# Etude de règles incertaines et imprécises satisfaisant certaines contraintes.

T. Pontet
INSA
Département Informatique. Bât. 502
20 av. Albert Einstein
69100 Villeurbanne
Tel : 72438383 poste 5928

## 1. Introduction.

Notre propos est d'explorer un certain nombre de méthodes de déduction en raisonnement incertain et imprécis. Ainsi, à partir de connaissances factuelles initiales (éventuellement imprécises), nous devons pouvoir déterminer certains faits nouveaux (éventuellement imprécis aussi) par l'utilisation de méthodes d'inférences. La première partie de cette étude va porter sur une réflexion sur ces méthodes. Celles-ci seront paramètrées et le choix de ces paramètres permettra de vérifier certaines contraintes. Ce sont ces contraintes qui détermineront certains résultats voulus pour les règles de déduction. Nous donnerons des informations générales sur les contraintes et leurs satisfactions dans la deuxième partie, alors qu'en troisième partie, nous chercherons des satisfactions particulières de contraintes particulières. C'est ainsi que l'on pourra construire des règles du type "si X est A alors Y est B" (modus ponens généralisé), "plus X est A, plus Y est B" (règles graduelles), "plus X est A, plus *q*".

## 2. Méthodes d'inférence.

### 2.1. Généralités.

Soient deux faits $p$ et $q$. Notre problème est de lier l'incertitude (ou l'imprécision) de $q$ à celle de $p$.

Toutes méthodes pourraient être employées mais certaines conditions de continuité et de monotonie nous semblent importantes à satisfaire. Ainsi, nous pourrions imposer que $\Pi(p \wedge q)$ soit fonction croissante et continue de $\Pi(p)$. Pour satisfaire ces hypothèses, nous poserons (analogie avec le calcul des probabilités) :

$$\Pi(p \wedge q) = \Pi(q \mid p) \otimes \Pi(p)$$

$\otimes$ étant un opérateur continu (appelé norme triangulaire) tel que :

- $\forall s, t \in [0,1], s \otimes t \leq \min(s,t)$
- $\forall s, t, u, v \in [0,1]$, si $s \leq u$ et $t \leq v$ alors $s \otimes t \leq u \otimes v$
- $\forall s \in [0,1], s \otimes 0 = 0 \otimes s = 0$
- $\forall s \in [0,1], s \otimes 1 = 1 \otimes s = s$

Cela permettra la détermination de méthodes de déduction purement incertaines (non imprécises) et de méthodes gérant l'imprécision (règles floues).

## 2.2. Règles floues.

Nous appelerons règles floues des règles manipulant (en prémisse ou en conclusion) des prédicats flous (du type "X est A", X étant une variable prenant ses valeurs dans un ensemble S et A un sous-ensemble flou de S). Deux cas pourront se produire :

### 2.2.1. Lien entre un prédicat flou et une proposition quelconque.

#### 2.2.1.1. Première méthode.

A partir de la formule générale présentée en 2.1. et tenant compte que :

$$\Pi(q) = \Pi((q \wedge p) \vee (q \wedge \neg p))$$

nous pouvons obtenir :

$$\begin{cases} \Pi(q) = \max(\Pi(q \mid p) \otimes \Pi(p), \Pi(q \mid \neg p) \otimes \Pi(\neg p)) \\ \mathrm{N}(q) = \min(\mathrm{N}(q \mid p) \oslash \mathrm{N}(\neg p), \mathrm{N}(q \mid \neg p) \oslash \mathrm{N}(p)) \end{cases}$$

avec :

$$\forall u, v \in [0,1], u \oslash v = 1 - (1-u) \otimes (1-v)$$

Nous appliquerons donc directement ces formules avec $p$="X est A" ou $q$="Y est B". Les nécessité et possibilité de $p$ (s'il est flou) pourront se calculer à l'aide des formules de filtrage flou (de Zadeh).

#### 2.2.1.2. Deuxième méthode.

Elle s'appliquera pour $p$="X est A". Nous remarquerons que :

$$\Pi(q) = \Pi\left(\bigvee_{x \in S} ((X = x) \wedge q)\right)$$

et ainsi :

$$\begin{cases} \Pi(q) = \sup_{x \in S} (\Pi_{q|X}(x) \otimes \pi_X(x)) \\ N(q) = \inf_{x \in S} (N_{q|X}(x) \oslash (1 - \pi_X(x))) \end{cases}$$

**2.2.1.3. Troisième méthode.**

Elle s'appliquera pour $q$="Y est B". On aura :

$$\forall y \in T, \Pi_Y(y) = \max(\Pi_{Y|p}(y) \otimes \Pi(p), \Pi_{Y|\neg p}(y) \otimes \Pi(\neg p))$$

**2.2.2. Lien entre deux prédicats flous.**

**2.2.2.1. Première méthode.**

C'est la première méthode précédente avec $p$="X est A" et $q$="Y est B".

**2.2.2.2. Deuxième méthode.**

C'est la deuxième méthode précédente avec $q$="Y est B".

**2.2.2.3. Troisième méthode.**

C'est la troisième méthode précédente avec $p$="X est A" (et utilisation du filtrage flou).

**2.2.2.4. Quatrième méthode.**

On démontre :

$$\forall y \in T, \pi_Y(y) = \sup_{x \in S} (\pi_{Y|X}(y, x) \otimes \pi_X(x))$$

# 3. Généralités sur les contraintes.

## 3.1. Définitions.

Une contrainte est un couple (condition, conclusion).

Une contrainte sera dite satisfaite par une règle ssi celle-ci déduit la conclusion de la contrainte lorsque sa condition est vérifiée.

## 3.2. Satisfaction de contraintes.

### 3.2.1. Contrainte unique.

Soit une contrainte. Soit un type de règles pouvant éventuellement satisfaire cette contrainte (par exemple, si la contrainte utilise en partie condition un prédicat flou et une proposition quelconque en partie conclusion, seules les règles permettant de déduire *q* à partir de *p*="X est A" pourront être utilisées). Nous chercherons des conditions suffisantes (d'où l'obtention d'un sous-ensemble de toute les règles du type choisi satisfaisant la contrainte), nécessaires (ensemble incluant l'ensemble de toute les règles satisfaisant la contrainte), ou nécessaires et suffisantes sur les paramètres des règles.

### 3.2.2. Ensemble de contraintes.

#### 3.2.2.1. Exclusivité des contraintes.

Dans ce cas, chaque contrainte sera traitée indépendamment. Nous obtiendrons en ensemble de conditions (suffisantes, nécessaires ou nécessaires et suffisantes) sur les paramètres des règles devant satisfaire cet ensemble de contraintes (indépendamment).

#### 3.2.2.2. Non-exclusivité des contraintes.

Nous devrons d'abord définir une méthode de combinaison des conclusions.

##### 3.2.2.2.1. Test de toutes combinaisons de contraintes.

Nous chercherons pour chaque combinaison de contraintes non exclusives, les conditions sur les paramètres des règles devant satisfaire (simultanément) ce sous-ensemble de contraintes (obtention de la combinaison des conclusions de ces contraintes lorsque leurs conditions sont vérifiées). On obtiendra un ensemble de conditions.

##### 3.2.2.2.2. Méthode de remplacement.

On remplacera chaque contrainte par un sous-ensemble de règles la satisfaisant. On obtient un ensemble d'ensembles de règles. On cherchera ensuite les conditions sur les paramètres des règles donnant dans tous les cas de configuration les mêmes résultats que l'ensemble d'ensembles de règles initial. Des conditions nécessaires seront faciles à trouver alors que des conditions suffisantes risquent de poser des problèmes (il faudra pour cela vérifier que pour toutes les règles les satisfaisant, des résultats identiques pourront être obtenus par n'importe quel ensemble de règles satisfaisant les contraintes).

**3.2.2.2.3. Remarque.**

La seconde méthode ne donnera pas en général les mêmes résultats que la première. En effet, dans le cas où certaines conditions de contraintes sont vérifiées, la conclusion qu'elle permettra d'obtenir ne sera pas forcément la combinaison des conclusions de ces contraintes. La raison en est que, même si l'on a vérifié que les règles correspondant aux conditions (des contraintes) satisfaites donneront les conclusions de ces contraintes, les autres règles donneront éventuellement des conclusions venant se combiner et modifier ainsi la conclusion générale. Si l'on veut que les résultats de la première méthode soient obtenus par la seconde, il faut rajouter l'ensemble des contraintes suivantes liées à chaque contrainte initiale (condition,conclusion) : (condition non totalement vérifiée, aucune conclusion).

**3.2.2.3. Equivalence entre une contrainte C et un ensemble de contraintes E.**

**3.2.2.3.1. Identité des ensembles de règles.**

Identité entre l'ensemble des règles satisfaisant C et l'ensemble des règles satisfaisant E (avec première ou deuxième méthode).

**3.2.2.3.2. Vérification des contraintes.**

On vérifie que les règles satisfaisant C satisfont E (première ou deuxième méthode) et inversement, que les règles satisfaisant E (première ou deuxième méthode) satisfont C.

# 4. Détermination de règles satisfaisant certaines contraintes.

On prendra comme opérateur de combinaison des conclusions l'intersection-minimum.

## 4.1. Un mot sur le modus ponens généralisé.

Il correspond à la contrainte (X est A, Y est B). Toutes les méthodes pourront être employées et donneront des résultats différents dans les cas où la condition de le contrainte n'est pas vérifiée.

## 4.2. Règles graduelles.

En reprenant la terminologie de Dubois et Prade, nous appellerons ainsi les règles du genre "plus X est A, plus Y est B". Deux ensembles de contraintes semblent tout à fait correspondre à ce genre de règles. Nous allons les étudier pour des règles du quatrième type (en 2.2.2.4.) et vérifierons qu'elles sont équivalentes (dans ce cas).

### 4.2.1. Contrainte $(X \in A_\alpha, Y \in B_\alpha)$ pour $\alpha \in ]0,1]$.

#### 4.2.1.1. Ensemble des règles satisfaisant la contrainte.

Les paramètres (des règles de type 4) doivent satisfaire aux conditions nécessaires et suffisantes suivantes :

$$\forall (x,y) \in S \times T / \mu_A(x) \geq \alpha, \begin{cases} \mu_B(y) < \alpha \Rightarrow \pi^\alpha_{Y|X}(y,x) = 0 \\ \mu_B(y) \geq \alpha \Rightarrow \pi^\alpha_{Y|X}(y,x) = 1 \end{cases}$$

Remarquons que ces règles ne donneront pas, lorsque la condition de la contrainte sera vérifiée, de conclusion plus restrictive (mais l'incluant néanmoins) que celle de la contrainte (c'est le résultat de la seconde condition sur les règles).

#### 4.2.1.2. Ensemble des règles satisfaisant les contraintes pour tous $\alpha$.

Après avoir remarqué que les contraintes n'étaient pas exclusives, nous décidons d'appliquer la méthode de remplacement. Nous trouvons les conditions nécessaires et suffisantes suivantes (toujours pour des règles de type 4) :

$$\forall (x,y) \in S \times T, \begin{cases} \mu_A(x) > \mu_B(y) \Rightarrow \pi_{Y|X}(y,x) = 0 \\ \mu_A(x) = \mu_B(y) = 1 \Rightarrow \pi_{Y|X}(y,x) = 1 \end{cases}$$

#### 4.2.1.3. Contraintes supplémentaires.

De façon à obtenir les résultats de la méthode test de toutes combinaisons de contraintes, nous rajouterons les contraintes suivantes :

$\forall \alpha \in ]0,1], (X \notin A_\alpha, Y \in T)$

Les conditions pour $\alpha \in ]0,1]$ s'écrivent de la façon suivante :

$$\exists x \in S / \begin{cases} \mu_A(x) < \alpha \\ \pi_X(x) \neq 0 \end{cases}$$

Cela ne permettant pas de conclure, nous prendrons des contraintes avec les conditions plus restrictives :

$$\exists x \in S / \begin{cases} \mu_A(x) < \alpha \\ \pi_X(x) = 1 \end{cases}$$

ce qui nous permettra d'obtenir les conditions nécessaires et suffisantes :

$$\forall (x, y) \in S \times T, \mu_A(x) < \alpha \Rightarrow \pi^{\alpha}_{Y|X}(y, x) = 1$$

Ainsi les conditions nécessaires et suffisantes sur les règles satisfaisant l'ensemble des contraintes deviennent :

$$\forall (x, y) \in S \times T, \begin{cases} \mu_A(x) > \mu_B(y) \Rightarrow \pi_{Y|X}(y, x) = 0 \\ \mu_A(x) \leq \mu_B(y) \Rightarrow \pi_{Y|X}(y, x) = 1 \end{cases}$$

#### 4.2.2. Contrainte $(X = x, Y \in B_{\mu_A(x)})$ pour $x \in S$.

Nous chercherons l'ensemble des règles satisfaisant les contraintes pour tous $x$. Les contraintes étant exclusives, nous les traiterons indépendemment. Cela nous permettra de déduire que les règles (de type 4) les satisfaisant doivent satisfaire aux conditions nécessaires et suffisantes suivantes :

$$\forall (x, y) \in S \times T, \begin{cases} \mu_A(x) > \mu_B(y) \Rightarrow \pi_{Y|X}(y, x) = 0 \\ \mu_A(x) \leq \mu_B(y) \Rightarrow \pi_{Y|X}(y, x) = 1 \end{cases}$$

Ces contraintes sont donc équivalentes à l'ensemble des contraintes du 4.2.1.

### 4.3. Règles semi-graduelles.

C'est ainsi que nous appellerons des règles du genre "plus X est A, plus $q$" ("..., plus $q$ est certain"). Un certain nombre de contraintes seront alors testées pour un certain nombre de types de règles.

#### 4.3.1. Contrainte $(X \in A_{\alpha}, N(q) = \alpha)$ pour $\alpha \in ]0, 1]$.

##### 4.3.1.1. Ensemble des règles satisfaisant la contrainte.

Nous chercherons des règles de type 1 avec $N^{\alpha}(q | \neg p) = 0$ (cas particulier pour simplifier la démonstration).

Ces règles satisferont la contrainte si on pose que $N^\alpha(p) \geq \alpha$ lorsque la condition de la contrainte est vérifiée, $N^\alpha(q \mid p) = \alpha$ et $\Pi^\alpha(q \mid p) = 1$. Ces conditions sur les paramètres ne sont nécessaires que si nous ajoutons une contrainte supplémentaire sur le sens de variation (nécessité sur $q$ inférieure ou égale à $\alpha$ lorsque la condition de la contrainte n'est pas vérifiée).

Une de ces règles sera celle pour laquelle $N^\alpha(p) = 1 - \operatorname*{Sup}_{\mu_A(x)<\alpha}(\pi_X(x))$ ce qui est obtenu si $p$ est égal à $X \in A_\alpha$ et si l'on calcule sa nécessité par la formule de filtrage flou. Dans ce cas, si la condition de la contrainte est vérifiée, la nécessité de la prémisse de la règle est égale à 1 (donc est bien supérieure ou égale à $\alpha$).

Dans ce cas, on aura :

$$\begin{cases} N^\alpha(q) = \min\left(\alpha \oslash \left(1 - \operatorname*{Sup}_{\mu_A(x)\geq\alpha}(\pi_X(x))\right), 1 - \operatorname*{Sup}_{\mu_A(x)<\alpha}(\pi_X(x))\right) \\ \Pi^\alpha(q) = \max\left(\operatorname*{Sup}_{\mu_A(x)\geq\alpha}(\pi_X(x)), \Pi^\alpha(q \mid \neg p) \otimes \operatorname*{Sup}_{\mu_A(x)<\alpha}(\pi_X(x))\right) \end{cases}$$

suivi d'une renormalisation éventuelle.

#### 4.3.1.2. Ensemble des règles satisfaisant les contraintes pour tous $\alpha$.

Nous appliquerons la méthode de remplacement ce qui donnera :

$$\begin{cases} \Pi(q) = \operatorname*{Inf}_{\alpha\in]0,1]}(\Pi^\alpha(q)) \\ N(q) = \operatorname*{Sup}_{\alpha\in]0,1]}(N^\alpha(q)) \end{cases}$$

suivi d'une renormalisation éventuelle.

### 4.3.2. Contrainte ($X \in A_\alpha$, B est $\alpha$-certain pour Y) pour $\alpha \in ]0,1]$.

Ceci est (à une inégalité près) une formulation différente des contraintes précédentes pour lesquelles $q$="Y est B". B est $\alpha$-certain pour Y signifie en effet :

$$\forall y \in T, \pi_Y(y) \leq \max(\mu_B(y), 1 - \alpha)$$

#### 4.3.2.1. Ensemble des règles satisfaisant la contrainte.

Cherchons les règles de type 4 satisfaisantes. Elles seront telles que :

$$\forall(x,y)\in S\times T/\mu_A(x)\geq\alpha,\pi^{\alpha}_{Y|X}(y,x)\leq\max(\mu_B(y),1-\alpha)$$

Rajoutons maintenant les contraintes :

$$\forall\alpha\in]0,1],(X\notin A_\alpha,Y\in T)$$

ce qui donne :

$$\forall(x,y)\in S\times T,\mu_A(x)<\alpha\rightarrow\pi^{\alpha}_{Y|X}(y,x)=1$$

#### 4.3.2.2. Ensemble des règles satisfaisant les contraintes pour tous $\alpha$.

Elles vérifieront (type 4) :

$$\forall(x,y)\in S\times T,\pi_{Y|X}(y,x)\leq\max(\mu_B(y),1-\mu_A(x))$$

ce qui est différent des résultats obtenus en 4.3.1.

### 4.3.3. Contrainte $(X=x,N(q)=\mu_A(x))$ pour $x\in S$.

Nous rappellerons que les contraintes du 4.3.1. ont pu être vérifiées par des règles de type 1 pour lesquelles on pris $p$="$X\in A_\alpha$". Dans ce nouveau cas, des règles de type 1 avec $p$="X=x" ne fonctionneraient pas car la nécessité de $p$ serait dans la plupart des cas nulle. Par contre, nous remarquerons que les règles obtenues en 4.3.1. satisfont ces nouvelles contraintes.

### 4.3.4. Contrainte $(X=x,$ B est $\mu_A(x)$-certain pour Y$)$ pour $x\in S$.

L'ensemble des règles de type 4 satisfaisant les contraintes pour tous $x$ (indépendamment) est tel que :

$$\forall(x,y)\in S\times T,\pi_{Y|X}(y,x)\leq\max(\mu_B(y),1-\mu_A(x))$$

ce qui est identique aux règles correspondant aux contraintes du 4.3.2.

## 5. Conclusion.

Nous n'avons étudié dans ce papier que certaines contraintes et ne les avons appliquées que pour certaines formulations de règles. Remarquons que l'on pourrait faire une exploration la plus exhaustive possible (en utilisant toutes les formulations de règles "connues" et tous les types de contraintes "intéressants").

En particulier, nous pensons à des contraintes du type "plus *p*, plus Y est B". Nous pourrions ainsi recenser les "avantages" et "inconvénients" (qui dépendent de l'utilisation qu'on va en faire) de chaque type de règle pour chaque contrainte qu'on veut respecter et fournir ainsi un guide de choix.

## Bibliographie.


. Dubois D. & Prade H. [1987] Théorie des possibilités : applications à la représentation des connaissances en informatique, Masson, France.

. Prade H. [1988] Raisonner avec des règles d'inférence graduelle. Une approche basée sur les ensembles flous. Revue d'intelligence artificielle. Volume 2. no 2/1988, p. 29-44.