme médian (le k-ième terme s'il y a $2k-1$ termes discrets).

Si f n'est pas croissante, la définition est la même à condition de ranger préalablement les valeurs prises par f dans l'ordre croissant.

L'intégrale de Choquet [Choquet 53] consiste en une autre définition, qui, elle, coïncide avec l'intégrale de Lebesgue dans le cas où c est additive, c'est à dire :

$$c(A \cup B) = c(A) + c(B)$$

$$\int_{\text{Cho}} f \circ c = \int_{[0, 1]} c\{x / f(x) > \alpha\} d\alpha = a_1 + (a_2 - a_1)c(A_2) + (a_3 - a_2)c(A_3) + \dots \text{ dans le cas discret.}$$

REMARQUE Une mesure est dite décomposable s'il existe une t-conorme S telle que :

$c(A \cup B) = S[c(A), c(B)]$, en particulier $c(A \cup B) = c(A) + c(B) + \lambda c(A)c(B)$ avec le coefficient $\lambda > -1$ qui sont les λ -mesures de Sugeno.

PROPRIÉTÉS

Les deux opérateurs sont croissants et continus mais non linéaires.

Si c est une mesure de possibilité (nécessité) et f la fonction d'appartenance à un ensemble flou, alors l'intégrale de Sugeno est la possibilité (nécessité) de cet ensemble flou.

Si f est constante égale à k , alors les deux intégrales Choquet et Sugeno ont bien la valeur k .

Si $c \leq c'$ alors $\int_{\text{Sug}} f \circ c \leq \int_{\text{Sug}} f \circ c'$ et $\int_{\text{Cho}} f \circ c \leq \int_{\text{Cho}} f \circ c'$

$\inf_X (f) \leq \int f \circ c \leq \sup_X (f)$ pour les deux définitions car on retrouve inf et sup pour deux mesures particulières que sont $c_0(\Omega) = 1$ et 0 ailleurs, et $c_1(\emptyset) = 0$ et 1 ailleurs.

Continuité pour les deux définitions : si $\lim (f_n) = f$ sur R , alors $\lim \int f_n \circ c = \int f \circ c$

Exercice 3.11

Calculer les deux intégrales pour $f(x) = x$ puis x^2 avec $c([a, b]) = (b-a)^2$ sur $[0, 1]$

APPLICATION À L'ANALYSE MULTICRITÈRE [Grabisch 93]

On évalue chaque objet, par exemple une voiture, par son degré de performance à l'égard d'un certain nombre de critères (prix, confort, consommation...) dont les attributs sont définis par des ensembles flous. Un utilisateur donnant ses poids à chacun de ces critères, on cherchera l'objet ayant la plus forte intégrale avec la mesure floue définie sur l'ensemble des critères.

APPLICATION À UNE MESURE DU FLOU [Benvenuti 94]

On peut définir $A < B$ (A moins flou que B) par $\mu_A \leq \mu_B \leq 0.5$ ou $0.5 \leq \mu_B \leq \mu_A$, grâce à cette définition on peut montrer que c est un ordre partiel et que les ensembles exacts sont minimaux et l'ensemble (noté $E/2$) défini par $\mu = 0.5$ est l'élément maximal unique. D'autre part le sup de deux ensembles existe toujours, mais pas l'inf.

En prenant une fonction f croissante sur $[0, 0.5]$ décroissante sur $[0.5, 1]$ et symétrique par rapport à 0.5, vérifiant $f(0.5) = 1$ et $f(0) = f(1) = 0$, un degré de flou peut aussi être défini par :

$$d(F) = \int_{\text{Sug}} f \circ \mu_F \circ c = \sup_{\alpha \in [0, 1]} \min(\alpha, c\{x / \mu_F(x) > \alpha\})$$

Exercice 3.12

Montrer que $d(F) = 0 \Leftrightarrow F$ ensemble exact, $d(E/2) = 1$, $d(\neg F) = d(F)$, et que d est croissante vis à vis de cette relation «moins flou que»:

$$A < B \Rightarrow d(A) < d(B)$$

AGRÉGATION D'ENSEMBLES FLOUS

On peut utiliser une des méthodes ci-dessus appliquées aux valeurs des fonctions d'appartenance pour chaque x . Cependant si $h = \sup \min(\mu_A, \mu_B)$ est le degré de chevauchement de deux ensembles flous A et B , [Dubois, Prade 92, 94] proposent un opérateur d'agrégation donnant une distribution normalisée réalisant d'autant plus la conjonction que le conflit est faible (h proche de 1), et d'autant plus la disjonction que le conflit est fort (h proche de 0).

$$\mu_A * \mu_B = \max [\min(\mu_A, \mu_B) / h, \min((1-h), \max(\mu_A, \mu_B))]$$

L'inconvénient est que cette formule n'est pas associative et surtout que l'ensemble flou produit peut ne pas être convexe, le pic à 1 peut fort bien être encadré de deux minima puis de deux plateaux plus élevés. Une extension de la formule à plusieurs distributions est néanmoins donnée de la façon suivante. Si $\mu_1, \mu_2, \dots, \mu_r$ sont des ensembles flous normalisés sur un univers U , on pose m le nombre maximal de ces sources présentant une concordance totale, c'est à dire :

$m = \max \{\text{card}(I) / I \subseteq \{1, \dots, r\} \text{ et } \sup(\inf\{\mu_i / i \in I\}) = 1\}$, et n le nombre maximal de sources présentant une certaine concordance (un mutuel chevauchement), soit $n = \max \{\text{card}(I) / I \subseteq \{1, \dots, r\} \text{ et } \sup(\inf\{\mu_i / i \in I\}) > 0\}$. On a donc $m \leq n$ et on pose :

$\pi_k(x) = \max \{\min \{\mu_i(x) / i \in I\} / \text{card}(I) = k \text{ et } I \subseteq \{1, \dots, r\}\}$ la meilleure possibilité obtenue avec k sources sans les préciser au sein des $\mu_1, \mu_2, \dots, \mu_r$.

Le plus grand degré de concordance établi sur n ensembles flous parmi $\mu_1, \mu_2, \dots, \mu_r$, ayant une certaine concordance est :

$$h = \max \{\sup(\{\min \{\mu_i(x) / i \in I\} / x \in U\}) / \text{card}(I) = n \text{ et } I \subseteq \{1, \dots, r\}\}$$

La fonction agrégée des $\mu_1, \mu_2, \dots, \mu_r$ sur l'univers U est alors : $Ag(x) = \max [(\pi_n(x) / h, \min((1-h), \pi_m(x)))]$.

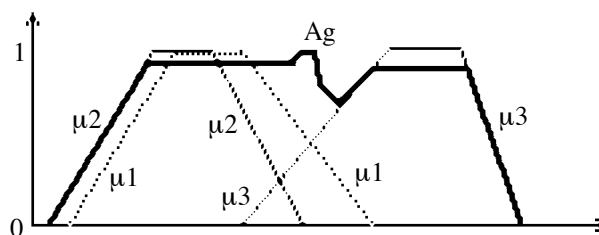


Figure 3.6 Agrégation Ag de deux ensembles flous suivant Dubois-Prade

Cette méthode est utilisée pour la fusion de capteurs donnant une évaluation de distance non fiable [Oussalah, Maaref, Barret 96].

Dans le même ordre d'idée, [Kelman, Yager 95], en considérant un opérateur d'agrégation H sur R^n , formulent l'agrégation des ensembles flous A_1, \dots, A_n par $H(A_1, \dots, A_n)$ d'après une généralisation du principe d'extension de Zadeh, comme un ensemble flou B de fonction :

$$\mu_B(H(x_1, \dots, x_n)) = \min(\mu_{A_1}(x_1), \dots, \mu_{A_n}(x_n), R(x_1, x_2), R(x_1, x_3), \dots, R(x_{n-1}, x_n))$$

au moyen de R une «mesure de compatibilité» qui est une relation floue réflexive et symétrique et vérifiant $[d(x, y) < d(x, z) \text{ alors } R(x, y) > R(x, z)]$ si d est une distance sur l'univers considéré.

Au cas où R est la relation toujours égale à 1, on retrouve le principe d'extension.

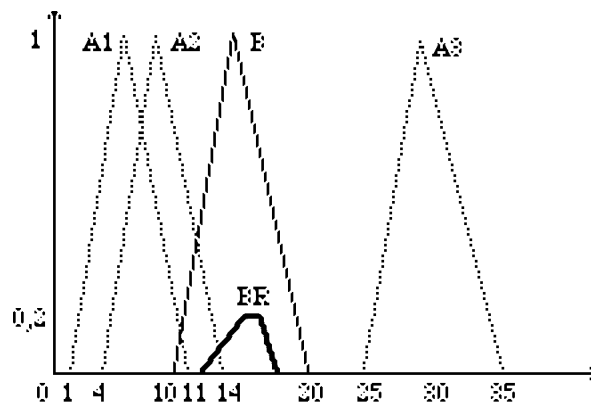


Figure 3.7 En prenant l'exemple de $R(x, y) = \max(0, 1 - 0.05|x - y|)$ dans l'intervalle $[0, 40]$, on obtient un ensemble flou BR de plus grande valeur 0.2 suivant cette compatibilité, et une agrégation normalisée sans la considérer.

Exercice 3.13

On définit une somme symétrique comme une application de $[0, 1]^P$ vers $[0, 1]$ continue et croissante pour chacun de ses arguments, commutative (ne dépendant pas de l'ordre de ses arguments) vérifiant :
 $S(0, \dots, 0) = 0$ et $S(1, \dots, 1) = 1$. S est dite auto-duale si elle vérifie de plus :
 $S(1 - x, 1 - y, 1 - z, \dots) = 1 - S(x, y, z, \dots)$. Trouver une telle fonction.

Signalons pour finir que pour les couples de $[0, 1]$, cette fois, l'opérateur de Dempster réalise une sorte de «moyenne» qui en fait renforce les valeurs données, soit vers le vrai, soit vers le faux. Dempster est associatif mais non idempotent (voir chapitre 2).

3.3. Différentes formes de l'implication

En désignant par le symbole \rightarrow l'implication booléenne classique, nous passons en revue les différentes généralisations, toutes notées de la même façon. On peut définir une implication associée à une t-norme T par :
 $(p \rightarrow q) = \sup \{r / T(p, r) \leq q\}$ ainsi Gödel-Sanchez, Goguen et Lukasiewicz sont-ils associés aux normes du même nom. Ces opérateurs vérifient pour la plupart la relation $p \leq q \Leftrightarrow (p \rightarrow q) = 1$ et aussi la définition classique $(p \rightarrow q) = \neg(T(p, \neg q))$

DÉFINITION DE REICHENBACH $(P \rightarrow Q) = 1 - P + P.Q$ résulte de $(p \rightarrow q) = \neg p \vee q$ mais avec le produit pour la conjonction.

DÉFINITION DE WILLMOTT $(P \rightarrow Q) = \max(1 - p, \min(p, q))$ c'est une implication tempérée par $(p \vee \neg p) \geq 0.5$

DÉFINITION DE MAMDANI

$(p \rightarrow q) = \min(p, q)$ caractérisée par une symétrie, cette définition pose problème car, ainsi que Larsen $(p \rightarrow q) = p.q$, ce n'est pas une généralisation de l'implication booléenne.

DÉFINITION DE RESCHER ET DE PRZYMUSINSKI $(P \rightarrow Q) = 1$ si $p \leq q$ sinon 0

DÉFINITION DE KLEENE ET DIENES

On généralise simplement la logique binaire classique où $p \rightarrow q$ est légitimé en disant que p ne peut être vrai sans que q le soit, en d'autres termes, il faut que $p \wedge \neg q$ soit faux, on a donc :

$(p \rightarrow q) = \neg(p \wedge \neg q) = \neg p \vee q = \max(1 - p, q)$ résulte de $(p \rightarrow q) = \neg p \vee q$ en logique min-max.

DÉFINITION DE GÖDEL (reprise par Brouwer) $(P \rightarrow Q) = 1$ si $p \leq q$ sinon q

Elle est associée aux t-normes min ou de Weber pour : $(p \rightarrow q) = \sup \{r / T(p, r) \leq q\}$.

Cette définition est utilisée dans le système-expert Diabeto [Buisson, Farreny, Prade 86], non pour donner une valeur de vérité à la proposition $(p \rightarrow q)$, mais pour donner une distribution de possibilité de q conditionné par p.

DÉFINITION PROPOSÉE PAR GOGUEN (probabiliste)

$(p \rightarrow q) = [si\ p = 0\ alors\ 1\ sinon\ \min(1, q / p)] = [si\ p \leq q\ alors\ 1\ sinon\ q / p]$.

Il est à noter que cette formule coïncide avec celle de Lukasiewicz, mais uniquement dans le cas de la logique à trois valeurs.

L'intérêt de cette définition réside dans le fait que la résolution de l'équation en q :

$(p \rightarrow q) = r$ se fait facilement par : $q = si\ r < 1\ alors\ p.r\ sinon\ [p, 1]$

Cette question est en effet importante en ce qui concerne le mécanisme d'inférence d'une règle $p \rightarrow q$ pondérée par un coefficient dans un système-expert.

DÉFINITION DE LUKASIEWICZ $(P \rightarrow Q) = \min(1, 1 - p + q) = 1 - \max(0, p - q)$

(Reprise par Zadeh et utilisée dans le système-expert Protis)

Cette formule a l'avantage de respecter la contraposition $(\neg q \rightarrow \neg p) = (p \rightarrow q)$

Dans un système multivalué comportant m valeurs t_0, t_1, \dots, t_{m-1} elle se définit par :

$$t_i \rightarrow t_j = [si (i \leq j) \text{ alors } t_{(m-1)} \text{ sinon } t_{(m-i+j-1)}]$$

AUTRES DÉFINITIONS

On trouve encore $(p \rightarrow q) = (p \wedge q) \vee (1 - p)$ définition de Zadeh analogue à celle de Wilmott, et :

$$(p \rightarrow q) = [(p \wedge q) \wedge (1 - p)] \vee [q \vee (1 - p)] \text{ [Turksen 88]}$$

$$(p \rightarrow q) = p \cdot q \text{ (Opérateur de Larsen)}$$

$$(p \rightarrow q) = p(1 - lp - ql) \text{ [Vincent, Dujet 94] qui généralise Mamdani.}$$

$$(p \rightarrow q) = q^p \text{ généralise aussi l'implication booléenne avec la convention } 0^0 = 1.$$

REMARQUE Les définitions de Mamdani et Larsen ne sont pas à proprement parler des implications car elles ne généralisent pas l'implication booléenne.

Pour $p = 0$ et $q = 1$ on doit en effet avoir $(p \rightarrow q) = 1$ et non pas 0.

EXEMPLES D'IMPLICATIONS AVEC 3, 4, OU 5 VALEURS DE VÉRITÉ

	P	Q	KLEENE	LUKASIEWICZ	GODEL
3 valeurs	1	Q	Q	Q	Q
	1/2	1	1	1	1
	1/2	1/2	1/2	1	1
	1/2	0	1/2	1/2	0
	0	Q	1	1	1
4 valeurs	1	Q	Q	Q	Q
	2/3	1	1	1	1
	2/3	2/3	2/3	1	1
	2/3	1/3	1/3	2/3	1/3
	2/3	0	1/3	1/3	0
	1/3	1	1	1	1
	1/3	2/3	2/3	1	1
	1/3	1/3	2/3	1	1
	1/3	0	2/3	2/3	0
5 valeurs	0	Q	1	1	1
	1	Q	Q	Q	Q
	3/4	1	1	1	1
	3/4	3/4	3/4	1	1
	3/4	1/2	1/2	3/4	1/2
	3/4	1/4	1/4	1/2	1/4
	3/4	0	1/4	1/4	0
	1/2	1	1	1	1
	1/2	3/4	3/4	1	1
	1/2	1/2	1/2	1	1
	1/2	1/4	1/2	3/4	1/4
	1/2	0	1/2	1/2	0
	1/4	1	1	1	1
	1/4	3/4	3/4	1	1
	1/4	1/2	3/4	1	1
1/4	1/4	3/4	1	1	
1/4	0	3/4	1	1	
0	Q	1	1	1	

Exercice 3.14

Comme on le voit, le résultat peut être fort différent pour ces trois implications. Cependant on peut constater que l'implication de Lukasiewicz est toujours supérieure aux deux autres, montrer en étudiant les formules que cela est toujours vrai.

CONTRAPOSITION

On vérifie facilement que pour les fonctions booléennes qui définissent l'implication au sens de Kleene ou de Lukasiewicz, l'égalité:

$$(p \rightarrow q) = (\neg q \rightarrow \neg p) \text{ est alors assurée.}$$

Par contre la définition de Gödel n'assure plus la contraposition car :

$$(\neg q \rightarrow \neg p) = si p \leq q \text{ alors } 1 \text{ sinon } 1 - p$$

$$(p \rightarrow q) = si p \leq q \text{ alors } 1 \text{ sinon } 1 - q$$

Si on se tient seulement à la théorie des possibilités et si P est connu avec un intervalle [x, y] et une règle P → Q avec un degré de suffisance [s, ...], par modus ponens $nc(Q) \geq \min(x, s)$, par modus tollens $nc(\neg Q) = 1 - ps(Q)$ donc $ps(P) \leq \max(1 - s, ps(Q))$ et c'est tout ce qui peut être dit.

Si un degré de nécessité [n,...] est donné pour la règle inverse Q → P c'est à dire

$\neg P \rightarrow \neg Q$, alors $nc(\neg Q) = 1 - ps(Q) \geq \min(nc(\neg P), n)$ soit $ps(Q) \leq S(y, 1-n)$

L'approche Bayésienne, préférée par certains systèmes, consiste à attribuer des coefficients [s, S] et [r, R] pour les règles respectives P → Q et $\neg P \rightarrow Q$ alors par Bayes : $nc(Q) = S(T(s, y), T(r, 1 - y)) \leq ps(Q) = S(T(S, y), T(R, 1 - x))$

En posant sous forme de possibilité conditionnelle $ps(p \text{ et } q) = \min(ps(p), ps(q / p))$ et $nc(q / p) = 1 - ps(\neg q / p)$ on obtient une écriture matricielle (mais avec les opérations min-max) :

$$\begin{bmatrix} ps(Q) \\ ps(\neg Q) \end{bmatrix} = \begin{bmatrix} ps(Q | P) & ps(Q | \neg P) \\ ps(\neg Q | P) & ps(\neg Q | \neg P) \end{bmatrix} \begin{bmatrix} ps(P) \\ ps(\neg P) \end{bmatrix}$$

C'est la relation de Bayes transposée aux coefficients de possibilités pour Q et pour $\neg Q$ qui s'écrit matriciellement [Farreny, Prade, Wyss 86].

LE PROBLÈME DE LA RÉOLUTION (P → Q) = R

Aucune définition de $p \rightarrow q$ ne donne de réponse simple à la solution q de l'équation $(p \rightarrow q) = r$, seule la formule de Goguen peut donner une borne inférieure $q = p * r$ d'un calcul pratique, c'est ce qui a été utilisé dans certains systèmes-experts (coefficients de vraisemblance).

Une autre borne inférieure utilisée est $q = \min(p, r)$, et c'est avec la définition de Gödel que cette formule s'accorde le mieux.

LE POINT DE VUE APPORTÉ PAR LA THÉORIE DES RELATIONS FLOUES

Dans [Kaufmann 77] le calcul propositionnel est également introduit à partir de :

$$\neg p = 1 - p \quad p \wedge q = \min(p, q) \quad p \vee q = \max(p, q)$$

La recherche des «implications» a conduit à étudier les relations floues entre deux ensembles X, Y : si les propositions sont des sous-ensembles flous, donner une définition de $A \rightarrow B$ où A, B sont sous-ensembles flous respectivement de X, Y; équivaut à se donner une relation, dite de causalité, R permettant de trouver B connaissant A, c'est-à-dire : $R(A) = B$ (voir aussi [Turksen 89])

C'est alors la définition de Gödel qui joue le plus grand rôle, en effet si on définit \rightarrow l'opérateur de Sanchez-Gödel par : $(x \rightarrow y) = [si\ x \leq y\ alors\ 1\ sinon\ y]$

la composition de deux relations floues S entre X, Y puis R entre Y, Z par :

$$\mu_{R \circ S}(x, z) = \max\{ \min(\mu_S(x, y), \mu_R(y, z)) / y \in Y \}$$

et enfin l' α -composition par $\mu_{S \rightarrow R}(x, z) = \min\{ \mu_S(x, y) \rightarrow \mu_R(y, z) / y \in Y \}$

on démontre les théorèmes suivants :

1° Si Q et T sont deux relations floues entre X et respectivement Y et Z, et s'il existe une solution R à l'équation $R \circ Q = T$ alors il existe une solution maximale de Y dans Z qui est $Q^{-1} \rightarrow T$

2° Si A, B sous-ensembles flous respectivement de X, Y et s'il existe une relation floue R telle que $R(A) = B$ alors il existe une telle relation maximale qui est $(A \rightarrow B)$ où $\mu_{A \rightarrow B}(x, y) = (\mu_A(x) \rightarrow \mu_B(y))$

3° Si R et T sont deux relations floues entre respectivement Y et X, et Z, et s'il existe une solution Q à l'équation $R \circ Q = T$ alors il existe une solution maximale de X dans Y qui est $(R \rightarrow T^{-1})^{-1}$

C'est le deuxième résultat qui nous intéresse. Ainsi par exemple : si $X = \{x_1, \dots, x_5\}$ et $Y = \{y_1, y_2, y_3\}$, que la «cause» soit le sous-ensemble flou :

$A = \{x_1|0.8, x_2|1, x_3|0, x_4|0.3, x_5|0.5\}$, et $B = \{y_1|0.9, y_2|0.2, y_3|0.7\}$ l'effet, alors $R = (A \rightarrow B)$ est donné par le tableau :

α	y1	y2	y3
x1	1	0.2	0.7
x2	0.9	0.2	0.7
x3	1	1	1
x4	1	0.2	1
x5	1	0.2	1

On vérifie d'ailleurs que la matrice-ligne A multipliée (min-max) par la matrice ci-dessus, donne la matrice-ligne B.

UNE GÉNÉRALISATION DE L'ÉQUIVALENCE

En attachant à chaque règle $P \Rightarrow Q$ (éventuellement $A(x) \Rightarrow B(y)$) un couple (s, n) représentant cette fois les plausibilités de la règle et de la règle inverse on cherche la plausibilité pour un couple de valeurs précises (x, y) d'avoir B(y) conditionné par A(x), en d'autres termes pour une règle de la forme : si «X est A» alors $_{(s,n)}$ «Y est B».

On peut proposer [Gacôgne 90] : $\begin{bmatrix} ps(Q) \\ ps(\neg Q) \end{bmatrix} = \begin{bmatrix} 1 & 1-n \\ 1-s & 1 \end{bmatrix} \begin{bmatrix} ps(P) \\ ps(\neg P) \end{bmatrix}$

Ce qui conduit à :

$$ps(Q) = \max(ps(P), 1 - \max(n, nc(P))) \text{ et } nc(Q) = \min(nc(P), \max(s, 1 - ps(P)))$$

Les différents systèmes existants utilisent des généralisations de l'implication de Kleene ou de Gödel en distinguant les règles suivant des conditions certaines, ou floues et une conclusion incertaine floue ou non floue [Buisson Farreny Prade 86]. Si on choisit

comme dans [Martin-Clouaire, Prade 86], une définition qui tente une généralisation comme la plus simple fonction π donnant les quatre valeurs du tableau suivant dans lequel on compare avec l'implication $P \rightarrow Q$ classique.

p	q	$p \rightarrow q$	$p \leftarrow q$	$p \leftrightarrow q$	$p \leftrightarrow_{s,n} q$
1	1	1	1	1	1
1	0	0	1	0	1 - s
0	1	1	0	0	1 - n
0	0	1	1	1	1

On voit donc seulement avec ces quatre cas de la logique binaire classique, que plus s est grand et n petit, plus on se rapproche de l'implication et plus s et n sont grands, plus on se rapproche de l'équivalence.

Précisément, si s = 1 et n = 0 on a l'implication, si s = 0 et n = 1, on a l'implication inverse, et pour s = n = 1, on retrouve l'équivalence. Il s'agit donc d'une généralisation de l'équivalence pouvant trouver son utilité dans la mesure où le raisonnement dit de «sens commun» s'intéresse toujours lors d'une relation de cause à effet, à la relation inverse.

L'APPROCHE DES VOISINAGES [AKDAG 92]

Dans une logique multivaluée finie, on prend généralement 7 valeurs linguistiques : $L_7 = \{\text{faux, peuvrai, assezvrai, moyen, plutôtvrai, trèsvrai, vrai}\}$.

$L_n = \{t_0, \dots, t_{n-1}\}$ est muni d'un ordre total $t_i < t_j$ si $i < j$, du min et max et de la négation $\neg t_i = t_{n-1-i}$.

L'implication choisie est celle de Lukasiewicz ($t_i \rightarrow t_j = t_{[n-1-\max(0, i-j)]}$) = si $i \leq j$ alors t_{n-1} sinon $t_{n-1-i+j}$

Grâce à une définition de voisinage $x \forall_i y \Leftrightarrow (x \rightarrow y) \geq i$ et $(y \rightarrow x) \geq i$, le t_k -voisinage de t_i est alors $\{t_j / \max(0, k-n+1+i \leq j \leq n-1+i-k)\}$

3.4. Le modus ponens généralisé

Le modus ponens généralisé consiste en la donnée d'une règle $(X \text{ est } A) \rightarrow (Y \text{ est } B)$ où A et B sont des relations floues unaires, et d'un fait X est A', on en déduit Y est B', naturellement suivant certaines modalités où A' étant «proche» de A, alors B' est construit comme «voisin» de B.

Pour une t-norme T et un opérateur π d'implication défini par :

$$\pi(x, y) = (\mu_A(x) \rightarrow \mu_B(y))$$

la définition donnée par [Zadeh 73] en est $\mu_{B'}(y) = \sup_{x \in U} (T(\mu_{A'}(x), \pi(x, y)))$.

$$\mu_{B'}(y) = \sup_{x \in \text{supp}(A')} (T(\mu_{A'}(x), \mu_A(x) \rightarrow \mu_B(y)))$$

On peut voir l'analogie avec la formule probabiliste $p(B) = \sum p(A_i).p(B/A_i)$, mais aussi une définition résultant du principe d'extension de Zadeh.

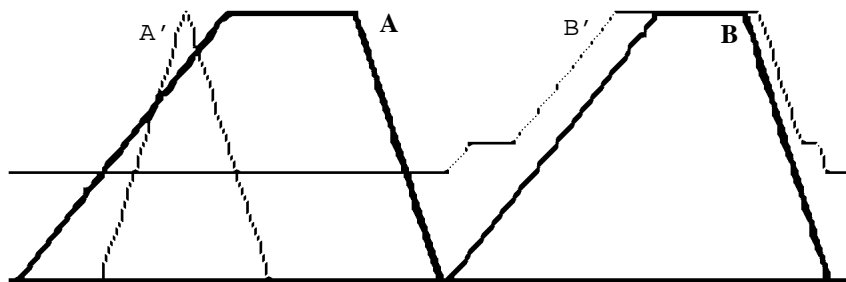


Figure 3.8 Exemple avec la t-norme de Zadeh et l'implication de Lukasiewicz

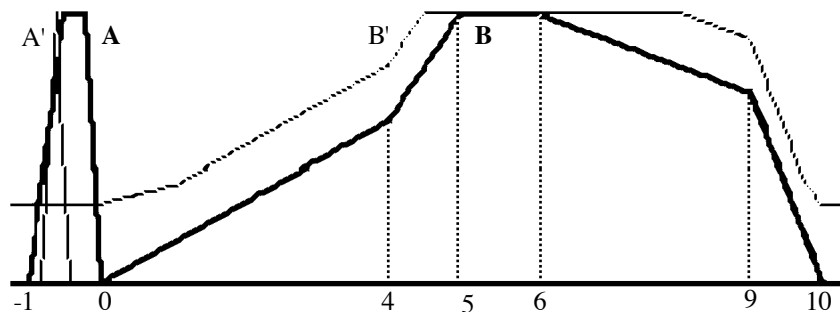


Figure 3.9 Exemple pour la t-norme Hamacher et l'implication de Lukasiewicz

Exercice 3.15

Pour $A' = A$ puis A' coupant A triangulaire et B triangulaire, représenter B' avec π de Mamdani, Lukasiewicz, Kleene et Gödel.

Exercice 3.16

Comparaison des implications lors d'inférence floue (X est A) \rightarrow (Y est B)

Montrer que si $\mu_{A'} \leq \mu_A$ alors on a $\mu_{B'} = \mu_B$ avec Kleene, Gödel, Lukasiewicz, Reichenbach, Willmott, Mamdani, tandis que l'on a $\mu_{B'} \leq \mu_B$ pour Rescher.

Si maintenant on a $\mu_{A'} \geq \mu_A$ alors $\mu_{B'} \geq \mu_B$ pour Kleene, Gödel, Goguen, Lukasiewicz, Reichenbach, Willmot, alors que $\mu_{B'} = \mu_B$ pour Mamdani.

Exercice 3.17

Au cas d'une observation précise, c'est à dire qu'il existe x_0 tel que $\mu_{A'}(x_0) = 1$ et $\mu_{A'}(x) = 0$ pour tout autre x , montrer alors qu'il y a incertitude avec Kleene, Lukasiewicz, Reichenbach, Willmott, mais pas avec Mamdani, et pas avec Gödel, Goguen et Rescher dans le cas où $\mu_A(x_0) \neq 0$ pour ces trois derniers.

Exercice 3.18

Montrer qu'en général, si $\mu = ps$ (A' est A) alors, en dehors du support de B , B' vaut $1 - \mu$ pour Kleene et Reichenbach.

Exercice 3.19

Cas d'une conclusion précise : B est défini par $\mu(y_0) = 1$ et si $y \neq y_0$ alors $\mu(y) = 0$, en ce cas $\pi(x, y_0) = \mu_A(x)$ pour Mamdani, et $\pi(x, y_0) = 1$ sauf pour Willmott et Mamdani.

Une étude systématique a été réalisée dans [Després 88].

ALGORITHME APPROCHÉ DE CALCUL DU MODUS PONENS GÉNÉRALISÉ, DANS LE CAS DES λ -TRAPÈZES

Etant donnés les λ -trapèzes $A = (a_A, b_A, \alpha_A, \beta_A, \lambda_A)$, $B = (a_B, b_B, \alpha_B, \beta_B, \lambda_B)$, intervenant dans la règle $A \rightarrow B$, et la valeur floue $A' = (a_{A'}, b_{A'}, \alpha_{A'}, \beta_{A'}, \lambda_{A'})$ approchant A , l'algorithme renvoie un λ -trapèze $B' = (a_{B'}, b_{B'}, \alpha_{B'}, \beta_{B'}, \lambda_{B'})$.

On calcule les niveaux $\xi = \sup \{\mu_{A'} / \mu_A = 0\}$, $\zeta = \inf \{\mu_A / \mu_{A'} = 1\}$. Un autre indicateur a été utilisé, c'est $\eta = \inf \{\mu_A / \mu_{A'} \geq ps$ (X est A)} sorte de degré de proximité.

Dans le cas où on prend la conjonction min et l'implication de Gödel, on examine dans l'ordre les cas :

Conditions :

Si $A' \subseteq A$ alors $B' = (a_B, b_B, \alpha_B, \beta_B, \max(\lambda_B, \lambda_{A'}))$

Si $A \cap A' = \emptyset$ alors $B' = (a_B, b_B, \alpha_B, \beta_B, 1)$

Si $\text{supp}(A') \subseteq \text{supp}(A)$, $B' = (a_B - (1-\zeta)\alpha_B, b_B + (1-\zeta)\beta_B, \zeta\alpha_B, \zeta\beta_B, \max(\lambda_B, \lambda_{A'}))$

Si $\text{noy}(A') \subseteq \text{noy}(A)$ alors $B' = (a_B, b_B, \alpha_B, \beta_B, \max(\xi, \lambda_{A'}))$

Sinon $B' = (a_B - (1 - \zeta)\alpha_B, b_B + (1 - \zeta)\beta_B, \alpha_B, \beta_B, \max(\xi, \lambda_{A'}))$

RÈGLES GRADUELLES

Aucune formalisation efficace et convaincante de règles telles que «plus X est A , alors plus Y est B », n'a encore été réalisée. On peut les représenter par plusieurs règles utilisant une hiérarchie petit, moyen, grand etc pour X ainsi que pour Y , mais on voudrait bien sûr éviter les problèmes de seuil pour passer d'une règle à l'autre. D. Dubois distingue par ailleurs : «plus X est A , plus il est certain que Y est B » avec par exemple «plus on se lève tard, plus on est certain de rater son train» qui est le plus souvent utilisé avec l'implication de Kleene $IK(p, q) = \max(1 - p, q)$, le modus-ponens généralisé donnant une distribution de possibilité. En ce cas, si l'agrégation se fait avec le min, il suffit que l'une des règles donne 0 pour interdire les autres.

Par ailleurs «plus X est A , plus il est possible que Y soit B » avec par exemple : «plus 2 voitures sont semblables, plus il est possible que leurs prix soient identiques», où on utilisera plutôt l'implication de Gödel $IG(p, q) = \text{si } (p < q) \text{ alors } 1 \text{ sinon } q$.

Exercice 3.20

On rappelle que l'image d'un ensemble flou A par une relation R est un ensemble flou B défini par :

$\mu_B(y) = \sup \{\min(\mu_A(x), \mu_R(x, y))\}$. Vérifier que pour les α -coupes, $B_\alpha = R_\alpha(A_\alpha)$ et que cela correspond à un ensemble de règles «si $x \in A_\alpha$ alors $y \in B_\alpha$ ».

En prenant l'implication de Gödel $(p \rightarrow q) = (\text{si } p \leq q \text{ alors } 1 \text{ sinon } q)$, montrer qu'alors $R \subseteq [A \rightarrow R(A)]$.

INTERPOLATION ENTRE LES RÈGLES

Pour une collection de n règles «si X est A_i , alors Y est B_i », deux points de vue peuvent être défendus.

a) Le point de vue logique qui consiste à prendre la conjonction des règles :

$x R^+ y = T (1 \leq i \leq n) (\mu_{A_i}(x) \rightarrow \mu_{B_i}(y))$ où T est une t -norme et \rightarrow une définition d'implication. Les règles produisent des degrés de certitudes, $\mu_{B'}(y) = \min_{1 \leq i \leq n} (\max(1 - \mu_{A_i}(x), \mu_{B_i}(y)))$ et pour tout i on doit avoir $B' \subseteq B_i$, et pour une collection de règles certaines, la relation R_f entre x et y est telle que pour tout i on ait $A_i \circ R_f \subseteq B_i$ d'où vient $R_f = \bigcap (\neg A_i \cup B_i)$ encadrement noir de la figure.

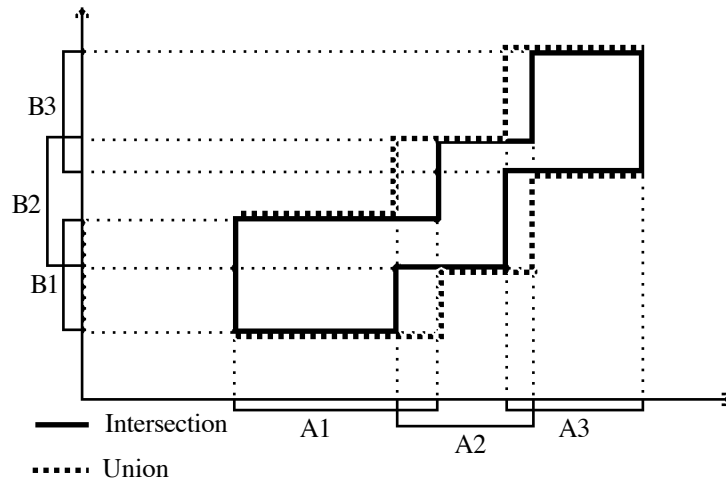


Figure 3.10 Conjonction ou disjonction de règles se chevauchant.

b) Le point de vue de la méthode de Mamdani (c'est le contrôle flou du chapitre suivant) :

$x R y = S (\mu_{A_i}(x) T \mu_{B_i}(y))$ où S est une t -conorme et $T = \min$, consiste à prendre la disjonction des règles, donc l'union des conclusions ce qui peut parfaitement se justifier dans la mesure où il considère que chaque règle peut apporter une réponse partielle et que c'est leur totalité qui doit apporter une réponse consensuelle. Alors les règles produisent des degrés de possibilités $\mu_B(y) =$

$\max_{(1 \leq i \leq n)} (\min (\mu_{A_i}(x), \mu_{B_i}(y)))$ et pour tout i on doit avoir $B_i \subseteq B$. Ce que fait Mamdani est de prendre la relation $R = \bigcup (A_i * B_i)$ (produit cartésien) correspondant au pourtour pointillé.

Pour des entrées précises x :

$$\mu_{B_f}(y) = \sup_{x \in A} (\min(\mu_A(x), \min_{1 \leq i \leq n} \pi_i(x, y)))$$

$$\mu_{B_g}(y) = \sup_{x \in A} (\min(\mu_A(x), \max_{1 \leq i \leq n} \min(\mu_{A_i}(x), \mu_{B_i}(y))))$$

APPLICATION AUX SYSTÈMES-EXPERTS FLOUS

Dans les systèmes experts flous, si on trouve deux assignations μ et μ' pour une conclusion, et leur compatibilité $h = \sup \min(\mu, \mu')$, si elle n'est pas nulle, permet de prendre par exemple $\mu_{ag} = \min(\mu, \mu') / h$ comme agrégation possible de la conclusion.

En fait, aucun problème concret lié aux systèmes-experts, comme on l'a déjà souligné, n'est résolu de façon universelle. Comment l'utilisateur doit-il donner ses appréciations (une échelle linguistique de 5 ou 7 valeurs est largement suffisante), mais ensuite comment doit se réaliser la conjonction des hypothèses (quelle t -norme choisir ?), comment opérer l'inférence ?, comment doit-on gérer les contradictions et réaliser l'agrégation des différentes conclusions ?

Chapitre 4

Contrôle flou

Le contrôle flou tire son nom des applications de contrôle ou de commande en automatique, mais il déborde maintenant ce cadre par ses multiples applications, partout où une modélisation mathématique est difficile. Le principe de cet algorithme de contrôle est très simple, il consiste à réaliser une «interpolation» entre un petit nombre de situations connues données par un expert sous la forme de règles floues du genre «si x est petit et y est modéré, alors u doit être très grand». La mise au point des prédicats évoqués par ces règles se fait généralement de façon empirique, mais de plus en plus, différentes méthodes d'apprentissage ont été appliquées dans le but d'avoir des systèmes de contrôle auto-adaptatif.

4.1. Algorithme du contrôleur flou

Le contrôle flou est l'étude, la mise au point et l'expérimentation de systèmes basés sur des règles exprimées par des catégories linguistiques mal définies telles que :

si (X est A₁) et (Y est A₂) alors (U est B)

où A₁, A₂, B sont des prédicats mentionnés comme «grand positif», «moyen», «presque nul».

Ces systèmes ont en commun d'avoir un petit nombre de règles (généralement entre 5 et 20), et de les considérer en disjonction : on ne veut pas voir de contradictions entre ces règles, mais au contraire, les prédicats se chevauchant, toutes les règles concourent à la détermination du paramètre U à contrôler.

La plupart du temps ces règles ont une, deux voire trois prémisses portant souvent sur un paramètre d'entrée E (erreur ou écart entre une valeur mesurée et une valeur cible ou «consigne» C) et sur la variation ΔE de ce paramètre entre deux mesures consécutives (E et sa dérivée E' si on préfère).

CONTRÔLE FLOU ET SYSTÈMES-EXPERTS

Nés vers les années 70, ces systèmes se sont développés séparément, et se distinguent donc des systèmes-expert sur plusieurs points : le petit nombre de règles, un traitement simple des contradictions, mais encore par le fait qu'ils ne servent pas à déterminer un objet à partir de données initiales au cours d'une session, mais à modifier en permanence un paramètre en bouclant continuellement, chaque règle apportant sa contribution, le paramètre de sortie est obtenu par superposition des informations contenues dans les règles. Il y a là un passage du quantitatif vers le qualitatif puis retour au quantitatif. Par ailleurs il n'y a pas d'enchaînement logique de déductions, sauf à faire fonctionner deux contrôleurs successivement dans le but de déterminer plusieurs paramètres.

LES RÈGLES

En fait il faut regarder un système de règles de contrôle flou comme une fonction (en général partiellement définie, dont on connaît des points imprécis du graphe) de

$$\Phi : \{\text{petit, moyen, grand}\}^* \{\text{petit, moyen, grand}\} \rightarrow \{\text{petit, moyen, grand}\}$$

dans le cas de neuf règles du type [si x est A₁ et y est A₂ alors u est B].

On définit ainsi partiellement une relation R sur des ensembles flous de dénomination maintenant traditionnelles NB (negative big), ... à PB (positive big) définies sur l'univers U*V de (x y).

L'ensemble flou inféré pour u lors de la règle [si (x, y) est A₁ et A₂ alors u est B] est défini par le modus ponens généralisé :

$$\mu_B(u) = \max_{(x, y) \in U*V} \{\mu_I(\Gamma[\mu_{A_1}(x), \mu_{A_2}(y)], \mu_B(u))\}$$

La plupart du temps, on aura des règles très simples portant sur E qui est la différence entre la sortie réelle et la consigne, et ∂E = variation de E, avec par exemple «si E est trop grand positif il faut le réduire, d'autant plus que «∂E» est grand algébriquement. Les transversales représentant à peu près des situations équivalentes. Par exemple, si E est PS (trop élevé) et ∂E est ZE (il ne varie pas) alors la conclusion NS indique qu'il faut le réduire légèrement.

PB	ZE	NS	NM	NB	NB
PS	PS	ZE	NS	NM	NB
ZE	PM	PS	ZE	NS	NM
NS	PB	PM	PS	ZE	NS
NB	PB	PB	PM	PS	ZE
$\partial E / E$	NB	NS	ZE	PS	PB

Comme on le verra sur certains exemples, les jeux de règles peuvent s'écarter notablement de cette configuration, notamment, ne conserver qu'une symétrie centrale de la table plutôt qu'une symétrie par rapport à la diagonale et surtout, ne pas être entièrement rempli.

D'autre part, dans la plupart des systèmes concrets existants, seules quelques cases du tableau sont remplies, le contrôleur réalise alors une sorte d'interpolation pondérée entre les règles, et renvoie 0 si aucune règle ne s'applique.

La conjonction (des prémisses) est usuellement interprétée par l'opération min (Zadeh), et la disjonction (des règles) comme le max, ainsi pour deux valeurs précises x et y et trois règles affectées de poids, on aura une troncature de chaque conclusion puis un sous-ensemble flou non obligatoirement normalisé pour la sortie U , et enfin une normalisation («défuzzification» ou «déflousification» par centre de gravité).

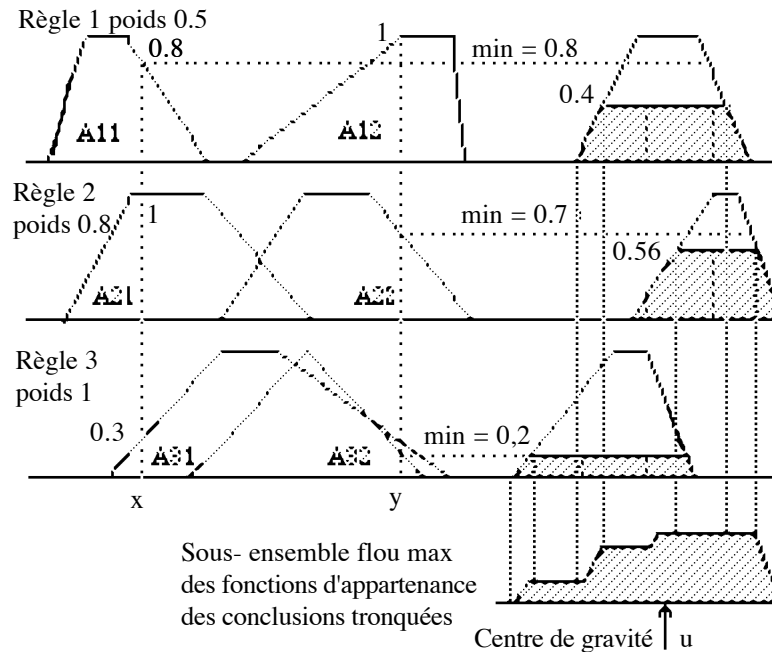


Figure 4.1 Schéma général de u fonction des entrées x et y

REMARQUE : s'il est général de choisir «min» pour la conjonction des prémisses, il paraît plus naturel de pondérer la conclusion avec le poids de la règle par l'opération «produit», en effet, prenons le premier schéma ci-dessus, en prenant «min», on aurait une troncature identique à 0.5 pour des «niveaux de vérité» des prémisses aussi différents que 1, 0.8, 0.6, 0.5. Si les règles s'appliquent peu ou si leurs poids sont faibles (zone peu élevée) il y aura quand même une valeur u (centre de gravité) déduite.

LES PRÉDICATS

Les problèmes qui se posent lors de la mise au point, sont ceux évoqués ci-dessus du choix des formules logiques, et de l'opérateur d'agrégation utilisé pour les conclusions, mais aussi des définitions apportées aux prédicats (problèmes de seuils, forme des fonctions d'appartenance etc). Les applications concrètes existant à l'heure actuelle utilisent des fonctions triangulaires ou trapézoïdales, du type courbe de Gauss, Arctangente, ou encore homographiques [Buckley 90], elles sont la plupart du temps établies du point de vue expérimental à partir de description subjectives d'opérateurs humains [Willaeys 79]. Afin de se ramener à un début d'universalité, on peut prendre $2N+1$ nombres flous triangulaires répartis sur $[-1,1]$, où les «pieds» de chaque triangle correspondent aux «sommets» de ses deux voisins, alors pour des entrées e_1, e_2, \dots, e_k , on a des théorèmes de convergence quand $N \rightarrow \infty$ [Buckley, Ying 89].

DÉFUZZIFICATION, MÉTHODES DE MAMDANI ET SUGENO

Si un paramètre U est acquis de plusieurs façons floues par plusieurs règles, chaque règle ayant une conclusion B_i , aboutit par le modus ponens généralisé à un B'_i . Se pose alors le problème de lui donner une représentation floue approximative, ou une valeur précise (defuzzification). Plusieurs méthodes peuvent être employées. La méthode des aires consiste à prendre la médiane qui fait le partage de l'aire en deux, $u = (\sum y_i \text{aire}(B'_i)) / \sum \text{aire}(B'_i)$. Une autre méthode préconise un mode (valeur donnant le max de u), mais le plus souvent c'est la méthode de la moyenne ou du centre de gravité, dite de Mamdani [Mamdani 75] consistant à prendre pour valeur finale en sortie, l'abscisse du centre de gravité de l'ensemble flou agrégeant les conclusions.

On pourra noter que si le prédicat PB est représenté par la diagonale dans $[0, 1]$ et que le contrôleur est réduit à une seule règle (x PB alors u PB), pour $x = 1$, u ne vaut pas 1 mais la moyenne de la fonction représentant PB, c'est à dire $2/3$.

Cet inconvénient est levé par la méthode de [Sugeno 85] qui consiste à écrire des règles de la forme : si (X est A₁) et (Y est A₂) alors U = u, (conclusion précise) le niveau de satisfaction de la règle étant calculé comme précédemment, alors pour plusieurs règles cette méthode prend simplement la moyenne pondérée des conclusions. Outre la plus grande rapidité d'exécution de cette méthode, sa différence essentielle avec celle de Mamdani réside dans le fait qu'une règle appliquée 2 fois compte double dans le résultat final.

En résumé, pour un jeu de règles données grâce à des prédicats flous, pour chaque valeurs précises des entrées x et y, chaque règle est déclenchée suivant son niveau de satisfaction, sa conclusion est tronquée par ce niveau, et la sortie exacte du contrôleur est dans les deux cas, le centre de gravité de l'agrégation de ces conclusions «pondérées».

EXEMPLE Une variable x initialement égale à 1 doit rejoindre la valeur cible 0. A chaque instant sont mesurées cette variable x et la variation dx entre sa valeur et la valeur précédemment mesurée. A ces deux entrées, la table de règles donnée ici, va permettre de calculer une valeur de sortie u qui devra être ajoutée à x.

PB			NS (2)		
PS		PS			
ZE	PB (1)				NB
NS				NS	
NB			PS		
dx / x	NB	NS	ZE	PS	PB

Ainsi par exemple si x est NB et dx est ZE (x est stabilisé trop en dessous de 0) la règle correspondant au numéro 1 dans la table, indique qu'il faut lui ajouter une grande valeur positive, mais si x est ZE (à sa place) et dx est PB (mais en augmentation, règle 2), il faut lui ajouter une valeur négative.

Dans cet exemple, x mesure la différence entre une variable et une cible, elle est initialement à -1, et la cible change de 1 à -1 toutes les 20 prises de données, il faut donc contrôler une variable pour lui faire suivre des créneaux.

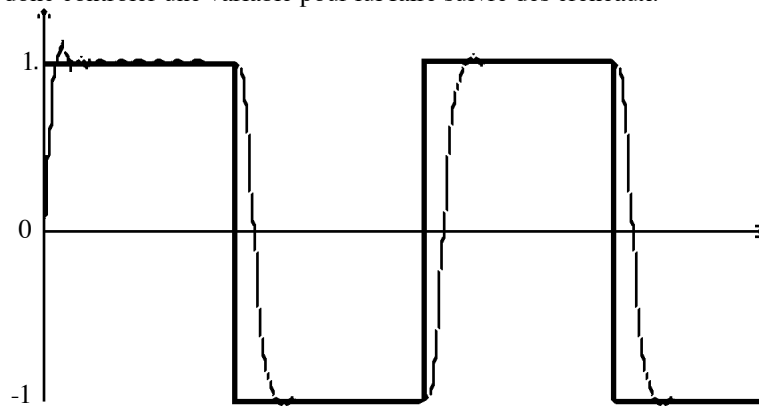


Figure 4.2 Consigne en créneaux.

Exercice 4.1

Mamdani est un cas particulier de modus ponens généralisé. On prend $x_0 = 36,7^\circ$ et les prédicats : A = anormale défini par 0 si $x = 37$, 1 si $|x - 37| > 1$ et affine par morceaux continue, R = rapide, défini par un trapèze de noyau [0, 25] et de support [0, 30], calculer le B' construit par modus ponens généralisé suivant π de Lukasiewicz, de Gödel ou de Mamdani avec une t-norme T quelconque.

Plus généralement dans une règle «x est A \rightarrow y est B» si $\mu_A(x_0) = p$ et si l'on utilise π de Mamdani, montrer que le modus ponens généralisé calcule un B' tronqué à p conformément à l'algorithme du contrôleur flou.

AVANTAGES DU CONTRÔLE FLOU

Les variables linguistiques sont bien adaptées en traduisant le raisonnement qualitatif humain, comme lui, on décrit des situations locales avec des règles représentant chacune une information partielle. Un opérateur expérimenté prend en effet ses décisions seulement sur des situations spécifiques dont il n'a qu'une connaissance imparfaite, mais qu'il agrège le long de son expérience. Le fait d'utiliser des prédicats flous, loin de traduire une pauvreté de l'information, permet d'exprimer des situations graduelles.

Le contrôle flou s'est surtout montré robuste, son comportement vis à vis des situations pathologiques, sa flexibilité (adaptation facile à des domaines, et ils sont nombreux, dont on ne possède pas de modèle mathématique) sont bons, et bien sûr il est rapide en temps réel.

La mise au point consiste aussi souvent en suppression de règles, il est curieux de constater qu'une amélioration passe par une simplification. On peut s'interroger néanmoins sur le paradigme du contrôle flou [Tong 84] quand est-ce approprié ? l'inconvénient majeur qui est l'empirisme dans le choix des règles, des prédicats et des univers de chacun des paramètres ne doit-il pas conduire à des systèmes plus compliqués d'apprentissage ? C'est effectivement le cas depuis plusieurs années, et le peu de méthodologie pour établir les règles et les prédicats qui était le talon d'Achille du contrôle flou a tendance à s'atténuer.

RÉSULTATS THÉORIQUES

On vérifiera que pour une valeur précise x, la commande floue est un modus-ponens généralisé employé avec l'implication de Mamdani, il est donc possible de l'effectuer avec différents opérateurs.

Les systèmes à base de règles (x est A_i) \Rightarrow (y est B_i) en nombre fini, forment un ensemble de fonctions, dense pour la norme de la convergence uniforme, dans l'ensemble des fonctions continues sur un compact, pourvu que les A_i et B_i soient gaussiennes [Wang, Mendel 92]. En fait, le résultat est étendu à toutes les A_i et B_i continues recouvrant tout l'espace et toute t-norme continue [Kosko 92], alors toute fonction continue à support compact peut être approchée par un nombre fini de règles floues. Le résultat est étendu de \mathbb{R}^n dans \mathbb{R}^p [Nguyen, Kreinovitch 93].

Prenons le cas de la dimension deux où un contrôleur flou décrit par n règles «si x est A_i et y est B_i alors u est C_i », pour une t-norme T , une t-conorme S , une implication I et un opérateur de défuzzyfication Df , réalise une fonction :

$$(x, y) \in [-1, 1]^2 \rightarrow u(x, y) = Df \left(\sum_{i=1}^n I [T(\mu_{A_i}(x), \mu_{B_i}(y)), \mu_{C_i}] \right)$$

où généralement $Df(f) = \frac{\int_{\text{supp}(f)} xf(x)dx}{\int_{\text{supp}(f)} f(x)dx}$ est le centre de gravité de f .

Il arrive que le produit soit choisi pour la conjonction (logique probabiliste) ou que la t-conorme soit celle de Lukasiewicz $S(p, q) = \min(1, p + q)$ ou encore, que des contrôleurs mixent ces lois. On trouvera une comparaison des opérateurs choisis dans le cas d'un contrôleur non flou «proportionnel-intégral» dans [Siler Ying 89]. Cependant, il faut bien remarquer que les différentes implications mises en oeuvre dans un contrôleur flou, ne peuvent donner satisfaction en général.

En effet, rappelons les principales implications :

de Lukasiewicz $I_L(a, b) = \min(1 - a + b, 1)$,

de Goguen ou probabiliste $I_p(a, b) = \text{si } a < b \text{ alors } 1 \text{ sinon } b / a$,

de Gödel $I_G(a, b) = \text{si } a < b \text{ alors } 1 \text{ sinon } b$,

et de Kleene $I_K(a, b) = \text{si } b < 1 - a \text{ alors } 1 - a \text{ sinon } b$.

Pour toutes ces implications, il est clair qu'une règle non satisfaite ($a = 0$) donnera toujours la conclusion au niveau 1. C'est le vieux paradoxe entre l'implication booléenne qui est telle que le faux implique le vrai, et le raisonnement de sens commun qui s'y refuse. Supposons donc que pour une entrée précise, il faille appliquer le modus-ponens généralisé, toutes les règles donneront leurs conclusion, et, comme c'est souvent le cas, si le jeu de règles est symétrique, l'agrégation de celles-ci donnera zéro.

EXEMPLE : On trace les 15 zones de niveaux de u entre -0.7 et 0.7 à 0.01 près pour x, y dans $[-1, 1]^2$ et la définition suivante (jeu de trois prédicats NB, ZE, PB et de cinq règles :

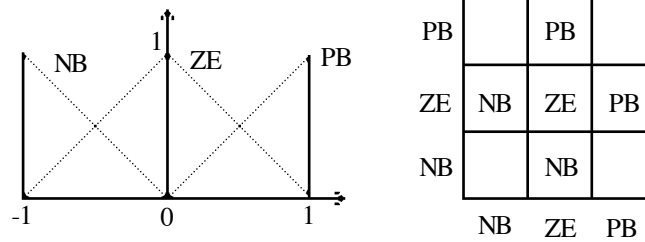


Figure 4.3 Prédicats et règles utilisées pour une étude comparative des opérateurs.

La première comparaison ci-dessous est effectuée avec la méthode de Mamdani, où d'après la remarque faite plus haut, u est limité entre -0.7 et 0.7 . Dans le cas de la méthode de Sugeno, ce sont les valeurs des modes $-1, 0, 1$ qui sont prises comme conclusions, et u parcourt véritablement l'intervalle $[-1, 1]$.

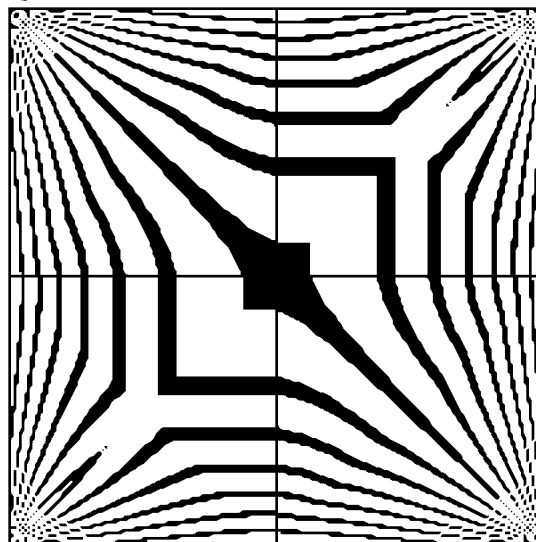


Figure 4.4 Opérateurs Mamdani $T = I = \min$, $S = \max$. Toujours dans $[-1, 1]$, la diagonale correspond à la valeur $u = 0$. Les autres lignes allant de $-0,1$ à $-0,7$ en dessous et de $0,1$ à $0,7$ au dessus.

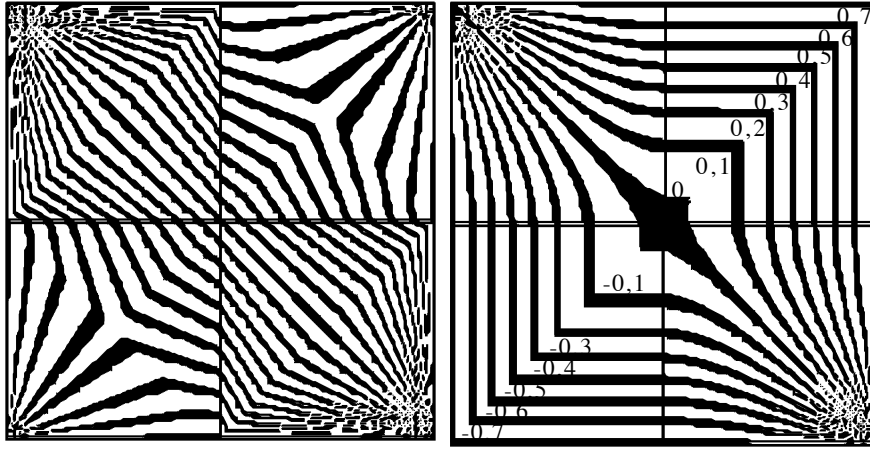


Figure 4.5 Pour x, y dans $[-1, 1]$, méthode de Sugeno à gauche (21 lignes de niveaux, seul le mode de la conclusion est pris en compte). La diagonale est la ligne de niveau 0, les autres sont celles de -1 à 1 avec un pas de 0.1 . A droite : opérateurs Larsen $T = I =$ produit, $S = \max$, (mêmes prédicats et règles).

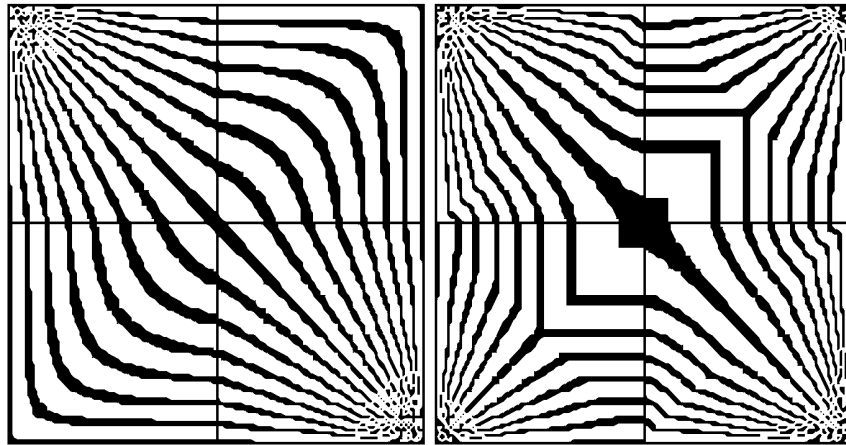


Figure 4.6 Larsen $T = I =$ produit, $S =$ proba à gauche, et opérateurs Larsen $T = \min$, $S = \max$, $I =$ produit pour le schéma de droite.

Dans le cas des figures ci-dessous, la méthode de Mamdani a été employée avec les mêmes règles mais en modifiant les prédicats NB et PB ci-contre de façon à ce qu'il y ait un chevauchement plus important des règles. La plupart du temps pour x et y dans $] -1, 1[$, les cinq règles s'appliquent à des degrés divers, aussi des règles de conclusions antagonistes s'appliquent davantage, la conclusion s'en trouve plus modérée (entre -0.4 et 0.4).

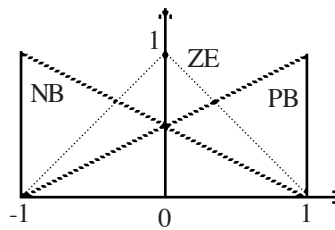


Figure 4.7 Modification des prédicats pour un plus grand chevauchement.

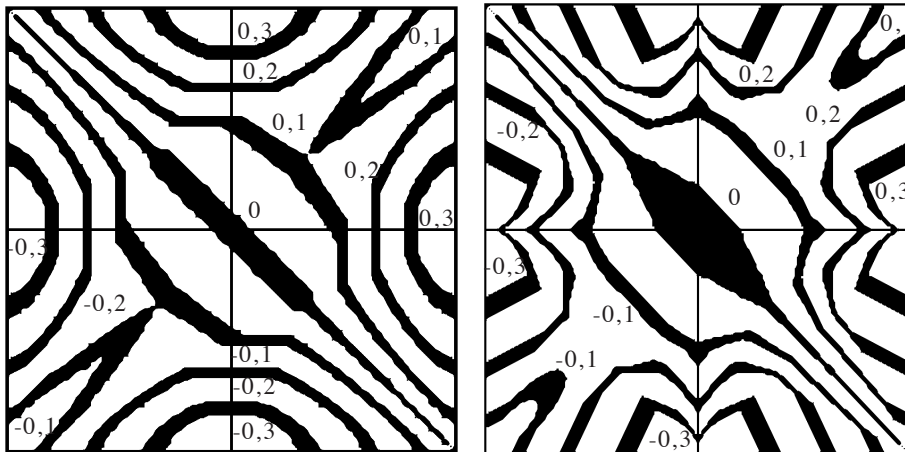


Figure 4.8 Avec ce jeu de prédicats, Mamdani : $T = I = \min$, $S = \max$, puis T, S de Lukasiewicz et $I = \min$ (Mamdani).

4.2. Applications

Beaucoup d'applications sont nées au Japon sous l'impulsion du Laboratory for International Research on Fuzzy Engineering (LIFE) à Yokohama et aussi du Fuzzy Logic Institute. Les premières applications industrielles ont également été développées au Japon.

MACHINE À VAPEUR EN MODÈLE RÉDUIT

C'est une des premières expérimentations de commande floue [King, Mamdani 77], [Assilian, Mamdani, Gaines 81]. Un premier contrôleur est chargé d'acquérir une pression donnée, un second mettant en cause SE (speed error) et TC (throttle change = modification de l'accélération, qui est en fait un débit contrôlé par une vanne) a pour but d'acquérir une vitesse donnée. Le contrôle de HC (heat change) suivant PE (pressure error = différence entre la pression actuelle et la pression but à atteindre) et CPE (change in error pressure = $PE_{t+1} - PE_t$) dans la boucle d'un moteur à vapeur, se fait par l'examen en disjonction des règles suivantes :

PE est (NB ou NM) et CPE est NS \rightarrow HC est PM
PE est NS et CPE est PS \rightarrow HC est PM
PE est NO et CPE est (PB ou PM) \rightarrow HC est PM
PE est NO et CPE (NB ou NM) \rightarrow HC est NM
PE est (PO ou NO) et CPE est NO \rightarrow HC est NO
PE est PO et CPE (NB ou NM) \rightarrow HC est PM
PE est PO et CPE est (PB ou PM) \rightarrow HC est NM
PE est PS et CPE est (PS ou NO) \rightarrow HC est NM
PE est (PB ou PM) et CPE est NS \rightarrow HC est NM

La plupart du temps, plusieurs règles contribuent à la constitution de la valeur floue en sortie. Les règles ayant le plus contribué lors d'une session sont signalées de façon à pouvoir les modifier par ajustements à la réalité. Les «et», «ou» correspondent aux opérations min, max.

Un autre prédicat ANY (le «NG» de Yamakawa) toujours à 1, est utilisé. Les prédicats sont définis sur 15 points de [-1, 1], ainsi par exemple si PB est (000 000 000 00 0.1 0.4 0.8 1) et PM est (000 000 000 0.2 0.7 1 0.7 0.2 0) on aura (not PB or PM) représenté par la séquence (111 111 111 0.8 0.3 0 0.3 0.2 0).

Ces noms de prédicats sont d'ailleurs différemment valués pour les différentes grandeurs PE, HC, TC.

MAINTIEN DE LA VITESSE DE ROTATION D'UN MOTEUR [Kiszka, Kochanska, Sliwiska 85]

Le rôle du système de contrôle est ici, de maintenir une vitesse constante pour un moteur électrique, en dépit des effets variateurs. L'idée générale que soulignent les auteurs est que l'établissement du modèle mathématique s'avère difficile alors que l'expert exprime verbalement les règles simplement, d'autre part il y a une très grande possibilité de modifier les performances du système en altérant les différentes descriptions verbales telles que «petit» etc. Ces prédicats se chevauchent fortement.

Ce système a une vingtaine de règles du type : [si S petit et I moyen alors U zéro], où les paramètres observés sont E = $\Delta\omega$ (erreur observée dans la vitesse angulaire), I intensité, S somme des erreurs, U variable de contrôle servant ensuite à déterminer la vitesse de rotation Ω par une quinzaine de règles du type : U grand et I petit alors Ω très grand

Trois formules ont été expérimentées pour définir la relation floue R entre fonctions d'appartenances définies par le premier type de règles suivant que les règles sont examinés en l'implication de lukasiewicz ($P \rightarrow Q$) = $\min(1, 1 + p - q)$ les résultats avec les deux dernières formules donnent des résultats semblables, alors que la première diverge.

AJUSTEMENT D'UNE VANNE DANS UNE USINE DE FONDERIE [Bartolini, Casalino, Davoli, Minciardi, Morten 85]

L'objectif de ce système est simplement de régler un paramètre u servant au débit d'une vanne entre un réceptacle contenant du métal en fusion, et un deuxième bassin dont le niveau est mesuré par la hauteur h. Ce dernier se déversant dans un moule.

Les prédicats pour qualifier h sont définis dans l'intervalle de 75 cm à 85 cm.

On a «moyen» par une fonction gaussienne et «petit», «haut» caractérisés par des fonctions d'appartenance du type $y = 1/(1 + a(x - c)^b)$.

Les prédicats «très négatif», «négatif», «positif», et «très positif» caractérisant dh sont définis entre -0,9 et 1,2 cm/s de la même manière.

Les paramètres a, b, c, m, s sont modifiés expérimentalement pour enlever les effets non désirés. La commande u prend les valeurs -1, 0 et 1 avec des possibilités 1, 1/2 ou 0 suivant qu'il est «bas», «haut» ou «moyen haut» :

L'originalité de ce système (ce qui n'est pas la règle) est qu'il prend en compte systématiquement en compte les douze possibilités, définissant complètement une relation floue.

h petit et dh très négatif \rightarrow u est haut	h petit et dh négatif \rightarrow u est haut
h petit et dh positif \rightarrow u est haut	h petit et dh très positif \rightarrow u moyen-haut
h moyen et dh très négatif \rightarrow u est haut	h moyen et dh négatif \rightarrow u est haut
h moyen et dh positif \rightarrow u est bas	h moyen et dh très positif \rightarrow u est bas
h haut et dh très négatif \rightarrow u est bas	h haut et dh négatif \rightarrow u est bas
h haut et dh positif \rightarrow u est bas	h haut et dh très positif \rightarrow u est bas

L'objectif de cette application est de doser l'adjonction de constituants chimiques comme le chlore, un agent alcalin, et une addition coagulante le PAC (chlorure d'aluminium polymérisé) pour purifier de l'eau de rivière. On reproduit le jugement d'opérateurs humains pour déterminer la quantité de PAC afin que la turbidité (en mg/l) de l'eau soit maintenue sous un certain seuil.

Les prédicats utilisés sont nombreux et définis par des courbes de Gauss dont on donne ici les paramètres moyenne, et écart-type :

SS small small	0	0,15	NB	-1	0,4
SA small	0	0,3	NM	-0,5	0,2
SM small medium	0,3	0,15	NS	-0,2	0,2
MM medium	0,3	0,15	ZR	0	0,2
ML medium large	0,7	0,15	PS	0,2	0,2
LA large	1	0,3	PM	0,5	0,2
LL very large	1	0,15	PB	1	0,4

Les données d'entrée sont, la température θ (dans l'intervalle $[0, 30]$), les turbidités initiale $TU_1 \in [0,50]$, de l'eau traitée $TU_2 \in [0,3]$, son accroissement $TUP \in [0, 1]$, l'alcalinité $ALK \in [8, 18]$, la flocculisation FLOC et le démarrage STAR. On les ramène à $[0, 1]$. La donnée retournée est le montant M de compensation du taux de PAC $\in [-10, 10]$ et les dix règles sont (Composition des informations suivant min-max) :

$TU_1 = MM, TU_2 \neq LA, \theta \neq SA \rightarrow M = NM$	$TU_1 = SS \rightarrow M = PM$
$TU_1 = SA, ALK = SA, \theta = SA \rightarrow M = NM$	$TU_1 = LA, ALK = SA \rightarrow M = NM$
$TU_2 = LA \rightarrow M = NM$	$TUP = LL \rightarrow M = PB$
$TUP = ML \rightarrow M = PM$	$TUP = MM \rightarrow M = PS$
$FLOC = SA \rightarrow M = PM$	$STAR = LA \rightarrow M = PS$
qui peuvent se simplifier en :	$TU_1 = LA, ALK = SA \rightarrow M = NM$
$STAR = LA \rightarrow M = PS$	$\Delta TUP = LL \rightarrow M = PB$
$TU_1 = SA, ALK = SA, \theta = SA \rightarrow M = NM$	$FLOC = SA \rightarrow M = PM$

CONTRÔLE DE LA VITESSE D'UNE VOITURE [MURAKAMI, MAEDA 85]

On considère un fait F tel que « X est A » où A est un ensemble flou d'un univers U , possède une valeur τ considérée elle aussi comme un ensemble flou de $[0,1]$. Le problème est alors de définir un ensemble flou B de U tel que « $(X \text{ est } A) \text{ est } \tau$ » soit équivalent à « X est B » par $\mu_B = \mu_\tau \circ \mu_A$. Inversement quand A et B sont spécifiés on trouve une valeur de vérité linguistique τ par $\mu_\tau = \mu_B \circ \mu_A^{-1}$.

L'implication étant définie sous la forme de Lukasiewicz $(p \rightarrow q) = \min(1, 1 - p + q)$ on définit la fonction d'appartenance par modulus généralisé :

$$\mu_{p \rightarrow q}(r) = \sup\{\min(\mu_p(p), \mu_Q(q)) / \min(1, 1 - p + q) = r\}$$

Cette formule étant utilisée pour inférer la valeur de q quand p et $r = (p \rightarrow q)$ sont connus.

L'objectif du système est, étant donné une vitesse v_0 fixée (par exemple 40km/h puis 60km/h), de calculer une variable de contrôle u servant à modifier la vitesse v .

A chaque instant la vitesse réelle mesurée étant v , on pose l'écart $e = v_0 - v$ et la différence entre deux observations (mesurant l'accélération) $\Delta e_t = e_t - e_{t-1}$. On pose de même $\Delta u_t = u_t - u_{t-1}$, ce système ne possède alors que 4 règles :

- si e est PE alors Δu est PU
- si e est NE alors Δu est NU
- si Δe est PDE alors Δu est PDU
- si Δe est NDE alors Δu est NDU

Les prédicats P (ositif), N (égatif) pour les grandeurs e , Δe , Δu , sont définis grâce à des fonctions d'appartenance du type $\mu(x) = 0,5 + (\tan(\alpha t))/\pi$ où $\alpha = (\tan(0,45))/r$ de telle sorte que pour la constante r on ait une possibilité 0,95. En rajoutant deux règles faisant intervenir $\Delta^2 e_t = \Delta e_t - \Delta e_{t-1}$, les auteurs améliorent le fonctionnement du contrôle.

CONTRÔLE D'UNE RAME DE MÉTRO [Yasunobu, Miyamoto 85]

Le métro de Sendai a marqué le début de la «fuzzymania» au Japon. (mis en service le 15-7-87 par Hitachi)

La commande est l'accélération/décélération suivant 19 règles traduisant les limitations sur la vitesse, la sécurité, le confort, de consommation d'énergie et la «traceability».

T est le profil de vitesse imposé «traceability» C le confort quantifié par le nombre d'action apporté au levier «notch» DN le nombre de crans d'accélération - décélération, $DN = PN - BN$

Les règles sont de la forme si (u est $C \rightarrow x$ est A et y est B) alors u est C :

- Si (N est $(N(t)+B_{max})/2 \rightarrow S$ est SD) alors N est $(N(t)+B_{max})/2$
- Si (N est $O \rightarrow S$ est SS, C est CG, E est ES) alors N est O
- (Si levier non changé \rightarrow le train stoppe dans la zone prévue alors levier non changé)
- Si (N est $P7 \rightarrow S$ est SS, C est CG, T est TL) alors N est $P7$
- Si (DN est $O \rightarrow S$ est SS, T est TG) alors DN est O
- Si (DN est $n \rightarrow S$ est SS, C est CG, T est TA) alors DN est n pour $n = \pm 1, \pm 2, \pm 3$
- Si (DN est $O \rightarrow R$ est RP, G est GG) alors DN est O
- Si (N est $O \rightarrow R$ est RF, C est CG) alors N est O
- Si (N est $B1 \rightarrow R$ est RP, C est CG) alors N est $B1$
- Si (DN est $n \rightarrow R$ est RP, C est CG, G est GA) alors DN est n , pour $n = \pm 1, \pm 2, \pm 3$

Les prédicats sont définis avec des fonctions homographiques :

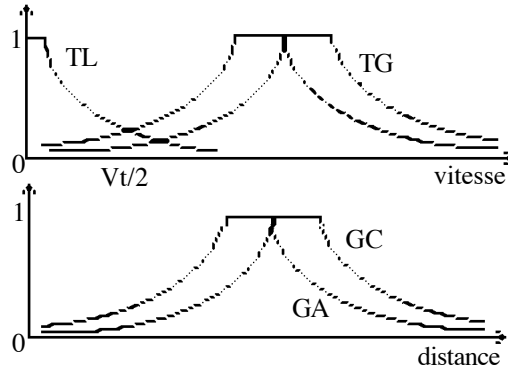


Figure 4.9 Ensembles flous TG : good trace, TL : low speed, TA : accurate trace, GG : good stop, GA accurate stop, CG : good confort, ES : energy saved running, et RP : ponctuel, RF : retard.

Ce sont les formules de Mamdani qui sont appliquées, mais, dans ce système, on ne déclenche que la règle qui a la meilleure pondération.

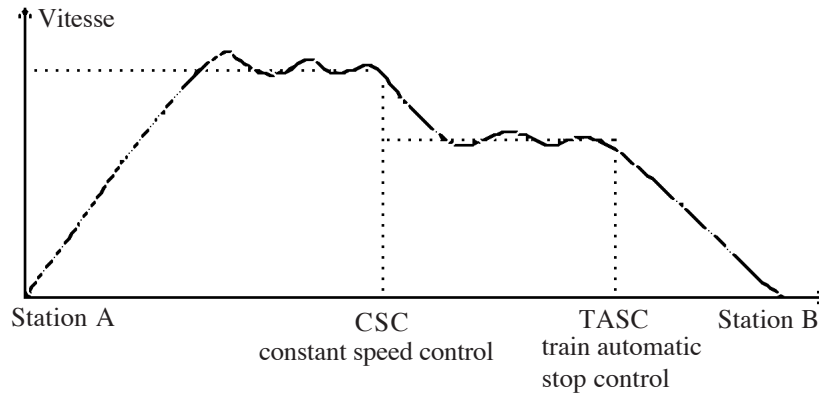


Figure 4.10 Vitesse d'un train entre deux stations.

CSC et TASC sont deux marqueurs sur la voie entre les stations. Les résultats montrent environ deux fois moins de modifications de N et une économie d'énergie de 10%, un confort supérieur au contrôle PID ainsi qu'une meilleure précision du point d'arrêt (écart-type de l'ordre de 10cm). Dans le même ordre d'idées une étude a été faite sur le contrôle du balancement des charges pour les grues.

CONTRÔLE FLOU D'UNE VOITURE EN MODÈLE RÉDUIT [SUGENO, MURAKAMI 89]

Une voiture 56*28 cm équipée de capteurs à ultra-sons mesure à chaque instant les distances frontales et latérales aux murs par rapport auxquels elle se déplace. La route est large de 80cm et doit se garer dans un garage. Elle reçoit des ordres vocaux tels que «tout droit», «tourner à gauche» etc.

Si θ est l'angle du véhicule avec la route, et ϕ l'angle que fait la direction des roues avant avec l'axe du véhicule, les règles pour se maintenir tout droit sont :

d gauche et θ moyen $\rightarrow \phi$ droite d droite et θ moyen $\rightarrow \phi$ gauche
 θ gauche $\rightarrow \phi$ droite θ droite $\rightarrow \phi$ gauche

Les connecteurs utilisés sont mixtes :

(p et q) = $\min(p, q)$ (Zadeh) et : (p ou q) = $\min(1, p + q)$ (Lukasiewicz)

Les prédicats sont définis triangulairement sur [-20cm, 20cm] pour d, [-90°, 90°] pour θ , et [-30°, 30°] pour ϕ (orientation dans le sens d'une montre).

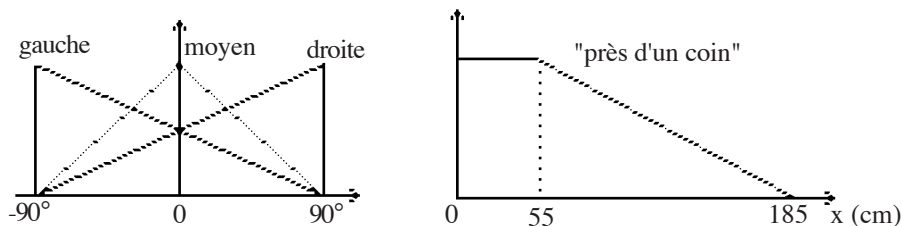


Figure 4.11 Prédicats utilisés pour le contrôle d'une voiture.

Sont également utilisés des valeurs de vérité correspondant à des appréciations linguistiques comme :

pas complètement faux	0,1	plutôt vrai	0,3
vrai	0,5	très vrai	0,7
très très vrai	0,9		

La «défuzzification» se fait par centre de gravité.

Trois paramètres interviennent dans la conduite de ce robot : la distance d au mur droit, la distance g au mur gauche, l'angle θ de l'axe du véhicule avec la ligne médiane. On les compare entre la position courante, la position à L mètres devant le véhicule, et la position au prochain noeud. On pose $L = 0,25v + 5$ où v est la vitesse du véhicule en m/s.

Soient $\delta_1 = [\text{si } \theta > 0 \text{ alors } d - d' \text{ sinon } |g| - |g'|]$, $\delta_2 = [\text{si } \theta > 0 \text{ alors } d_0 - d' \text{ sinon } |g'| - |g_0|]$, $\delta_3 = [\text{si } \theta > 0 \text{ alors } \theta - \theta_d \text{ sinon } \theta - \theta_g]$, $d_4 = [\text{si } q \leq 0 \text{ alors } \theta_0 - \theta_d \text{ sinon } \theta_0 - \theta_g]$, et $drf = \min(1, dr/5)$ où on est en «path command» si $dr > 5$ et en «node command» sinon.

A partir de ces paramètres, on définit $\Delta x = |1 - drf_1 \cdot \delta_1 + drf_2 \cdot \delta_2 + \Delta \theta| = |1 - drf_1 \cdot \delta_3 + drf_2 \cdot \delta_4|$, et c'est seulement sur ces 2 paramètres que vont porter les règles floues.

Cinq prédicats sont définis triangulairement, déterminant 25 règles, mais l'originalité de cette expérience réside dans le fait que les règles utilisent des prémisses floues mais des conclusions précises (Sugeno) en donnant l'angle de braquage en degrés (table ci dessous).

PB	0	-2	-4	-6	-10
PS	1	0	-1	-3	-6
ZE	3	1	0	-1	-3
NS	6	3	1	0	-1
NB	10	6	4	2	0
$\Delta \theta / \Delta x$	NB	NS	ZE	PS	PB

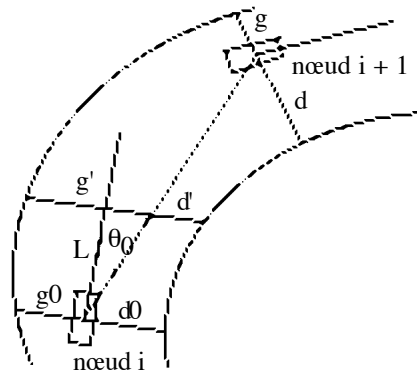


Figure 4.12 Suivi d'un couloir.

LE PENDULE INVERSÉ [Yamakawa 89]

C'est l'application la plus fameuse à titre de démonstration. Une tige métallique est montée à sa base sur un pivot appartenant à un charriot mobile, et à chaque instant le pendule fait avec la verticale un angle θ .

En contrôlant la vitesse algébrique v du charriot, on doit pouvoir maintenir le pendule en équilibre. Le modèle mathématique se traduit par un système différentiel non linéaire, donc très difficile à résoudre, alors que des règles floues permettent de décider de v à chaque instant en tenant compte de θ et de sa dérivée, la vitesse angulaire θ' , seulement 7 règles permettent d'arriver au résultat :

- (θ est PM) et (θ' est ZR) \rightarrow (v est PM)
- (θ est PS) et (θ' est PS) \rightarrow (v est PS)
- (θ est PS) et (θ' est NS) \rightarrow (v est ZR)
- (θ est NM) et (θ' est ZR) \rightarrow (v est NM)
- (θ est NS) et (θ' est NS) \rightarrow (v est NS)
- (θ est NS) et (θ' est PS) \rightarrow (v est ZR)
- (θ est ZR) et (θ' est ZR) \rightarrow (v est ZR)

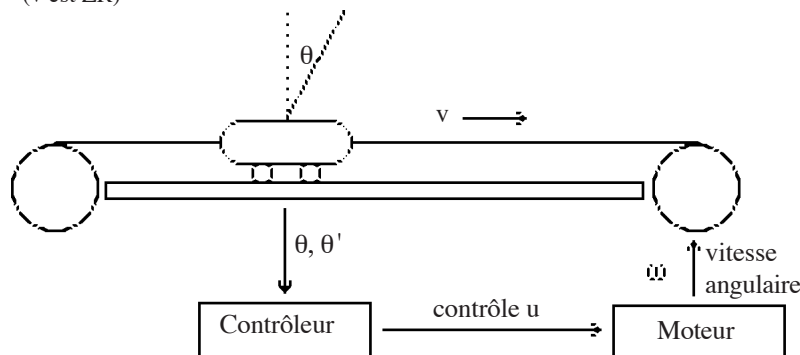


Figure 4.13 Schéma du pendule inversé.

Bien entendu, la stabilisation de la position du charriot requière un deuxième groupe de règles comportant toutes (θ est ZR) dans leur prémisses, et qui renforceront le contrôle de v .

Une foule d'expérimentation ont été faite sur ce thème, notamment la comparaison des opérateurs [Cordon, Herrera 97]. On peut signaler un système ne comportant que 3 prédicats N, Z, P, pour les entrées et 7 pour la sortie :

P	ZE	PS	PB
Z	NM	ZE	PM
N	NB	NS	ZE
$\partial\theta / \theta$	N	Z	P

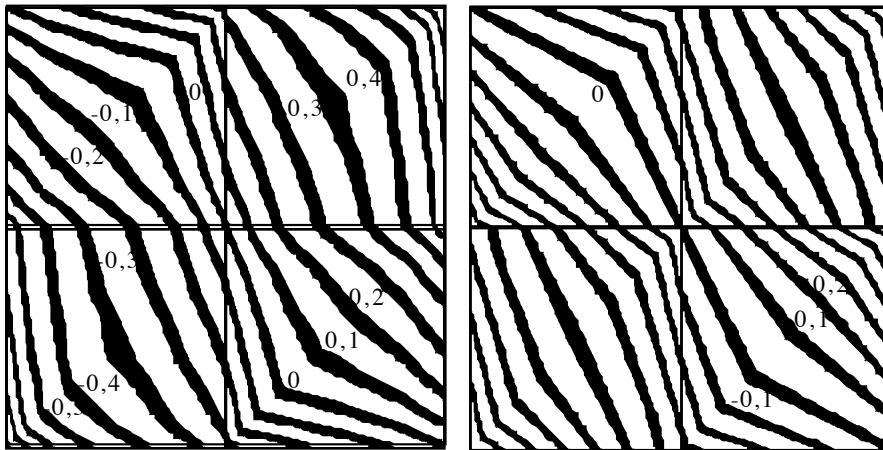


Figure 4.14 Lignes de niveaux pour θ et θ' dans $[-1, 1]^2$ de -0.8 à 0.8 (17 lignes) obtenues à 0.01 près pour cette table de règles suivant Mamdani. A droite, 21 lignes de niveaux pour la méthode de Sugeno où v est entre -1 et 1 (les conclusions étant en ce cas les modes des triangles). Hormis les règles :

- $(\theta \text{ est N}) \text{ et } (\theta' \text{ est N}) \rightarrow (v \text{ est NB})$ $(\theta \text{ est N}) \text{ et } (\theta' \text{ est Z}) \rightarrow (v \text{ est NM})$
 $(\theta \text{ est N}) \text{ et } (\theta' \text{ est P}) \rightarrow (v \text{ est ZE})$ $(\theta \text{ est Z}) \text{ et } (\theta' \text{ est N}) \rightarrow (v \text{ est NS})$
 $(\theta \text{ est Z}) \text{ et } (\theta' \text{ est Z}) \rightarrow (v \text{ est ZE})$ $(\theta \text{ est Z}) \text{ et } (\theta' \text{ est P}) \rightarrow (v \text{ est PS})$
 $(\theta \text{ est P}) \text{ et } (\theta' \text{ est N}) \rightarrow (v \text{ est ZE})$ $(\theta \text{ est P}) \text{ et } (\theta' \text{ est Z}) \rightarrow (v \text{ est PM})$
 $(\theta \text{ est P}) \text{ et } (\theta' \text{ est P}) \rightarrow (v \text{ est PB})$

On trouve encore un système de 11 règles avec 5 prédicats pour les trois paramètres, que l'on peut résumer par :

- $(\theta \text{ est ZE}) \rightarrow (v \text{ est } -\theta')$ $(\theta' \text{ est ZE}) \rightarrow (v \text{ est } -\theta)$
 $(\theta \text{ est PS}) \text{ et } (\theta' \text{ est NS}) \rightarrow (v \text{ est NE})$ $(\theta \text{ est NS}) \text{ et } (\theta' \text{ est PS}) \rightarrow (v \text{ est ZE})$

Le double pendule inversé consiste à considérer deux tiges articulées, les paramètres en entrée sont donc toujours x et x' pour la position du charriot, q_1 et q_1' pour l'angle que fait la première tige avec la verticale, mais aussi q_2 et q_2' pour l'angle de la seconde tige avec la verticale. On trouvera une intéressante recherche des règles par algorithme génétique (chapitre 5) dans [Renhou, Yi 96].

RECTIFICATION DES TEINTES D'UNE PHOTOCOPIEUSE COULEUR [Geno, Fujiwara 90]

On cherche à réaliser un duplicata de meilleure qualité que l'original. Pour cela les teintes de la peau du visage sont décrites par des prédicats tels que «désirable», «saine», «jeune», «noble» représentés par des ensembles flous de R^3 s'exprimant par 3 nombres flous triangulaires en coordonnées cylindriques notées la saturation $Cab = \rho \in [10, 50]$, Hab (longueur d'onde) $= \theta \in [0, 90]$, et $L = z \in [20, 80]$. 3×27 règles et la méthode Sugeno sont utilisées du type «L (luminosité) est A, Cab est B, Hab est C \rightarrow modification de L», Cab, Hab, A, B, C étant dans tous les cas petit, moyen, grand. Ces prédicats sont définis grâce aux réponses humaines apportées sur une échelle de 5 graduations.

Exemples de règles: 1) Ton des lèvres L moyen, Cab grand, Hab petit \rightarrow Accentuer le rouge $\Delta l = 8, \Delta Cab = 10, \Delta Hab = 0$ 2) Ton jaunâtre et sombre L moyen, Cab grand, Hab grand \rightarrow aller vers le rouge et augmenter la luminosité $\Delta l = 8, \Delta Cab = -10, \Delta Hab = -14$

SUIVI D'UNE LIGNE DE MARQUEURS [KAMADA, YOSHIDA 90]

Un véhicule doit suivre une ligne de marqueurs dans un repère mobile en repérant deux marqueurs consécutifs tq $|x_1 - x_2| < c_1$ et $|y_1 - y_2| > c_2$ car en général l'angle A que fait la direction du véhicule avec la ligne marquée, est petit. A chaque instant A peut être mesuré par $\tan(A) = |x_1 - x_2| / |y_1 - y_2|$, puis la distance latérale c'est à dire perpendiculaire à la route du véhicule par $D = (x_1 \cos(A) + y_1 \sin(A)) \cos(A)$.

Deux expériences ont été effectuées pour déduire l'angle U de braquage, soit à partir de A et de D, soit à partir de D et de ∂D . Les résultats $-5 < D < 5$ en 0,67 pixel, $-2 < \partial D < 2$ en 0,67 pixel et $-5/20 < A < 5/20$ en radian, montrent que l'angle A est un paramètre plus important que D ou ∂D . 5 prédicats triangulaires réguliers sont pour cela utilisés sur l'intervalle $[-1, 1]$, le premier jeu de règles «(A, D) \rightarrow virage U», est :

- A est ZR \rightarrow U est égal à D (5 règles condensées par cette écriture)
A est NB et D est NS \rightarrow U est NB A est NS et D est PS \rightarrow U est NB
A est PS et D est PS \rightarrow U est PB A est PB et D est NS \rightarrow U est PB

9 autres règles réalisent (D, ∂D) \rightarrow virage, donnés par la table :

PB		PS			
PS		ZE		PS	
ZE	NB		ZE		PB
NS		NS		ZE	
NB				NS	
$\partial D / D$	NB	NS	ZE	PS	PB

Ci-dessous ces règles avec une famille de prédicats triangulaires de base 2 au lieu de 1, il y a donc un chevauchement étendu des règles comblant les vides de la table. Une entrée (D, dD) pouvant vérifier jusqu'à cinq règles. Cette table réalise une fonction sur $[-1, 1]^2$ à valeurs dans $[-0.8, 0.8]$ dont on figure 15 lignes de niveaux à gauche suivant la méthode de Sugeno en prenant le mode de chaque prédicat (par exemple une conclusion PB signifie 1). Pour la méthode de Larsen ou Mamdani, les valeurs sont dans $[-0.5, 0.5]$ dont on figure à droite 11 lignes de niveaux.

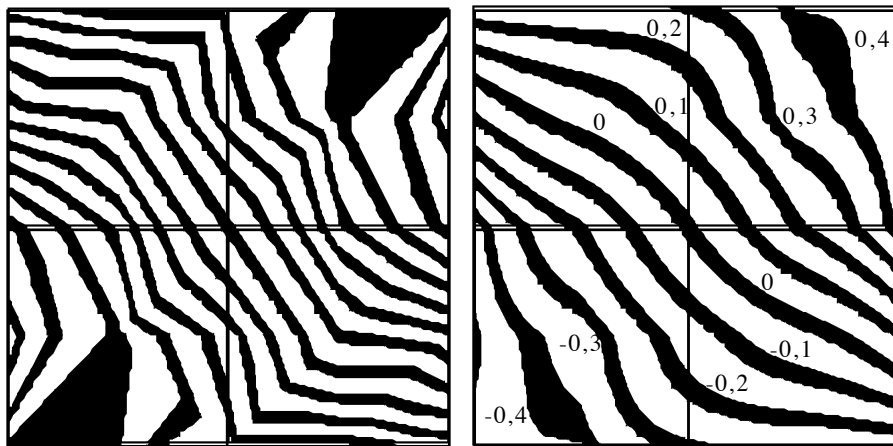


Figure 4.15 Pour x, y dans $[-1, 1]$, la méthode de Sugeno avec $T = \min$ à gauche et la méthode de Larsen accompagnée de $T = I = \text{produit}$, S probabiliste à droite. Il est très difficile dans ce genre d'expérience d'établir des comparaisons entre opérateurs, mais la méthode de Sugeno est la plus employée à cause de sa rapidité d'exécution.

De plus il y a plusieurs lignes de marques, on ne souhaite pas utiliser la distance entre le véhicule et la ligne des marques la plus proche car cela provoque de grands virages, alors que la distance latérale c'est à dire sur la perpendiculaire à la trajectoire, D est calculé par $D = (x_1 \cos(A) + y_1 \sin(A)) \cos(A)$, et de A et D ou bien de D et dD , on déduit l'angle de braquage et on réitère.

Il est impossible de mentionner toutes les applications même les plus marquantes et les plus originales qui ont été réalisées ou qui sont à l'étude car elles sont trop nombreuses, de plus les applications industrielles ont la plupart du temps des spécificités confidentielles. On citera cependant brièvement quelques domaines d'application très variés.

L'HÉLICOPTÈRE DE SUGENO

La stabilité et le contrôle d'un hélicoptère en modèle réduit se font grâce aux paramètres positions x, y, z , vitesses x', y', z' et les angles ϕ (inclinaison suivant l'axe longitudinal), θ (inclinaison suivant l'axe transversal), ψ (suivant la verticale) et leurs variations.

On a des règles du style de (x est NS) et (y' est ZE) et (ψ est ZE) alors (θ est NS)

4 sorties sont contrôlées : la longueur (avant-arrière), le latéral (droite-gauche), le «pedal» (angle d'attaque par rapport à l'horizontale) et le «collectif» (régime du moteur pour l'altitude). Chacun des 4 contrôles se fait avec 9 règles de 2 entrées confrontées à seulement 3 prédicats, les conclusions étant numériques (méthode Sugeno). Une maquette obéissant à des commandes vocales a été mise au point.

AUTO-FOCUS (réalisé par Canon)

On a trois points de mesure, centre C, en haut à gauche L, en bas à droite R, et on utilise cinq règles :

C proche \rightarrow Pc haut L proche \rightarrow Pl haut R proche \rightarrow Pr haut

L loin C moyen R proche \rightarrow Pc haut R loin C moyen L proche \rightarrow Pc haut

En réalisant une inférence sur une courte durée de 6ms, les résultats annoncés sont de 96,5% de réussite à comparer avec 73,6% par une mise au point manuelle. Le problème de la stabilisation d'une caméra a aussi été traité par 8 règles portant sur 20 points représentatifs de l'image.

STABILISATION D'UNE BALLE DE PING-PONG

Sur un plateau carré mouvant, la position x, y de la balle est captée par une caméra. Un contrôleur de 12 règles corrige l'orientation transversale du plateau, et le même contrôleur corrige l'orientation longitudinale (P.Horacek Université technique de Prague). Dans le même ordre d'idée, une valise en équilibre est stabilisée par deux contre-poids a, da en milliradians (11 règles) \rightarrow position des poids.

CONTRÔLE D'UNE DOUCHE (Matsushita)

Le système dispose de 4 règles pour modifier la température, de 4 autres pour la maintenir, et d'autant pour le maintien du débit. Par exemple, si Dq est PB alors augmenter beaucoup le débit d'eau chaude. Les résultats en ce qui concerne les variations brusques de températures sont de 13.5° pour un thermostat classique, 8.5° pour un thermostat «à cire» et 0.8° avec le contrôle flou.

LE ROBOT CENTAURE (A.DELMAS CEA)

Ce robot est formé de deux parties articulées et de 4 patins, il peut se diriger dans un couloir et monter ou descendre des escaliers en étant télécommandé mesurant 16 distances et grâce à 8 contacteurs, les tangage, roulis, pliage des deux parties et position du plateau sont fusionnés en 8 entrées. 292 règles permettent alors de déterminer la configuration de l'appui du robot sur le sol afin que l'opérateur puisse le commander.

LE ROBOT KHEPERA

Ce robot miniature peut être télécommandé et mu par tout logiciel, il est utilisé pour l'expérimentation de ceux-ci. Par exemple, à l'Ecole polytechnique de Lausanne, il a été actionné par des règles floues. 3 entrées (des distances captées sur quelque centimètres plus la distance arrière à un obstacle, contrôlent 2 sorties qui sont les vitesses angulaires des deux roues par les règles (où RL = recule lent, AL = avance lente ...)) :

Distance gauche	Distance avant	Distance droite	Distance arrière	Moteur gauche	Moteur droit
grande	grande	grande		AR	AR
grande	grande	grande	petite	RL	AL
petite	petite	grande		RL	AL
petite	petite	moyenne		RL	AL
grande	petite	petite		AL	RL
moyenne	petite	petite		AL	RL
petite	petite	petite		RL	RL

CONTRÔLE DE CAMÉRA [Robert L. Jani 92]

Une simulation du contrôle de caméra lors de rendez-vous orbital, cherche à contrôler deux variations angulaires de son axe dans l'intervalle $[-6^\circ/s, 6^\circ/s]$. Les deux entrées sont respectivement les positions horizontale et verticale entre 0 et 170 pixels qualifiées par 5 prédicats, et l'altitude qualifiée par 3 prédicats. Les deux sorties sont commandées séparément mais exactement par les mêmes règles avec 3 prédicats. Ce sont x est NB et z est low alors NB, x est PB et z est low alors PB, (x est NS et z est (low ou med)) ou x est NB et z est (med ou high) alors NS, (x est PS et z est (low ou med)) ou x est PB et z est (med ou high) alors PS, la conclusion est ZE dans les 5 autres cas.

RECONNAISSANCE DE SIGNATURES [Simon, Levrat, Brémont, Sabourin 96]

Contrairement aux autres méthodes d'analyse de manuscrits, cette méthode n'a pas besoin de segmenter le mot (ici, une signature) en lettres et traits. On compare une signature (ramenée à un cadre 512×128) avec le maillage de l'original. Les niveaux de gris de chaque point peuvent être interprétés avec trois prédicats linguistiques «posé», «corps» et «levé» auxquels on rajoute le «fond». La proximité avec l'original peut être qualifiée par «proche», «moyen» ou «loin». Les prédicats sont trapézoïdaux. Un contrôleur flou min-max-centre de gravité classique de 9 règles intuitives telles que [si proximité est «loin» et dynamique est «levé» alors codage est petit] ou encore [si proximité «proche» et dynamique «posé» alors codage «grand»]. La sortie peut donner un niveau de confiance dans la signature proposée.

Citons encore la reconnaissance de 3000 caractères kanji par Panasonic-Hitachi, la boîte de vitesse floue de Nissan, la gestion d'ascenseurs par Mitsubishi (problème extrêmement complexe), le problème du ravitaillement en vol (Dassault), le contrôle du régime ralenti d'une voiture [Boverie 93], le contrôle longitudinal d'une voiture [Kim, Dickerson, Kosko 96], [Zalila 94] où 2 entrées l'écart de distance sur $[-100m, 100m]$ et l'écart de vitesse sur $[-10, 50 m/s]$ sont qualifiés par 5 prédicats et contrôlent une accélération découpée grâce à 6 prédicats. Le contrôle (toujours min-max-Sugeno) transversal, beaucoup plus délicat est également expérimentée grâce à 7 entrées ramenées à 3 distances exprimées par 5 prédicats non trapézoïdaux dépendant de la largeur du véhicule et de la voie [Zalila 96].

Le contrôle flou est encore présent dans la suspension de véhicules [Rao 97], l'outillage (recharge des perceuses Bosh, Makita, Hitachi), les aspirateurs Hitachi, les téléviseurs Sony, les caméras Sanyo, le contrôle de niveau en raffinerie Elf, les hauts fourneaux de Dunkerque, la détection d'incendie (Cerberus-Guignard)...

4.3. Réalisations en matériel

LE PROCESSEUR FLOU DE YAMAKAWA

Pour une règle à trois entrées du type :
si (x est X) et (y est Y) et (z est Z) alors (c est C), un circuit repéré par (X, Y, Z, C) est conçu suivant les huit labels principalement utilisés PB, PM, PS, ZE, NS, NM, NB représentés par des fonctions d'appartenance sur $[-1,1]$ triangulaires ou trapézoïdales de pentes ajustables, et aussi le label NG (ou ANY) toujours à 1 servant pour une variable non utilisée dans les prémisses.

Pour trois entrées x, y, z dans ce circuit, chacune des trois prémisses ressort un signal (concrètement un voltage entre 0 et 5V) puis l'opération min restitue un μ , qui réalise une troncature du prédicat C, c'est-à-dire une sortie sous forme d'un bus de 25 signaux représentant la distribution de possibilité de c sur $[-1,1]$.

Cette règle peut être affectée d'un poids.

15 circuits semblables sont prévus en parallèle, c'est ainsi que pour une application donnée, on pourra avoir un maximum de 15 règles, (un circuit non utilisé aura son prédicat C réglé sur une fonction d'appartenance nulle NUL), chaque règle ayant 1, 2 ou 3 prémisses partagera les mêmes entrées x, y, z . Les 15 bus de 25 fils représentant c_1 à c_{15} arrivent dans un circuit réalisant le OR ou ALSO par l'opération max, et produisant un bus de 25 dont un dernier circuit (defuzzyfier) est chargé de calculer la sortie c par centre de gravité. La réponse d'un circuit d'appartenance de u à U est de 30 ns, celle du circuit min, de 20 ns. Une boucle se fait en $1\mu s$ ce qui fait 1Mfip (méga fuzzy inférence par seconde). Expérimenté sur le pendule inversé, ce circuit donne satisfaction, mais surtout il a une vocation générale pour la réalisation de systèmes de contrôle flou très divers.

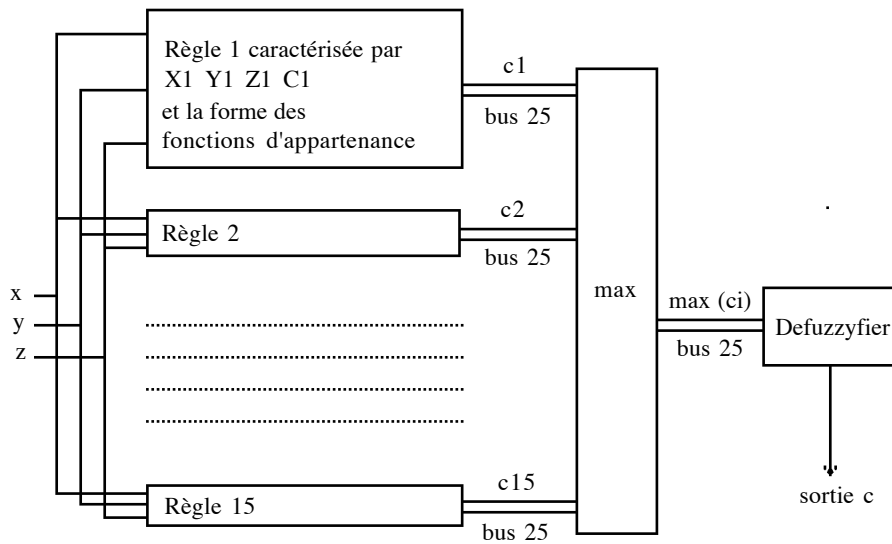


Figure 4.16 Processeur flou de Yamakawa.

LE PROCESSEUR SIEMENS

Sur le même principe, ce processeur (SAE 81C99) à 20MHz tient sur 7 mm² et infère 10 millions de règles par seconde. Il permet 4 entrées, 256 règles en nombre et formes variables ainsi que le choix de la méthode de défuzzyfication. La possibilité de linking permet jusqu'à 256 entrées, 64 sorties et 16384 règles. D'autres «fuzzy-chips» existent chez Omron, Thomson et à l'université de Karlsruhe.

LES CAPTEURS FLOUS DE L.FOULLOY [FOULLOY 93]

La couleur est un bon exemple pour lequel l'appréciation subjective dépend de plusieurs composantes. Un capteur ne peut pas renvoyer avec précision une couleur, mais il peut renvoyer des informations symboliques. Un capteur peut faire différentes mesures comme les fonctions d'appartenance r, v, b à trois prédicats flous triangulaires correspondant aux couleurs respectives rouge, vert et bleu (longueurs d'onde longues, moyennes et courtes). Il peut donc calculer la luminance $m = \max(r, v, b)$.

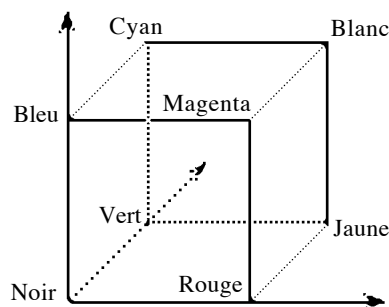


Figure 4.17 Le cube RVB des couleurs.

Définir le plan de chrominance consiste à se ramener de trois composantes à seulement deux grâce à une normalisation (division de r, v, b par la luminance m) puis à un point du plan dans lequel les couleurs rouge en (0, 1), jaune, vert, cyan en (0, -1), bleu et magenta forment un hexagone régulier. Les nouveaux axes sont la bissectrice des axes V et B, (axe des C1) et la bissectrice du trièdre (R = V = B) pour l'axe des C2. La transformation est :

$$\begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \frac{1}{m} \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & -\sqrt{3}/2 & \sqrt{3}/2 \end{bmatrix} \begin{bmatrix} r \\ v \\ b \end{bmatrix}$$

Le gris étant en (0, 0), on peut définir 7 prédicats «pyramidaux» ayant leurs noyaux aux points précités et leurs supports délimités chacun par les deux couleurs voisines ainsi que par le gris. A noter que des 3 intensités de couleurs R, V, B, on peut déduire 3 grandeurs plus perceptibles humainement que sont l'intensité $I = (R + V + B) / 3$, la saturation $S = 1 - \min(R, V, B) / I$ et l'argument polaire du point (C1, C2) qui vaut $H = \text{Arctan}(B - V) / (2R - V - B)$.

Chapitre 5

Problèmes d'apprentissage

Dans ce chapitre, nous examinons différentes approches pour acquérir de la connaissance à partir de données incomplètes et mal définies. Trois thèmes dominent, en premier lieu, celui de la classification, c'est un exemple type de problème mal spécifié, en effet, à partir d'un ensemble d'objets, qui, suivant la perception humaine, sont «plus ou moins» répartis en sous-ensembles aux frontières «plus ou moins bien nettes», on souhaite élaborer des méthodes pour arriver à une répartition satisfaisante en classes. A un classement en groupes prédéfinis, s'ajoute le problème plus ambitieux de l'énonciation de règles à partir d'exemples ayant des attributs vagues et de l'extraction de prédicats flous. Le deuxième thème est celui du connexionisme, c'est-à-dire des différents modèles de réseaux de neurones artificiels chargés d'apprendre à reconnaître une forme à partir d'un ensemble d'exemples sur lequel l'apprentissage a été effectué, ou bien à décider de la détermination de classes à partir de la donnée de prototypes et de reconnaître des objets voisins. Le troisième thème est celui des algorithmes génétiques ou plus généralement des stratégies d'évolution artificielle, heuristiques destinées aux problèmes d'optimisation, qui, comme on le verra, permettent d'apprendre un comportement simplement à partir de contraintes. Après avoir exposé les principales méthodes, nous envisageons comment tous ces sujets sont naturellement liés au flou.

5.1. Classement et classification

Etant donné un ensemble $X = \{x_1, x_2, \dots, x_n\}$ de n objets dans un espace métrique, et un nombre m de classes fixé à l'avance, le but du problème de classification ou coalescence est de séparer X en une partition $C = \{c_1, c_2, \dots, c_m\}$ suivant les «nuages de points» observés. Ce nuage se présente bien sûr (et ce n'est bien visible qu'en dimension deux) avec des zones de plus ou moins fortes densités. Par ailleurs, le problème en lui-même n'est pas vraiment bien défini, et des éléments sont manifestement toujours à cheval sur plusieurs classes, c'est pourquoi l'usage de classes floues semble bien indiqué, cependant diverses approches ne donneront pas nécessairement les mêmes résultats pour la raison que le but n'est pas véritablement spécifié.

Nous réservons le terme de «classement» pour l'opération consistant à ranger des objets dans des classes prédéfinies, et celui de «classification» à l'opération consistant à créer des classes, leur nombre n'étant même pas fixé initialement.

L'ALGORITHME DES K PLUS PROCHES VOISINS

Un premier algorithme très simple consiste à débiter avec un point (ou m points) et à classer chaque nouveau point dans la classe la plus représentée chez ses k (de l'ordre de 3 à 5) plus proches voisins (il ne s'agit donc pas de trouver les classes, ce qui est un vrai problème d'apprentissage, mais de classer de nouveaux candidats). Cette méthode est très simple, mais lente et son inconvénient est que les résultats sont sensiblement différents suivant les valeurs de k adoptées.

ALGORITHME FLOU DES K PLUS PROCHES VOISINS

Si les objets x_1, x_2, \dots, x_n sont déjà classés et si x est un nouvel exemple, on calcule les distances $d_j = d(x, x_j)$ en retenant l'ensemble V des indices des k exemples les plus proches, puis, λ étant plus grand que 1, on calcule une fonction d'appartenance du nouvel individu x relative à chaque classe C_i par :

$$\mu_i(x) = \frac{\sum_{j \in V} \mu_{ij} d_j^{\frac{2}{1-\lambda}}}{\sum_{j \in V} d_j^{\frac{2}{1-\lambda}}}$$

Chaque voisin intervient donc pour chaque classe avec un coefficient décroissant suivant sa distance à x . La décroissance étant réglée par le coefficient λ .

Il s'agit d'un raffinement de la méthode précédente. Cet algorithme prend les points à classer dans l'ordre où ils se présentent et forment des classes au fur et à mesure en s'inspirant de la méthode des k plus proches voisins, ceux-ci étant tempérés par une fonction décroissante de leur distance au point courant. L'avantage de cette méthode est que les points peuvent être ajoutés en temps réel et que le nombre de classes m n'est pas fixé à l'avance mais dépend bien sûr d'un certain nombre de paramètres.

On fixe un seuil d'appartenance s (par exemple 0.5), un paramètre a (de l'ordre de 0.45) et un nombre k (de l'ordre de 3) pour l'examen des plus proches voisins.

Au cours de l'algorithme, si une classe C_i est définie, pour un nouvel objet x (non encore placé), on calcule un degré d'appartenance par :

$\mu_i(x) = \max\{\mu_i(y) \cdot \exp(-ad^2(x, y) / d_m^2) / y$ parmi les k plus proches voisins de x} où d_m est la distance moyenne entre éléments de la classe C_i et a est un réel fixé entre 0 et 1. S'il est supérieur au seuil s, x est retenu dans la classe courante et d_m est alors mis à jour, x peut donc être placé dans une des classes déjà formées, ou bien rejeté (placé dans un ensemble R).

Pour chaque $x \in R$ on calcule donc les appartenances de x à tous les C_i pour i compris entre 1 et la valeur provisoire de m, s'il existe un x tel que $\mu_i(x) > s$, alors il est placé dans la classe C_i , sinon il est rejeté.

L'ensemble de rejet est alors analysé de la même façon.

Le premier calcul consiste en celui du centre de gravité G du nuage et on lui donne la valeur d'appartenance $\mu_1(G) = 1$, et on pose $m = 1$. On classe ensuite les points dans l'ordre décroissant de leur distance à G, on ne retient que ceux qui ont une $\mu_1(x) = \max\{\mu_1(y) \cdot \exp(-ad^2(x, y) / d_m^2) / y$ parmi les k plus proches voisins de x} supérieure à s.

Si le nombre de points restants dans R dépasse un seuil fixé à l'avance et que le nombre de classes m n'est pas trop important (une limite est également fixée), on initialise une nouvelle classe avec R comme la première.

La méthode s'achève avec une éventuelle fusion de classes C_i et C_j c'est à dire si $\forall x |\mu_i(x) - \mu_j(x)| < \text{seuil}$.

UN ALGORITHME DE CLASSIFICATION FLOUE EN DEUX CLASSES

Si $X = \{x_1, x_2, \dots, x_n\}$ sont n objets à classer dans un espace muni de la distance d et si l'on dispose de la donnée des distances mutuelles, on peut en déduire une relation floue de «proximité» définie par $r_0(x, y) = 1 - d(x, y) / \max(d)$. On pose R_0 la matrice symétrique n*n de cette relation. Initialement, on part de la partition U_0 formée par les n singletons $\{x_i\}$.

En prenant les deux objets les plus proches x_k et x_l , on construit une seconde partition U_1 formée par les n - 2 autres singletons $\{x_i\}$ et la paire $\{x_k, x_l\}$. On construit alors la matrice carrée R_1 (n - 1)*(n - 1) identique à R_0 sauf pour $r(\{x_i\}, \{x_k, x_l\}) = \min(r(x_i, x_k), r(x_i, x_l))$ et $r(\{x_k, x_l\}, \{x_i\}) = \max(r(x_k, x_i), r(x_l, x_i))$. Cette matrice n'est donc plus symétrique.

A chaque étape, à partir de la matrice R_{k-1} et de la distance d_{k-1} , on redéfinit une distance d_k entre les éléments de la partition U_k de fonctions d'appartenance μ et μ' par : $d_k(\mu, \mu') = \sum_{x \in X} |\mu^\lambda(x) - \mu'^\lambda(x)|$ avec un $\lambda > 0$. La relation floue r est modifiée $r(\mu, \mu') = 1 - d_k(\mu, \mu') / \max(d_k)$.

En prenant le couple de parties floues les plus proches au sens de d_{k-1} , on construit une partition U_k avec $\mu_C(x) = \max\{r(x, y) / y \in C\}$.

En itérant n - 2 fois ce procédé, on obtient deux ensembles flous [Dubuisson, Oppenham 80].

MÉTHODE DES CENTRES MOBILES

C'est la méthode classique en analyse des données, on se donne également un ensemble de points de R^n et le nombre m, choisi a priori, de classes voulues.

On choisit aléatoirement des centres G_1, G_2, \dots, G_m , puis on place dans la classe i, tous les points plus proches de G_i que des G_j pour $j \neq i$. On recalcule les centre G_i obtenus pour cette partition.

On recherche à nouveau la partition autour de ces nouveaux centres et ainsi de suite jusqu'à stabilisation relativement à un certain seuil, des G_i . Cette méthode très simple dans son principe a inspiré les algorithmes itératifs de Bezdek.

CLASSIFICATION AUTOMATIQUE

Le théorème de König-Huygens [Henry, Labordère 77] exprime la relation ci-dessous pour n points P_i répartis en m classes C_j (G_j est le centre de gravité de la classes C_j).

$$\sum_{i=1}^n \overrightarrow{GP_i}^2 = \sum_{j=1}^m \text{card}(C_j) \overrightarrow{GG_j}^2 + \sum_{j=1}^m \sum_{P_i \in C_j} \overrightarrow{G_jP_i}^2$$

Comme cette somme est fixe (G est le centre de gravité du système), on voit que le but de la classification étant d'avoir des classes les plus séparées possibles, donc de maximiser la première somme nommée d_{inter} , et donc de minimiser la seconde nommée d_{intra} . Cette somme étant la même quelle que soit la partition, on voit que d_{intra} a des chances de diminuer si deux classes sont fondues en une. Le problème est alors de trouver une heuristique séparant en classes, mais sans arriver à une seule classe.

Si on souhaite cette fois, avec plus d'ambition, que le nombre de classes soit lui-même trouvé automatiquement, une méthode itérative est donnée ci-dessous.

On donne un ensemble de points de R^n et la matrice de leurs distances deux à deux.

En procédant par itérations avec un incrément d_0 , on regroupe en classes tous les éléments p, q tels que $d(p, q) < d$ ($d = d_0$ initialement puis d sera incrémenté de d_0 à chaque étape). Le nombre de classes diminue donc à chaque fois.

Si, à chaque étape, on cherche les centres de gravités G_i de ces classes, on pose «l'inertie» de la partition : $I = \sum d^2(G_i, G)$ en notant G est le centre de tout le nuage.

On peut alors chercher l'étape (donc le nombre de classes) minimisant l'inertie I, car si I est minimum, cela entraîne que chaque dérivée partielle par rapport à une des distances est nulle : $I' = \sum d_i d'_i = 0$ signifie que chaque centre est stabilisé par rapport au centre G de tout le nuage.

La donnée d'une partition floue en m classes, équivaut à la donnée d'une matrice $m \times n$ (avec $m \leq n$) d'éléments génériques $\mu_{ik} = \mu_{C_i}(x_k)$ unistochastique c'est à dire vérifiant $\sum_{1 \leq i \leq m} \mu_{ik} = 1$ (la somme de chaque colonne est 1). Cette matrice a une et une seule valeur 1 par colonne (les autres étant 0) si et seulement si la partition est exacte. Le problème de trouver une telle partition réalisant un «bon équilibre» correspond à l'optimisation d'une heuristique.

Si maintenant, on tolère des valeurs autres que 0 ou 1, en notant β_i le «prototype» de la classe ou centre de gravité provisoire, la fonction :

$$J(B, C, X) = \sum_{1 \leq i \leq m} \sum_{1 \leq k \leq n} d^2(x_k, \beta_i)$$

n'est plus suffisante pour apprécier les nuances.

Toutes les méthodes de classification d'un ensemble X de n points d'un espace métrique consistent à tenter de minimiser une certaine fonction (analogue au d_{intra} du paragraphe précédent) définie à partir de X et des partitions C formées par un nombre m de classes. [Ruspini 69, 70] a étudié quelle genre de fonction des valeurs $\mu_{ij} = \mu_{C_i}(x_k)$ il convient de minimiser pour obtenir de bons résultats.

D'autres idées ont été avancées, ainsi l'optimisation de J peut être faite par algorithmes génétiques, un chromosome étant une liste $B = (\beta_1, \beta_2, \dots, \beta_m)$ de prototypes [Schulte 94], ceux-ci peuvent être employés pour obtenir des prototypes initiaux [Nascimento, Moura-Pirès 96].

Cependant nous exposerons ici la méthode du «fuzzy c-means» que [Bezdek 74, 81] a introduit pour minimiser la fonction :

$$J_\lambda(B, C, X) = \sum_{1 \leq i \leq m} \sum_{1 \leq k \leq n} (\mu_{ik})^\lambda d^2(x_k, \beta_i)$$

avec un coefficient $\lambda \geq 1$ (on prend généralement 2 dans les applications), la contrainte de partition floue étant toujours $\sum_{1 \leq i \leq m} \mu_{ik} = 1$ pour tout élément x_k , et $B = (\beta_1, \beta_2, \dots, \beta_m)$, β_i désignant le «prototype» de la classe C_i (ce sont tout simplement les barycentres suivant les pondérations μ^λ).

On peut partir initialement de centres de gravités aléatoires ou fixés suivant les concentrations du nuage ou encore régulièrement disposés dans une fenêtre englobant X .

La méthode consiste à chaque étape à minimiser J grâce à la partition courante.

Justification et rappel sur la méthode des multiplicateurs α de Lagrange :

Si $H(u_1, u_2, \dots, u_m)$ est une fonction différentiable de plusieurs variables dont on cherche les extremums relatifs alors que les variables sont liées par une relation $\Phi(u_1, u_2, \dots, u_m) = 0$, on construit la fonction auxiliaire (Lagrangien de H) $L(u_1, u_2, \dots, u_m) = H(u_1, u_2, \dots, u_m) - \alpha \Phi(u_1, u_2, \dots, u_m)$ qui est donc égale à H indépendamment de α . Un point singulier de H est donc en même temps point singulier de L et donc admet toutes ses dérivées partielles nulles.

Dans le cas de la fonction J , il est possible de raisonner pour chaque point x_k séparément, et donc de considérer la fonction :

$$H_k(\mu_{1k}, \mu_{2k}, \dots, \mu_{mk}) = \sum_{1 \leq i \leq m} (\mu_{ik})^\lambda d^2(x_k, \beta_i) - \alpha \left(\sum_{1 \leq i \leq m} \mu_{ik} - 1 \right)$$

dont les dérivées partielles sont faciles à calculer (on note $d_{ik} = d(\beta_i, x_k)$) :

$$\frac{\partial H_k}{\partial \alpha} = \left(\sum_{i=1}^m \mu_{ik} \right) - 1 = 0 \quad \text{et} \quad \frac{\partial H_k}{\partial \mu_{ik}} = \lambda \mu_{ik}^{\lambda-1} d_{ik}^2 - \alpha = 0 \quad \text{d'où on déduit} \quad \mu_{ik} = \left(\frac{\alpha}{\lambda d_{ik}^2} \right)^{\frac{1}{\lambda-1}}$$

$$\text{Et comme} \quad \sum_{i=1}^m \mu_{ik} = \left(\frac{\alpha}{\lambda} \right)^{\frac{1}{\lambda-1}} \sum_{i=1}^m \left(\frac{1}{d_{ik}^2} \right)^{\frac{1}{\lambda-1}} = 1, \quad \text{on déduit} \quad \mu_{ik} = \frac{1}{\sum_{j=1}^m \left(\frac{d_{ik}^2}{d_{jk}^2} \right)^{\frac{1}{\lambda-1}}}$$

On vérifie que chaque μ_{ik} est compris entre 0 et 1, sauf si le dénominateur est nul si $d_{ik} = 0$ auquel cas, il est normal de prendre $\mu_{ik} = 1$.

La distance d peut être calculée par $\|x - y\|^2 = (x - y)^t A (x - y)$ où A est une matrice symétrique définie et positive $n \times n$. En reprenant la fonction J , pour C et X fixés, et en considérant la dérivée suivant une direction quelconque w à partir du point $B = (\beta_1, \beta_2, \dots, \beta_m)$, c'est la dérivée par rapport à h :

$$H'(B, w) = \sum_{1 \leq k \leq n} (\mu_{ik})^\lambda (x_k - \beta_i - hw)^t A (x_k - \beta_i - hw) \Big|_{h=0}$$

la nullité de cette dérivée donne $-2 \sum_{1 \leq k \leq n} (\mu_{ik})^\lambda (x_k - \beta_i)^t A w = 0$ pour tout w , et donc on a $\sum_{1 \leq k \leq n} (\mu_{ik})^\lambda (x_k - \beta_i) = 0$ d'où en développant :

$$\beta_i = \left(\sum_{1 \leq k \leq n} (\mu_{ik})^\lambda x_k \right) / \left(\sum_{1 \leq k \leq n} (\mu_{ik})^\lambda \right)$$

L'algorithme consiste donc en résumé, à modifier à chaque étape les μ_{ik} en :

$$\mu_{ik} = \frac{1}{d_{ik}^{\left(\frac{2}{\lambda-1}\right)} \sum_{j=1}^m d_{jk}^{\left(\frac{2}{1-\lambda}\right)}}, \quad \text{en supposant au moins deux classes distinctes.}$$

Si le dénominateur est nul, on posera $\mu_{ik} = 1$. Ces calculs sont suivis d'une normalisation pour obtenir $\sum_{1 \leq i \leq m} \mu_{ik} = 1$ pour chaque point x_k . Puis à mettre à jour les prototypes, ceux-ci devenant $\beta_i = \left(\sum_{1 \leq k \leq n} (\mu_{ik})^\lambda x_k \right) / \left(\sum_{1 \leq k \leq n} (\mu_{ik})^\lambda \right)$.

L'arrêt peut être décidé lorsque chaque variation de μ_{ik} est inférieure à un seuil donné comme par exemple 1%.

REMARQUES

Pourquoi un tel plutôt λ que 1? car cela accentue les faibles niveaux d'appartenance, donc contribue à mieux séparer les classes. Lorsqu'il est proche de 1, les degrés d'appartenance sont proches de 0 ou de 1, on a donc un renforcement des classes floues vers des classes nettes, en revanche, lorsqu'il est plus grand (de l'ordre de 2, 3, 5 ...) la convergence est plus rapide, mais les classes restent d'autant plus floues que λ est grand. L'expérience montre que si les prototypes sont initialement trop rapprochés, des classes ont tendance à fusionner, en tout cas, le résultat dépend largement de ces positions initiales. Néanmoins, cette démarche de classification donne de bons résultats, car elle est appliquée dans la pratique à des ensembles d'objets sur lesquels on a déjà une certaine idée a priori de leur classement, c'est pourquoi, les prototypes initiaux peuvent grossièrement être placés dès le départ.

La distance d est quelconque, en particulier celle de Hamming $d(x, y) = \sum |x_i - y_i|$, qui est la plus simple, mais elle peut être calculée avec la norme $\|Z\|^2 = Z^t A Z$ où t est l'opérateur de transposition, A est la matrice identité pour la distance euclidienne ou bien l'inverse de la matrice de covariance des exemples pour la distance de Mahalanobis, en ce cas on a $d^2(x_k, \beta_i) = (x - \beta_i)^t C_i^{-1} (x - \beta_i)$ et C_i est la matrice $\sum_{1 \leq k \leq n} (\mu_{ik})^\lambda (x_k - \beta_i) (x_k - \beta_i)^t / (\sum_{1 \leq k \leq n} \mu_{ik}^\lambda)$ [Gustafson, Kessel 79].

Si A est une matrice symétrique, comme on sait que $Z^t A Z = 0$ définit une quadrique, [Krishnapuram, Frigui, Nasraoui 95] ont étendu le fuzzy c-means à la reconnaissance de classes qui ont la forme de coniques planes en mettant à jour les matrices A correspondants aux contours à chaque étape. La classification est à l'heure actuelle très employée en traitement d'images, afin de reconnaître des zones similaires (contours, coins, régions homogènes ...) et en reconnaissance des formes en général.

La classification «possibiliste» au sens de Krishnapuram est la méthode du c-means dans laquelle on retire la contrainte $\sum_{1 \leq i \leq m} \mu_{ik} = 1$, pour chaque exemple k .

DU CLASSEMENT À LA CLASSIFICATION

Deux idées simples peuvent être exploitées pour accompagner la méthode précédente, tout d'abord l'initialisation des prototypes peut être arbitraire suivant un positionnement régulier dans l'espace contenant les exemples. Il est toutefois préférable de les placer approximativement sur les nuages de points que l'on peut déceler. Pour ce faire, on regarde les distributions des projections des points sur chacun des axes, prenons par exemple, en dimension 2, n points de $[-1, 1]$.

En choisissant un nombre de tranches de l'ordre de la dizaine, nous pouvons obtenir les histogrammes relatifs aux coordonnées des exemples sur les deux axes. Il est aisé de repérer les modes de ces histogrammes, (p tranches modales pour x et q tranches modales pour y) puis par croisement, de fixer pq prototypes initiaux.

Une stratégie simple consiste alors à projeter les points à classer sur d'autres axes comme $y = ax$ pour $a = 1, -1, 2, -1/2, -2, 1/2, \dots$ pour vérifier que ces prototypes initiaux correspondent à des tranches modales jusqu'à ce qu'il n'y ait plus d'élimination de faux prototypes.

L'inconvénient réside dans l'arbitraire sur la fixation du nombre de tranches des histogrammes, heureusement, une éventuelle fusion de classes voisines peut y pallier dans la suite du traitement.

À l'issue de cette méthode, le nombre m de prototypes est provisoirement fixé, elle s'accompagne ensuite des itérations de celle de Bezdek, en examinant à chaque fois si deux prototypes ne sont pas suffisamment voisins.

Si, pour $i < j$, $|\beta_i - \beta_j| < \vartheta$ (fixé au regard de $[-1, 1]$ par exemple à 0.1) alors pour tout exemple x_k , μ_{ik} est remplacé par $(\mu_{ik} + \mu_{jk}) / 2$ et le prototype β_j est supprimé, donc le nombre m des prototypes est décrémenté.

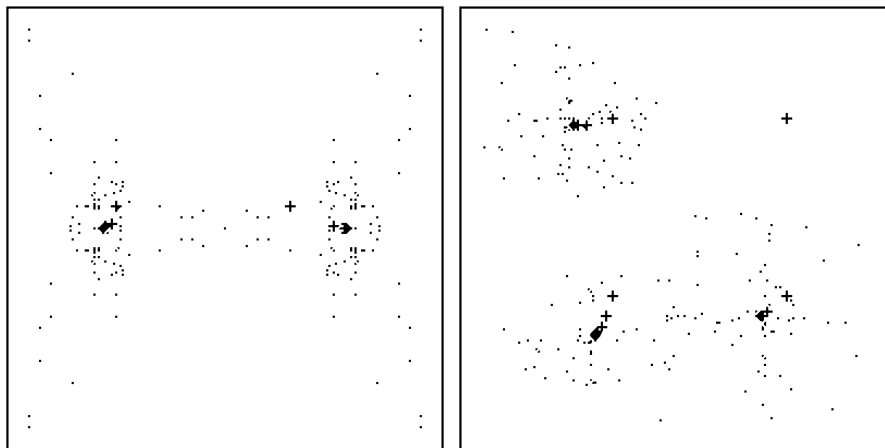


Figure 5.1 Exemple classique d'un papillon symétrique à gauche. Les deux classes sont immédiatement détectées grâce à la position des axes. Quelques itérations suffisent à déplacer les centres vers leur position définitive au seuil fixé. À droite, 300 points de $[-1, 1]^2$ sont placés en trois classes, grossièrement en L, quatre prototypes sont initialisés par croisement, une seule projection supplémentaire suffit à éliminer la classe parasite. 5 itérations suffisent pour assurer la convergence des prototypes.

Dans ces exemples, λ est fixé à 3 et on montre que la trajectoire des prototypes (marqués par des croix) converge bien vers les «centres» (gros points) des nuages.

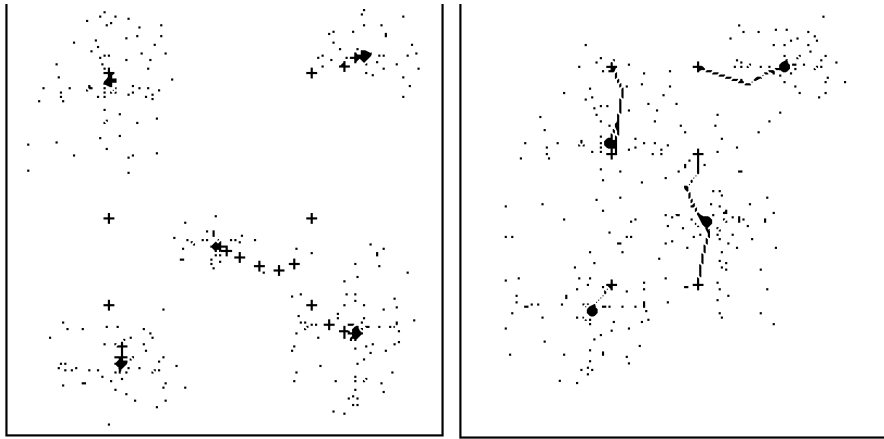


Figure 5.2 Cinq classes relativement séparées sont données à gauche. Six sont détectées par croisement des modes sur les deux axes, une est éliminée (la croix au centre à gauche), celle de centre droite, se déplaçant vers le petit nuage central. A droite, les points sont beaucoup plus diffus et le prétraitement a été anihilé de façon à observer la fusion de classes. De 6 classes fixées initialement (croix), il ne subsiste que 4 classes à la fin. Les trajectoires suivies par les prototypes laissant leur trace.

APPLICATION À LA RECONNAISSANCE DE CONTOURS

La méthode de Bezdek a été utilisée à divers propos, notamment la segmentation d'images (pour la reconnaissance de poissons) afin de reconnaître des contours [Shahin, Demko, Coisne 95] où on définit le contour (flou) ∂C d'une classe floue C par :

$$\mu_{\partial C}(x) = \min(\mu_C(x), \max\{\min(\mu_{V(x)}(y), 1 - \mu_C(y)) / y\})$$

où $V(x)$ est un voisinage déterminé de x (par exemple x et les 8 pixels autour de lui). Etre dans le contour de C signifie être dans C et avoir un voisin y qui ne l'est pas. On définit également la frontière (floue) entre deux classes C et C' par :

$$\mu_{F(C, C')}(x) = \min(\mu_{\partial C}(x), \max\{\min(\mu_{V(x)}(y), \mu_{\partial C'}(y)) / y\})$$

(x est dans la frontière de C s'il est dans son contour et a un voisin y dans le contour de C'). Le meilleur taux de reconnaissance d'images a été obtenu avec les t-normes et t-conormes de Zadeh (les formules ci-dessus) comparées avec les t-normes de Lukasiewicz, probabiliste et de Weber au lieu de min et max.

5.2. Les arbres de décision flous

La recherche de règles à partir d'un ensemble d'exemples est un problème connu et d'ailleurs résolu depuis longtemps par l'algorithme simple ID3 [Quinlan 79, 86]. L'introduction du flou est alors tout à fait naturelle dans la mesure où un problème réel comportera des individus dont les attributs qualitatifs sont incertains (couleur, profession ...) ou numériques (taille, prix ...). Dans ce dernier cas, il se peut qu'une partition pour l'attribut soit donnée a priori (petit si inférieur à a , grand si supérieur à b ...) et que cette partition soit floue, auquel cas le problème est d'examiner l'ensemble des exemples donnés avec leurs attributs afin d'en extraire des règles floues (si X est de taille grande et de catégorie moyenne alors le prix de X est assez élevé). Mais le problème se complique encore si les exemples ne sont donnés qu'avec leurs attributs numériques, comment trouver automatiquement une bonne partition de chaque univers d'attribut avec des prédicats flous de façon à obtenir des règles floues simples et parlantes ? On voit qu'il s'agit d'un but typiquement mal défini, mais diverses méthodes vont y parvenir avec un certain succès.

UN PREMIER ALGORITHME D'EXTRACTION DE RÈGLES FLOUES À PARTIR D'UNE BASE DE RÈGLES, MÉTHODE DE WANG

Cette méthode inspirée de [Michalski 83] peut s'utiliser dès que l'on dispose d'un ensemble d'exemples de la forme $(x_1, x_2, \dots, x_n, y)$, on sait donc de quelle conclusion « y » on souhaite parler, et que l'univers de chaque prémisse x_i est déjà partitionné par une famille de prédicats A_{i1}, A_{i2}, \dots

En ce cas on construit une règle $(A_{1,k1}, A_{2,k2}, \dots, A_{n,kn}, C_k)$ à partir de chaque exemple $(x_1, x_2, \dots, x_n, y)$, en choisissant simplement pour prédicat $A_{i,ki}$, celui qui est le plus vérifié par la valeur x_i , et pour prédicat C_k celui qui est le plus vérifié par y .

Le degré de cette règle est défini tout simplement comme le produit des fonctions d'appartenance aux dits prédicats apparaissant dans la règle pour l'exemple considéré $(x_1, x_2, \dots, x_n, y)$.

Dans le cas de l'établissement de règles de Sugeno, une autre solution consiste à prendre comme degré la moyenne de ces conclusions exactes, pondérée par les degrés d'appartenance aux dits prédicats, pris à une puissance supérieure à 1 [Nozaki 97]. Une puissance de l'ordre de 10 accentue fortement ce degré.

On supprime ensuite les règles inutiles, une règle $R = (A_{1,k1}, A_{2,k2}, \dots, A_{n,kn}, C_k)$ est dite inutile s'il existe une autre règle $R' = (A'_{1,k1}, A'_{2,k2}, \dots, A'_{n,kn}, C'_k)$ telle que $C'_k \subset C_k$ et pour chaque prémisse, on a $A_{i,ki} \subset A'_{i,ki}$.

Le conflit (règles ayant les mêmes prémisses) peut être réglé en prenant celle qui possède le plus grand degré [Wang 97].

L'étape suivante consiste à amplifier chaque règle $(A_{1,k1}, A_{2,k2}, \dots, A_{n,kn}, C_k)$ en remplaçant $A_{i,ki}$ par $(A_{i,p}$ ou $A_{i,q}$ ou $A_{i,r} \dots)$ disjonction de tous les prédicats relatifs à x_i qui ne sont présents dans aucune autre règle, puis par $\neg(A_{i,s}$ ou $A_{i,t} \dots)$ si cette expression équivalente est plus simple.

Naturellement cette méthode ne donnera de bons résultats que si elle est raffinée par une optimisation de la définition des prédicats, ce qui est plus difficile à réaliser.

Le principe de cet algorithme [Quinlan 79], est de créer un arbre de classification à partir d'un ensemble d'exemples dont chacun est une liste de descripteurs. Prenons l'exemple de 8 appartements caractérisés par :

Exemple	Ensoleillement	Taille	Balcon	Prix
A1	non	studio	oui	cher
A2	oui	studio	non	moyen
A3	oui	2 pièces	oui	moyen
A4	non	appartement	oui	moyen
A5	oui	appartement	oui	moyen
A6	oui	studio	oui	cher
A7	oui	appartement	non	moyen
A8	non	studio	non	moyen

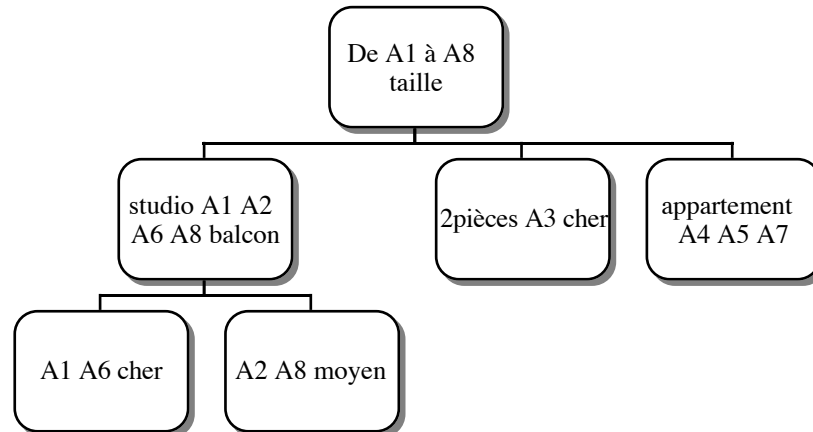


Figure 5.3 Exemple d'arbre résultant d'un choix d'attributs.

On crée un arbre de décision en choisissant l'un des attributs (par exemple la taille) avec autant de fils qu'il y a de valeurs (ici studio, 2pièces, appart) et on recommence pour chaque noeud tant que la partie des exemples concernés par ce noeud n'est pas uniforme du point de vue d'un attribut fixé à l'avance, (par exemple on veut des règles déterminant des tranches de prix). L'inconvénient de cet algorithme est que la forme de l'arbre dépend des choix successifs et reste limité à des problèmes de classification assez simples. En fait il faut calculer la séquence d'attributs commune à tous les chemins partant de la racine, permettant d'obtenir l'arbre minimal. Pour des données symboliques, l'algorithme est facile à implémenter et donne de bons résultats. Généralement, on décide à chaque noeud de l'arbre, de sélectionner le prochain attribut comme celui qui apporte le plus d'information (entropie minimale) $I(a) = -\log p(a)$, donc l'attribut «a» qui a la probabilité minimale.

LE CONCEPT D'ENTROPIE

Exemple, pour coder 32 cartes, 5 questions suffisent pour choisir des questions discriminantes, n questions suffisent afin de déterminer un objet particulier parmi 2^n , on dira que la quantité d'information pour identifier un objet dans un ensemble de m éléments est $I(E_m) = \log_2(m)$. Dans toute la suite, les logarithmes choisis sont dans une base supérieure à 1 quelconque.

Pour une variable aléatoire discrète, elle est définie par

$$I(X) = -\sum_x \text{pr}(X = x) \log(\text{pr}(X = x))$$

L'entropie conditionnelle de X connaissant Y est :

$$I(X / Y) = -\sum_j \text{pr}(y_j) \sum_i \text{pr}(x_i / y_j) \log(\text{pr}(x_i / y_j))$$

Si p est une mesure de probabilité, l'entropie de Shannon (1948) pour une partie finie A est :

$$H(A) = -\sum_{x \in A} p(x) \cdot \log_2 p(x)$$

C'est donc une fonction dans R^+ , sur un univers continu avec une densité de probabilité f, ce sera alors :

$$H(A) = -\int_{x \in A} f(x) \cdot \log_2 f(x) \cdot dx$$

Cette notion a été élargie à un ensemble flou F (entropie de Higashi-Klir) avec :

$$I(F) = \int_{0 \leq \alpha \leq 1} \log_2(\text{card } F_\alpha) \cdot d\alpha$$

On a définit également une «mesure de confusion» pour une masse m par :

$$C(m) = -\sum_{A \in F} m(A) \cdot \log_2 cr(A) \quad \text{où } F \text{ est l'ensemble des parties focales ainsi qu'une «mesure de dissonance» } E(m) = -\sum_{A \in F} m(A) \cdot \log_2 pl(A) \quad \text{où } cr \text{ et } pl \text{ sont définis comme au chapitre 2.2.}$$

L'entropie est en outre élargie à la détermination d'une classe parmi deux classes d'effectifs n et m avec :

$$H = -n / (n+m) \log(n / (n+m)) - m / (n+m) \log(m / (n+m))$$

En particulier si des classes C_1, \dots, C_m sont déterminées dans un ensemble et qu'un attribut A peut prendre les valeurs v_1, \dots, v_p , on définira l'entropie conditionnelle par rapport à une valeur par $d(v_i) = -\sum_{1 \leq j \leq m} p(c_j / v_i) \log p(c_j / v_i)$ qui est une mesure du désordre local lorsque l'attribut vaut v_i .

Une mesure du désordre global est alors l'entropie moyenne par rapport à l'attribut A à savoir $H(C / A) = -\sum_{1 \leq i \leq p} p(v_i) \sum_{1 \leq j \leq m} p(c_j / v_i) \log p(c_j / v_i)$ sa minimisation peut rendre le service de guider un apprentissage. Enfin on peut définir un :

$$\text{gain}(A) = H - H(C / A).$$

Dans le cas où chaque classe est déterminée par une valeur précise de l'attribut, on a $H(C/A)$ qui est nulle, c'est la valeur minimale pour le «désordre» et il est maximal lorsque toutes les décisions sont équiprobables (c'est d'ailleurs 1 si on prend le logarithme en base p). On a donc une notion adéquate que l'on peut chercher à minimiser grâce à un algorithme de recherche opérationnelle.

Au cas où les classes sont floues, l'entropie d'une partition $C = \{C_i / 1 \leq i \leq m\}$ en classes floues est $H(C) = - \sum p(C_i) \log p(C_i)$ avec $p(A) = \int_{x \in U} \mu_A(x).dp(x)$.

D'autres définitions d'entropie ayant la même propriété de minimalité pour une probabilité 1, les autres nulles et de maximalité pour équidistribution, peuvent être données, ainsi $1 - \sup\{p_i\}$, ou bien $\sum p_i(1 - p_i)$ (entropie quadratique), ou encore $\sum \sum_{i < j} (p_i p_j)^{1/2}$ (entropie de Bhattacharyya).

LA «RELAXATION» Basée sur l'entropie, cette méthode consiste à laisser évoluer un système formé d'un certain nombre d'exemples x classés en m classes en minimisant une «certaine énergie» qui peut être l'entropie de Shannon $H(x, C) = - \sum \mu_i(x) \log \mu_i(x)$ pour chaque x, cette entropie, qui est une mesure du désordre, étant nulle lorsque x est vraiment dans une classe et maximale si x est équidistribué dans toutes les classes de la partition $C = \{C_1, C_2, \dots, C_m\}$.

La mesure globale du désordre, si aucun μ_{ik} n'est nul, est l'entropie de la partition :

$$H(C, X) = - \sum_{1 \leq k \leq n} \sum_{1 \leq i \leq m} \mu_{ik} \log(\mu_{ik}) / n \text{ comprise entre } 0 \text{ et } \log(m).$$

Elle est nulle dans le cas d'une seule classe, et maximale dans le cas d'une équidistribution (on peut rendre ce maximum égal à 1 si on utilise le logarithme en base m).

ENTROPIE STAR Si $C = (c_1, \dots, c_m)$ est une partition floue de E fini, $\text{card}(E) = n$, et $\text{card}(c_i) = \pi_i = \sum \mu_{c_i}(x)$, on a donc une probabilité d'être dans la classe c_i qui s'exprime par $p_i = \pi_i / n$, et on définit [Ramdani 94] $I^* = - \sum p_i \log(p_i)$.

On montre $I^*(CC') = I^*(C).I^*(C')$, si C et C' sont deux partitions floues, pour les m.m' classes de CC'.

Si un attribut A admet les valeurs v_1, \dots, v_p , on pose comme pour l'entropie conditionnelle :

$I^*(C/A) = - \sum_{1 \leq i \leq p} p(v_i) \sum_{1 \leq j \leq m} p(c_j / v_i) \log p(c_j / v_i)$ qui va mesurer le degré d'incertitude d'une prise de décision (dire dans quelle classe on est), connaissant les valeurs imprécises v_i . (Si A exact, alors $I^* = I$).

Si I^*_i est l'entropie de la décision sachant que A est v_i , alors $I^*(C/A)$ est la moyenne de ces entropies $p_1 I^*_1 + p_2 I^*_2 + \dots + p_p I^*_p$.

On a de plus les résultats :

I^* nulle (minimale) \Leftrightarrow Chaque I^*_i nulle \Leftrightarrow Chaque v_i est entièrement déterminé

I^* maximale \Leftrightarrow Indépendance stochastique entre les classes et les valeurs de A

Cas particulier de partition :

Si S est une relation de similarité dans E ensemble de n éléments, muni d'une distribution de probabilité p,

$$H(p/S) = - \sum_{1 \leq i \leq n} p_i \log \sum_{1 \leq j \leq n} S(x_i, x_j)$$

On a toujours $0 = H(P/S^0) \leq H(P/S) \leq H(P) = H(P/S^1)$ où les relations extrêmes sont les relations triviales $S^0(x, y) = 1$ (une seule classe) et :

$S^1(x, y) = [si (x = y) \text{ alors } 1 \text{ sinon } 0]$ (n classes).

EXTRACTION DE RÈGLES FLOUES

Une classification à l'aide de l'algorithme ID3, lorsque les données sont imprécises et floues n'a que des performances réduites. Trouver des règles floues à partir d'un ensemble d'exemples (où les attributs sont donnés avec des degrés), pose d'abord le problème du partitionnement du domaine de chaque attribut numérique, et donc de la définition de prédicats flous sur chaque domaine. Trouver la meilleure coupure entre deux prédicats «petit» et «grand», trouver des fonctions d'appartenance qui les définissent, faire abstraction des cas pathologiques (on souhaite toujours des ensembles flous convexes) et pour finir adapter l'algorithme ci-dessus pour obtenir des règles floues, ne sont pas des problèmes simples.

En ce qui concerne la détermination des prédicats flous relatifs à l'univers d'un attribut, le problème se pose aussi bien dans le cas discret où des attributs A_j ont des valeurs exactes $V_{j1}, V_{j2}, \dots, V_{jk}$ que dans le cas continu où il faut trouver une ou plusieurs coupures pour éventuellement déterminer les valeurs floues (petit, moyen, grand) sur lesquelles on pourrait fonder des classes. Une première méthode consiste à construire l'histogramme des valeurs prises par l'attribut A. Cet histogramme peut lui-même être flou dans la mesure où une valeur d'attribut pour un exemple donné serait pondéré par un coefficient. Les deux histogrammes aux bornes de l'univers de A sont extrapolés.

[Marsala 94] indique une méthode d'inférence automatique de partitions floues fondée sur la morphologie mathématique.

Coupure : la meilleure coupure peut être trouvée en minimisant l'entropie-star.

Arrêt : Une méthode simple à chaque noeud de l'arbre, consiste à arrêter la construction du sous-arbre en ce noeud si les exemples qui y sont représentés le sont à un seuil près tel que 90% (degré cumulé par «min» des degrés avec lequel cet exemple est arrivé en ce noeud depuis la racine) ou bien s'ils sont peu nombreux [Ishigashi 96].

EXEMPLE

Après un prétraitement consistant à séparer des chiffres manuscrits et en isoler de 1 à 6 éléments parmi 12 types, [Chi, Yan 96] obtiennent grâce à un arbre de 828 noeuds, un certain nombre de règles. Les types reconnus sont V (segment vertical), H (horizontal), P (montant), N (descendant), de type C, D, A, U, S, Z, O et «courbe» pour le reste. Une matrice fixée empiriquement indique que par exemple un segment N peut être à 1 un N, mais à 0.25 une ligne de type H et à 0.1 une des courbes S ou Z. Les règles sont du type «si le plus grand segment est de type O et le type du deuxième est C et l'ordonnée normalisée du centre de l'image est supérieure à 0.586 alors c'est un 6».

5.3. Réseaux de neurones et neuro-flou

LE PERCEPTRON

Le perceptron est une machine parallèle de conception déjà ancienne [MacCulloch, Pitts 43], [Rosenblatt 58], réalisant un modèle de réseau de neurones assez simplifié, dans lequel une première couche de cellules (rétine) recevant des informations binaires (0 ou 1) transmettent cette information à des cellules de décision (éventuellement une seule reconnaissant ou non par 0 ou 1 l'image sur la rétine) par l'intermédiaire d'une couche cachée (par la suite de plusieurs autres couches). Chaque cellule reçoit la somme des informations de la couche précédente, pondérée par des poids attachés à chacun des arcs, et modifie son état suivant que cette somme pondérée dépasse ou non un certain seuil. C'est le prototype du système auto-organisateur grâce à différentes stratégies de modification des poids en vue de minimiser l'erreur entre la sortie du réseau et la sortie attendue pour un exemple, ces réseaux dits «à couches» sont appliqués à l'apprentissage «supervisé» (on donne un lot d'exemples constitués d'entrées de données et des sorties correspondantes que l'on souhaite obtenir, base sur laquelle le réseau s'entraîne, puis on le teste sur une base dite de généralisation beaucoup plus vaste).

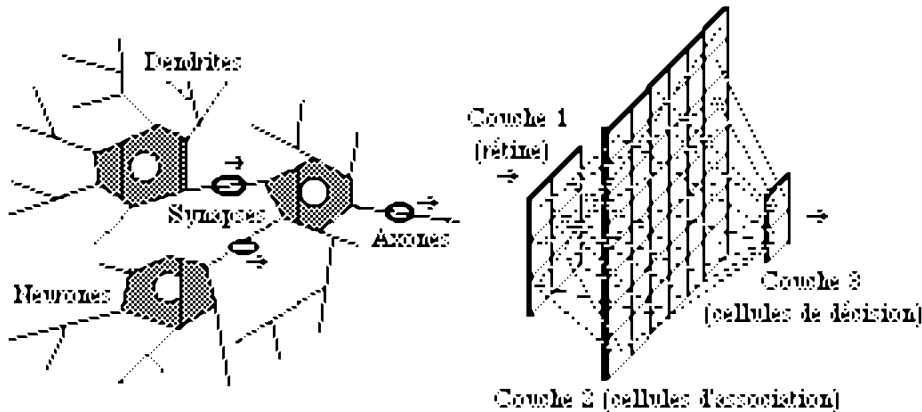


Figure 5.4 Neurones et liaisons synaptiques

Si maintenant on définit un état de la rétine par une fonction d'appartenance μ dans $[0, 1]$ (par exemple des gris au lieu du noir et blanc), on pourra définir la propriété à reconnaître par un prédicat flou P sur les ensemble flous de la rétine, et le réseau constituant une fonction F , celle-ci ne sera plus booléenne, mais à valeurs dans l'intervalle $[0, 1]$. On pourra toujours dire que la propriété est vérifiée à $F(e_1, \dots, e_n)$ près.

Exemple : si la rétine comporte 4 cases ternaires (noir ou gris ou blanc d'où 81 entrées possibles), on peut définir un prédicat P_1 «foncé» en donnant une fonction d'appartenance suivant le nombre de points (0 pour un blanc, $1/8$ pour un gris, et $1/4$ pour un noir). Par ailleurs un prédicat P_2 «pauvre en symétrie» par 5 valeurs $1 - \text{nbsym}/4$, alors la reconnaissance des figures foncées et pauvres en symétrie au seuil de 0,6 se fera par $\min(p_1, p_2) \geq 0,6$.

Le perceptron à une couche caché est capable de reconnaître des classes linéairement séparables.

LES RÉSEAUX À COUCHES

Les réseaux de neurones sont donc des schématisations de cerveaux, dont on postule qu'ils sont constitués d'un très grand nombre d'unités simples en interaction. L'un des modèles les plus séduisants et les plus utilisés pour la reconnaissance des formes, est celui du réseau à couches (la plupart du temps une ou deux couches cachées). Dans chacune des couches, chaque neurone est relié à ceux de la couche précédente dont il reçoit les informations, et à chaque neurone de la couche suivante à qui il transmet des informations, mais il n'est pas relié aux autres neurones de sa propre couche. On considère que chaque neurone reçoit par ses «dendrites» une certaine activation électrique dont une somme pondérée constitue l'entrée e (un certain poids $w_{i,j}$ sera affectée à la liaison entre les neurones i et j). Par suite, le neurone passe à un certain «état» $s = f(e)$ qui sera la mesure de sa sortie portée par son «axone» (f peut être modélisée par une simple fonction de seuil ou par une fonction du type arctangente ou tangente hyperbolique). L'axone transmet ensuite sa valeur par l'intermédiaire de «synapses» aux dendrites d'autres neurones. (Les poids mesurent en fait l'efficacité des synapses) L'apprentissage du réseau, consiste en une modification de ces poids au fur et à mesure des expériences, c'est à dire de la confrontation entre le vecteur sortant de la dernière couche et celui qui est attendu en fonction d'un vecteur d'entrée.

EXEMPLE : on voudrait qu'en traçant sur une grille carrée des dessins formés par une seule ligne, on ait en sortie deux neurones dont le premier réponde 1 si la courbe est fermée et 0 sinon, le second donnant 1 s'il y a un point double et 0 sinon.

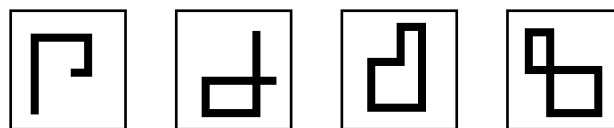


Figure 5.5 Dessins codés successivement par 00, 01, 10 et 11.

Les réseaux à couches sont surtout utilisés en reconnaissance de formes (écriture manuscrite ...)

Les différentes règles (Hebb, Widrow-Hoff) de modification des poids utilisées cherchent à exprimer l'erreur entre la sortie réelle et la sortie désirée puis à calculer la contribution à cette erreur de chaque poids comme dérivée partielle. Pour Widrow, si l'entrée est (x_1, x_2, \dots, x_m) d'un réseau à deux couches, et l'erreur globale $E = \sum_{1 \leq i \leq p} (y_i - s_i)^2$ alors, après calcul de la dérivée partielle de E par rapport à w_{ij} , la règle est de modifier chaque poids w_{ij} en $w_{ij} - \eta(s_j - y_j)x_i$.

Dans un réseau multicouche, chaque neurone n, à l'intérieur, ayant $e = \sum w_{i,n} s_i$ pour entrée, où i parcourt les neurones de la couche précédente, aura une sortie $s = f(e)$. Lorsque l'on propose le vecteur $X = (x_1, x_2, \dots, x_m)$, à la première couche, la dernière restitue le vecteur $S = (s_1, \dots, s_p)$ alors qu'on attend $Y = (y_1, \dots, y_p)$ comme réponse.

Le but de cet algorithme est d'exprimer l'erreur quadratique $E = \sum_{1 \leq i \leq p} (y_i - s_i)^2$ et de chercher à la minimiser en modifiant chaque poids w, suivant l'influence qu'il a sur E : $w(t + dt) - w(t) = -\eta \cdot \partial E / \partial w$ où η est un coefficient positif, le «pas» du gradient. Le choix de ce pas est délicat (il est en général pris assez grand entre 0.01 et 0.1 au début de l'apprentissage pour être ensuite diminué). On cherche donc à exprimer ce gradient qui est le vecteur formé par tous les $\partial E / \partial w$
Plaçons nous entre le neurone i d'une couche et un neurone n fixé :

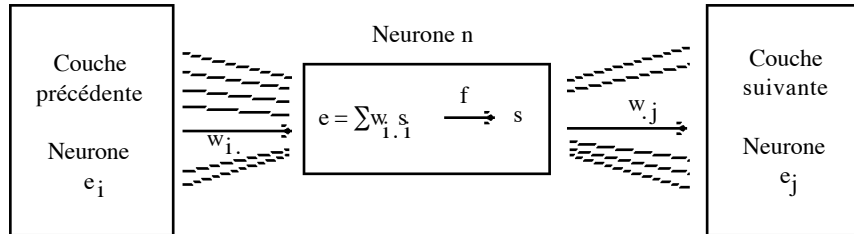


Figure 5.6 Schéma d'un neurone formel.

Pour la commodité de la lecture, on notera i l'indice parcourant la couche précédente, et j celui de la couche suivante, si d_n est la dérivée partielle de l'erreur E par rapport à e, on a :

$$\frac{\partial E}{\partial w_{i,n}} = \frac{\partial E}{\partial e} \cdot \frac{\partial e}{\partial w_{i,n}} = \frac{\partial E}{\partial e} \cdot \frac{\partial \sum_j w_{j,n} \cdot s_j}{\partial w_{i,n}} = \frac{\partial E}{\partial e} \cdot s_i = d_n \cdot s_i$$

Car tous les termes de la sommation pour $j \neq i$ ne dépendent pas de $w_{i,n}$.

Si le neurone n'est pas en sortie, alors on calcule :

$$d_n = \frac{\partial E}{\partial e} = \sum_{j \text{ suivant}} \frac{\partial E}{\partial e_j} \cdot \frac{\partial e_j}{\partial e} = \sum d_j \frac{\partial e_j}{\partial s} \cdot \frac{\partial s}{\partial e} = \sum d_j \cdot w_{n,j} \cdot f'(e)$$

Et si n est en sortie :

$$d_n = \frac{\partial E}{\partial e} = \frac{\partial \sum_{j=1}^p (Y_j - S_j)^2}{\partial e} = 2(S_n - Y_n) \cdot f'(e)$$

La règle d'apprentissage est donc à chaque présentation d'exemple, (X, Y), de mesurer la sortie S, l'erreur E, et de modifier chaque poids (de i à j) $w_{i,j}$ en le remplaçant par $w_{i,j} - \eta d_j s_i$ avec $d_n = (\sum d_j w_{n,j}) f'(e_n)$ pour les indices j en aval de n, sauf si n est en sortie auquel cas $d_n = 2(S_n - Y_n) f'(e_n)$. Il y a rétro-propagation de l'erreur commise dans la mesure où les modifications vont devoir être effectuées de la dernière couche vers la première.

Ce processus est répété sur plusieurs exemples (X, Y) jusqu'à ce qu'il y ait convergence suivant un seuil fixé. Les poids initiaux sont en principe distribués aléatoirement autour de 0,5. Expérimentalement les meilleurs résultats, de manière assez générale, ont été obtenus pour des réseaux où le nombre de poids est grossièrement de l'ordre du nombre d'exemples, ainsi que pour la stratégie (dite stochastique) de rétropropagation après chaque présentation d'exemple. La stratégie par lots consiste au contraire à n'opérer de rétropropagation qu'après la présentation de tous les exemples, en cumulant les erreurs.

RÉSULTATS RÉCENTS

Le théorème de Cybenko (89) montre que l'ensemble des fonctions réalisées par un réseau à une seule couche cachée dont la fonction d'activation est continue et vérifie $f(-\infty) = 0$ et $f(+\infty) = 1$, est dense dans l'ensemble des fonctions continues à support compact (borné suffit) [Hornik 88, 89].

Cet ensemble est fermé dans l'ensemble de toutes les fonctions définies sur R^m dans le cas où l'activation est gaussienne (Poggio T. Girossi F.).

Le théorème de Palmer (91) montre enfin que les réseaux à une seule couche cachée suffisent pour les problèmes linéairement séparables, et deux couches cachées suffisent pour approcher n'importe quelle fonction.

RÉSEAU RBF «RADIAL BASIS FUNCTION NETWORK»

Ce sont des réseaux [Broomhead D.S. Lowe D. 88], [Poggio 90] et [Platt 91] à trois couches dont la première possède p neurones correspondant à la dimension des vecteurs d'entrées.

La couche cachée possède un nombre de neurones qui peut être le nombre n d'exemples proposés pour l'apprentissage ou un peu moins, (ce nombre peut même être variable si on crée un neurone supplémentaire chaque fois que l'erreur dépasse un certain seuil) mais sa particularité est que les fonctions de transfert qui leur sont associées sont des fonctions gaussiennes avec un centre et un écart-type propre à chacun de ces neurones et qui sont modifiés en fonction des performances. La troisième couche constitue la

sortie (un ou plusieurs neurones) telle que les poids associés constituent dans le cas d'une sortie, en quelque sorte les conclusions des «règles floues». Une méthode pour mettre au point ces fonctions gaussiennes est d'utiliser la rétropropagation, mais ce n'est pas la seule possible : les moyenne et variance constituent deux poids arrivant à chaque neurone, et qui vont être modifiés à chaque rétropropagation. Le but de l'apprentissage est que chaque neurone de la couche cachée devienne un spécialiste reconnaissant un exemple, il le laissera passer et non les autres.

Une autre méthode inspirée des moyennes floues a été proposée par [Mustawi 92] pour le problème suivant de classement. Supposons que l'on dispose de n exemples X_1, X_2, \dots, X_n tous dans \mathbb{R}^p , et que ces points-exemples soient classés parmi m classes déjà définies. Le réseau va être entraîné sur ces exemples par un algorithme dépendant des paramètres η et c , afin de généraliser par la suite.

0) Le réseau est formé de 3 couches dont la première possède p neurones (une entrée X est acceptée à raison de ses p composantes), la couche cachée possède n neurones et la couche de sortie m neurones correspondant aux m classes prédéfinies. A chaque exemple X_i , est associé un neurone de fonction matricielle de transfert :

$f(X) = \exp[-(1/2)(X - C_i)^t Q_i (X - C_i)] = \exp[-(1/2)\sum_{1 \leq k \leq p} (x_k - c_{ik})^2 / \sigma_{i,k}^2]$ où C_i est fixé initialement en X_i , et Q_i est la matrice diagonale des inverses des variances $\sigma_{i,k}^2$. $\sigma_{i,k}$ est l'écart-type de la gaussienne correspondant au i -ième neurone de la couche cachée pour la k -ième composante (il peut être initialement fixé à 1). On peut remarquer que $f(x)$ étant l'exponentielle d'une somme est donc le produit (t-norme probabiliste) des degrés d'appartenance des différentes composantes du vecteur d'entrée X aux différents prédicats gaussiens.

1) Du premier au dernier de ces neurones cachés i :

2) On examine tour à tour tous les neurones j correspondant à des exemples x_j de la même classe. On envisage alors le regroupement des deux neurones en calculant le barycentre c_k entre c_i et c_j .

On calcule la distance r_k entre c_k et l'exemple de la même classe qui est le plus éloigné (rayon de la classe), ainsi que la distance d_k entre c_k et le barycentre de la classe distincte la plus proche.

Si $d_k > \eta r_k$ avec la constante de Mustawi η située entre 1 et 3, alors le regroupement des deux neurones est définitif, sinon on recommence en 2) avec un autre neurone j jusqu'à une acceptation ou bien jusqu'à ce que tous les neurones aient été examinés.

3) Revenir en 1) jusqu'à ce qu'il n'y ait plus de regroupement possible.

4) Le nombre et les centre c_i des neurones cachés étant déterminés, on prend alors $\sigma_i = cr_i$ avec une constante qui peut être expérimentalement fixée à la valeur $c = 1.2$.

Cette architecture et cet algorithme sont à l'heure actuelle employés en reconnaissance des formes en concurrence avec les réseaux multicouches à rétropropagation. Leur intérêt, pour le contrôle flou, est de réaliser une interpolation avec un petit nombre de neurones cachés [Cho, Wang 96].

RÉSEAU DE HOPFIELD [Hopfield 82], [Kurbel 94]

Il s'agit d'une structure adéquate à la reconnaissance de m classes par n neurones complètement connectés (il n'y a pas de couches). Soit un ensemble de n «neurones» d'état $x_i = \pm 1$, tous connectés entre eux par des «poids» $w_{i,j} = w_{j,i}$ avec $w_{i,i} = 0$. A chaque instant, l'entrée du neurone x_i , est la somme pondérée des états de tous les autres neurones qui lui sont connectés, c'est à dire $\sum_{1 \leq j \leq n} w_{i,j} x_j$. La règle d'évolution du réseau est donnée par la modification simultanée de tous les neurones suivant laquelle x_i est remplacé par signe de $(\sum_{1 \leq j \leq n} w_{i,j} x_j)$.

Une modification asynchrone des neurones est étudiée dans [Abdi 94].

Hopfield démontre la décroissance au cours de l'évolution de l'énergie du réseau définie par la fonction $H = (-1/2)\sum_{1 \leq i, j \leq n} x_i x_j$ et que pour un état initial, le réseau converge vers un état «attracteur».

Soit maintenant un ensemble X formés par m vecteurs de $\{-1, 1\}^n$ appelés prototypes, le problème est de constituer un réseau susceptible de reconnaître des vecteurs voisins (bruités) de ces prototypes. On peut alors montrer que chacun de ces prototypes possède un «bassin attracteur» dans lequel des vecteurs bruités (correspondant à un état du réseau) vont évoluer avec une convergence plus ou moins rapide grâce à la règle ci-dessus vers un des exemples de X . Appelons S la matrice $n \times m$ formée par les vecteurs exemples (et S^t sa matrice transposée), son élément générique $s_{i,j}$ désigne donc la i -ième composante de l'exemple j .

On définit $W = SS^t - I_n$ la matrice des poids, et on voudrait obtenir une stabilité s'exprimant par un «point fixe» S , c'est à dire $WS = S$. W possède une diagonale nulle $w_{i,i} = 0$ et soit $w_{i,j}$ l'élément i, j de SS^t c'est à dire $\sum_{1 \leq k \leq m} s_{ik} s_{jk}$

On remarque que pour un rang k entre 1 et m ,

$$\sum_{1 \leq j \leq n} w_{ij} s_{ik} = \sum_{1 \leq j \leq n} (\sum_{1 \leq p \leq m} s_{ip} s_{jp}) s_{jk} = \sum_{1 \leq j \leq n} \sum_{1 \leq p \leq m} s_{ip} (s_{jp} s_{jk}) = \sum_{1 \leq p \leq m} s_{ip} \sum_{1 \leq j \leq n} (s_{jp} s_{jk})$$

Dans le cas où les exemples sont deux à deux orthogonaux c'est à dire si $SS^t = I$, alors cette dernière expression est $s_{i,k}$ ce qui signifie que le réseau est stable (chaque état de neurone est égal à son entrée). En fait une hypothèse plus large sur le fait d'avoir le même signe suffit.

Une autre façon d'initialiser les poids est de prendre (théorème de Penrose) la matrice de poids $W = SS^+ + Z(I_n - SS^+)$ où Z est quelconque $n \times n$ et S^+ est la pseudo-inverse de S , car alors $WS = SS^+S + ZS - ZSS^+S = S$ Empiriquement de bons résultats sont obtenus pour la proportion $0.03 n \leq m \leq 0.14 n$, au delà, le réseau a tendance à oublier.

LES RÉSEAUX RÉCURRENTS

Il s'agit d'un réseau [Elman 90] très simple avec une couche d'entrée, une couche cachée et une couche de sortie, utilisant l'algorithme de rétropropagation. Cependant il existe une autre couche dite de «contexte» qui est une simple copie de la couche cachée, de telle sorte qu'à chaque propagation, la couche cachée reçoit les entrées accompagnée du «souvenir» de l'état de la couche cachée à l'étape antérieure. Le nouvel état de la couche cachée est alors recopié dans la couche de contexte et la rétropropagation est opérée de façon classique. Une autre architecture a été réalisée où la couche d'entrée est cette fois accompagnée du souvenir de la couche de sortie

relative à l'étape antérieure. Ces réseaux ont été utilisés pour le traitement des séries temporelles, mais aussi en informatique linguistique.

RÉSEAU DE KOHONEN

Le but de cette architecture [Kohonen 89] est un apprentissage cette fois non supervisé, où les entrées possibles sont classées. Il n'y a que deux couches, l'entrée (x_1, \dots, x_n) comportant autant de neurones que la dimension n des exemples, et la sortie comportant p neurones en compétition. On définit le voisinage V_j de la sortie j , par exemple si $p = 16$, chaque neurone peut avoir son voisinage décrit par ses 4 voisins, puis 3, puis 2. Chaque poids w_{ij} est modifié par la règle suivante :

A chaque sortie j (avec $1 \leq j \leq p$), on calcule la distance $\sum_{1 \leq i \leq n} (x_i - w_{ij})^2$ et si j_0 est la sortie pour laquelle cette distance est minimale (le neurone «vainqueur»), on met les poids à jour pour tous les i , mais seulement pour les j dans le voisinage de j_0 par la règle $w_{ij} \leftarrow w_{ij} + \eta(x_i - w_{ij})$. Ainsi donc le neurone j_0 se spécialise pour la reconnaissance d'une certaine classe de données en entrée, mais aussi ses neurones voisins, dans une moindre mesure.

Puis on fait décroître la taille du voisinage, et enfin η qui mesure le taux d'apprentissage, doit décroître aussi de 0.8 initialement avec un décrement de 0.005 par exemple. La convergence doit se faire en une dizaine de cycles vers un réseau reconnaissant p classes.

LE RÉSEAU NEURO-FLOU

Plusieurs structures [Sanchez 90], [Berenji 92], [Yamakawa, Uchino, Miki, Kusagani 92] ont été proposées, nous présentons la plus claire de [Glennec 91]. L'esprit de ce réseau est d'intégrer a priori, une connaissance même imparfaite, mais permettant un apprentissage plus rapide qu'avec des poids initiaux aléatoires. Un réseau neuro-flou est donc un réseau de 4 couches (3, 11, 25, 1) conçu de la façon suivante pour deux entrées exactes x, y il fournira une sortie u :

(Si n prédicats et p entrées, on aura un réseau $p + 1, np + 1, n^p, 1$).

La couche 1 possède les deux entrées x, y et un troisième neurone toujours à 1.

La couche 2 est constituée par les 10 états $\mu_{A_i}(x), \mu_{B_i}(y)$ dans l'hypothèse où on a choisi 5 prédicats NB, NS, ZE, PS, PB. Ceux-ci sont gaussiens donc entièrement déterminés par des couples (a_i, b_i) de réels fixés initialement par exemple à $(a, a), (a, a/2), (a, 0), (a, -a/2), (a, -a)$.

Cela constitue une famille de prédicats se coupant à la hauteur $\exp(-a^2/16)$ soit 0.5 si $a = 4 \ln^{1/2}(2)$.

a_i est le poids initial depuis x , et b_i celui depuis y vers chaque neurone de la seconde couche. Le second terme du couple est le poids initial du neurone fixe 1 vers chacun de ceux de la seconde couche. Si la fonction de transfert est donnée par $f(t) = \exp(-t^2)$ on obtient bien les prédicats gaussiens en sortie c'est à dire les états.

Ainsi l'état du premier neurone de la seconde couche est $f(ax + b) = \exp(-(ax + b)^2) = \mu_A(x)$ pour un certain prédicat gaussien A .

Les poids vers la 3^e couche sont tous fixés à 1.

Un 11^e neurone est fixé à 1 et les poids le joignant à la couche aval sont fixés à -1.

La couche 3 est constituée par 25 règles a priori (si 2 entrées et 5 prédicats).

Chaque entrée sera donc $\mu_A(x) + \mu_B(y) - 1$.

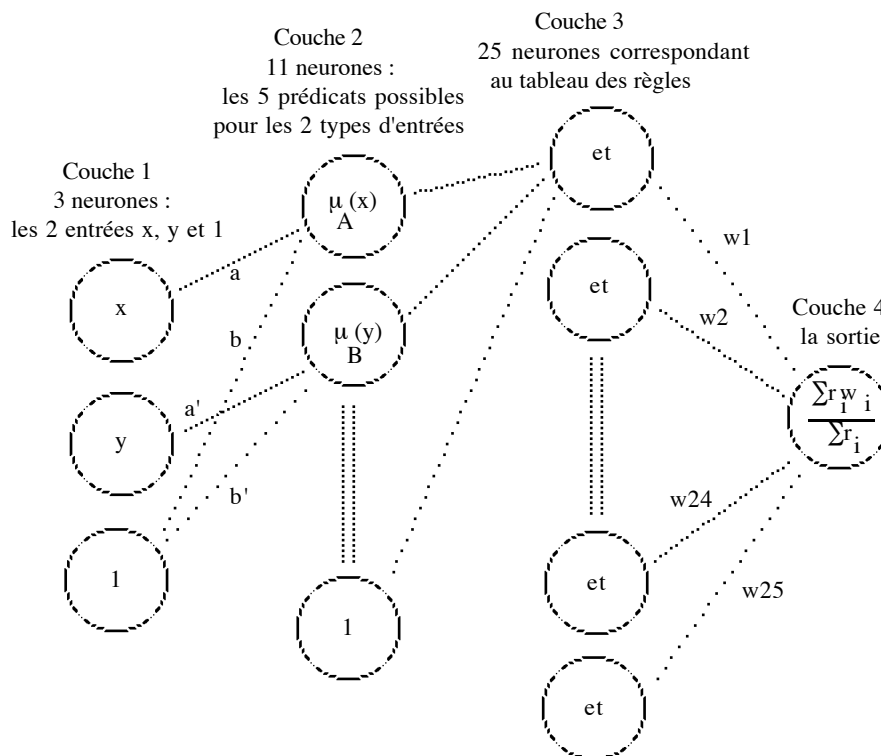


Figure 5.9 Schéma général du réseau neuro-flou.

La fonction de transfert de la troisième couche est $g(t) = 1 / (1 + \exp(2 - 4t))$ elle approche ainsi $\max(0, t)$. Par ce stratagème, la sortie de chaque neurone réalise donc approximativement une version lissée (c'est à dire dérivable) du «et» de Lukasiewicz.

Les 25 derniers poids w_1, \dots, w_{25} sont mis initialement de façon intuitive comme conclusions numériques des 25 règles. Si r_i est le niveau de satisfaction de la i -ième règle l'entrée du dernier neurone est $\sum r_i * w_i$.

La sortie sera alors calculée comme la moyenne $(\sum r_i * w_i) / (\sum r_i)$.

La différence avec l'algorithme de rétropropagation est que les fonctions de transfert sont distinctes à chaque couche et que la dernière est variable, son calcul est bien spécifique de la méthode de Sugeno.

Cette architecture a été appliquée par Berenji au pendule inversé avec 4 entrée (x, x', θ, θ') . Elle est testée dans bien d'autres domaines, notamment au domaine bancaire.

Tous les poids peuvent changer au cours de l'apprentissage y compris ceux du «et», cependant il est prudent de n'appliquer la rétropropagation que par étape, après un tel apprentissage séparé (couche par couche) [Brunet 96] a pu observer une déformation de la famille des 5 prédicats NB, NS, ZE, PS, PB vers une stabilisation dans laquelle NS tend à se confondre avec NB, PS avec PB et où on a une perte de la symétrie aussi bien en ce qui concerne les valeurs des poids en sortie que pour ZE légèrement déplacé vers la gauche.

REMARQUE, La sigmoïde de pente p en $(m, 1/2)$ est peut être donnée par la fonction $f(t) = 1 / [1 + \exp(-4p(t - m))]$.

LA MÉTHODE DU GRADIENT STOCHASTIQUE

Un réseau $(2, 10, 25, 1)$ est constitué de la même façon, mais avec la véritable t -norme choisie et des prédicats triangulaires, seuls les poids de la dernière couche seront modifiés grâce à :

$\Delta w(t) = -\epsilon [y(t) - d(t)] r_i / \sum r_i + \eta \Delta w(t-1)$ où $r_i / \sum r_i$ est la force relative de la règle y , la sortie réelle et d la sortie désirée. ϵ et η sont le «gain» et le «moment». (La méthode du gradient de Cauchy consiste à suivre la ligne de plus grande pente en chaque point pour optimiser une fonction différentiable).

L'intérêt de la méthode par rapport à la rétropropagation classique est la diminution du calcul et aussi que pour deux entrées, si la famille de prédicats est une partition floue, alors de 1 à 4 règles sont modifiées à chaque fois seulement.

5.4. Algorithmes génétiques et stratégies d'évolution

Les algorithmes génétiques [Bagley 67], [Goldberg 82, 89] sont inspirés par les lois de l'évolution en biologie où par croisements, la population d'une espèce se renouvelle et où des mutations accidentelles de très faibles probabilité peuvent modifier dans un sens ou dans l'autre les aptitudes d'un individu et par suite favoriser ou non la possibilité qu'il aura de se reproduire. Un individu muté dans le sens d'une plus grande performance (suivant certains critères, ne serait-ce que la longévité) aura alors plus de chance de se reproduire et donc de transmettre à sa descendance son nouveau code.

C'est la fameuse théorie darwinienne de la sélection naturelle [Monod 70], [Dessalles 96]. Cette théorie a récemment inspiré une méthode stochastique appliquée à la résolution des problèmes non seulement mal aisés à résoudre mais à définir. C'est avec un certain succès que des problèmes mal posés se voient offrir des solutions souvent inattendues par une simulation d'évolution :

Pour un problème quelconque d'optimisation, les solutions vont d'abord être codées sous forme de chaînes de caractères appelées «chromosomes», les caractères étant les «gènes». L'algorithme consiste alors à reproduire par étapes une «population» P de chromosomes.

Au départ on prendra donc des chromosomes plus ou moins arbitraires, générés aléatoirement ou réalisant déjà un semblant de solution intuitive, ou bien de réelles solutions mais non optimales.

De génération en génération, la population doit s'améliorer suivant une fonction d'évaluation dictée par le problème, on peut alors classer la population suivant cette fonction pour permettre aux meilleurs chromosomes de se reproduire.

L'illustration est assez simple à faire sur le problème de la minimisation d'une fonction numérique définie sur l'intervalle $[a, b]$. Toutes les méthodes de descentes du gradient (adaptées aux fonctions différentiables et convexes), ou bien les méthodes stochastiques telle que celle du recuit simulé consistent à itérer l'évaluation de la fonction sur une suite de points «voisins» en partant d'un point initial arbitraire. Si la fonction n'est pas convexe et présente plusieurs minima, on ne peut garantir la convergence vers le minimum global même si une certaine remontée est tolérée suivant une probabilité à définir comme c'est le cas dans le recuit simulé.

Nous signalons à présent deux méthodes d'optimisation présentant un certain rapport avec les algorithmes génétiques qui suivent.

MÉTHODES STOCHASTIQUES D'OPTIMISATION, LE RECUIT SIMULÉ

Cette méthode [Kirkpatrick 83], [Aarts Korst 89], [Reeves 93] consiste à obtenir par itérations successives le minimum absolu d'une fonction, elle est inspiré d'une technique de refroidissement consistant à accepter dans certains cas une remontée de la fonction (pour ne pas descendre trop vite vers un minimum local). En thermodynamique la probabilité d'avoir une augmentation d'énergie ΔE (la fonction que l'on veut minimiser suivant la loi de Boltzmann) est $e^{-\Delta E/\theta}$. On accepte un état voisin s_1 augmentant la fonction, dans la mesure où la probabilité ci-dessus est décroissante suivant ΔE . Le paramètre de contrôle (la température) va décroître, ce qui fait que pour un même ΔE , la probabilité d'accepter une remontée diminue suivant le refroidissement. Plus précisément :

a) Choisir un état initial s_0

b) Faire le «palier» consistant à répéter n fois

Chercher un voisin s_1 de s_0 , on calcule $\Delta = f(s_1) - f(s_0)$

Si $\Delta < 0$ alors $s_0 \leftarrow s_1$

Si $\Delta > 0$ alors on accepte la même affectation de s_0 avec une probabilité $e^{-\Delta/\theta}$

c) On abaisse la température θ et on réitère ces paliers jusqu'à un test d'arrêt défini par le problème.

La difficulté de cette méthode, qui par ailleurs donne de bons résultats, réside dans la détermination des paramètres. On peut montrer la convergence sous certaines conditions reliant n et la loi de décroissance de la température.

La température initiale peut être déterminée par une probabilité, par exemple de $1/2$, d'accepter une hausse au cours du premier palier, si m est la valeur moyenne de ces hausses, alors $\theta_0 = m / \ln(2)$. Le nombre d'itérations sur chaque palier peut être de l'ordre de la centaine, la loi de décroissance de la température est généralement prise comme une suite géométrique telle que $\theta_{n+1} = 0,9 \theta_n$

MÉTHODE TABOU

On part de la même façon d'une «solution» initiale quelconque x . A chaque point courant x du domaine de définition du problème, on cherche grâce à une discrétisation du problème, le meilleur voisin non tabou de x (ce sera le point suivant). La liste des «tabous» est simplement constituée d'états antérieurs sur lesquels on ne souhaite pas revenir. Cette liste doit avoir une taille fixe et chaque état n'y reste donc qu'un nombre fini d'itérations. On la gère en file : chaque mise-à-jour y place un nouveau venu et en retire le plus ancien [Glover 86].

Les algorithmes génétiques et plus généralement toutes les stratégies d'évolution partent d'une population de points (10 sur la figure ci-dessous), il est clair que moyennant un temps de calcul multiplié, les chances d'obtenir plusieurs minima et le minimum global en sont augmentés. Mais ces stratégies ne se contentent pas de poursuivre des itérations séparées dans plusieurs régions de «l'espace de recherche» en recréant des «niches écologiques» si l'on poursuit l'image de l'évolution naturelle des espèces, elles innovent dans l'exploration de cet espace de recherche en se croisant et en brouillant les éventuelles solutions par des «opérateurs génétiques».

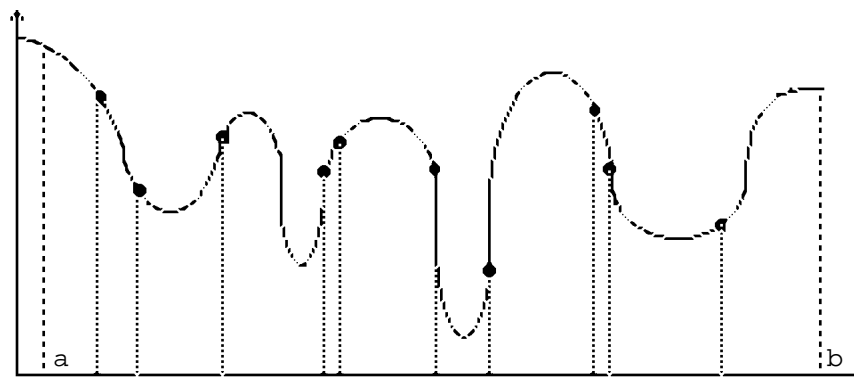


Figure 5.10 Population aléatoire dans un intervalle

LES STRATÉGIES D'ÉVOLUTION EN VUE D'UNE OPTIMISATION

Les différentes stratégies se distinguent sur les 7 points suivant :

a) Définition du problème : la fonction d'évaluation.

Naturellement les algorithmes génétiques seront utilisés pour l'optimisation de fonctions non triviales (non continues, bruitées et multimodales), sans propriétés de dérivabilité connues et définies sur des domaines plus complexes qu'un intervalle. Elles sont utilisées là où les méthodes analytiques ont échoué, mais aussi sur les problèmes symboliques pour lesquels les méthodes de dénombrement sont trop coûteuses. En fait pour des problèmes d'optimisation multicritère, nous verrons plus loin qu'il est très difficile de donner une fonction d'évaluation, le seul fait de bien poser le problème, n'est pas trivial.

b) Codage des solutions

Les algorithmes génétiques ont débuté avec un codage binaire des éventuelles solutions de façon à ce qu'il ait une grande possibilité d'exploration de l'espace de recherche avec simplement de petites transformations comme la mutation consistant à modifier un 1 en 0 ou le contraire au sein de la chaîne chromosomique. Si les codages les plus adaptés aux stratégies que l'on a en tête ne sont pas ceux qui permettent la plus grande capacité d'exploration, le manque de signification est l'inconvénient du codage binaire. En fait, comme cela a été pratiqué par la suite sous le nom d'algorithmes évolutifs, le codage peut être quelconque ainsi un nombre peut être codé par sa suite décimale et rester «lisible» et une solution d'un problème symbolique peut rester une suite finie (ou non) de symboles. Le «génotype» ou «génome» est l'ensemble des gènes, la question est d'établir le lien avec le «phénotype» ou ensemble des caractéristiques de morphologie et de comportement.

c) Choix de la population initiale

On peut partir d'un ensemble de solutions approchées déjà connues, mais d'une manière générale il est souvent avantageux de partir d'une population aléatoire (sa taille peut aller de quelques dizaines à quelques centaines). Dans l'algorithme «standard» qui est une reproduction assez fidèle de l'évolution, la population est toujours de grande taille, mais avec beaucoup de recopies d'individus identiques. [Cerf 94] montre que pour ces algorithmes, la vitesse de convergence est de l'ordre de la taille de la population.

d) Choix des opérateurs

Les transitions entre générations se font par «opérateurs génétiques» c'est à dire des «fonctions aléatoires» de $P \rightarrow P$ ou $P^2 \rightarrow P^2$ si P est la population.

L'opérateur le plus simple est la mutation, un gène est remplacé par un autre aléatoirement choisi, par exemple au chromosome ABCBBAC \rightarrow ABEBBAC. Des mutations bien particulières peuvent être envisagées suivant le problème, ainsi la transposition : deux gènes aléatoirement choisis sont permutés ABCEBACD \rightarrow ABAEBCCD, ou l'inversion : une coupure est choisie et les deux

parties sont permutées ABCBDEA → DEAACBCB, l'inversion partielle : une partie du chromosome est inversée ABCDEFDCD → ABCDDFECD etc.

Le cross-over : deux parents engendrent deux enfants en échangeant une partie choisie aléatoirement : AABEDABC, DACBCDAE → AACBCABC, DABEDDAE

C'est l'opérateur le plus puissant car intuitivement cela correspond à deux solutions comportant chacune une bonne partie du code, aussi en regroupant ces parties peut-on améliorer la solution globale.

La reproduction : si les valeurs des individus sont v_1, v_2, \dots, v_n et que $p_i = v_i / (\sum v_k)$, en tirant au hasard dans la population un numéro entre 1 et n avec cette distribution (p_1, p_2, \dots, p_n) le chromosome choisi est tout simplement recopié à la génération suivante.

D'autres opérateurs peuvent être imaginés notamment des opérateurs respectant la cohérence dans la représentation du problème en chromosomes (exemple : respecter un aspect injectif ou bijectif...).

Les opérateurs non uniformes [Michalewicz 92], consistent à effectuer par exemple des mutations de plus en plus faibles, c'est à dire si l'ensemble des gènes est métrisable, à opérer dans des voisinages de plus en plus fins lors du remplacement d'un gène par un autre.

Parmi toutes les idées, on peut citer celle du «poisson de corail» [Koza 94] consistant à déclarer «étalon» les 10% individus les plus performants et à réserver le croisement entre un étalon et un autre individu afin de favoriser le brassage et l'exploration de la population.

e) Mode d'application des opérateurs

Dans l'algorithme génétique désormais appelé «standard» les opérateurs sont appliqués suivant une certaine probabilité choisie au départ et suivant le principe de roue de loterie biaisée, on donne l'avantage aux meilleurs individus (stratégie «ranking»), le cardinal de la descendance d'un individu est a (en moyenne) pour le meilleur, et b pour le moins bon. La différence $a - b \in [1, 2]$ est appelée la pression de sélection, et il a été observé une uniformisation de la population d'autant plus rapide que cette pression est grande.

Afin de garder aussi longtemps que possible tous les minimums d'une fonction, il est possible de faire une partition de la population en «niches» [Horn 93] et d'effectuer périodiquement des «migrations» entre les sous-population [Franz 72]. Un autre point de vue est de changer la fonction d'évaluation [Goldberg 87] :

Si sh est une fonction de partage (sharing function) sur $[0, 1]$ alors on peut voir comme une «accentuation» de la négation avec $sh(x) = 1 - x^\alpha$ où $\alpha < 1$ pour avoir $sh(x) < x$ ou $\alpha > 1$ pour le contraire $sh(x) > x$. Soit enfin $sh(x) = 0$ si $x > 1$.

Si n est la taille de la population et σ un paramètre déterminant la taille des niches, au cas où il est possible de définir une distance d dans l'espace de tous les chromosomes, alors [Horn 93] modifie la fonction d'évaluation. Si f_0 est l'évaluation à la génération courante où les individus sont c_1, c_2, \dots, c_n alors la nouvelle évaluation f_1 pour la génération suivante est :

$f_1(c_i) = f_0(c_i) / \sum_{1 \leq k \leq n} sh [d(c_i, c_k) / \sigma]$. Dans le but de maximiser cette évaluation, on peut voir que plus un chromosome possède d'autres chromosomes voisins, meilleure est son évaluation, en cherchant à pénaliser les zones denses, l'exploration est ainsi favorisée. [Goldberg, Richardson 87] ont montré qu'une stabilisation est observée quand les valeurs des maximums $f(i)$ sont en proportion avec la taille des ensembles flous $m_i = \sum_{1 \leq k \leq n} sh [d(c_i, c_k) / \sigma]$ des niches.

f) Renouvellement des générations

A chaque étape ces opérateurs sont tirés et appliqués au hasard, on peut alors prendre les chromosomes et leurs images et éliminer les plus mauvais (ce qui fait par exemple deux suppressions pour quatre individus dans le cas du cross-over) ou alors trier toute la population et sa descendance et ne conserver que les meilleurs en assurant à P un cardinal constant.

La stratégie standard consiste à appliquer (suivant de très faibles probabilités) des mutations (plutôt, d'après ce qui précède, aux meilleurs individus issus du cross-over) et de garder une taille fixe pour la population, en supprimant les plus mauvais.

Les stratégies dites élitistes (évolution lamarckienne) consistent à trier immédiatement entre parents et enfants à chaque application d'un opérateur génétique, pour ne garder que les meilleurs. Mais on trouve également les stratégies étudiées depuis 1964 en Allemagne, suivant laquelle les opérateurs sont appliqués à chaque génération à chacun des μ individus et produisent λ enfants (le rapport λ / μ étant de l'ordre de 7).

Le choix est alors la stratégie $(\mu + \lambda)$ où μ parents produisent λ enfants, et le tout est trié à l'issue de la génération ou bien la stratégie (μ, λ) où les μ meilleurs de la descendance sont retenus, en ce cas, la meilleure performance n'est pas obligatoirement monotone comme pour les algorithmes standard [Bäck, Kursawe 94].

g) Arrêt de l'évolution

Il peut être décidé si la solution connue est obtenue à un seuil de précision fixé, ou bien au bout d'un nombre de générations limité ou encore après un nombre donné d'évaluations de la fonction. L'intérêt de la méthode est son applicabilité aux architectures parallèles.

EXEMPLE

Pour la fonction de Shaffer $f(x, y) = [1 - \sin^2 \rho] / [1 + 0,001 \rho^2]$ qui possède le maximum 1 pour (0, 0), (ρ est la distance à l'origine de (x, y)) il ne faut pas moins de 1500 évaluations avec un algorithme standard où $p_c = 0.6$ et $p_m = 0.001$, pour y accéder, compte tenu du grand nombre d'oscillations sur $[-100, 100]^2$.

REMARQUES Si $G = \{g_1, \dots, g_{ng}\}$ est l'ensemble des gènes sur lequel on définit une distance $d(g_j, g_i) = \delta_{ij}$ ou par $|i - j| / (ng - 1)$ au cas où on dispose d'un ordre canonique sur G. Si G_n est l'ensemble des chromosomes, donc de topologie discrète, on peut cependant définir une distance dans $[0, 1]$ par :

$d(C, C') = (\sum_{1 \leq i \leq n} d(g_i, g'_i)) / n$, on aura par exemple :

Pour une mutation (ou une recopie d'un gène à droite) $d(C, mut(C)) \leq 1 / n$

Pour une transposition (un échange de deux gènes) $d(C, transpo(C)) \leq 2 / n$

Pour un cross-over où k gènes sont échangés, si C_1 and C_2 sont les fils de C_1 et C_2 alors : $d(C_1, C_1) \leq k / n$ mais surtout : $\max [d(C_1, C_1), d(C_1, C_2), d(C_2, C_1), d(C_2, C_2)] \leq d(C_1, C_2)$ se qui s'exprime par le fait que la distance entre les parents détermine deux voisinages dont l'intersection contient les enfants.

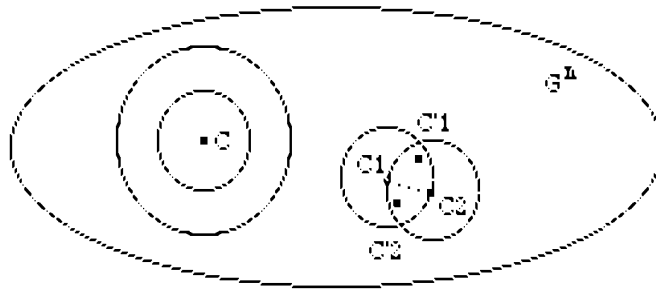


Figure 5.11 Dans l'espace métrique des chromosomes, visualisation de l'exploration par mutation, transposition et cross-over.

P ne doit pas être trop petit, ni la fonction d'évaluation trop restrictive sinon on risque de voir les solutions converger vers un optimum local.

La fréquence d'application de chacun des opérateurs doit être surveillé afin de d'utiliser les «bonnes» transitions entre générations. Il a été proposé de mesurer la diversité de la population, si $m = \text{card}(P)$, par l'indice $\partial(P) = (2 / m(m - 1)) \sum \sum_{i < j} d(C_i, C_j)$ (le cross-over ne la change pas) afin de trouver des opérateurs qui ne baissent pas trop cette diversité.

EXEMPLE, dans le problème classique du voyageur de commerce, une solution est une permutation de toutes les villes, ABCDEF par exemple s'il y en a 6. La fonction d'évaluation à minimiser est alors la somme des distances, mais les transitions devront être adaptées de façon à ce que les «solutions» restent toujours bijectives sur l'ensemble des villes. Les algorithmes génétiques sont appliqués à toutes sortes d'autres problèmes d'optimisation depuis la recherche du dessin d'un graphe minimisant le nombre d'arêtes se coupant jusqu'au problème de l'ordonnancement d'atelier où il faut minimiser la charge totale (temps utilisé par chaque machine où différentes pièces occupent différentes machines pour des durées différentes et suivant des contraintes de préséances) [Ghedjati 94].

LE THÉORÈME FONDAMENTAL

Pour un alphabet de k symboles (plus le symbole * utilisé pour signifier un gène quelconque) et des chromosomes de longueur l , il y a $(k + 1)^l$ schémas. Un schéma est une famille de chromosomes décrite par un mot de l'alphabet avec *, ainsi pour le mot 0^*1^* , il y a une unification possible avec les chromosomes 0010, 0011, 0110, 0111 sur l'alphabet $\{0, 1\}$.

Chaque chromosome choisi appartient à 2^l schémas.

Pour un schéma H , on note $o(H)$ le nombre de gènes définis, ainsi $o(10^*1^*) = 3$, et $d(H)$ la longueur entre les deux gènes définis extrêmes ainsi $d(10^*1^*) = 5$ et $d(1^*1^*) = 0$.

Soit f la fonction d'évaluation, (ici à maximiser) et f_m sa moyenne sur la population entière à chaque génération. Pour un schéma H , on notera $f(H)$ la moyenne de f sur la famille représentée par H dans la population.

Si à la génération t on note $m(H, t)$ le nombre de chromosome ayant le schéma H le théorème est alors :

$m(H, t + 1) \geq m(H, t) f(H) [1 - p_c d(H) / (l - 1) - o(H) p_m] / f_m$ où p_m est la probabilité d'une mutation, p_c celle du croisement. Il signifie que les schémas de courte longueur, d'ordre bas et de bonne valeur (les «building blocks») vont se reproduire exponentiellement.

On a pour une reproduction $m(H, t + 1) = m(H, t) f(H) / f_m$, en effet si C est un chromosome présentant le schéma H , la probabilité qu'il se reproduise est le quotient $f(C) / \sum f_i$ et $f(H) / f_m = n \sum \{f_i / C \in H\} / \text{card}(H) \cdot \sum f_i$ d'où la relation :

$$m(H, t) f(H) / f_m = n \sum \{f_i / C \in H\} / \sum f_i = n \text{proba}(H)$$

En notant $f(H) = (1 + c) f_m$, la relation s'écrit $m(H, t + 1) = (1 + c) m(H, t)$ soit plus précisément $m(H, t) = (1 + c)^t$ ce qui veut dire que si $c > 0$ le bon schéma croît exponentiellement et que si $c < 0$ le mauvais schéma décroît exponentiellement.

Pour l'action d'un croisement simple (échange des parties droite et gauche d'une coupure), la probabilité p_s de survivance du schéma H de longueur $d(H)$ est que la coupure survienne hors de la zone définie :

$$p_s = 1 - d(H) / (l(H) - 1) \geq 1 - p_c d(H) / (l(H) - 1)$$

Pour l'action d'une mutation, la probabilité de survivance de chaque gène est $1 - p_m$ et donc la probabilité de survivance du schéma H est $(1 - p_m)^{o(H)}$ approchée par la valeur $1 - o(H) p_m$ dans la mesure où $p_m \ll 1$, d'où l'inégalité annoncée.

NOUVELLES IDÉES MISES EN OEUVRE DANS LES STRATÉGIES D'ÉVOLUTION

Le problème essentiel des stratégies d'évolution est d'arriver à un compromis entre l'exploration de l'espace de recherche et l'exploitation des régions déjà explorées. C'est pourquoi, beaucoup d'idées implémentées, cherchent à éviter une trop grande homogénéité de la population. La population peut être de taille variable, les opérateurs peuvent être notés pour être triés à chaque génération, ces opérateurs peuvent évoluer au cours (mutation non uniforme), on peut créer des sous-populations mises en compétition et surtout créer un partage de la population en «niches» correspondant à des minima locaux.

Si le problème du codage est délicat, on avance souvent que l'exploration de l'espace est favorisée par un codage très raffiné comme un codage binaire, le problème de la définition de la performance est encore plus difficile. En effet, dans bien des questions pratiques, l'adéquation d'un individu à un objectif n'est pas clairement formalisable, le plus souvent, il s'agit d'une agrégation de plusieurs critères hétérogènes (performances physiques, prix, complexité du codage) dont les importances relatives sont sujettes à discussion. En vue d'optimiser plusieurs critères, il est possible de ne conserver au sein de la population que des individus incomparables entre eux au sens de Pareto, c'est à dire qu'il existe au moins deux critères pour lesquels deux à deux les individus sont classés de façon inverse [Fonseca, Fleming 93], [Gacogne 97].

La programmation génétique [Koza 92, 94] constitue un intéressant domaine d'expérimentation où chaque chromosome est un arbre (une fonction structurée en Lisp) dans le but de trouver des expressions de fonctions correspondant à un ensemble de couples (x, y) ou une courbe séparatrice de points par exemple.

Plusieurs travaux ont été réalisés en vue de déterminer une partie ou tout un contrôleur flou (les prédicats, les règles, certains paramètres numériques ...) [Geyer-Shulz 95], [Yuan Zhuang 96]. On pourra citer l'approche dite de Pittsburgh où la population est formée d'éventuelles solutions au problème, et celle de Michigan dans laquelle ce sont les règles qui sont individuellement notés pour leur contribution [Valenzuela, Rendon 91], [Cordon, Herrera 95]. C'est la première approche qui est présentée dans les exemples suivants.

APPLICATION AU CHOIX DES RÈGLES DE CONTRÔLEURS FLOUS POUR LE SUIVI D'UNE LIGNE

Un robot doit suivre une trajectoire en mesurant à chaque fois sa distance (entre 0 et une distance maximale de reconnaissance $d_m = 25$) à $\pm 45^\circ$ de sa direction. Les deux entrées sont donc cette distance x (négative à gauche, positive à droite) et la variation dx de cette distance entre deux prises de données (cette variation est qualifiée par les mêmes prédicats dans $[-a_m, a_m]$ avec un angle maximal de braquage $a_m = 30^\circ$). La sortie u suivant l'algorithme de Mamdani est un angle toujours qualifié par les mêmes prédicats mais dans $[-30^\circ, 30^\circ]$. Pour 5 prédicats, on veut trouver 25 termes linguistiques codant une table de règles identifiant ce contrôleur flou pour donner des conclusions aux règles. On utilisera un prédicat NUL (fonction d'appartenance nulle) pour les règles supprimées puisque NUL est neutre vis à vis du max.

Une «solution» sera donc ici une suite de 25 symboles, on peut prendre les diagonales successives du tableau des règles (numérotées sur la figure). Nous devons faire intervenir le nombre de virages effectués par le robot afin de minimiser les oscillations, mais si nous voulons obtenir un tracé le long de la courbe, ce nombre doit peser moins que la somme des distances.

En s'arrêtant sur l'heuristique attachant 3 fois plus d'importance à la somme des distances, on prend la définition :

Valeur $(C) = \sum |x| + (\sum |a_m * u|) / 3 - \text{vides}(C)$, (u est la sortie du contrôleur et a_m l'angle maximal de braquage) le dernier terme (nombre de cases vides) servant à classer les solutions suivant leur nombre de règles.

Il est assez fascinant de remarquer qu'un mécanisme d'évolution aveugle, uniquement guidé par une telle fonction de performance, peut orienter la population vers des solutions notablement différentes : très proches de la trajectoire mais oscillantes ou bien plus rapides mais s'en éloignant.

Le long d'une sinusoïde amortie, l'expérience montre que si cette valeur est calculée sur 30 pas (le premier virage de forte courbure) ou bien sur 150 pas (l'ensemble de la fenêtre) les populations ne convergent pas véritablement vers les mêmes tables et l'apprentissage sur un virage très serré ne donne pas nécessairement de bons résultats pour la suite.

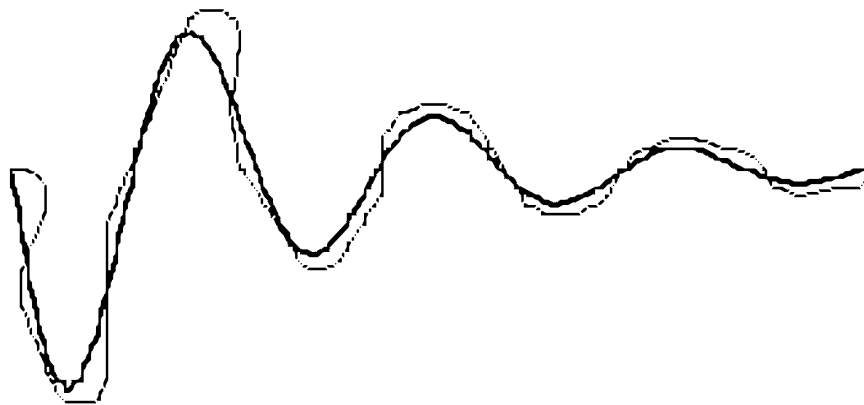


Figure 5.12 Sept règles obtenues codées par la chaîne (pb nul nul nul nul pb nul nul nul nul nul ze ze ze nul nul nul nul nul nb nul nul nul nul nb), la valeur est 2288.

La symétrie est imposée de sorte que seuls les 13 premiers symboles constituent réellement le chromosome, les meilleurs résultats comportent toujours PB dans les cases numérotées ci-dessous suivant un ordre cantorien, 1 et 6, et ZE au centre. Il semble au cours de l'apprentissage, que l'apparition d'un ZE (voire d'un NS ou NB dans d'autre sessions) en case 10 soit responsable d'une nette amélioration. En 7 et 8 plusieurs individus conservent PS ou PB et la différence se joue sur les cases 11 et 12 supportant ZE, NS ou NUL pour la première et NB, NS, ZE pour la seconde. Le prototype (on notera les effets conjugués des prédicats se disposant autour du centre ainsi que les couples de cases 6 et 10 et 16 et 20 dans les deux tables) est :

PB					25 NB
PS	10 ZE	14 PS	18 NB		
ZE	6 PB		13 ZE		20 NB
NS			8 PB	12 NS	16 ZE
NB	1 PB				
dx / x	NB	NS	ZE	PS	PB

APPLICATION À UN PROBLÈME DE POURSUITE

Ce problème est extrêmement intéressant à plus d'un titre, il s'agit pour le chat d'attraper la souris, et pour la souris, de lui échapper, s'il est assez facile d'imaginer des règles floues pour le chat, elles n'auront de valeur que si elles réussissent en toutes circonstances, mais s'il s'agit toujours de la même souris, ces règles seront trop particulières et n'auront pas d'effet face à une autre cible.

On voit là la difficulté qu'il y a à définir la fonction à optimiser pour le chat. D'autre part, quelles sont les paramètres sur lesquels le chat doit se fonder pour modifier sa trajectoire ?

Même à vitesse constante, ce qui est déjà une grande simplification du problème, la réponse n'est pas facile. Enfin, et surtout, il paraît très difficile de trouver des règles intuitives permettant à la souris d'échapper au chat.

Nous posons que le chat va m fois plus vite que la souris, mais que celle-ci peut virer m fois mieux. L'apprentissage a été effectué avec $m = 2$, un pas $ds = 8$ et un angle maximal $a_m = 60^\circ$. Le chat avance donc d'un pas ($m.ds$) et la souris peut tourner jusqu'à ($m.a_m$) par rapport à sa direction.

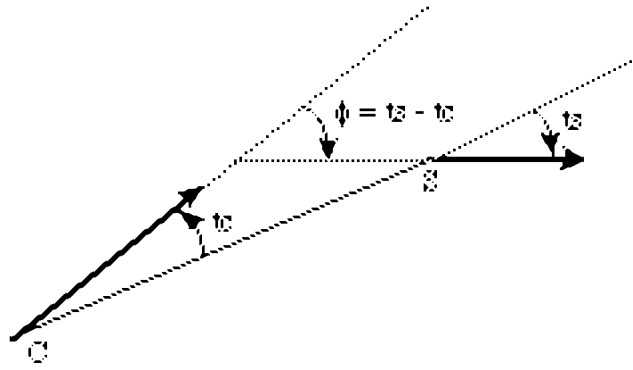


Figure 5.13 Les deux entrées t_c et t_s des contrôleurs flous.

Une première question est le choix des arguments à retenir pour exercer un contrôle, nous pouvons faire intervenir la distance CS , les angles t_c , t_s , $\phi = t_s - t_c$ et leurs variations. Des expériences préalables ont montré que les résultats pouvaient être bons en ne considérant que t_c et t_s .

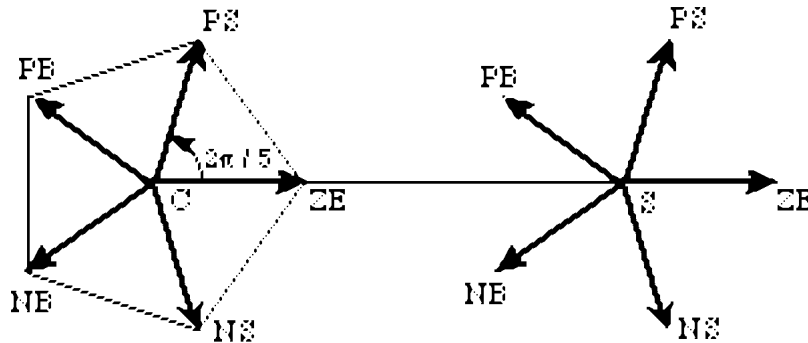


Figure 5.14 Les modes des cinq prédicats qualifiant les angles t_c et t_s .

Mais le problème de trouver les règles pour t_c et t_s n'est pas simple, car en prévoyant toutes les situations, ces deux angles appartiennent à $]-\pi, \pi]$ où π serait à la fois PB et NB. Nous nous ramenons donc aux cinq prédicats habituels par $t \rightarrow 5t / (4\pi)$ de telle façon que PB sera entièrement réalisé pour les angles de $4\pi / 5$ à π , et NB entre $-\pi$ et $-4\pi/5$. Les prédicats correspondent aux rayons de la figure 5.14.

VALEUR DU CHAT Pour évaluer un ensemble de règles de poursuite d'une cible mobile, nous avons sommé les 5 nombres de pas nécessaires pour atteindre une souris au comportement aléatoire lors de départs respectifs à une distance de $6.ds$ avec des valeurs initiales respectives $t_s = 0, -\pi/4, \pi/2, -3\pi/4$ et π , en majorant chaque poursuite à 15 pas. Nous ajoutons à ce total le nombre de règles et cherchons donc à minimiser cette valeur du chat. L'inconvénient majeur de cette définition est qu'il ne s'agit pas à proprement parler d'une fonction puisque deux évaluations du même individu ne donneront pas la même valeur et les différents chats de la population ne sont que grossièrement comparés lors de l'évolution. Parmi les règles intuitives du chat, on pourrait avoir par exemple : (t_c est PB) et (t_s est ZE) alors (d_{tc} est NB) ou encore la règle (t_c est PB) et (t_s est NS) alors (d_{tc} est PB), mais le risque d'agrégation nulle entre NB et PB montre que des cases voisines du tableau des règles ne doivent pas porter des valeurs linguistiques opposées. Une des meilleures solutions trouvées par algorithmes génétiques.

PB					
PS		14 PS	18 ZE		
ZE		9 PB		17 NS	
NS			8 ZE	12 NS	
NB					
ts / t_c	NB	NS	ZE	PS	PB

La vérification de ces résultats en faisant un cheminement aléatoire de la souris est excellente.



Figure 5.15 Départs $t_c = \pi$, $t_s = 0$.

VALEUR DE LA SOURIS II est difficile d'évaluer un comportement sur des situations initiales identiques, il faudrait pouvoir obtenir une moyenne sur un nombre très important d'épreuves, ce qui rend l'évaluation très longue, aussi avons-nous arrêté un compromis. Pour évaluer le comportement d'une souris, nous cherchons (avec les mêmes entrées t_c et t_s) à maximiser le total des pas effectués lors de 6 orientations initiales : la poursuite ($t_c = t_s = 0$), l'attaque frontale ($t_c = 0, t_s = \pi$), les départs opposés ($t_c = \pi, t_s = 0$), tête-bêche ($t_c = \pi/2, t_s = -\pi/2$) enfin ($t_c = \pi/2, t_s = \pi$) et ($t_c = 0, t_s = -\pi/2$).

Lors de cet apprentissage on voit se dessiner petit à petit une stratégie globale consistant à tourner au plus court et faire une boucle lorsque le chat est juste derrière puis à se maintenir derrière le chat dans son «angle mort». Ces boucles sont provoquées par des règles PB en position 10, 11 et 12 de la table.

En suivant cette stratégie, la souris retrouve un résultat élémentaire : en tournant toujours dans le même sens mais en augmentant sa courbure, la souris arrive quasiment à décrire un cercle, le chat décrivant «devant» elle un cercle de rayon m fois plus grand, les deux mobiles ayant même vitesse angulaire arrivent à des mouvements quasi-périodiques.

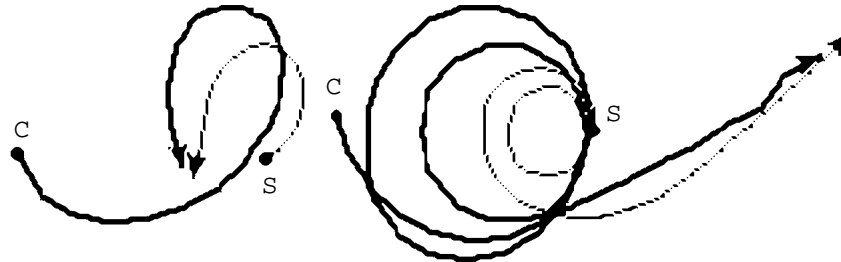


Figure 5.16 Deux exemples de départs $t_c = -\pi/2, t_s = 0$.

L'observation plus attentive des blocs performants comme PS NB PS en positions 7, 8, 9 ou seulement NB en 8 comme la suite l'a montré, permet d'énoncer la règle :

Si le chat va dans la direction de la souris (t_c est ZE) et que celle-ci a une direction légèrement négative (t_s est NS), alors elle doit la rendre plus négative encore (NB en 8), ce qui signifie virer brusquement devant le chat.

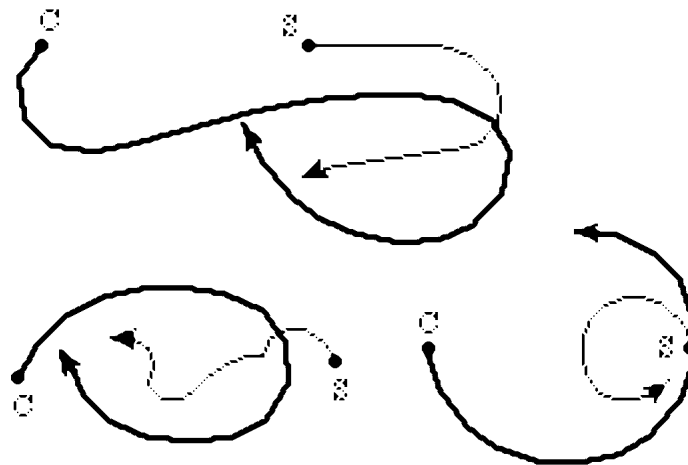


Figure 5.17 L'observation des trajectoires en cours d'apprentissage révèle aussi cette stratégie que l'on pourrait décrire par : obliger le chat à tourner, et celui-ci ne pouvant réduire sa vitesse, est contraint de dépasser la souris, à ce moment, celle-ci oblique pour passer derrière lui, ou plutôt se maintient à environ 135° de la direction du chat en l'obligeant à poursuivre son virage.

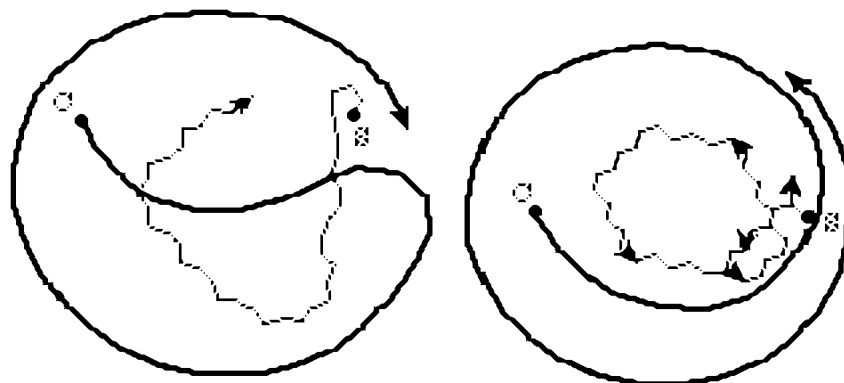


Figure 5.18 Grâce à NB en 8 et NS en 4 qui, moins indispensable, mais présent dans toutes les meilleures tables, on arrive à une trajectoire en ligne brisée qui oblige le chat à tourner en le contraignant de surcroît à décrire un large cercle. Le chat ne sait, en effet, si la souris va se maintenir en gros à 135° de sa direction, ce qu'elle fait une fois sur deux, ou bien passer derrière lui. Ici, NB est remplacé par NS en 8.

Les règles de la souris ne sont pas véritablement toujours efficaces, notamment au cas où $t_c = t_s = 0$ (la case centrale) la souris devrait obliquer, mais ceci contrarie les règles 8 et 18 qui, elles, sont indispensables. Les règles 4 et 22 le sont un peu moins, quant aux règles 10 et 16, elles peuvent être omises ou déplacées en 11 et 15. Le prototype est donc réduit à deux règles (8 et 18) avec éventuellement 2 ou 4 autres règles :

PB			22 PS		
PS	10 PB		18 PB		
ZE					
NS			8 NB		16 NB
NB			4 NS		
ts / tc	NB	NS	ZE	PS	PB

La robustesse de ces règles est néanmoins surprenante, si on modifie l'angle a_m (par exemple 30°) et surtout le rapport m , en ne conservant que les deux règles 8 et 18, on peut observer que la souris arrive véritablement à se placer derrière le chat pour $m = 1$ et à lui faire faire de larges boucles pour $m = 1.2$.

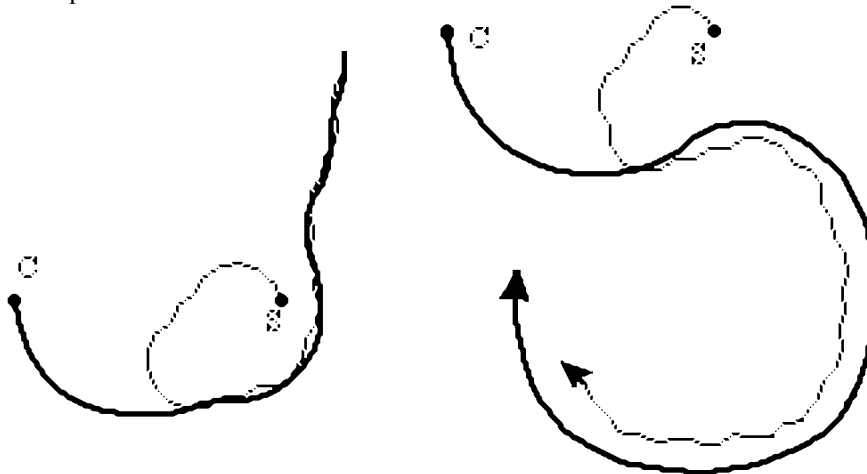


Figure 5.19 Succès ou échec du chat dans des conditions très voisines.

APPLICATION À LA RECHERCHE DES PRÉDICATS ET DES RÈGLES, EXEMPLE DU PARCOURS D'OBSTACLES (2 ENTRÉES, 2 SORTIES)

Dans [Glorennec 92], on part d'une famille de prédicats sur $[-1, 1]$ se coupant à 0,5 et définis par des sigmoïdes $f(x) = 1 / (1 + \exp(-ax+b))$, pour deux entrées et 5 prédicats on peut n'avoir que 10 valeurs continues à régler. Une solution sera une suite de 10 réels qui peut être mutée par modifications d'un nombre aléatoire d'éléments ou par addition d'un «bruit» gaussien.

Dans l'application ci-dessous, nous avons toujours 5 prédicats qui sont les troncatures dans $[0, 1]$ des triangles de hauteur h et de demi-base r .

Les éventuelles solutions sont codées de manière structurées comme $(r \ h \ R_1 \ R_2 \ R_3 \dots)$ où les règles R_i sont maintenant en nombre variable et de la forme $(p \ (\partial \text{dir} \ \partial \text{pas}) \ d \ a)$. Il y a deux entrées d et a qui sont la position polaire du plus proche obstacle vu dans le secteur limité par la distance d_m et les angles $\pm a_m$, et deux sorties ∂dir (variation de direction appréciée dans $[-a_m, a_m]$) et $\partial \text{pas} \in [-acc, acc]$ la variation du pas tel que celui-ci soit toujours entre une vitesse minimale avec d_s et une vitesse maximale p_m .

La priorité d'une règle signifie que les règles particulières de priorité 0 s'appliquent pour conclure à un couple de conclusions, et si elles ne peuvent s'appliquer, des règles plus générales (de priorité 1) s'appliquent alors.

Les opérateurs génétiques utilisés pour cette expérience sont les bruits gaussiens apportés à r et h , les mutations dans une règle, la création d'une nouvelle règle aléatoire, la suppression d'une règle, les changements de priorité et le cross-over.

La fonction à minimiser est $(nb \ de \ pa. (p_m - \sum \text{pas}) + nb \ de \ symboles + \text{tenue de route})$

Exemple de solution obtenue par algorithme évolutif :

$ds = 3$
 $pm = 15$
 $am = 45$
 $acc = 5$
 $dm = 35$

 Nb. symboles 8
 $r = 0.95$
 $h = 1.85$

 Règles :
 (0 0 (nb ns) ps pb)
 (0 0 (pb ns) ps nb)
 (1 0 (ns pb) ps ze)
 (1 0 (ps pb) ps ze)

 Reste distance 63
 Tenue de route 41
 (100 pas)
 Valeur 61

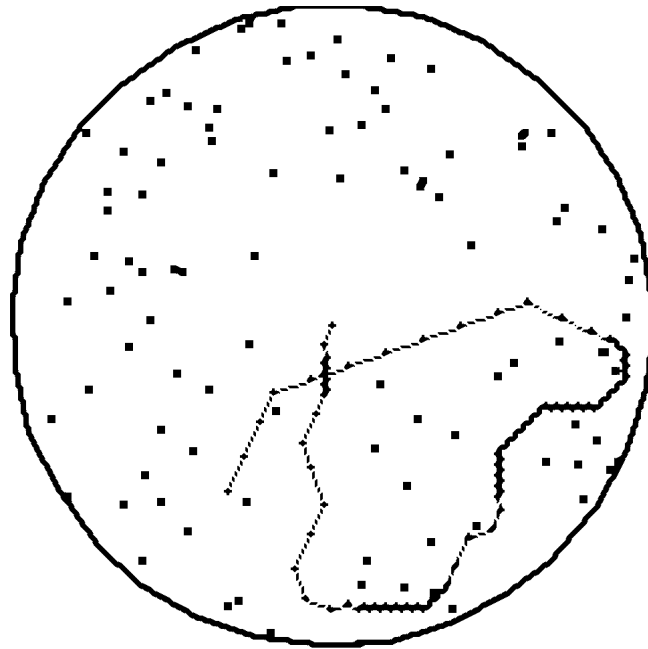


Figure 5.20 Exemple du parcours d'obstacles.

APPLICATION À LA RECHERCHE COMPLÈTE D'UN CONTRÔLEUR FLOU POUR UN ROBOT DEVANT FRANCHIR DES PORTES DE SLALOM

On souhaite maintenant la mise au point de contrôleurs flous utilisant des prédicats relativement variables mais définis par un petit nombre de paramètres. Il est assez difficile de considérer le problème dans son entière généralité. La plupart du temps de fortes restrictions permettent d'obtenir des résultats acceptables mais la recherche ne s'appliquant que sur un aspect du problème (les règles ou une famille de prédicats bien définie). Nous avons cherché ici à réaliser un compromis entre une famille générale et un petit nombre de paramètres. Le quadruplet (nb, contr, trap, fuz) défini ci-dessous permet d'obtenir des familles très diverses recoupant un grand nombre de familles de prédicats utilisées couramment dans les applications concrètes. Nous définissons ces familles à partir de fonctions d'appartenance sur l'intervalle $[-1, 1]$ grâce à :

Le nombre $p \in \{1, 2, 3, 4\}$ permet de déduire $2p + 1$ prédicats.

Le second $r \in [0, 1]$ nommé «contraction» des prédicats, il mesure la manière dont ils sont resserrés autour de 0.

Le coefficient suivant $h \in [0, 1]$ est un indice («trap») de chevauchement, et $c \in]0, 1]$ («fuz») est une mesure du flou grâce à la «pente» des trapèzes.

Soit m_k le «sommet» d'une fonction triangulaire pour $0 \leq k \leq p$ (avec $m_{-1} = m_1$) définie par :

$$m_k = r \left(\frac{k}{p}\right)^2 + (1 - r) \frac{k}{p} \quad \text{et :}$$

$$\mu_k(x) = \min \left(1, \max \left(0, h + 1 - \frac{1}{c} \left| \frac{m_k - x}{m_k - m_{k-1}} \right| \right) \right)$$

Chaque fonction d'appartenance est la troncature d'une «fonction triangulaire» dont le maximum est $1 + h + r \cdot m_k$ pour m_k , (ainsi $m_0 = 0$) et qui est linéaire entre ce sommet et le point $(m_{k-1}, 0)$, symétrique autour de m_k et bornée par $[0, 1]$. On nomme ZE l'ensemble flou de fonction d'appartenance μ_0 et on impose une symétrie telle que $\mu_{-k}(x) = \mu_k(-x)$. On peut alors appeler les prédicats PS et NS pour $k = 1$ ou -1 dont les fonctions sont respectivement μ_1 and μ_{-1} , PM and NM pour μ_2 and μ_{-2} et PB, NB pour ± 3 . Les figures suivantes montrent quelques possibilités de familles sur $[0, 1]$.

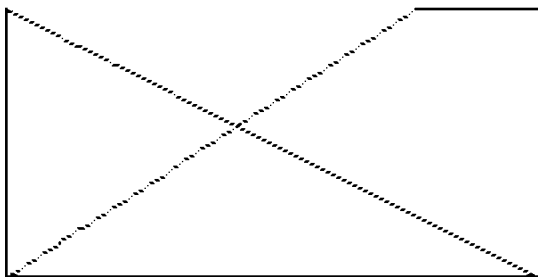


Figure 5.21 Famille de predicats sur $[0, 1]$ avec $nb = 1$, $contr = 0.3$, $trap = 0$ et $fuz = 1$.

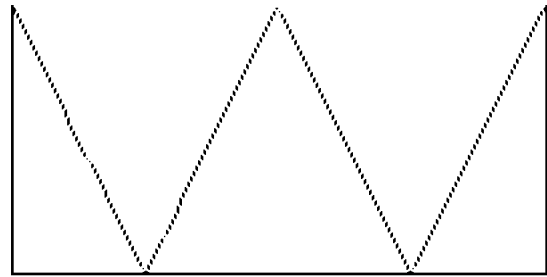


Figure 5.22 $nb = 2$, $contr = 0$, $trap = 0$ et $fuz = 0.5$

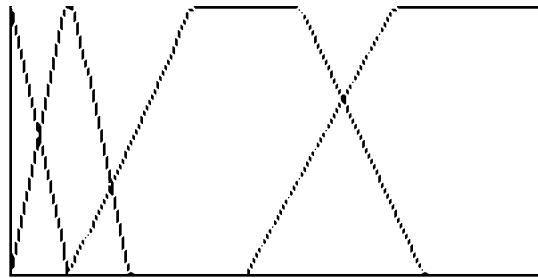


Figure 5.23 nb = 3, contr = 1, trap = 0 et fuz = 1.

CODAGE D'UN CHROMOSOME Les chromosomes n'ont pas une longueur fixe, leur structure est (nb contr trap fuz ouv R₁ R₂ R₃ ...) où nb, contr, trap, fuz sont les coefficients décrits plus haut, «ouv» un angle en degrés, et R_i les règles, toutes de la forme (pr f h₁ ... h_{nh} c₁ c₂ ... c_{nc}).

Les hypothèses h_i sont des symboles ∈ {any, nb, ns, ze, ps, pb} et les conclusions c_j sont des entiers [-100, 100], pr est la priorité de la règle, et f est sa force. Cette force est simplement le niveau d'utilisation de la règle au cours d'une épreuve, elle n'est présente que pour l'utilisation de deux opérateurs, la mutation sur la règle la plus utilisée et la suppression de la règle la moins utilisée.

OPÉRATEURS GÉNÉTIQUES Ce sont, outre les deux derniers mentionnés, le bruit sur les paramètres numériques, la création d'une nouvelle règle ou d'un nouvel individu aléatoire (migration) et la suppression d'une règle. Ces opérateurs sont notés suivant leur capacité cumulée à améliorer les 4 critères définis plus bas.

A chaque génération, la population est classée et les opérateurs le sont également, chaque opérateur est ensuite appliqué à l'individu de même rang de sorte que les meilleurs opérateurs s'essaient sur les meilleurs individus. D'autre part il n'y a pas reproduction, mais les individus les moins bons au sens de Pareto pour les 4 critères sont éliminés? C'est pourquoi la population peut être réduite à quelques individus incomparables deux à deux.

EPREUVE RÉALISÉE Il s'agit de faire suivre un parcours matérialisé par des portes, le robot M devant effectuer un slalom en mesurant à chaque étape, les angles qui séparent sa direction des visées des deux côtés P₁, P₂ de la prochaine porte, en mesurant de plus la distance d qui le sépare du milieu de cette porte, il va considérer les deux entrées constituées par a₁+a₂ et d.

Plus précisément, nous allons chercher des contrôleurs flous de [-1, 1]*[0, 1] vers le carré [-1, 1]². Une règle ayant deux prémisses pour a₁+a₂ apprécié dans [-ouv, ouv], et d apprécié dans [0, d_m] avec d_m = 40 et deux conclusions dans [-100, 100] donnant les pourcentages de variations de direction (le maximum est a_m = 60°) et de vitesse (maximum acc = 10 pixels) à effectuer.

Le pas variable restant dans l'intervalle [4, 35].

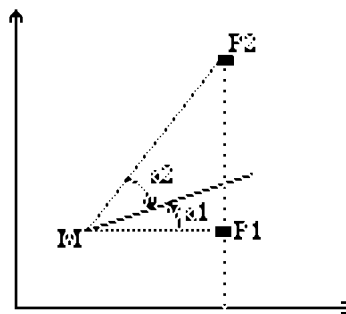


Figure 5.24 Le mobile M mesure les deux angles et sa distance au centre de la prochaine porte à passer, le but qui lui est assigné par les règles serait a₁+a₂ = 0.

CRITÈRES À OPTIMISER

Nous allons orienter notre recherche avec les quatre critères suivant à minimiser :

- 1 - Le nombre «np» de portes restantes à l'issue de 35 pas du robot.
- 2 - Le second critère est un malus attribué aux portes non franchies, c'est la moyenne «val» des distances mesurées au passage de chaque porte si celles-ci sont passées à l'extérieur.
- 3 - Un autre critère est la «tenue de route» [Gacogne 94]. C'est le nombre en pourcentage :

$$tr = \left| \frac{lda}{am} + \frac{dp}{2acc} - \frac{1}{2} \right| \in [0, 1]$$

Dans lequel da et dp sont les changements de direction et de pas réalisés à chaque prise de données, «am» et «acc» étant les maxima de ces changements. Plus tr est proche de 0, meilleure nous estimons la trajectoire (accélération en ligne droite, freinage dans les virages), lorsqu'il est proche 50, nous l'estimons mauvaise, et plus tr est proche de 100, plus la trajectoire est à l'opposé d'une bonne conduite.

- 4 - Nous tenons enfin compte du nombre «ns» de symboles significatifs dans une liste de règles, afin de favoriser les petits nombres de règles.

EXPERIENCE D'EVOLUTION

Afin d'explorer continuellement l'espace des solutions, nous avons choisi une taille minimale de 5 individus. Ainsi avec 2 ou 3 critères, il est fréquent que l'élimination des solutions comparables les plus mauvaises amène à 3 solutions retenues à l'issue d'une génération. En regard de ces 4 critères mentionnés, la taille de la population oscille entre une et deux dizaines, c'est pourquoi, nous avons choisi une taille maximale de 25.

Nous avons fait 20 évolutions, chaque expérience étant limitée à 5000 évaluations (5000 parcours du slalom). Ces expériences ont été faite pour moitié en triant la population suivant l'ordre lexicographique sur (np val tr ns), et pour moitié sur (val np tr ns) attendu que les deux premiers critères sont manifestement les plus importants.

A l'issue de ces évolution nous observons la variété des solutions trouvées et faisons concourir ensemble les meilleurs individus.

CONCLUSIONS DE L'EXPERIENCE

L'observation des opérateurs lors des différentes sessions montre l'importance de l'opérateur de «migration» qui permet, surtout en début d'évolution de ne pas converger trop rapidement vers une solution locale. Il est néanmoins évident que ce sont des évolutions séparées suivies de confrontations qui permettent d'obtenir de bons résultats. Ce fait a été remarqué par de nombreux chercheurs qui ont cherché à simuler des «niches écologiques» [Goldberg, Richardson 87]. L'action du «cross-over» doit être freiné en début de parcours par un grand nombre d'autres opérateurs, et on remarque en effet que le croisement n'opère bien qu'en fin de parcours. On peut remarquer que si on regroupe les opérateurs les plus perturbants (le plus liés à une exploration aléatoire uniforme), ceux-ci sont largement classés en tête.

En ce qui concerne les individus trouvés, on peut remarquer que presque tous réalisent des contrôleurs flous basés sur des familles de 5 prédicats (un seul est à 3 prédicats, 4 le sont des avec 7 prédicats) se chevauchant fortement (contr = 0.43 et trap = 0.45 en moyenne) et avec des trapèzes peu spécifiques (fuz = 0.77 en moyenne) c'est à dire assez larges. Ces contrôleurs ont peu de règles (généralement deux et leurs symétriques la plupart du temps ayant le même niveau de priorité) et l'univers d'appréciation des angles est donné par «ouv» de 25° à 94° (moyenne 60°).

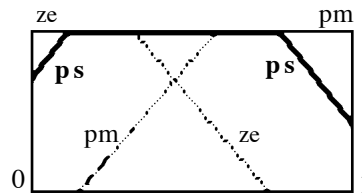


Figure 5.25 Prédicats ZE, PS, PM se chevauchant fortement, obtenus sur [0, 1] par algorithme génétique.

Un des meilleurs résultats (3 portes ratées de peu). Les prédicats NM, NS, ZE, PS, PM signifient respectivement «negative medium» et «small», zero, «positive small» et «medium». Les constantes sont ds = 4, pm = 35, am = 60, acc = 10 et dm = 40. Le chromosome trouvé a les caractéristiques nb = 2, contr = 0.21, trap = 0.76, fuz = 0.95. Les angles sont appréciés dans [-86°, 86°] et les règles toutes de même priorité sont :

$$a_1 + a_2 \text{ ze \& dist ps} \rightarrow da = -53 \ \& \ dp = 75$$

$$a_1 + a_2 \text{ ze \& dist ps} \rightarrow da = 53 \ \& \ dp = 75$$

$$a_1 + a_2 \text{ pm \& dist pm} \rightarrow da = -86 \ \& \ dp = -22$$

$$a_1 + a_2 \text{ nm \& dist pm} \rightarrow da = 86 \ \& \ dp = -22$$

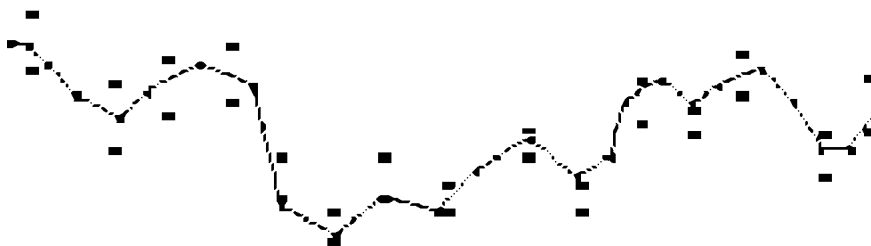


Figure 5.26 Robot traversant des portes de slalom (toutes les portes sont atteintes, mais malus de 5 pour celle qui est ratée et tenue de route 17).

Conclusion

En reprenant les différents concepts énoncés lors de l'introduction, on peut s'apercevoir que la plupart sont liés à des aspects linguistiques, certains ayant reçu à l'heure actuelle, un début de formalisation, d'autres non.

Si les concepts théoriques d'incertitude et d'imprécision ne reçoivent pas chacun une interprétation unique, on peut dire que plusieurs théories cohérentes ont vu le jour depuis quelques décennies.

En ce qui concerne les prédicats vagues, le concept simple de sous-ensemble flou est largement employé. Par contre, on trouve peu de travaux portant sur des applications concrètes utilisant des modificateurs de prédicats ou des quantificateurs flous, bien que l'idée soit présente en permanence. En tous cas, si tous ces termes évoquent le traitement du langage naturel, jusqu'à présent, ce n'est pas l'informatique linguistique qui a bénéficié de ces travaux, mais des aspects beaucoup plus proches du monde industriel.

Les applications actuelles sont essentiellement la régulation industrielle par commande floue, partout où la construction d'un modèle mathématique s'avère difficile. Les premiers contrôleurs flous industriels opérationnels ont été vendus en 1987. Leurs points forts sont maintenant connus en ce qui concerne la mise au point, certes souvent empirique, mais qui a le précieux avantage de garder une transparence, leur rapidité et leur robustesse. Dans ce domaine, la logique floue continuera certainement sur sa lancée pour toutes les tâches de surveillance.

Le principe de la commande floue n'est d'ailleurs pas réservé à la régulation d'un paramètre sur une valeur cible, mais commence à être appliqué à la modélisation de fonctions non directement exprimables, comme un comportement animal ou humain, individuel ou social.

Mais ces travaux issus du principe simple du contrôleur flou posent encore le problème de l'apprentissage des règles floues et de leur validation bien que de gros progrès aient été faits cette dernière décennie.

La seconde application est la reconnaissance des formes avec la classification floue, notamment en traitement d'images. Ce domaine qui est celui de la reconnaissance des formes en général, reconnaissance vocale, de l'écriture manuscrite, des éléments constituant une photo-satellite, d'un visage etc, fait l'objet de beaucoup de travaux de plus en plus liés au flou.

Quel est l'avenir de la logique floue et quels autres domaines pourra-t-elle pénétrer ? Outre l'informatique linguistique déjà évoquée où on ne peut parler que du long terme, on peut penser à l'intelligence artificielle distribuée où de petits automates cellulaires apprennent à coopérer en vue de la résolution collective d'un problème.

Annexe 1

Rappels de logique classique

Nous présentons en annexes de brefs rappels sur les résultats de la logique classique suivis de la présentation d'un certain nombre de voies explorées dans le but, soit d'étendre cette logique, soit de rendre compte de notions telles que la constructibilité (l'existence d'une solution à un problème n'est assurée que par celle d'un procédé effectif pour sa construction), la connaissance, la compatibilité avec les connaissances actuelles, la vérité dans le passé ou le futur...

Enfin, nous présentons dans les annexes suivantes, deux langages de programmation, Fril inspiré de Prolog raisonne par induction avec des couples (crédibilité, plausibilité), et Mvl également fondé sur Prolog, mais laissant le choix entre plusieurs logiques ; classique, avec défauts ou temporelle.

Pour leur part, les réseaux bayésiens constituent un système rigoureux quoiqu'assez lourd pour formaliser des relations causales.

La logique classique formalise la notion de preuve grâce à une axiomatique précise à l'aide d'axiomes et de règles (aspect syntaxique). Une preuve est aussi appelée une dérivation, c'est à dire une suite de formules dont chacune est soit un axiome soit le résultat de l'application d'une règle sur des axiomes ou des formules antérieures de cette suite.

D'un autre côté, une sémantique est définie par des tables de vérité dans une théorie des "modèles". On a alors le «théorème de complétude» tant pour le calcul propositionnel que pour le calcul des prédicats qui s'exprime par :

F est valide (vrai dans toute réalisation, ce qui s'écrit : $\models F$)

\Leftrightarrow F est un théorème

(prouvable d'après un système d'axiomes et de règles, noté par $\vdash F$)

Un troisième aspect (algébrique) est étudié avec les algèbres de Boole et Lindenbaum- Tarski. Nous passerons brièvement en revue les grands axes et résultats en logique et calculabilité dans ce qui suit.

A.1.1. Langage des propositions

Les formules bien formées de la logique propositionnelle du premier ordre sont les mots formés à partir d'un alphabet comportant un ensemble infini V de «variables propositionnelles» (les majuscules A, B,... qui suivent) et des signes dits connecteurs $\{\neg, \vee\}$ de telle sorte que l'on définisse :

Les formules atomiques sont les éléments de V, puis, si F et F' sont des formules $\neg F$ en est une, $F \vee F'$ également.

On définit la conjonction comme abréviation par : $A \wedge B = \neg(\neg A \vee \neg B)$, l'implication $(A \rightarrow B) = \neg A \vee B$, l'équivalence $A \leftrightarrow B = (A \rightarrow B) \wedge (B \rightarrow A)$ ainsi que l'exclusion réciproque «ou bien» notée \oplus ici par $A \oplus B = (A \wedge \neg B) \vee (\neg A \wedge B)$, et le symbole de Sheffer noté par une barre | avec $A | B = \neg(A \wedge B)$.

ASPECT SÉMANTIQUE

Une assignation α est une application (dite valeur de vérité) de l'ensemble des variables propositionnelles V vers $\{0, 1\}$ appelés «faux» et «vrai», assignation qui s'étend à toutes les formules.

Cette extension est définie par les fameuses tables de vérité à savoir

$\alpha(\neg P) = 1 - \alpha(P)$ pour la négation et :

P	Q	$P \wedge Q$	$P \vee Q$	$P \rightarrow Q$	$P \leftrightarrow Q$	$P \leftarrow Q$	$P \oplus Q$	$P Q$
1	1	1	1	1	1	1	0	0
1	0	0	1	0	0	1	1	1
0	1	0	1	1	0	0	1	1
0	0	0	0	1	1	1	0	1

Les tautologies, sont les formules vraies pour toute assignation (notation $\models F$).

Les premières tautologies du calcul propositionnel sont les 18 suivantes :

Involutivité de la négation $\neg(\neg P) \Leftrightarrow P$

Les lois de Morgan, $\neg(P \wedge Q) \Leftrightarrow (\neg P \vee \neg Q)$ et $\neg(P \vee Q) \Leftrightarrow \neg P \wedge \neg Q$

Les connecteurs \wedge et \vee sont commutatifs :

$$P \wedge Q \Leftrightarrow Q \wedge P \quad \text{et} \quad P \vee Q \Leftrightarrow Q \vee P$$

Idempotents :

$$P \wedge P \Leftrightarrow P \quad P \vee P \Leftrightarrow P$$

Associatifs :

$$P \wedge (Q \wedge R) \Leftrightarrow (P \wedge Q) \wedge R \quad \text{et} \quad P \vee (Q \vee R) \Leftrightarrow (P \vee Q) \vee R$$

Distributivités mutuelles :

$$P \wedge (Q \vee R) \Leftrightarrow (P \wedge Q) \vee (P \wedge R) \quad \text{et} \quad P \vee (Q \wedge R) \Leftrightarrow (P \vee Q) \wedge (P \vee R)$$

Eléments neutre et absorbant : 0 et 1 désignant respectivement le faux et le vrai, P étant une proposition quelconque :

$$0 \wedge P \Leftrightarrow 0 \quad 0 \vee P \Leftrightarrow P \quad 0 \text{ est absorbant pour } \wedge, \text{ et neutre pour } \vee.$$

$$1 \wedge P \Leftrightarrow P \quad 1 \vee P \Leftrightarrow 1 \quad 1 \text{ est neutre pour } \wedge, \text{ et absorbant pour } \vee.$$

Enfin les deux propriétés $P \wedge \neg P \Leftrightarrow 0$ et $P \vee \neg P \Leftrightarrow 1$ font du calcul propositionnel une algèbre de Boole.

Notons encore la contraposition $(\neg Q \rightarrow \neg P) \Leftrightarrow (P \rightarrow Q)$

ASPECT SYNTAXIQUE

Il existe une bonne douzaine d'axiomatics équivalentes du calcul des propositions, signalons la plus courte, celle de Nicod avec un seul symbole, la barre de Sheffer |, (le «nand» bien connu en informatique), un seul axiome :

$$(A | (B | C)) | ((D | (D | D)) | ((E | B) | ((A | E) | (A | E))))$$

et une seule règle, le fameux modus-ponens (si A et $A \rightarrow B$, sont connus alors B peut l'être) qui s'énonce dans ce langage :

$$A, A | (B | B) \vdash B.$$

La négation y est définie par $\neg A = A | A$.

Cependant, la syntaxe la plus courante est celle de Hilbert définie sur l'alphabet $V \cup \{\neg, \vee, \wedge, \rightarrow\}$ où V est un ensemble dénombrable de «variables» (ici les majuscules) avec les 10 axiomes :

$$A \rightarrow (B \rightarrow A) \quad (A \rightarrow B) \rightarrow (((A \rightarrow (B \rightarrow C)) \rightarrow (A \rightarrow C))$$

$$A \rightarrow (B \rightarrow (A \wedge B)) \quad A \wedge B \rightarrow A \quad A \wedge B \rightarrow B$$

$$A \rightarrow A \vee B \quad B \rightarrow A \vee B \quad (A \rightarrow B) \rightarrow ((A \rightarrow \neg B) \rightarrow \neg A)$$

$$(A \rightarrow C) \rightarrow ((B \rightarrow C) \rightarrow (A \vee B \rightarrow C))$$

Enfin l'axiome $\neg\neg A \rightarrow A$ (modifié dans l'intuitionnisme) et la règle unique du «modus-ponens» $A, A \rightarrow B \vdash B$.

A.1.2. Langage des predicats

SÉMANTIQUE

Un langage de prédicats est construit sur un alphabet comportant un ensemble dénombrable de variables V, des symboles relationnels (dont éventuellement le symbole d'égalité =), des symboles fonctionnels à 0 places (constantes) 1, 2 ... places, les connecteurs $\{\neg, \vee\}$ et le quantificateur \exists (il existe au moins un).

TERMES Les constantes et variables sont les termes les plus simples. La juxtaposition d'un symbole fonctionnel et de termes séparés (en nombre égal au nombre de places du symbole de fonction), constitue un nouveau terme, c'est-à-dire une expression pouvant être interprétée dans la mesure où les variables le sont.

FORMULES OU PROPOSITIONS

Les propositions «atomiques» sont le «vrai», le «faux», et tous les mots écrits avec un symbole de relation (la plupart du temps binaire) et des termes appartenant aux types sur lesquels portent la relation. Les propositions structurées sont construites à partir des propositions atomiques en les assemblant grâce aux connecteurs logiques comme pour le calcul propositionnel et aux quantificateurs (si F est une formule, $\exists x F$, en est une).

\forall (quelquesoit) est défini de façon duale par $\forall x P(x) = \neg \exists x \neg P(x)$.

PROPOSITION CLOSE

Les propositions closes ou «sentences» sont les propositions où toutes les variables y figurant sont «muettes», ou «liées» par un quantificateur ce qui signifie que le sens de la proposition ne change pas si on les remplace par d'autres variables.

ASPECT SÉMANTIQUE

Une réalisation du langage est une application ρ dans une structure M telle qu'à tout symbole fonctionnel est associé une fonction de M dans M avec autant de places (en particulier aux constantes sont associées des éléments de M) et à tout symbole relationnel est associé une relation dans M avec autant de places (en particulier, = est réalisé comme l'égalité). Une assignation α est une application de V dans l'ensemble M, elle se prolonge donc naturellement à tous les termes et par définition elle satisfait une formule si :

A atomique de la forme $R t_1 t_2 \dots t_n$:

$$(M, \alpha) \models A \quad \Leftrightarrow \rho(R)(\alpha(t_1), \dots, \alpha(t_n)) \text{ est vrai.}$$

Sinon :

80 **Logique floue**

$(M, \alpha) \models \neg A \Leftrightarrow \text{non } (M, \alpha) \models A$
 $(M, \alpha) \models A \vee B \Leftrightarrow (M, \alpha) \models A \text{ ou } (M, \alpha) \models B$
 $(M, \alpha) \models \exists x A \Leftrightarrow \text{Il existe un } c \text{ dans } M \text{ tel que } (M, \alpha(x \leftarrow c)) \models A \text{ où } \alpha(x \leftarrow c) \text{ désigne l'assignation donnant la valeur } c \text{ à } x \text{ et coïncidant avec } \alpha \text{ pour les autres variables. On aura donc :}$
 $(M, \alpha) \models \forall x A \Leftrightarrow \text{Pour tout élément } c \text{ de } M, (M, \alpha(x \leftarrow c)) \models A$

M est dit modèle de la formule close (notation $M \models F$) si F vraie dans M pour toute assignation, et F est une tautologie valide si elle est vraie dans toute réalisation.

HIÉRARCHIE DES FORMULES

On pose $\Sigma_0 = \prod_0$ l'ensemble des formules sans quantificateurs (closes ou non) puis $\Sigma_{p+1} = \exists \prod_p$ est défini comme l'ensemble des formules qui (sous forme préfixe, les quantificateurs en premier) s'expriment comme une suite de quantificateurs débutant par \exists et ayant p alternances sur le type de quantificateur, suivie d'une formule sans quantificateurs. De même $\prod_{p+1} = \forall \Sigma_p$ et $\Delta_p = \Sigma_p \cap \prod_p$.

ASPECT SYNTAXIQUE

La syntaxe du «calcul des prédicats» de Hilbert est celle du calcul propositionnel où sont ajoutées deux règles de particularisation $[A(x) \rightarrow C] \vdash [\exists x A(x) \rightarrow C]$ et de généralisation $[C \rightarrow A(x)] \vdash [C \rightarrow \forall x A(x)]$ si x non variable libre dans C.

A.1.3. Arithmétique

Une des premières axiomatiques ayant servi à la fondation des mathématiques est celle de Peano (1899) pour l'arithmétique, l'alphabet des symboles utilisés est encore un ensemble dénombrable (au sens naïf) de variables V et les symboles $\{\neg, \vee, \exists, 0, S, +, *, =\}$ où S va se réaliser comme la fonction successeur. Les connecteurs \wedge et \rightarrow n'étant que des abréviations.

Les axiomes et règles sont ceux de la logique des prédicats, plus les axiomes de l'égalité et les 6 axiomes dits de Peano :

$$\forall x \forall y \quad Sx = Sy \rightarrow x = y \quad x + 0 = x \quad \neg(Sx = 0)$$

$$x + Sy = S(x + y) \quad x * 0 = 0 \quad x * Sy = (x * y) + x$$

Ainsi que le schéma d'axiome de récurrence (ce n'est pas un axiome, mais une famille infinie d'axiomes), pour toute formule F à une variable libre :

$$[F(0) \wedge \forall x F(x) \rightarrow F(Sx)] \rightarrow \forall y F(y)$$

On montre que l'arithmétique n'est pas finiment axiomatisable. L'arithmétique du second ordre, elle, considère deux types d'objets (donc deux ensembles de variables), en ce cas, on peut trouver un nombre fini d'axiomes.

Le résultat logique le plus important est que la consistance (ou non-contradiction) de la théorie de l'arithmétique est exprimable dans cette théorie par une formule, mais celle-ci ne peut être ni démontrée, ni réfutée. Cet énoncé de non-contradiction est donc indécidable et montre «l'incomplétude» de l'arithmétique. C'est le célèbre théorème de Gödel (1940) établi grâce à une numérotation de toutes les formules et démonstrations.

Le théorème de Tarski (1944) montre pour sa part qu'il n'existe pas de formule qui puisse assurer de la validité d'un énoncé.

A.1.4. Théorie des ensembles et ensembles flous

Les mêmes résultats subsistent pour la théorie des ensembles de Zermelo-Fraenkel et d'ailleurs toute autre théorie pouvant fonder les mathématiques. La théorie des ensembles est à l'heure actuelle le cadre (plus vaste que l'arithmétique) le plus habituel pour fonder l'ensemble des mathématiques. Il s'agit de définir une structure (plusieurs structures, mais ZF+AC suffit aux mathématiques ordinaires) de graphe (U, =, \in) dans laquelle les relations binaires = et \in doivent vérifier les 5 axiomes suivants (avec l'écriture de l'inclusion $x \subset y$ pour $\forall z (z \in x \Rightarrow z \in y)$) :

AXIOME DE L'UNION

$$\forall x \exists y \forall z (z \in y \Leftrightarrow \exists t z \in t \text{ et } t \in x) \quad y \text{ est noté } U(x)$$

AXIOME DES PARTIES

$$\forall x \exists y \forall z (z \in y \Leftrightarrow z \subset x) \quad y \text{ est noté } P(x)$$

$$\text{AXIOME D'EXTENSION} \quad \forall x \forall y \forall z (z \in x \Leftrightarrow z \in y) \Rightarrow x = y$$

(deux ensembles sont égaux si et seulement si ils ont mêmes éléments)

SCHÉMA D'AXIOME DE SUBSTITUTION

Pour toute relation binaire fonctionnelle F, l'image d'un ensemble est un ensemble

$$\forall x \exists y \forall z z \in y \Leftrightarrow \exists u (u \in x \text{ et } F(u, z))$$

On montre alors l'existence d'un élément minimal pour \in (l'ensemble vide \emptyset), et l'existence des paires pouvant se former avec deux objets quelconques de U :

$\forall x \forall y \exists z \forall t (t \in z \Leftrightarrow (t = x \text{ ou } t = y))$, z étant noté $\{x, y\}$, et on peut définir alors le produit, l'union et intersection, les entiers finis $0 = \emptyset, 1 = \{\emptyset\}, 2 = \{\emptyset, \{\emptyset\}\}, \dots, n+1 = n \cup \{n\}$. On définit également les ordinaux comme ensembles transitifs (a tel que $\forall x \forall y (x \in a \text{ et } (y \in x) \Rightarrow y \in a)$ où \in est un bon ordre strict.

Cependant, les mathématiques classiques imposent un dernier axiome :

AXIOME DE L'INFINI

$\exists x (\emptyset \in x)$ et $\forall y (y \in x \Rightarrow y \cup \{y\} \in x)$, il existe un ordinal non fini.

En résumé, il est possible alors de montrer que tous les ordinaux finis forment un ensemble (le «dénombrable» noté ω), et de définir les cardinaux comme ordinaux non équipotents à un ordinal inférieur, \aleph_0 est le premier cardinal infini.

Cette axiomatique ZF n'est pas finiment axiomatisable à cause du schéma de substitution, les résultats de Gödel et de Tarski sont encore valable dans ZF et ses prolongements. A partir de sa consistance (l'existence d'une telle structure) on peut montrer l'existence d'univers parallèles tels que ZF + AF, ZF + AC, ZF + HGC, ou bien ZF + \neg AF, ZF + \neg AC, ZF + \neg HGC, pour citer les énoncés les plus connus avec :

AXIOME DE FONDATION AF : $\forall x \neq \emptyset \exists y \in x (x \cap y = \emptyset)$, en particulier il n'y a pas d'atome (c'est à dire d'ensemble a vérifiant $a = \{a\}$)

AXIOME DU CHOIX AC : Sur tout ensemble il existe au moins un bon ordre, ce qui fait que tout ensemble est en bijection avec un cardinal (une douzaine d'énoncés importants sont équivalents à AC, notamment celui de Zorn dont il est question à l'annexe 4).

HYPOTHÈSE DU CONTINU (généralisée) HGC : Le cardinal infini $\aleph_{\alpha+1}$ suivant \aleph_{α} est celui de $P(\aleph_{\alpha})$, c'est une conséquence de AF et l'axiome de constructibilité (tout ensemble est définissable par une formule à une variable libre), par ailleurs, il est prouvé que HGC entraîne AC.

D'autres théories permettent de fonder les mathématiques, ainsi celle de Gödel-Bernays dont on peut montrer l'équivalence avec ZF. Dans cette théorie il existe un symbole supplémentaire de prédicat unaire M ($M(x)$ sera interprété par «x est un ensemble»).

AXIOMATIQUE DE ZERMELO

C'est la structure moins forte $(U, =, \in)$ munie des axiomes d'extensionnalité, de la paire : $[\forall x \forall y \exists z \forall t (t \in z \Leftrightarrow t = x \vee t = y)]$, z étant noté $\{x, y\}$, de l'union, des parties et du schéma d'axiome de compréhension : pour toute formule à une seule variable libre A (donc éventuellement paramétrée par des constantes)

$[\forall x \exists y \forall z (z \in y \Leftrightarrow z \in x \wedge A(z))]$, y étant noté $y = \{z \in x / A(z)\}$.

Cette axiomatique permet de définir les opérations ensemblistes usuelles et les ordinaux, mais pas les cardinaux. La meilleure introduction en théorie des ensembles se trouve dans [Krivine 72].

THÉORIE AXIOMATIQUE DES ENSEMBLES FLOUS

Plusieurs axiomatiques de la logique floue (avec une infinité de valeurs ont été réalisées [Pavelka 79], mais nous présentons un point de vue plus original. Une théorie du second ordre introduite par [De Glas 84] formalise la théorie des ensembles flous. Il est nécessaire d'avoir une classe V de variables (les ensembles) et une autre (notées par des lettres grecques, les degrés), le symbole d'égalité, un symbole d'ordre < pour les degrés et un symbole de relation ternaire noté \in , (on notera $x \in_{\alpha} y$ et on lira «x appartient à y avec le degré α »). Les 12 axiomes sont :

1 Extensionnalité : $\forall x \forall y \forall z \forall \alpha (z \in_{\alpha} x \Leftrightarrow z \in_{\alpha} y) \Rightarrow x = y$

2 L'appartenance est une fonction $\forall x \forall y \forall \alpha \forall \beta (x \in_{\alpha} y \wedge x \in_{\beta} y) \Rightarrow \alpha = \beta$

3 Les degrés forment un ordre total avec extrémités, on pose $D(\alpha) = \exists x \exists y (x \in_{\alpha} y)$ qui signifie «être un degré» :

$\forall \alpha \forall \beta \forall \gamma [(\alpha < \beta \Rightarrow D(\alpha) \wedge D(\beta)) \wedge (D(\alpha) \Rightarrow \alpha < \alpha) \wedge (D(\alpha) \wedge D(\beta) \wedge D(\gamma) \wedge \alpha < \beta \wedge \beta < \gamma \Rightarrow \alpha < \gamma) \wedge (D(\alpha) \wedge D(\beta) \wedge \alpha < \beta \wedge \beta < \alpha \Rightarrow \alpha = \beta) \wedge (D(\alpha) \wedge D(\beta) \Rightarrow \alpha < \beta \vee \beta < \alpha) \wedge (\exists 0 \exists 1 \forall \alpha D(0) \wedge D(1) \wedge (D(\alpha) \Rightarrow 0 < \alpha \wedge \alpha < 1)]$

On prend les définitions :

$E(x) = (\forall y \forall \alpha (y \in_{\alpha} x) \Rightarrow \alpha = 0 \vee \alpha = 1)$ pour «être un ensemble exact»

et pour l'inclusion : $(x \sqsubset y) = \forall z \forall \alpha (z \in_{\alpha} x \Rightarrow \exists \beta \beta < \alpha \wedge z \in_{\beta} y)$

4 Axiome de la paire : $\forall x \forall y \forall z \exists p (E(p) \wedge (z \in_1 x \Leftrightarrow z = x \vee z = y))$

p sera noté $\{x, y\}$ y compris pour des degrés et un couple (x, y) est par définition, comme dans la théorie classique des ensembles, $\{x, \{x, y\}\}$.

5 Existence de degrés complémentaires

$\exists c [\forall x \exists \alpha \exists \beta (E(c) \wedge (x \in_1 c \Leftrightarrow D(\alpha) \wedge D(\beta) \wedge x = (\alpha, \beta))$

$\wedge \forall \alpha \forall \beta \forall \gamma ((\alpha, \beta) \in_1 c \Rightarrow (\beta, \alpha) \in_1 c) \wedge$

$[(\alpha, \beta) \in_1 c \wedge (\alpha, \gamma) \in_1 c \Rightarrow \beta = \gamma)]$

$\wedge \forall \alpha \forall \beta \forall \alpha' \forall \beta' (\alpha, \beta) \in_1 c \wedge (\alpha', \beta') \in_1 c \wedge (\alpha < \alpha') \Rightarrow \beta' < \beta)$

Cet ensemble c est l'ensemble des «couples complémentaires», on peut alors noter $\neg \alpha$ pour β si $(\alpha, \beta) \in_1 c$.

6 Axiome du complémentaire (x' sera noté $\neg x$) $\forall x \exists x' \forall y (y \in_{\alpha} x) \Rightarrow (y \in_{\neg \alpha} x')$

On montre alors que $\forall x \forall y \forall \alpha (y \in_{\neg \alpha} x \Leftrightarrow y \in_{\alpha} \neg x)$

7 Axiome de l' α -coupe $\forall x \forall \alpha \forall \beta \exists ! y \forall z (E(y) \wedge (\alpha < \beta \wedge z \in_{\beta} x \Leftrightarrow z \in_1 y))$

On note $\gamma = \max(\alpha, \beta)$ la formule $D(\gamma) \wedge (\alpha < \beta \Rightarrow \gamma = \beta) \wedge (\beta < \alpha \Rightarrow \gamma = \alpha)$

8 Axiome de l'union (r noté $U(x)$) :

$\forall x \exists r \forall y \forall \gamma (y \in_{\gamma} r \Leftrightarrow \exists z \exists \alpha \exists \beta y \in_{\alpha} z \wedge z \in_{\beta} x \wedge \gamma = \max(\alpha, \beta))$

9 Axiome des parties : $\forall x \forall y \exists p (E(p) \wedge (y \in_1 p \Leftrightarrow y \sqsubset x))$

10 Axiome de fondation : $\forall x \forall \alpha \neq 0 \neg(x \in_{\alpha} x)$

11 Axiome de l'infini :

$\exists x (\exists y \exists \alpha \neq 0 y \in_{\alpha} x) \wedge \forall y \forall \alpha (y \in_{\alpha} x) \Rightarrow \exists \beta U(\{y, \{y\}\}) \in_{\beta} x$

12 Schéma d'axiome de substitution

Soient F et G deux formules avec au moins deux variables libres, alors :

$$\forall a \forall b \forall x \forall y \forall y' [(F(x, y, a) \wedge F(x, y', a) \Rightarrow y = y')$$
$$\wedge \forall \alpha \forall \beta (G(y, \alpha, b) \wedge G(y, \beta, b) \Rightarrow D(\alpha) \wedge \alpha = \beta)]$$
$$\Rightarrow \forall z \exists t \text{ (appelé image de } x \text{ par } F)$$
$$\forall y \forall \alpha (y \in_{\alpha} t) \Leftrightarrow \exists x \exists \beta \alpha < \beta \wedge (z \in_{\beta} x) \wedge F(x, y, a) \wedge G(y, \alpha, b)$$

Avec ces axiomes, il est possible de définir l'union et l'intersection en prouvant qu'il s'agit bien d'opération correspondant respectivement aux max et min avec toutes les propriétés usuelles de treillis distributif complété ainsi que les lois de Morgan évoquées au chapitre I, et enfin de prouver l'existence de l'ensemble vide. On montre que cette axiomatique est consistante relativement à ZF.

A.1.5. La Récursivité

Toutes les notions suivantes sont définies sur \mathbb{N} et les produits cartésiens \mathbb{N}^P , des définitions analogues peuvent être faites dans la théorie plus vaste des ensembles avec les mêmes résultats.

LES FONCTIONS ÉLÉMENTAIRES sont définies comme le plus petit ensemble de fonctions contenant la fonction constante zéro, la fonction successeur, les projections, l'addition, la multiplication, la quasi-différence définie par $x \div y = [\text{si } (x \leq y) \text{ alors } x - y \text{ sinon } 0]$, et stable par sommation, produit et composition.

On peut montrer alors que ces fonctions sont exactement celles qui sont PL_2 -calculables par programmes comprenant 0, «entrée», «sortie», «incrémenter», «boucle simple répéter n fois», et limités à 2 boucles imbriquées.

LES FONCTIONS PRIMITIVES RÉCURSIVES sont une classe FRP plus vaste de fonctions constituée par le plus petit ensemble de fonctions contenant la constante zéro, la fonction successeur, les projections, et stable par composition et récurrence simple.

Une autre identification de FRP est l'ensemble des fonctions PL -calculables par programme comprenant 0, «successeur», «entrée», «sortie», «affectation», «test», «boucle», mais pas le débranchement «goto».

Un ensemble de \mathbb{N}^P (un prédicat) est récursif primitif, si par définition, sa fonction caractéristique est récursive primitive.

UN PRÉDICAT (OU PARTIE DE \mathbb{N}^P) EST RÉCURSIVEMENT ÉNUMÉRABLE (semi-décidable) s'il est la projection d'un ensemble récursif primitif. Il s'agit en fait du plus petit ensemble de parties contenant $\{1\}$, les graphes de l'addition et de la multiplication, et stable par changement de places, union, intersection, quantification existentielle et quantification universelle bornée. C'est l'ensemble des objets définissables par les formules Σ_1 de la hiérarchie arithmétique des formules, et également l'ensemble des prédicats reconnus par automate de Turing (A est semi-décidable si l'automate reconnaît les énoncés $x \in A$, mais pas ceux du type $x \notin A$). Les ensembles récursivement énumérables ont également 5 autres identifications par différentes familles de systèmes de règles tel que système de Post.

LES FONCTIONS RÉCURSIVES (calculables) sont le plus petit ensemble de fonctions contenant zéro, la fonction successeur, les projections et stable par composition, récurrence et minimisation, c'est aussi l'ensemble des fonctions dont le graphe est récursivement énumérable (FRP est énumérable dans FR).

Les fonctions récursives sont également identifiées suivant quatre autres points de vue : les fonctions Turing-calculables, Markov-calculables, PLD-calculables et les fonctions représentables dans le λ -calcul :

UN AUTOMATE DE TURING est un quintuplet $(A, Q, \{G, S, D\}, P)$ où A est un alphabet fini, par exemple deux signes $\{0, 1\}$, Q un ensemble fini «d'états» comportant notamment l'état initial q_0 et l'état final q_1 , et P le «programme» constitué par une fonction partielle de A^*Q dans $A^*\{G, S, D\}^*Q$.

Un calcul à partir d'un mot u de A^* (ensemble des suites finies d'éléments de A) est alors une suite de «descriptions» notée $q_0 u \vdash, \dots, \vdash (v, q_1)$ telle qu'à chaque étape on ait, dès lors que $P(a, q) = (a', M, q')$, (u ou v pouvant être vides) $uqav \vdash uq'a'v$ au cas où $M = S$, $uqav \vdash uq'fa'v$ si $M = G$ et $u = u'f$ avec $f \in A$ et enfin $uqav \vdash ua'q'v$ si $M = D$. Le calcul s'arrête dès que q' est l'état final.

Une fonction f est T-calculable si pour l'écriture binaire de x, elle s'arrête sur l'état final pour la valeur binaire de f(x) et ne s'arrête pas si f(x) n'est pas défini.

UNE GRAMMAIRE DE MARKOV sur l'alphabet $A = \{0, 1\}$ est la donnée d'un alphabet auxiliaire B et d'une liste P de règles de «production» de couples (u, u') dans $(A \cup B)^*$. Une dérivation à partir d'un mot de A^* (ensemble des suites finies d'éléments de A) est une suite de mots telle qu'à chaque étape la première production applicable dans la lecture gauche à droite, fasse passer au mot suivant. Une fonction f est alors dite M-calculable s'il existe une grammaire de Markov telle que pour tout x en représentation binaire, il y a une dérivation débutant en x et se terminant en f(x), ou ne terminant pas si f(x) n'est pas défini. Une fonction est PLD-calculable si elle se calcule par programme comprenant 0, «successeur», «entrée», «sortie», «affectation», «test», «boucle», et le débranchement «goto».

REPRÉSENTABILITÉ EN λ -CALCUL

Le λ -calcul est le plus petit ensemble de termes contenant un ensemble dénombrable de variables et stable par «l'application» (uv) si u et v sont des termes, et par "l'abstraction" $\lambda x t$ où x est une variable et t un terme. On définit une α -équivalence des termes $\lambda x u$ et $\lambda x' u'$ dans la mesure où u' est équivalent à u dans lequel x' est substitué à toutes les occurrences de x (ils ne diffèrent que par des changements de nom des variables). Λ est l'ensemble quotient.

La β -réduction consiste à déduire d'un terme, un terme sans «rédex» c'est à dire sans sous-terme de la forme $(\lambda x u)t$, terme que l'on remplace par u [où t prend la place de x]. Cela signifie concrètement que l'on applique la fonction u de variable x au terme t . La forme normale (sans rédex) est unique dans Λ .

On pose $I = \lambda x x$ (l'identité) et $F = \lambda x \lambda y y$ (le faux ou zéro), $K = V = \lambda x \lambda y x$ (le vrai), ensuite peuvent être définis la condition $IF = \lambda b \lambda x \lambda y b x y$, puis la négation $\neg = \lambda x (IF x F V)$, les connecteurs $ET = \lambda x \lambda y x y F$, $OU = \lambda x \lambda y x V y$, enfin les entiers de Church :

$\underline{0} = F$, $\underline{1} = \lambda f \lambda x (f x)$, $\underline{2} = \lambda f \lambda x (f (f x))$ et $\underline{n} = \lambda f \lambda x (f^k x)$.

On dit alors que la fonction f de N^m dans N est représentable en λ -calcul par F si et seulement si pour tous entiers k_1, \dots, k_n si $f(k_1, \dots, k_n)$ n'est pas défini alors l'expression $F \underline{k}_1, \dots, \underline{k}_n$ n'est pas réductible sinon si $f(k_1, \dots, k_n) = k$ alors l'expression $F \underline{k}_1, \dots, \underline{k}_n$ se réduit en \underline{k} .

On peut en effet construire toutes les fonctions récursives telles que :

$SUC = \lambda n \lambda f \lambda x ((n f) (f x))$, la somme $+$ $= \lambda n \lambda m \lambda f \lambda x ((n f) ((m f) x))$ et le produit $*$ $= \lambda n \lambda m \lambda f \lambda x ((n (m f)) x)$.

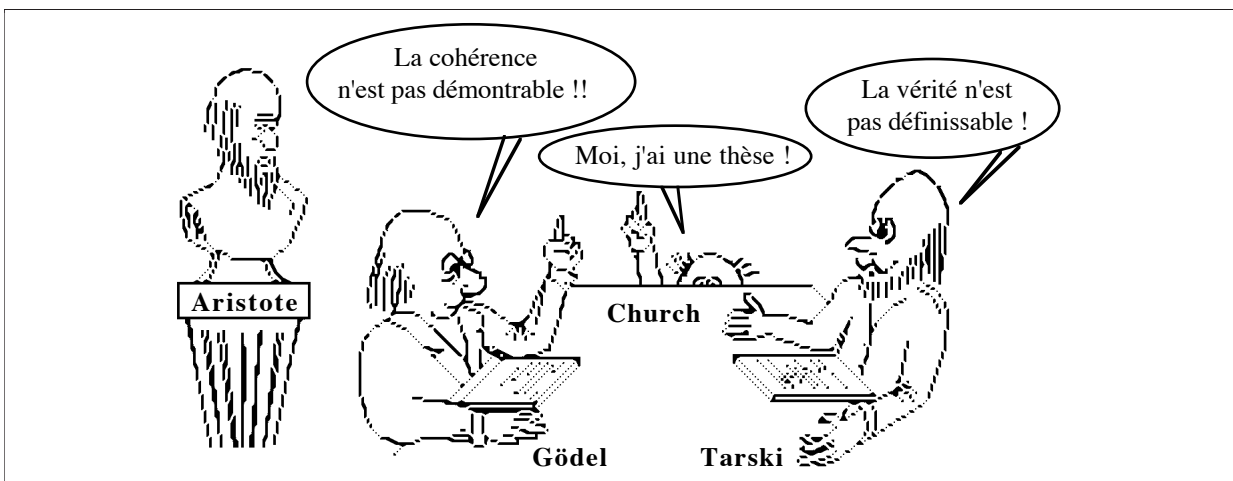
UN ENSEMBLE EST RÉCURSIF (décidable) si sa fonction caractéristique est récursive, cela revient au même de dire qu'il est récursivement énumérable ainsi que son complémentaire. Ce sont les ensembles $PR = \Delta_0$ définissables par une formule sans quantificateur. On montre que PRE est auto-énumérable et que l'on a les inclusions strictes $PRP \subset PR \subset PRE$.

D'autres notions ont été élaborées dans le but de formaliser la notion "d'ensemble aléatoire", ainsi un ensemble est «inapproximable» s'il est infini et ne contient aucun ensemble infini récursivement énumérable.

Un automate de Turing qui peut être assimilé à la suite finie des instructions de son programme P , est donc (codé en binaire) assimilé à un entier k de taille m (m bits). Or, pour une taille m donnée, la probabilité pour un automate de Turing de taille m de s'arrêter sur k à partir d'une donnée vide, est $1 / 2^m$. Pour tout entier n , on pose $H(n)$ la complexité de Chaitin-Kolmogorov, comme la taille du plus petit programme s'arrêtant sur n . Le nombre de Chaitin Ω est défini comme la probabilité qu'un automate de Turing s'arrête, donc $\Omega = \sum_{(n \in N)} 1/2^{H(n)}$. On montre que ce réel, (ce qui revient au même, cette suite binaire, ou cet ensemble d'entiers) est définissable dans $\Delta_2 - (\sum_1 \cup \prod_1)$ et qu'il est incompressible [Delahaye 94].

Un ensemble A est dit aléatoire ou "incompressible" s'il existe un entier c tel que pour tout programme p , les énoncés $n \in A$ ou leur négation qui sont reconnus par p sont en nombre inférieur à $H(p) + c$.

Enfin, au second ordre, les problèmes résolus en temps polynômial par une machine de Turing non déterministe (NP) sont ceux qui sont définis au second ordre par une formule \sum_1^1 , c'est à dire quantifiée existentiellement sur des relations suivi par une formule du premier ordre.



La thèse de Church n'est pas un énoncé mathématique, mais l'assertion suivant laquelle, jusqu'à preuve du contraire, les fonctions récursives sont les fonctions «effectivement calculables dans la pratique».

Il existe des extensions de la logique classique telles que les langages infinis (conjonctions infinies ou quantifications infinies) et la logique du second ordre (quantification sur des relations par exemple).

Annexe 2

Logiques non classiques

A.2.1. L'Intuitionisme

La logique intuitioniste (Heyting 1930) s'attache davantage à la connaissance que l'on peut avoir de la vérité des énoncés, elle est en cela plus proche de la programmation informatique, dans la mesure où elle ne prend en compte que la constructibilité de preuves. Elle est née du rejet du principe du tiers-exclu et du raisonnement par l'absurde. Pour l'intuitionnisme, une preuve est un moyen effectif, par exemple une preuve de $A \rightarrow B$ est un algorithme de passage de toute preuve de A à une preuve de B (aspect constructiviste). A chaque dérivation intuitionniste, on peut associer un algorithme qui est un programme lié au λ -calcul (voir l'arithmétique fonctionnelle de J.L.Krivine).

Une axiomatique de l'intuitionnisme est réalisée par le système de Hilbert de l'annexe 1 dans lequel l'axiome $\neg\neg A \rightarrow A$ est remplacé par $\neg A \rightarrow (A \rightarrow B)$. Sans l'axiome $\neg\neg A \rightarrow A$, le raisonnement par l'absurde consistant en $P \wedge \neg Q \rightarrow 0$ c'est à dire $\neg P \vee \neg\neg Q$ n'est pas équivalent à $\neg P \vee Q$ qui est $P \rightarrow Q$.

On prouve alors la formule $\neg(A \wedge \neg A)$ indiquant qu'il n'y a pas contradiction, mais pas le tiers-exclu $A \vee \neg A$ comme en logique classique. On montre cependant que le calcul propositionnel intuitionniste est décidable.

Par ailleurs il existe des sémantiques de Kripke ou topologique de Heyting assurant le théorème de complétude.

Un modèle de Kripke est $K = (C, F)$ où $C = \{\alpha, \beta, \gamma, \dots\}$ est un ensemble pré-ordonné de «conditions» et F est un ensemble de couples (α, p) du produit cartésien $C^* \text{Atomes}$ vérifiant $(\alpha, A) \in F$ et $\alpha \leq \beta \Rightarrow (\beta, A) \in F$. On définit la relation «force» notée \Vdash par les 6 clauses :

- non $(\alpha \Vdash 0)$, si A est atomique $\alpha \Vdash A \Leftrightarrow (\alpha, A) \in F$
- $\alpha \Vdash \neg A \Leftrightarrow \forall \beta \geq \alpha \text{ non } (\beta \Vdash A)$
- $\alpha \Vdash (A \wedge B) \Leftrightarrow (\alpha \Vdash A) \text{ et } (\alpha \Vdash B)$
- $\alpha \Vdash (A \vee B) \Leftrightarrow (\alpha \Vdash A) \text{ ou } (\alpha \Vdash B)$
- $\alpha \Vdash (A \rightarrow B) \Leftrightarrow \forall \beta \geq \alpha (\beta \Vdash A \Rightarrow \beta \Vdash B)$

On définit alors A valide dans K si et seulement si $\forall \alpha \alpha \Vdash A$ et on montre la consistance, à savoir que $(\alpha \Vdash A \text{ et } \alpha \Vdash \neg A)$ est impossible, la monotonie : $(\alpha \Vdash A \text{ et } \alpha \leq \beta \Rightarrow \beta \Vdash B)$ ainsi que la complétude :

A est valide dans tout modèle de Kripke $\Leftrightarrow A$ démontrable intuitionnistiquement.

Cette sémantique s'étend aux langages de prédicats intuitionnistes.

A.2.2. La logique modale

Alors que l'intuitionnisme est plus faible que la logique classique, la logique modale la complète. Elle est fondée sur le fait que la définition de $A \rightarrow B = \neg A \vee B$ vide la notion d'implication de sa signification causale s'il n'y a aucun lien entre A et B car il suffit en effet, en logique binaire classique, que A soit faux pour avoir $A \rightarrow B$ vrai.

On distingue alors deux connecteurs duaux le nécessaire \Box et le possible \Diamond , liés par la relation $\Diamond P = \neg \Box \neg P$. $\Box A$ signifie que A est assuré comme vrai, et $\Diamond A$, qu'il est simplement possible c'est à dire compatible avec les connaissances actuelles.

Si ces deux connecteurs sont idempotents, et si on remplace chaque P par $\Box P$, alors tout théorème de logique intuitionniste est un théorème de logique modale.

Sémantiquement un modèle est un triplet $M = (W \text{ mondes possibles}, R \text{ relation binaire sur } W, v \text{ une fonction valeur du produit } W^* \text{Atomes sur } \{0,1\})$ et on définit :

- $(M, w) \models A \Leftrightarrow v(w, A) = 1$
- $(M, w) \models \neg A \Leftrightarrow \text{non } (M, w) \models A$
- $(M, w) \models A \wedge B \Leftrightarrow (M, w) \models A \text{ et } (M, w) \models B$
- $(M, w) \models \Box A \Leftrightarrow \forall w' \text{ si } (w R w') \text{ alors } (M, w') \models A$

On aura donc : $(M, w) \models \Diamond A \Leftrightarrow \exists w' (w R w') \text{ et } (M, w') \models A$
 Pour une sémantique dans un espace ordonné (le temps) voir [Gabbey 82].

Axiomatiquement, on prend l'axiomatique du calcul propositionnel plus l'axiome de distribution :

$\Box (A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)$ équivalent à $(\Box (A \rightarrow B) \wedge \Box A) \rightarrow \Box B$, la règle du modus-ponens et la règle de nécessité $A \vdash \Box A$ pour le système appelé K, d'autres axiomatiques plus complètes ont été imaginé, pour lesquelles existe un théorème de complétude :

Système KT : R est réflexive ce qui revient à indiquer en plus la «connaissance» par l'axiome $\Box A \rightarrow A$.

Système S4 : R est réflexive et transitive ce qui revient à l'axiome supplémentaire $\Box A \rightarrow \Box \Box A$ (introspection positive, équivalente à $\Diamond \Box A \rightarrow \Box A$).

Système B : R est réflexive et symétrique, soit en plus : $A \rightarrow \Box \Diamond A$

Système S5 : R est une équivalence, c'est le système S4 plus l'axiome d'introspection négative $\Diamond A \rightarrow \Box \Diamond A$.

LA LOGIQUE AUTO-EPISTÉMIQUE

La logique autoépistémique [Moore 83, 88], est la logique classique, augmentée d'un seul connecteur modal \Box dont l'interprétation sémantique est :

$\Box A$ est vrai $\Leftrightarrow A \in T$ (la théorie)

L'objectif est de formaliser les concepts de croyance et de justification. On montre, si $th(S)$ désigne la théorie engendrée par les axiomes de S, dans le calcul des propositions, l'équivalence de stabilité :

$T \models th(S \cup \{\Box A / A \in T\} \cup \{\neg \Box A / A \notin T\})$

T est dit «basé» ou «fondé» sur les axiomes S

\Leftrightarrow Toute assignation satisfaisant les formules de S (un modèle de S) est aussi modèle de T

Une assignation v donne une valeur de $\{0, 1\}$ à chaque variable propositionnelle et est étendue de façon classique en une interprétation des formules en rajoutant la définition autoépistémique $[v(\Box A) = 1] \Leftrightarrow A \in T$.

Puis on montre une «complétude» :

T est complète (contient toutes les formules vraies dans chaque modèle de T)

$\Leftrightarrow T = th(T)$ et $(A \in T \Rightarrow \Box A \in T)$ et $(A \notin T \Rightarrow \neg \Box A \in T)$ (T point fixe de th et stable).

EXEMPLES

Si $S_1 = \{p\}$ alors $T = th(\neg \Box p, \neg \Box \neg p, \Box \neg \Box p, \dots)$ est une théorie autoépistémique complète basée sur $\{p\}$.

Soit maintenant, $S_2 = \{\Box p \rightarrow p\}$, il possède cette fois deux extensions stables $T_1 = th(p, \Box p, \neg \Box \neg p, \dots)$ et $T_2 = th(\neg \Box p, \Box \neg \Box p, \dots)$ dont l'union est inconsistante, (deux mondes parallèles).

$S_3 = \{\neg \Box p \rightarrow q, \neg \Box q \rightarrow p\}$ possède deux extensions stables $T_1 = th(p, \neg \Box q, \Box p, \dots)$ et $T_2 = th(q, \neg \Box p, \Box q, \dots)$ dont l'union est également inconsistante.

La sémantique est celle décrite plus haut, mais avec $(M, w) \models \Box A \Leftrightarrow \forall w' (M, w') \models A$, ce qui revient à la structure S5 complète où la relation R est la relation d'équivalence triviale ($\forall w, w'$ on a $w R w'$).

A.2.3. La logique des défauts

Introduit par [Reiter 80], voir aussi [Froidevaux 85] et [Grégoire 90], ce point de vue veut gérer des connaissances typiques ou particulières, mais sans s'engager dans des problèmes d'ordre ou de nombre comme avec le flou, par exemple, "en général les routes sont assez larges pour que deux voitures se croisent», «en général, les oiseaux volent».

Il s'agit de donner un sens au raisonnement révisable. Plus précisément, si S est un ensemble d'axiomes et $th(S)$ la théorie qu'il engendre, la monotonie :

$S \subseteq T \Rightarrow th(S) \subseteq th(T)$ est remise en cause, ce qui signifie qu'ajouter un axiome n'élargit plus nécessairement la théorie engendrée.

Par exemple [Besnard 89] si $S = \{ami(Alex, Luc), ami(Luc, Max)\}$, la règle $ami(x, y) \wedge ami(y, z) : ami(x, z) \vdash ami(x, z)$ va prouver $ami(Alex, Max)$, le fait de rajouter $\neg ami(Alex, Max)$ dans S ne le permettra plus, bien que S reste consistante.

Les règles sont de la forme $(A : \text{défaut } B) \vdash C$, signifiant que lorsque A est vrai et qu'on ne peut pas prouver $\neg B$ alors on déduit C. On note $A \vdash_{\text{d}} C$ pour une déduction par défaut (A entraîne C tant que l'hypothèse B n'est pas contredite).

On a par exemple : $A \vdash B \Rightarrow A \vdash_{\text{d}} B$ ($\vdash A \Leftrightarrow B$ et $A \vdash_{\text{d}} C \Rightarrow B \vdash_{\text{d}} C$)

$[A \vdash_{\text{d}} B \text{ et } \vdash (B \rightarrow C)] \Rightarrow A \vdash_{\text{d}} C$ ($A \vdash_{\text{d}} B$ et $A \vdash_{\text{d}} C \Rightarrow A \vdash_{\text{d}} B \wedge C$)

La théorie engendrée par un ensemble d'axiomes S et de règles R de ce type, se définit comme en logique classique, (une extension), c'est le plus petit ensemble de formules contenant S et stable par les règles. Un point fixe de l'opérateur th , qui peut encore être défini par induction comme : $E = \cup_{i \in \mathbb{N}} E_i$ avec $E_0 = S$ et $E_{i+1} = E_i \cup \{C / (A : B_1, B_2, \dots, B_n \vdash C) \in R \text{ et } \forall i \neg B_i \notin E\}$

EXEMPLE 1

Pour les règles $R = \{1 : A \vdash C, C : B \vdash \neg D, 1 : D \vdash B\}$, si 1 désigne le vrai, on déduit $\{C, \neg D\}$ qui est l'unique extension.

EXEMPLE 2

$S = \emptyset$ et $R = \{1 : A \vdash B, 1 : \neg A \vdash C, B : D \vdash D, D : B \vdash A, C : \neg B \vdash \neg A\}$, possède deux extensions (deux mondes parallèles) $\{A, B, D\}$ et $\{C, \neg A\}$

EXEMPLE 3

$S = \emptyset$ et $R = \{1 : \neg A \vdash \neg A, 1 : \neg B \vdash \neg B, 1 : A \vee B \vdash A \vee B\}$, possède les trois extensions $\{\neg A, \neg B\}$, $\{\neg A \vdash \neg A, A \vee B\}$ et $\{\neg B, A \vee B\}$.

EXEMPLE 4

$S = \{A\}$ et $R = \{A : B \vdash C, A : D \vdash \neg C\}$, n'a pas extension.

Il est possible que l'union de deux extensions soit consistante, c'est le cas pour $S = \emptyset$ et $R = \{1 : \neg A \vdash B, 1 : \neg B \vdash A\}$ dont les deux extensions sont $\{A\}$ et $\{B\}$ pouvant cohabiter.

Un défaut est dit «normal» s'il est de la forme $A : B \vdash B$.

En particulier «H est une hypothèse» peut s'écrire $1 : H \vdash H$. Lorsque tous les défauts sont normaux, il y a au moins une extension, et toutes les extensions sont mutuellement incompatibles.

LA LOGIQUE DES PRÉDICATS NON MONOTONE DE MAC DERMOTT

C'est une logique [Mc Dermott 82] plus forte que le système S5 augmenté de l'axiome de Barcan $(\forall x \Box A) \rightarrow \Box(\forall x A)$ et de la règle d'inférence de généralisation $A \vdash \forall x A$, ainsi que d'une dernière règle dont l'expression non formalisée est «On ne peut inférer $\neg A \vdash \Diamond A$. Si S est un ensemble d'axiomes, à côté de $\text{Th}(S)$ définie traditionnellement comme la théorie engendrée par S, on pose l'ensemble des hypothèses de S relativement à U c'est à dire consistantes avec U comme :

$$\text{Hyp}(S / U) = \{\Diamond p / p \text{ formule close et } \neg p \notin U\} - \text{Th}(S)$$

Ce que l'on note «Th~ théorie non monotone» engendrée par les axiomes de S est le plus petit point fixe U vérifiant $U = \text{Th}(S \cup \text{Hyp}(S / U))$. On pose alors $p \vdash_{\sim} q$ si et seulement si $q \in \text{th}_{\sim}(p)$.

Cette règle permet d'admettre que A est possible jusqu'à ce que $\neg A$ soit démontré, auquel cas $\Diamond A$ et toutes ses conséquences doivent être retirées. Mac Dermott montre qu'il y a équivalence entre la théorie de S5 et cette théorie non monotone.

Exemples, pour $S = \{\Diamond p \rightarrow \neg q, \Diamond q \rightarrow \neg p\}$, $S \cup \{\neg p\}$ et $S \cup \{\neg q\}$ sont les deux plus petit points fixes.

Pour l'unique axiome $\Diamond p \rightarrow \neg p$, il ne peut y en avoir.

A.2.5. Les logiques temporelles

«Irma est veuve, Max a épousé Irma donc Max a épousé une veuve, mais alors Irma n'est plus veuve». C'est le genre de contradiction qui apparaît lorsqu'on ne tient pas compte de l'évolution d'un système dans le temps. Des systèmes de prédiction ou de contrôle en temps réel (gestion des alarmes etc...) doivent pouvoir raisonner, outre dans l'incertain, dans le temps. Ces raisonnements sont donc non monotones, à savoir que des conclusions peuvent être remises en cause par de nouvelles informations arrivant dans le temps. Une proposition dans la logique temporelle, peut avoir différentes valeurs de vérité à des instants différents.

On introduit dans la logique des opérateurs temporels unaires F et P :

Fp : p sera vrai au moins une fois dans le futur

Pp : p a été vrai au moins une fois dans le passé.

Gp : p sera toujours vrai dans le futur (dorénavant), Gp est défini par $\neg F\neg p$

Hp : p a toujours été vrai dans le passé (jusqu'à présent), Hp est défini par $\neg P\neg p$

SÉMANTIQUE

Afin de formaliser les différentes logiques temporelles qui suivent, il est nécessaire, comme en logique modale, de considérer un graphe appelé «cadre temporel» (T, R) pour représenter le temps, dans lequel R est une relation binaire (au minimum transitive pour le système L₀, mais pas obligatoirement un ordre total).

Si F est l'ensemble des formules, un modèle est (T ensemble d'instants, R relation binaire sur T, v fonction F^*T vers $\{0,1\}$) et :

$$v(Gp, t) = 1 \Leftrightarrow \forall u \text{ si } (t R u) \text{ alors } v(p, u) = 1$$

$$v(Hp, t) = 1 \Leftrightarrow \forall u \text{ si } (u R t) \text{ alors } v(p, u) = 1$$

$$v(Fp, t) = 1 \Leftrightarrow \exists u \text{ (t R u) et } v(p, u) = 1$$

$$v(Pp, t) = 1 \Leftrightarrow \exists u \text{ (u R t) et } v(p, u) = 1$$

(ces deux dernières étant déduites de $v(\neg p, t) = 1 \Leftrightarrow v(p, t) = 0$)

et naturellement la sémantique des propositions classiques :

$$v(\neg p, t) = 1 \Leftrightarrow v(p, t) = 0$$

$$v(p \wedge q, t) = 1 \Leftrightarrow v(p, t) = 1 \text{ et } v(q, t) = 1$$

$$v(p \vee q, t) = 1 \Leftrightarrow v(p, t) = 1 \text{ ou } v(q, t) = 1$$

L₀ La plus simple axiomatique est alors le système L₀ contenant les axiomes de la logique classique avec les 4 axiomes supplémentaires :

$$\begin{aligned} G(p \rightarrow q) &\rightarrow (Gp \rightarrow Gq) & p &\rightarrow GPp \\ H(p \rightarrow q) &\rightarrow (Hp \rightarrow Hq) & p &\rightarrow HFp \end{aligned}$$

et la règle ordinaire du modus-ponens : si A et $A \rightarrow B$ sont prouvés, alors B l'est aussi, ainsi que la règle dite de généralisation temporelle : si p est un théorème, alors Gp et Hp le sont aussi.

Un certain nombre de théorèmes de complétude ont été prouvés, il s'agit des théorèmes montrant qu'une proposition est déduite par la syntaxe si et seulement si elle est vraie relativement à une sémantique.

Les différentes axiomatiques ci-dessous correspondent alors à une propriété particulière de la relation d'antériorité R.

L₁ (ordre) est L₀ augmenté de l'axiome $Gp \rightarrow GGp$ qui indique la transitivité. Remarquons que, par définition de G, cet axiome (ou plutôt ce schéma d'axiome valable pour toute proposition p) est équivalent à : $(\neg F\neg p \rightarrow \neg F\neg\neg F\neg p) = (F\neg p \vee \neg FF\neg p) = (FF\neg p \rightarrow F\neg p)$ soit : $(FFp \rightarrow Fp)$.

L_{arb} (ordre arborescent) est L₁ augmenté de :

$Pp \wedge Pq \rightarrow P(p \wedge Pq) \vee P(p \wedge q) \vee P(Pp \wedge q)$ l'ordre est linéaire en arrière (une arborescence, donc déterministe pour le passé et ramifié pour le futur).

L₂ (ordre total) est L₀ augmenté de deux axiomes : $Fp \wedge Fq \rightarrow F(p \wedge Fq) \vee F(p \wedge q) \vee F(Fp \wedge q)$ (l'ordre est linéaire vers l'avant, donc dans le futur) et aussi linéaire en arrière (donc dans le passé) : $Pp \wedge Pq \rightarrow P(p \wedge Pq) \vee P(p \wedge q) \vee P(Pp \wedge q)$

L₃ (ordre total avec extrémités) est L₂ et 2 axiomes (0 et 1 désigne le faux et le vrai) : $G0 \vee FG0$ et $H0 \vee PH0$.

L₄ (ordre total sans extrémités) est L₂ plus 2 axiomes $Gp \rightarrow Fp$, $Hp \rightarrow Pp$, (l'ordre n'a pas d'extrémités)

L₅ (ordre total dense) est L₂ et $Fp \rightarrow FFp$

L_Q (ordre total dense dénombrable) est la réunion de L₄ et L₅ ($Fp \rightarrow FFp$ qui entraîne $Pp \rightarrow PPp$). L_Q est isomorphe à l'ensemble Q des rationnels, et un théorème du à Hamblin montre que toutes les combinaisons de «temps» avec les opérateurs G, H, F, P, se ramène à 15 temps reliés par des implications :

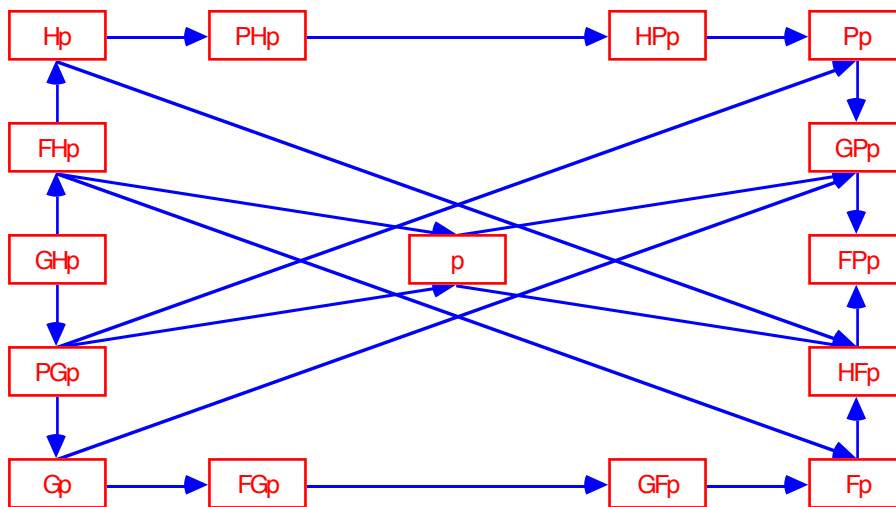


Figure A.2.1 Relations d'inférences entre les 15 temps dans L_Q.

L₆ (ordre total discret) est L₂ et 2 axiomes $p \wedge Hp \rightarrow FHp$, $p \wedge Gp \rightarrow PGp$

L₇ (ordre total complet c'est à dire sans coupure ouverte. Une coupure non ouverte, (fermée) est une partition de T en deux intervalles I et J telle qu'il existe s avec $s = \max(I)$ ou bien $s = \min(J)$). R est complet, Q ne l'est pas en prenant par exemple $I = \{x / x^2 < 2\}$ et $J = \{x / x^2 > 2\}$)

L₇ est L₂ augmenté de l'axiome :

$$Fp \wedge FG\neg p \rightarrow F(HFp \wedge G\neg p), \text{ et } Pp \wedge PH\neg p \rightarrow P(HPp \wedge H\neg p)$$

L_R est L_Q avec $Fp \wedge FG\neg p \rightarrow F(HFp \wedge G\neg p)$ et aussi :

$$Pp \wedge PH\neg p \rightarrow P(GPp \wedge H\neg p)$$

L₈ (bon ordre : toute partie de T possède un plus petit élément), c'est L₂ avec :

$$H(Hp \rightarrow p) \rightarrow Hp$$

L₉ (treillis, temps ramifié où chaque couple a un majorant commun et un min) L₂ augmenté de $Gp \rightarrow Fp$, $Hp \rightarrow Pp$, $FGp \rightarrow GFp$ et $PHp \rightarrow HPp$.

Les relations d'Allen [Allen 81] sur les intervalles de temps sont :

- $I < J \Leftrightarrow \forall u \in I \forall v \in J \quad u < v$, I précède J
- $I m J \Leftrightarrow \max(I) = \min(J)$, I touche J
- $I o J \Leftrightarrow \min(I) < \min(J) \text{ et } \min(J) < \max(I) < \max(J)$, I chevauche J
- $I d J \Leftrightarrow I \subset J$, I est dans J
- $I s J \Leftrightarrow \min(I) = \min(J) \text{ et } \max(I) < \max(J)$, I débute J
- $I e J \Leftrightarrow \min(I) > \min(J) \text{ et } \max(I) = \max(J)$, I termine J

Le système d'Allen est un ensemble d'intervalles muni des 13 relations formées par l'égalité, les 6 relations (antiréflexives) ci-dessus et leurs relations réciproques. Si $I = [x, y]$ est un intervalle, on peut représenter I par le point (x, y) du demi-plan supérieur à la diagonale $x = y$ (qui représente d'ailleurs les intervalles réduits à un point), en ce cas les images de I par ces 13 relations sont figurées par les 6 zones ouvertes, les 6 portions de droites et le point I lui-même.

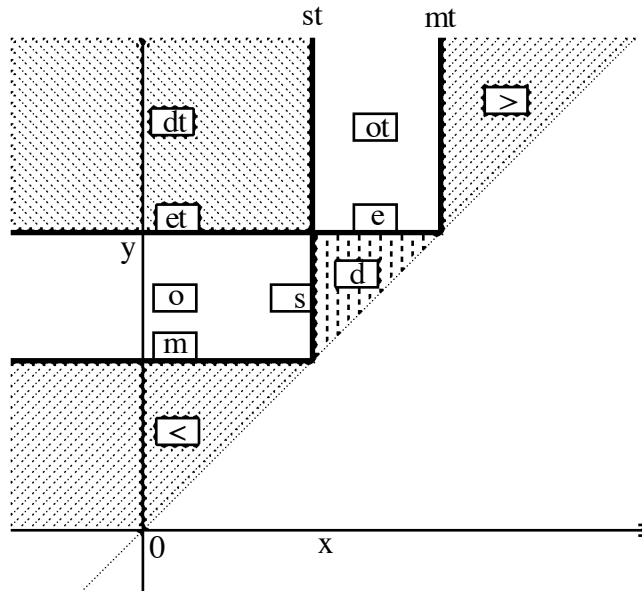


Figure A.2.2 Sur ce dessin du à [Bestougeff, Ligozat 89] la partie marquée «d» par exemple, est l'ensemble $d(I)$ des intervalles J tels que $J d I$ c'est à dire des intervalles J inclus dans I. La demi-droite marquée mt est l'ensemble des $J = [x', y']$ tels que $I m J$, c'est à dire «prolongeant» I avec $x' = y$.

LA LOGIQUE CAUSALE DE SHAFER [Shafer 96]

Il s'agit d'une construction axiomatique d'un système de logique temporelle et probabiliste faisant appel aux relations intuitives de «peut précéder», «peut suivre», «peut diverger», «fusionne» ...

Le but est de définir une structure «espace d'évènements» sur des éléments appelés «situations», par plusieurs relations d'ordres : $x \succ y$ (x se raffine en y, signifiant concrètement que y est une situation particulière de l'évènement x). Cet ordre doit être tel que toute partie A possède une borne inférieure appelée fusion de A, c'est la situation la plus grossière résumant toutes celles de A.

$x \rightarrow y$ (x permet y) relation devant vérifier $x \rightarrow y$ et $z \succ y \Rightarrow x \rightarrow z$

$x \Rightarrow y$ (x requis pour y, ou y requiert x) devant vérifier $x \Rightarrow y$ et $z \succ x \Rightarrow z \Rightarrow y$

$x < y$ (x précède y) $\Leftrightarrow x \rightarrow y$ et $x \Rightarrow y$ (x permet y et y requiert x)

$x \div y$ (x et y se chevauchent) x et y ont un raffinement commun (en d'autres termes sont «confluents» pour la relation «être raffiné par» \succ , c'est le cas de T et U ci-dessous dont Y est un raffinement).

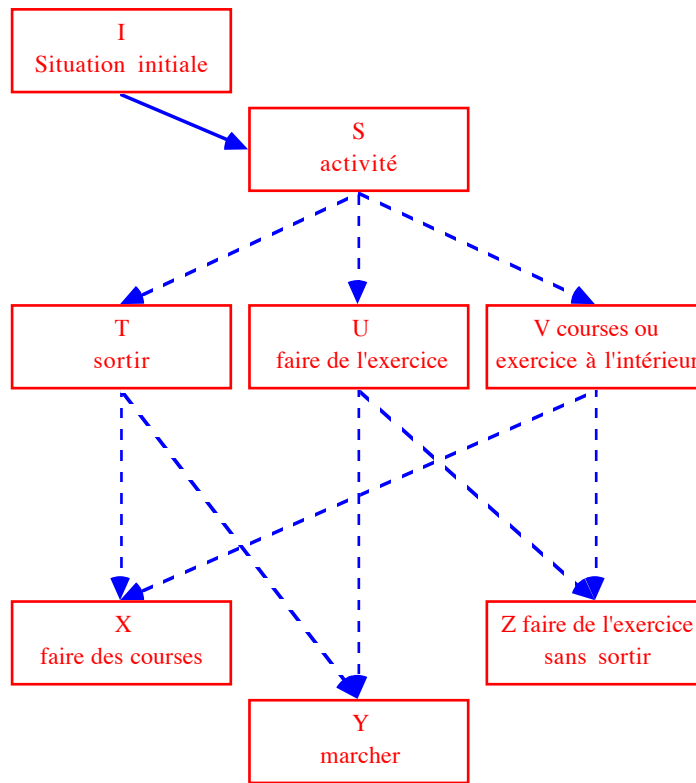


Figure A2.3 Exemple de raffinement de situations, en pointillé, la relation «se raffine en», en trait plein «précède». On imagine la complexité d'un tel arbre en augmentant le détail (raffinement des situations).

La complexité de cette structure réside dans le fait qu'on ne peut représenter ces situations dans un arbre (pour la relation $<$) ni même dans un arbre où chaque nœud serait une composante connexe vis à vis de la relation \succ . C'est pourquoi la définition formelle est donnée grâce à 5 relations de base dont R_1 qui est exactement \neq , et une axiomatique comportant notamment :

Axiome 1 (pour toutes situations, on a toujours au moins l'une des 5 relations satisfaite : $x R_1 y$ ou $x R_2 y$ ou $x R_3 y$ ou $x R_4 y$ ou $x R_5 y$, nommées respectivement «chevauche», «peut requérir», «peut appeler», «peut entraîner» et «peut diverger».

Axiome 5, à savoir : si $x R_i y$ pour l'une quelconque de ces 5 relations, alors il existe toujours un raffinement x' de x lié à y par R_i mais non lié à y par aucune des 4 autres.

On pose alors les définitions :

$x R_2 y$ (x peut requérir y) $\Leftrightarrow y R_3 x$ ou $y R_4 x$

$x \succcurlyeq y$ (x requis pour y) $\Leftrightarrow x R_2 y$, mais unique relation vérifiée parmi les 5.

$x \succ y$ (x se raffine en y) $\Leftrightarrow \neg(y R_2 x)$ et $\neg(y R_3 x)$ et $\neg(y R_4 x)$ et $\neg(y R_5 x)$

$x \multimap y$ (x permet y) $\Leftrightarrow \neg(x R_1 y)$ et $\neg(x R_2 y)$ et $\neg(y R_5 x)$

$x < y$ (x précède y) $\Leftrightarrow \neg(x R_2 y)$ et $\neg(x R_5 y)$ et $\neg(y R_3 x)$ et $\neg(y R_4 x)$ et $\neg(y R_5 x)$

$x \rightarrow y$ (x implique y , avant ou après) $\Leftrightarrow \neg(x R_4 y)$ et $\neg(x R_5 y)$

$x \rightarrow y$ (x oblige ou appelle y)

$\Leftrightarrow x \multimap y$ et $x \succcurlyeq y$ et $\neg(x R_4 y)$

$\Leftrightarrow x R_3 y$, mais unique relation parmi les 5

$x \updownarrow y$ (x et y divergent, ne peuvent avoir lieu en même temps)

$\Leftrightarrow x R_5 y$, mais vérifie cette unique relation parmi les 5.

Une «clade» sera une partie formée d'éléments deux à deux divergents.

On montre alors qu'étant donnés x et x' distincts avec x' raffinant x , il existe toujours une situation notée $x \setminus x'$ raffinant x et réalisant la fusion de tous les raffinements de x qui divergent de x' . On dit que x' et $x \setminus x'$ décomposent x .

Annexe 3

Langage Fril

A.3.1. Rappel sur le langage Prolog

Prolog est un langage déclaratif [Colmerauer 85] d'une conception totalement différente des langages classiques. Programmer en Prolog, c'est énoncer une suite de faits et de règles puis poser des questions.

Tout programme Prolog constitue un petit système-expert qui va fonctionner en «chaînage-arrière» ou «induction», c'est-à-dire qui va tester les hypothèses pour prouver une conclusion.

Soit une base de règles BR constituée d'une part de règles sans prémisses : des faits comme : (frères Caïn Abel) ou non nécessairement clos comme : (égal X X), et d'autre part de règles sous forme de clauses de Horn comme :

(père X Y) et (père Y Z) \Rightarrow (grand-père X Z) que l'on écrira dans le sens de la réécriture, la conclusion en premier, celle-ci devant s'effacer afin d'être remplacée par les hypothèses : C si H₁ et H₂ et ... et H_n.

Une liste de clauses Q étant écrites, l'interpréteur va alors chercher à unifier les différentes propositions élémentaires (faits) constituant le but Q avec la conclusion de chaque règle. Pour cela, il n'y a pas de distinction dans la base de clauses entre les faits et les règles, la conclusion est toujours en tête, les prémisses suivent et si elles ne sont pas présentes, c'est que la «règle» est un fait initial, en quelque sorte un axiome.

Si par un jeu de substitutions de variables, une telle unification est possible, alors avec ces mêmes substitutions partout dans Q, cette conclusion est remplacée par les hypothèses qui pourraient l'entraîner. C'est «l'effacement» et la constitution d'une «résolvante».

En Prolog une clause (Q si P₁ et P₂ et ... et P_n) s'interprète comme : pour prouver Q, il faut prouver P₁, prouver P₂ etc... Dans les principales implémentations du langage Prolog, la syntaxe est : Q :- P₁ , P₂ , ... , P_n.

Le point virgule, note le «ou» Q :- P ; R. étant équivalent aux deux clauses successives Q :- P. et Q :- R.

Les commentaires, comme en Fril, sont encadrés par /* ... */.

Prouver signifie «effacer» (unification avec un fait). Prolog ne se contente pas de fournir une telle «preuve», c'est-à-dire une instanciation ad hoc des variables, mais va les donner toutes, c'est en cela que l'on parle de non-déterminisme.

Il se pose alors un problème pour la négation, le «faux» est remplacé par le concept de «non prouvable».

Prolog fonctionne grâce au principe de résolution : un programme est une «forme normale» c'est à dire une conjonction de «clauses de Horn». Chaque clause de Horn est une disjonction $Q \vee P_1 \vee P_2, \dots, \vee P_n$ de littéraux (faits ou négations de faits) dont au plus un peut être positif, ce qui revient à dire que chaque clause est une proposition de la forme $P_1 \wedge P_2, \dots, \wedge P_n \rightarrow Q$. Le principe est que le programme est équivalent à sa "résolvante» obtenue en effaçant le fait p de la clause C et le littéral $\neg p$ de C' et en remplaçant C et C' par la disjonction obtenue ainsi, et ceci chaque fois qu'un tel couple (p, $\neg p$) est décelable.

EXEMPLE D'UNE AGENCE MATRIMONIALE

On établit un programme comme une liste de faits «bruts» qui vont être traduits au moyen de prédicats que l'on décide d'appeler «caractéristiques», «gouts», «recherche» ... et quelques règles définissant «être bien assortis», soit 23 clauses.

caracteristiques (alfred, homme, grand, brun, mur).

/* tout ce qui commence par une minuscule est constant */

caracteristiques (victor homme moyen blond, jeune).

caracteristiques (hector, homme, petit, brun, mur).

caracteristiques (irma, femme, moyenne, blonde, moyen).

caracteristiques (rosa, femme, petite, blonde, jeune).

caracteristiques (olga, femme, petite, brune, mur).

gouts (alfred, class, aventure, velo).

gouts (victor, pop, sf, ski).

gouts (hector, jazz, polar, ski).

gouts (irma , class, aventure, velo).

gouts (rosa , pop, sf, S) :- dif (S, boxe).

/* le sport est tout sauf de la boxe pour Rosa */

gouts (olga, M, aventure, velo). /* M peut être instancié par n'importe quoi */

```

recherche (alfred, grande, rousse, jeune). /* les majuscules sont des variables */
recherche (victor, T, blonde, jeune) :- dif (T, grande).
/* Victor ne veut pas de grande */
recherche (hector, petite, blonde, moyen).
recherche (irma, grand, brun, moyen).
recherche (rosa, moyen, blond, jeune).
recherche (olga, moyen, brun, mur). /* C'est tout pour les faits bruts */
dif (X, X) :- fail.
dif (X, Y).
/*ces deux clauses ne peuvent pas être permutées, elles définissent «être différents»*/
convient (X, Y) :- caracteristiques (X, homme, T1, C1, A1),
caracteristiques (Y, femme, T2, C2, A2),
recherche (X, T2, C2, A2), recherche (Y, T1, C1, A1).
memesgouts (X, Y) :- gouts (X, M, L, S), gouts (Y, M, L, S).
assortis (X, Y) :- convient (X, Y), memesgouts (X, Y).
/* Ne pas oublier que la conclusion est en premier */

```

Le programme étant achevé, on lance une recherche par la question :
assortis (X, Y).
La réponse que donnera Prolog est X = victor et Y = rosa
On pourra vérifier à la main qu'il n'y a pas d'autres solutions.

EXEMPLE DES TOURS DE HANOÏ

Il s'agit de déplacer une tour formée de N disques de diamètres échelonnés (le plus petit en haut), d'un emplacement dit gauche à un autre dit droite, en se servant d'un troisième (milieu). La règle est de ne jamais placer un disque sur un autre dont le diamètre serait plus petit.
Trois clauses suffisent avec le prédicat "mouv" qui traduit récursivement ce qu'il faut faire comme mouvements et le prédicat "hanoi" associé au nombre de disques, qui permet de démarrer.

```

mouv (1, A, _, C) :- write ("transport de ", A, " sur ", C) , !.
mouv (N, A, B, C) :- K is N-1, mouv (K, A, C, B),
write ("transport de ", A, " sur ", C), mouv (K, B, A, C).
hanoi (N) :- mouv (N, gauche, milieu, droite).

```

L'explication est transparente, pour déplacer les N disques de A sur C, il faut déplacer les N-1 premiers sur B, puis le dernier sur C, et enfin les N-1 de B sur C. (Il y a 2^{N-1} transports) Remarque "write" n'accepte qu'un seul argument en C-prolog, il faut donc en écrire plusieurs, et se servir de "nl" (retour à la ligne).

EFFACER UN ÉLÉMENT DANS UNE LISTE

Les listes d'objets abstraits s'écrivent en Prolog par énumération [a, b, c, ...] ou en faisant appel à une liste L, alors [a, b | L] désignera par exemple la liste débutant par a et b suivis de tous les éléments de L. La concaténation de deux listes est l'enchaînement des deux.

On construit «eff (X, L, R)» où R est la liste L dans laquelle la première occurrence de X est effacée.

```

eff (_, [], []).
eff (A, [A | L], L).
eff (A, [B | L], [B | M]) :- eff (A, L, M).

```

Mais attention le ";" donne les autres solutions ainsi "eff (a, [a, b, a], X)." donne X = [b, a] puis X = [a, b]. Il faut rajouter la prémisse "coupure" dans les deux premières clauses.

Avec eff2 pour effacer toutes les occurrences de X dans L, on remplace la seconde clause par "eff2 (A, [A | L], M) :- eff2 (A, L, M)." Mais là aussi "eff2 (a, [b, a, b, a], X)." donnerait X = [b, b] puis X = [b, a, b] puis X = [b, b, a] d'où la nécessité de la coupure.

Grâce à eff3, on efface X à tous les niveaux de L.

```

eff3 (_, X, X).
eff3 (X, [X | L], M) :- eff3 (X, L, M).
eff3 (X, [Y | L], [Z | M]) :- eff3 (X, Y, Z), eff3 (X, L, M).

```

EXEMPLE DE LA CONCATÉNATION DE DEUX LISTES

Deux clauses vont suffire: conc ([], L, L).
conc ([X | L], M, [X | N]) :- conc (L, M, N).

Avec ce programme très réduit, il est possible de l'interroger sous forme fonctionnelle, par exemple en demandant «conc ([a], [b, c], X)». En ce cas Prolog répondra X = [a, b, c]. Si on demande à présent une résolution d'équation «conc ([a], X, [a, b, c, d])», on obtiendra X = [b, c, d], mais pour «conc ([a], X, [b, c])» il n'y aura pas de solution. Pour «conc ([a], X, Y)», il y a une infinité de solutions, enfin, pour «conc ([a], [b], [a, b])», la réponse est booléenne.

Maintenant, à la question «conc (U, V, [a, b])» dans laquelle U et V sont les variables demandées, Prolog va fournir une première solution U = [] et V = [a, b], puis les autres si l'utilisateur le désire.

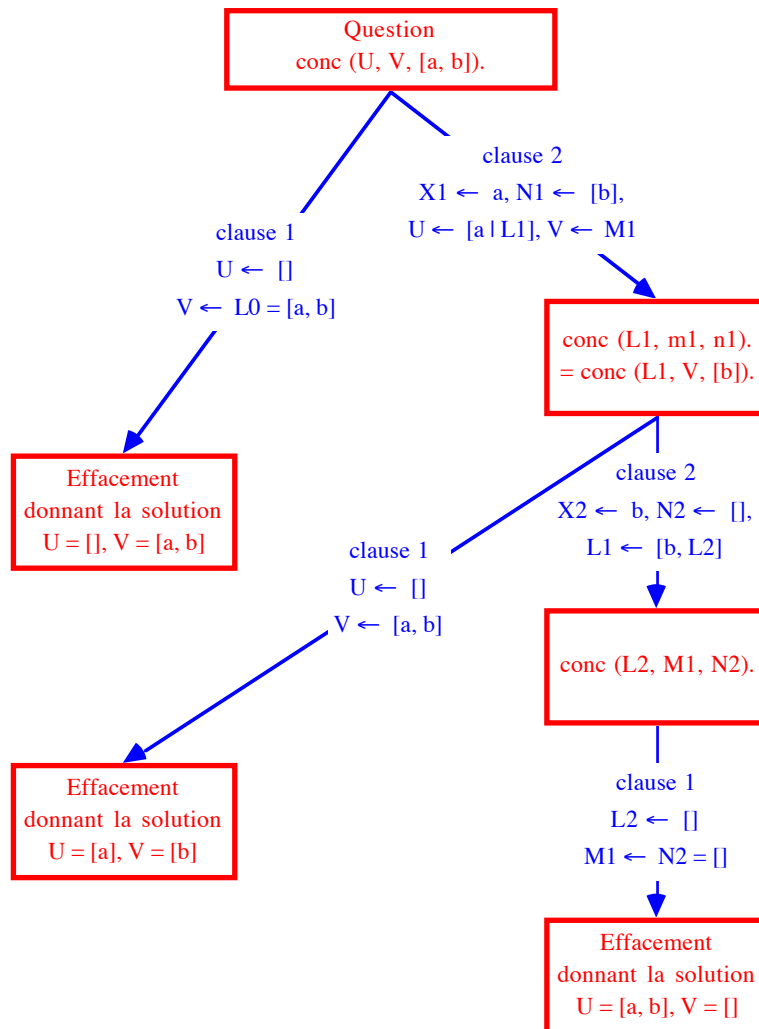


Figure A.3.1 Arbre de résolution du programme «conc» à partir de la question initiale.

A.3.2. Langage Fril (Fuzzy Relational Language de Baldwin)

Fril est un langage déclaratif comme Prolog, cependant, contrairement à Prolog l'arborescence complète est toujours explorée, car toutes les preuves d'une même conclusion vont être confrontées [Baldwin 90].

Dans Fril, chaque fait est donné avec un «support» (n, p). Ce support est considéré dans l'esprit des auteurs comme l'intervalle pouvant contenir la probabilité, il s'agit donc des notions de probabilités basse et haute parfois notées L et U (chapitre 2).

Chaque règle $P \Rightarrow Q$ est donnée avec le support (a, b) de l'événement conditionnel Q/P. Si un deuxième support (c, d) est donné, c'est celui de $\neg P \Rightarrow Q$, donc un intervalle pour la probabilité conditionnelle $p(Q/\neg P)$.

En termes de probabilités cela se justifie par $a \leq p(Q/P) \leq b$ et $c \leq p(Q/\neg P) \leq d$.

NÉGATION

La relation entre les supports de P et $\neg P$ est nécessairement $\neg(n, p) = (1-p, 1-n)$

CONSISTANCE D'UNE BASE DE CONNAISSANCES

Prenons l'exemple de probabilités $p(A/Q) = 0,5$ $p(A/\neg Q) = 0,4$ $p(A/S) = 0,8$ $p(A/\neg S) = 0,4$ $p(Q) = 0,7$ $p(S) = 0,175$ alors $p(A) = p(A/Q)p(Q) + p(A/\neg Q)p(\neg Q) = 0,47$ et le même calcul avec S donnant 0,47. Si cela n'avait pas été le cas la base aurait été dite inconsistante.

CONJONCTION ET DISJONCTION

Si dans un corps de prémisses, deux faits F_1 et F_2 de supports respectifs (n_1, p_1) et (n_2, p_2) se présentent, on suppose par défaut que ces faits sont indépendants et donc « F_1 et F_2 » aura dans ce système le support $(n_1 n_2, p_1 p_2)$ conformément à la logique probabiliste. « F_1 ou F_2 » aura le support $(n_1 + n_2 - n_1 n_2, p_1 + p_2 - p_1 p_2)$

Dans le cas où la dépendance est précisée la conjonction peut être calculée par :

$L(P \text{ et } Q) = \max(0, L(P) + L(Q) - 1)$ et $U(P \text{ et } Q) = \min(U(P), U(Q))$ et la disjonction par $L(P \text{ ou } Q) = \max(L(P), L(Q))$ et $U(P \text{ ou } Q) = \min(1, U(P) + U(Q))$, en suivant un panachage de Zadeh et Lukasiewicz.

Etant donnée une règle $P \Rightarrow Q$ connue avec les couples (a, b) et (c, d) et si l'hypothèse P est connue avec le support (x, y), l'opération consistant à calculer le support (X, Y) de Q est inspirée, pour $P \Rightarrow Q$, de la formule de Bayes dite formule des probabilités totales $p(Q) = p(Q/P)p(P) + p(Q/\neg P)p(\neg P)$.

Le calcul de (X, Y) pour la conclusion Q se fait donc :

$$X = \min\{p(Q/P)p(P) + p(Q/\neg P)p(\neg P)\} = \min \{az + c(1-z)\} \text{ en notant } z = p(P)$$

en regardant simplement le min de la fonction $z \in [x, y] \rightarrow az - cz + c$, dans le cas de $a \leq c$, le min a lieu pour $z = y$, c'est $X = ay + c(1-y)$ sinon $X = ax + c(1-x)$

De manière analogue $Y = by + d(1-y)$ sinon $Y = bx + d(1-x)$

Remarque : dans le cas où un seul support est donné $(X, Y) = (ax, bx + 1 - x)$ on remarque que y n'intervient pas. Ainsi il suffit que la règle $P \Rightarrow Q$ soit donnée avec un support (a, 1) pour qu'une hypothèse P (x, y) même mauvaise infère une conclusion Q (ax, 1).

La valeur (X, Y) attribuée à la conclusion Q lorsque P vaut (x, y) est :

$$X = \text{si } a \leq c \text{ alors } ay + c(1-y) \text{ sinon } ax + c(1-x)$$

$$\text{et : } Y = \text{si } b \leq d \text{ alors } bx + d(1-x) \text{ sinon } by + d(1-y)$$

Si seulement (a, b) est fourni, on a $(X, Y) = (ax, bx + 1 - x)$

REMARQUE

Dans le dernier cas, la valeur y n'a pas d'importance, ainsi par exemple si la prémisse est connue avec le support (0.4 0.8) et la règle avec (0.9 0.9), la conclusion aura (0.36 0.96) et il suffit que b soit 1 (règle pouvant être jusqu'à certaine) pour que la conclusion ait une probabilité supérieure 1. Il serait d'ailleurs difficile de prendre une autre position.

AGRÉGATION

1 Premier cas, l'intersection : si P se trouve avoir les supports (n1, p1) et (n2, p2) par deux preuves différentes, alors l'interpréteur Fril donne le support (max(n1, n2), min(p1, p2)) mais sous réserve que $n \leq p$. C'est la règle d'intersection notée «inter» des supports utilisée par défaut, elle suppose la base de règles consistante.

2 Sinon on utilise la règle de Dempster (chapitre 2), si cela est précisé, tous les faits intervenant dans un prédicat dit «de Dempster» sont supposés indépendants. Deux preuves de P correspondent à des points de vue indépendants P1, P2 de supports (L1, U1), (L2, U2)

On a donc $L = (L1 + L2 - L1L2 - c)/(1-c)$ et $U = U1U2/(1-c)$ avec le coefficient de conflit $c = L1(1-U2) + L2(1-U1)$

3 Troisième cas : par souci de mieux traduire les problèmes concrets où une conclusion peut très souvent être acquise par différentes voies, Fril prévoit la syntaxe suivante :

(conclusion IF (evlog S(x) (x1 is A1) w1 ; (x2 is A2) w2 ; ; (xn is An) wn)) : ((n1 p1) (n2 p2))

Règle dans laquelle les différentes hypothèses éventuellement structurées comme (and (or P Q) (not R)) exactes ou floues, sont affectées de poids w_i et séparées par des point-virgules ; notant «ou bien» comme en Prolog, la somme des poids $\sum w_i$ devant être 1. La fonction S réalise une sorte «d'accentuation» vers une valeur de vérité plus proche de 0 ou de 1, plus simple que la règle de Dempster.

Si (a_i, b_i) est le support de (x_i is A_i), plus généralement de la prémisse P_i, alors le support de la conclusion est calculé par l'application de la règle de Jeffrey sur (S($\sum w_i a_i$), S($\sum w_i b_i$)).

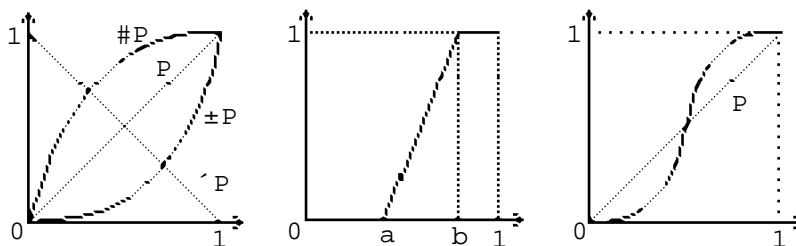


Figure A.3.2 Si P représente une valeur de vérité, on a (à gauche) les représentations de nonP, de «très»P souvent évoquée comme P² et de «plus ou moins»P par \sqrt{P} . Les fonctions S nommées «filtre» (centrale) proposées par [Baldwin, Martin 94] sont du type ci-dessous avec a et b égaux à 0.5 / 0.8 pour «most», 0.7 / 1 pour «very», 0 / 0.5 pour «fairly» ou encore 0 / 1 pour «trueline». A droite l'opérateur de Dempster.

REMARQUES PRATIQUES

Les constantes de Fril peuvent débiter par une majuscule ou non comme Jean ou jean, et les variables sont les majuscules isolées ou les mots débutant par au moins 2 majuscules J, J1, JEan, JEAN, ...

Un prédicat prédéfini (supp_query X S) établit un lien entre un fait X et son support S, on peut donc accéder aux valeurs par exemple (supp_query X (n p)) puis formuler sur n et p. Ainsi ((supp—query ((predicat X)) (X Y)) (p (X + Y) / 2)) réalisera une impression «p» de la demi-somme des probabilités basse et haute.

Les clauses s'écrivent dans une fenêtre d'édition (ainsi éventuellement que la question «qs»... «qh»... ne donnant pas les supports). Puis «reload selection» lance l'interpréteur.

EXEMPLE INTRODUCTIF DE BALDWIN

Une boîte contient 6 boules rouges et 4 bleues, 8 sur les 10 sont solides.
 On ne peut donc donner exactement $p(\text{solide} / \text{rouge})$ mais un encadrement. Les données seront :
 ((rouge X) : (0.6 0.6))
 ((solide X) (rouge X) : ((0.67 1) (0.5 1)))
 Si on demande «qs ((solide uneballe)) on aura le calcul (0.402 1), «uneballe» désignant une constante.
 Maintenant, avec des prédicats «solide», «rouge», «grand», «voisin», et des constantes a, b, on a la base de règles :

- R1 ((solide a) : (1 1))
 - R2 ((solide b) : (0 1))
 - R3 ((rouge a) : (0.8 0.9))
 - R4 ((rouge b) : (0 0))
 - R5 ((grand a) : (0.85 1))
 - R6 ((grand X) (solide X) : (0.6 0.9) (0.3 0.5))
 - R7 ((voisin X Y) (grand X) (rouge Y) : ((0.9 1) (0 0.1)))
- On demande «qs (voisin X Y)»

En posant ce but (voisin A B), les solutions sont comme en Prolog, vues comme les instantiations des variables le long des chemins amenant à un effacement des clauses. Pour chaque solution, son support est calculé en remontant des feuilles vers la racine de l'arbre. Une première solution est trouvée $A = a, B = a$ avec deux preuves en partant des supports (0.612, 0.91) une autre avec (0.612, 0.829), l'agrégation se fait ici par intersection. Toutes les solutions sont calculées, et on obtient $A = B = a$ (0.612, 0.829) puis $A = a, B = b$ (0, 0.1), $A = b, B = a$ (0.216, 0.829) enfin $A = B = b$ (0, 0.1).
 A titre d'exercice, on pourra recalculer les valeurs obtenues 0.216...

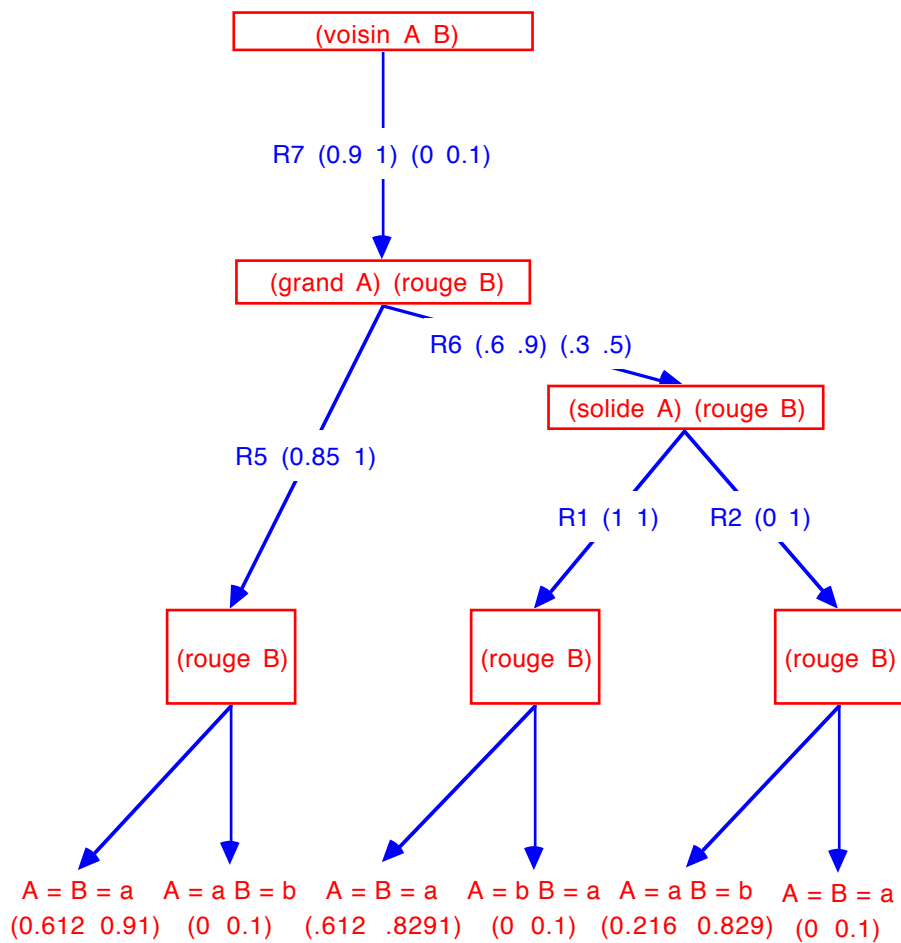


Figure A.3.3 Arbre de résolution de l'exemple de Baldwin.

EXEMPLE DE BASE DE CONNAISSANCES

- ((aime Alexandre Johan) : (0.5 0.8))
- ((aime Alexandre Julien) : (0.8 1))
- ((aime Julien Alexandre) : (0.7 0.9))
- ((freres Valerian Alexandre) : (1 1))
- ((amis X X) : (0 0) /* Une façon de rendre compte de la négation */
- ((amis X Y) (aime X Y) (aime Y X) : (0.7 1))
- ((amis X Z) (freres X Y) (amis Y Z) : (0.3 0.7))
- /* Difficulté comme en Prolog de définir une relation symétrique et transitive */
- /* On pose une question */
- qs ((amis Valerian B))
- ((aime Johan Alexandre) : (0.6 0.8))
- ((aime Laureline Valerian) : (0.6 1))
- ((aime Valerian Laureline) : (0.8 0.9))

```

/* Voilà les réponses */
((amis Valerian Valerian)) : (0 0)          /* antireflexivité */
((amis Valerian Laureline)) : (0.336 1)
((amis Valerian Alexandre)) : (0 1)
((amis Valerian Johan)) : (0.063 0.937)
((amis Valerian Julien)) : (0.1176 0.8824)
no (more) solutions

```

```

/* On souhaite une «transitivité» de l'amitié, mais attention aux appels récursifs infinis avec «amis» seul*/
((relation X Z) (amis X Y) (amis Y Z)) : (0.5 0.7)
qs ((relation A B))
((relation Alexandre Alexandre)) : (0.076832 0.953901)
((relation Johan Johan)) : (0.02205 0.98677)
((relation Johan Julien)) : (0.04116 0.975304)
((relation Julien Johan)) : (0.04116 0.975304)
((relation Julien Julien)) : (0.076832 0.953901)
no (more) solutions
/* On voit rapidement une incertitude presque totale pour ces 5 derniers résultats */

```

TRAITEMENT DES ENSEMBLES FLOUS EN FRIL

On les définit par une suite de couples $x : \mu(x)$, l'interpréteur fera alors des interpolations linéaires et prendra les deux valeurs des extrêmes pour tout ce qui est inférieur ou supérieur à ces extrêmes. Ainsi «grand» est-il défini ci-dessous par un triangle, et «trèsgrand» par une marche.

La confrontation d'une assignation floue A' de la variable X face à une référence floue A , si A représente un prédicat tel que «grand», de la proposition « X est A » est calculée par le support (n p) :

$p = \sup \min(\mu_A, \mu_{A'})$ comme cela est fait habituellement et :

$n = 1 - p(X \text{ est non } A) = 1 - \sup \min(\mu_A, 1 - \mu_{A'}) = \inf \max(1 - \mu_{A'}, \mu_A)$ contrairement à [Dubois, Prade 85].

```

(grand [165:0 175:1 185:0])
(moyen [160:0 170:1 175:0])
(gros [60:0 75:1 90:0])
(tresgros [80:0 95:1]) /* ou bien [80:0 95:1 lim:1] qui signifie jusqu'à l'infini */
((taille Max grand))
((taille Luc moyen))
((taille Fred 72))

```

```

qs ((taille X grand))          /* On pose un but */
/* Le schéma des triangles explique les supports, les trois réponses sont */
((taille Max [165:0 175:1 185:0])) : (0.5 1)
((taille Luc [165:0 175:1 185:0])) : (0.25 0.666666)
((taille Fred [165:0 175:1 185:0])) : (0 0)

```

A titre d'exercice, en définissant (petit [0:0 4:1 7:1 9:0]) et donnant le fait (salaire Max [2:0 3:1 9:0]), la réponse pour (salaire Max petit) sera (0.6 0.9), faire le dessin.

DÉFINITIONS RÉCURSIVES DES ENSEMBLES FLOUS

Une façon intéressante de définir récursivement des ensembles flous est de dire que si H_1 est une grande taille, alors $H_0 = H_1 - 1$ l'est encore, mais un peu moins, d'où un support (0.9 1) par exemple à cette règle.

```

((grand H)(less 175 H)!)
/* définition par une contrainte, essayer sans la coupure ! ferait boucler indéfiniment avec la seconde clause */

```

```

((grand H0) (sum H0 1 H1)(grand H1)) : (0.9 1) /* définition récursive */
((petit H) (less H 160) (!))
((petit H0) (sum H1 1 H0) (petit H1)) : (0.9 1)

```

```

/* voici les questions */          /* et voilà les réponses */
qs ((grand 175))                    ((grand 175)) : (0.9 1) yes
qs ((grand 172))                    ((grand 172)) : (0.6561 1) yes
qs ((grand 170))                    ((grand 170)) : (0.531441 1) yes
qs ((grand 160))                    ((grand 160)) : (0.185302 1) yes
qs ((petit 165))                    ((petit 165)) : (0.531441 1) yes
qs ((petit 160))                    ((petit 160)) : (0.9 1) yes
qs ((petit 155))                    ((petit 155)) : (1 1) yes

```

```

/* on rajoute des prédicats de Dempster en assumant la non-contradiction des règles */
dempster grand_ht
((grand_ht H) (grand H)) : (1 1)
((grand_ht H) (petit H)) : (0 0)
dempster petit_ht
((petit_ht H) (petit H)) : (1 1)
((petit_ht H) (grand H)) : (0 0)

```


Questions	Réponses
qs ((grand_ht 170))	((grand_ht 170)) : (0.437658 0.823531) yes
qs ((petit_ht 165))	((petit_ht 165)) : (0.437658 0.823531) yes
qs ((grand_ht 160))	((grand_ht 160)) : (0.022239 0.120015) yes
qs ((petit_ht 155))	((petit_ht 155)) : (1 1) yes

A titre d'exercice, rajouter ((poids X gros) (taille X grand)) (0.7 0.9) et demander (gros Max) etc.

OPÉRATIONS

Soient deux ensembles flous A et B définis comme suit :

(A [0:0 6:1 9:0])

(B [2:0 3:1 4:0])

qs (+ A B) /* réponse */ [2:0 9:1 13:0]

qs (- A B) /* réponse */ [-4:0 3:1 7:0]

qh ((sum [0:0 1:1 2:0] [3:0 4:1 5:1 7:0] X))

/* réponse approximative) */

X = [3:0 5:1 6:1 9:0]

Plusieurs applications sont en projet, on verra notamment [Baldwin 96].

Annexe 4

Langage Mvl

A.4.1. Rappels sur les treillis et bitreillis

Un treillis est une structure mathématique définie soit à partir d'un triplet (T, \vee, \wedge) où T est un ensemble et \vee, \wedge deux lois binaires internes idempotentes commutatives associatives et vérifiant l'«absorption»

$$[\forall x, y \quad x \wedge (x \vee y) = x \quad \text{et} \quad x \vee (x \wedge y) = x]$$

ou alors, un treillis est défini à partir de (T, \leq) où \leq est une relation d'ordre réticulée $[\forall x, y \exists \text{ inf et sup de } \{x, y\}]$, les opérations sont alors $x \wedge y = \text{inf } \{x, y\}$ et $x \vee y = \text{sup } \{x, y\}$. Réciproquement pour un treillis (T, \vee, \wedge) , on parlera de l'ordre associé défini par $x < y \Leftrightarrow x \wedge y = x$.

ORDRE INDUCTIF : ordre tel que toute chaîne est majorée, tout ensemble inductif possède un max (axiome du choix version Zorn).

ORDRE NOETHÉRIEN : ordre où toute chaîne admet un max (toute suite croissante est stationnaire). Pour la décroissance, on parle d'ordre artinien.

BON ORDRE : ensemble ordonné où toute partie possède un plus petit élément. L'ordre est alors total et tout élément possède un successeur.

TREILLIS COMPLET : ensemble ordonné tel que toute partie possède un «inf» et un «sup».

TREILLIS DISTRIBUTIF : \vee, \wedge sont mutuellement distributives

TREILLIS MODULAIRE : treillis tel que l'on ait :

$\forall a, b, c \quad a < c \Rightarrow a \vee (b \wedge c) \geq (a \vee b) \wedge c$, (tout treillis distributif est modulaire).

TREILLIS COMPLÉMENTÉ : treillis possédant deux éléments (notés 0 et 1) tels que $\forall x \exists \neg x$ (appelé complément de x) $x \wedge \neg x = 0$ et $x \vee \neg x = 1$

0 est alors minimal pour \leq , neutre pour \wedge et absorbant pour \vee

1 est maximal pour \leq , absorbant pour \wedge et neutre pour \vee

TREILLIS DE BOOLE : treillis distributif et complémenté, c'est alors avec la loi définie par $a + b = (a \wedge \neg b) \vee (\neg a \wedge b)$ et \wedge , un anneau unitaire commutatif et idempotent.

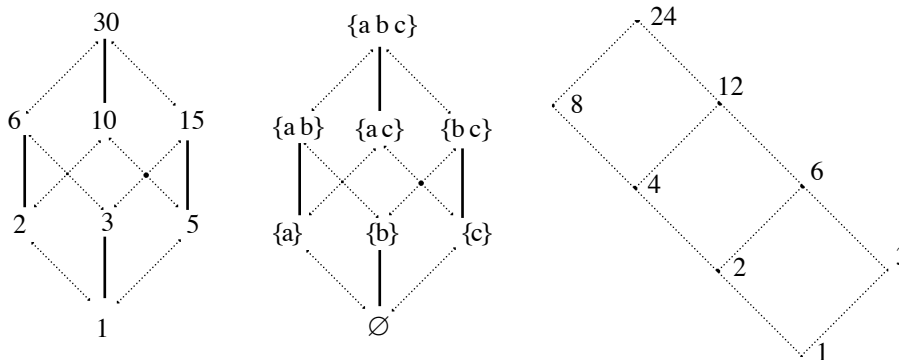


Figure A.4.1 Exemples d'algèbres de Boole Soit l'ensemble des diviseurs de 30 pour la relation de divisibilité, 1 est le minimum, 30 le maximum, \vee, \wedge sont les opérations pgcd, ppcm (figure gauche). Soit, maintenant l'ensemble des parties de $\{a, b, c\}$ pour la relation d'inclusion, \vee, \wedge sont les opérateurs \cap, \cup (figure centrale). On voit l'isomorphisme de ces deux structures, différentes par ailleurs du treillis des diviseurs de 24 (à droite).

IDÉAL I : c'est une partie stable pour \vee et permise pour \wedge (section commençante de \leq stable par \vee). Un idéal premier est un idéal maximal $\forall x \quad x \in I$ ou $\neg x \in I$

FILTRE F : partie stable pour \wedge et permise pour \vee . Un ultrafiltre est un filtre maximal $\forall x \quad x \in F$ ou $\neg x \in F$

ALGÈBRE MONADIQUE [Halmos 62] : algèbre de Boole possédant une application notée \exists normalisée, $\exists 0 = 0$ extensive $p \leq \exists p$ et quasi-multiplicative $\exists (p \wedge \exists q) = \exists p \wedge \exists q$

Cette structure est utilisée en logique formelle pour asseoir les notions de modèle et d'interprétations d'une théorie :

On montre que \exists est idempotente, croissante et réalise un morphisme pour \vee .

Un idéal monadique étant un idéal stable par \exists , une logique monadique est la donnée d'une algèbre monadique et d'un idéal monadique I .

I est appelé l'ensemble des antithèses et $T = \{p / \neg p \in I\}$ qui est un filtre, est l'ensemble des thèses.

(B, I) est complète $\Leftrightarrow I$ maximal $\forall p, p \in I$ ou $\neg p \in I$

(B, I) est consistant $\Leftrightarrow I$ propre $\forall p, \text{non}(p \in I \text{ et } \neg p \in I)$

Un modèle est une logique monadique où I est vide, et une interprétation est un homomorphisme d'une algèbre monadique dans un modèle, qui est telle que l'image de I soit 0 .

BITREILLIS

$(B, \wedge, \vee, \cdot, +, \neg)$ est un bitreillis si et seulement (B, \wedge, \vee) et $(B, \cdot, +)$ sont des treillis et si \neg est une involution de B réalisant un isomorphisme de \wedge sur \vee de \vee sur \wedge et de \cdot et $+$ sur eux mêmes.

La «négation» est la symétrie par rapport à la droite (I TF) dans les figures suivantes. Certains auteurs ont introduit une négation faible comme symétrie par rapport à la bissectrice.

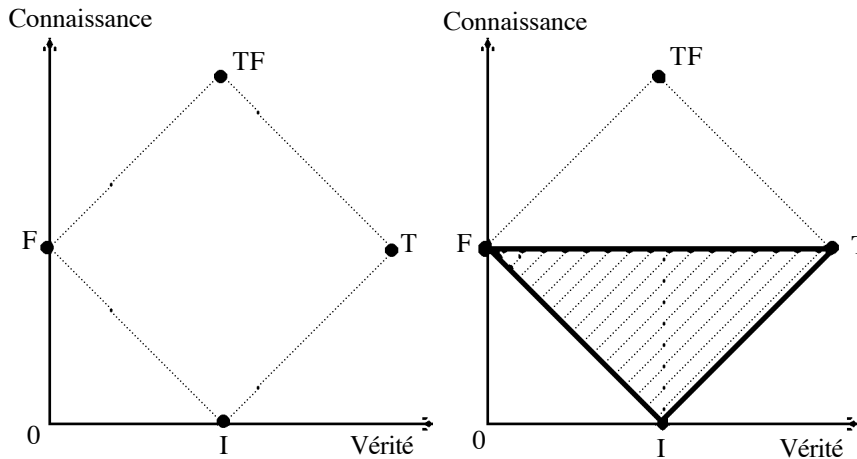


Figure A.4.2 Le plus petit bitreillis non trivial est celui de la logique booléenne [Belnap 77] (il comprend 4 éléments) : T (vrai), F (faux), I (incertain), TF (contradiction).

La théorie des possibilités correspond aux points des segments FI et IT, les logiques multivaluées à ceux du segment FT, et la théorie des probabilités haute et basse à la zone hachurée (I et TF faisant partie à chaque fois du bitreillis).

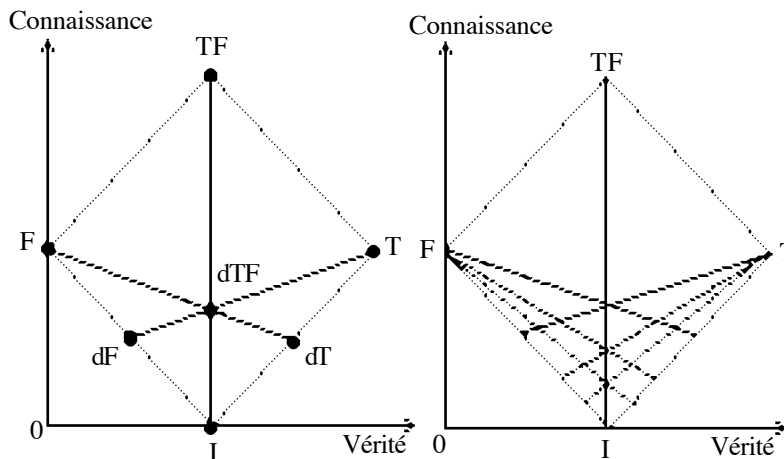


Figure A.4.3 En ce qui concerne la logique des défauts elle peut être représentée par le bitreillis $\{T, F, I, TF, dT, dF, dTF\}$ à gauche, et la logique dénombrable des défauts ou défauts hiérarchiques à droite par 3 chaînes notées T, dT, d2T, d3T F, dF, d2F, d3F,..... et TF, dTF, d2TF, d3TF,.....

Dans le système Mv1 [Ginsberg 88, 91], [Desvignes 92] ces quatre opérations sont appelées respectivement «and», «or», «dot» et «plus».

Les ordres partiels associés seront notés \leq_t et \leq_k afin d'ordonner suivant une vérité croissante et une connaissance croissante.

BITREILLIS BASÉ SUR DES MONDES

Soit W un ensemble de «mondes» (intuitivement des univers possibles), on dira que la proposition P du langage L a la valeur de vérité (U, V) avec U, V inclus dans W , si P est vrai de façon booléenne dans chaque élément de U et faux dans chacun de ceux de V . Pour un tel couple (U, V) , un monde de $U \cap V$ est dit inconsistant alors qu'un monde de $W - (U \cup V)$ est dit indéterminé.

En posant : $(U, V) \leq_t (U', V') \Leftrightarrow U \subseteq U'$ et $V' \subseteq V$

et : $(U, V) \leq_k (U', V') \Leftrightarrow U \subseteq U' \text{ et } V \subseteq V'$

on obtient un bitreillis $BW = P(U)*P(W)$ dans lequel les éléments distingués sont $F = (\emptyset, W)$ pour le faux, $T = (W, \emptyset)$, $I = (\emptyset, \emptyset)$ l'indéterminé et enfin $TF = (W, W)$ la contradiction.

Cette façon de construire un bitreillis est particulièrement adaptée aux systèmes de maintien de la cohérence ATMS [Doyle 79], [DeKleer 86].

Dans un tel bitreillis, à chaque proposition P , on associera le couple $(J_p, J_{\neg p})$ formés pour J_p par les «justifications» de P , ainsi si $Q \rightarrow P$ et $R \& S \rightarrow P$ sont des règles, on aura $J_p = (\{Q\}, \{R, S\})$ formé par deux justifications possibles amenant à P , et pour $J_{\neg p}$, par les justifications de $\neg P$.

Les relations d'ordres sont donc :

$J \preceq J'$ (J moins général que J') $\Leftrightarrow J \subseteq J'$

$P_1 \leq_t P_2$ (plus de vérité pour P_2) $\Leftrightarrow (J_1, K_1) \leq_t (J_2, K_2) \Leftrightarrow J_1 \preceq J_2 \text{ et } K_1 \preceq K_2$

$P_1 \leq_k P_2$ (plus d'information pour P_2) $\Leftrightarrow (J_1, K_1) \leq_k (J_2, K_2)$

$\Leftrightarrow J_1 \preceq J_2 \text{ et } K_1 \preceq K_2$

Les hypothèses qui ont une justification vide sont des éléments minimaux pour \leq_t , les axiomes et leurs déductions sont maximaux pour \leq_t .

Plus précisément, se plaçant donc dans la logique des propositions ordinaire avec ($T = \text{vrai}$, $F = \text{faux}$, $I = \text{incertain}$, $TF = \text{contradictoire}$), (il ne sera pas question de la négation c'est à dire qu'au lieu d'avoir systématiquement $[A \text{ et non } (A) \rightarrow TF]$, on aura seulement des règles particulières $[A \text{ et } B \rightarrow TF]$). Dans le fonctionnement du système, on met des axiomes (des faits ayant une justification vide), des hypothèses (des faits H ayant (H) comme justification), et une base de règles sous forme de clauses de Horn $[A_1 \text{ et } A_2 \text{ et } \dots \rightarrow A]$ est une règle, alors cela signifie que $(A_1 \ A_2 \ \dots)$ sera une justification pour A .

Un environnement est une conjonction de propositions, il est dit consistant s'il n'entraîne pas de contradiction.

Une justification est une liste d'hypothèses.

Un label $L(A)$ pour une proposition A est une disjonction de justifications (les différents «mondes» qui permettent de déduire A)

$L(A)$ est minimal si aucune des justifications présentes n'est sur-ensemble d'une autre (ce sont les plus générales et il n'y a pas de redondance).

$L(A)$ est consistant si chaque justification présente est consistante.

$L(A)$ est complet si tout environnement consistant où A peut être déduit est un sur-ensemble d'une des justifications déjà présente. (aucune justification n'est oubliée)

$L(A)$ est correct (ou sain ou encore bien construit) si chaque justification présente entraîne A .

Le rôle du système est de mettre à jour la liste de justifications de chaque assertion, en en faisant un label sain, consistant minimal et complet, et de le recalculer chaque fois que l'on rajoute une justification c'est à dire une nouvelle règle. Par exemple, si on rajoute (A, B) comme justification de TF , cela signifie que A et B ne peuvent plus coexister.

A.4.2. Le langage de programmation MVL (Multi Valued Logics)

Il s'agit, comme pour FRIL, d'un Prolog fonctionnant essentiellement en chaînage-arrière, mais où les "valeurs de vérité" des propositions peuvent être de type très divers y compris calculées par exemple I ou $(T \text{ et } F)$ pour la valeur dT , et le mode d'inférence choisi selon différentes logiques non classiques.

Afin d'illustrer le fonctionnement de MVL, nous reproduisons l'exemple du manuel de référence sans trop insister sur les détails de syntaxe et de codage un peu particuliers. Signalons seulement que les clauses, comme en Prolog, se présentent comme clauses de Horn avec la conclusion en tête et parenthésées, sous la forme :

$(\leftarrow C \ P_1 \ P_2 \ \dots \ P_n)$ et pour le chaînage arrière, la conclusion C étant remplacée par les prémisses $P_1 \ \dots$ ou bien $(\Rightarrow C \ P_1 \ P_2 \ \dots \ P_n)$ pour le chaînage avant non utilisé ici, ou encore $(\text{if } C \ P)$ pour une règle $P \rightarrow C$ et sa contraposée $\neg C \rightarrow \neg P$.

: value true ; permet de définir comme vraies les clauses qui vont suivre
(idn P) ; fonction renvoyant la valeur de vérité de la proposition P
(bcs 'P) ; permet de poser un but, de procéder à la résolution de P
(contents) ; donne la liste des clauses chargées
(denotes n) ; renvoie la clause de numéro n
(stash 'P : value false) ; permet de rajouter une clause P avec sa valeur («assert»)
(load-logic) ; permet au début d'une utilisation, de choisir la logique (premier ordre, ATMS, défauts, défauts hiérarchiques, temporelle, probabiliste)

EXEMPLE 1 LE SYSTÈME ATMS (MAINTIEN DE LA COHÉRENCE)

Les valeurs de vérité déclarées sont alors les constantes f , t , i , tf , ou bien des couples de la forme $(J_p, J_{\neg p})$ ou encore nil pour des faits non justifiés (les hypothèses).

On définit alors le programme P_0 comme la suite des clauses définissant un petit arbre généalogique :

(femme pam) (femme pat) (femme ann)
(femme liz) (homme tom) (homme bob)
(homme jim) (parent pam bob) (parent tom bob)
(parent bob ann) (parent tom liz) (parent bob pat)
(parent pat jim)
(\leftarrow (enfant x y) (parent y x))
(\leftarrow (mere x y) (parent x y) (femme x))
(\leftarrow (ancetre x y) (parent x y))
(\leftarrow (ancetre x y) (parent x z) (ancetre z y))

```

(<= (diff x y) (not (equal x y))) ; ce sont ici les fonctions lisp «not» et «equal» qui sont utilisées
(<= (soeur x y) (parent p x) (parent p y) (femme x) (diff x y))
(<= (frere x y) (parent p x) (parent p y) (homme x) (diff x y))
(if (pauvre x) (not (riche x)))
(if (vivant x) (not (mort x)))
(if (triste x) (not (heureux x)))

```

```

(riche bob) ; un nouveau fait assez simple
(<= (riche x) (enfant x y) (riche y))

```

```

: value nil ; les règles A, B, C, D sont de contexte vide, ce sont les hypothèses
(not (riche x)) ; règle A, tout le monde est pauvre a priori
(not (mort x)) ; règle B
(<= (heureux x) (riche x) (vivant x)) ; règle C
(<= (triste x) (mort y) (vivant x)
 (or (frere x y) (soeur x y) (ancetre x y) (ancetre y x))) ; règle D

```

On pose la question (riche bob), la réponse fournie par le système est ((nil) (A)) ce qui est conforme.

On rajoute alors le fait (mort bob)

En reposant la question (riche ann), la réponse est encore ((nil) (A)).

pour la question (triste ann), la réponse ((B D) (B C)) signifie qu'avec les hypothèses B et D, Ann doit être triste, mais qu'avec les hypothèses B et C, elle est heureuse.

EXEMPLE 2 LA LOGIQUE DES DÉFAUTS

Les seules valeurs de vérité sont ici les constantes f, t, i, tf, df, dt, dtf.

On prend le même programme P₀ avec la valeur t, et les mêmes règles A, B, C, D mais avec la valeur dt (vrai par défaut).

A la question (riche x), le système répond x = bob avec la valeur vraie.

A la question (not (riche x)), il donne toutes les personnes (y compris bob) avec la valeur dt (il ne va donc pas vérifier le contraire d'un but automatiquement).

A la question (heureux x), la réponse est x = bob (valeur dt)

On introduit alors (mort bob) et on demande (riche x), la réponse est x = pat, ann, bob : vrai (ce dernier reste riche tant qu'une nouvelle clause n'est pas présente pour dire qu'un mort n'est plus riche).

La demande (heureux x) fournit x = pat, ann (valeur dt) à cause de C.

La demande (not (heureux x)) donne x = pat, ann, pam, liz, jim, tom (valeur dt) par la règle D.

EXEMPLE 3 LA LOGIQUE DES DÉFAUTS HIÉRARCHIQUES

Les valeurs sont les mêmes avec dt₁ df₁ et dt₂, df₂ ...

Les règles de P₀ sont les mêmes (valeur vraie) A, B, D sont données avec la valeur dt₁, enfin C est donnée avec dt₂.

A la question (heureux x), la réponse est toujours x = bob (valeur dt₂)

On introduit alors (mort bob) et on demande (heureux x) qui ne donne aucune réponse car «triste» devient prioritaire. Enfin la demande (not (heureux x)) donne encore x = pat, ann, pam, liz, jim, tom (valeur dt₁), on a donc une connaissance plus fine qu'avec l'exemple 2.

EXEMPLE 4 LOGIQUE TEMPORELLE

Les valeurs sont les constantes ainsi que des entiers par exemple (true-at 4) signifiant vrai à l'instant 4. Ainsi peut-on déclarer une même proposition P (true-at t₀) puis (false-at t₁), un prédicat prédéfini permet d'indiquer la persistance d'une valeur de vérité dans l'intervalle de t₀ à t₁ - 1 c'est (propagate P).

(delay t P) est vrai en t + t₀ dès que P est vrai en t₀

Le programme est cette fois constitué par les mêmes clauses vraies de P₀ sauf les deux dernières et par :

```

: value true
(heritage bob 6) ; dans 6 mois, puis la règle A est remplacée par :
(<= (riche x) (enfant x y) (propagate (riche y)) (heritage y) (delay t (mort y)))
(<= (heureux x) (riche x) (propagate (vivant x))) ; règle B
(<= (triste x) (propagate (vivant x)) (propagate (mort y)) (mere x y))
; règle C la mère reste toujours triste de la mort de son enfant

(<= (triste x) (propagate (vivant x)) (or (frere x y) (soeur x y) (ancetre x y)
 (ancetre y x))
 (or (mort y) (delay 1 (mort y)) (delay 2 (mort y))
 (delay 3 (mort y)) (delay 4 (mort y))))
; règle D les autres membres de la famille observent 5 mois de deuil
: value (true-at 0)
(riche bob)
(vivant x)
: value (true-at 3)
; la proposition est différée 3 mois après, on aurait pu dire aussi : value (true-at 6) (heritage bob)
(mort bob)

```

: value (true-at 20)
(mort jim)

La question (triste pat) fournit les valeurs de vérité étalées dans le temps :

(0 i) (3 t) (6 tf) (7 t) (8 i) (20 t)

En effet en 6 on a le deuil plus l'héritage qui amène une contradiction de sentiments pour Pat, à 7 mois le deuil dure toujours, à 8 mois rien ne permet comme en 0 de se prononcer, et à 20 mois la mort de son fils Jim provoque à nouveau la tristesse.

LOGIQUE PROBABILISTE

Pour une règle A et B et C et D et ... \rightarrow Q la probabilité est calculée :

par $p(Q) = p(A).p(B).p(C)...$

Si $P \rightarrow Q$ et $R \rightarrow Q$ sont 2 règles de même conclusion alors l'indépendance est supposée et $p(Q) = p(P) + p(R) - p(P).p(R)$. Mais la version actuelle ne possède pas $p(\neg A) = 1 - p(A)$.

Annexe 5

Les réseaux bayésiens

A.5.1. Rappels de probabilités

Dans un univers Ω (la plupart du temps fini) dont les éléments sont appelés «éventualités» et les parties «événements», une probabilité a été définie (chapitre 2) comme une mesure donnant à chaque partie (événement) une valeur entre 0 et 1.

Les axiomes d'une telle mesure de probabilité sont, dans le cas fini, les données $p(\emptyset) = 0$, $p(\Omega) = 1$ (les événements impossible et certain), p est croissante pour l'inclusion $A \subseteq B \Rightarrow p(A) \leq p(B)$, ainsi que l'additivité :

$$\forall A \forall B \quad A \cap B = \emptyset \Rightarrow p(A \cup B) = p(A) + p(B)$$

Lorsqu'un événement A est réalisé, on appelle probabilité conditionnelle (connaissant A) la mesure définie par $p_A(B) = p(B / A) = p(A \cap B) / p(A)$.

On peut vérifier qu'il s'agit d'une autre mesure de probabilité et on a la relation :

$$p(A \cap B) = p(A / B)p(B) \text{ assortie de la définition :}$$

$$\begin{aligned} A \text{ et } B \text{ événements indépendants} &\Leftrightarrow p(A \cap B) = p(A).p(B) \\ &\Leftrightarrow p_A(B) = p(B) \Leftrightarrow p_B(A) = p(A) \end{aligned}$$

Formule de Bayes, elle permet de relier une probabilité conditionnelle à la probabilité conditionnelle écrite en sens inverse :

$$p_A(B) = \frac{p_B(A) p(B)}{p_B(A) p(B) + p_{\overline{B}}(A) p(\overline{B})} \text{ et pour une partition de } \Omega \text{ en } B_1, \dots, B_n,$$

$$\text{on a } p(B_k / A) = \frac{p(A / B_k) p(B_k)}{p(A / B_1) p(B_1) + p(A / B_2) p(B_2) + \dots + p(A / B_n) p(B_n)}$$

Une variable aléatoire numérique X est une fonction de Ω dans \mathbb{R} dont les probabilités associées aux différentes valeurs que peut prendre X , constituent la loi de probabilité.

La loi vérifie $\sum p(X = x) = 1$ pour toutes les valeurs x que peut prendre X , formule dans laquelle « $X = x$ » désigne en fait l'ensemble des éventualités $\omega \in \Omega$ pour lesquelles $X(\omega) = x$.

Deux variables aléatoires X et Y définies sur le même univers sont dites :

X, Y indépendantes

$$\Leftrightarrow \forall x \text{ valeur de } X, \forall y \text{ valeur de } Y, p(X = x \text{ et } Y = y) = p(X = x).p(Y = y)$$

A.5.2. Définition d'un réseau bayésien

Un réseau bayésien est un graphe où les noeuds sont des faits (ces faits seront plus précisément associés à des variables aléatoires, par exemple au noeud associé à la variable X , les faits pourront être $X = x_1$ ou $X = x_2, \dots$ ou $X = x_n$, si X peut prendre n valeurs), on pourra donc confondre dans la suite les noeuds et les variables aléatoires, quant aux arcs, ce sont des relations de causalité. Sur l'arc reliant le fait A au fait B , il y aura donc une pondération qui sera la probabilité conditionnelle notée $p(B / A)$.

Les réseaux bayésiens sont donc des graphes orientés généralement acycliques, c'est à dire sans boucles. Le problème est maintenant de décrire un algorithme de mise à jour des probabilités marginales reliant les noeuds en fonction de l'information disponible à chaque instant. En effet, lorsqu'une nouvelle information arrive en un noeud du graphe, les probabilités de chacun des états doivent être recalculés conformément à l'axiomatique des probabilités et en supposant toujours l'indépendance entre frères et soeurs dans l'arbre. C'est cette propagation de l'information que nous allons devoir détailler dans ce qui suit.

NOTATIONS

Le produit tensoriel (termes à termes) de deux vecteurs est noté \otimes , il est défini par :

$$(a, b, c) \otimes (a', b', c') = (aa', bb', cc').$$

On considère qu'en chaque noeud se trouve une variable aléatoire discrète et que l'état de cette variable X est décrit par le vecteur V_X des probabilités d'avoir différentes valeurs pour X . Pour le fait $X = x$, nous noterons $p(x)$ sa probabilité.

On note e_X^- toute l'information disponible chez les descendants de X , c'est à dire dans le sous-arbre de racine X , et e_X^+ , celle qui est disponible dans le reste du graphe.

Dans le cas simple d'une arborescence où chaque noeud X n'a qu'un parent U , l'état sera décrit par la matrice $M_{X/U}$ à n lignes et p colonnes dans la mesure où U possède p valeurs.

Chaque élément de cette matrice est donc de la forme $p(X = x_i / U = u_j)$

PROPAGATION DE LA CONFIANCE DANS UN RÉSEAU BAYÉSIEN

Les calculs vont se propager de telle façon qu'ils vont être traités localement en chaque noeud.

CALCUL A PRIORI DANS LE CAS D'UNE ARBORESCENCE

A la racine U , la donnée du vecteur V_U des probabilités $p(u_j)$ et la donnée de la matrice $M_{X/U}$ pour chaque fils X de U , permet de calculer le vecteur V_X des probabilités de X par simple produit matriciel $V_X = M_{X/U} \cdot V_U$ c'est à dire pour chaque ligne $p(x_i) = \sum_{1 \leq j \leq p} p(x_i / u_j) \cdot p(u_j)$.

En chaque noeud X on peut donc stocker les valeurs $p(x_i)$. Si X et Y sont deux fils du même père U , on supposera toujours X et Y indépendantes.

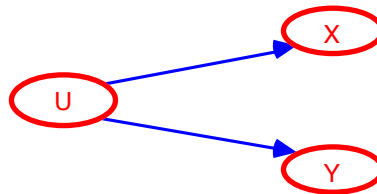


Figure A.5.1 Propagation d'un père U vers ses fils X et Y .

Ce type de calcul permet de faire descendre l'information dans toute l'arborescence.

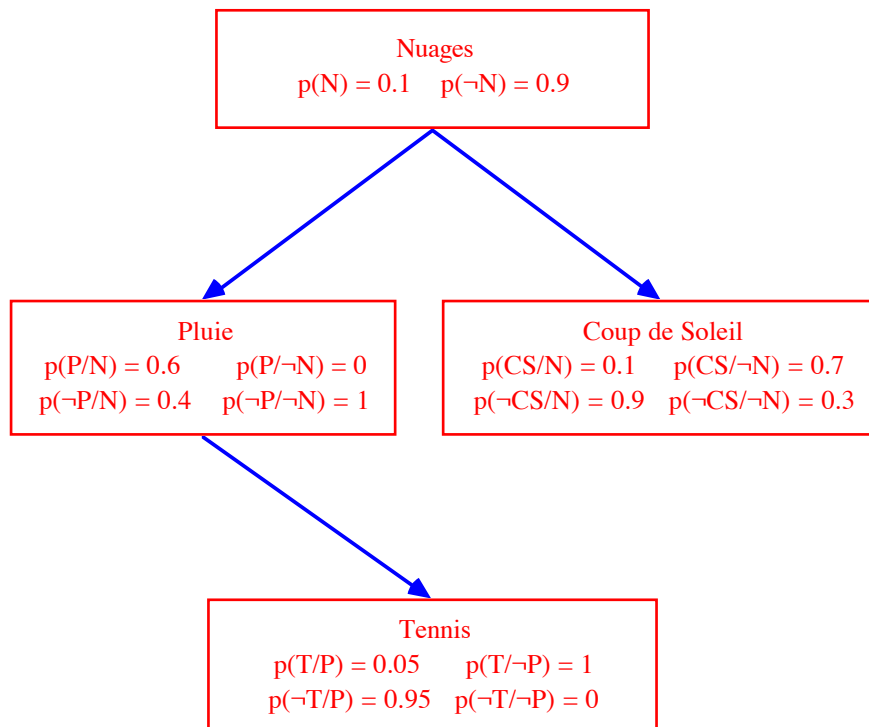


Figure A.5.2 Exemple de [Person 91], P = pluie, N = nuages, T = tennis, CS = coup de soleil.

CAS D'UN ARBRE QUELCONQUE

(X a par exemple deux parents U et V), on va toujours supposer l'indépendance de ces deux variables. Si U possède les valeurs u_1, \dots, u_p et V les valeurs v_1, \dots, v_k , alors la matrice des probabilités conditionnelles de X , notée $M_{X/U, V}$ comportera n lignes et pk colonnes, et le calcul descendant sera identique quoique le vecteur V_U, V_V doit être de dimension pk (c'est le produit extérieur des messages des pères).

CALCUL A POSTERIORI

Considérons à présent le cas d'une arborescence où le père est unique et dans la chaîne U père de X, père de Y, au cas où on apprend que $Y = y_0$ pour le noeud Y, on va remonter l'information pour actualiser la loi de X puis celle de U.

La formule de Bayes donne $p(x / y_0) = \alpha p(y_0 / x) \cdot p(x)$ où α est un facteur multiplicatif tel que ait $\sum p(x / y_0) = 1$ pour toutes les valeurs x de X.



Figure A.5.3 Propagation le long d'une chaîne.

En termes vectoriels, on va définir le «message ascendant de Y vers X» qui est noté par $\Lambda_{y_0 X}$, comme le vecteur ligne formé par les $p(y_0 / x)$.

On a donc $V_{X/y_0} = \alpha \Lambda_{y_0 X} \otimes V_X$.

Intéressons nous maintenant au père U de X, pour chaque valeur u possible pour U, nous avons : $p(u / y_0) = \alpha p(y_0 / u) p(u)$

$$= \alpha \left[\sum_{\text{valeur de X}} p(y_0 / x, u) p(x / u) \right] p(u)$$

$$= \alpha \left[\sum p(y_0 / x) p(x / u) \right] p(u)$$

car Y est indépendante de U connaissant X.

Nous avons donc plus généralement, pour une donnée de probabilités sur Y, une matrice Λ_{YX} formée par les lignes $\Lambda_{y_i X}$, dite «message ascendant» contenant les probabilités conditionnelles $p(y / x)$ et qui permet le calcul $V_{X/Y} = \alpha \Lambda_{YX} \otimes V_Y$. Le calcul précédent s'exprime alors par le produit matriciel $\Lambda_{XU} = \Lambda_{YX} \cdot M_{X/U}$, c'est dire que le message ascendant de chaque étape ne nécessite que la matrice locale et le message ascendant qui précède.

CIRCULATIONS DESCENDANTE ET ASCENDANTE COMBINÉES DANS UNE CHAÎNE

Le message ascendant se propage jusqu'au bout de la chaîne. C'est seulement après cela que se propage le message descendant, et qu'en chaque sommet X, le vecteur V_X sera obtenu par produit tensoriel des deux messages reçus $\prod X$ et Λ_X , puis normalisation.

CAS D'UNE ARBORESCENCE OU D'UN ARBRE QUELCONQUE

La mise à jour d'un noeud X doit se faire en consultant le père U et les fils Y_i .

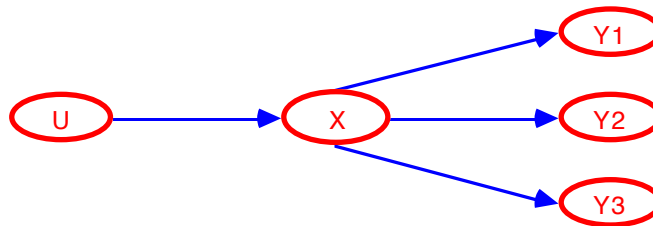


Figure A.5.4 Propagation dans une arborescence.

Posons $\prod X$ le vecteur des probabilités $p(x / e_{X+})$ c'est à dire connaissant tout ce qui est situé au dessus de X il est calculé comme expliqué ci-dessus (même dans le cas d'un arbre quelconque) $M_{X/U} \cdot V_U$

Posons d'autre part Λ_X celui des probabilités $p(e_{X-} / x)$ de l'état de toute la descendance de X connaissant les différentes valeurs de X.

Le calcul est le même que précédemment, mais supposant les fils Y_i de X indépendants, on peut faire, grâce à cette hypothèse, le produit des probabilités pour avoir $p(e_{X-} / x) = \alpha \prod p(Y_i = y_{i0} / x) \cdot p(x)$ où α est un facteur multiplicatif tel que $\sum_{\text{valeur de X}} p(x / e_{X-}) = 1$.

On donne donc à X le vecteur Λ_X des probabilités $p(e_{X-} / x)$ calculé par le produit termes à termes des messages des fils :

$$\Lambda_X = \Lambda_{Y1X} \otimes \Lambda_{Y2X} \otimes \dots \otimes \Lambda_{Y_i X} \otimes \dots$$

On aura en définitive le vecteur d'état de X connaissant le reste du réseau :

$$[p(x / e_{X+}, e_{X-})] = \alpha \Lambda_X \otimes \prod X$$

Donc encore une fois, le produit tensoriel des messages ascendant et descendant, suivi d'une normalisation.

REMARQUES : Si X est la racine de l'arbre, $\prod X = V_X$ et si X est une feuille, Λ_X est le vecteur formé par des 1 uniquement. Lorsqu'une information est du type $X = x_0$, alors Λ_X est calculé comme le produit tensoriel de ses fils et d'un «fils fictif» formé par le vecteur nul sauf de composante 1 pour l'indice de x_0 , donc, seul cet indice aura une valeur non nulle dans le résultat.

PROPAGATION DES INFORMATIONS

Un message ascendant est calculé à partir de la donnée locale et des messages ascendants antérieurs, par contre un message descendant est calculé à partir du ou des messages amont, qui eux reçoivent d'autres messages ascendant. La règle de propagation des informations est donc la suivante : lorsque une information sur les probabilités marginales arrive en un ou plusieurs noeuds de l'arbre, ce sont les messages ascendants qui sont calculés et envoyés en premier. Les messages descendants sont ensuite propagés depuis la racine.

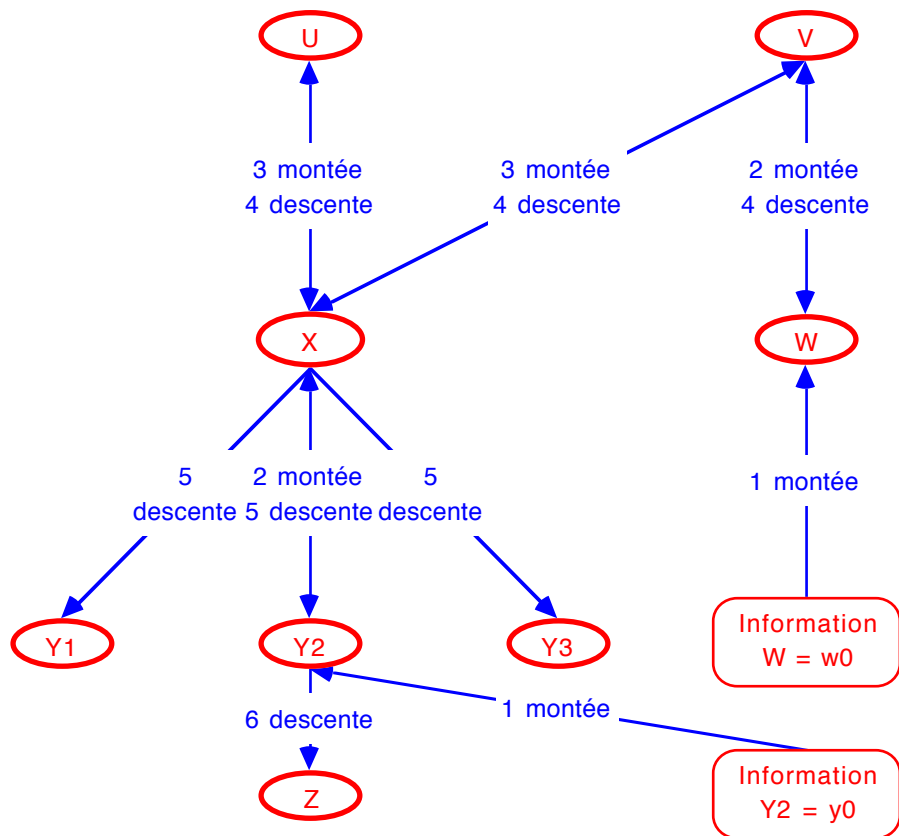


Figure A.5.5 Exemple de propagation des informations dans l'arbre suivant, deux informations portant sur Y_2 et W , vont se propager dans un ordre marqué par les étapes de 1 à 6.

EXEMPLE DE [Cooper 84], [Pearl 88]

On suppose établies les probabilités conditionnelles figurant dans l'arbre ci-dessous où la présence de tumeur au cerveau et d'un taux élevé de calcium dans le sang est causé par un cancer, ces deux faits pouvant entraîner le coma.

Dans l'hypothèse où les maux de tête sont observés, l'information remonte en T puis en C pour redescendre en Ca et en T, enfin en Co et en M.

Après circulation des messages et normalisation, les probabilités stabilisées sont $p(\text{Cancer}) = 0.097$, $p(\text{Calcium}) = 0.094$, $p(\text{Tumeur}) = 0.031$ et $p(\text{Coma}) = 0$.

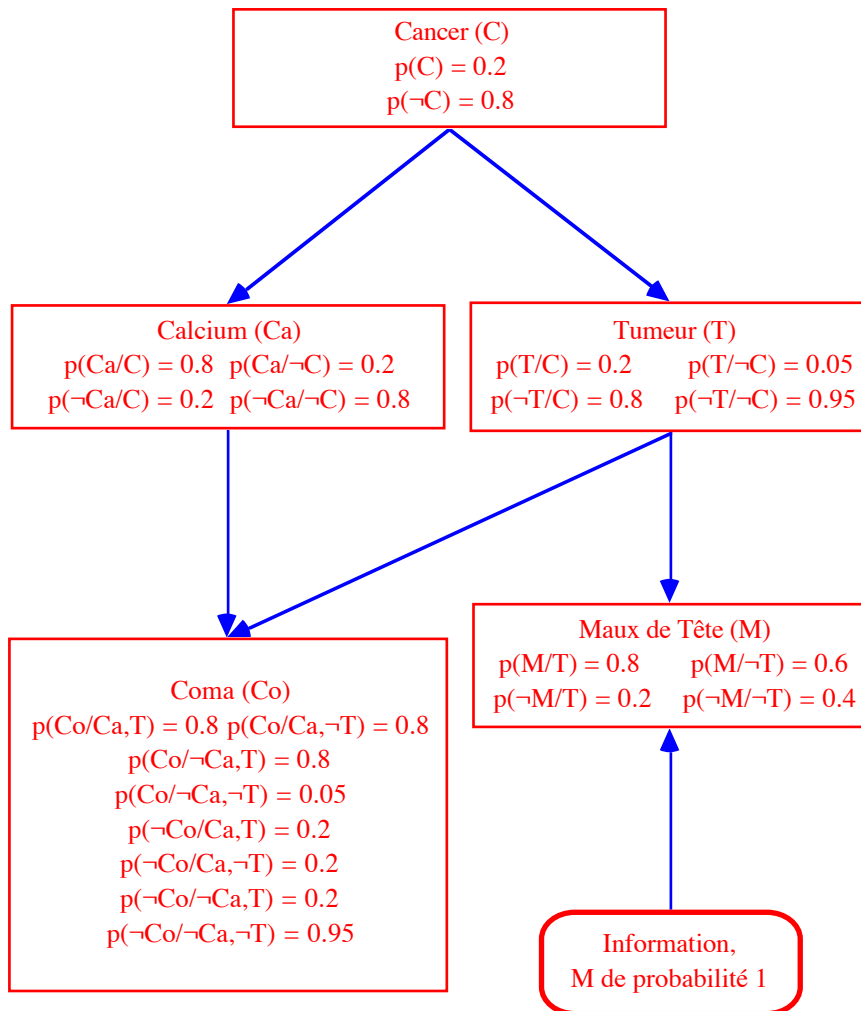


Figure A.5.6 Exemple de Cooper.

CAS DE GRAPHES AVEC CIRCUITS Dans le cas où il y a des boucles, le processus précédent peut être infini, le problème est donc bien plus compliqué, aussi a-t-il été proposé de partitionner le graphe en arbres de façon permanente (ce qui supprime le problème) ou de façon temporaire.

Annexe 6

Aperçu d'automatique

Un système, dont on doit contrôler un paramètre de «sortie» y, se présente suivant le schéma ci-dessous.

Les entrées d'un contrôleur sont la «sortie» réelle et l'erreur, ou bien l'erreur et la variation de cette erreur, ou plus généralement plusieurs paramètres physiques. Quant à sa sortie, elle constitue l'entrée du système à asservir.

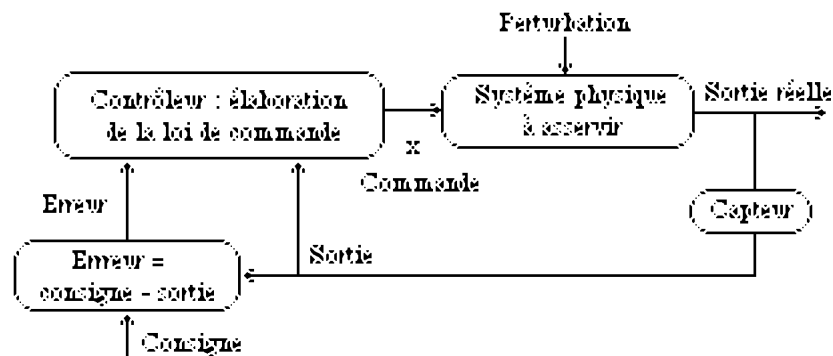


Figure A.6.1 Schéma d'un contrôleur en boucle fermée.

Les contrôleurs sont des systèmes calculant une commande notée x ou u en fonction de l'erreur e et de sa dérivée, leurs types sont (t est le temps) :

P	proportionnel	$x(t) = ke(t)$	k est appelé un "gain"
I	intégral	$x(t) = \int_0^t e(u)/T \, du$	risque d'oscillations
D	dérivé	$x(t) = T' \, de(t)/dt$	on amortissement
PI		$x(t) = ke(t) + \int_0^t e(u)/T \, du$	Laplace $X(p) = (k + 1/Tp)E(p)$
PD		$x(t) = ke(t) + T' \, de(t)/dt$	$X(p) = (k + T'p)E(p)$
PID		$x(t) = ke(t) + \int_0^t e(u)/T \, du + T' \, de(t)/dt$	$X(p) = (k + 1/Tp + T'p)E(p)$

Les problèmes auxquels sont confrontés ces contrôleurs sont ceux du réglage des paramètres k, T, T' ... afin d'obtenir un compromis entre une réponse rapide mais oscillante voire instable, ou bien trop lente. Ces réglages sont difficiles, tout comme les contrôleurs flous, et les critères d'optimisation à réduire peuvent être l'intégrale de la valeur absolue ou du carré de l'erreur sur un temps plus ou moins long. L'exemple de la consigne en créneaux du chapitre 5 est un exemple industriel typique.

Après mise au point, les PID fonctionnent soit grâce à leur formule (PID numériques), soit en calculant la fonction de transfert en Z à partir d'un échantillon de mesures précédentes (PID discrets). Pour un système, nous notons donc son entrée u qui est la sortie engendrée par le contrôleur, y sa sortie réelle, $x = e$, l'«erreur». Un système est dit linéaire (le contrôle flou rend service précisément dans le cas non linéaire) si on a une relation du type :

$$a_n \partial y(t)/\partial t + \dots + a_1 \partial y(t)/\partial t + a_0 y(t) = b_k \partial u(t)/\partial t + \dots + b_1 \partial u(t)/\partial t + b_0 u(t)$$

En ce cas, grâce à la transformée de Laplace, on peut écrire :

$$H(p) = \frac{Y(p)}{U(p)} = \frac{b_k p^k + \dots + b_1 p + b_0}{a_n p^n + \dots + a_1 p + a_0}$$

Ces systèmes sont dits du premier, second, troisième ... ordre suivant que $n = 1, 2, 3 \dots$ avec $k = 0$. H est la transmittance ou fonction de transfert, son original de Laplace donne y par convolution : $y(t) = (h * u)(t)$, h est la réponse impulsionnelle du système.

En considérant un système à plusieurs dimensions, déterminer le vecteur de commande U, c'est résoudre les équations matricielles $X' = AX + BU$, $Y = CY + DU$. On cherche alors les fonctions de transfert réunies dans une matrice T vérifiant $Y(p) = T(p) \cdot U(p)$ et les transformées de Laplace des deux équations donnent :

$$pX(p) = AX(p) + BU(p) \text{ et } Y(p) = [C(pI - A)^{-1}B + D]U(p) \text{ soit :}$$

$$T(p) = C(pI - A)^{-1}B + D$$

Les valeurs propres de A sont les pôles de T.

Annexe 7

Quelques procédures

NOYAU D'UN CONTRÔLEUR FLOU (SOLUTION AVEC DES RÈGLES FIGÉES À DEUX PRÉMISSSES EN PASCAL)

```
Const   nr = 9; {nr = nb de règles}
type predicat = (N, Z, P, nb, nm, ns, ze, ps, pm, pb, ANY);
type trapeze = record   g, pg, d, pd : integer end;
           {bornes droite et gauches et pentes des trapèzes tous définis dans [-10,10]}
type regle = record pd: real; h1, h2, co : predicat end;
           {chaque règle est de la forme «si dl est h1 et dr est h2 alors u est co»}
type distribution = array [-10 .. 10] of real;
           {représente les possibilités pour -1, -0,9, -0,8 .... 0,9, 1}
type baserel = array [N .. ANY] of trapeze;
type basereg = array [1..NR] of regle;
           {Ces 2 types imposés par pascal dans les en-têtes de procedures}
var BP : baserel; BR : basereg;
procedure initialisation (var BP : baserel; var BR : basereg, pe : integer);
var i : integer;   {pe = pente des trapèzes }
begin
  BP[ANY].g:= -100; BP[ANY].pg:=0; BP[ANY].d:=100; BP[ANY].pd:= 0;
  BP[N].g := -100 ; BP[N].pg := 0 ; BP[N].d := -100 ; BP[N].pd := 200;
  BP[P].g := 100 ; BP[P].pg := 200 ; BP[P].d := 100 ; BP[P].pd := 0;
  BP[Z].g := 0 ; BP[Z].pg := 100 ; BP[Z].d := 0 ; BP[Z].pd := 100;
  BP[NB].g := -100 ; BP[NB].pg := 0 ; BP[NB].d := -100 ; BP[NB].pd := pe ;
  BP[NM].g := -67 ; BP[NM].pg := pe ; BP[NM].d := -67; BP[NM].pd := pe;
  BP[NS].g := -50 ; BP[NS].pg := pe ; BP[NS].d := -50 ; BP[NS].pd := pe ;
  BP[ZE].g := 0 ; BP[ZE].pg := pe ; BP[ZE].d := 0 ; BP[ZE].pd := pe ;
  BP[PS].g := 50 ; BP[PS].pg := pe ; BP[PS].d := 50 ; BP[PS].pd := pe ;
  BP[PM].g := 67 ; BP[PM].pg := pe ; BP[PM].d := 67 ; BP[PM].pd := pe ;
  BP[PB].g := 100 ; BP[PB].pg := pe ; BP[PB].d := 100 ; BP[PB].pd := 0 ;
  { ci-dessous les règles du pendule inversé pe = 33 } for i : 1 to nr do BR[i].pd := 1;
  BR[1].h1 := n; BR[1].h2 := n; BR[1].co := nb; BR[2].h1 := n; BR[2].h2 := z; BR[2].co := nm; BR[3].h1 := n; BR[3].h2 := p; BR[3].co :=
  ze; BR[4].h1 := z; BR[4].h2 := n; BR[4].co := ns; BR[5].h1 := z; BR[5].h2 := z; BR[5].co := ze; BR[6].h1 := z; BR[6].h2 := p; BR[6].co :=
  ps; BR[7].h1 := p; BR[7].h2 := n; BR[7].co := ze; BR[8].h1 := p; BR[8].h2 := z; BR[8].co := pm; BR[9].h1 := p; BR[9].h2 := p; BR[9].co :=
  pb} end;

function ver (x : integer; t : trapeze) : real; {donne son résultat t(x) entre 0 et 1}
begin if x <= t.d then if t.g <= x then ver := 1
      else if t.g - t.pg <= x then ver := 1 - (t.g - x) / (t.pg)
      else ver := 0
      else if x <= t.d + t.pd then ver := 1 - (x - t.d) / (t.pd)
      else ver := 0 end;

function min (a, b : real) : real; begin if a < b then min := a else min := b end;
function conj (n : integer; a, b : real) : real; {n est un numéro d'opérateur}
begin case n of   1 : if a + b < 1 then conj := 0 else conj := a + b - 1 ;
                2, 3 : conj := a*b ; {0 Weber, 1 Luk, 2 et 3 Larsen, 4 Mamdani}
                0, 4, 5 : conj := min(a, b) end end;

function disj (n : integer; a, b : real) : real; {une t-conorme, 2 proba}
begin case n of   1 : disj := min(1, a + b) ;
                2 : disj := a + b - a*b;
                0, 3, 4, 5 : if a < b then disj := b else disj := a {max} end end;

function impl (n : integer; a, b : real) : real; {une implication 0 Weber}
begin case n of   1, 4 : impl := min(a, b); {Mamdani}
                0, 2, 3, 5 : impl := a*b {Larsen 2 ou 3} end end;
```

```

procedure confrontation (n, x, y : integer; R : regle ; var U : distribution ); {construit une distribution U en confrontant 2 valeurs x, y
à une règle R déterminant U, si aucune règle ne s'applique U = 0 et le u sera nul}
  var i : integer; v : real;
  begin v := (R.pd) * conj (n, ver (x, BP[R.h1]), ver (y, BP[R.h2]));
  for i:= -10 to 10 do U[i] := impl (n, v, ver (i*10, BP[R.co])) end;

procedure fuzzy (n, x, y : integer; var U: distribution );
{réalise l'agrégation en examinant les NR règles du tableau BR, x et y dans [-100,100]}
  var UR : distribution; i, r : integer ;
  begin for i := -10 to 10 do U[i] := 0 ;{U est construit suivant disj}
  for r := 1 to NR do begin confrontation (n, x, y, BR[r], UR) ;
    for i:= -10 to 10 do U[i] := disj (n, U[i], UR[i]) end end;

function moyenne (U : distribution) : integer {renvoyé entre -100 et 100} ;
{la distribution est formée par 21 valeurs d'appartenance dans [0, 1]}
  var i : integer; c, s : real;
  begin c := 0; s := 0;
  for i := -10 to 10 do begin s := s + i*U[i] ; c := c + U[i] end;
  if c = 0 then moyenne := 0 else moyenne := round ((10*s) / c) end;

function f (n, x, y : integer) : integer; {renvoyé entre -100 et 100}
{fonction générale calculant le résultat u dans [-100,100] en fonction de x, y}
  var U : distribution; r : integer; s, c, v : real;
  begin if op = 5
    then begin s := 0; c := 0; for r := 1 to nr do
      begin v := (BR[r].pd)*conj(op, ver(x, BP[BR[r].h1]), ver(y, BP[BR[r].h2]));
      s := s + impl(op, v, (BP[BR[r].co].g + BP[BR[r].co].d)/2); c := c + v end;
      if c = 0 then f := 0 else f := round(s / c) end
    else begin fuzzy (n, x, y, U); f := moyenne (U) end end;

```

NOYAU D'UN CONTRÔLEUR FLOU MAMDANI (SOLUTION AVEC DES RÈGLES FIGÉES À DEUX PRÉMISSSES EN LISP)

```

(de histog (TR) (cond; TR trapèze représenté en histogramme de 21 valeurs entre -1 et 1
((eq TR 'ANY) (compte 21 1))
((eq TR 'NUL) (compte 21 0))
(t (let ; compteur, décrement, liste des valeurs, a, b, al, bt est le trapèze TR
(k 1) (d 0.1) (H nil) (a (car (eval TR))) (b (cadr (eval TR))) (al (caddr (eval TR)))
(bt (caddr (eval TR))))
(while (< (+ b bt) k) (set 'H (cons 0 H)) (set 'k (- k d)))
(while (< b k) (set 'H (cons (1+ (divide (- b k) bt)) H)) (set 'k (- k d)))
(while (< a k) (set 'H (cons 1 H)) (set 'k (- k d)))
(while (< (- a al) k) (set 'H (cons (1+ (divide (- k a) al)) H)) (set 'k (- k d)))
(while (< -1 k) (set 'H (cons 0 H)) (set 'k (- k d)))
(cons (if (eq TR 'NB) 1 0) H) )))

(de compte (n v) (if (eq n 0) nil (cons v (compte (1- n) v))))
(setq PRED '(NB NS ZE PS PB) NB '(-1 -1 0 0.5) NS '(-0.5 -0.5 0.5 0.5)
ZE '(0 0 0.5 0.5) PS '(0.5 0.5 0.5 0.5) PB '(1 1 0.5 0) npr 5 ; nb de prédicats
HISTO (mapcar (lambda (x) (cons x (histog x))) PRED) am 1) ; définit les constantes

(de poss (x P) (cond ; donne la fonction d'appartenance de x au prédicat P
((eq P 'NUL) 0)
((eq P 'ANY) 1)
((and (<= x -1) (eq P 'NB)) 1)
((and (<= 1 x) (eq P 'PB)) 1)
(t (apply 'posbis (cons x (eval P))))))
(de posbis (x a b al bt) (cond ; un prédicat trapézoïdal est donné par (a b alpha beta)
(<= x (- a al)) 0)
(< x a) (divide (- (+ x al) a) al))
(<= x b) 1)
(< x (+ b bt)) (divide (- (+ b bt) x) bt))
(t 0)))

(de mini (x y) (if (< x y) x y))
(de maxi (L M) (cond ; réalise la liste composée des max dans les listes L et M
((null L) M)
((null M) L)
((< (car L) (car M)) (cons (car M) (maxi (cdr L) (cdr M))))
(t (cons (car L) (maxi (cdr L) (cdr M))))))
(de troncature (L v) ; L est un histogramme, on renvoie le min avec v
(if (null L) nil (cons (mini (car L) v) (troncature (cdr L) v))))
(de moyenne (L) (mbis L -1 0 0)) ; donne l'abscisse du barycentre
(de mbis (LR k S pr) (if (null LR) (if (eq pr 0) 0 (divide S pr))
(mbis (cdr LR) (+ k 0.1)(+ (* k (car LR)) S) (+ pr (car LR)))))

```

(de cantor (i j) (if (<= (+ i j) (1+ npr)) ; on parcourt le tableau en diagonale
 (+ (1- j) (div (* (- (+ i j) 2) (1- (+ i j))) 2))
 (- (* npr (1+ npr)) i (div (* (- (1+ (* 2 npr)) i j) (- (+ 2 (* 2 npr)) i j)) 2))))

(de fuzzy (C x y) ; x et y sont les entrées; C la base de règles
 (fuzzybis C x y PRED PRED PRED 1 1 nil nil))

(de fuzzybis (C x y PRED Px Py i j R v)
 (cond ; donne le sous ensemble flou max, Px et Py sont les listes de prédicats
 ((null Px) (* am (moyenne R))); am désigne plus loin l'angle maximal
 ((or (null Py) (eq v 0)) (fuzzybis C x y PRED (cdr Px) PRED (1+ i) 1 R nil))
 ((null v) (fuzzybis C x y PRED Px Py i j R (poss x (car Px))))
 (t (fuzzybis C x y PRED Px (cdr Py) i (1+ j)
 (maxi R (troncature (cassoc (nth (cantor i j) C) HISTO)
 (mini v (poss y (car Py)))) v)))

NOYAU D'UN CONTRÔLEUR FLOU (SOLUTION AVEC DES RÈGLES SYMBOLIQUES DE N PRÉMISSSES ET P CONCLUSIONS EN LISP)

Les règles sont du type ((c1 c2 ... cp) p1 p2 pn) pour n entrées et p sorties (que des symboles pour les prémisses et des symboles dont on prendra les sommets, ou bien des nombres en conclusions)

(de repete (x L) (cond ; produit une liste d'autant de fois x qu'il y a d'éléments dans L
 ((listp L) (repete x (length L)))
 ((zerop L) nil)
 (t (cons x (repete x (1- L))))))

(de maxi (a b) (if (< a b) b a))

(de ramene (x a) (cond ; réalise une bijection de [-a, a] dans [-1, 1]
 ((< x (- a)) - 1)
 ((< a x) 1)
 (t (divide x a))))

(de sommet (S) ; donne l'abscisse du sommet (qui va rester fixe) du prédicat (r = 1/2)
 (selectq S (NB -1) (NS -0.5) (ZE 0) (ANY 0) (PS 0.5) (PB 1)))

(de opp (pred) ; donne l'opposé du prédicat pred
 (selectq pred (NB 'PB) (NS 'PS) (ZE 'ZE) (ANY 'ANY) (PS 'NS) (PB 'NB)))

(de coef (E P) ; coef d'application de la règle de liste de prémisses P pour l'entrée E = (e1 e2)
 (coefbis E P nil))

(de coefbis (E P LC) ; LC liste de coefficients
 (if (or (null E) (null P)) (tnorm LC) (coefbis (cdr E) (cdr P)
 (cons (if (eq (car P) 'ANY) 1
 (maxi 0 (- 1 (divide (abs (- (sommet (car P)) (car E))) r)))) LC))))

(de maxi (a b) (if (< a b) b a)) (de mini (a b) (if (< a b) a b))

(de tnorm (L) (if (null L) 1 (minibis (car L) (cdr L)))) ; t-norme de Zadeh

(de minibis (X L) (cond
 ((null L) X)
 ((> X (car L)) (minibis (car L) (cdr L)))
 (t (minibis X (cdr L))))

(de fuzzy (E LR) ; donne une liste en sortie dans [-1, 1]p avec LR liste de règles

(fuzzybis (coef E (cdar LR)) E (caar LR) (repete 0 (caar LR)) 0 (cdr LR)))

(de fuzzybis (co E LC S SC LR) ; c coeff E entrées LC liste des conclusions S sortie, SC somme des coef

(fuzzyter E (ajou co (if (numberp (car LC)) LC (mapcar 'sommet LC)) S) (+ co SC) LR))

(de fuzzyter (E S SC LR) ; E vecteur entrée, S vecteur sortie

(if (null LR) (if (eq SC 0) (repete 0 S) (mapcar (lambda (x) (divide x SC)) S))

(fuzzybis (coef E (cdar LR)) E (caar LR) S SC (cdr LR))))

(de ajou (co L M) ; produit la liste co*L + M terme à terme

(if (or (null L) (null M)) nil (cons (+ (* co (car L)) (car M)) (ajou co (cdr L) (cdr M))))

EVITEMENTS D'OBSTACLES EN LISP

Dans ce programme, (exemple du parcours d'obstacles du chapitre 5) un contrôleur flou est déterminé par la structure (r h R1 R2 R3) où r est la valeur de la demi-base des prédicats (r = 1/2 normalement) et h ≥ 1 leur hauteur, les règles R = (pr deg (c1 c2 ... cp) p1 p2 pn) pour n entrées et p sorties (uniquement des symboles NB NS ZE PS PB ANY pour les prémisses et des symboles dont on prendra les sommets, ou bien des nombres en conclusions.

Exemple de contrôleur ayant deux règles indiquant qu'un obstacle proche nécessite un virage et un ralentissement brusques, et une règle accélératrice si la distance au plus proche obstacle est grande:

(set 'C '(24 0.7 1.15 (0 0 (pb nb) ze ns) (0 0 (nb nb) ze ps) (0 0 (ze ps) pb any)))

On verra un exemple voisin en fin de paragraphe.

La fonction "coef" est modifiée par rapport au paragraphe précédent :

(de coef (E prm r h) ; coef d'application de la liste de prémisses prm pour l'entrée E = (e1 e2)
 (coefbis E prm nil r h))

(de coefbis (E prm LC r h) ; LC liste de coefficients, r demi-base h hauteur
 (if (or (null E) (null prm)) (tnorm LC)
 (coefbis (cdr E) (cdr prm) (cons (if (eq (car prm) 'ANY) 1
 (mini 1 (maxi 0 (* h (- 1 (divide (abs (- (sommet (car prm)) (car E))) r)))))) LC) r h)))

La fonction "fuzzy" donne, suivant la méthode de Sugeno, la liste des sorties dans [-1, 1]^P avec LR liste de règles de même priorité la précédant c.a.d (LR u1 u2 up) où LR a été modifiée, renvoie nil si LR ne s'applique pas.

```
(de fuzzy (E LR r h)
  (fuzzybis nil (car LR) (coef E (cdddar LR) r h) E (caddar LR)
    (repete 0 NC) 0 (cdr LR) r h)) ; LV règles vues, Rc règle courante
(de fuzzybis (LV Rc co E LC S SC LR r h)
; co coeff E les entrées LC liste des conclusions S sortie, SC somme des coef
  (fuzzyter (cons (mcons (car Rc) (+ co (cadr Rc)) (cddr Rc)) LV)
; la règle Rc est modifiée et ajoutée à LV
  E (ajou co (if (numberp (car LC)) LC (mapcar 'sommet LC)) S) (+ co SC) LR r h))
(de fuzzyter (LV E S SC LR r h) ; E vecteur entrée, S vecteur sortie
  (if (null LR) (cons LV (if (eq SC 0) nil ; (repete 0 NC)
    (mapcar (lambda (x) (divide x SC)) S)))
  (fuzzybis LV (car LR) (coef E (cdddar LR) r h) E (caddar LR) S SC (cdr LR) r h)))

(de decision (E C) ; renvoie pour une liste d'entrées E et un contrôleur C = (r h R1 R2 R3 ...)
; le vecteur sortie précédée de C où ont été modifiés les degrés d'application des règles
  (remettre (car C) (cadr C) (decibis E (car C) (cadr C) nil (list (caddr C)) 0 (cdddd C))))
```

```
(de decibis (E r h LV Lpr pr LR)
; LV règles vues non appliquées, Lpr règles de priorité pr, LR reste à voir
  (cond ((null LR) (deciter E r h LV (fuzzy E Lpr r h) pr nil))
        ((equal (caar LR) pr) (decibis E r h LV (cons (car LR) Lpr) pr (cdr LR)))
        (t (deciter E r h LV (fuzzy E Lpr r h) pr LR))))
(de deciter (E r h LV F pr LR) ; F est de la forme (Règles de forces modifiées s1 ... sp)
  (if (null (cdr F)) (if (null LR) (cons (append LV (car F)) (repete 0 (cddr (cdaar F))))
    (decibis E r h (append LV (car F)) nil (caar LR) LR)
    (cons (append (append LV (car F)) LR) (cdr F)))) ; -> (Règles s1 ... sp)
(de remettre (a b L) ; renvoie ((a b X) m n k...) si L = (X m n k ...)
  (cons (mcons a b (car L)) (cdr L)))
```

Fonctions graphiques pour le dessin du champ d'obstacles :

```
(de tan (x) (divide (sin x) (cos x)))
(de degre (a) (truncate (* 180 (/ a pi))))
(de arc (a b c d) (cond ; donne l'arc en radians du vecteur (a, b) (c, d) avec l'horizontale
  ((eq b d) (if (< a c) 0 pi))
  ((eq a c) (if (< b d) (/ pi 2) (- (/ pi 2))))
  ((< c a) (if (< b d) (+ pi (arc c d a b)) (- (arc c d a b) pi)))
  (t (atan (divide (- d b) (- c a))))))
```

```
(de rpi (a) (cond ; représentation dans ]-pi, pi]
  ((< pi a) (rpi (- a (* 2 pi))))
  ((<= a (- pi)) (rpi (+ a (* 2 pi))))
  (t a))
```

```
(de point (u v) (move-to u v) (line-to u v) )
```

```
(de croix (u v) (move-to (1- u) v) (line-to (1+ u) v) (move-to u (1- v))(line-to u (1+ v)))
(de sqr (x) (* x x))
```

```
(de obstacles (n) ; produit une liste de n obstacles aléatoires symbolisés par des points
  (let ((E nil) (r 0) (a 0))
  (repeat n (setq r (distrib ry) a (* pi (divide (random -98 98)99))
    E (cons (list (truncate (+ u0 (* r (cos a)))) (truncate (+ v0 (* r (sin a)))) E))) E))
```

```
(de dessin (E) (if (null E) nil (apply 'point (car E)) (dessin (cdr E))))
(de cercle (x y r) (let ((ag 0) (move-to (+ x r) y) (repeat 32
  (line-to (truncate (+ x (* r (cos (set 'ag (+ ag 0.2)))))) (truncate (+ y (* r (sin ag))))))))
(de radar (u v dir L d a) ; donne le couple (d a) de l'obstacle de L le plus proche de (u v) dans la dir
  (if (null L) (list (ramene d dm) (ramene a am))
  (radar1 u v dir (abs (- u (caar L))) (abs (- v (cadar L))) L d a)))
(de radar1 (u v dir x y L d a)
  (if (or (< dm x) (< dm y)) (radar u v dir (cdr L) d a)
  (radar2 u v dir x y (- (arc u v (caar L) (cadar L)) dir) (cdr L) d a)))
(de radar2 (u v dir x y ag L d a)
  (if (or (< am ag) (< ag (- am))) (radar u v dir L d a)
  (radar3 u v dir (sqrt (+ (* x x) (* y y))) ag L d a)))
(de radar3 (u v dir di ag L d a) (if (< di d) (radar u v dir L di ag) (radar u v dir L d a)))
```

```
(de bord (u v a) ; donne la distance relative à dm, de (u, v) au bord du disque dans la direction a
  (ramene (let ((co (cos a)) (ta (tan a)))
  (if (eq co 0) (- (sqrt (- (sqr ry) (sqr (- u u0)))) (* (sgn a) (- v v0)))
  (let ((del (sqrt (- (* (sqr ry) (1+ (sqr ta))) (sqr (+ (- (* ta (- u u0)) v) v0))))
    (x (- u0 u (* ta (- v v0))))
    (set 'x (- x del)) (if (< (* x co) 0) (set 'x (+ x (* 2 del)))) (* x co)))) dm))
```



```
(de cap (u v dir) (capbis (radar u v dir OB dm 0) (bord u v (+ dir am)) (bord u v (- dir am))))
(de capbis (rad d1 d2) ; renvoie le couple relatif (d a) du plus proche obstacle (bord compris)
  (if (< d1 (car rad)) (if (< d2 d1) (list d2 -1) (list d1 1))
    (if (< d2 (car rad)) (list d2 -1) rad)))
```

La fonction principale "parcours" permet d'effectuer une expérience en prenant des mesures.

```
(de parcours (C) ; C=(r h (pr f (da dx) g-d f) (pr f (da' dx') g'-d' f') .....)
  (let ((u u0) (v v0) (dir0 2)(dir 2) (pas0 ds) (Z nil) (k 0) (tr 0) (cp 0) (val 0) (da 0)
    (pas ds) (drap t))
    (print "" Règles : ") (mapcar 'print (caddr C))
    (pen-size 2 2) (dessin OB) (cercle u0 v0 ry) (pen-size 1 1) (move-to u v) (croix u v)
    (while (and (< k km) drap) ; drap signifie "plus petite distance à un obstacle > ds"
      (setq val (+ val pas) cp (cap u v dir) Z (decision cp C) C (car Z) da (* (cadr Z) am)
        dir (+ dir da) pas (mini (maxi ds (+ pas (* (caddr Z) acc))) pm))
      (if (and (< (* (car cp) dm (cos (- (cadr cp) da))) pas)
        (< (* (car cp) dm (sin (abs (- (cadr cp) da)))) ds)) (set 'drap nil)
        (line-to (set' u (+ u (truncate (* pas (cos dir)))))
          (set' v (+ v (truncate (* pas (sin dir))))) (croix u v)
          (if (< (sqr ry) (+ (sqr (- u u0)) (sqr (- v v0)))) (set 'drap nil)
            (setq tr (+ tr (abs (- 0.5 (divide(- pas pas0)(* 2 acc)) (divide (abs (- dir dir0)) am))))
              pas0 pas dir0 dir k (1+ k)))
          (cons (print "" Reste distance " ; val est la somme des pas donc la distance parcourue
            (set 'val (truncate (* 100. (- 1 (divide val (* km pm)))))) ; val % parcours restant
              "" Tenue de route " (set 'tr (truncate (/ (* 100. tr) k)))
                "" (" k " pas) Valeur " (truncate (agreg (nbsymbol C) tr val))))))
```

Agreg est une fonction d'agrégation entre les trois critères à optimiser.

Données globales :

```
(setq PRED '(ANY NB NS ZE PS PB) NH 2 NC 2 pi 3.14159 ry 130 u0 300 v 0 150 OB (obstacles 200) km 30 ds 3 pm 30 acc 4
  dm 35 am (/ (* pi 45) 180))
```

Exemple :

```
dx = 3
ym = 20
am = 45
ze = 2
dm = 35

Nb. symboles 7
r = 0.7
h = 1.15

2 règles
prioritaires :
[0 0 [nb nb] ze ze]
[0 0 [pb nb] ze ze]
1 règle générale ne
s'appliquant que si
les deux premières
ne sont pas satisfaites :
[1 0 [ze pb] pb ze]

Reste distance 53 (50 pas)
Tenue de route 19%
```

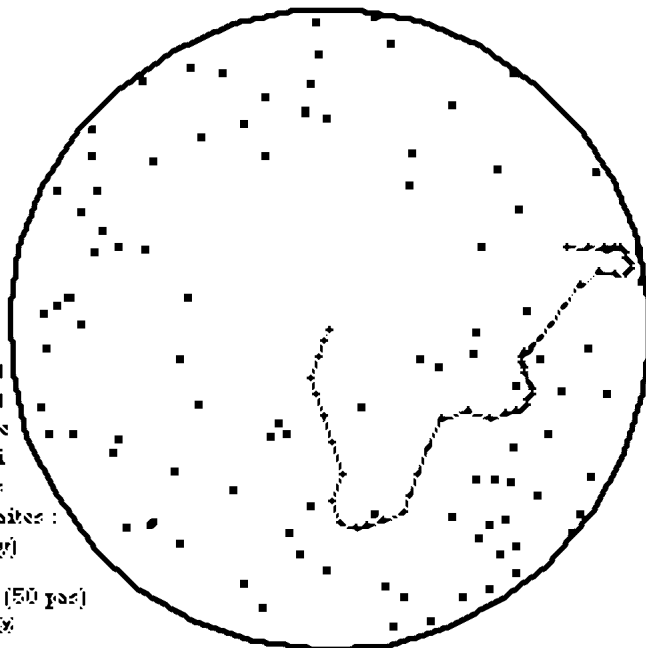


Figure A7.1 Parcours d'un robot à 3 règles modifiant sa direction et sa vitesse.

Tout un module de fonctions lisp complète le programme afin de simuler une évolution pour trouver de "bons contrôleurs" C, ceux-ci sont en fait codés pour moitié, le programme se chargeant de les compléter par symétrie et de vérifier la cohérence de leurs paramètres.

NOYAU D'UN CONTRÔLEUR FLOU

(SOLUTION EN LISP AVEC DES RÈGLES NUMÉRIQUES DE R^2 DANS R^2 DU TYPE (A B R C D) SIGNIFIANT «SI (X, Y) VÉRIFIE LE PRÉDICAT CÔNE DE CENTRE (A, B) ET RAYON R, ALORS (U, V) EST (C, D)».

```
(de fuzzy (LR x y) ; renvoie le couple (u v) de [-1 1] en fonction de x y de [-1, 1]
  (if (null LR) '(0 0) (fuzzybis (car LR) x y 0 0 0 (cdr LR))))
(de fuzzybis (R x y u v Sc RR) ; règle R, u, v numérateurs, Sc somme des coefficients
  (fuzzyter (maxi (- 1 (divide (maxi (abs (- x (car R))) (abs (- y (cadr R)))) (caddr R)))0) (caddr R) (car (last R)) x y u v Sc RR))
(de fuzzyter (cf c d x y u v Sc RR)
  ; cf dans [-1, 1] est le coef d'application de la dernière règle étudiée
  (if (null RR) (if (eq (+ cf Sc) 0) '(0 0) (list (divide (+ u (* cf c)) (+ Sc cf)) (divide (+ v (* cf d)) (+ Sc cf))))
    (fuzzybis (car RR) x y (+ u (* cf c)) (+ v (* cf d)) (+ Sc cf) (cdr RR))))
```

Exemple : (print (fuzzy '((0 0 2 2 4) (2 0 2 6 8)) 1 0)) = (4 6)

MODUS-PONENS GENERALISE

L'intérêt d'un langage fonctionnel comme Lisp est de pouvoir ici définir la fonctionnelle appelée «mpg», qui, ayant pour données une t-norme, une relation d'implication et trois fonctions d'appartenance μ_A , μ_B , et $\mu_{A'}$, va renvoyer la fonction $\mu_{B'}$.

```
(de weber (a b) (cond ((eq a 1) b) ((eq b 1) a) (t 0))) ; t-normes
(de lukas (a b) (maxi 0 (1- (+ a b))))
(de einstein (a b) (divide (* a b) (- (+ (* a b) 2) a b)))
(de proba (a b) (* a b))
(de hamacher (a b) (if (eq (zadeh a b) 0) 0 (divide (* a b) (- (+ a b) (* a b)))))
(de zadeh (a b) (if (< a b) a b)) ; également nommée maxi dans les paragraphes précédents.
```

```
(de reichenbach (p q) (1+ (- (* p q) p))) ; implications
```

```
(de willmott (p q) (maxi (- 1 p) (zadeh p q)))
(de mamdani (p q) (zadeh p q))
(de rescher (p q) (if (< p q) 1 0))
(de kleene (p q) (maxi (- 1 p) q))
(de godel (p q) (if (< q p) q 1))
(de goguen (p q) (if (eq p 0) 1 (zadeh 1 (divide q p))))
(de implukas (p q) (- 1 (maxi 0 (- p q))))
(de larsen (p q) (* p q))
```

```
(df mpg (norme implic muA muB muAp) ; modus-ponens
```

```
(list 'lambda 'y)
(list 'mpgbis norme implic muA (list 'funcall muB 'y) muAp -1 0)))
(de mpgbis (norme implic muA my muAp x sup)
(if (> x 1) sup
; c'est la valeur de muB' renvoyée, les ensembles considérés étant dans [-1, 1]
(mpgbis norme implic muA my muAp (+ 0.1 x)) ; 0.1 est l'incrément utilisé
(maxi sup (funcall norme (funcall muAp x) (funcall implic (funcall muA x) my))))))
```

Exemple, on définit A, B, A' et des fonctions de mise à l'échelle de l'écran

```
(de muA (x) (cond ((< x -1) 0) ((< x -0.5) (* 2 (1+ x))) ((< x -0.2) 1) ((< x 0) (* x -5)) (t 0)))
(de muAp (x) (cond ((< x -0.8) 0) ((< x -0.6) (+ 4 (* 5 x))) ((< x -0.4) (- -2 (* 5 x))) (t 0)))
(de muB (x) (cond ((< x 0.1) 0) ((< x 0.6) (- (* 2 x) 0.2)) ((< x 0.8) 1)
((< x 1) (+ (* -5 x) 5)) (t 0)))
(de phi (x) (truncate (+ 50 (* 200 (1+ x)))))
(de phir (u) (divide (- u 250) 200)) ; est la réciproque de phi
(de psi (y) (truncate (+ 50 (* 100 (- 1 y)))))
```

```
(de graphe (f) (move-to 0 150); trace la fonction f pour 0 < y < 1 et -1 < x < 1
(let ((x 0)) (repeat 500 (line-to x (psi (funcall f (phir x)))) (set 'x (1+ x))))))
(de essai (norme implic)
```

```
(pen-size 2 2) (move-to 500 150) (line-to 0 150) (graphe 'muA) (graphe 'muB)
(pen-size 1 1) (graphe 'muAp) (graphe (mpg norme implic 'muA 'muB 'muAp))
(print ""t-norme : " norme "" implication : " implic))
```

Nous présentons ci-dessous un échantillon (pour la même prémisse «X est A», la même conclusion «Y est B» et le même fait observé A' comme ci-dessus) de la conclusion approchée (en traits fins) pour différentes t-normes et différentes implications.

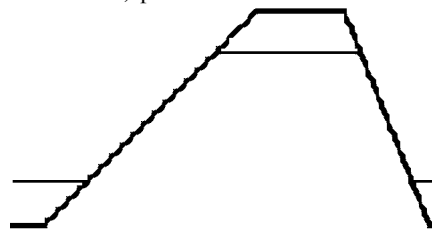


Figure A.7.2 Toute norme sauf Weber, implication de Willmott

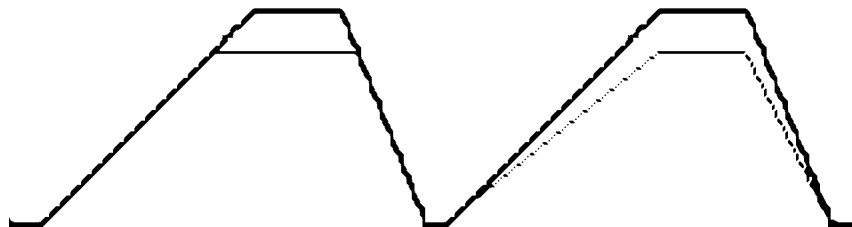


Figure A.7.3 Toutes normes sauf Weber, imp Mamdani et Proba, Hamacher et imp Larsen

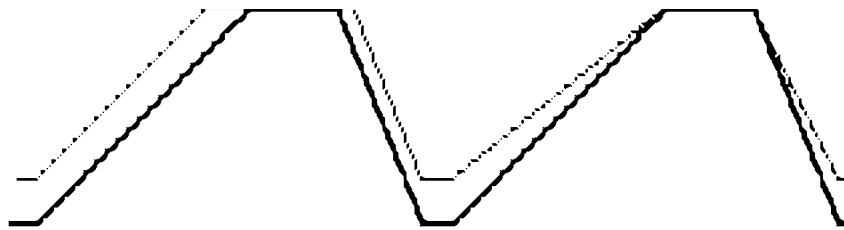


Figure A.7.4 Norme proba, implications Lukasiewicz et Reichenbach

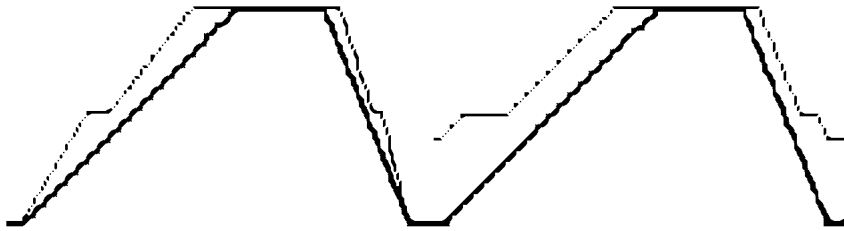


Figure A.7.5 Norme de Zadeh, implications Goguen et de Lukasiewicz

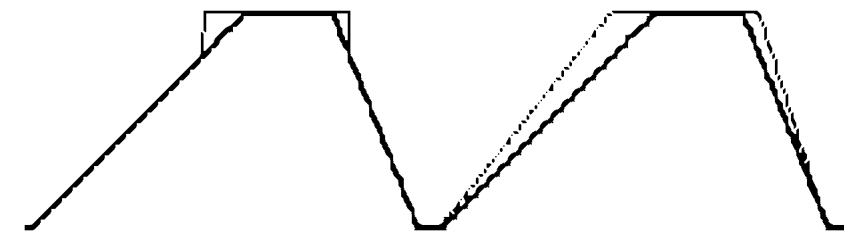


Figure A.7.6 Toute norme sauf Weber
implication Gödel

Toute norme sauf Weber et Zadeh
implication de Goguen

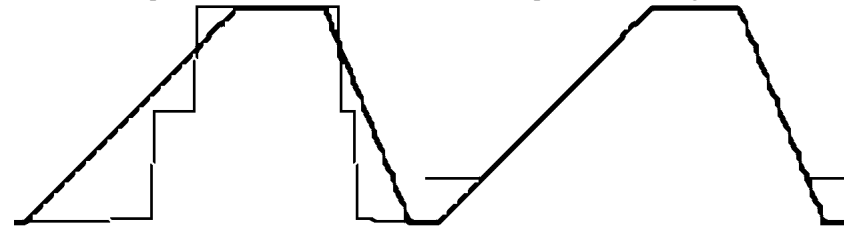


Figure A.7.7 Toute norme, implication de Rescher, Toute norme sauf Weber, imp. Kleene
Norme Weber et implications de Gödel, Goguen et Lukasiewicz

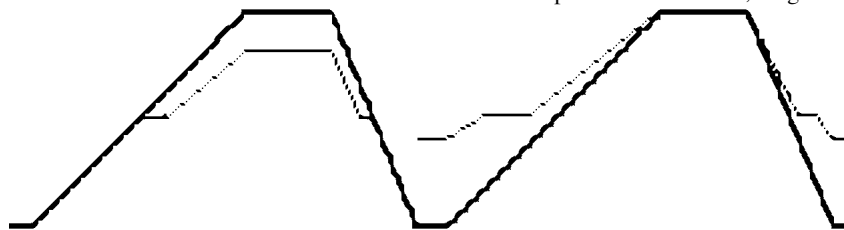


Figure A.7.8 Norme de Zadeh, implications de Larsen et de Reichenbach

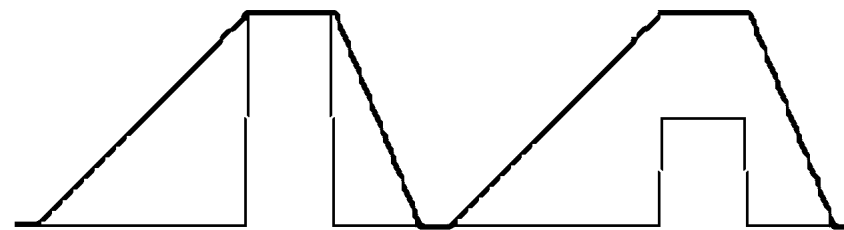


Figure A.7.9 Norme Weber, implication Kleene et à droite de Mamdani, Larsen, Willmott ou Reichenbach

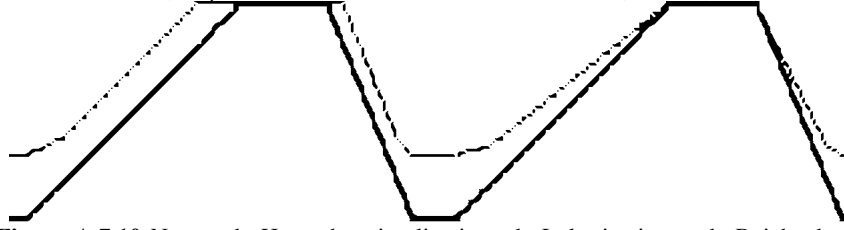


Figure A.7.10 Norme de Hamacher, implications de Lukasiewicz et de Reichenbach

Remarque, les implications de Lukasiewicz, Reichenbach et Willmott ont tendance à donner une incertitude en dehors du support de B, elles peuvent être considérées comme trop fortes.

Problèmes

Tous les énoncés qui suivent ont fait l'objet de projets réalisés au cours des dernières années par des étudiants principalement de l'IIE, mais aussi de l'Université Paris VI ou de l'ENSTA.

Problème 1 Programmation en Fril

a) Soit le programme :

((vole X) (oiseau X)) : (0.9 1)

((vole X) (pingouin X)) : (0 0)

((oiseau X) (pingouin X))

((oiseau X) (aigle X))

((pingouin hector))

((aigle edgar))

On pose la question : qs ((vole Hector)) quelle sera la réponse ?

On rajoute alors au début, avant les clauses, la "déclaration" : dempster vole

Quelle sera la réponse à qs ((vole X)) ?

b) Soit maintenant le programme Fril :

(haut [0.4 : 0, 0.7 : 1])

(assez_haut [0.1 : 0, 0.6 : 1])

(moyen [0.2 : 0, 0.5 : 1, 0.8 : 0])

(bas [0.4 : 1, 0.7 : 0])

(cher [30 : 0, 70 : 1])

(correct [20 : 0, 50 : 1, 60 : 1, 90 : 0])

(pascher [30 : 1, 80 : 0])

dempster choix

((choix X ok) (performance X bonne) (aspect X moderne)) : (0.9 1)

((choix X ok) (prix X cher)) : (0 0.05)

((choix X ok) (securite X bas)) : (0 0.2)

((performance X bonne) (moteur X ok) (securite X haut)) : (0.9 1)

((performance X bonne) (securite X haut)) : (0.6 1)

((aspect panhard moderne)) : (0.8 1)

((aspect levassor moderne)) : (0.9 1)

((securite panhard assez_haut)) : (1 1)

((securite levassor moyen)) : (1 1)

((moteur panhard ok)) : (0.7 1)

((moteur levassor ok)) : (0.9 1)

((prix panhard correct)) : (1 1)

((prix levassor pascher)) : (1 1)

Réponses à la question qs ((choix X Y)) ?

c) Trouver des exemple didactique (ni trop complexe, ni trop trivial) entre vingt et cinquante règles d'utilisation de Fril utilisant des ensembles flous. Le système doit faire intervenir des opérations arithmétique sur les trapèzes et aussi éventuellement manipuler les supports avec "supp_query".

Problème 2 MVL

Explorer le système MVL, en particulier donner un compte-rendu sur l'aspect probabiliste et trouver un exemple de démonstration non trivial.

Problème 3 Modus ponens généralisé

Pour la règle $X \text{ est } A \rightarrow Y \text{ est } B$, créer un outil visualisant B' par rapport à B lorsque A , B sont des triangles et X une valeur précise vérifiant plus ou moins A . L'utiliser pour produire une table de schémas suivant les différents opérateurs T de conjonction et π d'implication.

Problème 4 Opérateurs d'agrégation

Créer un petit logiciel où il soit possible de définir et représenter deux ensembles flous trapézoïdaux, et d'en déduire une agrégation des deux suivant les méthodes exposés au chapitre 3. Prévoir plusieurs relations possibles.

Problème 5 Systeme expert flou

Construire un système expert en chaînage-avant (analyse financière, évaluation du prix d'un appartement ...) dans lequel on puisse discuter des différents modes d'implication.

Problème 6 La machine à laver

On souhaite étudier la normalisation de différentes valeurs floues inférées par un système expert destiné à contrôler une machine à laver (degré de saleté = degré de transparence).

Les règles sont :

synthétique ou laine → fragile	fragile → essorage court
laine → température froide	synthétique → durée courte
laine → durée très courte	saleté moyenne → durée courte
coton → durée longue	saleté grande → durée longue
durée courte → essorage court	saleté faible → durée très courte
couleur vive → RD élevé	saleté grande → trempage
blanc → température très chaude	RD élevé → température froide
poids élevé → durée élevée	RD moyen → température tiède
RD faible → température chaude	saleté très grande → trempage et pré-lavage

On pourra prendre une normalisation différente pour les valeurs des différents arguments: le min pour la durée et la température, le max pour le risque de déteindre RD, ou encore l'intersection et éventuellement en signalant les contradictions pour les autres variables (programmation en lisp).

Problème 7 Détermination de l'âge d'un enfant

Le but est de déterminer un âge ou de vérifier le développement par rapport à l'âge réel en répondant vaguement par "ne sait pas", "vrai", "plutôt vrai", "à moitié vrai", "plutôt faux", "faux", à des questions. Ces questions correspondent à des quadruplets de possibilités (m, a, b, M) dont la signification est la suivante : possibilité 1 entre a et b, possibilité réduite entre m et a et entre b et M de façon à obtenir une représentation trapézoïdale. Chaque règle s'applique avec un coefficient fonction de la valeur de vérité donnée. Le résultat est une distribution de possibilité sur [1, 36] que l'on peut visualiser (programmation en lisp).

sourire (1 1 1 2)	suivre un objet horizontalement (1 2 2 3)
suivre un objet (2 3 3 4)	tient sa tête (2 3 3 4)
gazouillis (1 3 4 5)	attraper un objet d'une main (3 3 4 5)
se retourne (1 2 3 6)	2 incisives en bas (3 6 8 10)
tenir entre le pouce et l'index (8 9 9 10)	dire une syllabe répétée (6 9 9 10)
passer un objet d'une main à l'autre (6 7 7 8)	reconnaissance des parents (6 8 10 10)
position assise (6 7 9 11)	monter un escalier (9 12 12 15)
debout avec soutien (8 9 11 12)	2 incisives en haut (3 8 10 12)
marcher à 4 pattes (6 6 12 12)	manger seul à la cuillère (15 18 24 24)
tient son biberon (10 12 18 18)	8 incisives (6 10 14 20)
faire au-revoir (10 12 15 18)	debout sans soutien (8 12 15 24)
marcher (9 12 15 24)	dire un mot signifiant (10 12 15 24)
connaître des verbes (12 13 17 18)	2 premières molaires (12 14 16 18)
empiler des cubes (12 18 20 24)	désigner les parties du corps (12 15 16 24)
canines (12 15 20 24)	chercher un objet caché (18 20 24 24)
assembler deux mots (18 18 24 28)	propreté le jour (15 24 30 30)
s'habiller (18 24 30 36)	propreté la nuit (18 30 36 36)
descendre un escalier (16 24 30 30)	dessin d'une tête (30 31 35 36)
acquisition du moi (30 31 35 36)	

Problème 8 Détermination de l'époque d'une peinture

Même problème avec la base de connaissance suivante :

scène religieuse (700 1000 1550 1700)
peinture sur bois (1000 1100 1450 1650)
présence de dorure (1150 1200 1400 1475)
portrait (1450 1450 1600 1850)
perspective déformée (1000 1300 1400 1450)
architecture classique (1400 1450 1600 1850)
tableau solennel (1550 1600 1700 1830)
peinture précise très détaillée (1500 1550 1850 1875)
personnages statufiés (1750 1775 1840 1870)
jeux d'ombres et de lumières (150 1600 1700 1800)
personnages assis (1550 1620 1700 1750)
représentation de société (1650 1700 1800 1850)
les couleurs se mélangent (1720 1730 1990 1990)
caractère historique (1750 1800 1850 1900)
ombres de couleurs (1800 1850 1900 1950)
scène réaliste de la vie quotidienne (1750 1800 1850 1900)
grosse couche de peinture (1850 1870 1990 1990)
beaucoup de points (1850 1880 1900 1920)
personnages déformés (1890 1910 1950 1990)

tableau abstrait (1900 1920 1990 1990)
 produit de consommation (1940 1950 1990 1990)
 aucune forme distinguable (1920 1960 1990 1990)
 mouvement des corps et des tissus (1575 1600 1875 1900)

On pourra sortir les résultats sous forme des 0,25 ou 0,5 ou 0,75 ou 1-coupes de l'ensemble flou obtenu.

Problème 9 Aide à la décision dans l'achat d'une voiture

On qualifie les attributs d'une voiture de la façon suivante :

Cylindrée : pet. = (1100 1100 1300 1500) moy. = (1300 1500 1700 1900) grande =(1700 1900 2000 2500)

Puissance : petite = (30 30 50 55) moyenne = (50 55 70 85)

grande = (70 85 120 140) en kW

Vitesse pet.= (120 140 160 175) moy.= (165 175 185 195)

grande = (185 195 210 220)

Prix petit = (45 50 70 80) moyen = (70 80 95 105)

grand = (95 105 120 140) en kF

Taille à définir en m.

Chaque voiture existante, pour ces 5 attributs reçoit un triplet de possibilités correspondant à chaque fois aux 3 qualificatifs. L'utilisateur va répondre pour ces 5 attributs, le qualificatif qu'il souhaite et l'importance qu'il y attache. (voir intégrale de Sugeno)

Problème 10 Un système hiérarchique de règles définissant une fonction

Les règles toutes du type "u est A \rightarrow v est B", sont classées par ordre décroissant de priorités allant des plus précises au moins précises [Yager 92]. Si F(R) désigne l'ensemble des sous ensemble flous de R, chaque paquet de règles va réaliser une fonction de F(R) dans F(R). Soit h(E) le sup de la fonction d'appartenance de l'ensemble flou E (la hauteur), et faible(x) = 1-x un prédicat sur [0,1].

Soit u une entrée réelle, on pose V_0 l'ensemble vide donc faible(V_0) = 1.

Si V_{i-1} est la sortie du paquet i-1 réalisant une fonction f_{i-1} de F(R) dans F(R), alors f_i sera le résultat de n_i règles modifiées par rapport à celles qui ont été données du type "u est A_{ij} et h(V_i) est faible \rightarrow v est B_{ij} "

$h(V_i)$ mesure en fait la façon dont on a pu atteindre l'étape i+1, il se peut donc qu'une conclusion précise fasse sortir avant toutes les étapes.

A chaque paquet, on fait la disjonction des conclusions à la Mamdani ce qui fait le résultat $B_i = \cup B_{ij}$ et on pose $V_i = V_{i-1} \cup B_i$. La sortie v sera la moyenne du dernier V_n .

Faire l'exemple suivant (en lisp) et la représentation de la fonction u \rightarrow v, si les règles sont :

- 1° paquet U est 5 \rightarrow V est 13
 U est 75 \rightarrow V est 180
 U est 85 \rightarrow V est 100
- 2° paquet U est environ 10 \rightarrow V est environ 20
 U est environ 30 \rightarrow V est environ 50
 U est environ 60 \rightarrow V est environ 90
 U est environ 80 \rightarrow V est environ 120
 U est environ 100 \rightarrow V est environ 150
- 3° paquet U est petit \rightarrow V est environ 40
 U est moyen \rightarrow V est environ 85
 U est grand \rightarrow V est environ 130
- 4° paquet U est quelconque \rightarrow V est 2*U

dans lequel "environ x" est le triangle isocèle centré sur x de demi-base 10 sur [0,100] et 20 sur [0, 200], petit, moyen, grand étant une famille de triangles de demi-base 50 sur [0, 100]. Programmer en lisp et donner d'autres exemples.

Problème 11 Système expert fonctionnant avec la logique des défauts

Dans un système déductif où on a des règles du type A \rightarrow C, et A et B \rightarrow \neg C, on pourra dire que A entraîne C en général (par défaut d'une information B supplémentaire, auquel cas la première règle ne doit plus se déclencher). Ainsi si T et T' sont des ensembles de règles et d'axiomes avec T inclut dans T', on n'a pas nécessairement la théorie de T incluse dans celle de T' (logique non monotone).

On souhaite marier les approches flou et défauts dans un petit système déductif limité au calcul propositionnel. Les propositions y reçoivent un degré de vérité v satisfaisant à $v(\neg P) = 1 - v(P)$.

Définitions :

Axiome : règle sans prémisses

Preuve de P : arbre de racine P dont les feuilles sont des axiomes et dont chaque noeud est la conclusion d'une règle dont les prémisses sont les fils du noeud.

Longueur d'une preuve : nombre d'arêtes du plus long chemin

Cardinal d'une preuve : nombre de variables propositionnelles ayant une occurrence dans l'arbre.

On considère un système de règles maintenant affectées de coefficients et la théorie engendrée grâce aux métarègles suivantes :

- 1) Si A_1 et A_2 et A_3 et ... \rightarrow C est une règle de coefficient r avec chaque A_i connu avec le coefficient a_i , alors C est admis (provisoirement) avec le coefficient $r * \min(a_1, a_2, a_3, \dots)$
- 2) Si P est obtenu par deux preuves de longueurs distinctes, P aura le coefficient correspondant à la preuve de plus courte longueur. (Ainsi on donne la priorité à des déductions plus immédiates et on évite le fait d'avoir à agréger par exemple un axiome avec son éventuelle démonstration, cas fâcheux où une connaissance initiale pourrait se trouver affaiblie.)
- 3) Si P est obtenu par deux preuves de mêmes longueurs mais de cardinaux distincts, c'est la preuve de plus grand cardinal qui l'emporte (Ainsi des précisions supplémentaires entraînent une priorité sur des règles générales.)
- 4) Si P est obtenu par deux preuves de mêmes longueurs et cardinaux, alors on prend simplement la moyenne, mais comme cette opération n'est pas associative, il faudra tenir compte du nombre d'occurrences.

5) Si $\neg Q$ et $P \rightarrow Q$ alors $\neg P$ (contraposition).

Soit la base de connaissances suivante :

En général les oiseaux volent. Les pingouins sont des oiseaux qui ne volent pas, sauf ceux des Kerguelen qui volent légèrement. Hector et Victor sont des pingouins des kerguelen, mais Victor ne vole pas vraiment. Nestor est un autre pingouin.

a) Dérouler à la main l'enchaînement des déductions suivant la stratégie ci-dessus.

b) Dessiner et expliquer le graphe pour la théorie de $T_1 = \{A, E, A \rightarrow B, B \rightarrow C, E \rightarrow B\}$ ainsi que pour $T_2 = \{A, B, D, E, A \& B \rightarrow C, B \& C \rightarrow D, C \& E \rightarrow F, D \& F \rightarrow C\}$. L'ensemble de toutes les conclusions est-il toujours parfaitement défini ? Donner un algorithme pour le construire.

c) Trouver une représentation qui va permettre de mettre en oeuvre un tel moteur d'inférence et le programmer en lisp.

d) Trouver un exemple plus étoffé pour le tester et commenter les résultats.

Problème 12 Morphologie mathématique

Si E , ensemble flou de R^2 , est caractérisé par sa fonction d'appartenance μ , et f est un élément structurant (une définition floue de voisinage) défini par exemple par $f(t) = \min(1, \max(0, 1 - t/a))$ (un cône) où a est de l'ordre de 1 à 5 pixels).

On rappelle que l'on définit la dilatation $Dil_{\mu}(x) = \sup_{y \in E} \{T(\mu_E(y), f(d(x, y)))\}$ où T est une t -norme et S sa conorme associée et l'érosion $Ero_{\mu}(x) = \inf_{y \in E} \{S(\mu_E(y), 1 - f(d(x, y)))\}$.

On veut une image floue mais non bruitée à partir d'une image floue.

Faire un petit logiciel de démonstration avec quelques images tests dans R^2 , réalisant la dilatation de l'érosion floue en permettant un choix de a . On prendra la t -norme de Zadeh et la norme Hamming dans R^2 pour des raisons de simplicité, ainsi, si $a = 1$, en tout point on teste ses 8 voisins, et si $a = 2$ ses 24 voisins.

Problème 13 Classification floue

On prendra l'algorithme de Bezdek modifiant les centres de gravité des différentes classes.

Le programme devra visualiser les itérations de la façon suivante : chaque point de l'ensemble sera représenté par un carré de 10 sur 10 pixels (pour cela ils ne doivent pas être trop voisins). S'il y a par exemple 3 classes fixées, (rouge, verte et jaune) chaque exemple sera représenté par un rectangle contenant (régulièrement répartis) des pixels de chaque couleur au prorata de ses 3 appartenances. On pourra par ailleurs consulter la table des points avec ces taux d'appartenance aux différentes classes, à la fin des itérations. Prévoir deux ou trois bases d'exemples avec 2, 3, 4 classes.

Problème 14 Régulateur cardiaque

On veut modifier de minute en minute les doses de deux substances A et R destinées à accélérer ou retarder le rythme cardiaque (mesuré par B en battements par minutes) en temps réel dans un hôpital. Si B_0 est le rythme fixé, on note $E = B - B_0$ et on fixe les règles par une liste comportant un poids, la conclusion et les deux prémisses :

(0,5 A PB E NB dE PS) (0,8 A PM E NM dE ZE) (1 A PS E NS dE NS)

(1 A ZE E ZE dE NM) (1 R ZE E ZE dE PM) (1 R PS E PS dE PS)

(0,8 R PM E PM dE ZE) (0,5 R PB E PB dE NS)

Si chaque prédicat est représenté par une liste de 21 valeurs (les possibilités sur $[-1, 1]$) programmer (en lisp) la fonction B_0 (cible), B_1, B_2 (deux mesures consécutives) \rightarrow couple A, R des doses réelles, en passant par des fonctions φ, ψ, χ ramenant à $[-1, 1]$ à définir.

Problème 15 Evitement d'obstacles

On suppose une voiture équipée de 8 capteurs de distances aux obstacles se trouvant sur 8 directions réparties régulièrement AV (avant), AR, GA (gauche), DR, VG (avant gauche), VD, RG, RD ("angles morts").

Ecrire des règles destinées à contrôler la vitesse v et sa rotation α , du type :

(ΔAV est NB) et (v est PB) \rightarrow (Δv est NB)

Que doit-il se passer lors d'une contradiction avec (ΔAR est NB) \rightarrow (Δv est PB) ?

Effectuer une simulation avec simplement trois capteurs dans une fenêtre occupée par différents obstacles aléatoirement distribués (entre 10 et 100), une sorte de labyrinthe.

Règles de Simiand-Noury, pour un pas maximal $pm = 10$, un angle de braquage maximal $am = 60^\circ$ une accélération comprise entre -3 et 2 pixels ainsi que 3 prédicats ZE (1 avant 5, 0 après 15), PS (support [5, 25], noyau {15}) et PB (nul avant 15, 1 après 25), les règles portent sur les distances gauche g , frontale f , et droite d .

4 règles modifiant le pas :

$f ZE \rightarrow dp NB$ $f PB \rightarrow dp PB$ $d ZE \rightarrow dp NS$ $g ZE \rightarrow dp NS$

Si $d < g$ les 4 règles :

$g ZE \rightarrow NB$ $g PS \rightarrow da NS$ (f ou $d ZE$) $\rightarrow da PB$ (f ou $d PS$) $\rightarrow da PS$

Si $d < g$, les 4 règles analogues.

Règles de Nguyen Kiet - Queulin, pour un pas maximal $pm = 5$, un angle de braquage maximal $am = 30^\circ$, $ds = 1$, $dm = 100$ une accélération comprise entre -5 et 5 pixels ainsi que 3 prédicats N, Z, P ramenés à $[-100, 100]$, deux règles pour éviter l'obstacle frontal :

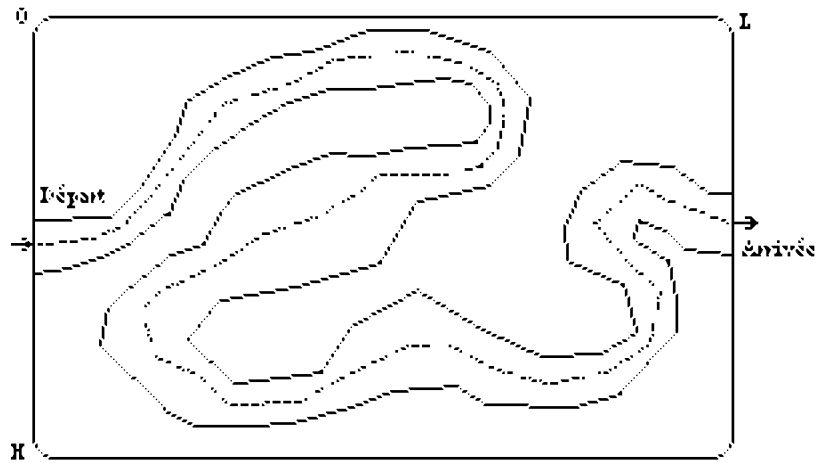
$f Z$, pas P $\rightarrow dp N$, da N, $f Z$, d Z, g Z $\rightarrow da P$, une pour accélérer : $f P \rightarrow dp P$, et 4 règles pour tourner : $d Z$, g P $\rightarrow da N$, g Z, d P $\rightarrow da P$,

g P, f Z $\rightarrow da N$, d P, f Z $\rightarrow da P$

Problème 16 Simulation d'un suivi de route

En traçant sur l'écran un couloir de largeur variable, coudé avec des angles aigus, on cherche à contrôler le trajet d'un robot depuis l'entrée à gauche jusqu'à la sortie à droite.

Le robot doit avancer à vitesse constante (ou en modifiant à la fois sa direction et sa vitesse) en suivant les coudes et en restant si possible au milieu du couloir, il ne dispose pas de capteur frontal, mais seulement de deux capteurs mesurant les distances l et r perpendiculairement (ou plus généralement à un angle \pm ouv de sa direction), aux murs du tunnel.



Ecrire un système de règles floues $(l, \Delta l, r, \Delta r) \rightarrow \Delta\alpha, \Delta v$, avec 3 ou 5 prédicats dans $\{NB, NM, NS, ZE, PS, PM, PB\}$, contrôlant l'angle de la direction et la variation de pas à chaque instant. Visualiser l'avancer du robot. On peut chercher à ce que le robot conserve sa droite, ou le milieu, ou encore à minimiser les variations de directions en conservant une certaine régularité de trajectoire à la façon de la navigation fluviale.

Problème 17 Simulation d'un robot suiveur

Dans une boucle infinie, un mobile A se déplace grâce aux flèches du clavier, un robot B cherche à le suivre en fixant son cap sur A en modifiant sa vitesse, et en maintenant une distance constante avec A.

Ecrire des règles de conduite destinées à définir une fonction : a) $(d, \Delta d) \rightarrow \Delta v$ pour une poursuite rectiligne. b) $(d, \alpha, \Delta\alpha, \Delta d) \rightarrow \Delta v$ pour le problème plan c) On laisse de côté le problème d'un missile dans l'espace.

Problème 18 Métro

Aux heures d'affluence on cherche à optimiser le débit des rames de métro, et pour cela on peut concevoir deux contrôleurs, l'un pour l'arrêt (dans une station ou à un feu rouge), l'autre pour ne pas perdre de temps dans les tunnels. C'est au second que l'on s'intéresse en souhaitant fixer l'allure d'un métro sur celle de son prédécesseur grâce à l'accélération u . Au métro n'est fourni qu'une seule donnée d , on note $e = d - d_0$ où d_0 est une distance minimale très réduite ne dépendant pas de la vitesse des deux rames.

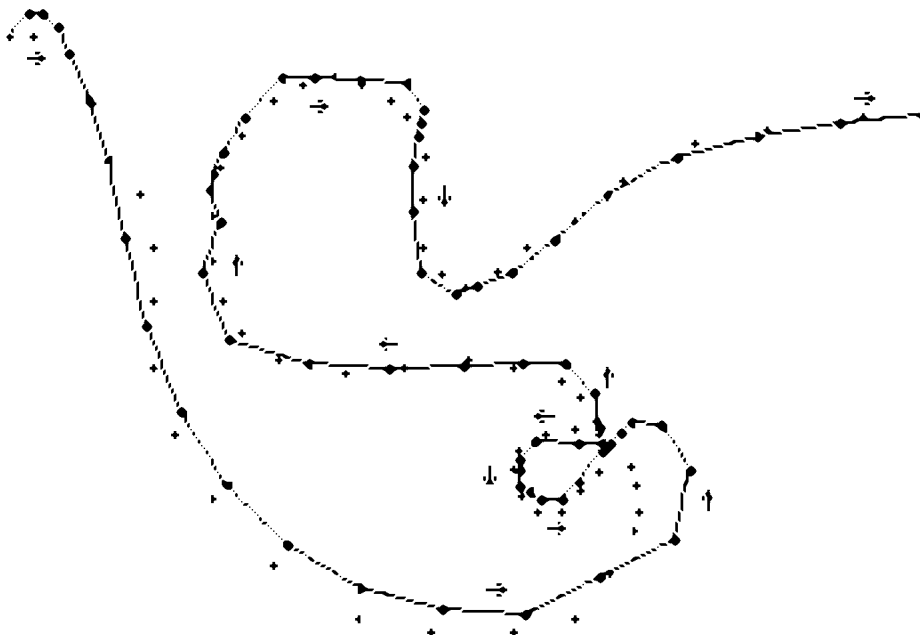
On doit donc avoir $e > 0$, mais le plus petit possible, en fait dès que e est négatif on ralentit le métro. C'est la rapidité du cycle du contrôleur flou qui vca assurer la sécurité. Former des règles telles que : $(e \text{ est NB}) \rightarrow (u \text{ est NB})$ $(e \text{ est PB}) \rightarrow (u \text{ est PB})$
 $(e \text{ est PM}) \text{ et } (\Delta e \text{ est PS}) \rightarrow (u \text{ est PB})$ $(e \text{ est ZE}) \text{ et } (\Delta e \text{ est PM}) \rightarrow (u \text{ est PM})$

Problème 19 Suivi d'une ligne de marqueurs

A chaque instant un robot mesure sa distance x et l'angle a de sa direction avec la prochaine borne, il modifie deux paramètres : sa variation de direction et sa variation de pas c'est à dire sa vitesse.

On utilisera les règles suivantes $(a \text{ ZE}) \rightarrow (\partial \text{dir ZE et } \partial p \text{ PB})$, $(x \text{ ZE et } a \text{ ZE}) \rightarrow (\partial \text{dir ZE et } \partial p \text{ NB})$, $(x \text{ PB et } a \text{ PB}) \rightarrow (\partial \text{dir NS et } \partial p \text{ is PB})$, $(x \text{ PB et } a \text{ NB}) \rightarrow (\partial \text{dir PS et } \partial p \text{ PB})$, $(x \text{ PB et } a \text{ PS}) \rightarrow (\partial \text{dir NB et } \partial p \text{ ZE})$, $(x \text{ PB et } a \text{ NS}) \rightarrow (\partial \text{dir PB et } \partial p \text{ PB})$, $(x \text{ ZE et } a \text{ PB}) \rightarrow (\partial \text{dir NB et } \partial p \text{ NB})$, $(x \text{ ZE et } a \text{ NB}) \rightarrow (\partial \text{dir PB et } \partial p \text{ NB})$. On fixe les paramètres $d_m = 50$, (distance maximale de reconnaissance) $acc = 7$ (augmentation maximale du pas), $vm = 90^\circ$, braquage $am = 50^\circ$ $pm = 35$ (pas maximal en pixels).

Exemple de circuit où les bornes sont numérotées :



Problème 20 Suivi d'une ligne blanche

Tracer la trajectoire oscillante d'un véhicule avec d sa distance algébrique à la ligne mesurée perpendiculairement à sa direction, et non perpendiculairement à la ligne d dans $[-dm, dm]$ et Dd dans $[-vm, vm]$ en entrée et a dans $[-A, A]$ en sortie (règles données par Kamada-Yoshida, programmation en C).

Problème 21 Même sujet

Mais on étudiera de façon systématique l'influence des facteurs d'échelle, du nombre de prédicats (3, 5, 7) de leur base 2^p , de la conjonction (min, produit, Lukasiewicz..) et surtout des règles en retirant ou modifiant chacune d'entr'elles partant par exemple de 5 prédicats et 25 règles.

Le but de ce travail est de décrire de façon rationnelle l'influence de certains facteurs sur les résultats d'un contrôleur flou, en regroupant des informations faites à partir d'une simulation graphique.

On dispose d'un programme C réalisant la trajectoire d'un robot devant suivre une sinusoïde, et d'une interface délivrant pour chaque expérience, un tracé légendé par les valeurs des paramètres utilisés, ainsi qu'une table fournissant le nombre d'occurrences des règles déclenchées.

Le robot mesure à chaque pas sa distance algébrique à la courbe, d mesurée suivant un angle \pm ouv par rapport à sa trajectoire. Sa variation de direction sera contrôlée suivant les deux entrées d et dd (variation de distance). L'angle de braquage étant dans l'intervalle $[-A, A]$, d dans $[-B, B]$, dd dans $[-C, C]$, l'avance ds, la portée maximale des capteurs dm, nsi le nombre de sommets de la courbe inscrits dans la fenêtre considérée, et dim la dimension de la fenêtre. On définit par ailleurs les critères de performance appréciés par la somme rot des valeurs absolues de rotations et la distance parcourue sd que l'on cherche à minimiser.

Travail demandé :

1° Partant des meilleurs résultats donnés dans le rapport de C.Deneuille, pour un jeu de règles complet avec 5 prédicats on désire visualiser le nuage de points correspondant aux meilleures expériences (au sens de rot + sd) et obtenir une loi par corrélation linéaire pour les fonctions empiriques ds / dim \rightarrow A, ds / B \rightarrow A, nsi \rightarrow A, on pourra chercher d'autres relations, attendu que dm, ouv ont moins d'importance que B et C qui ont eux-même moins que A ou ds. Le rapport B/C est-il toujours du même ordre ?

2° Faire l'examen pour les 25 conclusions de les retirer ou de les modifier. Il serait bon de définir une méthodologie pour ce problème afin d'avoir automatiquement cette recherche. Modification dans le poids des règles?

3° Reprendre cette recherche des "bonnes règles" en faisant varier p la demi-base des triangles avec 5, 7 ou 9 prédicats analogues, et k pour avoir $2k+1$ prédicats triangulaires non isocèles formant une famille binaire dont les sommets $0 = x_0 < x_1 < x_2 < \dots < x_k = 1$ tels que $x_i = i(i+1) / k(k+1)$.

Problème 22 Le chat et la souris

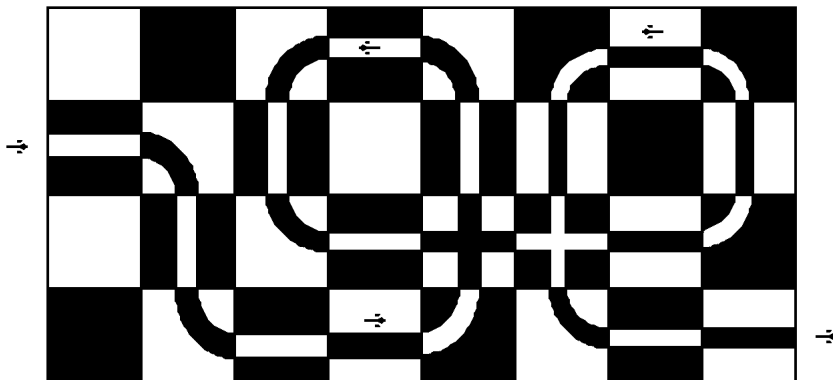
Un chat doit poursuivre une souris en allant m fois plus vite qu'elle mais avec un braquage maximal m fois plus petit. Chacun d'entre eux est contrôlé par des règles floues, le chat cherchant à rejoindre la souris va modifier sa direction suivant des paramètres d'entrées à définir et la souris va chercher à fuir en conservant une direction la plus perpendiculaire possible à celle du chat. Créer un logiciel permettant de modifier les prédicats grâce aux paramètres p et h s'ils sont définis par des triangles centrés sur m, de base $[m-p, m+p]$ et de hauteur $h \geq 1$, tronqués par 1.

Les deux entrées peuvent être les angles tc et ts que font respectivement les directions du chat et de la souris avec l'axe CS qu'ils forment, ces angles étant ramenés à $[-4\pi/5, 4\pi/5]$. Quant aux règles, on pourra essayer pour la souris : tc NB et ts PS alors δ dir PB, tc ZE et ts NB alors δ dir NS, tc ZE et ts NS alors δ dir NB, et leurs symétriques. Outre ce qui est donné au chapitre 5, voici deux autres exemples de systèmes de règles pour la souris :

ts					ts							
PB				NB	NB							
PS				NB	NB							
ZE			PB									
NS	PB	PB										
NB	PB	PB										
	NB	NS	ZE	PS	PB	tc	NB	NS	ZE	PS	PB	tc

Problème 23 Suivi de ligne sur un damier

On demande de créer un damier analogue à celui ci-dessous où un robot doit suivre la ligne marquée alternativement en blanc sur fond noir ou l'inverse avec d'éventuels croisements.



Le principe doit être le suivant : devant le robot une barre rectiligne munie de 16 capteurs binaires voyant le noir ou le blanc permet à chaque instant de mesurer un nombre en chiffres binaires (les 8 pixels de droite, l'unité étant le plus lointain) et un second nombre binaire (les 8 pixels à gauche, l'unité étant le plus lointain). L'entrée du contrôleur flou est unique, c'est la différence entre ces deux nombres, il est apprécié par 3 prédicats NB, ZE, PB dans un intervalle à définir suivant les dimensions des cases du damier. La sortie est la variation de direction à prendre qualifiée par les 3 même prédicats, mais sur l'intervalle de -90° à 90° . Trouver de bonnes règles (3 peuvent suffire) et

expérimenter (éventuellement sur plusieurs circuits). Ce thème a servi de concours de robotique mobile depuis à La Ferté Bernard, P.Darche de Paris VI a conduit une équipe et un robot de 18kg qui devait également ramasser des boules.

Problème 24 Lignes de niveaux

Etant donnée une fonction réelle f de deux variables, on souhaite tracer k lignes de niveaux régulièrement répartis dans un intervalle $[p, q]$ où l'on sait par un moyen quelconque que f prend ses valeurs sur un domaine donné $[a, b] \times [c, d]$. Par exemple f de $[0, 1] \times [0, 1]$ dans $[0, 1]$ où les niveaux sont $0, 0.1, 0.2, \dots, 0.9, 1$ à ϵ près. Le problème est de relier ces points de façon à former des lignes.

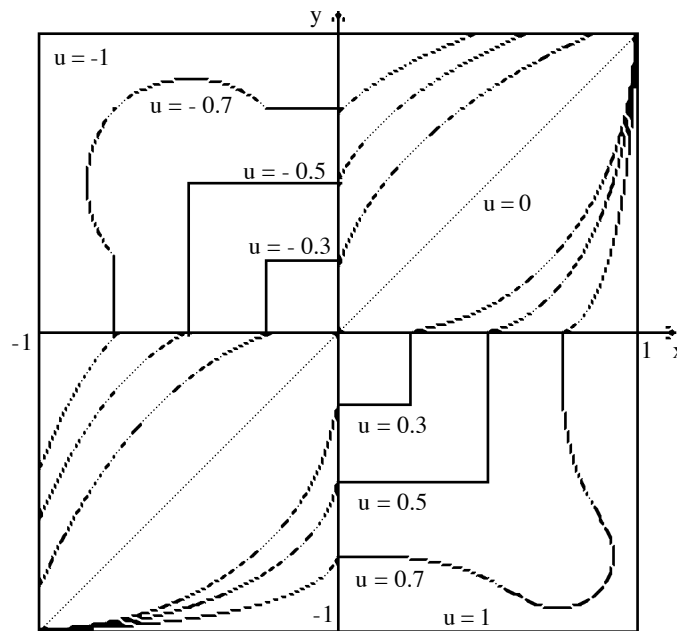
Chaque ligne correspondant à un fichier séquentiel dans lequel chaque nouveau point admis devra s'insérer suivant une stratégie à définir. La procédure, une fois mise au point, sera surtout utilisée pour des fonctions n'ayant pas d'expression mathématique simple. Ainsi pour un contrôleur flou abstrait $(x, y) \rightarrow u$, on veut visualiser les lignes de niveaux de la fonction et les comparer en faisant varier le nombre de prédicats, leur forme (triangle de demi-base p ou gaussiens d'écart-type s), les opérateurs "et" et "ou", les règles en débutant avec un tableau complet et en rompant la symétrie du tableau et en les rayant une par une.

Exemple de contrôleur $(x, y) \in [-1, 1]^2 \rightarrow u \in [-1, 1]$ donné par [Bühler 94].

On présente sur la partie haute gauche le résultat u obtenu par max-min et centre de gravité avec une base de 9 règles. Sur la partie droite basse, les mêmes lignes si 5 règles opèrent seulement.

P	N	N	Z
Z	N	Z	P
N	Z	P	P
y / x	N	Z	P

P		N	
Z	N	Z	P
N		P	
y / x	N	Z	P



Problème 25 Prototypes de commandes linguistiques approchées

Réaliser un logiciel acceptant des ordres de visualisation approchés. Les commandes seront soit des phrases, soit une suite de commandes à cliquer parmi un certain nombre de mots, de telle sorte que l'ordre donné soit du style "tracer / déplacer / supprimer / agrandir / réduire / déformer, (un peu / beaucoup), un / le, (petit / moyen / grand), rectangle / ellipse, (tres / peu, excentré), debout / couché, rouge / bleu / jaune ..., ((très, peu...) au dessus / sous / à droite, proche / loin... du GN) / en haut / bas, à droite / gauche". (GN = groupe nominal). On pourra simplifier l'excentricité comme une notion commune $1 - b/a$ aux rectangles et ellipses (0 pour le cercle et le carré).

Formaliser la syntaxe de ces phrases afin de n'accepter que celles qui ont un sens.

Donner une représentation floue des prédicats unaires "petit"... et binaires "loin de" ..., des modificateurs "très", "peu" ... Tout est relatif à la taille de la fenêtre de démonstration.

Le programme doit de plus pouvoir traiter une donnée par défaut et modifier l'interprétation des ordres donnés de telle façon que chaque figure soit entièrement incluse dans l'écran et qu'elles soient deux à deux disjointes. Il est permis d'introduire d'autres spécificités.

Problème 26 Réalisation d'un contrôleur neuro-flou

Devant un système présentant un tel caractère expérimental, il faut des outils permettant de conclure rapidement sur les résultats, aussi va-t-on utiliser une visualisation, entre 5 et 10 lignes de niveaux pour la fonction principale $x, y \rightarrow u$ comme pour les "et".

Programmer l'ensemble pour avoir un réseau global R, après expérimentation sur une fonction $x, y \rightarrow u$ obtenue au seuil près sur quelques exemples, (on pourra prendre plusieurs corps de règles initiaux).

Préparer une interface présentant clairement pour un corps de règles initial, au bout d'un certain nombre de rétropropagations, le graphique des prédicats modifiés, les différentes fonctions "et" par leurs lignes de niveaux sur $[-1,1]^2$, l'allure de la fonction principale et l'état des règles par une table des w_i .

Problème 27 Descente à ski

On matérialise une piste de largeur constante, par une bande pour $0 \leq x \leq L$ où la pente est $p_x = -\sin(\pi x / L) = \tan(t)$ où t est l'angle de la plus grande pente en x , avec l'horizontale. Un skieur la parcourt de gauche à droite sur l'écran en connaissant à chaque instant sa position $x, h_1 < y < h_2$ et sa direction α par rapport à la plus grande pente. Sa vitesse est alors donnée par un "pas" tel que $ds \leq pas < p_m$. Il ne peut contrôler que sa direction en modifiant son cap α grâce à $-a_m \leq \delta\alpha \leq a_m$.

A chaque instant la position (x, y, α) du skieur permet de calculer (le vérifier) son accélération (ou modification du pas) $\partial p = s \cdot \text{acc} \cdot \cos(\alpha) / (2 / (1 + s^2))^{1/2}$ où $s = \sin(\pi x / L)$. Vérifier que si le skieur fait α avec l'axe de la piste et β avec l'horizontale, $\sin \beta = \sin t \cdot \cos \alpha$, et que son accélération doit être proportionnelle à $\sin \beta$ donc maximale (valeur acc) pour $x = L/2$ et $\alpha = 0$.

Puis le pas est réellement modifié par l'ajout de $\partial p - f \cdot \text{pas}$ où f est un coefficient de frottement.

Enfin un contrôleur flou à 3 entrées doit déterminer la fonction $(y, \partial p, \alpha) \rightarrow \partial \alpha$.

Le parcours est stoppé dès que $\text{pas} \geq p_m$ où si y ne respecte plus $h_1 < y < h_2$ ou encore si $x \geq L$.

L'objet de la recherche est de trouver par algorithme génétique de bons contrôleurs flous. Dans tous les cas la performance à minimiser est le nombre de pas nécessaire à la condition $x > L$. Un skieur est une liste de règles floues réalisant une fonction $(y, \partial p, \alpha) \rightarrow \partial \alpha$. On se limitera aussi bien pour les entrées que pour la sortie aux 3 prédicats N, Z, P définis sur $[-1, 1]$. On imposera la symétrie qui veut que $(-y, \partial p, -\alpha) \rightarrow -\partial \alpha$. Un chromosome est donc formé par 15 règles.

Les paramètres $L, a_m, d_s, p_m, f, \text{acc}, p_0$ seront à fixer suivant la simulation désirée.

Ce problème s'écarte du suivi de ligne par un robot : il s'agit ici d'aller vite sans suivre la ligne de plus grande pente, ni de trop s'approcher des bords de la piste. Une stratégie d'évolution appropriée doit faire émerger un comportement global : une trajectoire sinueuse dont les oscillations sont d'autant plus resserrées que la pente est forte. Toute solution trouvée au terme d'une évolution doit pouvoir être réexécutée à l'écran avec l'affichage en clair des règles trouvées le tracé de la descente en vue de dessus et profil, et éventuellement être sauvegardée.

Bibliographie

- Abdi H. Les réseaux de neurones, Presses Universitaires de Grenoble 1994
- Aminzadeh F. Jamshidi M. Soft Computing p.29-43 Prentice Hall 1994
- Andrès V. Filtrage sémantique dans une base de données imprécises et incertaines : un système souple autorisant la formulation de requêtes composites pondérées. Thèse Université P.Sabatier Toulouse 1989
- Akdag H. Une approche logique du raisonnement incertain, Thèse d'état Paris VI 1992
- Allen An interval-based representation for temporal knowledge, 7th Int. Conference on AI, 1981
- Alliot J.M. Durand N. Medioni F. Algorithmes génétiques et programmation linéaire appliqués à la résolution de conflits, Evolution artificielle p.91-98 Cepaduès, 1994
- Appriou A. Procédure d'aide à la décision multi-informateurs. Applications à la classification multi-capteurs de cibles ONERA 92322 Châtillon, 1988
- Assilian S. Mamdani E.H. Learning control algorithm in real dynamic systems. Proceedings of the 4th international IFAC / IFIP Zurich, 1974
- Atanassov K. Intuitionistic fuzzy sets, FSS vol. 20 p.87-96, 1986
- Baldwin J.F. A new approach to approximate reasoning using a fuzzy logic, in FSS n°2, 1979
- Baldwin J.F. Evidential support logic programming, Fuzzy Sets and Systems n°24, 1987
- Baldwin J.F. Martin T. An abstract mechanism for handling uncertainty, 3° conférence IPMU Paris, 1990
- Baldwin J.F. Martin T. Pilsworth B. Fril, fuzzy and evidential reasoning in AI, Research studies press, 1994
- Baldwin J.F. Knowledge from data using Fril and fuzzy methods, Fuzzy logic, Wiley 96
- Baldwin J.F. Lawry J. Martin T.P. A mass assignment theory of the probability of fuzzy events, FSS vol 83 p.353-367, 1996
- Barendregt H.P. The lambda-calculus, its syntax and semantics, North Holland 1981
- Bartolini G. Casalino G. F.Davoli M.Minciardi E.Morten Development of performance adaptive fuzzy controllers with application to continuous casting plants. Industrial applications of fuzzy control M.Sugeno ed. North Holland, 1985
- Belnap N.D. A useful four-valued logic, Modern uses of multiple values logics, Epstein Dumm eds n°8-37, Dordrecht, 1977
- Béreau M. Dubuisson B. A fuzzy extended k-nearest neighbors rule, FSS 44 p.17-32, 1991
- Berenji H.R. A reinforcement learning-based architecture for fuzzy logic control, international journal of approximate reasoning n°6 p.267-292, 1992
- Besnard P. Default logic, Springer Verlag 1989
- Bezdek J.C. Cluster validity with fuzzy sets, Journal of cybernetics vol 3 p.58-73, 1974
- Bezdek J.C. Pattern recognition with fuzzy objective function algorithms, Plenum press New York, 1981
- Bloch I. Maître H. Constructing a fuzzy mathematical morphology : alternative ways, IEEE, 1993
- Bloch I. Maître H. Fuzzy mathematical morphologies : a comparative study, Pattern recognition vol 28 n° 9 p.1341-1387, 1995
- Bloch. I. Pellot C. Sureda F. Herment A. Fuzzy modelling and fuzzy mathematical morphology applied to 3D reconstruction of blood vessels by multi-modality data fusion, in Fuzzy sets methods and information Engineering : a guided tour of applications, Wiley, 1996
- Bonissone P.P. Summarizing and propagating uncertain information with triangular norms, Readings in uncertain reasoning Morgan Kaufmann Publ. p.239-253 San Mateo, 1987
- Bosc P. Galibourg M. Hamon G. Fuzzy querying with SQL : extensions and implementation aspects, FSS n°28 p.333-349, 1988
- Bosc P. Liétard L. Incertitude et division relationnelle de relations floues, Actes LFA p.126-133, 1995
- Bosc P. Pivert O. SQLf, A relational database language for fuzzy querying, IEEE Transactions of fuzzy systems, vol.3 p.1-17, 1995
- Bothorel S. Toward general measures of comparison of objects, FSS 84 p.143-153, 1996
- Bouchon-Meunier B. Management and uncertainty in knowledge-based systems in Encyclopedia of computer science and technology vol. 20 (Marcel Dekker inc. New York) 1989
- Bouchon-Meunier B. Stability of linguistic modifiers compatible with a fuzzy logic Uncertainty in intelligent systems Springer Verlag Lectures notes in computer science, vol 313, 1988
- Bouchon-Meunier B. Yao J. Linguistic modifiers and imprecise categories, Journal of intelligent systems, vol.7 p.25-36, 1992
- Bouchon-Meunier B. La logique floue, P.U.F. (Que-sais-je?), 1993
- Bouchon-Meunier B. La logique floue et ses applications, Addison Wesley, 1995
- Bouchon-Meunier B. Marsala C. Ramdani M. Arbres de décision et théorie des sous-ensembles flous, Actes du 5° PRC-GDR en IA p.50-53, 1995
- Bouchon-Meunier B. Nguyen H.T. Les incertitudes dans les systèmes intelligents, PUF 1996
- Boverie S. La commande floue dans les systèmes automobiles, Neuro-Nîmes p.21-29, 1993
- Broomhead D.S. Lowe D. Multivariable functional interpolation and adaptive networks, Complex systems, vol. 2 p.321-355, 1988
- Brunet Contribution au neuro-flou, Thèse Université d'Evry 1996

- Buchanan B.G. Shortliffe E.H. Rule-based expert systeme : the Mycin experiments of the Stanford Heuristic Programming Project, Addison Wesley, 1984
- Buckley J.J. siler W. tucker D. Flops : a fuzzy expert system : applications and perspectives, Fuzzy logic in knowledge engineering, Verlag TUV 1986
- Buckley J.J. Theory of the fuzzy controller : An introduction. University of Alabama, 1990
- Buckley J.J. Fuzzy controller : further limit theorems for linear control rules, FSS n°36, 1990
- Buckley J.J. Ying H. Fuzzy controller : limit theorems for linear fuzzy control rules. Automatica vol.25 n°3 p.469, 1989
- Buckley J.J. Sugeno type controllers are universal controllers FSS n°53 p.299-303, 1993
- Buisson J.C. Farency H. Prade H. Dealing with imprecision and uncertainty in the expert system Diabeto III (Actes du 2° colloque d'intelligence artificielle de Marseille) p.705-721, 1986
- Bühler H. Réglage par logique floue, Presses polytechniques et univ. romandes, 1994
- Cerf R. Une théorie asymptotique des algorithmes génétiques, Thèse Univ. Montpellier, 1994
- Chateaneuf A. Jaffray J.Y. Some characterizations of lower probabilities and other monotone capacities through the use of Möbius inversion, Mathematical Social Sciences n°17, 1989
- Chi Z. Yan H. ID3-derived fuzzy rules and optimized defuzzification for handwritten numeral recognition, IEEE transactions on fuzzy systems, vol 4 n°1 p.24-31, 1996
- Cho K.B. Wang B.H. Radial basis function based adaptative fuzzy systems and their application to system identification and prediction, FSS vol 83 p.325-339, 1996
- Chockalingam T. Arunkumar S. Genetic algorithm based heuristics for the mapping problem, Int. Journal Computers and Op. Research vol 22, p.55-64, 1995
- Choquet G. Théorie des capacités, Annales de l'institut Fourier 5, 1953
- Chuen Chien Lee Fuzzy logic in control systems : fuzzy logic controller IEEE Transactions n° 20, 1990
- Colmerauer A. Giannesini F Kanoui H. Pasero R. Van Caneghem M. Prolog, Inter Editions, 1985
- Cooper G.F. Nestor : a computer-based medical diagnostic aid that integrates causal and probabilistic knowledge, Ph.D. Stanford Univ. 1984
- Cordón O. Herrera F. A general study on genetic fuzzy systems, Genetic Algorithms in engineering and computer science, p.33-57 Wiley 1995
- Cordón O. Herrera O. Peregrin A. Applicability of the fuzzy operators in the design of fuzzy logic controllers, FSS vol 86 p.15-41, 1997
- Daniel M. Algebraic structures related to Dempster-Shafer theory, IPMU p.71-76, 1994
- Daniel M. Algebraic properties of structures related to Dempster-Shafer theory, IPMU p.441-447, 1996
- De Glas M. representation of Lukasiewicz' many-valued algebras; □he atomic case, FSS n°14, 1984
- DeKleer J. An assumption-based truth maintenance system, Artificial intelligence n°28 p.127-162, 1986
- Delahaye J.P. Information, complexité et hasard, Hermès 1994
- Dempster A.P. Upper and lower probability. Annals of mathematical statistics n°38 p.325-339, 1967
- Desprès S. Un apport à la conception de systèmes à base de connaissances : les opérations de déduction floue, Thèse Paris VI 1988
- Dessales J.L. L'odrateur génétique, Hermès 1996
- Desvignes S. Introduction à l'utilisation de MVL Rapport Laforia Paris VI n°92/29, 1992
- Doyle J. A A truth maintenance system, Artificial intelligence n°12 p.231-272, 1979
- Dubois D. Prade H. FSS n°2 1979
- Dubois D. Prade H. Théorie des possibilités Masson, 1985
- Dubois D. Prade H. Combination of fuzzy information in the framework of possibility theory, Data fusion in robotics and machine intelligence, Academic press New York p.481-505, 1992
- Dubois D. Prade H. La fusion d'informations imprécises, Traitement du signal vol II n°6 p.447-458 1994
- Dubois D. Prade H. Zimmermann H.J. Handbook of fuzzy sets and possibility theory, Kluwer academic, 1997
- Dubois D. Godo L. Lopez de Mantara R. Prade H. Qualitative reasoning with imprecise probabilities Journal of Intelligent Information and System 2 p.319-363, 1993
- Dubuisson B. Oppenham S. Distances entre sous ensembles flous et classification hiérarchique relative ascendante, Journée CNRS Applications de la théorie du flou, 1980
- Elman J.L. Finding structure in time, Cognitive science vol.14 p.179-211, 1990
- Farency H. Prade H. Default and inexact reasoning with possibility degrees (Rapport LSI n°247) 1986
- Farency H. Prade H. WyssE. Approximate reasoning in a rule-based expert system using possibility theory : a case study. (Information processing 86 North Holland) 1986
- Farinhas del Cerro Logique temporelle et valeur de programmes parallèles, Masson
- Felix R. Entscheidung bei qualitativen Zielen, Dissertation Univ. Dortmund, 1991
- Fonseca C.M. Fleming P.J. Genetic algorithms for multiobjective optimization : formulation, discussion and generalization, Proc. of the 5th conf. on the GA and their applications 1993
- Forbus Gentner Learning physical domains, Machine learning vol.2 (Morgan Kaufmann publishers) 1986
- Foulloy L. Benoit E. Capteurs flous multicomposantes : application à la reconnaissance des couleurs Actes des journées "Applications des ensembles flous" Nîmes p.167-175, 1993
- Froidevaux C. ISA hierarchies with exception, 5° congrès AFCET Rec. des formes et IA p.1127-1138, 1985
- Gabbay D. Intuitionistic basis for non-monotonic logic, Proc. Conf. on automated deduction, Lectures note on computer science n°6 p.260-273, Springer Verlag 1982
- Gacogne L. Contribution à la représentation des connaissances floues et mise en oeuvre d'un système de raisonnement approché, Thèse ParisVII, 1990
- Gacogne L. An extension of the possibility theory in view of the formalization of approximate reasoning. Lectures notes in computer science n° 548 p.176-181, 1991
- Gacogne L. About a foundation of the Dempster's rule, Rapport 93/27 Laforia, 1993
- Gacogne L. Apprentissage génétique global d'un contrôleur flou à deux variables basé sur la simulation d'un véhicule autonome, Actes IPMU p.1099-1104, 1994
- Gacogne L. About the fitness of fuzzy controllers whose rules are learned by genetic algorithms. EUFIT p.1523-1531, Aachen, 1994
- Gacogne L. Research of Pareto set by genetic algorithm, application to multicriteria optimization of fuzzy controller, EUFIT 97
- Gedjati Guessoum F. Résolution par des heuristiques dynamiques et des algorithmes génétiques du problème d'ordonnement de type job-shop généralisé. Thèse Paris VI, 1994

- Geno H. Fujiwara Y. Yoneda H. Fukushima K. Human sensory perception oriented image processing in color copy system. Proceedings of the international conference on fuzzy logic and neural networks Iizuka Japan p.423-427, 1990
- Geyer-Shulz A. Fuzzy rule based expert system and genetic machine learning, Physica-Verlag 1995
- Ginsberg M.L. Non monotonic reasoning using Dempster's rule, Proceedings of the fourth national conference on AI p.126-129, 1984
- Ginsberg M. Multivalued logics : a uniform approach to reasoning in artificial intelligence, Computational Intelligence 4 p.265-316, 1988
- Ginsberg M. A circumscriptive theorem prover, Artificial intelligence 39 p.209-230, 1989
- Glorennec P.Y. "An evolutionary neuro fuzzy network" Proceedings of the 4th neuro-Nîmes, 1991
- Glorennec P.Y. Applications des algorithmes génétiques pour l'optimisation des fonctions d'appartenance d'un réseau neuro-flou, Deuxièmes journées sur les applications des ensembles flous, p.219-226 Nîmes, 1992
- Glover Future paths for integer programming and links to artificial intelligence, Computer and operational research, vol 5 p.340-349, 1986
- Gochet P. Gribomont P. Logique volume II, Hermes
- Godo L. Jacas J. Valverde L. Fuzzy values in fuzzy logic Proc. 3° IFSA conf. Seattle p.829-832, 1989
- Goldberg D.E. Richardson J. Genetic algorithms with sharing for multimodal function optimization, GA and their applications p.41-49 Hillsdale New Jersey Lawrence Erlbaum ass., 1987
- Grabisch M. Le regroupement et la classification floue, lettre du club CRIN logique floue, mai 1992
- Grabisch M. On equivalence classes of fuzzy connectives - The case of fuzzy integrals, IEEE Trans. on fuzzy systems, vol.3 n°1, p.96-109, 1995
- Grégoire E. Logiques non monotones et Intelligence artificielle, Hermès 1990
- Gustafson D.E. Kessel W. Fuzzy clustering with a fuzzy covariance matrix, Proc. IEEE, vol. 2 p.761-766, 1979
- Hölle U. Klement E.P. Non classical logics and their applications to fuzzy subsets, Series B vol 32 Kluwer ac. publishers 1995
- Hornik K. Stinchcombe M. White M. Multilayer feedforward networks are universal approximators, Dep. of economics Univ. of California San Diego 1988
- Hornik K. Multilayer feedforward networks are universal approximators, Neural networks vol 2 p.359-366, 1989
- Hosaka A. Tanigushi M. Ueki S. Kurami K. Hattori A. Yamada K. Steering control of an autonomous vehicle using a fuzzy logic controller Rapport Nissan, 1989
- Ishigashi H. Shirai T. Nagasajka K. Miyoshi T. Neuro-fuzzy ID3 : a method of inducing fuzzy decision trees with linear programming for maximizing entropy and an algebraic method for incremental learning, FSS vol 81 p.157-167, 1996
- Jeffrey R. Probable knowledge, The problem of inductive logic, Ed. Lakatos North Holland, 1968
- Kacprzyk Hybrid genetic algorithms, Int. Journal of intelligent systems vol 11 n°8, 1996
- Kamada H. Yoshida M. A visual control system using image processing and fuzzy system IEEE round table discussion on vision-based vehicle guidance Tokyo 1990
- Kameyama M. Higuchi T. Multiple valued logic and special purpose processors : overview and future, Proc. of the 12th Int. symposium on multiple valued logic p.289-292, 1982
- Kaufmann A. Introduction à la théorie des sous ensembles flous, tome 1, Masson, 1973
- Kaufmann A. Introduction à la théorie des sous ensembles flous, tome 3, Masson, 1977
- Kelman A. Optimisation floue de trajectoires, préparation de missions aériennes. Actes des journées "Applications des ensembles flous" Nîmes p.137-144 1993
- Kelman A. Yager R.R. Compatibilité et agrégation partielle : méthode de fusion modulaire, Rencontres francophones sur la logique floue et ses applications, Paris, 1995
- Kim H. Dickerson J. Kosko B. Fuzzy throttle and brake control for platoons of smart carts, FSS vol 84 p.209-234, 1996
- Kiszka J. Kochanska M. Sliwinska D. The influence of some parameters on the accuracy of a fuzzy model. Industrial applications of fuzzy control M. Sugeno ed. North Holland p.187, 1985
- Klir C. Fuzzy sets, uncertainty and information Prentice Hall 1988
- Kohonen T. Self organization and associative memory, Springer Verlag 1989
- Kohout H. Fuzzy implications and generalized quantifiers, Int. journal of uncertainty and fuzziness, vol 4 n°3, 1996
- Kosko B. Fuzzy systems and neural networks, Prentice Hall 1992
- Kosko B. Fuzzy systems are universal approximators, IEEE conf. p.1153-1162, 1992
- Koza Genetic programming II, MIT Press, 1994
- Krishnakumar K. Micro-genetic algorithms for stationary and non-stationary function optimization, Proc. SPIE International Conf. Adaptive systems Philadelphia, 1989
- Krishnapuram R. Fuzzy clustering methods in computer vision EUFIT p.720-730, 1993
- Krishnapuram R. Frigui H. Nasraoui O. Fuzzy and possibilistic shell clustering algorithms and their application to boundary detection and surface approximation, IEEE Trans. on fuzzy systems vol.3 n°1 p.29-60, 1995
- Krivine J.L. Théorie axiomatique des ensembles, PUF 1972
- Lasseigne R. Rougemont M. Logique et fondements de l'informatique, Hermès 1993
- Léa Sombé Raisonements sur des informations incomplètes, Revue IA n°2 ou éditions Hermès 1989
- Mac Culoch W.S. Pitts W.H. A logical calculus of the ideas imminent in nervous activity, Bulletin of mathematical biophysics vol 5 p.115-133, 1943
- Mac Dermott Non monotonic logic II, Journal of ACM n°29 p.34-57, 1982
- Mahler P.E. St Clair D. Uncertain reasoning in an ID3 machine learning framework, Proc. of the second IEEE International Conference on fuzzy systems, p.7-12, 1993
- Mamdani E.H. Application of fuzzy algorithms for control of simple dynamic plant, IEEE vol 121, 1974
- Mamdani E.H. Fuzzy reasoning and its applications Academic press p.325, 1981
- Marsala C. Arbres de décision et sous-ensembles flous, Rapport 94/21 Laforia Université Paris VI, 1994
- Marsala C. Bouchon-Meunier B. Fuzzy Partitionning using mathematical morphology in a learning scheme, Fuzz'IEEE 1996
- Martin-Clouaire R. Prade H. 86 SPII-1 A simple inference engine capable of accomodation both imprecision and uncertainty. Computer-assisted decision making (North Holland 86)
- Martin-Löf Constructive mathematics and computer programming, Proc. 6th congress of logic methodology and philosophy of science p.153-175, North Holland, 1982
- Menger K. Statistical metrics, Proc. of the national academy of sciences USA 28, p.535-537, 1942
- Michalski R.S. A theory and methodology of inductive reasoning, Machine Learning vol 1 p.83-134, Morgan Kaufmann 1983
- Milgram M. reconnaissance des formes, méthodes numériques et connexionistes, Armand Colin, 1993

- Miller G. Todd P. Hedge S. Designing neural networks using genetic algorithms, 3rd International Conference on GA, Morgan-Kaufmann 1989
- Mizumoto Masaharu Representations of fuzzy connectives FSS 31, 1989
- Monod J. Le hasard et la nécessité, Seuil 1970
- Moore R.C. Semantical considerations on non monotonic logic, IJCAI p272-279, 1983
- Moore R.C. Autoepistemic logic, Non standard logics for automatic reasoning, Ac. Press p.105-136, 1988
- Mukaidono M. Yasui H. Postulates and proposals of fuzzy Prolog EUFIT p.1080-1086, 1994
- Murakami S. Maeda M. Automobile speed control system using a fuzzy logic controller. Industrial applications of fuzzy control M.Sugeno ed. North Holland p.105, 1985
- Mustawi M.T. On the training of radial basis function classifiers, Neural networks, vol.5 p.595-605, 1992
- Nascimento S. Moura-Pirès F. A genetic fuzzy C-means algorithm, Proc. IPMU p.745-750, 1996
- Neapolitan R. Probabilistic reasoning in expert systems, Wiley, 1990
- Nguyen H.T. Kreinovich V. On approximation of fuzzy controls by fuzzy systems IFSA p.1414-1417, 1993
- Nguyen H.T. Sugeno Tong Yager R.R. Theoretical aspects of fuzzy control, Wiley 1995
- Nozaki K. Ishibushi H. Tanaka H. A simple but powerful heuristic method for generating fuzzy rules from numerical data, FSS vol 86 p.251-270, 1997
- OFTA Logique floue (ouvrage collectif) Masson 1994
- O'Hagan M. Using maximum entropy-ordered weighted averaging to construct a fuzzy neuron, Proc. 24th IEEE conf. on signals, systems and computers, p.618-623, 1990
- Oussalah M. Maaref H. Barret C. Kalman filter and possibility theory for the localization of a mobile robot, EUFIT p.1650-1655, 1996
- Pavelka J. On fuzzy logic, Zeitschr. fundamental mathematical logik und grundlagen des math. n°25, 1979
- Pawlak Z. Rough sets, International journal of information and computer sciences vol 11 p. 341-356, 1982
- Pawlak Z. Rough sets, Theoretical aspects of reasoning about data, Warsaw University of Technology, 1990
- Pearl J. Probabilistic reasoning in intelligent systems, Morgan Kaufmann eds, 1988
- Pearl J. Heuristics, Addison Wesley 1985, trad. chez Hermès 1996
- Perny P. Modélisation, agrégation et exploitation de préférences floues dans une problématique de rangement, Thèse Paris Dauphine 1992
- Person P. Les réseaux bayésiens : un nouvel outil de l'intelligence artificielle, Thèse, Université Paris VI, 1991
- Picques J.D. Uncertainty management system for embedded applications IPMU p.336-341, 1994
- Pivert O. Contribution à l'interrogation flexible de bases de données : expression et évaluation de requêtes floues. Thèse Univ. Rennes I, 1991
- Quinlan J.R. Induction of decision trees, Machine learning vol.1, 1986
- Ralescu A. Bouchon-Meunier B. Ralescu D. Combining fuzzy quantifiers, Fuzzy logic for the applications to complex systems, World Scientific p.246-252, 1995
- Ramdani Système d'induction formelle à base de connaissances imprécises, Thèse Univ. Paris VI 1994
- Rao M. Prahad V. A tunable fuzzy logic controller for vehicle-active suspension systems, FSS vol 85 p.11-21, 1997
- Reiter R. A logic for default reasoning, Artificial Intelligence n°3 Bobrow ed. 1980
- Renhou L. Yi Z. Fuzzy logic controller based on genetic algorithm, FSS 83 p.1-10, 1996
- Rescher N. Many-valued logic, Mac Graw Hill, 1966
- Rifqi M. Mesures de similitude et leur agrégation, Rencontres francophones sur la logique floue et ses applications, Paris, 1995
- Robert L. Jani Y. Fuzzy logic in autonomous orbital operations, International Journal of Approximate reasoning, p.151-184, 1992
- Rosenblatt F. The perceptron : a probabilistic model for information storage and organization in the brain, Psychological review vol. 65 p.386-408, 1958
- Rumelhart D. Jordan M.I. Internal world models and supervised learning, Proceedings of the eight international workshop on machine learning p.70-74, 1991
- Ruspini E. H. A new approach to clustering, Information Control 15 p.22-32, 1969
- Ruspini E. Numerical methods for fuzzy clustering, Information Sciences 2 p319-350, 1970
- Sanchez E. Equation de relations floues Thèse de biologie humaine Faculté de médecine de Marseille, 1974
- Sanchez E. Logique floue et réseaux neuronaux en intelligence artificielle et reconnaissances des formes, Science Technique Technologie n°19 1991
- Schmitt M. Mattioli J. Morphologie mathématique, Masson 1994
- Schulte C.M. Genetic algorithm for prototype based fuzzy clustering EUFIT p.913-921, 1994
- Schwefel H.P. Numerical Optimization of computer models, J.Wiley 1981
- Schwefel H.P. Systems analysis, systems design and evolutionary strategies, System analysis, Modeling and Simulation vol.7 p.853-864, 1990
- Schwefel H.P. Evolution and optimum seeking. Sixth generation computer technology series. Wiley, 1995
- Schweitzer B. Sklar A. Probabilistic metric spaces, North Holland 1983
- Shafer G. A mathematical theory of evidence Princeton university press, 1976
- Shafer G. The art of causal conjecture, MIT Press, 1996
- Shahin A. Demko C. Coisne D. Extraction floue de paramètres sur des images, influence du nombre de régions et du choix des opérateurs triangulaires, Actes LFA p. 9-16, 1995
- Siler W. Ying H. Fuzzy control theory : the linear case FSS n°33 p.275, 1989
- Simon J.C. La reconnaissance des formes par algorithmes, Masson 1985
- Simon C. Levrat E. Brémont J. Sabourin R. Codage d'images de signatures manuscrites pour la vérification hors ligne, Rencontres francophones sur la logique floue et ses applications p.23-30 Cépaduès 1996
- Smets P. Probability of fuzzy event : an axiomatic approach FSS n°7, 1982
- Smets P. Decisions and belief functions, Rap. techn. 90-10 Institut de rech. interdisciplinaires et de dév. en IA Univ. libre de Bruxelles, 1990
- Smets P. Mamdani E.H. Dubois D. Prade H. Non-standard logics for automated reasoning, Academic Press p.253-286, 1988
- Smets P. Constructing the pignistic probability function in a context of uncertainty, Uncertainty and Artificial Intelligence vol 5 p.29-39 North Holland 1990
- Sugeno Theory of fuzzy integrals and its applications, Doct.thesis Tokyo inst. of tech. 1974
- Sugeno M. An introductory survey of fuzzy control. Information and science n°36 p.59, 1985
- Sugeno M. Murakami K. An experimental study on fuzzy parking control using a model car FSS n° 32 p.207 1989
- Sugeno M. Nishida N. Fuzzy control of model car FSS n°16 p.103, 1985

- Sugeno M. Murohushi T. Nishino J. Miwa H. Helicopter flight control based on fuzzy logic, Proc. of fuzzy engineering towards human friendly systems IEEE, 1991
- Sugeno M. Gyei-Kark Park An approach to linguistic instruction based learning, Uncertainty fuzzyness and knowledge based systems vol 1 p.19-56, 1993
- Terano T. Asai K. Sugeno M. Fuzzy systems theory and its applications, Academic Press 1991
- Tong R.M. A retrospective view of fuzzy control systems FSS n°14 p.199, 1984
- Tong J.R. Logique floue, Hermès 1995
- Turksen I.B. Four methods of approximate reasoning with interval-valued fuzzy sets, in International journal of approximate reasoning vol.3, 1989
- Valenzuela-Rendón M. The fuzzy classifier system, a classifier system for continuous varying variables, Belew and Booker p.346-353, 1991
- Vincent N. Dujet C. Modus ponens and inference in control IPMU p.102-107, 1994
- Wang L. X. A course in fuzzy systems and control, Prentice Hall, 1997
- Watkins C. Q-learning, technical note, Machine learning vol.8 p.279-292, 1992
- Willaeys D. Malvache N. Contribution of the fuzzy sets theory and applications. Gupta Ragade ed. North Holland, 1979
- Willaeys D. Malvache N. Hammad Utilization of fuzzy sets for system modelling and control. IEEE Conference Decision and Control New Orleans, 1977
- Yager R.R. On general class of fuzzy connectives, in Fuzzy Sets and Systems n°4, 1980
- Yager R.R. Reasoning with uncertainty for expert systems Proc. of the 9° conf. IJCAI Los Angeles, 1985
- Yager R. "Hierarchical representation of fuzzy if-then rules" Proce. IPMU Conference Mallorca 1992
- Yager R. On structures for fuzzy modeling and control, Technical report MII 1213 Iona College 1992
- Yager R.R. Entropy and specificity in a mathematical theory of evidence, Int. Journal general Systems n°9 p.249-260, 1993
- Yager R.R. Filev D.P. Including probabilistic uncertainty in fuzzy logic controller modeling using Dempster-Shafer theory, IEEE Trans., Man and Cybernetics 25 p.3-28, 1995
- Yagishita O. Itoh O. Sugeno M. Applications of fuzzy reasoning to water purification process. Industrial applications of fuzzy control M.Sugeno ed. North Holland, 1985
- Yamakawa T. High speed fuzzy controller hardware system : the mega FIPS machine. Information and science n°45 p.113, 1988
- Yamakawa T. Stabilization of an inverted pendulum by a high speed fuzzy controller hardware system . FSS n° 32 p.161-180, 1989
- Yamakawa T. Uchino E. Miki T. Kusagani O. A neofuzzy neuron model and its applications to systems identification and prediction of system behavior, Proc. of the 2nd conf. on fuzzy logic and neural networks p.477-483, 1992
- Yasunobu S. Miyamoto S. Automatic train operation system by predictive fuzzy control. Industrial applications of fuzzy control M.Sugeno ed. North Holland 1985
- Yuan Y. Zhuang H. A genetic algorithm for generating fuzzy classification rules, FSS vol 84 p.1-19, 1996
- Zadeh L.A. Fuzzy sets Information and control n°8 p.338-353, 1965
- Zadeh L.A. Outline of a new approach to the analysis of complex systems and decision processes IEEE Trans system Man Cybernetic n°1 p.28-44, 1973
- Zadeh L.A. Fuzzy sets as a basis for a theory of possibility FSS n°1 p.3-28, 1978
- Zadeh L.A. PRUF Possibilistic relational universal fuzzy : A meaning representation language for natural languages, Int. journal of man-machine studies n°10 p.395-460, 1978
- Zadeh L.A. A theory of approximate reasoning machine intelligence n°9 p.149-194, 1979
- Zadeh L.A. The concept of linguistic variable and its application to approximate reasoning, Fuzzy sets and applications : selected papers p.219-327, Wiley 1987
- Zalila Z. Contrôle longitudinal d'un véhicule autonome par régulateur hybride flou classique, IPMU p.478-484, 1994
- Zalila Z. Système copilote flou pour le contrôle transversal d'un véhicule automobile, Rencontres francophones sur la logique floue et ses applications p.277-285, 1996
- Zimmermann H.J. Fuzzy set theory and its applications, Kluwer academic publishers, 1991