Chapitre 2 La Sélection de variables

Le domaine de la fouille de données est actuellement caractérisé par la présence de bases de données de taille relativement importante. En effet, la collecte d'information devient de plus en plus facile et donc, rapide. Cependant, la totalité de l'information collectée n'est pas forcément pertinente au vue du problème considéré. Aussi, appliquer, sur l'ensemble des données récupérées, les techniques de fouille de données est bien trop coûteux. Il paraît, ainsi, nécessaire de distinguer, avant tout apprentissage, l'information pertinente de l'information inutile et/ou redondante. Cette distinction peut s'effectuer à l'aide du processus de sélection de variables lors de la phase de prétraitement des données.

La sélection de variables est un processus qui permet de « sélectionner » un sous-ensemble de variables considérées par le processus comme pertinentes. Les données d'entrée du processus sont constituées par l'ensemble initial de variables qui forment l'espace de représentation et l'ensemble des données d'apprentissage du problème étudié. Le processus de sélection de variables se décompose de la manière suivante, figure 2 :

- A partir de l'ensemble initial des variables, le processus de sélection détermine un sous-ensemble de variables qu'il considère comme les plus pertinentes ;
- Le sous-ensemble est ensuite soumis à une procédure d'évaluation. Cette dernière permet d'évaluer les performances et la pertinence du sous-ensemble ;
- En fonction du résultat de la procédure d'évaluation, un critère d'arrêt du processus détermine si le sous-ensemble de variables peut être soumis à la phase d'apprentissage. Si tel est le cas, le processus de sélection s'arrête, sinon, un autre sous-ensemble de variables est généré.

Les principaux enjeux et conséquences de la sélection de variables sont divers :

- La sélection de variables va dans un premier temps nous permettre de déterminer les variables considérées comme pertinentes ;
- La sélection de variables nous permet de supprimer le bruit généré par certaines variables ;
- Les variables redondantes sont également supprimées ;

- La taille de l'espace de représentation est ainsi réduit. Le coût de calcul de la phase d'apprentissage est également réduit.

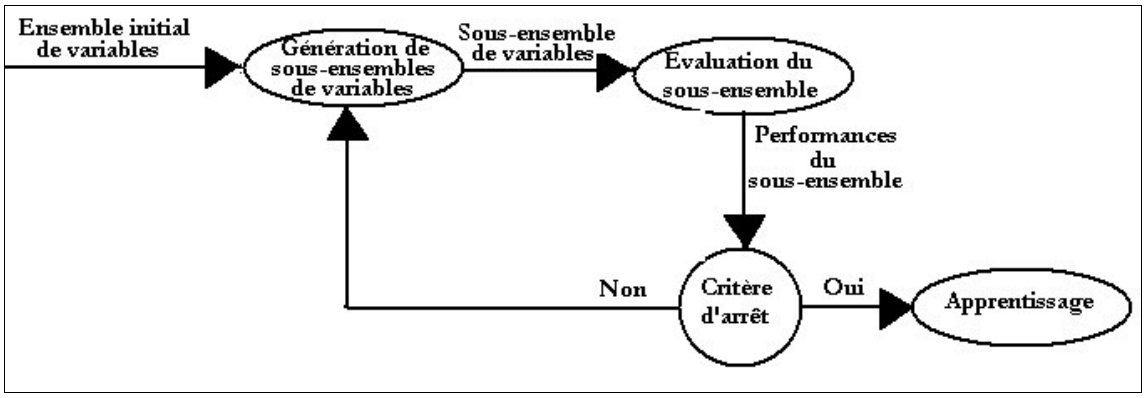


Figure 2 Processus de sélection de variables.

La première section est consacré à décrire l'ensemble des méthodes de sélection existantes. Ensuite, nous avons expérimenté un certain nombre de ces méthodes. Enfin, nous présentons et expérimentons la méthode de sélection que nous proposons.

1 Les algorithmes de sélection de variables

Les algorithmes de sélection de variables sont caractérisés par quatre éléments :

- Le type d'approche ;
- La direction de recherche ;
- La fonction d'évaluation ;
- Le critère d'arrêt.

1.1 Le type d'approche

Il existe deux types d'algorithmes visant à sélectionner un sous-ensemble de variables : les méthodes dîtes enveloppe, [6], et les méthodes filtre, [9]. La différence fondamentale entre ces deux types de méthodes réside dans le fait que la première est liée à l'algorithme d'apprentissage utilisé alors que la seconde en est totalement indépendante.

1.1.1 Les méthodes enveloppe

Ces méthodes utilisent l'algorithme d'apprentissage comme fonction d'évaluation, figure 3. Elles permettent la génération itératives de sous-ensembles de variables. L'algorithme d'apprentissage va permettre de tester les différents sous-ensembles de variables générés. Il intervient donc au sein même du processus de sélection de variables.

A partir de l'ensemble initial des variables, les méthodes enveloppe génèrent un sous-ensemble de variables qui est fourni à l'algorithme d'apprentissage. En fonction du résultat de l'algorithme d'apprentissage, le sous-ensemble est considéré ou non comme le sous-ensemble de variables optimal. Si tel est le cas, il est possible de poursuivre le processus de fouille de données et de passer à l'étape d'apprentissage.

Pour ce type de méthode, l'algorithme d'apprentissage fait partie intégrante du processus de sélection et sert de fonction d'évaluation.

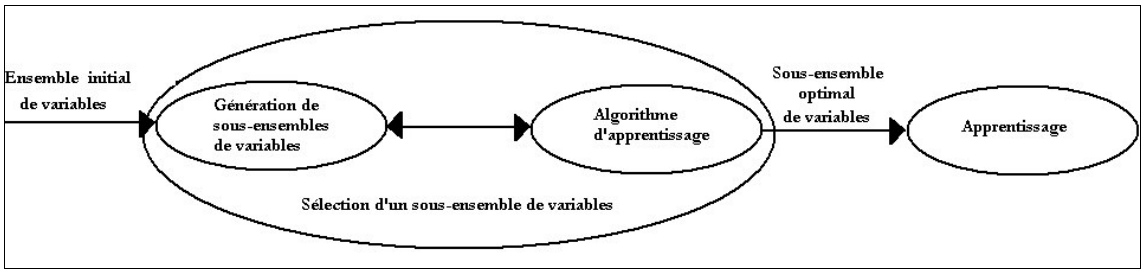


Figure 3 Approche enveloppe.

1.1.2 Les méthodes filtre

Les méthodes filtre, quant à elles, n'utilisent absolument pas l'algorithme d'apprentissage dans leur processus, figure 4. Ces méthodes permettent la génération de sous-ensembles de variables qui sont testés à l'aide d'une fonction d'évaluation.

A partir de l'ensemble initial de variables, les méthodes filtre génèrent un sous-ensemble de variables. Ce sous-ensemble de variables est soumis à une fonction d'évaluation propre à chaque méthode filtre. En fonction du résultat de la fonction d'évaluation, le sous-ensemble est considéré ou non comme optimal. Si tel est le cas, le sous-ensemble de variables peut alors être soumis à la phase d'apprentissage.

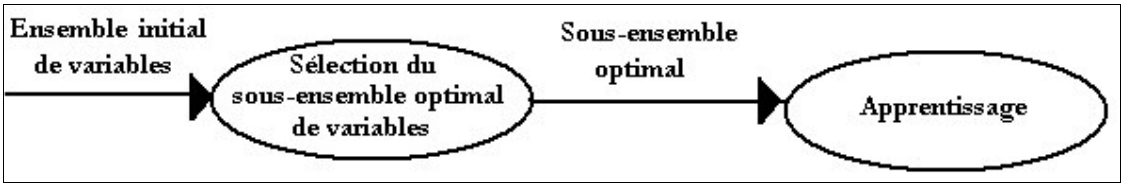


Figure 4 Approche filtre

1.2 La direction de recherche

La sélection de variables peut être vue comme un problème de recherche où chaque état de l'espace de recherche spécifie un sous-ensemble possible de variables. Le passage de l'état initial à l'état final peut être schématisé par un graphe partiellement ordonné où chaque état enfant possède un attribut de plus que ses parents. Si l'on prend un exemple avec un ensemble initial composé de quatre variables alors la figure 5 nous montre l'espace de recherche. La bille noire représente la présence de la variable et la bille blanche son absence.

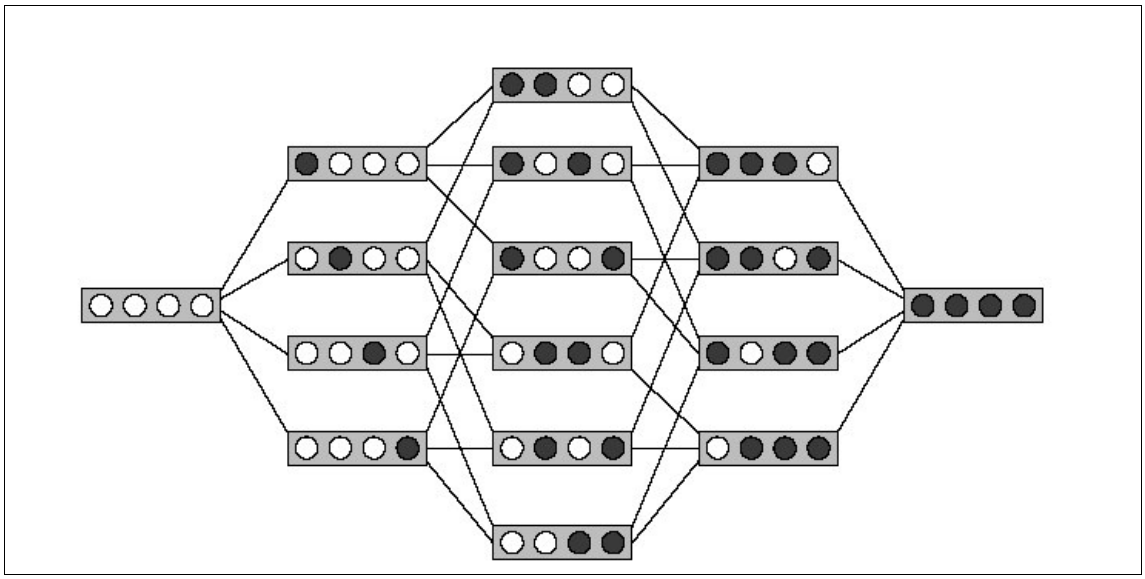


Figure 5 Espace de recherche [5]

Si p représente le nombre de variables initiales, il y a $O\left(2^p\right)$ états possibles. Donc, les méthodes de sélection de variables utilisent l'ordre partiel des variables pour organiser la recherche du sous-ensemble optimal de variables. Cet ordre partiel correspond à l'agencement des variables dans le temps, c'est à dire à leur utilisation lors du processus de sélection.

Les directions de recherche peuvent être de trois types : la forward selection, la backward elimination et les méthodes bidirectionnelles.

1.2.1 La forward selection (FS)

Cette stratégie part d'un ensemble vide. Les variables sont ajoutées une à une. A chaque itération, la variable optimale suivant un certain critère est ajoutée. Le processus s'arrête soit quand il n'y a plus de variable à ajouter, soit quand un certain critère est satisfait. Une fois qu'une variable a été ajoutée, la FS ne peut la retirer.

Entrée X L'ensemble initial des variables
φ La mesure de qualité des variables
ε Le seuil de qualité

Sortie X^* L'ensemble de variables sélectionnées

Début Forward Selection
$X^* = \{\ \}$
Répéter
Chercher la meilleure variable X_i^* de X
$X^* = X^* \cup \{X_i^*\}$
$X = X - \{X_i^*\}$
Jusqu'à ce que $\varphi(X^*) > \varepsilon$ ou $X = \{\ \}$
Fin Forward selection

Algorithme 1 Forward selection.

1.2.2 La backward elimination (BE)

Cette stratégie part de l'ensemble initial de variables. A chaque itération, une variable est enlevée de l'ensemble. Cette variable est telle que sa suppression donne le meilleur sous-ensemble selon un critère particulier. Une fois la variable supprimée, il est impossible de la réintégrer.

Entrée	X	L'ensemble total des variables initiales
	φ	La mesure de qualité des variables
	ε	Le seuil de qualité
Sortie	X'	L'ensemble de variables sélectionnées

Début Backward Elimination ¨

$X' = X$

Répéter

Chercher la variable la moins pertinente X_i' de X

$X' = X' - \{X_i'\}$

Jusqu'à ce que $\varphi(X') > \varepsilon$ ou $X' = \{\ \}$

Fin Backward Elimination

Algorithme 2 Backward elimination.

1.2.3 Les méthodes bidirectionnelles

Il est également possible d'utiliser une variation de l'ordre partiel des variables : Devijver et Kittler [10] définissent un opérateur qui ajoute k $(k < p)$ variables et en enlève une. La première décision à prendre est donc le point de départ de la recherche, il peut être de trois sortes :

- Un ensemble vide : il s'agit de la Forward Stepwise Selection ;
- Un ensemble complet : Backward Stepwise Elimination ;
- Un ensemble d'attributs choisis aléatoirement.

Ces méthodes permettent de pallier le problème de l'irrévocabilité de la suppression ou de l'ajout d'une variable, problème présent dans les deux autres directions de recherche. En effet, l'importance d'une variable peut se modifier ultérieurement. Ces méthodes autorisent l'ajout et la suppression d'une variable de l'ensemble des variables à n'importe quelle étape de la recherche autre que la première ou la dernière.

1.3 La fonction d'évaluation

Le but d'une fonction d'évaluation est de mesurer la capacité d'une variable ou d'un ensemble de variables exogènes à distinguer les classes de la variable endogène. L'optimalité d'un sous-ensemble est relative à la fonction d'évaluation utilisée.

Dash et Liu [9] considèrent que ces fonctions ou critères peuvent être regroupées en cinq catégories qui sont les suivantes :

- **Critères d'information** : C'est la quantité d'information apportée par une variable sur la variable endogène. La variable, ayant le gain d'information le plus élevé, sera préférée aux autres variables. Le gain d'information est la différence entre l'incertitude a priori et l'incertitude a posteriori.
- **Critères de distance** : Ces mesures s'intéressent au pouvoir discriminant d'une variable.
- **Critères d'indépendance** : Ils regroupent toutes les mesures de corrélation ou d'association. Ils permettent de calculer le degré avec lequel une variable exogène est associée à une variable endogène.
- **Critères de consistance** : Ils sont liés au biais des variables minimum (min-features bias). Ces mesures recherchent l'ensemble de variables le plus petit qui satisfait un pourcentage d'inconsistance minimum défini par l'utilisateur. Deux objets sont inconsistants si leurs modalités sont identiques et s'ils appartiennent à deux classes différentes. Ces mesures peuvent permettre de détecter les variables redondantes.
- **Critères de précision** : Ils utilisent le classifieur comme fonction d'évaluation. Le classifieur choisit, parmi tous les sous-ensembles de variables, celui qui est à l'origine de la meilleure précision prédictive. Cette catégorie de critères est celle utilisée par toutes les méthodes enveloppe.

1.4 Le critère d'arrêt

Le type de méthodes et la fonction d'évaluation peuvent influencer le choix du critère d'arrêt. Selon si l'on se base sur l'une ou l'autre de ces caractéristiques, le critère d'arrêt varie.

Un critère d'arrêt basé sur un type de méthode pourra être de deux sortes :

- Soit un nombre prédéfini de variables sélectionnées. Ce type de critère d'arrêt nous parait difficile à utiliser. En effet, il est rare que le nombre optimal de variables soit connu à l'avance. Cependant, certaines contraintes techniques ou calculatoires peuvent induire un nombre fixe de variables à sélectionner.
- Soit un nombre d'itérations préfixé. Ce genre de critère permet de limiter le temps de calcul. Bien sûr, le résultat obtenu ne sera pas forcément optimal.

Un critère d'arrêt basé sur une fonction d'évaluation peut être liée à deux faits :

- L'ajout ou la suppression d'une variable ne produit aucun sous-ensemble plus performant
- Le sous-ensemble obtenu est, d'après la fonction d'évaluation, le sous-ensemble optimal.

L'itération continue jusqu'à ce que le critère d'arrêt soit satisfait. Le processus de sélection de variables s'arrête en fournissant le sous-ensemble obtenu à la procédure de validation.

1.5 Les méthodes filtre

Le filtrage est un processus de prétraitement des données qui filtre les variables non pertinentes avant que n'intervienne la phase d'apprentissage. Il utilise les caractéristiques générales de l'ensemble d'apprentissage pour sélectionner certaines variables et en exclure d'autres. Le schéma le plus simple est d'évaluer individuellement chaque variable en considérant sa fonction d'évaluation et de sélectionner les variables possédant les plus grandes valeurs. La fonction d'évaluation est la plupart du temps sous la forme d'un critère nommé critère de sélection. Pour cette raison nous allons, dans un premier temps, présenter ces différents critères.

1.5.1 Les critères de sélection

Il convient de distinguer les critères issus de l'approche statistique de ceux basés sur la comparaison par paire d'objets. La première approche consiste à travailler sur des tableaux de contingence, tandis que la seconde consiste à comparer les partitions induites par deux variables, paires d'objets à paires d'objets.

1.5.1.1 Approche statistique

Soit T_k le tableau de contingence des variables X_k et Y . n_{ij} représente le nombre d'objets possédant la $j^{ème}$ modalité de la variable X et la $i^{ème}$ modalité pour la variable Y , tableau 1.

Le nombre d'objets appartenant à la classe i est $n_{i.} = \sum_{j=1}^{q^k} n_{ij}$, le nombre d'objets ayant la modalité m_j^k est $n_{.j} = \sum_{i=1}^{m} n_{ij}$ et le nombre total d'objets est $n = \sum_{i=1}^{m}\sum_{j=1}^{q^k} n_{ij}$.

$T_k =$ $Y \times X_k$	m_1^k	…	m_j^k	…	$m_{q^k}^k$
y_1	n_{11}	…	n_{1j}	…	n_{1q^k}
⋮	⋮	⋱	⋮		⋮
y_i	n_{i1}	…	n_{ij}	…	n_{iq^k}
⋮	⋮	…	⋮	⋱	⋮
y_m	n_{m1}	…	n_{mj}	…	n_{mq^k}

Tableau 1 Tableau de contingence.

Il existe deux types de critères statistiques : les critères myopes et les critères contextuels.

Les critères myopes : Ce sont des estimateurs de la qualité d'une variable hors du contexte des autres variables explicatives. Ils sont inadaptés pour les algorithmes traitant des données contenant des variables corrélées. Ils peuvent appartenir à 3 catégories :

- **Critères d'information :**
  - L'entropie de Shannon [11] de l'ensemble des classes se référant aux modalités de la variable Y est définie comme suit : $H_Y = -\sum_{i=1}^{m} \frac{n_{i.}}{n} \log_2 \frac{n_{i.}}{n}$, H_Y mesure l'indétermination de l'affirmation qu'un objet pris aléatoirement parmi l'ensemble Ω d'objets observés est classé dans l'une des m classes.

Si les objets de Ω sont caractérisés par un attribut $X_k \in \mathrm{X}$, l'information associée à X_k est : $H_{X_k} = -\sum_{j=1}^{q^k} \frac{n_{.j}}{n} \log_2 \frac{n_{.j}}{n}$.

L'information de couple apportée par les variables X_k et Y sur l'ensemble Ω est : $H_{X_k Y} = -\sum_{j=1}^{q^k} \sum_{i=1}^{m} \frac{n_{ij}}{n} \log_2 \frac{n_{ij}}{n}$.

Il est alors possible de définir l'entropie de la variable Y par rapport à la variable X_k : $H_{Y/X_k} = H_{X_k,Y} - H_{X_k}$. Cette valeur quantifie l'indétermination existante quand on classe un objet, sachant que l'on connaît la répartition des valeurs de la variable X_k par rapport aux valeurs de la variable Y.

- Le gain d'information mesure l'interaction des variables X_k et Y :

  $\mathrm{Gain}(X_k) = H_Y - H_{Y/X_k}$.

- Le ratio du gain, [12], est calculé pour les variables dont le gain est supérieur à la moyenne des gains. La variable maximisant le rapport sera sélectionnée : $\mathrm{Gain}_{rt}(X_k) = \frac{\mathrm{Gain}(X_k)}{H_{X_k}}$.

- Le gain normalisé [13] : $\mathrm{Gain}_n(X_k) = \frac{\mathrm{Gain}(X_k)}{\log_2 q^k}, \quad q^k \geq 2$.

- La distance de Mantaras [14] a été créée pour résoudre le problème de surestimation des variables multivaluées : $D(X_k) = 1 - \frac{G\,\mathrm{ain}(X_k)}{H_{X_k Y}}$. Cette mesure peut également être considérée comme une mesure de distance.

- **Critères de distance :**

  - Le critère de Gini [15] : Breiman utilise cette mesure pour l'induction d'arbres de décision dans le système CART : $\mathrm{Gini}(X_k) = \frac{1}{n}\left(\sum_{j=1}^{q^k} \sum_{i=1}^{m} n_{ij} - \frac{n_{ij}^{\,2}}{n_{.j}} \right)$.

- Le critère ORT [16] mesure la séparation des classes et repose sur l'évaluation de l'angle θ formé par les vecteurs V_1 et V_2 associés à un nœud d'une bi-partition. La variable maximisant ce critère sera sélectionnée : $\mathrm{ORT}(X_k) = 1 - \cos\theta(V_1, V_2)$, où $\cos\theta(V_1, V_2) = \frac{V_1 \circ V_2}{\|V_1\| * \|V_2\|}$, avec $V_1 \circ V_2 = \sum_{i=1}^{m} \frac{n_{i1}}{n_{.1}} \frac{n_{i2}}{n_{.2}}$ et $V_j = \left(\frac{n_{1j}}{n}, \frac{n_{2j}}{n}, ..., \frac{n_{pj}}{n} \right)$.

- **Critères d'indépendance :**

  - Le Khi2 [17] est une mesure statistique traditionnelle :

    $$\chi^2_{(Y,X_k)} = n \sum_{j=1}^{q^k} \sum_{i=1}^{m} \left(n_{ij} - \frac{n_{i.} n_{.j}}{n} \right)^2 \frac{1}{n_{i.} n_{.j}}.$$

  - Le critère de Tschuprow [18] et [19] est une normalisation du Khi2 dans l'intervalle [0,1].

    $$T_{(Y,X_k)} = \frac{\frac{\chi^2}{n}}{\sqrt{(q^k - 1)(m-1)}}$$ q^k est le nombre de modalités de la variable X_k.

    La ressemblance entre X_k et Y est d'autant plus grande que $T_{(Y,X_k)}$ est grand. Ce critère a le défaut d'être en faveur des cas où les nombres de modalités descriptives et de classes cibles sont égaux. En effet, le T de Tshuprow ne peut prendre sa valeur maximale que dans ce cas.

  - Le coefficient de Cramer est une mesure de dépendance et de similarité basée sur un tableau de contingence : $C_{(Y,X_k)} = \sqrt{\frac{\frac{\chi^2}{n}}{\min[(q^k - 1), (m-1)]}}$.

    Dans la plupart des cas (q >= m), le dénominateur est constant, ce qui fait que le C de Cramer se comporte comme le Khi2 et peut avoir les mêmes défauts.

Les critères contextuels : Ce sont les critères de consistance. Ils estiment la qualité d'une variable exogène dans le contexte des autres variables exogènes. Ces mesures sont plus coûteuses mais permettent de découvrir des dépendances indécelables par les mesures myopes.

La mesure la plus connue est le critère heuristique-statistique de Zhou, [20].

$$\tau(X_k)=\frac{\sum_{j=1}^{q^k}\sum_{i=1}^{m}\left[\left(\frac{n_{ij}}{n}\right)^2\Big/\left(\frac{n_{i.}}{n}\right)\right]+\sum_{j=1}^{q^k}\sum_{i=1}^{m}\left[\left(\frac{n_{ij}}{n}\right)\Big/\left(\frac{n_{.j}}{n}\right)\right]-\sum_{j=1}^{q^k}\left(\frac{n_{.j}}{n}\right)^2-\sum_{i=1}^{m}\left(\frac{n_{i.}}{n}\right)^2}{2-\sum_{j=1}^{q^k}\left(\frac{n_{.j}}{n}\right)^2-\sum_{i=1}^{m}\left(\frac{n_{i.}}{n}\right)^2}$$

1.5.1.2 Comparaison par paires

L'idée fondamentale des comparaisons par paires est à attribuée à Condorcet dès 1785. Elle consiste en la comparaison des partitions induites par deux variables à modalités, paires d'objets à paires d'objets. Pour n individus, on doit donc considérer des tableaux de dimension d'ordre n^2. Le concept de comparaison par paires appliqué au tableau de contingence est apparu pour la première fois dans les travaux de Kendall [21]. Ce type de tableaux établi uniquement sur des comparaisons par paires permet de définir un certain nombre de mesures d'association entre X_k et Y, compatibles à la fois avec la statistique des tableaux de contingence et avec l'utilisation des comparaisons par paires.

Représentation par paires : Les données sont représentées sous forme relationnelle où une variable est représentée par un tableau. Nous disposons d'une variable X_k ayant q^k modalités c'est à dire une partition à q^k classes.

$(\Omega\times\Omega)_{X_k}$	ω_1	…	ω_j	…	ω_n	Total
ω_1	$\varphi^k_{11}=1$	…	φ^k_{1j}	…	φ^k_{1n}	$\varphi^k_{1.}$
⋮	⋮	⋱	⋮		⋮	⋮
ω_i	φ^k_{i1}	…	φ^k_{ij}	…	φ^k_{in}	$\varphi^k_{i.}$
⋮	⋮	…	⋮	⋱	⋮	⋮
ω_n	φ^k_{n1}	…	φ^k_{nj}	…	$\varphi^k_{nn}=1$	$\varphi^k_{n.}$
Total	$\varphi^k_{.1}$	…	$\varphi^k_{.j}$	…	$\varphi^k_{.n}$	$\varphi^k_{..}$

Tableau 2 Tableau des comparaisons par paires

Un tableau de comparaisons par paires (tableau 2) a pour terme général φ_{ij}^k est défini comme suit :

$$\varphi^k : \Omega \text{x} \Omega :\mapsto \{0,1\} : \begin{cases} \varphi_{ij}^k = 1 \text{ si } X_k(\omega_i) = X_k(\omega_j) \\ \varphi_{ij}^k = 0 \text{ sinon} \end{cases}.$$

$\varphi_{i.}^k = \sum_{j=1}^{n} \varphi_{ij}^k$ est le nombre de concordances avec ω_i et $\varphi_{.j}^k = \sum_{i=1}^{n} \varphi_{ij}^k$;est le nombre de concordances avec ω_j.

Critères basés sur les paires : Ces critères sont moins nombreux que les critères statistiques. Ils sont également de deux types : myopes et contextuels.

- **Mesures myopes** :

  - Critère de Condorcet [22] : $C(i,j) = \sum_{i=1}^{n} \sum_{\substack{j=1 \\ i \neq j}}^{n} \left(\varphi_{ij}^Y \varphi_{ij}^k + \overline{\varphi}_{ij}^Y \overline{\varphi}_{ij}^k \right)$. Plus forte est la valeur de $C(i,j)$, plus fort est le pouvoir discriminant.

  - Critère de Zahn [23] : $Z(i,j) = \sum_{i=1}^{n} \sum_{\substack{j=1 \\ i \neq j}}^{n} \left(\overline{\varphi}_{ij}^Y \varphi_{ij}^k + \varphi_{ij}^Y \overline{\varphi}_{ij}^k \right)$. Plus faible est la valeur de $Z(i,j)$, plus fort est le pouvoir discriminant.

  - Critère de l'écart à l'indépendance [24], [25] : $M(i,j) = \dfrac{\sum_{i=1}^{n} \sum_{\substack{j=1 \\ i \neq j}}^{n} \left(\varphi_{ij}^Y - \overline{\varphi}_{ij}^Y \right) \varphi_{ij}^k + \sum_{i=1}^{n} \sum_{\substack{j=1 \\ i \neq j}}^{n} \left(\overline{\varphi}_{ij}^Y \right)}{n^2}$.

    Plus forte est la valeur de $M(i,j)$, plus fort est le pouvoir discriminant.

On peut remarquer que les trois critères de Condorcet, Zahn et Marcotorchino sont très proche les uns des autres. En utilisant l'égalité $\left(\overline{\varphi}_{ij}^k + \varphi_{ij}^k \right)\left(\overline{\varphi}_{ij}^Y + \varphi_{ij}^Y \right) = \left(\varphi_{ij}^Y \varphi_{ij}^k + \overline{\varphi}_{ij}^Y \overline{\varphi}_{ij}^k \right) + \left(\overline{\varphi}_{ij}^Y \varphi_{ij}^k + \varphi_{ij}^Y \overline{\varphi}_{ij}^k \right)$, on déduit que $Z(i,j) = n(n-1) - C(i,j)$. Enfin, avec $\left(\varphi_{ij}^Y - \overline{\varphi}_{ij}^Y \right) \varphi_{ij}^k + \overline{\varphi}_{ij}^Y = \varphi_{ij}^Y \varphi_{ij}^k + \overline{\varphi}_{ij}^Y \left(1 - \varphi_{ij}^k \right) = \varphi_{ij}^Y \varphi_{ij}^k + \overline{\varphi}_{ij}^Y \overline{\varphi}_{ij}^k$, on déduit que $M(i,j) = C(i,j) / n^2$.

- **Mesures contextuelles :**

  - Relief [26, 27] : Le critère à lui seul n'est pas contextuel car une variable est estimée en dehors du contexte des autres variables. Mais, l'algorithme dans lequel il s'intègre le rend contextuel. Le critère Relief se restreint aux problèmes à deux classes et repose sur le principe suivant : la variable idéale doit être en mesure, par des instanciations différentes de sa valeur, de séparer des individus voisins appartenant à des classes différentes. Si des individus sont de même classe, ses valeurs doivent être identiques. Il est définie de la manière suivante : $\text{Relief}(X_k) = \sum_{i=1}^{n} \left(\overline{\varphi}_{i\notin}^{k} - \overline{\varphi}_{i\in}^{k} \right)$ avec $\omega_{\in}$ est l'objet le plus proche de ω_i appartenant à la même classe $(Y(\omega_i) = Y(\omega_{\in}))$ et $\omega_{\notin}$ est l'objet le plus proche de ω_i n'appartenant pas à la même classe $(Y(\omega_i) \neq Y(\omega_{\notin}))$. Plus forte est la valeur de $\text{Relief}(X_k)$, plus fort est le pouvoir discriminant.

  - Le mérite contextuel [28] : Lorsque deux individus ω_i et ω_j , de classes différentes, sont comparés, seules les variables descriptives qui ont des valeurs différentes peuvent aider à la discrimination des deux individus. Soit $R_{ij} = \sum_{k=1}^{p} \overline{\varphi}_{ij}^{k}$ le nombre de variables discriminantes du couple d'individus (ω_i, ω_j). La difficulté de séparer ω_i et ω_j peut être vue comme une fonction décroissante du nombre de variables les discriminants : $\frac{1}{(R_{ij})^2}$. Le mérite contextuel est alors définit de la manière suivante : $M^k = \sum_{i=1}^{n} \sum_{j=1}^{s} \frac{\overline{\varphi}_{ij}^{k}}{(R_{ij})^2}$; *avec* $Y(\omega_i) \neq Y(\omega_j)$ avec $\omega_j, j = 1,..,s$ représente un objet de l'ensemble des s plus proches voisins de ω_i tel que ω_i et ω_j appartiennent à des classes différentes.

On peut remarquer que l'inconvénient lié à l'approche par paires est son coût calculatoire. Pour cette raison, l'approche statistique, présentée précédemment, est souvent préférée. Cependant, l'approche par paires permet l'addition des tableaux par paires associés à autant de variables exogènes et l'obtention d'un tableau collectif ayant une signification très importante. En effet, celui-ci permet de ne pas se borner aux relations entre une variable exogène et une variable endogène.

1.5.2 La stratégie de recherche

Les méthodes filtre sont caractérisées une stratégie de recherche. Il en existe cinq types :

- **Les méthodes complètes** : Ces méthodes génèrent la totalité des sous-ensembles de variables. Cependant, ces méthodes ne sont pas forcément exhaustives ([29]), comme nous le verrons plus loin avec les méthodes comme Focus 2, [30] : les sous-ensembles générés ne sont pas forcément tous évalués. Différentes méthodes d'élagage peuvent être utilisées afin de réduire l'espace de recherche sans compromettre les chances de trouver le sous-ensemble optimal.
- **Les méthodes heuristiques** : On ajoute (ou l'on ôte) pas à pas des variables à l'ensemble des variables sélectionnées (ou restantes) jusqu'à ce que le sous-ensemble ne puisse plus être amélioré. A chaque itération, toutes les variables restantes jusque là peuvent être sélectionnées. Il y a plusieurs variantes de ce simple processus mais la génération des sous-ensembles est fondamentalement incrémentale (soit croissante soit décroissante). La taille de l'espace de recherche est de $O\left(2^p\right)$ ou moins, avec p le nombre de variables initiales ; il y a quelques exceptions comme Relief [26] et DTM [31]. Cependant, ces méthodes ne permettent pas d'obtenir un sous-ensemble optimal.
- **Les méthodes aléatoires** : Ces méthodes recherchent aléatoirement le sous-ensemble suivant. La recherche d'un bon sous-ensemble s'effectue sur l'espace réduit aux sous-ensembles possibles. La taille de cet espace (inférieure à $O\left(2^p\right)$) est définie en fixant le nombre d'itérations. L'optimalité de la solution dépend à la fois des ressources disponibles et des valeurs assignées aux paramètres liés à la procédure de génération. Pour obtenir une solution, il n'est pas nécessaire d'attendre la fin de la recherche. Nous ne pouvons pas savoir si le sous-ensemble obtenu à l'instant t est optimal mais seulement qu'il est meilleur que les précédents.
- **Les méthodes myopes** : Ces méthodes utilisent des critères de sélection myopes pour sélectionner les variables. Ces méthodes ne prennent pas en compte l'interaction entre variables et permettent de classer les variables en fonction de leur pouvoir discriminant. Ce type de méthodes est efficace et très rapide en particulier sur des problèmes comportant à la fois beaucoup de variables et d'individus.
- **Les méthodes en un seul scan de base** : Le processus de sélection s'effectue de la manière suivante : lors de la première étape, la variable la plus corrélée avec la variable endogène est sélectionnée. Lors de la deuxième étape, la variable qui est la plus partiellement corrélée avec la

variable endogène est sélectionnée et ainsi de suite. Afin de n'avoir qu'un seul scan de base, les mesures rapides de corrélation sont utilisées (coefficient de corrélation linéaire de Pearson, coefficient de corrélation de rangs de Kendall, etc.).

Le choix de la stratégie de recherche dépend de la taille de l'espace de représentation des données. En effet, si le nombre de variables est trop élevé, les méthodes de type exhaustif ne pourront pas être utilisées.

1.5.3 Classification des méthodes filtre

Nous avons répertorié les différentes méthodes de filtrage suivant deux axes : le type de méthodes filtre et le critère d'évaluation, tableau 3.

Type de méthodes	Complète	Heuristique	Aléatoire	Myope	En un seul scan de base
Critères d'évaluation	Information				
	Distance				
	Indépendance				
	Consistance				

Tableau 3 Classification des algorithmes de filtrage.

1.5.4 Méthodes complètes

Les méthodes complètes que nous avons recensées utilisent soit un critère de distance soit un critère de consistance. Ces méthodes étudient l'ensemble des sous-ensembles possible de t variables, $1 \leq t < p$. Certaines méthodes ne sont pas exhaustives : un processus d'élagage leur permet de réduire l'espace de recherche du sous-ensemble optimal.

1.5.4.1 Critère de distance

Le message de description de longueur minimale (MDLM), [32] : Les auteurs définissent l'utilité d'une variable et basent leur algorithme de recherche d'un sous-ensemble de variables pertinentes sur la distinction entre sous-ensemble utile de variables et sous-ensemble inutile. Ils effectuent une recherche exhaustive sur l'espace. Cette méthode est impossible à appliquer car elle est trop coûteuse en temps de calcul. La stratégie de recherche de cette méthode est exhaustive : tous les sous-ensembles

possibles sont générés. Le sous-ensemble ayant la plus faible utilité sera considéré comme le sous-ensemble optimal.

```
Entrée  X       ensemble total des variables initiales
        P(X)    ensemble des parties de X

Sortie  X*      ensemble de variables sélectionnées

Début MDLM
        X* ← X
        Pour tout X' ∈ P(X) Faire
                Si l'utilité de (X') < l'utilité de (X*) Alors
                        X* ← X'
                Fin Si
        Fin Pour
        Retourner X*
Fin MDLM
```

Algorithme 3 MDLM

1.5.4.2 Critère de consistance

PRESET de Modrzejewski [33] : Modrzejewski s'intéresse au problème de la présence de classes non disjointes c'est à dire à la présence d'objets inconsistants. Des objets inconsistants sont des objets qui ont les mêmes modalités pour les variables exogènes mais qui appartiennent à deux classes différentes. PRESET est un algorithme basé sur la théorie des ensembles approximatifs (rough sets). La première étape de cette méthode réalise une recherche complète. Ensuite, PRESET sélectionne le sous-ensemble de variables entraînant la même consistance sur l'ensemble d'apprentissage que l'ensemble initial de variables. Ce sous-ensemble se nomme réduction R de X. Toutes les variables n'appartenant pas à cette réduction sont éliminées et le sous-ensemble optimal est R.

Focus de Almuallim et Dietterich : Almuallim et Dietterich [7] présentent une méthode qui implique un haut degré de recherche à travers l'espace. Leur algorithme Focus recherche la combinaison minimale d'attributs qui discrimine parfaitement les classes, c'est ce que l'on appelle l'ensemble minimum de variables ou Min-Feature Bias. L'ensemble minimum est défini comme suit : si deux sous-ensembles sont consistants alors celui qui est constitué par le plus petit nombre de variables est préféré. Cette méthode commence par s'occuper des variables une par une, puis des paires de variables, puis des triplets de variables, ... Elle s'arrête seulement quand elle trouve une partition pure de l'ensemble d'apprentissage : c'est à dire une partition dans laquelle chaque objet possède une classe

différente de celle des autres objets. Focus n'est pas affectée par l'introduction de variables non pertinentes. Elle travaille dans le domaine booléen sans bruit, c'est-à-dire que deux objets appartenant à deux classes différentes doivent nécessairement avoir des modalités différentes. Toutes les variables exogènes ainsi que la variable endogène sont de nature booléenne.

```
Entrée  X        ensemble initial des variables initiales

Sortie  X*       ensemble de variables sélectionnées

Début Focus
        Pour i = 1,⋯, p  Faire
                Pour tout X* ⊂ X de taille i  Faire
                        Si ¬∃ 2 individus qui ont les mêmes valeurs pour les variables exogènes mais qui
                        ∉ la même classe alors aller à 2 Alors
                                Retourner X*
                        Fin Si
                Fin Pour
        Fin Pour
        Retourner X*
Fin Focus
```

Algorithme 4 Focus

Focus-2, [30] : Cet algorithme est une amélioration de Focus. Toutes les variables exogènes ainsi que la variable endogène sont de nature booléenne. Trois nouvelles notions sont utilisées :

- Le conflit : un conflit est généré par deux paires d'objets ω_i et ω_j appartenant à deux classes différentes et est décrit par un vecteur composé de p bits (autant de bits que de variables descriptifs). Un 0 au $k^{ième}$ bit indique que la valeur de la $k^{ième}$ variable de l'individu ω_i est différente de celle de ω_j et un 1 indique qu'elles sont égales.
- Un sous-ensemble X' est suffisant si et seulement si tout conflit généré par l'ensemble d'apprentissage est couvert par des variables de X'.
- $M(A,B)$ représentent l'ensemble des sous-ensembles contenant au moins toutes les variables de A mais aucune variable de B.

On cherche toujours à obtenir l'ensemble minimum de variables pertinentes, les auteurs étudient les conflits générés par l'ensemble d'apprentissage. Ainsi, si une variable est nécessaire pour résoudre un conflit alors tout sous-ensemble ne contenant pas cette variable ne sera pas examiné.

Focus-2 utilise une queue FIFO dont chaque élément représente un sous-ensemble de l'espace des variables. L'idée principale de Focus-2 est de ne garder dans la queue que les parties intéressantes de l'espace des sous-ensembles, celles qui peuvent contenir la solution.

Entrée X ensemble initial des variables
G ensemble de tous les conflits générés à partir de l'ensemble d'apprentissage

Sortie X' ensemble des variables sélectionnées

Début Focus 2
Queue = $M_{\emptyset,\emptyset}$.
Répéter
Sortir le premier élément de la queue $(= M_{A,B})$
$\mathrm{X}' = A$
Soit g *un conflit de* G *qui ne couvre pas* A *tel que* $|X_g - B|$ *est minimum, où* X_g *est l'ensemble des variables couvrant* g
Pour tout $X \in (X_g - B)$
Si $(A \cup \{X\})$ est suffisant Alors
Retourner $(A \cup \{X\})$
Insérer $M_{(A\cup\{X\}),Out}$ en fin de queue
Fin Si
Fin Pour
$\mathrm{X}' = \mathrm{X}' \cup \{X\}$
Fin Répéter
Fin Focus 2

Algorithme 5 Focus 2

Les méthodes Branch and Bound : L'utilisation d'une recherche exhaustive n'est possible que pour les problèmes à faible dimension, c'est à dire comportant peu de variables. Une alternative à la recherche exhaustive sont les algorithmes Branch & Bound (B&B) présenté par Narendra et Fukunaga [34]. Ils sont basés sur l'hypothèse que la fonction d'évaluation adoptée remplie la condition de monotonie.

Définition : Une fonction possède la propriété de monotonie si elle considère qu'un sous-ensemble de variables n'est pas meilleur qu'un ensemble plus grand contenant ce sous-ensemble.

La sélection de variables de type B&B commence avec l'ensemble initial des variables (backward elimination) et cherche ensuite la variable à retirer en maximisant la fonction d'évaluation. Le premier algorithme de sélection de variables de type Branch & Bound est B&B de [34]. En partant du nœud

père X (l'ensemble initial des variables), pour chaque nœud fils $X^k = X - X_k$, si $F(X^k)$est supérieur à un seuil τ alors le noeud X^k sera développé sinon il sera élagué. Cette opération est effectuée de manière récursive pour chaque nœud obtenu jusqu'à ce que tous les nœuds soient explorés.

ABB [2] : L'algorithme ABB proposé par [2] reprend le même principe que B&B. La seule différence réside dans la fonction d'évaluation monotonique, ici le taux d'inconsistance. L'algorithme proposé est un algorithme de type Branch & Bound auquel les auteurs ont inclus un seuil d'inconsistance ς. L'algorithme débute avec l'ensemble total des variables X, retire une variable afin de générer le sous-ensemble X_j^l, où l est le niveau courant et j spécifie les différents sous-ensemble du niveau l. Si $F(X_j^l) > F(X_j^{l+1})$, X_j^l cesse de s'accroître (la branche est élaguée). Sinon, elle grandit au niveau $l+1$, en d'autres termes une variable est retirée. En résumé, l'algorithme ABB cherche le plus petit sous-ensemble X_j dont le taux d'inconsistance est ς.

1.5.5 Méthodes Heuristiques

1.5.5.1 Critère d'information

Algorithme gourmand basé sur l'information mutuelle (GIM), [35] : Cet algorithme approxime le Min Feature Bias, précédemment défini et utilisé dans l'algorithme Focus. Il n'utilise que des variables de nature booléenne. En partant de l'ensemble vide, à chaque itération, on ajoute à la solution partielle une variable dont on estime la pertinence à l'aide d'un critère basé sur l'information. Soit un ensemble de variables X', l'algorithme GIM crée, en premier lieu, la partition de l'ensemble d'apprentissage la plus fine au moyen de X', on obtient $2^{|X'|}$ éléments. Il est possible de visualiser cette partition au moyen d'un arbre binaire dont les sommets terminaux contiennent les éléments de la partition la plus fine.

Soit n_i et p_i, le nombre d'individus respectivement positifs et négatifs appartenant à l'élément i de la partition créée à partir de X'. Alors, l'entropie de X' est définie comme suit:

$$E(X') = -\sum_{i=0}^{2^{|X'|}-1} \frac{p_i + n_i}{n} \left[\frac{p_i}{p_i + n_i} \log_2 \frac{p_i}{p_i + n_i} + \frac{n_i}{p_i + n_i} \log_2 \frac{n_i}{p_i + n_i} \right].$$

GIM sélectionne la variable qui, ajoutée à la solution partielle X', entraîne l'entropie minimum.

```
Entrée  X      ensemble de toutes les variables
        X'     sous-ensemble de X

Sortie  X'     sous-ensemble optimal de variables

Début GIM
        X'= φ
        Tant que E(X') ≠ 0 Faire
                Pour k = 1, p − 1 Faire
                        Calculer E(X' ∪ X_k)
                Fin Pour
                Sélectionner la variable X* ayant l'entropie la plus faible
                X' = X' ∪ X*
        Fin Tant que
        Retourner X'
Fin GIM
```

Algorithme 6 GIM

<u>Algorithme basé sur le critère de couverture de Markov, [36]</u> : Les auteurs utilisent la couverture de Markov. L'algorithme développé est un algorithme de Backward elimination . Un ensemble de définitions est nécessaire :

- Soit X_{-k} un ensemble de variable ne contenant pas X_k, X_{-k} est la couverture de Markov de X_k si et seulement si X_k est conditionnellement indépendant de $\mathrm{X} - \mathrm{X}_{-k} - \{X_k\}$. C'est à dire si : $P(X_k / \mathrm{X}_{-k}, Z) = P(X_k / \mathrm{X}_{-k})$, $\forall Z \notin \mathrm{X}_{-k}$.
- Soit x un vecteur de valeurs des variables, $x_{\mathrm{X}'}$ est la projection surjective de x sur X'.

L'algorithme se déroule de la manière suivante : pour chaque variable X_k de X, un sous-ensemble X_{-k} est sélectionné. Ce sous-ensemble X_{-k} doit regrouper les variables les plus corrélées avec X_k, mais ne contiendra pas X_k. Ensuite, la distance entre X_{-k} et une couverture de Markov de X_k est calculée de la manière suivante : $\delta_{\mathrm{X}'}(X_k / \mathrm{X}_{-k}) = \sum_{x_{\mathrm{X}_{-k}}, x_k} P(\mathrm{X}_{-k} = x_{\mathrm{X}_{-k}}, X_k = x_k)$.

Si $\delta_{X'}(X_k / X_{-k})$ est égale ou très proche de 0 alors X_{-k} est un couverture de Markov pour X_k. La variable X_k pour laquelle $\delta_{X'}(X_k / X_{-k})$ est la plus proche de 0 sera éliminée. Le critère d'arrêt est sous la forme d'un nombre de variables restantes η spécifié par l'utilisateur.

Entrée X ensemble de toutes les variables
η nombre de variables restantes
Sortie X' ensemble optimal des variables

Début CM
X'= X
Répéter
Pour $X_k \in X'$ Faire
Sélectionner X_{-k} sous-ensemble de variables corrélées avec X_k
Calcul de $\delta_{X'}(X_k / X_{-k})$
Fin pour
Choisir X^*_k telle que $\delta_{X'}(X^*_k / X^*_{-k})$soit la plus petite
$X' = X - X^*_k$
Jusqu'à ce que $\text{card}(X') = \eta$
Retourner X'
Fin CM

Algorithme 7 CM

- **DTM, [37] :** L'auteur travaille sur le problème du traitement du langage naturel. Elle utilise une approche filtre hybride dans un algorithme de type apprentissage par cas. Cette approche utilise les arbres de décisions (C4.5) pour spécifier les variables à inclure dans la recherche des k plus proches voisins d'une instance.

1.5.5.2 Critère de distance

Algorithme gourmand simple (GS), [35] : Cet algorithme a comme base l'heuristique gourmande [38] de la couverture minimum d'un ensemble. En partant de l'ensemble vide, il sélectionne à chaque fois la variable qui couvre le plus grand nombre de conflits qui n'ont pas encore été exploités (au départ la totalité des conflits). Ces derniers sont alors éliminés de l'ensemble des conflits. Et on réitère jusqu'à ce que cet ensemble soit vide. Toutes les variables descriptives ainsi que la variable à prédire doivent être de nature booléenne.

Entrée X ensemble total des variables initiales
Ω ensemble des instances
G ensemble des conflits générés par Ω

Sortie X' ensemble de variables sélectionnées

Début GS
$X' = \phi$
Tant que G $\neq \varnothing$ Faire
Pour k=1 à r-1 Faire
Compter le nombre de conflits couverts par y_k
Fin Pour
Sélectionner la variable X^* couvrant le plus de conflits
$X' = X' \cup \{X^*\}$
éliminer de G les conflits couverts par X^*
Fin Tant que
Retourner X'
Fin GS

Algorithme 8 GS

1.5.5.3 Critère d'indépendance

Algorithme du Khi2 [39] : La majorité des algorithmes existants traite des variables discrétisées, l'auteur a élaboré un algorithme qui traite les variables continues. Dans le cas où les variables sont mixtes, l'algorithme n'est pas en mesure de déterminer l'ensemble minimal des variables pertinentes. Cet algorithme réalise en même temps la discrétisation des variables et l'élimination des variables non pertinentes. Il est basé sur la statistique du χ^2 et consiste à fusionner successivement les valeurs de la variable à discrétiser jusqu'à ce qu'une condition d'arrêt soit atteinte.

1.5.5.4 Critère de consistance

Algorithme gourmand pondéré, [35] : A la différence de l'algorithme GS, l'incrémentation dépend, ici, du nombre de variables qui couvrent le conflit. Si une variable couvre toute seule un conflit, alors elle doit obligatoirement faire partie de l'ensemble solution de variables. Cet algorithme n'utilise que des variables de nature booléenne. Soient X_k , un attribut, g_κ un conflit couvert par X_k, G_{X_k} l'ensemble des conflits couvert par X_k, alors le score de X_k sur le conflit g_K :

- $score_{X_k} = \sum_{g_k \in G_{X_k}} \frac{1}{(\text{nombre de variables couvrant } g_k) - 1}$

- Il est inversement proportionnel au nombre de variables couvrant g_K.
- Il est élevé si peu de variables couvrent le conflit g_K,
- Il devient infini si g est exclusivement couvert par X_k, dans ce cas, X_k fera obligatoirement partie du sous-ensemble de variables sélectionnées.

Cet algorithme calcule le score d'une variable en tenant compte du comportement des autres variables sur un ensemble précis de conflit. La meilleure variable est celle qui a le meilleur score.

Entrée X ensemble total des variables initiales
Ω ensemble des instances
G ensemble des conflits générés par Ω

Sortie X' ensemble de variables sélectionnées

Début GP
X'= ϕ
Tant que $G \neq \phi$
Pour $k = 1$ à $r - 1$ Faire
Calculer $score_{X_k}$
Fin Pour
Sélectionner la variable X^* ayant le plus grand score
X'= X'$\cup X^*$
Eliminer de G les conflits couverts par X^*
Fin Tant que
Retourner X'
Fin Gourmand Pondéré

Algorithme 9 GP

Relief, [27] : Avant d'expliquer cet algorithme, il est nécessaire de définir un ensemble de notions :

- L'instance de réussite la plus proche est l'instance qui est à la plus petite distance parmi toutes les instances appartenant à la même classe que l'instance choisie.
- L'instance d'échec la plus proche est l'instance qui a la plus petite distance parmi toutes les instances appartenant à une classe différente de celle de l'instance choisie.

Relief est un algorithme basé sur l'attribution de poids aux variables. L'algorithme commence par choisir un échantillon d'individus dont le nombre est fourni par l'utilisateur. Il recherche ensuite pour chaque instance la plus proche instance de réussite et la plus proche instance d'échec en se basant sur une mesure de distance. Il met à jour les poids des différentes variables qui sont initialisés à zéro au

début. Cette démarche est basée sur une idée intuitive qui est : une variable est plus pertinente qu'une autre si elle distingue une instance de son instance d'échec la plus proche, et moins pertinente si elle distingue une instance de son instance de réussite la plus proche. Après avoir épuisé toutes les instances de l'échantillon, Relief choisit toutes les variables ayant un poids supérieur ou égal à un certain seuil. Ce seuil peut être déterminé de manière automatique en utilisant une fonction qui dépend du nombre d'instances dans l'échantillon.

Entrée X ensemble total des variables initiales
Ω ensemble des instances

Sortie X' ensemble de variables sélectionnées

Début Relief
Fixer un seuil τ pour filtrer les variables ayant un poids supérieur ou égal
$X' = \phi$
Tirer aléatoirement un échantillon $\Omega' \subseteq \Omega$
Initialiser tous les poids $w_j; j = 1, \cdots, p$ à zéro
Pour $t = 1, T$ Faire /*T est le nombre d'itérations choisit arbitrairement */
Choisir aléatoirement une instance $\omega \in \Omega'$
Chercher sa plus proche instance $\omega_=$ de la même classe et sa plus proche instance $\omega_{\neq}$ de classe différente :
Pour $j = 1, ..., p$ Faire

$$w_j = w_j - \frac{\delta(X_j(\omega), X_j(\omega_=))}{T} + \frac{\delta(X_j(\omega), X_j(\omega_{\neq}))}{T}$$

Fin Pour
Fin Pour
Pour $j = 1, r$ Faire
Si $w_j \geq \tau$ Alors
$X' = X' \cup \{X_j\}$
Fin Si
Retourner X'
Fin Pour
Fin Relief

Algorithme 10 Relief

Avec $\delta(X_j(\omega), X_j(\omega')) = \begin{cases} \dfrac{|X_j(\omega) - X_j(\omega')|}{|\max_{X_j} - \min_{X_j}|} \text{ si } X_j \text{ est continue} \\ 1 \text{ si } X_j \text{ est qualitative et } X_j(\omega) \neq X_j(\omega') \\ 0 \text{ si } X_j \text{ est qualitative et } X_j(\omega) = X_j(\omega') \end{cases}$

La dissimilarité entre ω et $\omega' : d(\omega, \omega') = \sum_{j=1}^{r} \delta(X_j(\omega), X_j(\omega'))$.

Relief est capable de travailler avec des variables bruitées et corrélées et demande un temps d'exécution linéaire par rapport au nombre de variables et au nombre d'instances. Cet algorithme est également capable de traiter des données nominales et continues. L'inconvénient majeur est qu'il n'apporte pas d'amélioration avec des variables redondantes. Il génère souvent un sous-ensemble de taille non optimale en présence de variables non redondantes. Ce problème peut être résolu par une recherche ultérieure approfondie dans le sous-ensemble de variables obtenues par Relief. Aussi l'un des inconvénients de Relief est qu'il ne travaille que sur des classes binaires.

Variantes de Relief, [40] : Relief a connu de nombreuses variantes que nous allons succinctement présenté :

- **ReliefA :** Dans Relief, les variables bruitées ou redondantes peuvent affecter négativement la sélection du plus proche voisin. Pour accroître la fiabilité du résultat obtenu, Relief va subir une extension qui consiste à chercher les k plus proches voisins. L'itération pour la modification du poids de l'attribut X_j se calcule comme suit :

$$w_j = w_j - \frac{\sum_{k=1}^{K} \delta(X_j(\omega), X_j(\omega_{=}^{k}))}{TK} + \frac{\sum_{k=1}^{K} \delta(X_j(\omega), X_j(\omega_{\neq}^{k}))}{TK}$$

  Pour permettre à ReliefA de traiter des ensembles de données incomplets, la fonction $\delta(X_j(\omega), X_j(\omega_{=}))$ sera étendue aux valeurs manquantes des variables.

- **ReliefB :** Si au moins l'une des deux instances a une valeur d'une variable donnée inconnue, alors on affectera $1-\frac{1}{q^j}$ à la valeur de $\delta\left(X_j(\omega), X_j(\omega_=)\right)$, où q^j est le nombre de valeurs différentes observées sur X_j.

- **ReliefC :** Cette variante ressemble à ReliefB, sauf que durant la mise à jour de l'estimation de w_j, les contributions de telles différences (calculées à partir d'instance dont au moins une valeur d'une variable est inconnue) sont ignorées, avec une normalisation appropriée. L'idée est que les valeurs inconnues doivent être ignorées au cours de l'estimation et si un nombre suffisant d'instances d'apprentissage est fourni, l'estimation obtenue doit converger vers l'estimation correcte.

- **ReliefD :** Il calcule la probabilité que deux instances données ont des valeurs différentes pour une variable donnée :

  - Si une instance ω a une valeur inconnue X_j :

    $$\delta\left(X_j(\omega), X_j(\omega')\right) = 1 - P\left(X_j = X_j(\omega') / Y(\omega)\right)$$

  - Si les deux instances ont des valeurs inconnues X_j :

    $$\delta\left(X_j(\omega), X_j(\omega')\right) = 1 - \sum_{k=1}^{q^k} P\left(X_j = x_k / Y(\omega)\right) \times P\left(X_j = x_k / Y(\omega')\right)$$

- **ReliefE et ReliefF :** Les auteurs de Relief traitent les problèmes à classes multiples en les ramenant à plusieurs problèmes à deux classes. Cette solution semble insatisfaisante. Pour pouvoir utiliser Relief dans la pratique, l'algorithme doit être capable de traiter les problèmes à classes multiples sans avoir à effectuer des changements à priori et à déformer l'espace de représentation qui peuvent affecter le résultat final. Pour cela deux extensions de l'algorithme ReliefD ont été proposées pour traiter les problèmes à classes multiples :

  - ReliefE : La plus proche instance de classes différentes d'une instance donnée ω est défini comme le plus proche voisin de toutes les classes différentes. C'est une généralisation directe de Relief.

  - ReliefF : Au lieu de chercher la plus proche instance de classe différente, l'algorithme cherche la plus proche instance pour chaque classe différente et fait la moyenne de leur contribution

pour la mise à jour de w_j comme dans ReliefA. La moyenne est pondérée avec la probabilité antérieure de chaque classe :

$$w_j = w_j - \frac{\sum_{k=1}^{K} \delta(X_j(\omega), X_j(\omega_=^k))}{TK} + \sum_{C \neq Classe(\omega)} \left[P(C) \times \frac{\sum_{k=1}^{K} \delta(X_j(\omega), X_j(\omega_{\neq}^k))}{TK} \right].$$

L'idée est que l'algorithme doit évaluer la capacité d'une variable à séparer chaque paire de classes sans se soucier de savoir qu'elles sont les deux classes les plus proches par rapport aux autres.

Entrée X ensemble de toutes les variables
t nombre de variables à sélectionner
w_1 et w_2 poids
Sortie X' sous-ensemble de variables
Début POE ACC
X'= { }
$POE(X^*) = -1$
Pour k=1 à p Faire
Si $POE(X_k) > POE(X^*)$ Alors
$X^* = X_k$
Fin si
Fin Pour
X'= X'$\cup$ {X^*}
Tant que $|X'| \neq t$ Faire
$w_1(POE(X^*)) + w_2(ACC(X^*)) = -1$
Pour chaque $X_k \in X - X'$ Faire
Si $w_1(POE(X_k)) + w_2(ACC(X_k)) > w_1(POE(X^*)) + w_2(ACC(X^*))$
Alors
$X^* = X_k$
Fin Si
Fin Pour
X'= X'$\cup$ {X^*}
Fin Tant que
Retourner X'
Fin POE ACC

Algorithme 11 POE ACC

POE-ACC de Mucciardi et Gose, [41] : L'algorithme POE-ACC est un algorithme de forward selection. Le critère d'arrêt de cet algorithme est un nombre t, fixé par l'utilisateur, de variables

sélectionnées. Son critère de sélection pour la variable X_k est la somme pondérée de deux mesures : la probabilité d'erreur de X_k, $(POE(X_k))$ et le coefficient de corrélation de X_k avec les variables descriptives non encore sélectionnées, $(ACC(X_k))$. Comme les valeurs des paramètres sont choisies par l'utilisateur, les solutions obtenues ne sont pas optimales.

$POE(X_k)$ se définit de la manière suivante :

$POE(X_k) = \sum_{i=1}^{m^Y} \sum_{j=1}^{m^k} \left[P(X_k = m_j^k) \times P(Y = m_i^Y / X_k = m_j^k) \right] - P_{prédiction}$, avec m^y le nombre de modalités de la variable à prédire Y, et, $P_{prédiction} = \max_{i=1}^{m^Y} \left(P(X_k = m_v^k) \times P(Y = m_i^Y / X_k = m_v^k) \right)$ pour une modalité m_v^k de X_k.

Nous considérons comme des méthodes heuristiques, les méthodes utilisant les comparaisons par paires, les Support Vecteur Machine et les Réseaux de Neurones.

1.5.6 Sélection et comparaison par paires

Algorithme Pouvoir Discriminant But (PDOBut) [42] : Cet algorithme de filtrage effectue une recherche heuristique d'un sous-ensemble de variables et réalise la sélection de variables à l'aide d'un critère de sélection utilisant les comparaisons par paires. Avant de présenter le processus de cet algorithme, il convient de définir le critère du Pouvoir Discriminant Original But (PDOBut) d'une variable par rapport à un ensemble de variables. En effet, c'est ce critère que [42] utilise pour sélectionner des variables. Le PDOBut de la variable X_k par rapport au sous-ensemble de variables L, $(X_k \notin L)$, est une fonction du nombre de couples dont les individus sont discriminés par X_k et par aucune autre variable de L et qui ne possèdent pas la même modalité pour la variable endogène :

$$PDOBut(X_k | L) = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \overline{\varphi}_{ij}^{k} \overline{\varphi}_{ij}^{Y} - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \underset{\substack{k'=1 \\ k' \neq k}}{\overset{l}{Max}} \overline{\varphi}_{ij}^{k'} \overline{\varphi}_{ij}^{Y} .$$

Cette méthode se décompose en trois étapes :

- Sélection des variables essentielles : Les variables essentielles sont des variables qui permettent à elles seules de discriminer les objets de certaines paires. C'est à dire : Pour toute variable $X_k \in X$, si $PDOBut(X_k, X - \{X_k\}) \neq 0$ alors la variable $X_k \in X$ est sélectionnée. Ces variables sont définitivement sélectionnées. Si aucune variable n'est essentielle, l'algorithme débute avec la deuxième étape.
- Sélection des variables restantes : Les variables qui augmentent le plus le PDOBut du sous-ensemble S de variables sélectionnées à l'étape précédente sont ajoutées.
- Suppression des variables redondantes : Les variables redondantes après la sélection des nouvelles variables sont éliminées. Pour chaque variable X_k, si $PDOBut(X_k, S - \{X_k\}) = 0$ alors la variable X_k est supprimée de S.

1.5.7 Sélection et Support Vecteur Machine

La Méthode de Stoppiglia et Dreyfus, [43] : Cette méthode permet d'ordonner les variables à l'aide des SVM. Pour ordonner les variables exogènes, la procédure d'orthogonalisation de Gram-Schmidt [44] est utilisée. Son originalité réside dans un nouveau critère d'arrêt : [44] définissent un risque dont le valeur est fixée en fonction du problème étudié et calculent la fonction de distribution cumulative de l'ordre des variables à chaque itération. Si la valeur de cette fonction est inférieure au risque, la variable sélectionnée est conservée et la procédure est réitérée. Si la valeur de cette fonction est supérieure au risque, la variable sélectionnée est supprimée et la procédure s'arrête. Grâce à ce critère d'arrêt, cette méthode fournit un sous-ensemble de variables ordonnées.

L'algorithme SVM-RFE, [45] : C'est un algorithme de Backward Sélection : à partir de l'ensemble de toutes les variables, ces dernières sont éliminées itérativement. [45] considèrent le problème dual des SVM.

L'algorithme VS-SSVM, [46] : [46] appliquent les SVM sur différents échantillons d'individus et, à l'aide du bagging, agrègent leurs résultats. Après cette étape d'agrégation, on obtient un sous-ensemble de variables considérées comme pertinentes.

1.5.8 Sélection et Réseaux de neurones

Heuristique pour sélection de variables : La méthode HVS est basée sur les réseaux de neurones. Pour chaque variable X_j, [47, 48] estiment sa contribution relative aux résultats fournis par le réseau de neurones. C'est une méthode de backward selection. Le point de départ est l'ensemble initial des variables sur lequel on applique le réseau de neurones. Ensuite, la contribution relative de chaque variable est calculée. La variable qui possède la contribution la plus faible est supprimée. Et, ce processus est réitéré. On obtient ainsi p réseaux (autant que de variables). Le premier est constitué de la totalité des variables c'est à dire de p variables, le deuxième possède (p-1) variables, et ainsi de suite. Le sous-ensemble optimal de variables correspondra à celui utilisé par le réseau qui a un taux d'erreur minimal.

1.5.9 Méthodes aléatoires

1.5.9.1 Critère d'inconsistance

LVF [49] : LVF est une version filtre des algorithmes «Las Vegas». Ces derniers font des choix probabilistes qui les mènent plus rapidement vers une solution correcte. Un certain type de ces algorithmes utilise la stratégie aléatoire pour guider leur recherche de telle manière qu'une solution correcte est garantie même si des choix non judicieux ont été réalisés.

LVF choisit aléatoirement un sous-ensemble de variables et calcule sa cardinalité. Le meilleur sous-ensemble est initialisé avec l'ensemble complet des variables. Un sous-ensemble X' est choisi aléatoirement. Si X' a une cardinalité inférieure ou égale au meilleur sous-ensemble courant X^*, le pourcentage d'inconsistance de X' , Inconsistance(X') est calculé. Si ce dernier est inférieur ou égal à celui de X^*, X' devient le meilleur sous-ensemble courant. Le critère d'arrêt est soit un nombre d'itérations I fixé par l'utilisateur, soit un seuil τ, fixé par l'utilisateur, pour le pourcentage d'inconsistance. Le succès de cet algorithme réside dans son critère d'inconsistance. Ce critère spécifie jusqu'à quel point la réduction de la dimension des données peut être acceptée. Cet algorithme est simple à implémenter. Cependant, le nombre de variables sélectionnées est biaisé : il est toujours de l'ordre de la moitié du nombre de variables initiales.

Entrée X ensemble de toutes les variables
I nombre d'itérations
τ seuil d'inconsistance

Sortie X^* sous-ensemble optimal de variables

Début LVF
$X^* = X$
$i = 0$
Répéter
$i = i + 1$
Tirer aléatoirement un sous-ensemble X' de X
Si $\text{card}(X') \leq \text{card}(X^*)$ et $Inconsistance(X') \leq Inconsistance(X^*)$ Alors
$X^* = X'$
Fin Si
Jusqu'à $Inconsistance(X^*) \geq \tau$ ou $I = I_{\max}$
Retourner X^*
Fin LVF

Algorithme 12 LVF

Méthodes utilisant les Algorithmes Génétiques (AG) : Les AG sont des stratégies de recherche basées sur le principe de sélection naturelle. Une population de solutions possibles nommées chromosomes est maintenue. Les chromosomes sont sélectionnés, recombinés et mutés dans le but de faire évoluer une nouvelle population. Le processus est répété jusqu'à ce qu'une condition d'arrêt soit atteinte pour l'individu le plus adapté de la population ou quand un certain nombre de générations a été produit. Les AG sont très connus pour leur capacité à effectuer des recherches dans des espaces très grands et sans connaissance sur le domaine. Ils sont relativement insensibles au bruit. Donc ils sont idéaux pour des usages où les connaissances du domaine et les théories sont difficiles voire impossibles à obtenir. Lors de l'utilisation des AG, la chose la plus importante est de choisir une représentation bien appropriée et une fonction d'évaluation adéquate.

Dans les problèmes de sélection de variables, le principal intérêt est la représentation de l'espace de tous les sous-ensembles possibles de variables. La forme de représentation la plus simple est la représentation binaire. Chaque variable est un gène binaire et un individu est une chaîne binaire de longueur fixée, c'est à dire un sous-ensemble de variables. Un individu de longueur λ est un vecteur $X(.)$ constitué de λ composants.

Chaque composant x_j représente l'inclusion (1) ou l'élimination (0) de la variable qui lui est associée. La fonction d'évaluation F utilisée est, très souvent, la suivante : on considère qu'il y a m objets bien classés de ω_1 à ω_m et $(n-m)$ objets mal classés de ω_{m+1} à ω_n :

$F = \sum_{i=1}^{m} S_i * W_i - \sum_{j=m+1}^{n} S_j * W_j$, avec S_i le score de reconnaissance de l'objet i obtenu par le processus de classification, et W_i le poids assigné à l'objet i.

C'est la différence entre la somme des scores pondérés des reconnaissances correctes et la somme des scores pondérés des reconnaissances incorrectes. Les valeurs de F sont dépendantes du nombre d'exemples d'apprentissage et de leur poids. Afin de rendre utilisables les valeurs de F par les AG, on effectue les transformations suivantes : $F' = 100 - \left[\left(F / \sum_{i=1}^{n} W_i \right) * 100 \right]$. Ainsi, F' est toujours positives.

La sélection de variables s'effectue en suivant ces différentes étapes :

- Une population initiale est créée aléatoirement à partir de l'ensemble initial des variables. Chaque individu ou chromosome de la population est un vecteur de longueur p.
- La population est évaluée grâce à la fonction de fitness. Ceci permet de déterminer les meilleurs chromosomes de la population ou les chromosomes parents qui seront utilisés pour produire les nouveaux chromosomes.
- Les opérateurs génétiques (Crossover et mutation) sont appliqués aux chromosomes parents. Les nouveaux chromosomes ou chromosomes enfants sont ainsi produits.
- Les chromosomes enfants permettent de construire une nouvelle population.
- Si le critère d'arrêt est satisfait, l'AG fournit le chromosome solution qui correspond au meilleur sous-ensemble de variables, sinon, on continue le processus en allant à l'étape 2.

Les AG ont été utilisés par de nombreux auteurs tels que Guerra-Salcedo, Chen, Whitley et Smith [50], Vafaie et De Jong [51], [52].

GADistAI : Yang et Honovar [53], [54], présentent la méthode GADistAI (figure 8) qui mélange recherche locale et globale : c'est à dire les algorithmes génétiques qui représentent une recherche

globale et rapide dans des espaces de recherche très grands pour des problèmes d'optimisation difficiles et les réseaux de neurones qui représentent une recherche locale pour raffiner les solutions prometteuses qui ont déjà été trouvées. Les variables peuvent être de type numérique ou nominal.

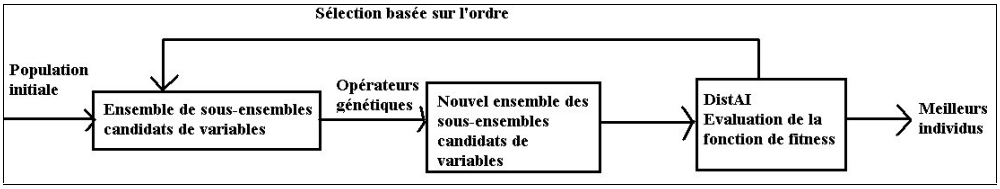


Figure 6. Processus de GADist AI.

L'algorithme débute avec une population initiale composée de différents sous-ensembles de variables. De nouvelles populations sont générées à partir des précédentes par l'application des opérateurs génétiques (crossover et mutation). DistAI est un algorithme d'apprentissage rapide et simple de construction de réseaux de neurones pour la classification. Il permet le calcul des fonctions d'évaluation, de la population et leur classification en fonction des valeurs de ces fonctions. Le meilleur individu est obtenu après la dernière génération.

1.5.10 Méthodes en un seul scan de base

1.5.10.1 Critère d'information

MIFS, [55] : Cette méthode utilise l'information mutuelle pour évaluer l'information contenue dans chaque variable et sélectionner le sous-ensemble de variables le plus informatif. Elle sélectionne le sous-ensemble de variables qui possède la plus grande information mutuelle avec la variable endogène tout en minimisant l'information mutuelle existant entre les variables exogènes sélectionnées. La ième meilleure variable sélectionnée X_i est celle qui satisfait :

$$X_i = Max_{X_i}\left[\left(I(Y,X_i)\right) - \beta\sum_{j=1}^{i-1} I\left(X_i,X_j\right)\right], \text{ Avec } I(X,Y) = H_X - H_{X|Y}.$$ H_X est l'entropie de la variable X et $H_{X|Y}$ est l'entropie de la variable X par rapport à la variable Y. β est un paramètre fixé par l'utilisateur.

1.5.10.2 Critère d'indépendance

La méthode CFS, [56] : Elle ordonne les différents sous-ensembles possibles de variables à l'aide d'une fonction d'évaluation basée sur une mesure de corrélation. Cette fonction d'évaluation est la suivante :

$$M_S = \frac{f\bar{r}_{ck}}{\sqrt{f + f(f+1)\bar{r}_{kk}}}$$

M_S se nomme le « mérite » du sous-ensemble S qui contient f variables. $\bar{r}_{ck}$ est la corrélation moyenne entre variable endogène et variable exogène. $\bar{r}_{kk}$ est la corrélation moyenne entre variables exogènes appartenant à S. Ce mérite permet de sélectionner des sous-ensembles contenant des variables qui sont fortement corrélées avec la variable endogène et non corrélées entre elles. Ainsi, les variables non pertinentes sont ignorées car elles sont trop faiblement corrélées avec la variable endogène et les variables redondantes sont également ignorées car elles sont trop fortement corrélées avec une ou plusieurs autres variables. Pour éviter de générer tous les sous-ensembles possibles de variables, la méthode CFS s'effectue soit dans un processus de forward selection soit dans un processus de backward selection. Dans un premier temps, les corrélations entre variable endogène et variable exogène et entre variables exogènes sont calculées. Pour la forward selection (resp. la backward selection), une variable est ajoutée (resp. ôtée) au sous-ensemble si l'ajout (resp. la suppression) de cette variable augmente le mérite du sous-ensemble. La complexité de cette méthode est en $O(p(n^2-n)/2)$.

Algorithme G3, [57] : Cet algorithme est basé sur une mesure de corrélation partielle. Le tableau $T_r(Y,X)$ de taille $(p+1)\times(p+1)$ contient les corrélations simples r_{YX_j} avec $j = 1 \grave{a}\ p$ et $r_{X_iX_j}$ et, son calcul est la première étape du processus de sélection. Cet algorithme fonctionne dans un processus de forward sélection donc il démarre avec un ensemble vide. La première variable ajoutée X* est celle qui est le plus corrélée avec Y. Le tableau des corrélations est alors recalculé. Il ne sera pas constitué de corrélations simples mais de corrélations partielles $(r(Y,X/X*))$. Ces corrélations partielles peuvent être calculés à partir des corrélations simples. Ainsi les accès à la base ne sont plus nécessaires. Dans les étapes suivantes, le processus recherche la variable la plus corrélée avec Y conditionnellement aux variables déjà sélectionnées. Le processus s'arrête quand le coefficient de détermination corrigé diminue. Le résultat fourni est un sous-ensemble de variables pertinentes.

1.5.11 Les méthodes myopes

Les méthodes myopes utilisent des critères de sélection myopes pour sélectionner les variables. Le résultat fourni par ce type de méthodes est une liste de variables triées par ordre de pertinence. Ces méthodes sont efficaces et très rapides en particulier sur des problèmes comportant à la fois beaucoup de variables et d'individus. La complexité de ce type de méthodes est de l'ordre de $O(NlogN)$ avec N le nombre d'individus. Tous les critères de sélection décrits dans la partie consacrée aux critères d'évaluation sont des méthodes de sélection en eux même.

1.5.12 Conclusion

L'avantage de approches filtre est qu'elles peuvent être utilisées en amont de n'importe quel algorithme d'induction, étant donnée leur indépendance totale vis à vis de celui ci. Cependant, elles ignorent totalement les effets du sous-ensemble de variables sélectionnées sur les performances de cet algorithme.

1.6 Les méthodes enveloppe

Ces approches ont été introduites par John, Kohavi et Plfeger [6]. Pour les auteurs, le défaut des méthodes filtre est lié au fait qu'elles ignorent totalement l'influence de l'ensemble de variables sélectionnées sur les performances de l'algorithme d'apprentissage utilisé. Pour résoudre ce problème, ils proposent une approche différente qui utilise le résultat de l'algorithme d'apprentissage comme fonction d'évaluation. L'algorithme d'apprentissage travaille sur la totalité des individus avec différents sous-ensembles de variables. Il fournit pour chacun d'eux la précision estimée sur le classement des nouveaux individus. Le sous-ensemble induisant le classifieur le plus précis est sélectionné. Le tableau 4 permet de présenter les différentes caractéristiques des méthodes enveloppe.

1.6.1 Oblivion, [58]

La méthode Oblivion de Langlay et Sage [58] utilise la méthode des plus proches voisins qui attribue à une nouvelle instance la classe du plus proche voisin mis en mémoire durant l'apprentissage. Oblivion prend comme point de départ l'ensemble de toutes les variables et supprime itérativement celles dont la suppression entraîne la meilleure amélioration.

Méthodes	**Point de départ**	**Méthode de recherche**	**Critère d'arrêt**	**Algorithme d'induction**
BEAM, [59]	aléatoire	comparaison	Pas mieux	plus proches voisins
CAP, [60]	comparaison	gloutonne	Toutes les variables utilisées	arbre de décision
Méthode de Doak, [61]	comparaison	comparaison	Pas assez mieux	arbre / bayésien
Méthode de John, Kohavi et Plfeger, [6]	comparaison	gloutonne	Pas mieux	arbre de décision
Oblivion, [58]	toutes les variables	gloutonne	pire	plus proches voisins
Bayes,[62]	aucune	gloutonne	pire	bayésien naïf
Race, [63]	comparaison	gloutonne	pas mieux	plus proches voisins
K2-AS, [64]	aucune variable	gloutonne	pire	réseaux bayésien
Skalak, [65]	aléatoire	mutation	limite de temps	plus proches voisins
Townsend-Weber/Kibler, [66]	toutes les variables	comparaison	pas mieux	plus proches voisins
NLC, [67]	liste triée de variables	comparaison	toutes les variables utilisées	bayésien naïf

Tableau 4 Les méthodes enveloppe

1.6.2 BEAM, [59]

La méthode BEAM de Aha et Bankert [59] utilise les k plus proches voisins comme algorithme d'induction et débute d'un ensemble de variables sélectionnées aléatoirement.

1.6.3 CAP, [60]

CAP de Caruana et Freitag [60] tente d'accélérer le processus d'évaluation à l'aide d'un schéma de mémorisation des arbres de décision.

1.6.4 La méthode de Moore et Lee

Moore et Lee, [63], propose un schéma qui accélère la sélection d'attributs en réduisant le pourcentage d'individus d'apprentissage durant l'évaluation.

1.6.5 La méthode NLC, [67]

[67] appliquent une approche enveloppe au résultat fourni par une méthode de sélection qui ordonne les variables et obtiennent ainsi un sous-ensemble de variables optimal. Les variables peuvent être de type continu ou discret et les problèmes multi-classe sont pris en compte.

En premier lieu, il convient de préciser la notion de NLC (Number of Label Change) qui représente le nombre de changements de classes. Dans ce but, nous prenons comme exemple la base Iris. Les individus sont projetés sur les axes Petal Width et Sepal Width. Par rapport à l'axe de Sepal Width, on ne peut pas obtenir d'intervalles ayant une classe majoritaire. Par contre, par rapport à l'axe de PetalWidth, sur l'intervalle $[0;0.6]$, seule la classe Setosa est touchée, sur l'intervalle $[1;1.3]$, seule la classe Versicolor est touchée, et sur l'intervalle $[1.8;2.5]$, seule la classe Virginica est touchée. Entre ces différents intervalles, 16 changements de classes sont comptabilisés et pour Sepal Width, 120. [67] comptabilisent pour chaque variable le nombre de changements de classes et concluent que la variable ayant le plus petit nombre de changements de classes classifie le mieux les individus. Leur méthode se déroule en deux étapes :

- Les variables sont ordonnées grâce au NLC.
- Cette étape est de type enveloppe. Le Bayésien Naïf est appliqué avec la première variable de la liste. Puis, il est appliqué à l'ensemble de variables composé de la première et de la deuxième

variable. Si le taux d'erreur obtenu est inférieur au précédent alors la deuxième variable est conservée sinon elle est supprimée. Et ainsi de suite, jusqu'à ce qu'il n'y ait plus de variables.

1.6.6 Conclusion

Le désavantage majeur des méthodes enveloppe est le coût calculatoire important dû à l'appel de l'algorithme d'induction pour chaque sous-ensemble considéré. Aussi ces méthodes sont peu utilisées en pratique.

1.7 Méthodes issues de l'économétrie

L'économétrie,[68] et [69], propose également un certain nombre de méthodes permettant de sélectionner les variables pertinentes. Des procédures statistiques permettent de déterminer quelles variables retirer ou ajouter dans un modèle. Ces démarches excluent tout raisonnement économique car elles aboutissent à des modèles économétriques qui sont souvent bons sur le plan statistique mais dont l'interprétation économique n'est pas toujours évidente. Il existe cinq méthodes qui permettent de retenir les variables les plus corrélées avec la variable à expliquer et les moins corrélées entre elles.

1.7.1 Toutes les régressions possibles

Cela revient à estimer toutes les combinaisons de régressions possibles ($2^p - 1$ possibilités avec p=nombre de variables explicatives). Le modèle sélectionné est celui dont le coefficient de détermination $R^2 = 1 - \frac{\sum e^2}{nVar(Y)}$ est maximal ($\sum e^2$ est la somme des erreurs au carré). Bien sûr, cette méthode ne peut être utilisée quand le nombre de variables explicatives est grand.

1.7.2 La méthode de backward elimination

Cette procédure consiste, sur le modèle complet à p variables explicatives, à éliminer les variables explicatives dont les t de Student sont en dessous du seuil critique. Le risque de colinéarité entre les variables initiales étant élevé, cette procédure n'est possible que si la première équation peut être effectivement estimée.

1.7.3 La méthode de forward selection

Cette méthode permet de sélectionner les variables exogènes qui maximise le coefficient de corrélation partielle entre elles et la variable endogène.

1.7.4 La régression pas à pas (stepwise regression)

Cette méthode est identique à la précédente avec la différence suivante : après l'ajout d'une nouvelle variable, le t de Student de chacune des variables explicatives préalablement sélectionnées est examiné. Les variables dont le t de Student est inférieur au seuil critique sont éliminées.

1.7.5 La régression par étage (stagewise regression)

C'est un processus permettant de minimiser les inter-corrélations entre les variables explicatives par étude du résidu.

1.8 Conclusion

Afin de récapituler l'ensemble des méthodes de sélection, nous avons construit le tableau 5. Ce tableau présente l'ensemble des méthodes ainsi que leur caractéristiques principale, à savoir le type d'approche, la direction de recherche, la type de méthode, le critère d'évaluation et le critère d'arrêt.

Il n'existe pas de méthode qui puisse traiter au mieux toutes les applications. Certaines sont plus efficaces que d'autres dans un contexte donné. Etant donnée la multitude de méthodes disponibles, il est difficile de décider quelle méthode utiliser pour une application particulière. Le temps imparti à la recherche d'un sous-ensemble et les caractéristiques des données (qualité, taille et type) peuvent nous aider à effectuer notre choix sur la technique la plus appropriée à la résolution du problème donné. Le choix de la méthode à appliquer se fera en fonction des ressources disponibles et des performances de chaque méthode.

La comparaison des algorithmes de sélection de variables semble difficile étant données leurs caractéristiques différentes. Comment comparer les résultats obtenus avec un algorithme traitant des données exclusivement quantitatives et ceux issus d'une méthode traitant uniquement des données qualitatives. Cependant, il ne nous paraît pas opportun d'utiliser des méthodes enveloppe car elles sont beaucoup trop coûteuses en temps de calcul et ne sont pas applicables pour des études réelles.

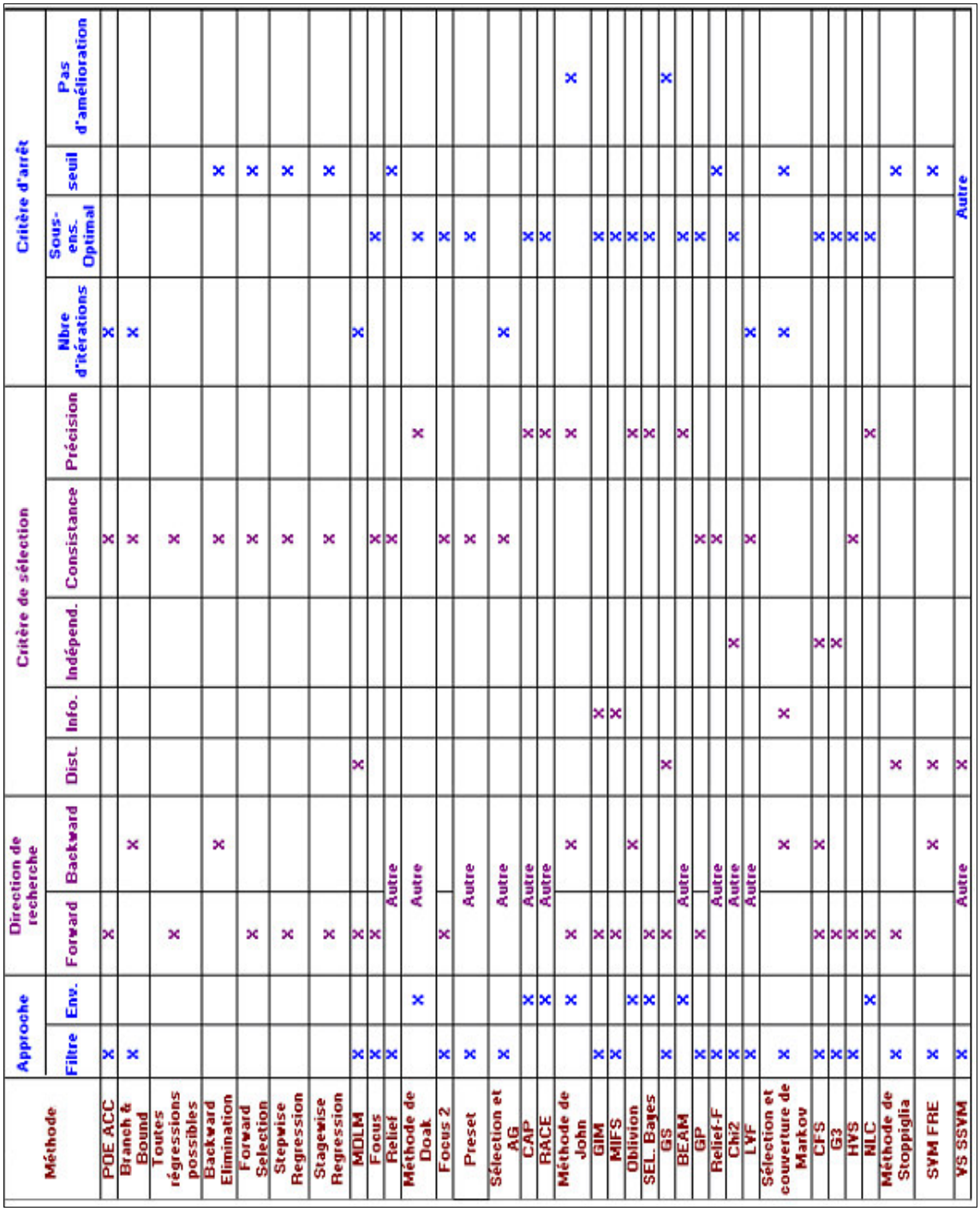
L'obtention systématique de la solution optimale n'est envisageable que par une recherche exhaustive, dans de très rares cas complète, sur l'espace des sous-ensembles de variables possibles. Tous les sous-ensembles de variables sont alors examinés : ceci est irréalisable lorsque l'ensemble d'apprentissage est volumineux. A l'inverse les méthodes heuristiques réduisent le nombre de sous-ensembles à étudier en guidant leur recherche sur l'espace des sous-ensembles possibles. Elles génèrent de bonnes solutions mais ne trouvent pas la solution optimale.

En raison de la quantité de plus en plus importante des données à traiter, il ne nous semble pas réaliste de nous attarder sur les méthodes filtre complètes ou sur les méthodes enveloppe. Les méthodes ne traitant que les variables booléennes telles que les méthodes de [30] ne sont pas adaptées pour les problèmes concrets. Les méthodes ne traitant qu'un seul type de données telles que Focus, Focus2 ou PRESET ne sont pas utilisables pour les mêmes raisons. Les méthodes telles que celle de Schlimmer qui ne traitent pas les données volumineuses ne pourra pas être utilisées pour les bases de données récentes. Les méthodes ne fournissant qu'une liste de variables classées par ordre de pertinence (ReliefF, les méthodes myopes,…) ne sont pas applicables telles quelles du fait même de la forme du résultat.

Les méthodes paraissant les plus efficaces sont les méthodes filtre en un seul scan de base. En effet, elles sont rapides, peu coûteuses, et présentent de bons résultats. Les méthodes filtre heuristiques utilisant les algorithmes génétiques ou les SVM ou encore les réseaux de neurones semblent également relativement efficaces.

Il n'existe malheureusement pas de méthode « miracle » capable de s'adapter à toutes situations. La réussite d'une méthode de sélection tient en grande partie au choix judicieux de la méthode à appliquer en fonction du contexte : nature des variables, taille de l'ensemble d'apprentissage,… Malgré tout, il est possible d'espérer un algorithme de sélection qui tende le plus souvent vers un sous-ensemble de variables pertinentes de taille minimale.

Dans la section suivante, nous effectuons une comparaison des méthodes existantes qui nous ont parues les plus intéressantes ou qui sont les plus connues.



Méthode	Approche		Direction de recherche		Critère de sélection					Critère d'arrêt			
	Filtre	Env.	Forward	Backward	Dist.	Info.	Indépend.	Consistance	Précision	Nbre d'itérations	Sous-ens. Optimal	seuil	Pas d'amélioration
POE ACC	x		x					x		x			
Branch & Bound	x			x				x		x			
Toutes régressions possibles			x					x					
Backward Elimination				x				x				x	
Forward Selection			x					x				x	
Stepwise Regression			x					x				x	
Stagewise Regression			x					x				x	
MDLM	x		x		x					x			
Focus	x		x					x			x		
Relief	x			Autre				x			x	x	
Méthode de Doak		x		Autre					x		x		
Focus 2	x		x					x			x		
Preset	x			Autre				x			x		
Sélection et AG	x			Autre				x		x			
CAP		x		Autre					x		x		
RACE		x		Autre					x		x		
Méthode de John		x	x	x					x				x
GIM	x		x			x					x		
MIFS	x		x			x					x		
Oblivion		x		x					x		x		
SEL. Bayes		x	x						x		x		
GS	x		x		x								x
BEAM		x		Autre					x		x		
GP	x		x					x			x	x	
Relief-F	x			Autre				x				x	
Chi2	x			Autre			x				x		
LVF	x			Autre				x		x			
Sélection et couverture de Markov	x			x		x				x		x	
CFS	x		x	x			x				x		
G3	x		x				x				x		
HVS	x		x					x			x		
NLC		x	x						x		x		
Méthode de Stoppiglia	x		x		x							x	
SVM FRE	x			x	x							x	
VS SSVM	x			Autre	x						Autre		

Tableau 5 Récapitulatif des méthodes de sélection de variables

2 Expérimentations et comparaisons des méthodes de sélection existantes.

Afin de mettre en évidence les caractéristiques et qualités qu'une méthode de sélection efficace devrait posséder, nous avons décidé de tester un ensemble de méthodes existantes et nous paraissant efficaces. Cela nous permettra de dégager les avantages et inconvénients de chaque méthode et d'apprécier la qualité de leurs résultats.

2.1 Choix des méthodes.

Nous avons choisi un ensemble de quatre méthodes de sélection : MIFS, LVF, ReliefF et une méthode myope utilisant comme critère d'évaluation l'entropie de Shannon. Nous allons expliquer pourquoi nous avons fait ce choix :

- Les méthodes enveloppe n'ont pas été retenues car elles sont bien trop coûteuses en temps de calcul et elles ne sont pas applicables sur des problèmes réels.
- Les méthodes filtre complètes ne sont pas testées parce qu'elles deviennent elles aussi trop coûteuses en temps de calcul dès que le nombre de variables augmentent.
- Les méthodes économétriques ne sont pas étudiées pour les mêmes raisons précédemment citées.
- ReliefF est le représentant des méthodes filtre heuristiques : c'est l'une des méthodes les plus connues. ReliefF permet de traiter tous types de variables, ainsi que les problèmes multiclasses. Il est robuste au données volumineuses et bruitées.
- La méthode MIFS a montré de bons résultats dans la littérature,[55]. C'est une méthode rapide et de faible complexité grâce au fait qu'elle n'effectue qu'un seul scan des données.
- Nous sélectionnons l'entropie de Shannon comme critère d'évaluation pour la méthode myope car c'est un critère de référence. Les méthodes myopes sont très rapides puisqu'elles traitent chaque variable indépendamment des autres
- La méthode PDOBut nous semble originale car elle utilise un critère basé sur les comparaisons par paires et les résultats présentés dans [42] sont intéressants.

2.2 Présentation détaillée des méthodes expérimentées.

2.2.1 PDOBut

Comme nous l'avons vu précédemment, l'algorithme PDOBut est essentiellement basé sur les comparaisons par paires. Il effectue une recherche heuristique d'un sous-ensemble de variables et réalise la sélection à l'aide de deux critères : l'un myope, l'autre contextuel. Il se termine lorsqu'il a obtenu un sous-ensemble de variables nécessaire et suffisant à la discrimination de tous les individus. En présence de variables bruitées et/ou non pertinentes, PDOBut ne cherchera pas à discriminer les individus de toutes les paires mais se déroulera jusqu'à l'obtention, avec le sous-ensemble de variables sélectionnées, de la même inconsistance qu'avec l'ensemble initial des variables.

D'après [42], la complexité algorithmique de PDOBut est en $O(pc)$ avec p le nombre de variables et c le nombre de paires dont les individus ne sont pas discriminés.

Le critère de sélection contextuel utilisé par cet algorithme permet de traiter les situations où les variables sont corrélées. De plus, l'algorithme possède une étape qui permet de supprimer les variables redondantes et les données bruitées peuvent être traitées. Le résultat fourni par PDOBut est sous la forme d'un sous-ensemble de variables.

Les données volumineuses peuvent être traitées mais avec un coût calculatoire relativement important. En effet, la vitesse d'exécution de cet algorithme est non optimale à cause du comptage des paires nécessité par les critères de sélection utilisés. De plus, seules les variables qualitatives sont traitées.

2.2.2 ReliefF

ReliefF est un algorithme basé sur l'attribution de poids aux variables. L'algorithme commence par choisir un échantillon d'instances dont le nombre est fourni par l'utilisateur. Il recherche ensuite pour T instances (T choisi par l'utilisateur), la plus proche instance de réussite et la plus proche instance d'échec en se basant sur une mesure de distance. Et, l'algorithme met à jour les poids des différentes variables qui sont initialisés à zéro. Après avoir épuisé toutes les instances de l'échantillon, Relief choisit toutes les variables ayant un poids supérieur ou égal à un certain seuil.

Cette démarche est basée sur une idée intuitive qui est : une variable est plus pertinente qu'une autre si elle distingue une instance de son instance d'échec la plus proche, et moins pertinente si elle distingue

une instance de son instance de réussite la plus proche. ReliefF suit le même principe que Relief. La seule différence réside dans le fait que ReliefF permet de traiter les problèmes multi-classes.

D'après [40], la complexité algorithmique de ReliefF est en $O(Tnp)$, avec T le nombre d'instances spécifié par l'utilisateur, n le nombre d'individus et p le nombre de variables initiales. Les données volumineuses et/ou bruitées peuvent être traitées par ReliefF ainsi que les problèmes multiclasses. ReliefF peut travailler aussi bien avec des variables qualitatives qu'avec des variables quantitatives.

Le résultat de ReliefF est une liste de variables triées par ordre d'importance de leur pertinence. Le sous-ensemble optimal n'est pas déterminé. C'est à l'utilisateur de spécifier la taille du sous-ensemble optimal.

2.2.3 Méthode myope

La méthode myope que nous avons choisie consiste à mesurer l'entropie de Shannon correspondant à chaque variable. Les variables ayant la plus faible valeur pour l'entropie sont préférées. Sa complexité est $O(n \log n)$ avec n le nombre de variables initiales.

Ce type de méthodes peut traiter les problèmes contenant des données bruitées et/ou volumineuses. De plus, ce sont des méthodes très peu coûteuses en temps de calcul

Le problème majeur des méthodes myopes est lié au fait que l'interaction des variables exogènes n'est pas prise en compte. De plus, le résultat fourni par l'ensemble des méthodes myopes est la liste des variables initiales triées par ordre d'importance de leur pertinence vis à vis de la variable endogène. Donc, comme pour ReliefF, l'utilisateur doit préciser la taille du sous-ensemble optimal de variables. Dans notre cas, c'est à dire celui de l'utilisation de l'entropie de Shannon, les variables traitées doivent être discrètes. Cependant, rien n'empêche l'utilisateur d'utiliser un critère de sélection traitant les variables quantitatives.

2.2.4 MIFS

A chaque itération, l'algorithme choisit une seule variable : celle qui maximise la valeur de l'information mutuelle avec la variable endogène. Cet algorithme sélectionne les variables qui ont une information mutuelle maximale avec la variable endogène mais qui minimisent l'information mutuelle entre elles. MIFS effectue $\binom{n}{2}+n$ calculs d'information mutuelle.

Le résultat fourni par MIFS est un sous-ensemble optimal de variables. MIFS permet de traiter les données volumineuses et/ou bruitées. C'est une méthode très rapide.

MIFS ne travaille qu'avec des variables discrètes. Ceci réduit considérablement son champs d'action.

Le tableau 6 permet de résumer l'ensemble des méthodes que nous allons tester.

	ReliefF	**PDOBut**	**Méthode myope**	**MIFS**
Type de méthodes	Filtre heuristique	Filtre heuristique	Filtre myope	Filtre en un seul scan de base
Critère d'évaluation	Consistance	Comparaison par paires	Information	Information
Type de données traitées	Quantitatives et qualitatives	Qualitatives	Qualitatives	Qualitatives
Données volumineuses	Oui	Oui	Oui	Oui
Données bruitées	Oui	Oui	Oui	Oui
Résultat fourni	Liste de variables triées	Sous-ensemble optimal	Liste de variables triées	Sous-ensemble optimal
Complexité	$O(Tnp)$	$O(pc)$	$O(n\log n)$	$\binom{n}{2}+n$

Tableau 6 Caractéristiques des méthodes testées.

2.3 Expérimentations

Pour nos expérimentations, nous avons sélectionné un ensemble de dix bases sélectionnées parmi celles de l'UCI [70]. Les caractéristiques de ces bases sont présentées en annexe. Toutes les variables ont été discrétisées afin de pouvoir comparer les quatre méthodes entre elles. La discrétisation s'est effectuée par la méthode FUSINTER, [71].

L'algorithme d'apprentissage sélectionné est ID3. Une 10-Cross-Validation a été utilisée pour calculer le taux d'erreur moyen. Le découpage de l'ensemble des données a été effectué comme suit : la totalité des individus a été partagée aléatoirement en deux parties, tout en gardant la répartition initiale des classes. Le premier sous-ensemble d'individus contient 30% des individus et nous servira pour appliquer le processus de sélection de variables. Le tableau 7 et la figure 7 présentent les résultats.

Jeu de Données	Avant sélection			ReliefF			MIFS			Méthode myope			PDOBut		
	Erreur	σ	n	Erreur	σ	Nbr. Var.	Erreur	σ	Nbr. Var.	Erreur	σ	Nbr. Var.	Erreur	σ	Nbr. Var.
Austra	17,16	6,21	14	15,31	5,23	2	17,17	4,12	13	16,51	3,76	7	16,52	3,42	10
Breast	5,9	2,64	9	5,29	3,16	6	5,9	2,64	9	5,08	2,58	5	4,28	2,32	5
Cleve	27,1	9,18	13	40,54	7,77	6	24,68	10,27	8	22,38	6,96	6	25,69	9,53	5
CRX	14,46	5,44	15	17,54	5,88	2	16,12	6,7	7	16,31	3,79	7	16,94	5,37	8
German	29,57	6,13	20	30,14	6,01	14	27,43	5,06	3	30,71	3,57	10	25,29	4,38	9
Heart	31,05	6,42	13	27,38	9,06	2	28,42	9,76	13	32,11	10,6	6	27,37	10,5	7
Iono	10,92	4,37	34	11,78	3,94	25	15,75	8,71	8	14,57	5,16	17	16,15	11,3	3
Pima	25,24	5,76	8	25,05	7,69	7	24,87	4,83	4	25,78	4,14	4	26,11	5,43	8
Tic tac toe	28,44	7,53	9	30,51	5,9	5	30,81	7,11	3	29,33	4,98	4	26,78	2,29	7
Vehicle	30,59	5,54	18	42,25	6,52	18	40,62	7,39	6	32,44	5,5	9	33,94	4,82	8

Tableau 7 Evaluation des méthodes de sélection pour une 10-Cross-Validation avec ID3.

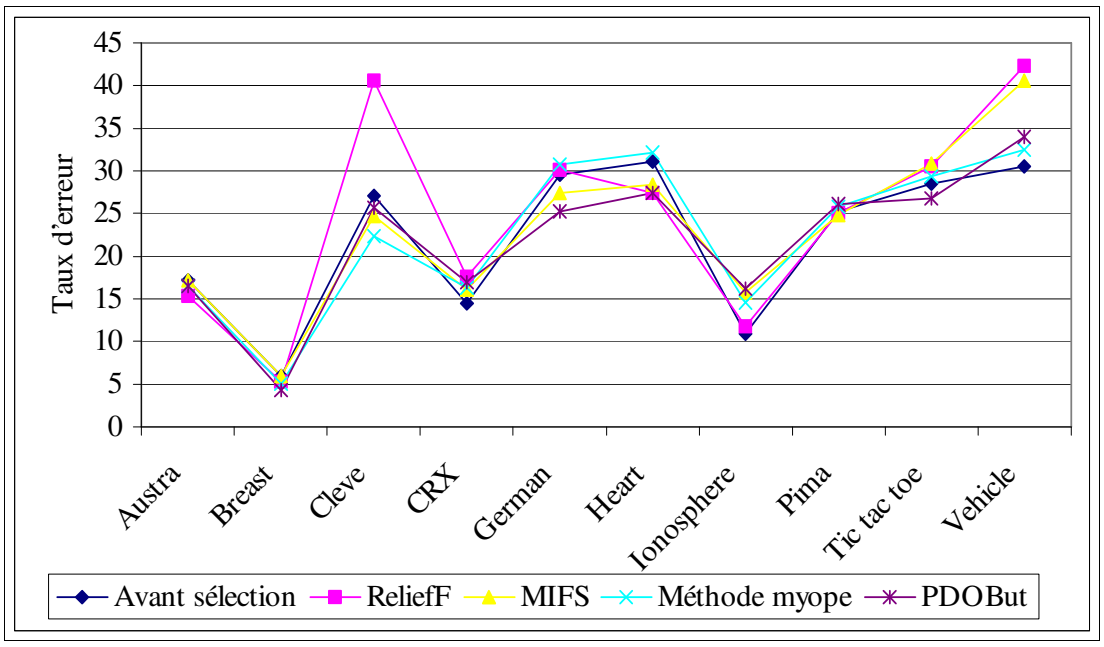


Figure 7 Taux d'erreur moyen des méthodes testées.

A partir de ces expérimentations, nous pouvons tirer un certain nombre de remarques.

- La détermination du nombre optimal de variables pose problème pour ReliefF et la méthode myope. Aucune indication n'est donnée quant à la taille du sous-ensemble optimal. Aussi, les résultats d'apprentissage peuvent énormément varier suivant le nombre de variables sélectionnées. Nous avons choisi, pour ReliefF, de ne sélectionner que les variables pour lesquelles le critère de consistance est supérieur ou égal à 0,1. En effet, c'est le seuil choisi de manière classique pour cette méthode. Pour la méthode myope, nous avons fonctionner par tâtonnement et nous avons sélectionné le sous-ensemble de variable minimisant le taux d'erreur moyen.
- Pour certains jeux de données, tels que Austra, pour lesquels les variables pertinentes sont connues, ces dernières ont été découvertes. Cependant elles ne sont pas toujours les seules à être sélectionnées.
- La phase de sélection a permis de réduire le nombre de variables et par conséquence, le temps d'apprentissage. Toutefois, il n'y a pas forcément d'amélioration du taux d'apprentissage (dans notre tableau 7, de diminution du taux d'erreur).
- Les temps de calcul avec ReliefF et PDOBut sont relativement important, de l'ordre de quelques minutes pour certaines bases. Pour MIFS et la méthode myope, les temps de calcul sont relativement courts.

	ReliefF	MIFS	Méthode myope	PDOBut
Austra	1	4	2	3
Breast	3	4	2	1
Cleve	4	2	1	3
CRX	4	1	2	3
Diabetes	4	2	1	3
German	3	2	4	1
Heart	2	3	4	1
Iono	1	3	2	4
Pima	2	1	3	4
Tic Tac Toe	3	4	2	1
Vehicle	4	3	1	2
Rang moyen	3	3	2	2

Tableau 8 Classement des méthode en fonction du taux d'erreur.

- Toutes les méthodes ne sont pas efficaces sur les mêmes jeux de données. Cependant, il n'est pas possible de préférer une méthode plus qu'une autre. Chaque méthode est efficace sur les jeux de données qui sont en relation avec ses caractéristiques et en particulier avec son critère d'évaluation. Le tableau 8 nous montre que toutes les méthodes apportent le meilleur taux d'erreur moyen pour au moins un jeu de données particulier. Si l'on considère le rang moyen de chaque méthode, la mesure myope et PDOBut sont les mieux classées en moyenne. Ce rang moyen est calculé pour l'ensemble des onze jeux de données. La méthode myope est, dans la majorité des cas, classée première ou deuxième.

2.4 Comparaisons des méthodes et profil-type de la méthode de sélection idéale

Du point de vue de l'amélioration de la qualité d'apprentissage après la sélection, aucune méthode ne surpasse les autres. L'étude du classement des méthodes en fonction du taux d'erreur moyen nous pousse à légèrement favoriser la méthode myope et PDOBut. Cependant, PDOBut et ReliefF sont les méthodes qui nécessitent le temps de calcul le plus important, la méthode la moins onéreuse en temps de calcul étant la méthode myope. Il nous simple donc naturel de ne pas utiliser ReliefF dont les temps de calcul explosent avec l'augmentation du nombre d'individus et de variables. Et, l'inconvénient de la méthode myope lié à la non prise en compte de l'interaction des variables exogènes n'a pas de conséquences graves sur les résultats de cette méthode. Elle peut même obtenir des résultats meilleur que les autres méthodes. Le problème principal de la méthode myope et de ReliefF reste le fait qu'elles ne fournissent pas de sous-ensemble optimal de variables. Ceci pose problème pour le choix du sous-ensemble destiné à l'apprentissage. Les modèles après sélection sont relativement stables quelle que soit la méthode de sélection utilisée. L'ensemble des méthodes de sélection permettent une nette réduction de l'espace de représentation des données. Les deux méthodes qui paraissent les plus efficientes, au vue de l'ensemble de ces remarques, sont MIFS et la méthode myope.

Nous pouvons ainsi dégager quelques propriétés que la méthode idéale doit posséder :

- Le résultat fourni doit être sous la forme d'un sous-ensemble de variables ou sous une forme s'y rapprochant,
- La méthode doit être rapide et peu coûteuse,
- Elle doit pouvoir traiter à la fois un grand nombre d'individus et de variables,
- Son critère d'évaluation doit tenir compte de la grande majorité des catégories de critères de sélection afin d'être compétent pour le plus grand nombre de problème,

- La méthode doit améliorer le plus souvent possible la qualité d'apprentissage, tout en réduisant la taille de l'espace de représentation des données.

	ReliefF	**MIFS**	**Méthode myope**	**PDOBut**
Résultat sous forme de sous-ensemble optimal	Non	Oui	Non	Oui
Méthode rapide et peu coûteuse	Non	Oui	Oui	Non
Efficace sur données volumineuses	Non	Oui	Oui	Non
Amélioration de la qualité d'apprentissage	Pas toujours	Pas toujours	Pas toujours	Pas toujours
Réduction de l'espace de représentation	Oui	Oui	Oui	Oui

Tableau 9 Caractérisations des méthodes testées.

Le tableau 9 nous indique que les méthodes ReliefF et PDOBut possèdent peu de qualités que devraient posséder la méthode de sélection idéale. Il nous semble utile de nous intéresser de plus près aux méthodes myopes qui possèdent déjà un certain nombre de qualités que devraient posséder la méthode de sélection idéale, à savoir la rapidité, le traitement de données volumineuses et un gain fréquent en qualité d'apprentissage après le processus de sélection.

3 Notre méthode de sélection

3.1 Point de départ

Les résultats de nos expérimentations sur les méthodes de sélection existantes nous permettent de conclure que les méthodes myopes sont plutôt efficientes : en effet, elles sont rapides, peu coûteuses et fournissent des résultats encourageants et équivalents à ceux des autres méthodes du point du taux d'erreur après sélection. Cependant, les méthodes myopes possèdent un certain nombre de défauts qu'il convient d'étudier plus en détail. Ce constat a été le point de départ de notre méthode.

3.1.1 Problèmes liés aux méthodes myopes

3.1.1.1 Interaction des variables exogènes

Le premier problème des méthodes myopes concerne le fait qu'elles traitent chaque variable indépendamment des autres variables. Chaque variable est considérée hors du contexte de l'espace de représentation. En effet, une variable peut être pertinente quand elle est prise individuellement et devenir non pertinente en présence d'autres variables. Toutefois, cet inconvénient n'a pas beaucoup d'impact sur la qualité des résultats sur les jeux de données traités précédemment et en comparaison aux autres méthodes de sélection de variables sur ces mêmes jeux de données.

3.1.1.2 La forme du résultat

La forme du résultat est un liste de variables triées par ordre de pertinence. Cette forme ne fournit aucun renseignement sur la taille du sous-ensemble optimal de variables. L'obtention d'une liste triée de variables limite l'intérêt de la sélection de variables : comment peut on déterminer la taille optimale du sous-ensemble ? L'utilisateur n'a qu'une seule solution pour découvrir cette taille : l'une des méthodes qui semble efficace pour obtenir un sous-ensemble optimal de variables est d'utiliser une approche enveloppe qui ajoute ou ôte itérativement les éléments de la liste triée au sous-ensemble de variables sélectionnées. A chaque itération, la méthode d'apprentissage est appliquée pour tester si l'ajout ou la suppression d'une variable entraîne une amélioration du taux d'apprentissage. Toutefois, ce processus est bien trop onéreux et long pour être appliqué. Nous désirons obtenir comme type de résultats :

- Le sous-ensemble de variables considéré comme optimal par la méthode de sélection. C'est la situation idéale.
- Néanmoins, il nous semble intéressant que l'utilisateur et/ou l'expert du domaine puissent intervenir lors du choix des variables.

Nous avons donc décidé, dans un premier temps, que la forme des résultats doit se situer entre une liste de variables et un sous-ensemble optimal. Une liste de variables laissent trop de liberté et trop peu d'indications sur le sous-ensemble optimal de variables à l'utilisateur. Et, l'obtention d'un sous-ensemble est une situation trop rigide.

3.1.1.3 Le choix du critère

De nombreux critères d'évaluation existent. Les critères mesurent différentes caractéristiques de variables. Il existe quatre catégories de critères : les critères d'information, les critères d'indépendance, les critères de distance et les critères de consistance.

La problématique à ce niveau est la suivante : Quel critère est le plus efficace et lequel doit on utiliser pour sélectionner les variables les plus pertinentes ? Il n'existe pas de critère meilleur ou plus efficace que les autres. Chaque critère met en avant certaines qualités spécifiques à chaque variable. Il semble intéressant d'obtenir un résultat tenant compte de l'avis de plusieurs critères différents, afin de ne pas éliminer ou sélectionner trop hâtivement les variables.

3.1.2 Solutions proposées

3.1.2.1 Un ensemble de critères

Afin d'avoir plusieurs avis différents sur les variables, la méthode que nous proposons utilise un ensemble de différents critères d'évaluation myopes. Cet ensemble de critères doit nous permettre de couvrir les différentes problèmes que peuvent engendrer les variables exogènes. Pour cette raison, il nous semble important de choisir un ou plusieurs critères de chaque catégorie de critères myopes. Si nous désirons traiter des variables quantitatives et/ou des variables qualitatives, nous choisirons des critères pouvant considérer le type de variables désirées. Nous fixons deux contraintes dans le choix des critères :

- La première contrainte imposée au choix de ces critères est qu'ils doivent être de type myope, afin de garantir la rapidité de leur calcul et ainsi la rapidité de la méthode de sélection.
- La deuxième contrainte est que parmi les critères choisis, il doit y en avoir au moins un de chaque catégorie de critère.

Ces deux contraintes laissent, cependant, de nombreuse possibilités à l'utilisateur. Il est ainsi possible de s'adapter au cas traité. A chaque problème auquel nous sommes confrontés, nous pouvons moduler notre méthode par le choix des critères et ce dans le but d'obtenir le meilleur résultat possible.

3.1.2.2 Une méthode d'agrégation

Pour prendre en compte l'avis de ces critères, notre méthode de sélection utilise une procédure d'agrégation. Ce type de procédure permet d'obtenir une vue d'ensemble tenant compte de tous les points de vue considérés. Les méthodes d'agrégation peuvent être vues suivant deux aspects :

- Soit suivant la nature des données [72] : dans ce cas, l'agrégation peut être :
  - Uni-représentationnelle, c'est à dire que l'ensemble des critères utilise la même représentation des données,
  - Multi-représentationnelle, c'est à dire que chaque critère utilise une représentation des données différentes. Ce peut être le cas si plusieurs jeux de données différents sont utilisés.
- Soit suivant la stratégie de combinaison mise en place : cette stratégie se définit à la fois :
  - Par rapport au nombre de critères différents utilisés. Deux situations peuvent être rencontrées, nous pouvons employer deux types d'approches différentes :

Soit *une approche mono-stratégique* : plusieurs occurrences d'un même critère sont appliquées au données. Les occurrences diffèrent par les conditions initiales…

Soit une *approche multi-stratégique* : plusieurs critères distincts sont combinés.

- Par rapport à la méthodologie employée. Deux procédures existent :

Soit une *procédure multi-étapes* : les critères travaillent alors en série, les uns à la suite des autres. Cela permet à chaque critère d'utiliser le résultat du critère précédent comme données d'entrée.

Soit une *procédure multi-experts* : les critères travaillent en parallèle. Chaque critère fournit une solution qui lui est propre. Cette procédure nécessite une unification de l'ensemble des résultats obtenus.

Il existe trois méthodes principales d'agrégation :

- **Le Bagging ou Bootstrap Agregating** : Cette méthode, proposée par [73], est une méthode de combinaison mono-stratégique et multi-expert basée sur le vote. Elle propose de générer un ensemble de classification à partir de plusieurs sous-ensembles de données. Ensuite, les différents résultats sont combinés ou agrégés par un algorithme de vote, en l'occurrence un vote simple à la majorité. En d'autres termes, elle "valide" la classification en générant différents échantillons à partir de l'échantillon d'apprentissage. Ces échantillons sont utilisés pour construire des critères.

Un critère final est ensuite construit à partir de tous ces critères. La classe prédite par ce critère final est celle qui apparaît le plus dans les critères intermédiaires.

- **Le Boosting** : cette méthode, introduite par [74], est mono-stratégique et multi-étape. Cette technique améliore la précision d'un critère en mettant à jour un ensemble de poids sur les différents individus de l'ensemble d'apprentissage. Le poids des individus mal classés sont augmentés de manière à ce que l'attention se porte sur ces individus mal classés. Donc, comme le Bagging, le Boosting génère un ensemble de critères. Mais Boosting les génère séquentiellement alors que le Bagging peut le fait en parallèle. Le but est de minimiser l'erreur attendue selon différentes distributions en entrée. Il faut spécifier un nombre d'essais, c'est-à-dire un nombre de critères à générer. Le poids de chaque critère pour la construction du critère final dépend de ses performances : le poids d'un critère est, ainsi inversement proportionnel à l'erreur commise.
- **Le Arcing ou Adaptatively Resample and Combine, [75] et [76]** : le terme Arcing a été introduit par Breiman dans [15]. Plusieurs critères sont calculés en série et un poids est affecté à chaque individu à classer. L'affectation du poids à chaque individu s'effectue de manière très simple : le poids des critères est proportionnel au nombre d'erreurs des critères précédents à la puissance 4, plus 1. Pour agréger les différents résultats obtenus, une méthode de vote simple est utilisée.

Le Bagging, le Boosting et le Arcing agrège un ensemble de critères. Dans notre cas, les critères sont l'ensemble des critères de sélection myopes que nous choisissons.

Notre méthode de sélection de variables utilise une méthodologie d'agrégation proche du Bagging. En effet, l'agrégation des critères de sélection est de type uni-représentationnelle, multi-stratégique et multi-experts. Notre méthode utilise une procédure d'agrégation uni-représentationnelle car chaque critère travaille sur une même représentation du jeu de données considéré. En effet, notre méthode travaille sur un même ensemble de données traitées par l'ensemble des critères de sélection. La procédure d'agrégation utilisée est multi-stratégique car un ensemble de critères différents sont appliqués sur un jeu de données particulier. La procédure employée est multi-experts car les critères sont lancés en parallèle et leurs résultats devront être agrégés et ne pourront pas utiliser comme données le résultat du critère précédent.

3.1.2.3 Agrégations des résultats des critères

Notre méthode étant multi-experts, il est nécessaire d'agréger les résultats obtenus par l'ensemble des critères de sélection. Afin d'unifier ces différents résultats, nous nous servons d'une méthode d'agrégation des préférences. Ceci nous permettra de tenir compte de l'avis de l'ensemble des critères de sélection choisis. L'emploi d'une méthode d'agrégation, [77], nécessite la définition d'un ensemble de juges et d'un ensemble d'individus. Ce sont les classements des individus par chaque juge qui sont agrégés.

Dans notre cas, l'ensemble des individus est l'ensemble des variables exogènes et les juges sont les différents critères de sélection myopes que nous employons. Nous utilisons la méthode d'agrégation d'opinion développée par [78, 79] et basée sur [22, 80]. Nous allons présenter le principe sous-jacent à cette méthode.

Pour tout couple d'individus, ici de variables, $(X_i, X_j) \in X \times X$, chaque juge, ici chaque critère, émet un avis $A_k(i, j)$. Cet avis est de type préférence large. En conséquence, l'opinion A_k du juge k est une application de $X \times X$ dans $PL = \{PREF, NPREF, EQ\}$.

Les différents éléments de PL se définissent de la manière suivante :

- $A_k(i, j) = PREF \Leftrightarrow$ le juge ou critère k préfère X_i à X_j ;
- $A_k(i, j) = NPREF \Leftrightarrow$ le juge ou critère k préfère X_j à X_i ;
- $A_k(i, j) = EQ \Leftrightarrow$ le juge ou critère k considère X_j et X comme ressemblants.

Chaque juge est considéré comme cohérent, ce qui signifie que :

- $A_k(i, j) = PREF \Leftrightarrow A_k(j, i) = NPREF$,
- et $A_k(i, j) = EQ \Leftrightarrow A_k(j, i) = EQ$.

On ne suppose aucune propriété de transitivité pour A_k. A_k est une opinion de type large et de ce fait, se compose de $p(p-1)/2$ avis distincts, avec p le nombre de variables.

Le problème peut se résumer de la manière suivante : Etant données les opinions des t juges $A_1, ..., A_k, ..., A_t$, nous cherchons à construire un opinion OP, c'est à dire l'opinion d'un méta-juge

virtuel qui en soit la meilleure agrégation possible. L'opinion cherchée OP doit engendre une relation de préordre total sur X. Nous sommes en face d'un problème de classement optimal où les ex-œquo sont admis. OP est une application de $X \times X$ dans $PL = \{PREF, NPREF, EQ\}$.

Il convient, en premier lieu, de définir un certain nombre de notions.

Définition 1 : le degré d'accord $\rho_{i,j}(OP, A_k)$ entre les avis $OP(i, j)$ et $A_k(i, j)$ est défini comme précisé dans le tableau 10.

OP / A_k	$PREF$	$NPREF$	EQ
$PREF$	1	0	0,5
$NPREF$	0	1	0,5
$NPREF$	0,5	0,5	1

Tableau 10 Degrés d'accord $\rho_{i,j}(OP, A_k)$.

Définition 2 : le degrés d'accord entre les opinions OP et A_k est

$$DA(OP, A_k) = \sum_{(X_i, X_j) \in X \times X} \rho_{ij}(OP, A_k)$$

.

Définition 3 : le degré d'accord entre l'opinion OP et l'opinion de tous les juges est

$$DA(OP) = \sum_{k=1}^{t} DA(OP, A_k)$$

.

Notre problème devient ainsi un problème de maximisation : nous désirons obtenir l'opinion OP qui maximise $DA(OP)$.

A une opinion OP de type préférence large, nous associons trois vecteurs $(op_{ij})_{(X_i, X_j) \in X \times X}$, $(\overline{op}_{ij})_{(X_i, X_j) \in X \times X}$, $(o\tilde{p}_{ij})_{(X_i, X_j) \in X \times X}$ définis comme suit :

- $op_{ij} = \begin{cases} 1 \text{ si } OP(i, j) = PREF \\ 0 \text{ sinon} \end{cases}$;

- $\overline{\overline{op}}_{ij} = \begin{cases} 1 \; si \; OP(i,j) = NPREF \\ 0 \text{ sinon} \end{cases}$;
- $o\tilde{\tilde{p}}_{ij} = \begin{cases} 1 \; si \; OP(i,j) = EQ \\ 0 \text{ sinon} \end{cases}$.

Notons que OP devant engendrer un préordre total sur X, on obtient la relation suivante : $op_{ij} + \overline{op}_{ij} + o\tilde{\tilde{p}}_{ij} = 1, \quad \forall (X_i, X_j) \in X \times X$.

Nous associons de la même manière à chaque opinion A_k les vecteurs :

- $(a_{kij})_{(X_i,X_j)\in X\times X}$ définis de la même manière que $(op_{ij})_{(X_i,X_j)\in X\times X}$;
- $(\bar{a}_{kij})_{(X_i,X_j)\in X\times X}$ définis de la même manière que $(\overline{op}_{ij})_{(X_i,X_j)\in X\times X}$;
- et $(\tilde{a}_{ikj})_{(X_i,X_j)\in X\times X}$ définis de la même manière que $(o\tilde{\tilde{p}}_{ij})_{(X_i,X_j)\in X\times X}$.

Le degré d'accord entre les avis $OP(i,j)$ et $A_k(i,j)$ peut alors s'écrire de la manière suivante :

$$\rho_{i,j}(OP, A_k) = op_{ij}\left(a_{kij} + \frac{1}{2}\tilde{a}_{kij}\right) + \overline{op}_{ij}\left(\bar{a}_{kij} + \frac{1}{2}\tilde{a}_{kij}\right) + o\tilde{\tilde{p}}_{ij}\left(\frac{1}{2}a_{kij} + \frac{1}{2}\bar{a}_{kij} + \tilde{a}_{kij}\right).$$

En tenant compte de $op_{ij} + \overline{op}_{ij} + o\tilde{\tilde{p}}_{ij} = 1, \quad \forall (X_i, X_j) \in X \times X$ et en éliminant $o\tilde{\tilde{p}}_{ij}$, il vient :

$$\rho_{i,j}(OP, A_k) = op_{ij}\left(\frac{1}{2}a_{kij} - \frac{1}{2}\tilde{a}_{kij} - \frac{1}{2}\bar{a}_{kij}\right) + \overline{op}_{ij}\left(\frac{1}{2}\bar{a}_{kij} - \frac{1}{2}\tilde{a}_{kij} - \frac{1}{2}a_{kij}\right) + \left(\frac{1}{2}a_{kij} + \frac{1}{2}\bar{a}_{kij} + \tilde{a}_{kij}\right)$$

Pour $(X_i, X_j) \in X \times X$, on pose :

- $r_{ij} = \sum_{k=1}^{t} a_{kij}$, le nombre de juges qui préfèrent X_i à X_j ;

- $\bar{r}_{ij} = \sum_{k=1}^{t} \bar{a}_{kij}$, le nombre de juges qui préfèrent X_j à X_i ;
- $\tilde{r}_{ij} = \sum_{k=1}^{t} \tilde{a}_{kij}$, le nombre de juges qui ne manifestent pas de préférence entre X_i et X_j ;
- $s_{ij} = r_{ij} - \bar{r}_{ij} - \tilde{r}_{ij}$ et $\bar{s}_{ij} = \bar{r}_{ij} - r_{ij} - \tilde{r}_{ij}$, les mesures de préférences algébriques.

Il s'en suit : $DA(OP) = \sum_{k=1}^{t} \sum_{(X_i, X_j) \in X \times X} \rho_{ij}(OP, A_k)$

$$= \sum_{(X_i, X_j) \in X \times X} \sum_{k=1}^{t} \rho_{ij}(OP, A_k)$$

$$= \sum_{(X_i, X_j) \in X \times X} \left[\frac{1}{2}\left(r_{ij} + \bar{r}_{ij}\right) + \tilde{r}_{ij} \right] + \frac{1}{2} \sum_{(X_i, X_j) \in X \times X} \left[op_{ij} s_{ij} + \overline{op}_{ij} \bar{s}_{ij} \right]$$

Nous pouvons remarquer que $\sum_{(X_i, X_j) \in X \times X} \left(r_{ij} + \bar{r}_{ij} + \tilde{r}_{ij}\right)$ est égal au nombre d'avis exprimé c'est à dire à $t \frac{p(p-1)}{2}$. Il s'en suit : $DA(OP) = \frac{1}{2}\left(t \frac{p(p-1)}{2} + \sum_{(X_i, X_j) \in X \times X} \left(\tilde{r}_{ij}\right) \right) + \frac{1}{2} \sum_{(X_i, X_j) \in X \times X} \left[op_{ij} s_{ij} + \overline{op}_{ij} \bar{s}_{ij} \right]$.

Un préordre L peut être décrit comme un ensemble de groupes ordonnés, $L = \{l_1, ..., l_m, ..., l_M\}$ conformément à la figure 8 où $\mapsto$ signifie que tout élément de l_1 est préféré à tout élément de l_2 …

$$(l_1) \mapsto (l_2) \mapsto ... \mapsto (l_M)$$

Figure 8 Description d'un préordre.

L'opinion OP et le préordre L correspondant se caractérisent l'un et l'autre selon les relations suivantes : $\begin{cases} X_i, X_j \in l_m \Leftrightarrow OP_{i,j} = EQ \\ X_i \in l_m, X_j \in l_{m'}, m < m' \Leftrightarrow OP_{i,j} = PREF \end{cases}$

Nous posons : $Val(L) = \sum_{m=1}^{M-1} \sum_{m'=m+1}^{M} \sum_{\substack{X_i \in l_m \\ X_j \in l_{m'}}} s_{ij}$.

Il s'en suit : $DA(OP) = \frac{1}{2}\left(t\frac{p(p-1)}{2} + \sum_{(X_i,X_j)\in X\times X} \tilde{r}_{ij} + Val(L) \right)$.

La recherche d'un opinion OP qui maximise $DA(OP)$ est donc équivalente au problème suivant :

$$\max\left[Val(L) \middle| L \text{ préordre sur } X \right].$$

Pour la maximisation, la méthode du recuit simulé [81] est utilisée. Ce problème de programmation linéaire NP- difficile peut être résolu par différentes méta-heuristiques, la méthode du recuit-simulé a été sélectionnée en raison de sa mise en œuvre relativement facile et de sa rapidité d'exécution de l'ordre de quelques secondes. Pour la mise en œuvre de la méthode du recuit simulé, on sait qu'il est important de disposer d'un concept de voisinage. Aussi, nous dirons qu'un préordre L' est voisin d'un préordre L, $L' \in V(L)$, si et seulement si L' dérive de L par le déplacement d'un seul individu $X_i \in l_m$ de la manière suivante :

- Soit X_i est transféré dans l_{m+1} (sous réserve que $m < M$) ou dans l_{m-1} (sous réserve que $m > 1$) ;
- Soit X_i constitue un nouveau groupe à lui tout seul, et X_i peut être préféré ou non à l_m.

La variation $\Delta = Val(L') - Val(L)$ est facile à calculer. Ainsi si l'on suppose que L' dérive de L par le déplacement de $X_i \in l_m$ dans le groupe l_{m+1}, il vient : $\Delta = \sum_{\substack{X_j \in l_m \\ j \neq i}} s_{ij} - \sum_{X_j \in l_{m+1}} s_{ij}$.

La procédure d'optimisation par le recuit simulé peut maintenant être définie dans l'algorithme 13. Nous avons choisi la valeur des paramètres initiaux conformément aux recommandations de [81] : $I_{Max} = 10 * card(X)$ et $ro = 0{,}98$. La valeur de T_0 est telle que $\exp(\Delta / T_0)$ soit proche de 1 en moyenne.

L'application de cette méthode d'agrégation des préférences sur l'ensemble des critères myopes choisis nous permet d'obtenir un classement de type préordre des variables exogènes qui va tenir compte des différents classement inférés par chaque critère.

Algorithme RS
Données : T_0 la température initiale,
I_{Max} le nombre d'itérations dans la boucle interne
ro le coefficient de refroidissement
Résultat : L^*
Début
Initialisation
$T = T_0$
Choisir au hasard un préordre L de X
$L^* = L$
Répéter
$change = FAUX$
Pour $i = 1$ à I_{Max} faire
Choisir au hasard L' dans $V(L)$
Calculer Δ
Tirer au hasard une valeur R dans $[0,1]$
Si $\Delta > 0$ ou $R < \exp(\Delta / T)$ alors
$L^* = L$
$change = VRAI$
Si $Val(L) > Val(L^*)$ alors
$L^* = L$
Fin Si
Fin Si
Fin Pour
$T = T * ro$
jusqu'à $change = FAUX$
Fin

Algorithme 13 Algorithme du Recuit Simulé.

3.1.2.4 La forme du résultat

Grâce à l'utilisation, lors de la procédure d'agrégation, d'une relation de préférence large qui autorise les ex-æquo, le résultat fourni ne sera plus un ordre sur les variables mais un préordre total sur les variables exogènes qui tient compte de l'avis de l'ensemble des critères myopes choisis. Ainsi, le résultat de notre méthode de sélection est désormais une liste de sous-ensembles de variables triés en fonction de leur pertinence. Nous avons donc deux solutions :

- Soit nous fournissons à l'utilisateur cette liste de sous-ensembles de variables. Cette solution peut être intéressante si l'utilisateur est un expert du domaine, car elle lui laissera une certaine marge d'action dans le choix du sous-ensemble de variables.

- Soit nous mettons en place une procédure, que nous présentons par la suite, qui nous permet de fournir à l'utilisateur le sous-ensemble de variables considéré comme optimal par notre méthode de sélection.

3.2 Présentation de notre méthode

La méthode de sélection que nous proposons est une méthode hybride à l'intersection des approches filtre et enveloppe de type forward selection. Tous les types de variables sont traitées, en fonction des critères choisis par l'utilisateur. Notre méthode agrège les classements de variables obtenus à l'aide de plusieurs critères de sélection myopes. Le résultat produit une liste triée de sous-ensembles disjoints de variables. Nous fournissons, cependant, à l'utilisateur le sous-ensemble de variables considéré comme optimal par notre méthode.

Nous pouvons décomposer notre méthode en trois étapes, figure 9 :

- Le calcul et la discrétisation des différents critères pour chaque variable,
- L'agrégation des résultats obtenus à l'étape précédente,
- Et, la recherche du sous-ensemble optimal de variables.

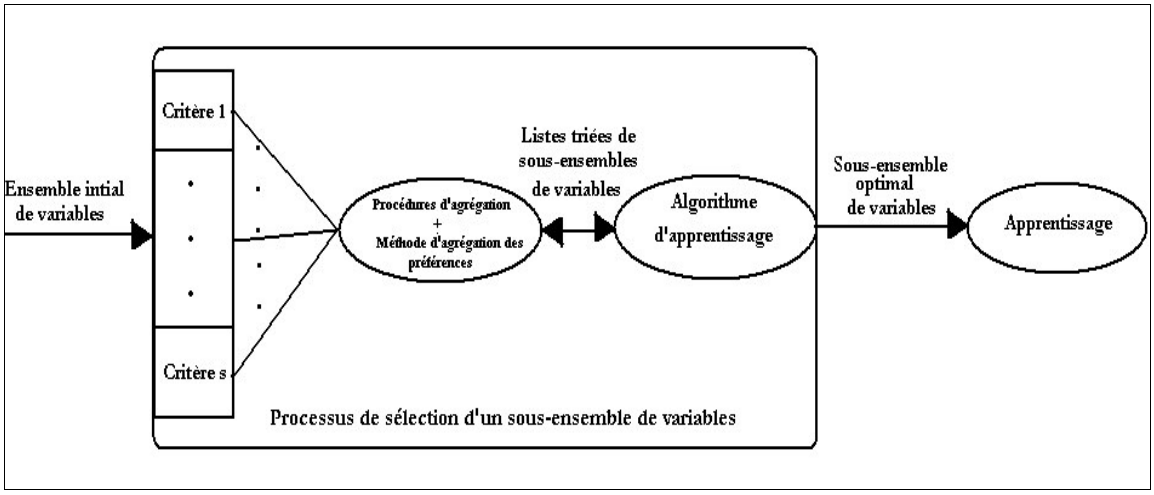


Figure 9 Déroulement de notre méthode de sélection de variables.

3.2.1 Calcul et discrétisation des critères myopes

L'ensemble des critères de sélection myopes dont nous nous servons est noté $CR = \{cr_1,...,cr_s,...,cr_S\}$ avec $S \in \aleph^+$ et $1 < s < S$.

3.2.1.1 Formalisme et définitions

Pour présenter cette première étape, nous avons besoin de définir un certain nombre de notions. Pour cela, nous considérons un problème d'apprentissage caractérisé par un ensemble d'individus $\Omega = \{\omega_1,...,\omega_i,...,\omega_n\}$ décrits par un ensemble de variables $X = \{X_1,...,X_k,...,X_p\}$ nommé ensemble initial de variables.

A chaque critère cr_s appartenant à CR est associé l'ensemble $CR_s = \{cr_{s1},...,cr_{sk},...,cr_{sp}\}$ qui représente l'ensemble des valeurs prises par le critère cr_s pour l'ensemble des variables.

Définition 1 : Le rang R_{sk} associé à la variable $X_k \in X$ pour le critère $cr_S \in CR$ représente la place de la variable $X \in X$ dans le classement obtenu à l'aide du critère $cr_S \in CR$. Ainsi, la variable la plus pertinente sera celle qui possèdera le rang le plus faible.

Deux variables pouvant être aussi pertinentes l'une que l'autre vis-à-vis de la variable endogène, même si elles n'apportent pas le même type d'information, nous introduisons la notion d'équivalence de variables.

Définition 2 : Deux variables X_i et X_j sont équivalentes du point de vue d'un critère particulier $cr_S \in CR$ si et seulement si pour ce critère, elles ont le même rang : $X_i \leftrightarrow X_j \Leftrightarrow R_{si} = R_{sj}$.

3.2.1.2 Déroulement de la première étape

Cette étape se déroule en trois actions :

- **Calcul des critères** : Les calculs de chaque critère pour la totalité des variables s'effectuent en parallèle. Nous obtenons subséquemment un ensemble constitué de S listes de valeurs de critères. Chaque liste est ordonnée dans l'ordre décroissant de pertinence des variables. Il convient de faire attention au fait que certain critères doivent être maximisés, et d'autres doivent être minimisés.

Les valeurs des premiers seront donc classées par ordre décroissant et les valeurs des deuxièmes seront classées par ordre croissant.

- **Normalisation** : Une fois ordonnées, les valeurs cr_{sk} de chaque critère sont normalisées à l'aide de la transformation suivante : pour une variable $X_k \in X$ et un critère $cr \in CR$, la valeur normalisée du critère est $cr_{sk,N} = \frac{cr_{sk} - Min(CR_s)}{Max(CR_s) - Min(CR_s)}$. Cette transformation nous permet d'obtenir une valeur comprise entre 0 et 1.

- **Discrétisation** : Après leur normalisation, ces valeurs sont discrétisées en déciles. Cette discrétisation permet d'affecter, pour chaque critère, un rang à chaque variable de la manière suivante :

  - Pour les critères qui doivent être minimisés :

Si $cr_{sk,N} \in [0;(1/S)[$ alors $R_{sk} = 1$;
Si $cr_{sk,N} \in [(1/S);2\cdot(1/S)[$ alors $R_{sk} = 2$;
…
Si $cr_{sk,N} \in [9\cdot(1/S);1[$ alors $R_{sk} = S$.

  - Pour les critères qui doivent être maximisés :

Si $cr_{sk,N} \in [0;(1/S)[$ alors $R_{sk} = S$;
Si $cr_{sk,N} \in [(1/S);2\cdot(1/S)[$ alors $R_{sk} = 9$;
…
Si $cr_{sk,N} \in [9\cdot(1/S);1[$ alors $R_{sk} = 1$.

La méthode de normalisation ainsi que la méthode de discrétisation que nous avons choisies d'utiliser sur les critères ne sont pas forcément les plus adaptées ni les plus efficaces. Tout dépend de la structure des valeurs des critères obtenues. Lors de nos expérimentations, nous avons testé plusieurs combinaisons différentes de méthodes de normalisation et de discrétisation. La combinaison que nous proposons ici est celle qui nous a donnée en moyenne les résultats les plus généraux et les plus intéressants pour l'ensemble des jeux de données testés.

Chaque critère nous propose maintenant un classement des variables. Il est intéressant de préciser que dès cette étape, nous pouvons être en présence de variables équivalents pour un critère donné et qui possèdent donc le même rang pour ce critère.

3.2.2 Agrégation des divers résultats

Afin d'obtenir un classement des variables tenant compte à la fois de l'ensemble total des variables, nous agrégeons les classements proposés par tous les critères à l'aide de la méthode d'agrégation des préférences présentée précédemment.

Après l'application de cette technique d'agrégation, nous obtenons une liste de sous-ensembles disjoints de variables ordonnés en fonction de leur pertinence. Cette liste est notée $L = \{l_1,...,l_m,...,l_M\}$ avec $l_i \in X^g, 0 < g \leq p$. X^g est l'ensemble des sous-ensembles de variables générés par la méthode d'agrégation des préférences. Il ne peut pas y avoir plus de p sous-ensembles de variables générés (au minimum, chaque sous-ensemble contient une variable). Un certain nombre de propriétés caractérisent ces sous-ensembles de variables :

- Les sous-ensembles constituant L sont disjoints : $l_i \cap l_j = \{\}, \forall i, j \in [1, M]$;
- La somme des cardinaux des l_m est égal au nombre total de variables : $|l_1| + ... + |l_m| + ... |l_M| = p$.
- La pertinence du sous-ensemble $l_1$est plus importante que la pertinence du sous ensemble l_2 et ainsi de suite : $Pertinence(l_1) > ... > Pertinence(l_m) > ... > Pertinence(l_M)$. La notion de pertinence a été défini au sein du chapitre 1.
- La pertinence est la même pour toutes les variables appartenant à un même sous-ensemble.

Notre méthode peut s'arrêter là. Son résultat est alors une liste triée de sous-ensembles de variables. A l'utilisateur de décider le nombre de sous-ensembles de variables et par la même occasion de variables qui vont constituer le sous-ensemble «optimal». Cependant, nous préférons fournir une procédure permettant le choix automatique des sous-ensembles composant le sous-ensemble optimal de variables.

3.2.3 Découverte du sous-ensemble optimal

Jusqu'à présent, notre méthode de sélection se situe dans une optique d'approche filtre : en effet, nous avons obtenu un premier résultat, à savoir une liste de sous-ensemble de variables, sans l'intervention d'aucun algorithme d'apprentissage. Nous allons maintenant nous placer dans une optique enveloppe. D'autres méthodes telle que [82] sont également à l'intersection des deux types d'approches.

L'avantage d'utiliser une approche enveloppe est lié au fait que l'influence du sous-ensemble de variables sélectionnée sur les performances de l'algorithme d'apprentissage est prise en compte.

Entrée X l'ensemble initial de variables
Ω l'ensemble des individus
Sortie X^* le sous-ensemble optimal de variables

Début
$X^* = \{\}$
Pour $m = 1$ à M faire
$X^* = X^* \cup l_m$
Lancement de l'algorithme d'apprentissage
Si $(Erreur_m > Erreur_{m-1})$ ou $Erreur_m = Erreur_{m-1} = Erreur_{m-2}$ Alors
Arrêt
$X^* = X^* - l_m$
Fin Si
Fin Pour
Fin

Algorithme 14 Processus de découverte du sous-ensemble optimal.

Le processus de détermination du sous-ensemble optimal s'effectue de la manière suivante :

- A la m^{ième} itération, le sous-ensemble de variables $l_m \in L$ est ajouté au sous-ensemble «optimal» de variables et l'algorithme d'apprentissage est appliqué sur le nouveau sous-ensemble «optimal»;
- Le critère d'arrêt est double : il y a arrêt du processus soit lorsque le taux d'erreur est constant sur deux itérations, soit lorsque l'on assiste à une augmentation du taux d'erreur.

L'intérêt de se servir d'une approche enveloppe, à ce stade du processus de sélection, est renforcé par le fait que la contrainte du temps de calcul est diminuée : les variables ne sont pas ajoutées une à une mais sous-ensemble par sous-ensemble. Ainsi nous pouvons tenir des caractéristiques et des besoins de l'algorithme d'apprentissage utilisé tout en garantissant un temps de calcul limité.

3.2.4 Spécification de notre méthode

Afin de tester notre méthode de sélection de variables, nous avons choisi un ensemble de dix critères qui sont les suivants :

- Trois critères d'informations : L'entropie de Shannon, [11], le gain d'information, le ratio de gain, [83], le gain normalisé, [13] ; Ces critères nous permettent de quantifier l'information apportée par chaque variable exogène au sujet de la variable endogène.
- Deux critères de distance : La distance de Mantaras, [14], le critère de Gini, [15] ; Ces critères mesurent le pouvoir discriminant de chaque variable exogène vis-à-vis de la variable endogène.
- Trois critères d'indépendance : Le Khi2, le Tschuprow, [18], le coefficient de Cramer ; Ces critères mesurent la corrélation existante entre variable exogène et variable endogène.
- Un critère de consistance : le tau de Zhou, [20] ; Ce critère sélectionne les variables satisfaisant le MinFeatures Bias et détecte les variables redondantes.

Ces critères ont été choisis car ils sont relativement connus. Nous avons sélectionnés plusieurs pour chaque catégorie de critère myope[1]. Tous ces critères ne traitent que des variables qualitatives. Aussi, pour les jeux de données testés qui comportent des variables quantitatives, ces dernières sont discrétisées à l'aide de la méthode de discrétisation supervisée FUSINTER, [71]. Cette discrétisation des variables quantitatives nous permet de traiter toutes les variables de la même manière et d'avoir la possibilité de comparer entre elles différentes méthodes de sélection de variables.

3.2.5 Exemple

Afin de mieux expliciter notre méthode, nous en présentons un exemple de déroulement. Cet exemple se compose de quatre variables exogènes qualitatives $X = \{X_1, X_2, X_3, X_4\}$, d'une variable endogène possédant trois classes $Y = \{A, B, C\}$ et d'un ensemble de dix individus $\Omega = \{\omega_1, ..., \omega_{10}\}$, tableau 11.

Les variables X_3 et X_4 permettent à elles seules de discriminer au mieux la variable endogène :

- Si $X_3 = V$ alors $Y = C$;

[1] Les critères qui nécessitent une minimisation sont : l'entropie de Shannon, la distance de Mantaras, le Tau de Zhou et le critère de Gini, et les critères qui nécessitent une maximisation sont : le gain d'information, le gain normalisé, le ratio de gain, le Khi2, le critère de Tschuprowet le coefficient de Cramer.

- Si $X_4 = V$ alors $Y = A$;
- Si $X_3 = F$ et $X_4 = F$ alors $Y = B$.

Nous aimerions donc que notre méthode ne sélectionne que les variables X_3 et X_4.

X_1	X_2	X_3	X_4	Y
V	V	F	V	A
V	F	F	V	A
F	V	F	V	A
F	F	F	V	A
V	V	F	F	B
V	F	F	F	B
V	V	F	F	B
V	F	F	F	B
F	V	V	F	C
V	F	V	F	C

Tableau 11 Données exemple.

La première étape de notre méthode consiste en le calcul (tableau 12), la normalisation et la discrétisation des valeurs de chaque critère pour l'ensemble des variables. Le résultat que nous obtenons est un ensemble de dix listes triées de variables. On peut remarquer sur le tableau 13 que les variables 4 est toujours classées 1ère sauf pour le Tau de Zhou. En effet, ce critère pénalise la redondance et même si des différences flagrantes à l'oeil nu existent entre les variables 1, 2 et 4, ce critère les considère comme les plus proches parmi l'ensemble total de variables. La variable 3 est relativement bien classée par l'ensemble des critères (Rang de valeur 1 ou 3). La variable 2 est quant à elle toujours classée dernière. La variable 1 est toujours classée avant dernière.

Maintenant, nous appliquons la procédure d'agrégation des préférences aux données du tableau 13. Nous obtenons la liste suivante : $L = \{\{X_3, X_4\}, \{X_1\}, \{X_2\}\}$.

La dernière étape peut débuter. Nous travaillons avec l'algorithme d'apprentissage ID3, [83]. Nous allons donc commencer avec le sous-ensemble $l_1 = \{X_3, X_4\}$. Puis nous rajouterons itérativement les

sous-ensembles $l_2 = \{X_1\}$et $l_3 = \{X_2\}$. Le tableau 14 nous permet de résumer les taux d'erreur obtenus successivement.

	Entropie de Shannon	Gain d'information	Ratio de Gain	Gain Normalisé	Distance de Mantaras	Critère de Gini	Khi2	Tshuprow	Coefficient de Cramer	Tau de Zhou
X_1	-1,1219	0,4000	0,4000	0,4000	-0,8115	-0,3000	4	0,2828	0,6325	0,5175
X_2	-1,5219	9,54E-08	9,54E-08	9,54E-08	-9,9999E-01	-0,5000	0	0	0	0,5351
X_3	-0,8000	0,7219	1	0,7219	-0,5256	0	10	0,7071	1	0,7708
X_4	-0,5510	0,9710	1	0,9710	-0,3620	0	10	0,7071	1	0,5179
	à min	à max	à max	à max	à min	à min	à max	à max	à max	à min

Tableau 12 Valeur des critères avant leur normalisation.

	Entropie de Shannon	Gain d'information	Ratio de Gain	Gain Normalisé	Distance de Mantaras	Critère de Gini	Khi2	Tshuprow	Coefficient de Kramer	Tau de Zhou
X_1	6	6	7	6	8	7	7	6	4	10
X_2	10	10	10	10	10	10	10	10	10	10
X_3	3	3	1	3	3	1	1	1	1	1
X_4	1	1	1	1	1	1	1	1	1	10

Tableau 13 Classement des variables.

$l_1 = \{X_3, X_4\}$		$l_1 = \{X_3, X_4\}$ et $l_2 = \{X_1\}$		$l_1 = \{X_3, X_4\}$ et $l_2 = \{X_1\}$et $l_3 = \{X_2\}$	
Erreur	Ecart - type	Erreur	Ecart - type	Erreur	Ecart - type
20	40	30	45,83	30	45,83
Passage à l'itération suivante		Arrêt du processus			

Tableau 14 Processus de détermination du sous-ensemble optimal.

Le processus de détermination du sous-ensemble optimal s'arrête à la deuxième itération et sont sélectionnées uniquement les variables X_3 et X_4, $\left(X^* = \{X_3, X_4\}\right)$. Notre méthode a effectivement permis de sélectionner seulement les variables pertinentes de cet exemple succinct.

3.3 Conclusions

Le tableau 15 résume les caractéristiques principales de notre méthode.

Notre méthode	
Type d'approches	Filtre puis enveloppe
Type de méthodes	Myope
Critères d'évaluation	Information, Distance, Indépendance et Consistance
Direction de recherche	Forward selection
Algorithme d'apprentissage	Tous les algorithmes d'apprentissage
Critère d'arrêt	Pas d'amélioration
Forme du résultat	Sous-ensemble optimal
Type de variables traitées	Qualitative et quantitative
Données volumineuses	Oui

Tableau 15 Caractéristiques de notre méthode de sélection.

Notre méthode est essentiellement caractérisée par une approche de type filtre myope lors de la phase de classement des variables initiales en fonction de leur pertinence. Seule la partie de détermination du sous-ensemble optimal est de type enveloppe.

Elle utilise une méthode d'agrégation des préférences pour agréger les classements de variables obtenus à l'aide d'un ensemble de critères myopes. L'utilisation de différents critères myopes nous permet d'obtenir un ensemble de listes composées des variables initiales triées en fonction de leur pertinence et ce à moindre coût grâce à l'utilisation même des critères myopes. La méthode d'agrégation des préférences permet de combiner les différentes listes de variables et ainsi d'obtenir une nouvelle et unique liste triée de variables. Ce nouveau classement des variables initiales tient compte des caractéristiques de l'ensemble des critères myopes choisis. Le résultat de l'agrégation des préférences est sous la forme d'un préordre total sur les variables c'est à dire d'une liste de sous-ensembles de variables triés en fonction de leur pertinence. Si l'utilisateur est un expert du domaine, cette liste de sous-ensembles peut être un résultat qui lui convient. Dans le cas contraire, notre

méthode utilise une approche enveloppe afin de lui fournir comme résultat un sous-ensemble de variables. Le fait d'utiliser une procédure de type enveloppe lors de la phase de détermination du sous-ensemble optimal permet de tenir compte de l'influence des variables sur l'algorithme d'apprentissage. Suivant l'algorithme choisi, les variables sélectionnées ne seront pas forcément les mêmes. Grâce à l'obtention d'un préordre de variables et donc de la limitation du nombre d'itération, les temps de calcul sont largement inférieurs à ceux d'une méthode enveloppe pure.

Notre méthode permet de traiter tous types de variables ainsi que les données volumineuses grâce au fait qu'elle est de type myope et donc très rapide et peu coûteuse. Elle utilise comme direction de recherche la forward selection.

Parce qu'elle laisse libre l'utilisateur de choisir à la fois les critères d'évaluation et la forme du résultat fourni, nous pouvons caractériser notre méthode de « meta-méthode ». En effet, elle peut s'adapter aux caractéristiques du problème traité, en terme de type de variables (quantitatives ou qualitatives), de causes de non pertinence (non pertinence dû au fait que la variable n'apporte pas suffisamment d'information, dû à de la redondance, dû à une variable inconsistante,…) et en terme de forme de résultat (sous-ensemble optimal ou simples indications sur ce sous-ensemble).

4 Expérimentations

4.1 Présentation du cadre expérimental

Nous avons effectué l'étude expérimentale sur un ensemble de quatorze jeux de données issus de la collection de l'UCI [70]. Les variables quantitatives ont été discrétisées à l'aide Fusinter [71].

Le découpage de l'ensemble des données a été effectué comme suit : la totalité des individus a été partagée aléatoirement en deux parties, tout en gardant la répartition initiale des classes. Le premier sous-ensemble d'individus contient 30% des individus et nous servira pour appliquer le processus de sélection de variables. Les tests avec MIFS, ReliefF et PDOBut sont effectués sur ces mêmes 30% d'individus. Les 70% d'individus restants sont utilisés pour les tests avant et après sélection.

Nous avons testés et comparés notre méthode de sélection avec les méthodes MIFS, ReliefF et PDObut. Notre méthode de sélection utilise, dans son fonctionnement, l'algorithme d'apprentissage. Pour cette raison, il nous paraît intéressant de tester l'efficacité de notre méthode en amont de différentes méthodes d'apprentissage :

- ID3, [12], permet la construction de graphes d'induction arborescents à l'aide d'un critère basé sur le gain d'informations. Le seuil minimal du gain d'information a été fixé dans nos expérimentations à 0,05.
- Sipina, [84] et [85], est une méthode conduisant à un graphe d'induction non arborescent. Sipina produit une succession de partitions par fusion et/ou éclatement des nœuds du graphe. Le nombre minimum d'individus que doit posséder chaque sommet est fixé à 5 et le paramètre contrôlant le développement du graphe est fixé à 1.
- Le modèle des bayésiens naïfs, [86], utilise le théorème de Bayes pour estimer les probabilités a posteriori de toutes les classes. Pour chaque individu, la classe avec la probabilité a posteriori la plus élevée est choisi comme prédiction.

Nous avons utilisé deux procédures de validation : une 10-Cross-Validation et cinq 2-Cross-Validation. Nous avons réalisé une étude comparative concernant d'une part le taux d'erreur moyen des diverses méthodes d'apprentissage selon la méthode de sélection employée et d'autre part le nombre de variables sélectionnées par chaque méthode de sélection. Notons de plus que :

- La version de MIFS que nous avons utilisée est la version classique : c'est à dire celle utilisant le critère classique et initial décrit par Battiti (voir [55]) et une stratégie de recherche gloutonne classique.
- La version de ReliefF employée est celle utilisant un critère de sélection contextuel et de consistance classique et une stratégie de recherche utilisant un échantillon d'individus de la taille de l'ensemble des individus du jeu de données.
- ReliefF fourni comme résultat une liste de variables triées en fonction de leur pertinence. Nous avons retenu les variables ayant un poids supérieur ou égal à 0,1.
- MIFS, PODBut et notre méthode fournisse quant à elles le sous-ensemble optimal de variables ou tout au moins un sous-ensemble l'approchant.

4.2 Analyse de l'évaluation expérimentale.

Les résultats des expérimentations sont regroupés dans les tableaux 16 à 39. Ces résultats sont également présentés de manière graphique dans les figures 10 à 18.

Les tableaux 16 à 18 permettent d'évaluer le comportement général des diverses méthodes d'apprentissage testées lorsqu'elles sont associées à une méthode de sélection de variables. En effet,

ils présentent la valeur moyenne du rapport « taux d'erreur avec Sélection/taux d'erreur sans Sélection » de chaque méthode de sélection pour l'ensemble des algorithmes d'apprentissage utilisés et dans le cadre soit d'une 10-Cross-Validation, soit de cinq 2-Cross-Validation. La moyenne de ce rapport est calculée sur l'ensemble des quatorze jeux de données considérés. Les résultats permettent de conclure que, de manière générale, l'ensemble des méthodes de sélection de variables impliquent l'obtention d'un taux d'erreur quasi-équivalent lorsque les variables fournies par ces méthodes ou l'ensemble total des variables sont utilisées. Ainsi, quelle que soit la méthode d'apprentissage utilisée, les taux d'erreur sont corrects et quasiment similaires. On peut toutefois remarquer qu'il existe un léger déficit au niveau de la qualité d'apprentissage, pour l'ensemble des méthodes de sélection, lorsqu'elles sont associées à la méthode d'apprentissage Sipina. Notre méthode de sélection ainsi que MIFS permettent lorsqu'elles sont associées aux Bayésiens Naïfs une nette amélioration de la qualité d'apprentissage. De plus, notre méthode est parmi les méthodes ayant les meilleurs résultats.

Les tableaux 19 à 30 et les figures 10 à 15 permettent d'appréhender de manières plus précises les résultats obtenus. Les tableaux 19 à 24 présentent les taux d'erreur moyen en validation et les écart-type pour notre méthode de sélection ainsi que pour MIFS, PDOBut, et ReliefF. Les figures 14, 15 et 16 présentent graphiquement ces résultats. Les tableaux 25 à 30 présentent, pour chaque jeu de données le taux d'erreur moyen en validation, l'écart type de ce taux d'erreur, l'écart et l'écart relatif existants entre les taux d'erreur avant et après sélection ainsi que la différence qu'il existe entre les écarts type avant et après sélection. Les figures 13, 14 et 15 se contentent de présenter le taux d'erreur moyen en validation.

L'étude de ces résultats nous permet de tirer un certain nombre de remarques qui sont les suivantes :

- Les tableaux 31 et 32 et les figures 16 à 18 permettent d'évaluer la réduction de la taille de l'espace de représentation des données induites par les différentes méthodes de sélection de variables. Il apparaît clairement que l'ensemble des méthodes entraîne une diminution significative de la taille de l'espace de représentation des données. Il existe un certain nombre de distinctions entre les méthodes de sélection :
    - Le nombre de variables sélectionnées par notre méthode varie en fonction de la méthode d'apprentissage qui est utilisée. Notre méthode réduit de manière très significative la taille de cet espace puisqu'en moyenne elle ne conserve que 49,7% (au maximum avec les Bayésiens Naïfs). Du point de vue de la diminution de la taille de l'espace de représentation, elle

constitue la méthode la plus efficace : cette efficacité n'est mise en défaut que sur quelques jeux de données.

- MIFS et ReliefF permettent également de réduire significativement la taille de l'espace de représentation des données. En effet, elles ne conservent en moyenne que 52,92% et respectivement 52,83% des variables initiales. Après notre méthode, ce sont les méthodes qui sont les plus efficaces du point de vue de la réduction de l'espace de représentation.
- PODBut, même si elle permet de réduire l'espace de représentation (62,69% des variables conservées en moyenne), paraît cependant en retrait vis à vis des autres méthodes.
- Cependant, bien que notre méthode soit en moyenne supérieure aux autres du point de vue de la diminution de l'espace de représentation des données, cette tendance peut s'inverser ponctuellement. Et, notre méthode peut ainsi être surpassée par ReliefF, MIFS ou PODBut, pour un jeu de données particulier.

- La tendance générale du taux d'erreur proche pour les apprentissages effectués avec et sans sélection, relevée par le rapport moyen « taux d'erreur avec sélection/taux d'erreur sans sélection » se confirme localement pour la plupart des jeux de données à l'aide des tableaux 16 à 18.
- La totalité des méthodes de sélection impliquent parfois des augmentations du taux d'erreur pour des jeux de données particuliers. Ces déficits en terme de correction demeurent cependant ponctuels. Cependant, les méthodes ReliefF et PODBut connaissent parfois des pics relativement importants, en particulier pour les bases Cleve et Monks-1.
- MIFS et notre méthode sont, du point de vue du taux d'erreur moyen en validation, quasiment équivalentes, bien qu'il arrive ponctuellement que l'une surpasse plus fortement l'autre.
- La stabilité des apprentissages est quasiment similaire, pour une base donnée, que l'on ait utilisé ou non une méthode de sélection de variables et ce, quelle que soit la méthode choisie.
- Du point de vue du coût calculatoire, ReliefF implique un temps de calcul plus important parfois de l'ordre de plusieurs minutes. Ceci s'explique par les nombreuses passes sur le jeu de données que cette méthode nécessite, ce qui n'est pas le cas pour les trois autres méthodes. La méthode PDOBut est relativement coûteuse en temps de calcul ce qui s'explique par le fait que sa complexité dépend du nombre de paires d'individus non encore discriminés. MIFS et notre méthode sont les méthodes les moins coûteuses en temps de calcul et donc les plus rapides.

- Les tableaux 34 à 39 présentent le classement des méthodes en fonction du taux d'erreur obtenu après leur application. Notre méthode est relativement bien classée : elle est classée première ou deuxième pour les algorithmes d'apprentissage ID3 ou les bayésiens naïfs. Elle est moins bien classée pour Sipina.
- L'utilisation d'une approche enveloppe permet à notre méthode de s'adapter à l'algorithme d'apprentissage utilisé. Cependant, cette caractéristique n'entraîne pas un coût calculatoire trop élevé. En effet, le tableau 33 nous indique le nombre d'itérations de l'algorithme d'apprentissage et l'on peut remarquer que le nombre d'itérations ne dépasse pas 9 pour ID3, 7 pour les bayésiens naïfs et 6 pour Sipina. Ainsi le nombre de fois où l'algorithme d'apprentissage est appliqué est relativement réduit.

En conclusion, l'ensemble de ces expérimentations tend à privilégier notre méthode par rapport MIFS, ReliefF et PDOBut. Il nous semble possible de rejeter ReliefF en particulier à cause de son coût calculatoire important. PDOBut paraît également moins efficiente par rapport à MIFS et à notre méthode. Les différentes caractéristiques de chacune de ces deux dernières méthodes semblent plaider en faveur de l'utilisation de l'une d'entre elles. Cependant, notre méthode surpasse MIFS en terme de réduction de l'espace de représentation des données et en terme de rapport moyen « taux d'erreur après sélection/taux d'erreur avant sélection ».

	Notre méthode	MIFS	ReliefF	PDObut
10 –Cross-Validation	1,0676	1,1612	1,2729	0,9936
Cinq 2-Cross-Validation	1,0973	1,1935	1,2785	1,0347

Tableau 16 Rapport «taux d'erreur avec sélection/taux d'erreur sans sélection» moyen avec ID3.

	Notre méthode	MIFS	ReliefF	PDObut
10 –Cross-Validation	1,1502	1,1060	1,2468	1,1474
Cinq 2-Cross-Validation	1,1603	1,0560	1,1853	1,1482

Tableau 17 Rapport « taux d'erreur avec Sélection/taux d'erreur sans Sélection » moyen avec Sipina.

	Notre méthode	MIFS	ReliefF	PDObut
10 -Cross-Validation	0,9576	0,9739	1,2533	1,0645
Cinq 2-Cross-Validation	0,9855	0,9807	1,1753	1,1179

Tableau 18 Rapport « taux d'erreur avec Sélection/taux d'erreur sans Sélection » moyen avec les Bayésiens Naïfs.

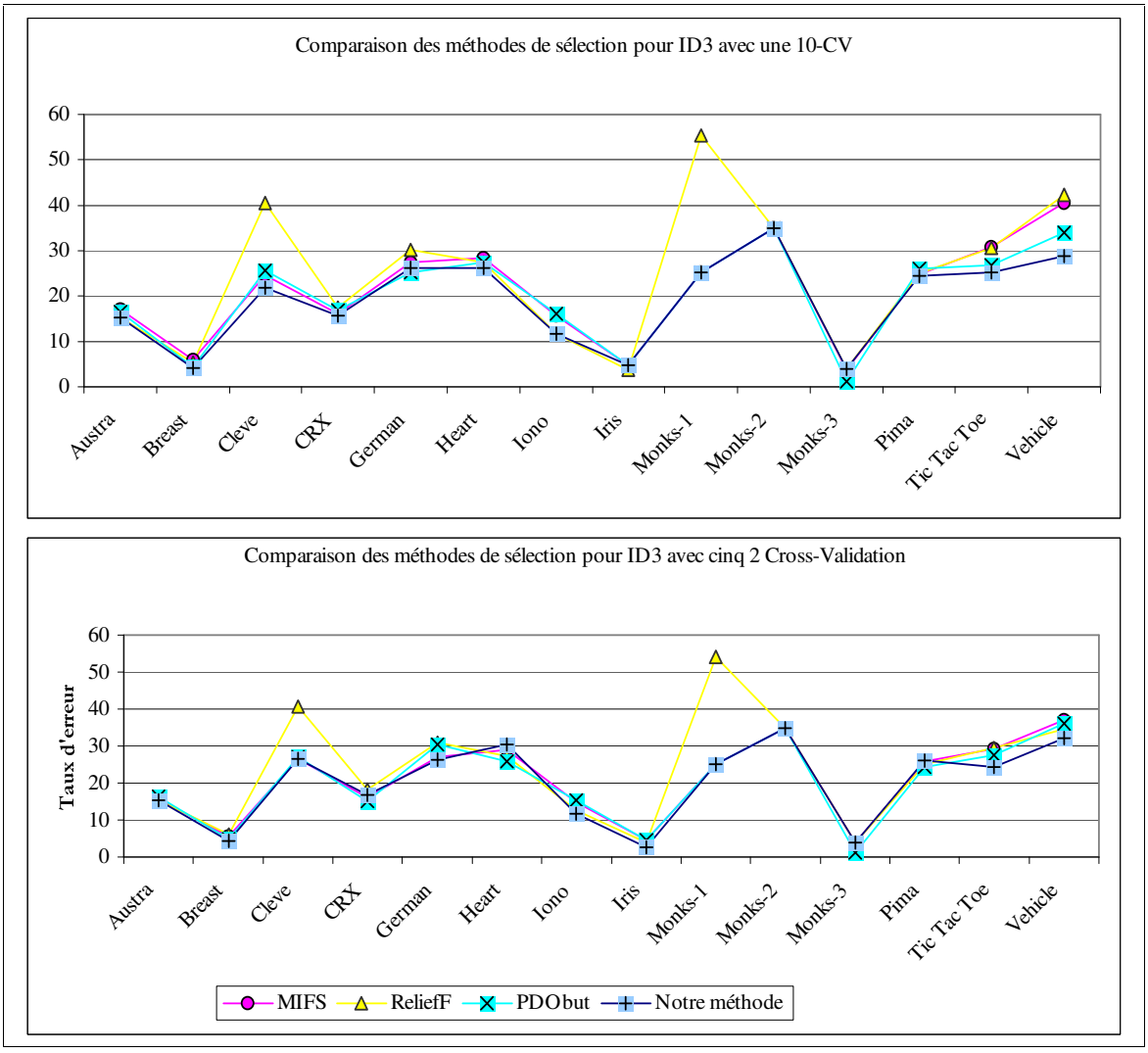


Figure 10 Evaluation de notre méthode de sélection de variables avec ID3.

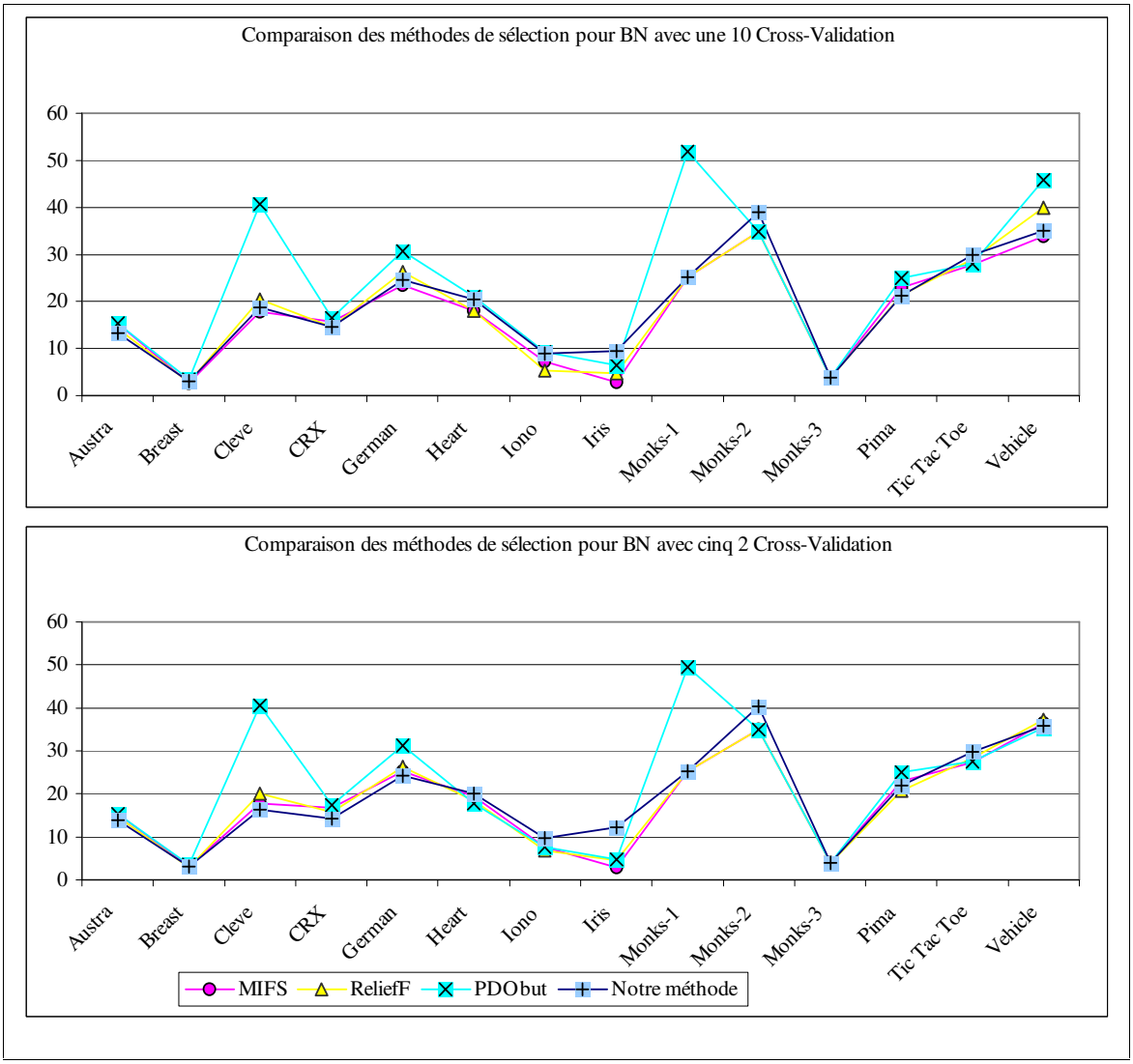


Figure 11 Evaluation de notre méthode de sélection avec les Bayésiens Naïfs.

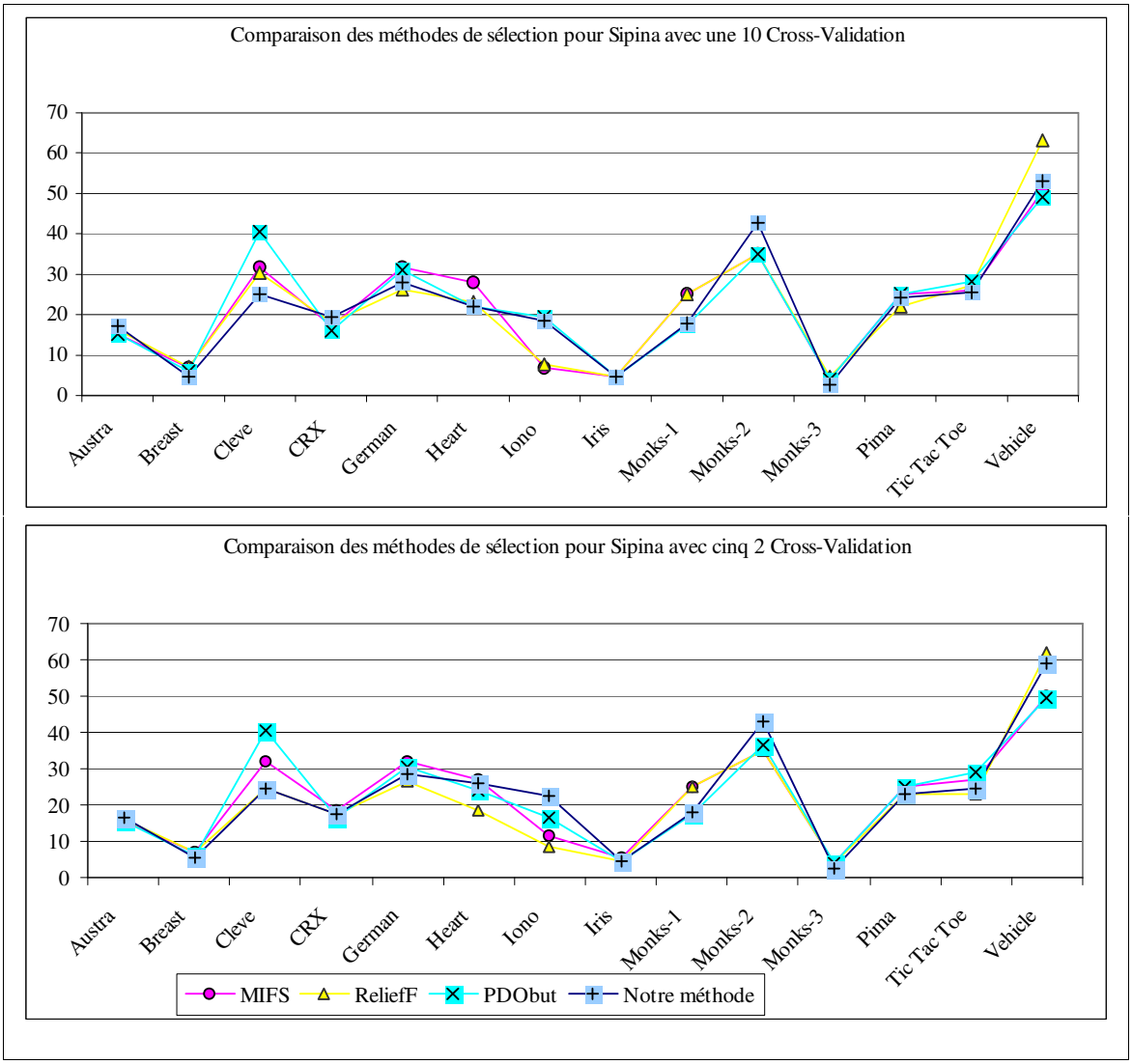


Figure 12 Evaluation de notre méthode de sélection avec Sipina.

Bases	Notre méthode		MIFS		ReliefF		PDObut	
	Taux d'erreur	σ	Taux d'erreur	σ	Taux d'erreur	σ	Taux d'erreur	σ
Austra	15,29	3,48	17,17	4,12	15,31	5,23	16,52	3,42
Breast	4,27	2,8	5,9	2,64	5,29	3,16	4,28	2,32
Cleve	21,9	8,67	24,68	10,27	40,54	7,77	25,69	9,53
CRX	15,7	3,1	16,12	6,7	17,54	5,88	16,94	5,37
German	26,14	4,87	27,43	5,06	30,14	6,01	25,29	4,38
Heart	26,32	11,04	28,42	9,76	27,38	9,06	27,37	10,47
Iono	11,73	5,59	15,75	8,71	11,78	3,94	16,15	11,31
Iris	4,73	4,74	4,82	6,58	3,73	4,57	4,82	6,58
Monks-1	25,18	7,56	25,20	7,71	55,52	3,34	25,19	4,82
Monks-2	34,89	6,71	34,91	6,7	34,9	8,63	34,92	5,25
Monks-3	3,88	2,69	3,86	2,86	3,88	3,34	1,29	1,29
Pima	24,5	5,15	24,87	4,83	25,05	7,69	26,11	5,43
Tic Tac Toe	25,16	6,31	30,81	7,11	30,51	5,9	26,78	2,29
Vehicle	28,75	5,44	40,62	7,39	42,25	6,52	33,94	4,82

Tableau 19 Comparaison des différentes méthodes avec ID3 pour une 10 Cross-Validation.

Bases	Notre méthode		MIFS		ReliefF		PDObut	
	Taux d'erreur	σ	Taux d'erreur	σ	Taux d'erreur	σ	Taux d'erreur	σ
Austra	15,29	1,78	15,71	2,7	15,28	5	16,32	2,69
Breast	4,28	1,76	5,7	1,89	6,11	2,05	4,89	1,35
Cleve	26,7	7,2	26,64	4,11	40,63	7,17	27,07	11,39
CRX	16,94	1,65	16,10	3,51	18,19	4,94	15,09	3,11
German	26,29	5,84	27,00	2,04	31	4,79	30,57	3,24
Heart	30,53	6,78	28,95	6,86	27,37	5,67	25,79	3,07
Iono	11,72	2,91	14,97	1,54	12,58	4,42	15,4	4,78
Iris	2,77	2,26	4,73	5,22	3,81	4,67	4,63	2,88
Monks-1	25,2	1,99	25,21	3,63	54,24	3,28	25,19	5,7
Monks-2	34,92	2,31	34,91	3,02	34,91	1,32	34,9	5,42
Monks-3	3,86	3,34	3,87	1,84	3,87	0,82	1,29	1,42
Pima	26,16	1,85	25,80	2,91	25,06	4,35	24,3	2,48
Tic Tac Toe	24,4	4,54	29,18	3,39	29,6	3,96	27,68	2,85
Vehicle	32,1	2,78	37,14	5,05	34,62	1,95	36,13	2,25

Tableau 20 Comparaison des différentes méthodes avec ID3 pour cinq 2 Cross-Validation.

Bases	Notre méthode		MIFS		ReliefF		PDObut	
	Taux d'erreur	σ	Taux d'erreur	σ	Taux d'erreur	σ	Taux d'erreur	σ
Austra	15,27	3,61	14,28	3,08	15,28	5,15	13,23	3,61
Breast	2,65	2,05	2,86	1,87	3,45	2,56	3,05	2,09
Cleve	17,77	6,14	20,52	11,34	40,67	4,33	18,72	7,57
CRX	15,69	3,99	14,66	5,7	16,53	2,8	14,5	4,63
German	23,43	4,62	26,29	3,63	30,71	4,96	24,57	5,95
Heart	17,89	7,14	17,89	10,04	21,05	10,53	20,53	7,24
Iono	7,25	5,88	5,22	4,4	9,32	6,22	8,87	5,85
Iris	2,82	4,31	4,64	6,17	6,45	7,14	9,45	7,53
Monks-1	25,19	4,68	25,20	7,18	51,9	8,2	25,19	4,82
Monks-2	34,92	5,11	34,92	6,24	34,92	6,65	38,96	3,95
Monks-3	3,85	3,67	3,86	2,87	3,85	3,85	3,87	1,74
Pima	22,83	5,73	21,33	4,3	25,04	3,41	21,14	5,42
Tic Tac Toe	27,83	3,92	28,87	5,42	27,97	4,19	29,93	6,02
Vehicle	33,95	4,18	39,85	8,01	45,82	8,78	34,95	4,33

Tableau 21 Comparaison des différentes méthodes avec les Bayésiens Naïfs pour une 10 Cross-Validation.

Bases	Notre méthode		MIFS		ReliefF		PDObut	
	Taux d'erreur	σ	Taux d'erreur	σ	Taux d'erreur	σ	Taux d'erreur	σ
Austra	15,3	3,83	14,67	3,29	15,29	5,73	13,84	2,86
Breast	3,06	1,45	2,86	1,98	3,46	1,04	3,06	0,92
Cleve	17,74	5,18	20,07	5,15	40,63	3,54	16,37	5,54
CRX	16,73	3,44	15,70	2,46	17,36	1,9	14,25	3,34
German	25,29	1,16	26,29	2,69	31,14	4,06	24,29	4,87
Heart	19,47	4,88	18,42	3,33	17,89	5,37	20	3,16
Iono	7,7	3,25	6,87	1,56	7,71	3,28	9,72	2,99
Iris	2,86	3,81	4,63	4,16	4,72	4,26	12,12	4,76
Monks-1	25,17	4,84	25,19	3,69	49,36	2,81	25,19	5,53
Monks-2	34,92	2,47	34,92	5,04	34,92	4,31	40,38	4,35
Monks-3	3,88	2,73	3,87	1,43	3,87	0,83	3,87	1,43
Pima	23	7,23	20,78	4,47	25,04	2,31	21,89	4,07
Tic Tac Toe	27,28	5,62	28,27	3,65	27,53	1,78	29,77	3,91
Vehicle	36,47	3,17	37,14	5,05	35,46	4,1	35,8	2,84

Tableau 22 Comparaison des différentes méthodes avec les Bayésiens Naïfs pour cinq 2 Cross-Validation.

Bases	Notre méthode		MIFS		ReliefF		PDObut	
	Taux d'erreur	σ	Taux d'erreur	σ	Taux d'erreur	σ	Taux d'erreur	σ
Austra	15,28	6,02	16,35	6,65	15,28	5,25	17,15	3,06
Breast	6,73	4,84	7,13	2,29	5,9	3,8	4,68	2,74
Cleve	31,67	10,87	30,41	10,7	40,56	10,4	25,13	12,45
CRX	17,13	6,05	17,95	5,23	16,12	4,72	19,42	3,24
German	31,71	4,51	26,29	4,53	31	4,61	28	3,79
Heart	27,89	7,82	23,16	6,74	22,11	6,57	22,11	5,67
Iono	6,88	2,58	7,70	6,22	19,4	6,85	18,5	11,67
Iris	4,64	6,17	4,55	9,32	4,64	6,17	4,73	6,24
Monks-1	25,18	3,72	25,19	6,35	17,48	8,4	17,74	5,9
Monks-2	34,89	8,79	34,91	4,86	34,93	8,83	42,8	10,43
Monks-3	3,87	3,09	4,63	2,99	3,86	4,02	2,58	3,25
Pima	25,05	4,36	22,07	4,84	25,07	6,43	24,3	4,46
Tic Tac Toe	26,06	7,5	27,40	6,06	28,27	5,16	25,6	3,53
Vehicle	50,58	5,63	63,17	6,75	49,07	5,07	53,12	5,07

Tableau 23 Comparaison des différentes méthodes avec Sipina pour une 10 Cross-Validation.

Bases	Notre méthode		MIFS		ReliefF		PDObut	
	Taux d'erreur	σ	Taux d'erreur	σ	Taux d'erreur	σ	Taux d'erreur	σ
Austra	15,29	2,74	16,11	2,01	15,3	2,55	16,74	1,89
Breast	6,93	2,78	7,13	2,27	6,11	2,12	5,3	2,18
Cleve	31,76	6,61	24,29	4,24	40,61	6,46	24,3	6,33
CRX	18,6	3,32	17,35	1,88	16,52	5,23	17,55	3,08
German	31,86	3,02	26,29	3,6	30,71	5,07	28,29	2,42
Heart	26,84	3,07	18,42	6	24,21	4,53	25,79	7,7
Iono	11,31	5,71	8,51	3,71	16,58	4,62	22,73	10,83
Iris	5,63	3,59	4,59	4,07	4,68	3,01	4,63	2,88
Monks-1	25,19	2,22	25,20	3,94	17,48	4,55	18,25	6,2
Monks-2	34,92	2,96	34,92	5,86	36,6	8,11	42,8	6,75
Monks-3	3,87	2,02	3,87	0,83	3,88	2,33	2,32	2,2
Pima	25,04	2,64	22,83	3,8	25,04	2,84	23,19	3,29
Tic Tac Toe	27,08	1,78	22,76	3,18	29,02	2,43	24,69	4,84
Vehicle	49,92	4,27	62,02	6,12	49,41	1,24	58,99	5,76

Tableau 24 Comparaison des différentes méthodes avec Sipina pour cinq 2 Cross-Validation.

Bases	Avant Sélection		Après Sélection		Ecart	Ecart relatif	Différence entre σ
	Taux d'erreur	σ	Taux d'erreur	σ			
Austra	16,60	4,57	15,29	3,48	-1,31	-8,57%	-1,09
Breast	5,95	1,95	4,27	2,8	-1,68	-39,34%	0,85
Cleve	18,53	8,68	21,9	8,67	3,37	15,39%	-0,01
CRX	14,73	5,68	15,7	3,1	0,97	6,18%	-2,58
German	31,86	7,53	26,14	4,87	-5,72	-21,88%	-2,66
Heart	27,05	10,29	26,32	11,04	-0,73	-2,77%	0,75
Iono	21,37	8,39	11,73	5,59	-9,64	-82,18%	-2,8
Iris	3,73	4,57	4,73	4,74	1,00	21,14%	0,17
Monks-1	25,22	8,3	25,18	7,56	-0,04	-0,16%	-0,74
Monks-2	34,91	6,79	34,89	6,71	-0,02	-0,06%	-0,08
Monks-3	1,28	1,28	3,88	2,69	2,60	67,01%	1,41
Pima	26,11	5,43	24,5	5,15	-1,61	-6,57%	-0,28
Tic Tac Toe	33,43	5	25,16	6,31	-8,27	-32,87%	1,31
Vehicle	34,24	4,96	28,75	5,44	-5,49	-19,10%	0,48

Tableau 25 Evaluation de notre méthode de sélection avec ID3 pour une 10 Cross-Validation.

Bases	Avant Sélection		Après Sélection		Ecart	Ecart relatif	Différence entre σ
	Taux d'erreur	σ	Taux d'erreur	σ			
Austra	15,91	2,58	15,29	1,78	-0,62	-4,05%	-0,8
Breast	5,7	1,89	4,28	1,76	-1,42	-33,18%	-0,13
Cleve	32,23	5,68	26,7	7,2	-5,53	-20,71%	1,52
CRX	14,66	2,43	16,94	1,65	2,28	13,46%	-0,78
German	28,57	4,58	26,29	5,84	-2,28	-8,67%	1,26
Heart	28,95	4,71	30,53	6,78	1,58	5,18%	2,07
Iono	13,39	3,62	11,72	2,91	-1,67	-14,25%	-0,71
Iris	3,81	4,67	2,77	2,26	-1,04	-37,55%	-2,41
Monks-1	25,19	5,7	25,2	1,99	0,01	0,04%	-3,71
Monks-2	34,92	3,57	34,92	2,31	0	0,00%	-1,26
Monks-3	1,29	0,81	3,86	3,34	2,57	66,58%	2,53
Pima	24,3	2,48	26,16	1,85	1,86	7,11%	-0,63
Tic Tac Toe	22,8	3,94	24,4	4,54	1,6	6,56%	0,6
Vehicle	29,41	3,49	32,1	2,78	2,69	8,38%	-0,71

Tableau 26 Evaluation de notre méthode de sélection avec ID3 pour cinq 2 Cross-Validation.

Bases	Avant Sélection		Après Sélection		Ecart	Ecart relatif	Différence entre σ
	Taux d'erreur	σ	Taux d'erreur	σ			
Austra	16,73	3,95	15,28	6,02	-1,45	-9,49%	2,07
Breast	7,13	2,29	6,73	4,84	-0,4	-5,94%	2,55
Cleve	21,47	8,57	31,67	10,87	10,2	32,21%	2,3
CRX	16,3	6,22	17,13	6,05	0,83	4,85%	-0,17
German	28,14	5,5	31,71	4,51	3,57	11,26%	-0,99
Heart	23,16	10,04	27,89	7,82	4,73	16,96%	-2,22
Iono	7,73	6,95	6,88	2,58	-0,85	-12,35%	-4,37
Iris	4,64	6,17	4,64	6,17	0	0,00%	0
Monks-1	20,11	4,89	25,18	3,72	5,07	20,14%	-1,17
Monks-2	38,24	7	34,89	8,79	-3,35	-9,60%	1,79
Monks-3	1,79	2,58	3,87	3,09	2,08	53,75%	0,51
Pima	24,3	4,46	25,05	4,36	0,75	2,99%	-0,1
Tic Tac Toe	20,67	3,77	26,06	7,5	5,39	20,68%	3,73
Vehicle	47,26	6,24	50,58	5,63	3,32	6,56%	-0,61

Tableau 27 Evaluation de notre méthode de sélection avec Sipina pour une 10 Cross-Validation.

Bases	Avant Sélection		Après Sélection		Ecart	Ecart relatif	Différence entre σ
	Taux d'erreur	σ	Taux d'erreur	σ			
Austra	17,36	1,58	15,29	2,74	-2,07	-13,54%	1,16
Breast	7,13	2,27	6,93	2,78	-0,2	-2,89%	0,51
Cleve	21,46	8,43	31,76	6,61	10,3	32,43%	-1,82
CRX	16,31	3,96	18,6	3,32	2,29	12,31%	-0,64
German	29,29	4,74	31,86	3,02	2,57	8,07%	-1,72
Heart	21,58	4,53	26,84	3,07	5,26	19,60%	-1,46
Iono	7,7	4,35	11,31	5,71	3,61	31,92%	1,36
Iris	4,68	3,01	5,63	3,59	0,95	16,87%	0,58
Monks-1	19,76	11,8	25,19	2,22	5,43	21,56%	-9,58
Monks-2	42,28	1,19	34,92	2,96	-7,36	-21,08%	1,77
Monks-3	2,31	2,2	3,87	2,02	1,56	40,31%	-0,18
Pima	23,19	3,29	25,04	2,64	1,85	7,39%	-0,65
Tic Tac Toe	24,12	4,26	27,08	1,78	2,96	10,93%	-2,48
Vehicle	47,56	1,96	49,92	4,27	2,36	4,73%	2,31

Tableau 28 Evaluation de notre méthode avec Sipina pour cinq 2 Cross-Validation.

Bases	Avant Sélection		Après Sélection		Ecart	Ecart relatif	Différence entre σ
	Taux d'erreur	σ	Taux d'erreur	σ			
Austra	14,26	4,58	15,27	3,61	1,01	6,61%	-0,97
Breast	2,65	1,31	2,65	2,05	0	0,00%	0,74
Cleve	21	6,63	17,77	6,14	-3,23	-18,18%	-0,49
CRX	14,67	3,14	15,69	3,99	1,02	6,50%	0,85
German	23,71	6,58	23,43	4,62	-0,28	-1,20%	-1,96
Heart	17,37	7,46	17,89	7,14	0,52	2,91%	-0,32
Iono	6,83	5,06	7,25	5,88	0,42	5,79%	0,82
Iris	6,45	7,14	2,82	4,31	-3,63	-128,72%	-2,83
Monks-1	25,22	6	25,19	4,68	-0,03	-0,12%	-1,32
Monks-2	38,94	4,14	34,92	5,11	-4,02	-11,51%	0,97
Monks-3	3,88	2,9	3,85	3,67	-0,03	-0,78%	0,77
Pima	21,14	5,42	22,83	5,73	1,69	7,40%	0,31
Tic Tac Toe	29,61	5,15	27,83	3,92	-1,78	-6,40%	-1,23
Vehicle	34,27	5,52	33,95	4,18	-0,32	-0,94%	-1,34

Tableau 29 Evaluation de notre méthode de sélection avec les Bayésiens Naïfs pour une 10 Cross-Validation.

Bases	Avant Sélection		Après Sélection		Ecart	Ecart relatif	Différence entre σ
	Taux d'erreur	σ	Taux d'erreur	σ			
Austra	14,47	3,52	15,3	3,83	0,83	5,42%	0,31
Breast	2,86	1,98	3,06	1,45	0,2	6,54%	-0,53
Cleve	20,08	2,25	17,74	5,18	-2,34	-13,19%	2,93
CRX	15,09	3,37	16,73	3,44	1,64	9,80%	0,07
German	23,86	3,9	25,29	1,16	1,43	5,65%	-2,74
Heart	17,37	6,36	19,47	4,88	2,1	10,79%	-1,48
Iono	7,71	2,42	7,7	3,25	-0,01	-0,13%	0,83
Iris	4,72	4,26	2,86	3,81	-1,86	-65,03%	-0,45
Monks-1	25,69	6,98	25,17	4,84	-0,52	-2,07%	-2,14
Monks-2	38,95	3,81	34,92	2,47	-4,03	-11,54%	-1,34
Monks-3	3,87	1,64	3,88	2,73	0,01	0,26%	1,09
Pima	21,89	4,07	23	7,23	1,11	4,83%	3,16
Tic Tac Toe	30,34	3,82	27,28	5,62	-3,06	-11,22%	1,8
Vehicle	36,3	1,24	36,47	3,17	0,17	0,47%	1,93

Tableau 30 Evaluation de notre méthode de sélection avec les Bayésiens Naïfs pour cinq 2 Cross-Validation.

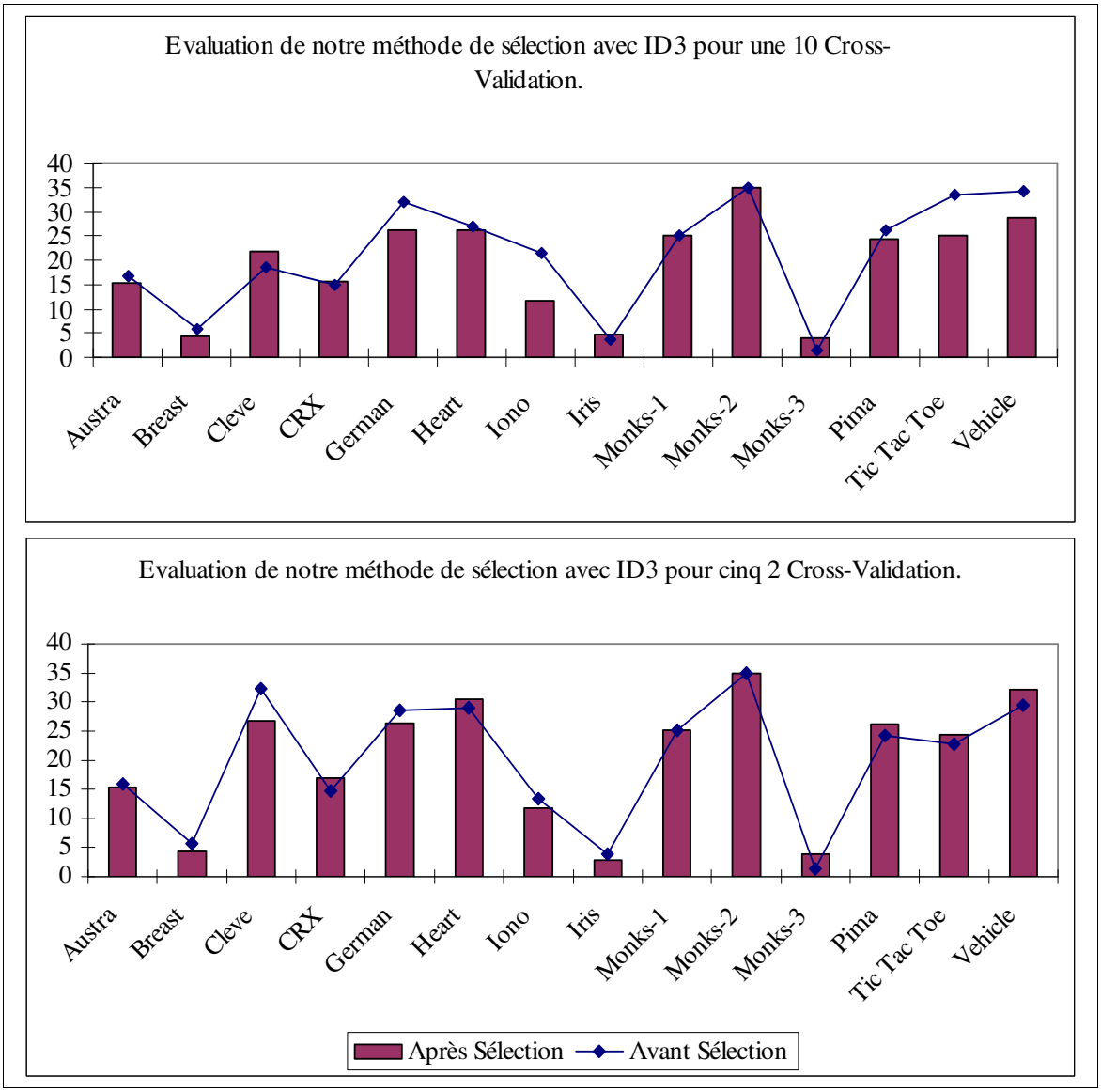


Figure 13 Evaluation de notre méthode de sélection avec ID3.

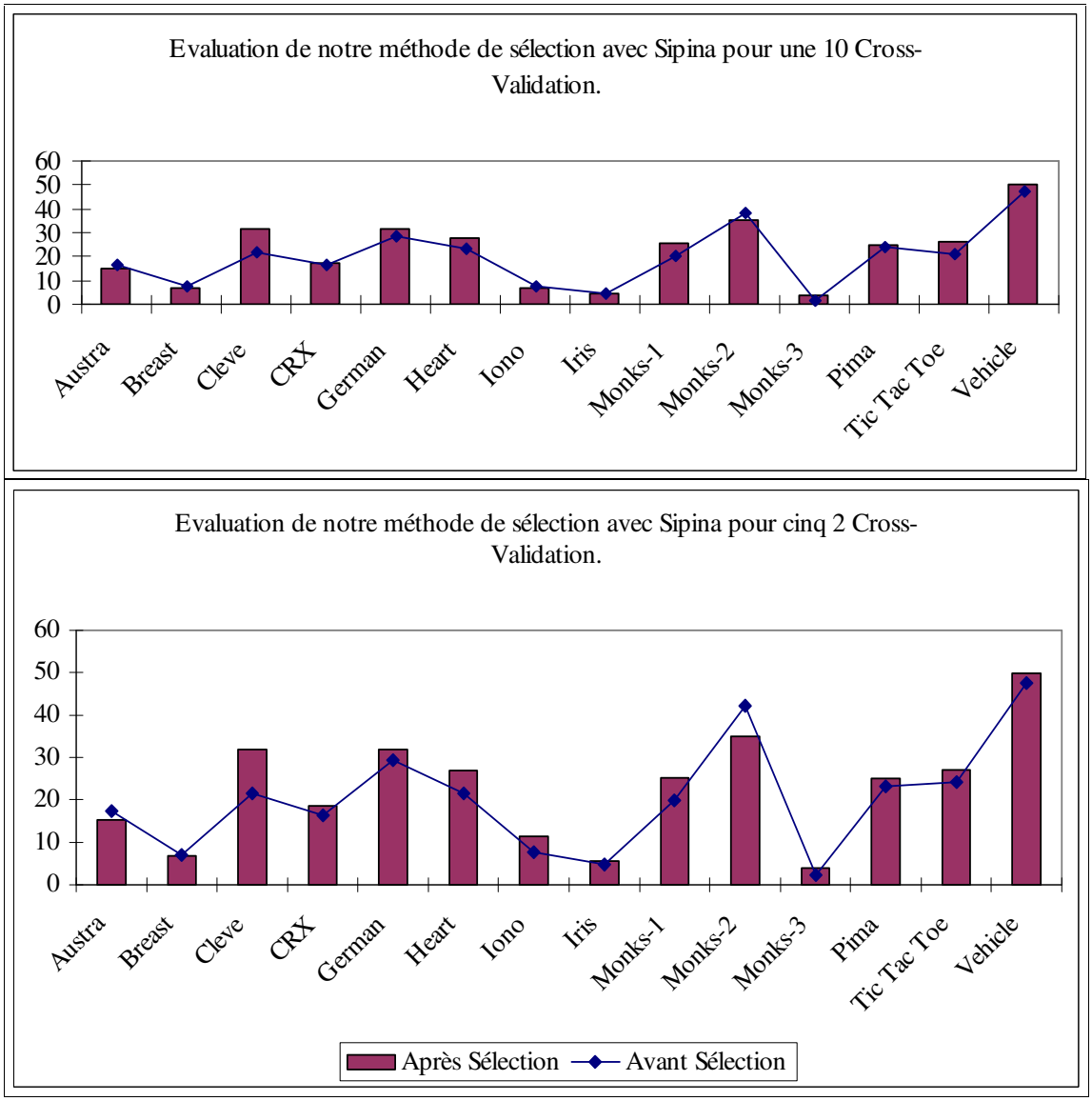


Figure 14 Evaluation de notre méthode de sélection avec Sipina.

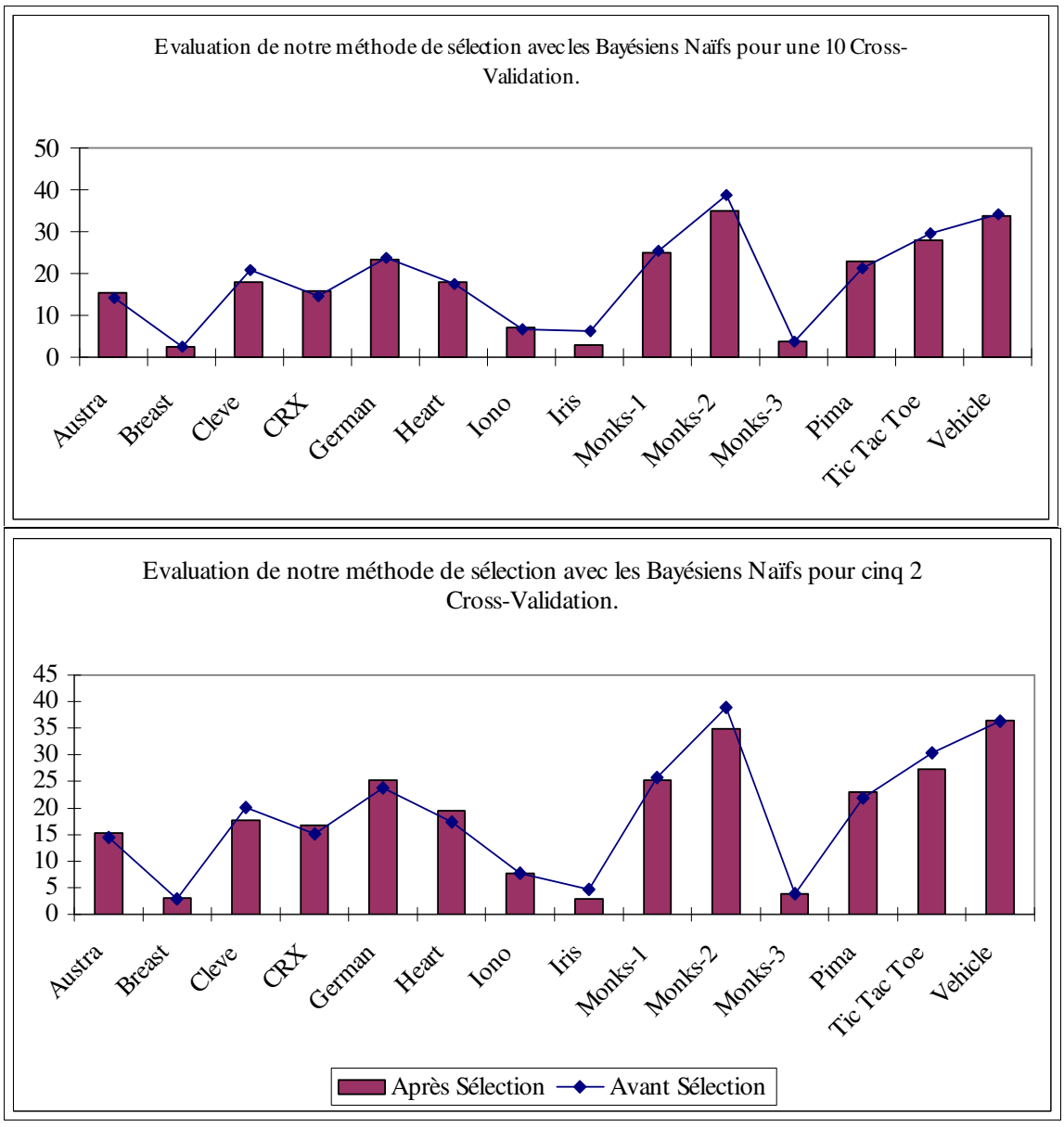


Figure 15 Evaluation de notre méthode de sélection avec les Bayésiens Naïfs.

Bases	Sans Sélection	Notre méthode avec ID3	Notre méthode avec les Bayésiens Naïfs	Notre méthode avec Sipina	ReliefF	MIFS	PDObut
Austra	14	1	2	1	2	13	10
Breast	9	3	7	4	6	9	5
Cleve	13	7	5	1	6	8	5
CRX	15	3	5	3	2	7	8
German	20	5	9	1	14	3	9
Heart	13	2	8	2	2	13	7
Iono	34	2	26	26	25	8	3
Iris	4	3	2	2	4	3	3
Monks-1	6	1	1	1	2	1	3
Monks-2	6	1	1	1	2	2	6
Monks-3	6	2	2	2	2	3	4
Pima	8	2	5	1	7	4	8
Tic Tac Toe	9	7	7	3	5	3	7
Vehicle	18	14	12	10	18	6	8

Tableau 31 Nombre de variables sélectionnées.

Bases	Notre méthode avec ID3	Notre méthode avec les Bayésiens Naïfs	Notre méthode avec Sipina	ReliefF	MIFS	PDObut
Austra	7,14%	14,29%	7,14%	14,29%	92,86%	71,43%
Breast	33,33%	77,78%	44,44%	66,67%	100,00%	55,56%
Cleve	53,85%	38,46%	7,69%	46,15%	61,54%	38,46%
CRX	20,00%	33,33%	20,00%	13,33%	46,67%	53,33%
German	25,00%	45,00%	5,00%	70,00%	15,00%	45,00%
Heart	15,38%	61,54%	15,38%	15,38%	100,00%	53,85%
Iono	5,88%	76,47%	76,47%	73,53%	23,53%	8,82%
Iris	75,00%	50,00%	50,00%	100,00%	75,00%	75,00%
Monks-1	16,67%	16,67%	16,67%	33,33%	16,67%	50,00%
Monks-2	16,67%	16,67%	16,67%	33,33%	33,33%	100,00%
Monks-3	33,33%	33,33%	33,33%	33,33%	50,00%	66,67%
Pima	25,00%	62,50%	12,50%	87,50%	50,00%	100,00%
Tic Tac Toe	77,78%	77,78%	33,33%	55,56%	33,33%	77,78%
Vehicle	77,78%	66,67%	55,56%	100,00%	33,33%	44,44%

Tableau 32 Proportion de variables sélectionnées.

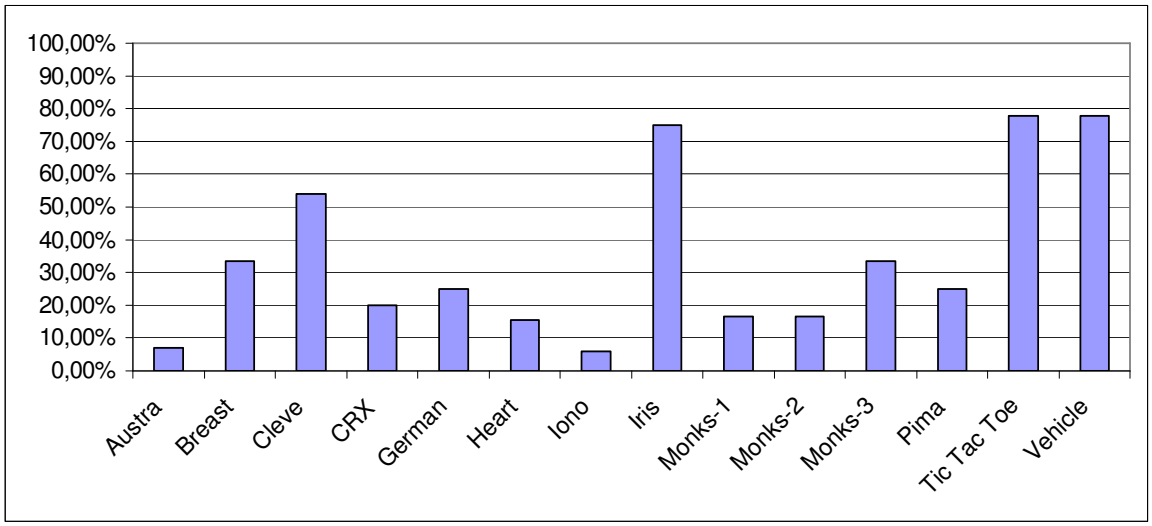


Figure 16 Evaluation de la diminution de l'espace de représentation avec notre méthode pour ID3.

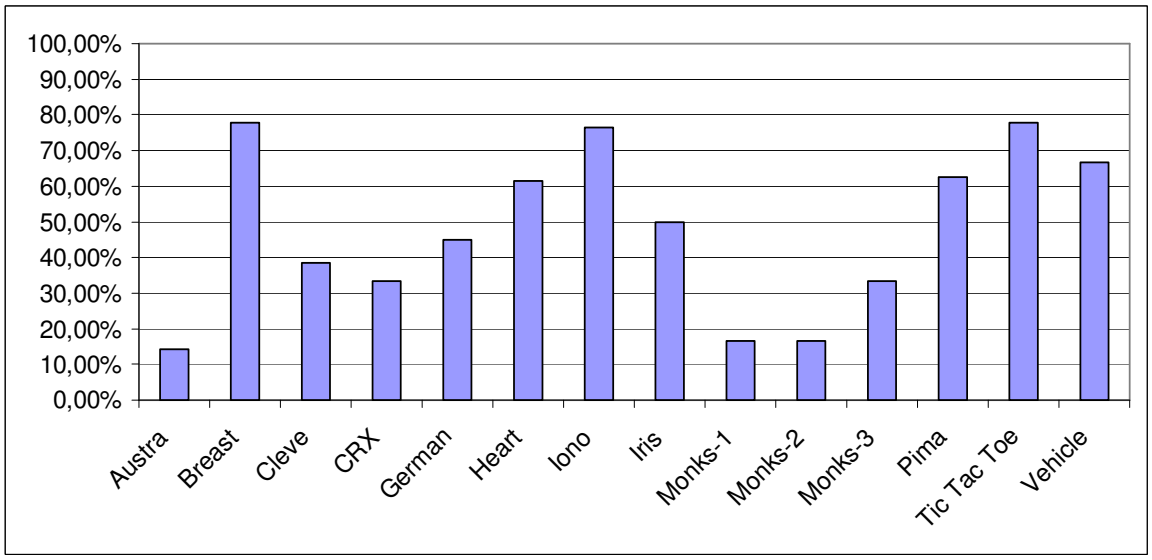


Figure 17 Evaluation de la diminution de l'espace de représentation avec notre méthode pour les bayésiens naïfs.

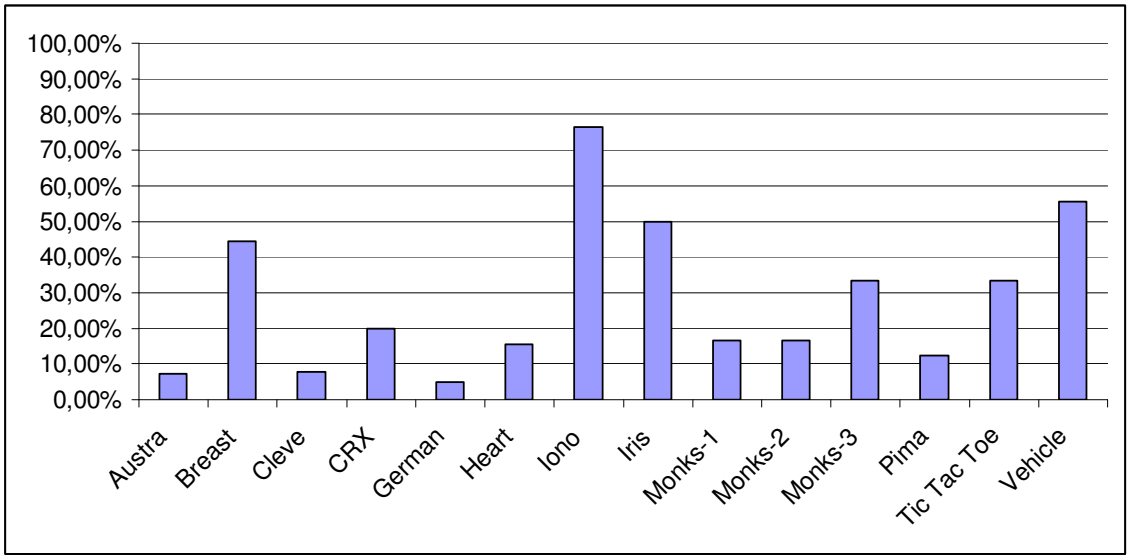


Figure 18 Evaluation de la diminution de l'espace de représentation avec notre méthode pour Sipina.

Bases	ID3	Bayésiens naïfs	Sipina
Austra	2	3	2
Breast	3	5	5
Cleve	5	4	2
CRX	2	3	2
German	5	7	2
Heart	2	4	2
Iono	3	6	6
Iris	3	3	3
Monks-1	2	2	2
Monks-2	2	2	2
Monks-3	2	2	2
Pima	3	4	2
Tic Tac Toe	4	4	3
Vehicle	9	7	6

Tableau 33 Nombre d'itérations lors de la partie enveloppe de notre méthode.

Bases	Notre méthode	MIFS	ReliefF	PDObut
Austra	1	4	2	3
Breast	1	4	3	2
Cleve	1	2	4	3
CRX	1	2	4	3
German	2	3	4	1
Heart	1	4	3	2
Iono	1	3	2	4
Iris	2	3	1	3
Monks-1	1	3	4	2
Monks-2	1	3	2	4
Monks-3	3	2	3	1
Pima	1	2	3	4
Tic Tac Toe	1	4	3	2
Vehicle	1	3	4	2
Rang moyen	1	3	3	3

Tableau 34 Rang des méthodes avec ID3 pour une 10 Cross-Validation.

Bases	Notre méthode	MIFS	ReliefF	PDObut
Austra	2	3	1	4
Breast	1	3	4	2
Cleve	2	1	4	3
CRX	3	2	4	1
German	1	2	4	3
Heart	4	3	2	1
Iono	1	3	2	4
Iris	1	4	2	3
Monks-1	2	3	4	1
Monks-2	4	2	2	1
Monks-3	2	3	3	1
Pima	4	3	2	1
Tic Tac Toe	1	3	4	2
Vehicle	1	4	2	3
Rang moyen	2	3	3	2

Tableau 35 Rang des méthodes avec ID3 pour cinq 2 Cross-Validation.

Bases	Notre méthode	MIFS	ReliefF	PDObut
Austra	3	2	4	1
Breast	1	2	4	3
Cleve	1	3	4	2
CRX	3	2	4	1
German	1	3	4	2
Heart	1	1	4	3
Iono	2	1	4	3
Iris	1	2	3	4
Monks-1	1	3	4	1
Monks-2	1	1	1	4
Monks-3	1	3	1	4
Pima	3	2	4	1
Tic Tac Toe	1	3	2	4
Vehicle	1	3	4	2
Rang moyen	2	2	3	3

Tableau 36 Rang des méthodes avec les bayésiens naïfs pour une 10 Cross-Validation.

Bases	Notre méthode	MIFS	ReliefF	PDObut
Austra	4	2	3	1
Breast	2	1	4	2
Cleve	2	3	4	1
CRX	3	2	4	1
German	2	3	4	1
Heart	3	2	1	4
Iono	2	1	3	4
Iris	1	2	3	4
Monks-1	1	2	4	2
Monks-2	1	1	1	4
Monks-3	4	1	1	1
Pima	3	1	4	2
Tic Tac Toe	1	3	2	4
Vehicle	3	4	1	2
Rang moyen	2	2	3	2

Tableau 37 Rang des méthodes avec les bayésiens naïfs pour cinq 2 Cross-Validation.

Bases	Notre méthode	MIFS	ReliefF	PDObut
Austra	1	3	1	4
Breast	3	4	2	1
Cleve	3	2	4	1
CRX	2	3	1	4
German	4	1	3	2
Heart	4	3	1	1
Iono	1	2	4	3
Iris	2	1	2	4
Monks-1	3	4	1	2
Monks-2	1	2	3	4
Monks-3	3	4	2	1
Pima	3	1	4	2
Tic Tac Toe	2	3	4	1
Vehicle	2	4	1	3
Rang moyen	2	3	2	2

Tableau 38 Rang des méthodes avec Sipina pour une 10 Cross-Validation.

Bases	Notre méthode	MIFS	ReliefF	PDObut
Austra	1	3	2	4
Breast	3	4	2	1
Cleve	3	1	4	2
CRX	4	2	1	3
German	4	1	3	2
Heart	4	1	2	3
Iono	2	1	3	4
Iris	4	1	3	2
Monks-1	3	4	1	2
Monks-2	1	1	3	4
Monks-3	2	2	4	1
Pima	3	1	3	2
Tic Tac Toe	3	1	4	2
Vehicle	2	4	1	3
Rang moyen	3	2	3	3

Tableau 39 Rang des méthodes avec Sipina pour cinq 2 Cross-Validation.

5 Conclusion

Il existe un très grand nombre de méthodes de sélection de variables. Certaines d'entre elles ne peuvent pas être utilisées sur des problèmes réels. En effet, les méthodes enveloppes ainsi que les méthodes filtres complètes et les méthodes économétriques sont extrêmement coûteuse en ressources et en temps de calcul. Les méthodes développées par [7, 30, 35] ne peuvent être employées que rarement du fait qu'elles ne traitent que des variables booléennes. Les méthodes ne traitant qu'un seul type de variables ne sont pas toujours applicables au problème traités.

Nous proposons une méta-méthode de type hybride combinant une approche filtre myope et une approche enveloppe. L'utilisateur doit sélectionner un ensemble de critères myopes appartenant à chacune des catégories existantes (information, indépendance, consistance et distance). Notre méthode peut ainsi s'adapter et se moduler en fonction du problème considéré. Notre méthode utilise une procédure d'agrégation des préférences dans le but d'obtenir un préordre sur les variables, c'est à dire une liste de sous-ensembles disjoints de variables triés en fonction de leur pertinence. Une approche enveloppe est employée pour obtenir non plus une liste de sous-ensembles de variables mais le sous-ensemble de variables considéré comme optimal par notre méthode. L'algorithme d'apprentissage est choisi par l'utilisateur. Ceci permet à notre méthode de s'ajuster à la phase d'apprentissage qui suivra et de tenir compte de l'influence du sous-ensemble de variables sélectionnées sur les performances de l'algorithme d'apprentissage. L'inconvénient des approches enveloppe lié au temps de calcul est ici contourné puisque l'algorithme d'apprentissage est appliqué à un nombre limité de sous-ensemble de variables : au maximum sept itérations.

Les caractéristiques de notre méthode sont les suivantes :

- Son type d'approche est à la fois filtre et enveloppe ;
- Dans sa partie filtre, notre méthode est de type myope ;
- Dans sa partie enveloppe, elle utilise l'algorithme d'apprentissage choisi par l'utilisateur ;
- Sa direction de recherche est la forward selection ;
- Elle peut traiter tous types de variables ;
- La catégorie de critères de sélection est choisie par l'utilisateur, plusieurs catégories peuvent être représentées ;

- Le résultat qu'elle fournit peut être soit sous la forme d'un sous-ensemble de variables soit sous la forme d'une liste de sous-ensembles de variables.

Les expérimentations nous ont montré que les méthodes telles que ReliefF et PDOBut ne sont pas applicables car elles nécessitent des temps de calcul beaucoup trop important dès que le nombre d'individus et de variables augmente. Au niveau de la diminution de l'espace de représentation des données et au niveau de la diminution du taux d'erreur après le processus de sélection, MIFS et notre méthode sont parmi les plus efficaces.

Nous avons évaluer notre méthode suivant plusieurs axes :

- Notre méthode permet de traiter les données volumineuses et bruitées ;
- C'est une méthode relativement rapide. Elle est comparable en temps de calcul aux méthodes telles que MIFS ;
- Elle permet de réduire conséquemment la taille de l'espace de représentation ;
- Elle permet relativement souvent d'améliorer la qualité d'apprentissage ou tout au moins de la garder constante et de garantir la stabilité du modèle ;
- L'utilisabilité de notre méthode semble bonne : le paramétrage de l'algorithme est facile, intelligible, intuitif et peut permettre l'intervention de l'utilisateur ou d'un expert du domaine, la présentation des résultats est explicite et intelligible.

Enfin, nous notons qu'il serait intéressant au vue des résultats des expérimentations d'aller vers une méta-méthode de sélection combinant plusieurs méthodes de sélection vraiment différentes telles que notre méthode, MIFS, une méthode utilisant les algorithmes génétiques et une méthode utilisant les réseaux de neurones et fournissant un résultat qui tient compte et met en commun les résultats des différentes méthodes.