

Marie-Jeanne Lesot
 Christophe Marsala
 Jean-Noël Vittaut
 Gauvain Bourgne

LIP6 – Sorbonne Université

XAI – 2023-2024

*C'est quand on a raison qu'il est difficile
 de prouver qu'on n'a pas tort.*

Pierre Dac

Plan du cours

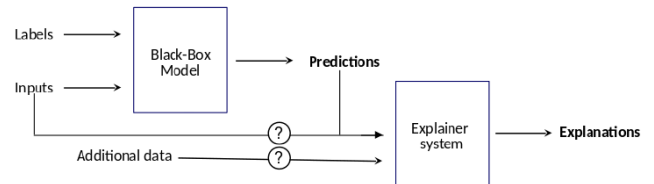
Approches par modèle de substitution
 contexte et rappels
 approche LORE

Apprentissage de règles floues

1 – Approches par modèle de substitution – contexte et rappels

Contexte

- Un modèle déjà construit : $f : \mathcal{D} \rightarrow \mathcal{Y}$
 - approche **agnostique** : on n'a que f (et on ne sait rien de plus)
 - boîte noire
 - hypothèse : on peut utiliser f autant que l'on veut
- Un exemple d'intérêt : $\mathbf{x} \in \mathcal{D}$
 - exemple à classer et pour lequel on voudrait une explication locale
- Approches post-hoc



Source : (Laugel, 2018)

Christophe Marsala – 2023-2024

XAI – cours 7 – 4

1 – Approches par modèle de substitution – contexte et rappels

1 – Approches par modèle de substitution – contexte et rappels

Les approches par modèle de substitution (surrogate)

- But : étant donné f (boîte noire), trouver un modèle interprétable à lui substituer pour fournir une explication à la décision $f(\mathbf{x})$ donnée pour \mathbf{x} .
 \Rightarrow substitut (**surrogate**) à f
- Cadre :
 - approche agnostique
 - approche globale ou locale
 - approche simple et interprétable
- Différentes approches
 - explications par "feature importance" : LIME, SHAP,...
 - explications par des règles : LORE, TREPAN, FoilTree, ANCHOR...

Rappels : LIME

- revoir le cours 3
- **LIME** : Local Interpretable Model-agnostic Explanations
 (Ribeiro et al., 2016)
 - génération d'un échantillon autour du centre des données
 - pondération des exemples selon leur proximité à \mathbf{x}
 - mise en évidence des caractéristiques (features) importantes
- Etapes principales :
 1. créer une base d'apprentissage étiquetée grâce à f
 2. pondérer les exemples à l'aide d'une mesure de proximité à \mathbf{x}
 3. apprendre un modèle interprétable g
 4. produire une explication pour $f(\mathbf{x})$ grâce à g
- Problèmes
 - aspect global de l'explication
 - interprétabilité ("peut mieux faire")

Une approche par substitution et explication contre-factuelle

- ▶ L'approche **LORE** : L^Ocal Rule-based Explain^Er (Guidotti et al., 2019)
- ▶ Combinaison des approches par substitution et contre-factuelle
 1. créer une base d'apprentissage dans le **voisinage** de \mathbf{x} et dont la classification par f couvre les 2 classes
 2. apprendre un **modèle interprétable** g :
⇒ choix de l'arbre de décision
 3. produire une explication pour $f(\mathbf{x})$ grâce à g
 - explication **contre-factuelle** dérivée de la structure de g
- ▶ Avantages
 - pas d'hyperparamètre lié au nombre de features (\neq LIME)
 - pas d'a priori (par exemple : discrétisation (\neq ANCHOR))

Génération de \mathbf{X}_x

- ▶ Algorithmes génétiques (Holland, 1975)
 - algorithmes évolutionnaires
- ▶ Approche inspirée par des lois de l'évolution
 - population d'individus qui évoluent de génération en génération
 - croisements entre individus
 - mutations d'individus : saut génétique
 - mesure de performance d'un individu : mesure de **fitness**
- ▶ Méthode d'optimisation d'une mesure de fitness M
 - trouver un optimum de M par une recherche aléatoire
 - croisement : mélange de propriétés de 2 individus pour tenter d'augmenter M
 - mutation : changer légèrement un individu pour tenter d'augmenter M

Algorithmes génétiques : l'algorithme général

1. étant donné
 - M la fonction à optimiser
 - taille de la population N
 - un taux de croisement pc et un taux de mutation pm
2. soit P_0 une population initial e à $t = 0$
3. sélectionner des chromosomes dans la population courante P_t
 - sélection selon leur valeur pour M
4. effectuer des croisements et des mutations pour obtenir P_{t+1}
5. recommencer en 3

Approche LORE : l'algorithme

1. Créer une base d'apprentissage \mathbf{X}_x dans le **voisinage** de \mathbf{x} et dont la classification par f couvre les 2 classes
 - génération de \mathbf{X}_x par **algorithme génétique**
 - 2 étapes : exemples positifs puis exemples négatifs (selon f)
 - fonction de fitness et opérateurs dédiés
2. Apprendre un **modèle interprétable** g
 - construction d'un arbre de décision g à partir de \mathbf{X}_x
3. Produire une explication pour $f(\mathbf{x})$ grâce à g
 - explication **contre-factuelle** dérivée de la structure de g
 - trouver la branche de g pour \mathbf{x} : classe $g(\mathbf{x})$
 - recenser les branches de g associées à la classe $\overline{g(\mathbf{x})}$
 - en déduire une explication pour $f(\mathbf{x})$

Algorithmes génétiques : éléments de base

- ▶ Contexte
 - soit la fonction $M : \mathcal{E} \rightarrow \mathbb{R}$ à optimiser (max)
 - \mathcal{E} : ensemble de solutions possibles pour M
- ▶ Population P_0 à l'instant t : sous-ensemble de \mathcal{E} de taille donnée
- ▶ Chromosome : représentation d'élément de \mathcal{E}
→ représenté pour l'approche génétique (vecteur de \mathcal{D})
- ▶ Opérations possibles
 - croisement (cross-over) : $c : \mathcal{D} \times \mathcal{D} \rightarrow \mathcal{D}$
 - mutation : $m : \mathcal{D} \rightarrow \mathcal{D}$
- ▶ Propriétés
 - aucune contrainte sur M
 - exploration aléatoire et guidée de l'espace des solutions
 - minimiser le risque de tomber dans un optimum local

Algorithmes génétiques : usage dans LORE (1)

- ▶ Génération de \mathbf{X}_x en 2 étapes
 1. générer des exemples proches de \mathbf{x} et de même classe ($f(\mathbf{x})$)
 2. générer des exemples proches de \mathbf{x} et de classe différente
- ▶ Mesures de fitness
 1. $M_{=}^{\mathbf{x}}(\mathbf{z}) = I_{f(\mathbf{x})=f(\mathbf{z})} + (1 - d(x, z)) - I_{\mathbf{x}=\mathbf{z}}$
 2. $M_{\neq}^{\mathbf{x}}(\mathbf{z}) = I_{f(\mathbf{x}) \neq f(\mathbf{z})} + (1 - d(x, z)) - I_{\mathbf{x}=\mathbf{z}}$avec :
 - $I_{true} = 1$ et $I_{false} = 0$
 - $d : \mathcal{E} \times \mathcal{E} \rightarrow [0, 1]$ une mesure de la distance
$$d(\mathbf{x}, \mathbf{z}) = \frac{h}{d} \text{match}(\mathbf{x}, \mathbf{z}) + \frac{d-h}{d} \text{norm}(\mathbf{x} - \mathbf{z})$$
$$\text{match}(\mathbf{x}, \mathbf{z}) : \text{comparaison sur les } h \text{ attributs catégoriels}$$
$$\text{norm}(\mathbf{x}, \mathbf{z}) : \text{norme euclidienne sur les } d - h \text{ attributs numériques}$$
- ▶ Interprétation de la fitness : on cherche des exemples \mathbf{z}
 - **proches** de \mathbf{x} ($1 - d(\mathbf{x}, \mathbf{z})$) et **différents** de \mathbf{x} ($-I_{\mathbf{x}=\mathbf{z}}$)
 - avec la **même classe** par f ou une **classe différente**

Algorithmes génétiques : usage dans LORE (2)

- Population initiale : duplication de l'exemple x
- Opérations de croisement et de mutation
 - croisement 2-points : tirage aléatoire de 2 attributs entre lesquels tous les attributs seront croisés
 - mutation : tirage aléatoire d'un attribut
 - la valeur de remplacement est tirée aléatoirement suivant la distribution des valeurs dans l'ensemble test
- Chromosome = exemple

parent 1	25	clerk	10k	yes
parent 2	30	other	5k	no
children 1	25	other	5k	yes
children 2	30	clerk	10k	no

Figure 1: Crossover.

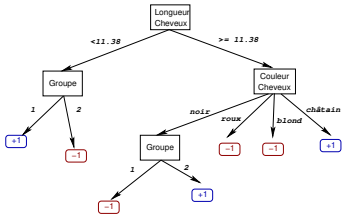
parent	25	clerk	10k	yes
children	27	clerk	7k	yes

Figure 2: Mutation

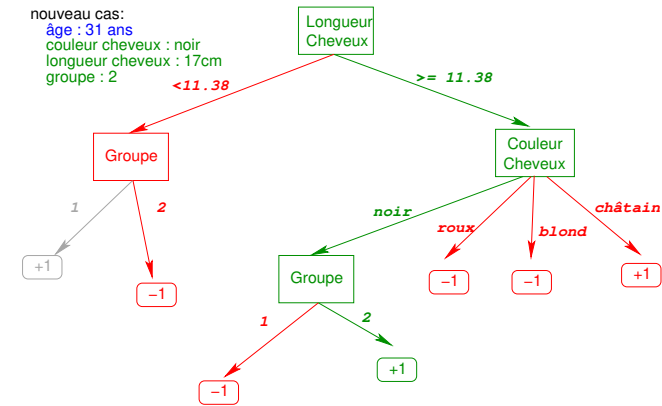
(Guidotti et al., 2018)

Approche LORE : l'algorithme

- Créer une base d'apprentissage X_x dans le voisinage de x et dont la classification par f couvre les 2 classes
 - génération de X_x par algorithme génétique
 - génération en 2 temps : exemples positifs puis exemples négatifs
 - fonction de fitness et opérateurs dédiés
- Apprendre un modèle interprétable g
 - construction d'un arbre de décision g à partir de X_x
 - algorithme classique



Rappel : classification avec un arbre de décision



Approche LORE : génération des exemples

Algorithm 2: GeneticNeigh($x, fitness, b, N, G, pc, pm$)

Input : x - instance to explain, b - black box, $fitness$ - fitness function, N - population size, G - # of generations, pc - crossover probability, pm - mutation probability

Output: Z - neighbors of x

```
1  $P_0 \leftarrow \{x | \forall 1 \dots N\}; i \leftarrow 0;$  // population init.
2  $evaluate(P_0, fitness, b);$  // evaluate population
3 while  $i < G$  do
4    $P_{i+1} \leftarrow select(P_i);$  // select sub-population
5    $P'_{i+1} \leftarrow crossover(P_{i+1}, pc);$  // mix records
6    $P''_{i+1} \leftarrow mutate(P'_{i+1}, pm);$  // perform mutations
7    $evaluate(P''_{i+1}, fitness, b);$  // evaluate population
8    $P_{i+1} = P''_{i+1}; i \leftarrow i + 1$  // update population
9 end
10  $Z \leftarrow P_i$  return  $Z;$ 
```

(Guidotti et al., 2018)

Approche LORE : algorithme

- Créer une base d'apprentissage X_x dans le voisinage de x et dont la classification par f couvre les 2 classes
 - génération de X_x par algorithme génétique
 - génération en 2 temps : exemples positifs puis exemples négatifs
 - fonction de fitness et opérateurs dédiés
- Apprendre un modèle interprétable g
 - construction d'un arbre de décision g à partir de X_x
 - algorithme classique
- Produire une explication pour $f(x)$ grâce à g
 - explication contre-factuelle dérivée de la structure de g
 - trouver la branche de g pour x : classe $g(x)$
 - recenser les branches de g associées à la classe $\overline{g(x)}$
 - en déduire une explication pour $f(x)$

Classification : ensemble de règles

- Règle de classification
 - si Longueur cheveux ≥ 11.38 et Couleur cheveux est noir et Groupe 2 alors classe +1
- Règles contre-exemples
 - si Longueur cheveux < 11.38 et Groupe 2 alors classe -1
 - si Longueur cheveux ≥ 11.38 et Couleur cheveux est roux alors classe -1
 - si Longueur cheveux ≥ 11.38 et Couleur cheveux est blond alors classe -1
 - si Longueur cheveux ≥ 11.38 et Couleur cheveux est châtain alors classe -1
 - si Longueur cheveux ≥ 11.38 et Couleur cheveux est noir et Groupe 1 alors classe -1

Approche LORE : explication contre-factuelle

- Soit r la règle déclenchée par \mathbf{x} pour donner $g(\mathbf{x})$
- Soit r' une règle représentant une branche produisant $\overline{g(\mathbf{x})}$
 - on note $n_{r'}$: nombre de tests de r' invalidés par \mathbf{x}
- Explication contre-factuelle : règle r_{best} qui minimise $n_{r'}$
 - il peut exister plusieurs telles règles
- Exemple précédent :
 - règle r :
 - si Longueur cheveux ≥ 11.38 et Couleur cheveux est noir et Groupe 2 alors classe +1
 - règles r' qui minimisent $n_{r'}$
 - si Longueur cheveux ≥ 11.38 et Couleur cheveux est roux alors classe -1
 - si Longueur cheveux ≥ 11.38 et Couleur cheveux est blond alors classe -1
 - si Longueur cheveux ≥ 11.38 et Couleur cheveux est châtain alors classe -1
 - si Longueur cheveux ≥ 11.38 et Couleur cheveux est noir et Groupe 1 alors classe -1

Approche LORE : explication contre-factuelle

- Explications pour notre exemple : il est de classe +1
 - car il est dans le groupe 2 (et pas le groupe 1)
 - car il a les cheveux noirs (et pas roux, blonds ou châains)

Approche LORE : quelques références

- "Factual and counterfactual explanations for black box decision making". R. Guidotti, A. Monreale, F. Giannotti, D. Pedreschi, S. Ruggieri, F. Turini. IEEE Intelligent Systems 34 (6), 14-23. 2019.
- "Local rule-based explanations of black box decision systems". R. Guidotti, A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini, F. Giannotti. arXiv 2018.

Approche LORE : explication contre-factuelle

- Soit r la règle représentant la branche fournissant $g(\mathbf{x})$
- Soit r' une règle représentant une branche telle que $\overline{g(\mathbf{x})}$
 - $n_{r'}$: nombre de tests de r' invalidés par \mathbf{x}
- Explication contre-factuelle :
 - règle r_{best} qui minimise $n_{r'}$
- Exemple précédent :
 - règle r :
 - si Longueur cheveux ≥ 11.38 et Couleur cheveux est noir et Groupe 2 alors classe +1
 - règles r' qui minimisent $n_{r'}$
 - si Longueur cheveux ≥ 11.38 et Couleur cheveux est roux alors classe -1
 - si Longueur cheveux ≥ 11.38 et Couleur cheveux est blond alors classe -1
 - si Longueur cheveux ≥ 11.38 et Couleur cheveux est châtain alors classe -1
 - si Longueur cheveux ≥ 11.38 et Couleur cheveux est noir et Groupe 1 alors classe -1

Approche LORE : bilan

- Approche efficace et basée sur l'utilisation d'arbres de décision
- Construction de contre-factuels
 - approche plus générale qui peut s'appliquer dans un cadre non agnostique...
 - ... si le modèle f est un arbre ou une base de règles
- Remarques
 - génération de la base d'exemples
 - hyper-paramètres
 - recherche des règles contre-factuelles (tout ou rien pour le test)
- D'autres pistes avec des surrogates
 - TREPAN, ANCHOR, FoITREE,...

Apprentissage inductif flou

- Théorie des sous-ensembles flous pour aider l'apprentissage
 - prise en compte de données numériques, imprécises ou floues
 - mise en œuvre d'un raisonnement flou
- La base d'apprentissage contient des exemples décrits par des données numériques ou des valeurs floues
- Discretisation / fuzzification des données
 - une valeur floue généralise un ensemble de valeurs précises
 - robustesse

Apprentissage inductif flou : pouvoir de généralisation élevé.

- Une valeur floue est déjà une généralisation

Plan du cours

Approches par modèle de substitution

- Apprentissage de règles floues
 - systèmes d'inférence floue
 - modèles de Mamdani
 - modèles de Takagi & Sugeno
 - approches neuro-floues
 - arbres de décision flous

Quelques approches floues (1)

- Fuzzy cluster analysis
 - prise en compte d'appartenance graduelle
 - les frontières entre les clusters ne peuvent pas être définies précisément
- Apprentissage de règles floues
 - classification ou régression
 - combinaison avec d'autres approches
 - p.ex. optimisation de fonctions d'appartenance par algorithmes génétiques
 - approches neuro-floues
 - représentation d'un système flou comme un réseau de neurones
 - techniques classiques d'optimisation
 - combine les avantages du flou (interprétabilité) et des réseaux de neurones (flexibilité)

Comment rendre flou ?

- Données floues
 - construction de fonctions d'appartenance
 - les données sont floues
 - connaissances subjectives : sensation, sentiment, opinion, perception
 - connaissances vagues : capteurs imprécis, bruit
- Frontières de décision floues
 - les classes ou groupes ne sont pas connus avec précision
 - mal définies ou mal identifiées
- Mesures floues : paramètres de l'algorithme
 - similarité entre individus
 - mesures d'information floues

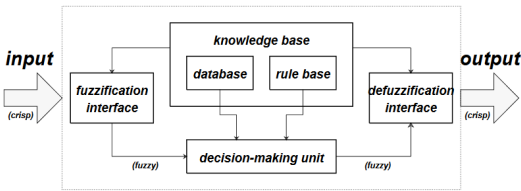
Apprentissage de règles floues

- Étant donné une base de règles floues,
 - apprendre les caractéristiques de chaque règle : optimiser des fonctions d'appartenance
- Étant donné une base d'apprentissage
 - apprendre une base de règles floues : trouver des relations entre les exemples de la base
 - exemple : construction d'une base de règles par arbres de décision flous

Quelques approches floues (2)

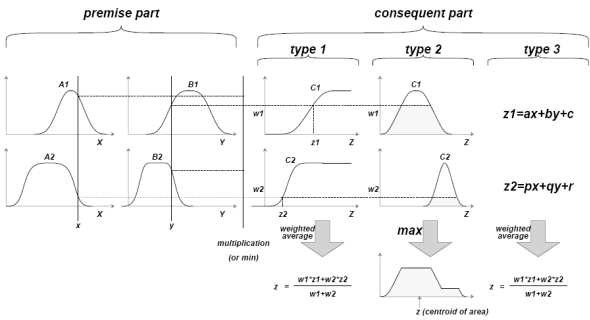
- Apprentissage d'arbres de décision flous
 - exemple type de fuzzification d'un algorithme
 - paramètres : interface avec les représentations floues / vagues
- Règles d'association floues
 - notion floue de support / confiance
 - mesures de qualité
- Représentations floues pour l'apprentissage à partir de cas
 - importance de la similarité
 - agrégation (opérateurs OWA ou approches de fusion)
- Réseaux possibilistes
 - lien possibilistes (≠ probabilistes) entre les informations

Systèmes d'inférence floue (SIF) à base de règles



- Règles : IF <premise> THEN <conséquent>
- Selon le type du conséquent :
 - modèle de Mamdani
 - modèle de Takagi-Sugeno

SIF : différents modèles



[Jang 94]

- ▶ types 1 & 2 : modèle de Mamdani
- ▶ type 3 : modèle de Takagi-Sugeno

Modèle de Mamdani

Règles de type Mamdani

R_1 : IF V_1 is $A_{1,1}$ and ... and V_k is $A_{1,k}$ THEN W is B_1
...
 R_p : IF V_1 is $A_{p,1}$ and ... and V_k is $A_{p,k}$ THEN W is B_p

- ▶ Variables linguistiques :
 $(V_1, X_1, T_1), \dots, (V_k, X_k, T_k), (U, Y, T_Y)$
- ▶ Observation de données précises : $(x_1, \dots, x_k) \in X_1 \times \dots \times X_k$
- ▶ Conclusion floue :
 - agrégation des conséquents des règles par un "or"
- ▶ Sortie précise si nécessaire :
 - $y \in Y$ obtenue par défuzzification :
centre de gravité, moyenne du noyau, milieu, etc.

Etape 1 : application d'une règle

- ▶ On considère
 - une règle $R = \text{Si } V_1 \text{ est } A_1 \text{ et ... et } V_n \text{ est } A_n, \text{ alors } W \text{ est } B$
 - l'entrée $x_0 = (x_1, \dots, x_n) \in X_1 \times \dots \times X_n$
- Calculer la compatibilité avec R
degré d'appartenance de x_0 au sous-ensemble flou prémisses de la règle (défini sur le produit cartésien)
$$a_R = f_A(x_0) = \min(f_{A_1}(x_1), \dots, f_{A_n}(x_n))$$
- Calculer le résultat de la règle R
avec la (fausse) implication floue de Mamdani
$$f_{resR} : Y \rightarrow [0, 1]$$
$$y \mapsto \min(a_R, f_B(y))$$

Principe théorique

(Zadeh, 73)

- ▶ Principe : modélisation sous la forme de règles floues
Si θ vaut environ 0 et si $\hat{\theta}$ vaut environ zéro alors F doit être approximativement zéro
 - mesure de $\theta, \hat{\theta}, \dots$: observations précises
 - modus ponens généralisé (MPG) pour en déduire la commande
- ▶ Avantages
 - connaissance mathématique du système non nécessaire
 - simulation de l'expert qui contrôle le système
 - flexibilité et adaptabilité de la commande

Traitement

- ▶ On considère
 - une règle $R = \text{Si } V_1 \text{ est } A_1 \text{ et ... et } V_n \text{ est } A_n, \text{ alors } W \text{ est } B$
 - l'entrée $x_0 = (x_1, \dots, x_n) \in X_1 \times \dots \times X_n$
- ▶ Etape 1 : application de la règle
 - utilisation des résultats sur le MPG
 - avec une observation précise
 - avec l'"implication" de Mamdani : $f_R(x, y) = \min(x, y)$

$$f_{res}(y) = \sup_{x \in X} \top(f_{A'}(x), \min(f_A(x), f_B(y)))$$
$$= \min(f_A(x_0), f_B(y))$$

$$A = A_1 \times \dots \times A_n \text{ défini sur } X = X_1 \times \dots \times X_n$$

Traitement

- ▶ On considère
 - un ensemble de règles
 $R_1 = \text{Si } V_1 \text{ est } A_1 \text{ et ... et } V_n \text{ est } A_n, \text{ alors } W \text{ est } B$
 $R_2 = \text{Si } V_1 \text{ est } A'_1 \text{ et ... et } V_n \text{ est } A'_n, \text{ alors } W \text{ est } B'$
 $R_3 = \text{Si } V_1 \text{ est } A''_1 \text{ et ... et } V_n \text{ est } A''_n, \text{ alors } W \text{ est } B''$
 - l'entrée $x_0 = (x_1, \dots, x_n) \in X_1 \times \dots \times X_n$
- ▶ Etape 2 : combinaison des résultats de toutes les règles
 - \Rightarrow sous-ensemble flou décrivant la commande à appliquer

Etape 2 : combinaison des règles

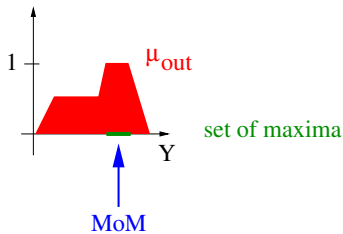
- Calcul de la compatibilité avec R
$$a_R = f_A(x_0)$$
$$= \min(f_{A_1}(x_1), \dots, f_{A_n}(x_n))$$
- Calcul du résultat de la règle R
$$f_{resR} : Y \rightarrow [0, 1]$$
$$y \mapsto \min(a_R, f_B(y))$$
- **Combinaison des sef**
obtenus avec chacune des règles,
par l'opérateur max
$$f_{res} : Y \rightarrow [0, 1]$$
$$y \mapsto \max_R f_{resR}(y)$$

Traitement

- On considère
 - un ensemble de règles
 $R_1 = \text{Si } V_1 \text{ est } A_1 \text{ et } \dots \text{ et } V_n \text{ est } A_n, \text{ alors } W \text{ est } B$
 $R_2 = \text{Si } V_1 \text{ est } A'_1 \text{ et } \dots \text{ et } V_n \text{ est } A'_n, \text{ alors } W \text{ est } B'$
 $R_3 = \text{Si } V_1 \text{ est } A''_1 \text{ et } \dots \text{ et } V_n \text{ est } A''_n, \text{ alors } W \text{ est } B''$
 - l'entrée $x_0 = (x_1, \dots, x_n) \in X_1 \times \dots \times X_n$
- Etape 2 : combinaison des résultats de toutes les règles
 - \Rightarrow sous-ensemble flou décrivant la commande à appliquer
- Etape 3 : **défuzzification**
 - \Rightarrow on en déduit une valeur précise

Etape 3 : moyenne des maxima

- Définition : **moyenne des y qui maximisent f_{res}**

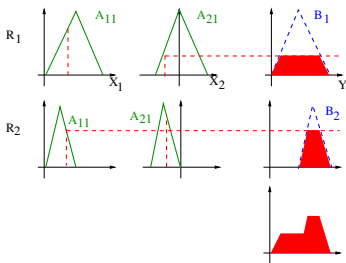


- Difficultés
 - suppose que l'on ait un intervalle
 - peut conduire à un contrôleur discontinu : le résultat dépend seulement de la règle qui a le plus haut degré de compatibilité

Etapes 1 et 2 : exemple

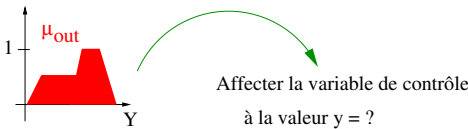
R : si V_1 est A_1 et V_2 est A_2 , alors W est B

- Compatibilité avec la règle R :
$$a_R = \min(f_{A_1}(x_1), \dots, f_{A_n}(x_n))$$
- Résultat de la règle R
$$f_{resR} : Y \rightarrow [0, 1]$$
$$y \mapsto \min(a_R, f_B(y))$$
- Combinaison des règles
$$f_{res} : Y \rightarrow [0, 1]$$
$$y \mapsto \max_R f_{resR}(y)$$



Etape 3 : défuzzification

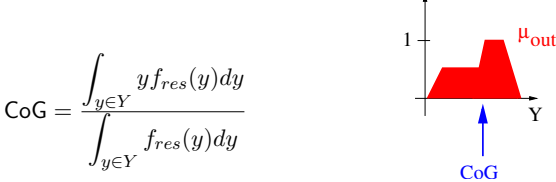
- Transformer le sous-ensemble flou obtenu en une valeur précise



- Méthodes principales
 - maximum : choisir y qui maximise f_{res}
 - moyenne des maxima
 - centre de gravité

Etape 3 : centre de gravité

- Définition : **centre de gravité de l'aire sous la courbe**



- Caractéristiques
 - conduit le plus souvent à un comportement continu
 - dépend de plusieurs règles, pondérées selon leurs degrés de compatibilité
 - coût de calcul élevé
 - difficile à justifier sémantiquement

Contrôleur de Takagi-Sugeno

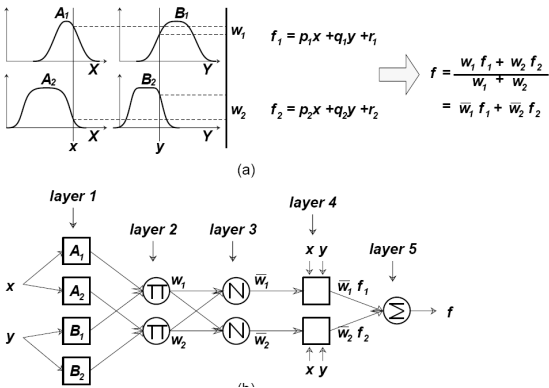


- Base de règles de **forme différente** : **conclusion précise**
 $R : \text{si } V_1 \text{ est } A_1 \text{ et ... et } V_n \text{ est } A_n, \text{ alors } y = g(x_1, ..., x_n)$
 - $g : X_1 \times ... \times X_n \rightarrow Y$
 - souvent g est linéaire : $g(x_1, ..., x_n) = a_0 + a_1x_1 + ...a_nx_n$

Comparaison

- | | |
|---|--|
| <p>Contrôleur de Mamdani</p> <ul style="list-style-type: none">► Prémisses et conclusions linguistiques► Avantages<ul style="list-style-type: none">• quand le modèle est imprécis, voire inconnu• facile à appréhender par un non-spécialiste► Inconvénients<ul style="list-style-type: none">• manque de preuves formelles (stabilité, optimalité) | <p>Contrôleur de Takagi-Sugeno</p> <ul style="list-style-type: none">► Conclusion arithmétique► Avantages<ul style="list-style-type: none">• preuve formelle de stabilité• extension des concepts de l'automatique linéaire au cas non-linéaire► Inconvénients<ul style="list-style-type: none">• nécessite la connaissance d'un bon modèle |
|---|--|

Exemple de système neuro-flou hybride : ANFIS



[Jang 94]

Adaptive Network based Fuzzy Inference System (pour TS)

Modèle de Takagi-Sugeno

Règles de type Takagi-Sugeno [Takagi & Sugeno 85]

$$R_1 : \text{IF } V_1 \text{ is } A_{1,1} \text{ and ... and } V_k \text{ is } A_{1,k} \text{ THEN } y_1 = g_1(x_1, \dots, x_k)$$
$$\dots$$
$$R_p : \text{IF } V_1 \text{ is } A_{p,1} \text{ and ... and } V_k \text{ is } A_{p,k} \text{ THEN } y_p = g_p(x_1, \dots, x_k)$$

avec $g_i : X_1 \times \dots \times X_k \rightarrow Y$ pour tout $i = 1, \dots, p$
en général : $g_i(x_1, \dots, x_k) = p_{i,0} + p_{i,1}x_1 + \dots + p_{i,k}x_k$ avec $p_{i,j} \in \mathbb{R}$

- Variables linguistiques : $(V_1, X_1, T_1), \dots, (V_k, X_k, T_k), (U, Y, T_Y)$
- Observation de données précises : $(x_1, \dots, x_k) \in X_1 \times \dots \times X_k$
- Conclusion précise : $y \in Y$
 - moyenne pondérée des y_i

Approches neuro-floues

Idee des approches neuro-floues

Combiner une représentation à base de règles (SIF) avec un réseau de neurones

- Système neuro-flou coopératif :
 - réseau de neurones pour apprendre des sous-parties du SIF
 - une fois fait, on ne garde que le système flou
- Système neuro-flou combiné
 - les 2 approches se combinent
 - un réseau détermine les entrées ou les sorties d'un système flou
- Système neuro-flou hybride
 - pour déterminer les paramètres du système flou
 - réseau + approches d'optimisation (descente de gradient,...)

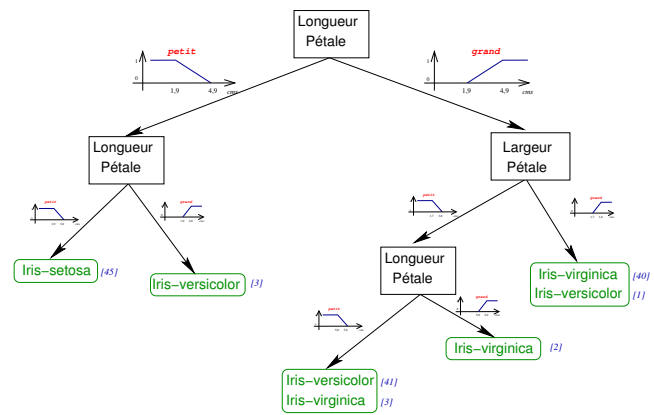
Construction / mise au point de SIF

- Selon le type, spécificités des approches :
 - modèle de Mamdani : arbres de décision flous, construction de règles, ...
 - modèle de Takagi-Sugeno : approches neuro-floues

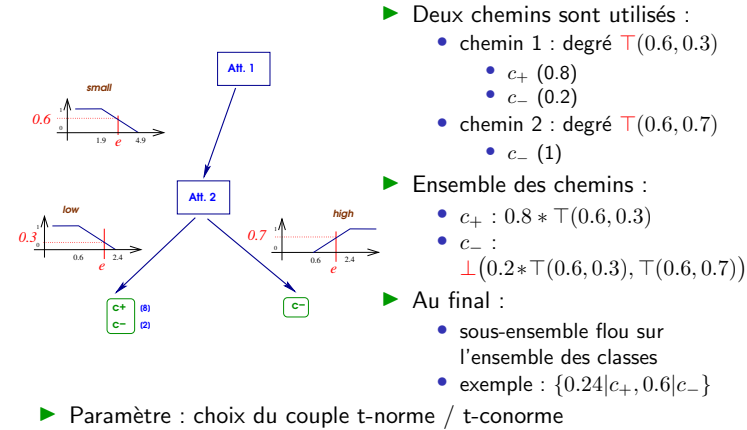
Conclusions sur les SIF et les approches neuro-floues

- Un système d'inférence flou est un approximateur universel
 - on peut trouver un SIF pour représenter toute fonction
 - modèle interprétable
- Système neuro-flou :
 - combiner un SIF et une représentation par réseau de neurones
 - plusieurs approches existent : ANFIS, NEFCON, NEFCLASS,...
 - utilisé en contrôle flou (cf. cours à venir)
 - conception pour hardware

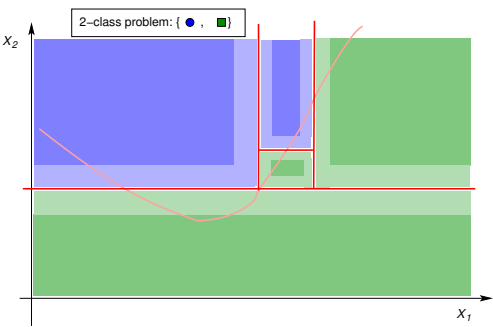
Exemple d'arbre de décision flou



Classification d'un exemple e

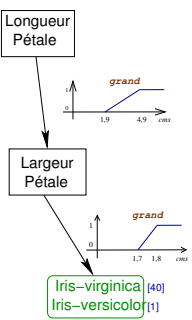


Arbres de décision flous : frontières floues



- Amélioration : "fuzzification" des frontières
 - arbres de décision flous

Classification avec un arbre de décision flou



SI
la longueur du pétale est grande
ET
la largeur du pétale est grande
ALORS
0.98 | Iris-Virginica
+ 0.02 | Iris-Versicolor

Extension d'une approche d'apprentissage

- Approche classique
 - validation formelle : théorie de l'information
- Caractérisation des données à prendre en compte
 - hypothèses ?
- Comment réaliser une extension ?
 - qu'est-ce qui doit être étendu ?
 - comment est-ce que cela doit être étendu ?
 - quelles propriétés peuvent être conservées ?
- Quelles spécificités de l'algorithme ?
 - comment évaluer l'impact de l'extension ? la valider ?
 - quels apports ?
- Etude de cas : les arbres de décision

Entropie d'événements flous

► Extension de l'**entropie de Shannon** aux événements flous

- v_1, \dots, v_m : m valeurs de l'attribut A
- c_1, \dots, c_K : K valeurs pour la classe C

$$H_E(C|A) = - \sum_{j=1}^m p^*(v_j) \sum_{k=1}^K p^*(c_k|v_j) \log(p^*(c_k|v_j))$$

- avec la **probabilité d'un événement flou** e d'un ensemble dénombrable X muni d'une probabilité p , définie par (Zadeh, 65) :

$$p^*(e) = \sum_{x \in X} \mu_e(x) p(x)$$

► Mesure du lien entre A et la classe C