Sélection rapide en apprentissage supervisé

Pierre-Emmanuel JOUVE *, Gaëlle LEGRAND*, Nicolas NICOLOYANNIS *

*LABORATOIRE ERIC, Université Lumière - Lyon2
Bâtiment L, 5 av. Pierre Mendès-France
69 676 BRON cedex FRANCE
{pierre.jouve, gaelle.legrand}@eric.univ-lyon2.fr, nicolas.nicoloyannis@univ-lyon2.fr
http://eric.univ-lyon2.fr

Résumé. La sélection de variables (SdV) permet de réduire l'espace de représentation des données. Ce processus est de plus en plus critique en raison de l'augmentation de la taille des bases de données. Traditionnellement, les méthodes de SdV nécessitent plusieurs accès au jeu de données, ce qui peut représenter une part relativement importante du temps d'execution de ces algorithmes. Nous proposons une nouvelle méthode efficiente et rapide (ne nécessitant qu'un unique accès aux données). Cette méthode utilise les algorithmes génétiques ainsi que des mesures de validité de classification non supervisée (cns).

1 Introduction

La taille des bases de données étant de plus en plus importante, l'amélioration de la qualité de l'espace de représentation des données (ERD) est devenue un problème majeur de l'ECD. L'une des difficultés majeures liée à l'ERD est la dimension des données (le nombre de variables descriptives caractérisant chacun des objets). Ce problème peut se résumer par la phrase de Liu et Motoda [Liu et Motoda, 1998] "Less is more." qui signifie que si l'on désire extraire de l'information utile et compréhensible à partir de nos données, il convient en premier lieu de retirer les parties non pertinentes. La sélection de variables (SdV) permet de résoudre ce problème. C'est un processus choisissant un sous-ensemble optimal de variables selon un critère particulier. Il permet l'élimination de variables inutiles et/ou redondantes, autorisant ainsi l'accélération et l'amélioration de la précision prédictive des processus d'apprentissage. Il existe deux familles d'algorithmes de SdV : les méthodes "Enveloppe" [Kohavi et John, 1997] et les méthodes "Filtre"[Kira et Rendell, 1992]. La différence fondamentale entre ces deux familles réside dans le fait que la première est liée à l'algorithme d'induction utilisé (ce qui lui confère un coût calculatoire bien souvent trop important) alors que la seconde est totalement indépendante.

Les approches filtre sont de 4 types : exhaustive, heuristique, probabiliste et sélection en un seul parcours de base. **Les méthodes exhaustives**(MDLM [Sheinvald *et al.*, 1990], FOCUS [Almuallim et Dietterich, 1991]...) testent tous les sous-ensembles possibles de variables, ces algorithmes sont donc le plus souvent impossible à appliquer du fait de leur coût calculatoire trop élevé. **Les méthodes heuristiques** sont très nombreuses, la plus connue est RELIEF[Kira et Rendell, 1992]. Sa complexité est linéaire selon le nombre d'objets et le nombre d'itérations effectuées. Il existe également des méthodes

du type Branch and Bound telle que ABB [Liu et Motoda, 1998]. Ces méthodes requièrent plusieurs accès à la base de données. **Les méthodes probabilistes** sont essentiellement représentées par LVF [Liu et Setiono, 1996]. Sa complexité est de l'ordre de celle de RELIEF. Comme les méthodes précédentes, ces méthodes requièrent plusieurs accès à la base de données. **Les méthodes de sélection en un seul parcours de base** sont des processus itératifs qui comme leur nom l'indique ne nécessitent qu'un seul scan de base. Pour ce faire, des mesures rapides de corrélation sont utilisées (coefficient de corrélation linéaire de Pearson, coefficient de corrélation de rangs de Kendall,...). Ce type de méthodes est représenté par MIFS [Battiti, 1994], CFS [Hall, 2000], et la méthode proposée par Lallich et Rakotomalala, [Lallich et Rakotomalala, 2000]. Ces méthodes, qui sont les plus rapides et qui sont relativement efficaces, paraissent les plus intéressantes.

Nous proposons une nouvelle méthode de SdV rapide et efficiente. Cette méthode requiert uniquement un passage sur les données. Elle utilise un algorithme génétique et une mesure d'evaluation de la validité de classification non supervisée (cns) [1]. La prochaine section introduit les concepts et formalismes utilisés, la troisième section consiste en la présentation de la méthode de SdV proposée. Nous proposons des evaluations expérimentales de notre méthode dans la dernière section.

2 Concepts et Formalismes Introductifs

Cette section est intégralement composée d'éléments essentiels à la présentation de notre méthode de SdV (notations, définitions...). Elle consiste en la présentation d'indices pour l'évaluation et la comparaison de la validité de cns.

Notation 1
$O = \{o_i, i = 1..n\}$ *ensemble d'objets*
$EV = \{V_1,..,V_p\}$ *l'ERD constitué de p variables décrivant les objets de O.*
$EV_\star \subseteq EV$ *un sous-espace de EV*
$o_i = \{o_{i_1},..,o_{i_p}\}$ *un objet de O,* o_{i_j} *correspond à la valeur de* o_i *pour la variable* V_j
$P = \{C_1,...,C_z\}$ *une partition de O en z classes* $(\forall k, C_k \subseteq O)$

Afin de rendre compte de la similarité entre objets, nous utiliserons ici la notion de lien (selon une variable) défini comme suit :

Définition 1 ***Lien entre 2 objets***
A chaque variable V_i *est associée une fonction* $Lien_i$ *qui définit un lien (une similarité) ou un non-lien (une dissimilarité) selon* V_i *entre deux objets de O* (o_a *et* o_b) *:*

$$Lien_i(o_{a_i},o_{b_i}) = \begin{cases} 1 \text{ si une propriété déterminant un lien (selon } V_i) \text{ entre } o_a \text{ et } o_b \text{ est vérifiée} \\ 0 \text{ sinon (non-lien)} \end{cases} \quad (1)$$

Exemples :
- *Pour une variable catégorielle* V_i*, on peut définir naturellement* $Lien_i$ *comme suit :*

$$Lien_i(o_{a_i},o_{b_i}) = \delta_{sim}(o_{a_i},o_{b_i}) = \begin{cases} 1 \text{ si } o_{a_i} = o_{b_i} \\ 0 \text{ sinon} \end{cases}$$

1. (une cns représente ici une partition de l'ensemble des objets d'un jeu de données)

- *Pour une variable quantitative V_i, on peut par exemple définir $Lien_i$ comme suit :*
$Lien_i(o_{a_i},o_{b_i}) = \begin{cases} 1 \text{ si } |o_{a_i} - o_{b_i}| \leq \delta, \text{ avec } \delta \text{ un seuil fixé par l'utilisateur} \\ 0 \text{ sinon} \end{cases}$

2.1 Indices pour l'Evaluation de l'homogénéité interne des classes et de la séparation entre classes d'une cns

Classiquement, l'évaluation de la validité de cns repose sur l'étude de l'homogénéité interne de ses classes ou de la séparation de ses classes. Nous introduisons ici des indices permettant l'évaluation de chacun de ces points.

Pour **évaluer l'homogénéité interne d'une cns** (une partition P de O) **dans un espace $\mathbf{EV_\star}$**, on peut utiliser l'indice LM (resp. NLM) qui dénombre le nombre de liens (resp. non-liens) entre objets de même classe de la cns :

$$LM_{EV_\star}(P) = \sum_{g=1..k} \sum_{\substack{o_a \in C_g, o_b \in C_g, \\ a<b}} \sum_{\substack{i \text{ tel que} \\ V_i \in EV_\star}} (lien_i(o_{a_i},o_{b_i}))$$

$$NLM_{EV_\star}(P) = \sum_{g=1..k} \sum_{\substack{o_a \in C_g, o_b \in C_g, \\ a<b}} \sum_{\substack{i \text{ tel que} \\ V_i \in EV_\star}} (1 - lien_i(o_{a_i},o_{b_i}))$$

Ainsi, l'homogénéité interne d'une cns P est d'autant plus forte que $LM_{EV_\star}(P)$ (resp. $NLM_{EV_\star}(P)$) est élevé (resp. faible).

Pour **évaluer la séparation entre classes d'une cns dans un espace $\mathbf{EV_\star}$**, on peut utiliser l'indice LD (resp. NLD) qui dénombre le nombre de liens (resp. non-liens) entre objets de classes différentes de la cns :

$$LD_{EV_\star}(P) = \sum_{\substack{f=1..k, g=1..k \\ f<g}} \sum_{o_a \in C_f, o_b \in C_g,} \sum_{\substack{i \text{ tel que} \\ V_i \in EV_\star}} (lien_i(o_{a_i},o_{b_i}))$$

$$NLD_{EV_\star}(P) = \sum_{\substack{f=1..k, g=1..k \\ f<g}} \sum_{o_a \in C_f, o_b \in C_g,} \sum_{\substack{i \text{ tel que} \\ V_i \in EV_\star}} (1 - lien_i(o_{a_i},o_{b_i}))$$

Ainsi, la séparation des classes d'une cns P est d'autant plus forte que $NLD_{EV_\star}(P)$ (resp. $LD_{EV_\star}(P)$) est élevé (resp. faible).

Notions Additionnelles Nous définissons deux indices additionnels, $M_{EV_\star}(P)$ et $D_{EV_\star}(P)$ qui correspondent respectivement au nombre total de liens et non liens entre objets de même classe de P ($M_{EV_\star}(P) = NLM_{EV_\star}(P) + LM_{EV_\star}(P)$) et au nombre total de liens et non liens entre objets de classes différentes de P ($D_{EV_\star}(P) = NLD_{EV_\star}(P) + LD_{EV_\star}(P)$).

Finalement, nous notons $L_{EV_\star}(O)$ (resp. $NL_{EV_\star}(O)$) le nombre total de liens (resp. de non-liens) entre objets de O : $L_{EV_\star}(O) = LM + LD$ (resp. $NL_{EV_\star}(O) = NLM + NLD$).

Résumé

$L_{EV_\star}(O) + NL_{EV_\star}(O) = \frac{n\times(n-1)}{2} \times card(EV_\star)$

$D_{EV_\star}(P) + M_{EV_\star}(P) = \frac{n\times(n-1)}{2} \times card(EV_\star)$

$M_{EV_\star}(P) = NLM_{EV_\star}(P) + LM_{EV_\star}(P)$

$D_{EV_\star}(P) = NLD_{EV_\star}(P) + LD_{EV_\star}(P)$

$L_{EV_\star}(O) = LM_{EV_\star}(P) + LD_{EV_\star}(P)$

$NL_{EV_\star}(O) = NLM_{EV_\star}(P) + NLD_{EV_\star}(P)$

Ces relations peuvent être synthétisées au sein d'une table de contingence de type comparaison par paires :

	liens	non liens	$Total$
même classes	$LM_{EV_\star}$	$NLM_{EV_\star}$	$M_{EV_\star}(P)$
classes diff.	$LD_{EV_\star}$	$NLD_{EV_\star}$	$D_{EV_\star}(P)$
$Total$	$L_{EV_\star}(O)$	$NL_{EV_\star}(O)$	$\frac{n(n-1)}{2}p$

2.2 Aspect calculatoire

Bâtir cette table de contingence ne nécessite qu'une seule passe sur le jeu de données. Dans le cas de données catégorielles, cela ne requiert que $O(np)$ comparaisons afin de bâtir p tables de contingence (croisant les p variables avec la variable catégorielle virtuelle impliquée par la partition P). Intuitivement, les définitions formelles de $LM_{EV_\star}$, $NLM_{EV_\star}$, $LD_{EV_\star}$ et $NLD_{EV_\star}$ semblent impliquer $O(n^2p)$ comparaisons mais des astuces de calcul permettent de réduire ce nombre de comparaisons, voir l'exemple illustratif ci-dessous. Ce nombre peut atteindre $O(n^2p)$ dans le cas de présence de variables quantitatives et d'utilisation de fonctions $lien_i$ telles que définies dans le cas 2 des exemples illustratifs précédents.

Du point de vue de l'utilisation mémoire, le cas de données catégorielles implique le stockage de p tables de contingence, ce qui correspond à un encombrement mémoire faible ; en cas de présence de données quantitatives, il est nécessaire de stocker la diagonale supérieure d'une matrice $n \times n$ (dans laquelle on stocke le nombre de liens pour chaque paire d'objets), cela implique donc un encombrement mémoire qui peut s'avérer trop important.

EXEMPLE : *Considérons un jeu de données synthétique composé de 4 objets (o_1 = [y,y,n,n], o_2 = [y,y,n,n], o_3 = [n,y,y,y], o_4 = [n,n,y,y]) décrits par 4 variables catégorielles ($EV = \{V_1,V_2,V_3,V_4\}$). Considérons également la partition P suivante : $P = \{C_1,C_2\} = \{\{o_1,o_2\},\{o_3,o_4\}\}$.*

	V_1	V_2	V_3	V_4
1	y	y	n	n
2	y	y	n	n
3	n	y	y	y
4	n	n	y	y

Nous exposons maintenant comment calculer les valeurs des divers indices présentés dans la section précédente. (Nous utilisons ici la fonction Lien telle qu'elle est définie dans l'exemple sur les données catégorielles de la définition 1.)

– *En une unique passe sur le jeu de données, on peut bâtir les tables de contingence croisant la variable catégorielle virtuelle V_A impliquée par la partition P (V_A possède deux modalités a et b qui correspondent respectivement aux classes $\{o_1,o_2\}$ et $\{o_3,o_4\}$) avec les p variables de EV (V_1, V_2, V_3, V_4.)* :

$V_A \setminus V_1$	y	n
a	2	0
b	0	2

$V_A \setminus V_2$	y	n
a	2	0
b	1	1

$V_A \setminus V_3$	y	n
a	0	2
b	2	0

$V_A \setminus V_4$	y	n
a	2	0
b	0	2

– *le calcul de la valeur de chaque indice est alors réalisé à partir de ces tables : si la table de contingence pour une variable V_i est notée:*

	V_{i_1}	...	$V_{i_{m_i}}$	
V_{A_1}	α_{1i1}	...	α_{1im_i}	$\alpha_{1i.}$
...	...	...	...	...
V_{A_z}	α_{zi1}	...	α_{zim_i}	$\alpha_{zi.}$
	$\alpha_{.i_1}$	...	$\alpha_{.im_i}$	n

V_A la variable catégorielle virtuelle à z modalités (associée à P),
V_i une variable exogène à m_i modalités notées $V_{i_j} (j = 1..m_i)$.
α_{lih} le nombre d'objets ayant la valeur V_{i_h} pour V_i et la valeur V_{A_l} pour V_A.
$\alpha_{.i_j} = \sum_{h=1..z} \alpha_{hij}$; $\alpha_{hi.} = \sum_{j=1..m_i} \alpha_{hij}$

– *on peut alors calculer :*

$$LM_{EV_\star}(P) = \sum_{\substack{i = 1..p \\ \text{tel que } V_i \in_E V_\star}} \sum_{j=1..m_i} \sum_{t=1..z} \frac{\alpha_{ti_j}(\alpha_{ti_j}-1)}{2}$$

$$M_{EV_\star}(P) = card(EV_\star) \times \sum_{t=1..z} \frac{card(C_t)(card(C_t)-1)}{2}$$

$$L_{EV_\star}(O) = \sum_{\substack{i = 1..p \text{ tel que} \\ V_i \in EV_\star}} \sum_{j=1..m_i} \frac{\alpha_{.i_j}(\alpha_{.i_j}-1)}{2}$$

$$NLM_{EV_\star}(P) = M(P)_{EV_\star} - LM_{EV_\star}(P)\,;\; LD_{EV_\star}(P) = L_{EV_\star}(O) - LM_{EV_\star}(P)$$

$$D(P)_{EV_\star} = \frac{n(n-1)}{2} \times card(EV_\star) - M_{EV_\star}(P)\,;$$
$$NLD_{EV_\star}(P) = D_{EV_\star}(P) - LD_{EV_\star}(P)$$

– *Pour l'exemple et en considérant $EV_\star = EV$, cela donne : $LM_{EV_\star}(P) = 7$; $M_{EV_\star}(P) = 8$; $L_{EV_\star}(O) = 9$; $NLM_{EV_\star}(P) = 1$; $LD_{EV_\star}(P) = 2$; $D_{EV_\star}(P) = 16$; $NLD_{EV_\star}(P) = 14$*

2.3 Caractérisation Statistique des valeurs de LM et NLD

Il apparaît intuitivement, qu'une cns valide doit être telle que les objets de même classe sont majoritairement reliés par des liens et que les objets de classes différentes sont majoritairement reliés par des non-liens. Ainsi, une cns valide doit présenter de fortes valeurs pour $LM_{EV_\star}$ et $NLD_{EV_\star}$ ce qui implique de faibles valeurs pour $NLM_{EV_\star}$ et $LD_{EV_\star}$ (cela signifie alors une forte homogénéité interne des classes et une forte séparation des classes de la cns). Cependant, la signification de fortes et faibles valeurs n'est elle pas totalement intuitive, nous utilisons donc une approche statistique de manière à déterminer dans quelle mesure des valeurs $LM_{EV_\star}$ et $NLD_{EV_\star}$ exhibées par une cns peuvent être considérées comme significativement élevées.

Faisons l'hypothèse H_0 d'une organisation aléatoire de l'ensemble d'objets (O) selon une partition P en z classes. Nous pouvons déterminer la loi statistique suivie par LM et $NLD_{EV_\star}$ sous cette hypothèse :

– $LM_{EV_\star}(P)$ suit une loi binomiale de paramètres $M_{EV_\star}(P)$ et $\frac{L_{EV_\star}(O)}{L_{EV_\star}(O)+NL_{EV_\star}(O)}$
– $NLD_{EV_\star}(P)$ suit une loi binomiale de paramètres $D_{EV_\star}(P)$ et $\frac{NL_{EV_\star}(O)}{L_{EV_\star}(O)+NL_{EV_\star}(O)}$

Par approximation avec la loi normale suivie d'une opération de centrage-réduction,

on obtient deux indices suivant la loi normale centrée réduite :

- $xv_1^{EV_\star} = \frac{LM_{EV_\star}(P) - M_{EV_\star}(P) \times \frac{L_{EV_\star}(O)}{L_{EV_\star}(O)+NL_{EV_\star}(O)}}{\sqrt{M_{EV_\star}(P) \times \frac{L_{EV_\star}(O)}{L_{EV_\star}(O)+NL_{EV_\star}(O)} \times (1 - \frac{L_{EV_\star}(O)}{L_{EV_\star}(O)+NL_{EV_\star}(O)})}}$
- $xv_2^{EV_\star} = \frac{NLD_{EV_\star}(P) - D_{EV_\star}(P) \times \frac{NL_{EV_\star}(O)}{L_{EV_\star}(O)+NL_{EV_\star}(O)}}{\sqrt{D_{EV_\star}(P) \times \frac{NL_{EV_\star}(O)}{L_{EV_\star}(O)+NL_{EV_\star}(O)} \times (1 - \frac{NL_{EV_\star}(O)}{L_{EV_\star}(O)+NL_{EV_\star}(O)})}}$

Si l'on considère H_1 l'hypothèse alternative à H_0 définie ainsi : "L'ensemble d'objets O est organisé de manière non aléatoire selon une partition P en z classes telles que cette partition représente une cns valide", alors $LM_{EV_\star}$ et $NLD_{EV_\star}$ doivent simultanément exhiber des valeurs exceptionnellement élevées.

3 La Nouvelle Méthode de Sélection de Variables

Nous proposons maintenant une nouvelle méthode efficiente et rapide pour la SdV dans le cadre de l'apprentissage supervisé. Cette méthode de type filtre ne requiert qu'une unique passe sur le jeu de données. Elle utilise un algorithme génétique (AG) et les indices pour l'évaluation/comparaison de la validité de cns (cf. section précédente) au sein d'un processus itératif pour la sélection d'un sous-ensemble de variables. Dans les sections suivantes, nous considérons un problème d'apprentissage caractérisé par un ensemble $O = \{o_1,...,o_n\}$ de n objets décrits par :

un espace de représentation des données (ERD) EV comprenant p variables catégorielles $EV = \{V_1,...,V_p\}$. ($o_i = \{o_{i_1},...,o_{i_p}\}$) ; une variable catégorielle V_A qui représente le concept à apprendre (variable endogène) possédant k modalités. Nous utilisons un problème d'apprentissage synthétique pour illustrer nos développements : $O = \{o_1,o_2,o_3,o_4,o_5,o_6,o_7\}$, $EV = \{V_1,V_2,V_3,V_4\}$, V_A a 3 modalités a,b,c.

	V_1	V_2	V_3	V_4	V_A
o_1	o	o	o	o	a
o_2	o	o	n	o	a
o_3	o	n	o	o	a
o_4	n	o	n	o	b
o_5	n	o	o	o	b
o_6	n	o	n	n	c
o_7	n	n	o	n	c

Hypothèses et Idées Fondamentales Nous donnons maintenant les idées et hypothèses qui constituent les bases de la méthode que nous proposons :

1. ***Hypothèse*** *: Si l'ERD, EV, d'un problème d'apprentissage est tel que le concept à apprendre implique une structure naturelle de l'ensemble des objets O dans cet ERD, alors cela doit permettre un bon processus d'apprentissage.*
2. ***Hypothèse*** *: Une cns valide de l'ensemble des objets O correspond à une structure naturelle de O.*
3. ***Hypothèse*** *: Sur la base de 1 et 2, on peut admettre que si l'ERD EV d'un problème d'apprentissage est tel que le concept à apprendre implique une organisation des objets de O selon une cns valide dans cet espace EV, alors l'ERD EV doit autoriser un bon processus d'apprentissage.*
4. ***Idée*** *: Dans le cadre de la SdV, nous pouvons considérer que l'ERD, $EV_\clubsuit \subseteq EV$, constitué des variables sélectionnées pour l'apprentissage doit être tel que le concept à apprendre implique une organisation des objets de O selon une cns valide dans l'espace $EV_\clubsuit$.*

5. ***Hypothèse*** *: La cns de l'ensemble d'objets O impliquée par le concept à apprendre est composée d'autant de classes qu'il existe de modalités du concept à apprendre, et chaque classe est exclusivement composée d'objets correspondant à la même modalité du concept à apprendre.(Cette cns est notée par la suite P)*
   Pour l'exemple : $P = \{\{o_1,o_2,o_3\},\{o_4,o_5\},\{o_6,o_7\}\}$.
6. ***Idée*** *: Dans le cadre de la SdV, si nous considérons l'ensemble de tous les ERDs potentiels (ces espaces sont les sous espaces non vides de EV), l'ERD que l'on sélectionne finalement (i.e. l'ERD constitué des variables sélectionnées pour l'apprentissage) doit être tel que la cns P apparaît comme la plus valide au sein de ce sous espace de EV .*

Nous montrons maintenant comment utiliser les indices de comparaison de validité de cns en l'associant à un AG pour bâtir une méthode de SdV pour l'apprentissage supervisé basée sur les idées et hypothèses émises.

La Méthode de Base : une méthode exhaustive Nous proposons tout d'abord une fonction permettant de caractériser la validité de cns. Cette fonction ($fit(P,EV_\star)$), basée sur l'observation que P doit présenter de fortes valeurs pour $xv_1^{EV_\star}$ et $xv_2^{EV_\star}$ pour être considérée comme valide dans $EV_\star$, est la suivante :

$$fit(P,EV_\star) = \begin{cases} \sqrt{(\tilde{x_1} - xv_1^{EV_\star})^2 + (\tilde{x_2} - xv_2^{EV_\star})^2} \text{ si } xv_1^{EV_\star} > 0 \text{ et } xv_2^{EV_\star} > 0 \\ 0 \text{ sinon} \end{cases}$$

Elle correspond donc, en quelque sorte, à une distance du point de vue de la validité entre une cns virtuelle particulière (dont les valeurs $xv_1^{EV_\star}$ et $xv_2^{EV_\star}$ seraient respectivement $\tilde{x_1}$ et $\tilde{x_2}$) et la cns P. En fait, dans ce cas, nous fixons $\tilde{x_1} = \tilde{x_2} = très forte valeur$ de manière à conférer à la cns virtuelle particulière l'aspect d'une sorte de cns idéale du point de vue de la validité (ou encore la validité de P dans un espace tel qu'il confère à P une validité idéale). Cette fonction correspond en somme à une distance du point de vue de la validité entre une cns virtuelle idéale du point de vue de la validité et la cns P. Ainsi, plus une partition présente une valeur faible pour cette fonction, plus cela signifie qu'elle constitue une cns valide.

L'idée de base de la méthode est de considérer la cns P et de tester la validité de cette cns dans chaque sous espace $EV_\star$ de EV (le test de validité consistant à déterminer la valeur de $fit(P,EV_\star)$) . L'ERD sélectionné est alors le sous espace impliquant la plus forte validité pour P. Ce processus requiert une unique passe sur les données pour obtenir toutes les tables de contingence utiles (qui ne nécessitent qu'une faible capacité de stockage), $2^p - 1$ calculs pour tester chaque sous espace non vide de EV, et $2^p - 1$ comparaisons pour déterminer le meilleur sous espace (ou l'ensemble des meilleurs sous espaces). Si le nombre de variables p est faible, l'utilisation de cette méthode est envisageable (car réalisable du point de vue calculatoire). Mais pour des nombres de variables un peu plus élevés, l'utilisation de cette méthode n'est pas envisageable du point de vue calculatoire. Nous devons alors adopter une heuristique pour déterminer le meilleur sous espace, ou au moins un bon sous espace, sans pour autant utiliser une phase de test exhaustive et ainsi limiter le coût calculatoire de la méthode. Nous avons choisi d'adopter les algorithmes génétiques (AGs) qui sont connus comme une solution efficace pour la résolution de problèmes combinatoires.

Réduction de la complexité par introduction d'un AG Le problème auquel nous sommes confronté (la découverte d'un bon sous-espace sans pour autant pratiquer une recherche exhaustive) peut effectivement être résolu efficacement par utilisation d'un AG de la manière suivante :
- chaque chromosome de l'AG correspond à un sous espace de EV qui est caractérisé par la présence/absence de variables de EV
- chaque chromosome possède p gènes, chaque gène correspond à l'une des p variables de EV, un gène a une valeur binaire (un gène est codé sur un seul bit) qui code la présence/absence de la variable dans le sous espace de EV codé par le chromosome
- la fonction de fitness de l'AG est la fonction fit proposée préalablement.
- pour le reste, l'AG est utilisé et défini de manière classique.

L'algorithme ci-dessous détaille le fonctionnement de cette méthode de SdV :
1. Données : la cns P, l'ERD EV
2. En une unique passe sur les données bâtir les tables de contingence nécessaires aux calculs des mesures de validité nécessaires à la méthodologie d'évaluation/comparaison de la validité de cns présentée préalablement.
3. Fixer les paramètres de l'AG : *nombre de générations, taille de la population, Proba. de Croisement, Proba. de mutation*
4. Lancer l'AG utilisant la fonction de fitness spécifique définie par la suite.
5. Sélectionner le meilleur sous-espace déterminé par l'AG

4 Evaluation Expérimentale

Présentation de l'Evaluation Expérimentale L'évaluation expérimentale a été réalisée sur 17 jeux de données issus de la collection de l'université de Californie à Irvine sur lesquels ont été menés divers apprentissages mettant en oeuvre 5 méthodes d'apprentissage différentes (ID3, Sipina, C4.5, 1-plus proche voisins et bayésiens naïfs) et utilisant des espaces de représentation respectivement issus d'un processus de SdV préalable réalisé par les algorithmes ReliefF, CFS, MIFS (ces 3 méthodes constituant des méthodes de référence du domaine), et par notre algorithme de SdV, ou encore sans sélection préalable. Ces divers apprentissages ont permis la réalisation d'une étude comparative concernant d'une part le taux d'erreur des divers apprentissages selon la méthode de SdV employée et d'autre part le nombre de variables sélectionnées par chaque méthode de SdV. L'évaluation du taux d'erreur est réalisée pour une 10-cross-validation ainsi que pour cinq 2-cross-validation.

Notons de plus que, (1) la version de CFS utilisée est telle que le critère employé est bien le critère classique et la stratégie de recherche est basée sur un AG ; (2) la version de MIFS employée est la version classique (critère classique et stratégie de recherche gloutonne classique) ; (3) la version de ReliefF employée est telle que : le critère employé est bien le critère à la fois de consistance et contextuel classique ; (4) la stratégie de recherche utilise quant à elle un échantillon d'objets de la taille de l'ensemble des objets du jeu de données ; (5) CFS, MIFS et notre méthode fournissent quant à elles le sous-ensemble optimal de variables (ou un sous-ensemble l'approchant) ; (6) ReliefF fournit la liste des variables classifiées selon leur pertinence (nous avons ensuite étudié cette

liste de valeur afin de déterminer le sous-ensemble de variables apparemment le plus intéressant) ; (7) l'AG utilisé pour CFS et notre méthode est une version élitiste des AGs de base, il est paramètré de la manière suivante : *nombre de générations =2000, taille de la population = 30, Proba. de Croisement = 0.98, Proba. de mutation = 0.3* .

Analyse de l'Evaluation Expérimentale Les résultats des expériences sont regroupés au sein des tableaux 1, 2, 3 qui présentent les résultats généraux suivants :
- Les tableaux 1, 2 permettent d'évaluer le comportement général des diverses méthodes d'apprentissage utilisées lorsqu'elles sont associées aux méthodes de SdV. En effet, ils présentent la valeur moyenne du rapport "taux de succès avec sélection / taux de succès sans sélection" pour chaque méthode d'apprentissage associée à chacune des méthodes de SdV, et ce, soit dans le cadre d'une 10-cross-validation (pour le tableau 1), soit dans le cadre de cinq 2-cross-validation (pour le tableau 2) (la moyenne est calculée sur l'ensemble des 17 jeux de données). Il apparaît que, de manière générale, l'ensemble des méthodes de SdV impliquent l'obtention de taux de succès quasi-équivalent lorsque l'on utilise les variables fournies par ces méthodes ou l'ensemble complet des variables. Ainsi, quelle que soit la méthode d'apprentissage utilisée et quelle que soit la méthode de SdV utilisée, les taux de succès sont corrects et quasiment similaires. On peut toutefois noter un très léger déficit de qualité d'apprentissage pour la méthode d'apprentissage Sipina lorsqu'elle est associée à la méthode de SdV CFS. On peut ainsi conclure que de manière générale ces 4 méthodes de SdV sont presque équivalentes du point de vue de la qualité des apprentissages qu'elles impliquent.

	ID3	C4.5	Sipina	B. Naïfs	1-PPV
Notre Méthode	0.9987	1.0001	0.9842	0.9951	1.0121
MIFS	1.0044	1.0086	0.9961	1.0030	1.0070
CFS	0.9951	0.9935	0.9679	0.9957	0.9955
ReliefF	0.9966	0.9999	0.9936	1.0011	1.0055

TAB. 1 – *Evaluation des Méthodes de SdV pour une 10-Cross-Validation*

	ID3	C4.5	Sipina	B. Naïfs	1-PPV
Notre Méthode	0.9960	1.0046	1.0074	1.0193	1.0042
MIFS	1.0030	1.0086	1.0024	1.0118	1.0078
CFS	0.9863	0.9988	0.9879	1.0351	1.0199
ReliefF	0.9928	1.0014	0.9997	1.0102	1.0107

TAB. 2 – *Evaluation des Méthodes de SdV pour cinq 2-Cross-Validation*

- Le tableau 3 permet l'évaluation de la réduction de la taille de l'ERD impliquée par l'utilisation des méthodes de SdV. Ainsi, il apparaît clairement que l'ensemble de ces méthodes permettent une réduction significative de la taille de l'ERD. De plus, il existe ici des distinctions claires entre les méthodes de SdV : (1) CFS réduit de manière générale très significativement la taille de cet espace puisqu'en moyenne elle ne conserve que 41,4% des variables. Elle constitue la méthode la plus efficace pour la réduction de l'ERD : son apparente plus grande capacité à réduire cet espace n'est mise en défaut que sur quelques rares jeu de données. (2) Notre méthode et MIFS permettent également, en général, de réduire significativement la taille de cet espace puisqu'en moyenne elles ne

conservent respectivement que 56,9% et 62,6% des variables. Elles constituent, derrière CFS les méthodes les plus efficaces pour la réduction de l'ERD. Leur proximité en moyenne sur leur capacité à réduire l'ERD ne reflète cependant pas leurs comportements largement différents : selon le jeu de données, il peut arriver que l'une surpasse fortement l'autre dans sa capacité à réduire l'ERD. On peut ainsi conclure que si notre méthode semble légèrement plus efficace que MIFS de ce point de vue, il est par contre clair que ponctuellement ce résultat peut être inversé. (3) La méthode ReliefF, même si elle permet de réduire l'ERD (74,4% des variables conservées en moyenne), semble cependant en retrait par rapport aux autres méthodes.

	sans SdV	Notre méthode	MIFS	ReliefF	CFS
GERMAN	20	$6^{30\%}$	$3^{15\%}$	$14^{70\%}$	$5^{25\%}$
MUSH.	22	$8^{36.36\%}$	$1^{4.55\%}$	$17^{77.27\%}$	$3^{13.64\%}$
SICK	28	$6^{21.43\%}$	$9^{32.14\%}$	$12^{42.86\%}$	$1^{3.57\%}$
VEHICLE	18	$12^{66.67\%}$	$6^{33.33\%}$	$18^{100\%}$	$10^{55.56\%}$
ADULT	14	$7^{50\%}$	$5^{35.71\%}$	$6^{42.86\%}$	$5^{35.71\%}$
MONKS 3	6	$3^{50\%}$	$6^{100\%}$	$2^{33.33\%}$	$1^{16.67\%}$
FLAGS	28	$14^{50\%}$	$21^{75\%}$	$27^{96.43\%}$	$3^{10.71\%}$
BREAST	9	$8^{88.89\%}$	$9^{100\%}$	$4^{44.44\%}$	$9^{100\%}$
ZOO	16	$12^{75\%}$	$16^{100\%}$	$14^{87.5\%}$	$9^{56.25\%}$
WINE	13	$11^{84.62\%}$	$13^{100\%}$	$11^{84.62\%}$	$9^{69.23\%}$
CANCER	9	$8^{88.89\%}$	$9^{100\%}$	$9^{100\%}$	$9^{100\%}$
PIMA	8	$2^{25\%}$	$4^{50\%}$	$7^{87.5\%}$	$3^{37.5\%}$
WAVE	21	$15^{71.43\%}$	$21^{100\%}$	$19^{90.48\%}$	$15^{71.43\%}$
CONTRA.	9	$2^{22.22\%}$	$2^{22.22\%}$	$2^{22.22\%}$	$5^{55.56\%}$
ION	34	$25^{73.53\%}$	$13^{38.24\%}$	$33^{97.06\%}$	$9^{26.47\%}$
SPAM	57	$25^{43.86\%}$	$51^{89.47\%}$	$57^{100\%}$	$12^{21.05\%}$
HVOTES	16	$10^{62.5\%}$	$11^{68.75\%}$	$14^{87.5\%}$	$1^{6.25\%}$
moyenne	19.29	$10.24^{56.9\%}$	$11.76^{62.6\%}$	$15.65^{74.4\%}$	$6.41^{41.4\%}$

TAB. 3 – *Evaluation des Méthodes de SdV sur 17 jeux de données de la collection de l'UCI : Nombre de variables sélectionnées*$^{\%\ de\ variables\ sélectionées}$

- Du point de vue du coût calculatoire, MIFS, CFS et notre méthode nécessitent un temps de calcul proche avec un avantage toutefois à MIFS qui utilise une stratégie de recherche gloutonne contrairement aux 2 autres méthodes (temps de calcul de l'ordre de quelques secondes à la minute selon les jeux de données). En effet, les AGs sont, en principe, plus lents que les méthodes d'optimisation gloutonnes telle que celle employée dans MIFS. En fait, CFS et notre méthode pourraient être plus rapides si nous remplacions l'AG par une telle méthode d'optimisation (bien que dans ce cas nous pourrions obtenir des résultats de moindre qualité du point de vue de la correction en prédiction, nous ne pensons pas que la réduction de qualité associée soit réellement significative, et envisageons actuellement de tester cette approche...). ReliefF, par contre, implique un temps de calcul plus important (parfois plusieurs minutes) ce qui s'explique par les multiples passes sur le jeux de données que cette méthode implique contrairement aux 3 autres méthodes.

Par manque d'espace nous omettons ici une présentation détaillée des résultats mais indiquons les points les plus intéressants que l'on peut extraire de leur analyse : (1) la tendance générale de taux de correction proches pour les apprentissages réalisés

avec et sans SdV est vérifiée localement ; (2) CFS semble impliquer parfois des déficits importants en terme de correction et notamment lorsqu'elle sélectionne un nombre faible de variables (le cas du jeu de données MONKS 3 par exemple) ; (3) tout comme pour la réduction de l'ERD, la méthode MIFS et la notre sont en général proches mais il arrive ponctuellement que l'une surpasse plus fortement l'autre ; (4) la stabilité des apprentissages est quasiment similaire pour les apprentissages sur un même jeu de données que l'on ait utilisé ou non la SdV et quelle que soit la méthode de SdV employée.

En définitive, selon nous, cette étude expérimentale tend à privilégier l'utilisation de CFS par rapport à MIFS et notre méthode. On peut rejeter l'idée d'employer ReliefF sans trop de soucis. Toutefois, le coût calculatoire faible de CFS, MIFS et notre méthode ainsi que la variabilité "ponctuelle" des résultats (déficit en terme de correction parfois important pour CFS, différentiel en terme de correction et de nombre de variables sélectionnées parfois significatif entre MIFS et notre méthode), semblent plaider en faveur d'une utilisation simultanée de ces 3 méthodes.

5 Conclusion

En résumé, nous proposons, une méthode basée sur l'hypothèse que l'espace de représentation des données doit être tel que le concept à apprendre doit impliquer qu'une cns représentant ce concept soit valide dans cet espace ; ne nécessitant qu'une unique passe sur le jeu de données, et ayant une complexité algorithmique faible ce qui lui confère une rapidité très intéressante ; utilisant un AG et une nouvelle fonction de fitness particulière afin de résoudre le problème combinatoire de la recherche du sous espace de V impliquant la validité la plus forte pour P.

Les évaluations expérimentales ont montré que (1) pour la précision prédictive, notre méthode se comporte, en général, comme les 3 autres méthodes testées (qui constituent des méthodes de référence du domaine) ; (2) pour le nombre de variables sélectionnées, la réduction du nombre de variables due à notre méthode est réelle même si elle est inférieure à celle impliquée par CFS ; (3) que notre méthode est un peu plus lente que MIFS qui est une méthode de sélection extrêmement rapide ; (4) et qu'une utilisation simultanée des méthodes CFS, MIFS et la notre semble réalisable et judicieuse.

Nous pouvons également conclure que (1) le paradigme de sélection sous-jacent à notre méthode est relativement différent de ceux de MIFS et CFS et peut être mieux adapté à certains jeux de données ; (2) notre méthode peut être améliorée du point de vue du coût calculatoire (voir ci dessous) ; (3) on peut aisément modifier la structure de l'AG de manière à pouvoir rechercher non pas l'ensemble "optimal" de variables mais le meilleur ensemble de variables tel qu'il comprenne au plus un nombre fixé de variables, afin de réduire le nombre de variables sélectionnées.

Enfin, bien que l'hypothèse 5 (une classe par modalité du concept à apprendre, cf. page 6) soit forte, notre méthode fournit des résultats de qualité proche ou supérieure à ceux des méthodes existantes. Les travaux futurs seront dirigés vers une amélioration de la méthode, par relaxation de l'hypothèse 5. (la relaxation de cette hypothèse pouvant éventuellement être réalisée par modification de la fonction de fitness utilisée et en donnant notamment plus d'importance à l'aspect séparation des classes ($xv_2^{EV_\star}$) par

rapport à l'aspect homogénéité interne des classes ($xv_1^{EV_\star}$)); une réduction du temps de calcul associé par substitution d'une méthode d'optimisation gloutonne à l'AG.

REMARQUE : La présence de variables quantitatives, si elle n'implique qu'une unique passe sur les données, est cependant handicapante du point de vue de la capacité de stockage nécessaire (besoin de stocker une matrice $n \times n$) et du point de vue de sa complexité calculatoire qui est alors en $O(n^2)$.

Références

[Kira et Rendell, 1992] K. Kira et L.A. Rendell. A practical approach to feature selection. In Morgan Kaufmann, editor, *Proceedings of the Tenth International Conference on Machine Learning*, 1992.

[Kohavi et John, 1997] Ron Kohavi et George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[Liu et Motoda, 1998] H. Liu et H. Motoda. *Feature Extraction, Construction, and Selection : A Data Mining Perspective, Kluwer Academic, Boston, MA*. 1998.

[Kira et Rendell, 1992] K. Kira et L. A. Rendell. The feature selection problem: Traditional methods and a new algorithm. In MIT Press, editor, *Tenth National Conference on Artificial Intelligence*, pages 129–134, 1992.

[Sheinvald *et al.*, 1990] Sheinvald, Dom, Niblack, et Rendell. A modeling approach to feature selection. In *10th Int. Conf. on Pattern Recognition*, 1990.

[Almuallim et Dietterich, 1991] H. Almuallim et T. G. Dietterich. Learning with many irrelevant features. In *Proc. of the 9th National Conf. on Artificial Intelligence (AAAI-91)*, volume 2, pages 547–552, Anaheim, California, 1991. AAAI Press.

[Liu et Setiono, 1996] Huan Liu et Rudy Setiono. A probabilistic approach to feature selection - a filter solution. In *Int. Conf. on Machine Learning*, pages 319–327, 1996.

[Battiti, 1994] R. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Trans. on Neural Networks*, 5:537–550, July 1994.

[Hall, 2000] Mark A. Hall. Correlation-based feature selection for discrete and numeric class machine learning. In *Proc. 17th International Conf. on Machine Learning*, pages 359–366. Morgan Kaufmann, San Francisco, CA, 2000.

[Lallich et Rakotomalala, 2000] S. Lallich et R. Rakotomalala. Fast feature selection using partial correlation for multi-valued attributes. In *Proc. of the 4th European Conf. on Knowledge Discovery in Databases, PKDD 2000*, pages 221–231, 2000.

Summary

Feature selection (FS) enables reduction of the number of features. Due to databases size increase, FS becomes a more and more critical process. Traditionally, FS methods need several accesses to data which may account for a large part of the execution time of FS algorithms. We propose a new efficient and fast method (which needs just one access to data). This method uses a genetic algorithm as well as a clustering validity index.