

Combinaison de classificateurs simples pour une sélection rapide de caractéristiques

Hassan Chouaib*, Florence Cloppet*
Salvatore-Antoine Tabbone**, Nicole Vincent*

*Laboratoire LIPADE, Université Paris Descartes
45, rue des Saints Pères 75006 - Paris, France
prenom.nom@mi.parisdescartes.fr,

**LORIA, Université de Nancy 2, Campus scientifique - BP 239
54 506, Vandoeuvre -les-Nancy Cedex, France
tabbone@loria.fr

Résumé. La sélection de caractéristiques est une technique permettant de choisir les caractéristiques les plus pertinentes, celles adaptées à la résolution d'un problème particulier. Les méthodes classiques présentent certains inconvénients. Par exemple, elles peuvent être trop complexes, elles peuvent faire dépendre les caractéristiques sélectionnées du classificateur utilisé, elles risquent de sélectionner des caractéristiques redondantes. Dans le but de limiter ces inconvénients, nous proposons dans cet article une nouvelle méthode rapide de sélection de caractéristiques basée sur la construction et la sélection de classificateurs simples associés à chacune des caractéristiques. Une optimisation par un algorithme génétique est proposée afin de trouver la meilleure combinaison des classificateurs. Différentes méthodes de combinaison sont considérées et adaptées à notre problème. Cette méthode a été appliquée sur différents ensembles de caractéristiques de tailles variées et construite à partir de la base de chiffres manuscrits MNIST. Les résultats obtenus montrent la robustesse de l'approche ainsi que l'efficacité de la méthode. En moyenne, le nombre de caractéristiques sélectionnées a diminué de 69,9% tout en conservant le taux de reconnaissance.

1 Introduction

Dans de nombreux domaines (vision par ordinateur, reconnaissance de formes, etc.) et dans de nombreuses applications, la résolution des problèmes se base sur le traitement de données extraites à partir des données acquises dans le monde réel, et structurées sous forme de vecteurs. La qualité du système de traitement dépend directement du bon choix de la constitution de ces vecteurs. Mais dans de nombreux cas, la résolution du problème devient presque impossible à cause de la dimensionalité trop importante de ces vecteurs ou des incohérences qui peuvent apparaître dans les données. Par conséquent, il est souvent utile, et parfois nécessaire, de réduire la dimension de l'espace de représentation à une taille plus compatible avec les méthodes de résolution, même si cette réduction peut conduire à une petite perte d'informations. Parfois, la résolution de phénomènes complexes avec des descripteurs de grande taille

Combinaison de classificateurs simples pour une sélection rapide de caractéristiques

peut être gérée en utilisant peu de caractéristiques extraites des données initiales si elles représentent les variables pertinentes pour le problème à résoudre.

Une méthode de réduction de la dimensionnalité est souvent définie comme un processus de prétraitement de données qui permet de supprimer les informations redondantes et bruitées. Avec la multiplication de la quantité de données, la redondance et le bruit dans les informations sont toujours présents. Un type de méthode de réduction de dimensionnalité est la sélection de caractéristiques. Ces méthodes consistent à sélectionner les caractéristiques les plus pertinentes, à partir de l'ensemble des données.

Les méthodes de sélection existantes dans la littérature présentent des limitations à différents niveaux (complexité, interactions entre les caractéristiques, dépendance du classificateur utilisé pour l'évaluation, etc.). Pour limiter ces inconvénients, nous présentons une nouvelle méthode de sélection de caractéristiques. Cette méthode est basée sur la sélection de la meilleure combinaison de classificateurs à partir d'une base de classificateurs simples construits sur chacune des caractéristiques. Cette sélection se fait à l'aide d'un algorithme génétique.

La suite de cet article est organisée comme suit : dans la section 2 nous motivons notre choix pour la sélection de caractéristiques en présentant les limitations des méthodes de base et nous abordons aussi le sujet de la sélection des caractéristiques par un algorithme génétique. Dans la section 3 nous présentons notre méthode de sélection et nous menons dans la section 4 une étude expérimentale approfondie. Enfin, nous concluons sur l'approche et donnons des perspectives à notre travail (section 5).

2 Sélection de caractéristiques : méthodes de base

La sélection de caractéristiques est généralement définie comme un processus de recherche permettant de trouver un sous-ensemble "*pertinent*" de caractéristiques parmi celles de l'ensemble de départ. La notion de pertinence d'un sous ensemble de caractéristiques dépend toujours des objectifs et des critères du problème.

Une méthode de sélection passe généralement par quatre étapes (Liu et Yu (2005)) :

Les deux premières consistent à initialiser le point de départ à partir duquel la recherche va commencer et à définir une procédure de recherche. Une fois la stratégie de recherche définie, les sous-ensembles sont générés, une méthode d'évaluation est définie dans la troisième étape. Les étapes deux et trois se répètent jusqu'à un critère d'arrêt. Ce critère d'arrêt représente la quatrième étape de la méthode. En général, les stratégies de recherche peuvent être classées en trois catégories : exhaustive, heuristique ou aléatoire.

Les méthodes utilisées pour évaluer un sous-ensemble de caractéristiques dans les algorithmes de sélection peuvent être classées en deux catégories principales : *filter* ou *wrapper*.

2.1 Filter

Le modèle "*filter*" est le plus anciennement utilisé pour la sélection de caractéristiques. Le critère d'évaluation utilisé juge la pertinence d'une caractéristique selon des mesures qui reposent sur des données d'apprentissage. Cette méthode est considérée comme une étape de prétraitement (filtrage). L'évaluation se fait généralement indépendamment d'un classificateur (John et al. (1994)).

Le principal avantage des méthodes de filtrage est leur efficacité calculatoire et leur robustesse face au sur-apprentissage. Malheureusement, ces méthodes ne tiennent pas compte des interactions entre caractéristiques et tendent à sélectionner des caractéristiques comportant de l'information redondante plutôt que complémentaire (Guyon et Elisseeff (2003)). De plus, ces méthodes ne tiennent pas compte de la performance des méthodes de classification utilisées après la sélection (Kohavi et John (1997)).

2.2 Wrapper

Le principal inconvénient des approches "*filter*" est d'ignorer l'influence des caractéristiques sélectionnées sur la performance du classificateur à utiliser par la suite. Kohavi et John ont introduit alors le concept "*wrapper*" pour la sélection des caractéristiques (Kohavi et John (1997)). Les méthodes "*wrapper*", appelées aussi méthodes enveloppantes, évaluent un sous-ensemble de caractéristiques par sa performance de classification en utilisant un algorithme d'apprentissage, pour un classificateur donné.

Les sous-ensembles de caractéristiques sélectionnés par cette méthode sont bien adaptés à l'algorithme de classification utilisé, mais ils ne sont pas forcément valides si on change le classificateur. La complexité de l'algorithme d'apprentissage, appliqué pour chaque sous-ensemble de caractéristiques testé, rend les méthodes *wrapper* très coûteuses en temps de calcul.

Le problème de la complexité de cette technique rend impossible l'utilisation d'une stratégie de recherche exhaustive (problème NP-complet). Par conséquent, des méthodes de recherche heuristiques ou aléatoires peuvent être utilisées. Les méthodes enveloppantes sont capables de sélectionner des sous-ensembles de caractéristiques de petite taille, performants pour le classificateur utilisé, mais il existe deux inconvénients majeurs qui limitent ces méthodes :

- La complexité et le temps de calcul nécessaire pour la sélection.
- La dépendance des caractéristiques pertinentes sélectionnées par rapport au classificateur utilisé.

2.3 Algorithme génétique et sélection de caractéristique

Les algorithmes génétiques (AG) constituent une des techniques les plus récentes dans le domaine de la sélection de caractéristiques (Yang et Honavar (1998), Oliveira et al. (2002), Kitoogo et Baryamureeba (2007), Duval et Hao (2010)). L'application d'un AG à la résolution d'un problème nécessite de coder les solutions potentielles à ce problème en des chaînes finies de bits afin de constituer les chromosomes constituant une population formée de candidats. Il s'agit de trouver une fonction sélective permettant une bonne discrimination entre les chromosomes et de définir les opérateurs génétiques.

La fonction de *fitness* dans le cas de sélection de caractéristiques peut être définie en utilisant le modèle *filter* ou *wrapper*. L'évaluation de tous les individus d'une génération à l'aide de la fonction de *fitness* risque de devenir très coûteuse en temps de calcul avec l'augmentation du nombre d'individus. Plus particulièrement, pour une évaluation de type *wrapper*, une construction d'un classificateur pour chacun des individus est indispensable. Pour limiter ce problème nous proposons de construire un classificateur qui peut profiter des avantages des deux approches *filter* et *wrapper*.

- La qualité est associée à chacune des caractéristiques (*filter*).

Combinaison de classificateurs simples pour une sélection rapide de caractéristiques

- La performance du classificateur est optimisée (*wrapper*) en combinant des classificateurs simples sans passer par une phase d'apprentissage d'un classificateur pour chacun des sous-ensembles de caractéristiques.

Cet objectif représente l'idée principale de notre nouvelle méthode de sélection que nous décrivons dans la section suivante.

3 Méthode proposée

Les méthodes de filtrage pour la sélection des caractéristiques présentent des limitations dans leur prise en compte des interactions potentielles entre les caractéristiques. Les méthodes enveloppantes (*wrapper*) souffrent de leur côté d'une complexité très élevée ainsi que d'une dépendance aux classificateurs utilisés pour l'évaluation. Dans le but de limiter ces inconvénients, nous proposons une nouvelle méthode de sélection de caractéristiques basée sur une sélection de classificateurs simples. Le filtrage tire sa rapidité de la prise en compte des caractéristiques individuellement, nous retenons cette idée en construisant un ensemble de classificateurs chacun associé à une caractéristique. La vision globale de la méthode *wrapper* est conservée en considérant un critère de sélection prenant en compte l'ensemble des caractéristiques retenues. Cela est mis en oeuvre dans un AG dont nous détaillerons la fonction de *fitness* dans la section 3.3. Nous tenons ainsi compte des interactions entre caractéristiques. Les caractéristiques associées au sous-ensemble final de classificateurs sélectionnés par notre méthode représentent le sous-ensemble final de caractéristiques.

3.1 Processus de sélection

On dispose d'un ensemble $F = \{f_1, f_2, \dots, f_N\}$ composé de N caractéristiques et un ensemble d'apprentissage $B_{app} = \{X_1, X_2, \dots, X_M\}$ composé de M échantillons (exemples) où chaque $X_i = (f_{i1}, f_{i2}, \dots, f_{iN})$ représente le $i^{\text{ème}}$ exemple. Un exemple est représenté par un vecteur dont les composantes sont les valeurs des caractéristiques (f_i) de dimension N , où N est le nombre total de caractéristiques. On dispose de l'ensemble $Y = \{y_1, y_2, \dots, y_M\}$ des étiquettes des exemples de cette base. Pour un problème de classification bi-classe $y_i \in \{-1, 1\}$

Nous divisons cet ensemble d'apprentissage en deux parties : une base d'apprentissage A qui contient M_A exemples pour construire l'ensemble des classificateurs, une base de validation V qui contient M_V exemples utilisés par l'algorithme génétique.

La figure 1 représente le processus de notre méthode de sélection. Ce processus comporte deux étapes :

- La construction de N classificateurs simples H_i , pour chacun n'est pris en compte que la $i^{\text{ème}}$ caractéristique f_i
- Une sélection sur les classificateurs (H_i) par un algorithme génétique.

La première étape de sélection consiste à constituer un ensemble de classificateurs qui représentent les caractéristiques initiales proposées ensuite comme entrées de l'algorithme génétique. Chaque classificateur est un modèle simple entraîné sur une seule caractéristique. A la fin de cette étape, un ensemble $H = \{H_1, H_2, \dots, H_N\}$ est construit où H_i représente le classificateur appris sur la $i^{\text{ème}}$ caractéristique.

Dans la deuxième étape, on applique un algorithme génétique pour sélectionner de proche en

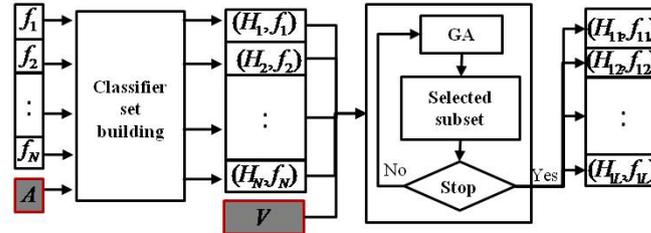


FIG. 1 – Schéma général du processus de sélection

proche un "bon" sous-ensemble de classificateurs. Les caractéristiques associées aux modèles finalement sélectionnés représentent le sous-ensemble de caractéristiques.

3.2 Construction de l'ensemble de classificateurs

Cette étape consiste à construire un ensemble de classificateurs où chaque élément est un classificateur entraîné sur une seule caractéristique. Ce classificateur doit être le plus performant possible.

Les méthodes usuelles sont basées sur un seul seuil de classification. Donnons deux exemples : Un classificateur binaire simple proposé par (Alamdari (2006)) basé sur le calcul d'un seuil statique. Le seuil utilisé est le milieu du segment dont les extrémités sont les isobarycentres des valeurs de la caractéristique des données de chacune des classes. Dans la suite, nous utilisons le nom "*Classif_Alamdari*" pour désigner ce type de classificateur.

Un autre classificateur de ce type est le "*decision stump*" (Iba et Langley (1992)). Ce classificateur fait partie des classificateurs faibles les plus connus, ceux utilisés par l'algorithme "*AdaBoost*".

Nous proposons d'introduire plusieurs seuils de classification pour construire un classificateur. Ces seuils peuvent être les seuils associés aux noeuds d'un arbre de décision (Breiman et al. (1984)) ou bien en utilisant un algorithme "*AdaBoost*" (Freund et Schapire (1995)) construit à partir de différents classificateurs faibles de type "*decision stump*". A partir d'un classificateur de type "*decision stump*" initial h_{s1} , nous définissons un nouveau classificateur de même type en modifiant les poids dans l'ensemble d'apprentissage des exemples qui sont mal classés par h_{s1} pour obtenir un nouveau classificateur $\alpha_1 h_{s1} + \alpha_2 h_{s2}$. La construction d'un classificateur est alors issue de l'application de ce principe de manière itérative. Nous avons fixé à 20 le nombre d'itérations afin d'éviter le problème de sur-apprentissage. Dans la suite, nous utilisons le nom " CS_i " pour désigner ce type de classificateur.

3.3 Critère de sélection et fonction de *fitness*

Dans les méthodes *wrapper* la fonction de *fitness* est liée à la construction d'un nouveau classificateur basé sur les caractéristiques impliquées dans l'individu. Pour contourner la lourdeur de cette approche, il nous fallait trouver un compromis, c'est à dire une grandeur qui pourrait remplacer l'erreur fournie par un nouveau classificateur basé sur les caractéristiques sélectionnées, sans avoir recours à un nouvel apprentissage. Les classificateurs déjà construits

Combinaison de classificateurs simples pour une sélection rapide de caractéristiques

donnent une information plus adaptée au problème que les caractéristiques seules. Chaque classificateur sélectionné doit participer à la décision. Nous notons par $C = (c_1, c_2, \dots, c_N)$ un chromosome où $c_i \in \{0, 1\}$ et S_c est un ensemble $S_c = \{i/c_i = 1\}$. Nous introduisons alors un classificateur construit comme une combinaison des classificateurs.

$$H^c = Comb_{i \in S_c}(H_i) \quad (1)$$

où $comb_{i \in S_c}(H_i)$ est la combinaison des classificateurs présents dans un individu. Ainsi la fonction de *fitness* de l'AG peut s'écrire :

$$fitness = erreur(H^c) \quad (2)$$

L'opérateur *Comb* peut prendre de multiples formes que nous étudions maintenant.

Pour la combinaison des classificateurs, nous avons utilisé des méthodes classiques de combinaison comme le vote majoritaire, le vote majoritaire pondéré, la moyenne, la moyenne pondérée, le médian ainsi que AWFO (**A**ggregation **W**eight-**F**unctional **O**perator) qui a été proposé dans (Dujet et Vincent (1998)). Néanmoins, pour la méthode AWFO, une adaptation est nécessaire pour l'appliquer dans notre cas. Dans la version initiale, il était sous-entendu que l'ensemble des valeurs à agréger appartenait à un intervalle sur lequel la qualité des valeurs relativement à un objectif était monotone. Dans notre cas d'une classification à deux classes, les deux classes, caractérisées respectivement par -1 et 1, ont un statut équivalent, ne permettant pas de définir une valeur distinguée globale significative. Si nous cherchons à combiner des réponses x_i de classificateurs, réponses comprises entre -1 et 1, nous disposons de deux valeurs "optimales". Les valeurs positives ou négatives n'ont pas le même propos, elles indiquent avec plus ou moins de confiance une appartenance à une classe ou à l'autre. Nous avons donc choisi d'agréger séparément les réponses positives et les réponses négatives. Ainsi, nous avons deux valeurs distinguées, -1 pour les valeurs négatives et +1 pour les valeurs positives (figure 2).

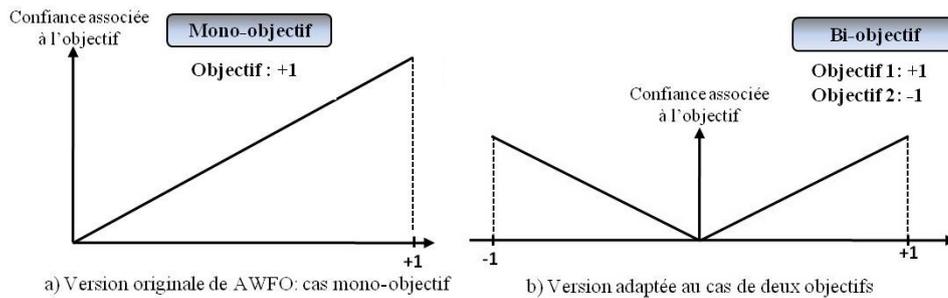


FIG. 2 – Contexte (a) mono et (b) bi-objectif dans la méthode AWFO

La méthode AWFO propose de ne pas considérer exclusivement la réponse du classificateur mais aussi la distribution de toutes les réponses pour réaliser l'agrégation. Le poids de chaque réponse est calculé par la formule suivante :

$$W(x_i) = \frac{d_{cum}(x_i)}{\sum_{signe(x_j)=signe(x_i)} d_j} \quad (3)$$

$$\text{o à } \left\{ \begin{array}{l} d_{cum}(x_i) = \sum_{j \in E_i} d_j \text{ avec } E_i = \{j / (d_j \geq d_i) \ \& \ (\text{signe}(x_j) = \text{signe}(x_i))\} \\ \text{et} \\ d_i = 1 - |x_i| \end{array} \right.$$

4 Expérimentations et validation

Dans cette section nous présentons les expérimentations que nous avons réalisées pour illustrer notre méthode de sélection. Nous commençons par décrire les bases de données que nous avons utilisées ainsi que les descripteurs associés. Nous montrons ensuite les apports des méthodes de combinaison utilisées dans notre approche et nous comparons les résultats avec ceux obtenus par d'autres méthodes de sélection.

4.1 Bases de donnée et protocole d'expérimentation

Pour nos expérimentation nous avons utilisé cinq ensembles de données dans des espaces de dimension variées, construits à partir de la base *MNIST*. C'est une base de chiffres manuscrits (de 0 à 9) isolés construite en 1998 (Lecun et al. (1998)). A chaque chiffre sont associées des images de taille 28 x 28 en 256 niveaux de gris (figure 3). La base *MNIST* est divisée en deux sous-ensembles, un ensemble d'apprentissage de 60 000 exemples et un ensemble de test de 10 000 exemples. Chaque base que nous allons utiliser dans nos expérimentations est



FIG. 3 – Exemples d'images extraites de la base *MNIST*

construite en calculant un descripteur particulier sur les images de la base *MNIST*. Comme descripteurs, nous avons utilisé plusieurs descripteurs de formes parmi lesquels, le descripteur de Zernike (Kim et al. (2000)), le descripteur de Fourier générique (GFD) (Zhang et Lu (2002)), la *R*-signature (Tabbone et Wendling (2003)) et la luminance d'un pixel. Le descripteur GFD est caractérisé par les deux variables *R* et *T* qui sont respectivement la résolution radiale et la résolution angulaire. Différentes valeurs de *R* et *T* ont été testées. Dans la suite nous utilisons les abréviations ZER, GFD1, GFD2, R-sig et pix pour désigner les bases utilisées où GFD1 représente la base construite à partir d'un GFD avec (*R* = 8 et *T* = 12) et GFD2 représente celle construite pour un GFD avec (*R* = 10 et *T* = 15). La première ligne du tableau 1 résume les dimensions des sous-espaces de représentation associés à chacun des descripteurs utilisés.

Nous traitons *a priori* des problèmes à deux classes, dans le cas plus général de *n* classes nous avons choisi de construire des sous-ensembles dans l'ensemble des exemples étiquetés qui permettent d'utiliser une approche "un contre tous". Chaque sous ensemble est associé à une classe. Soit $A = \{A_i\}$, $V = \{V_i\}$ et $T = \{T_i\}$ qui représentent respectivement les bases d'apprentissage, les bases de validation et les bases de test. A_i , V_i et T_i sont construits pour l'utilisation d'une méthode "un contre tous". Chacune des bases A_i et T_i contient $2 \cdot N$ exemples. N

Combinaison de classificateurs simples pour une sélection rapide de caractéristiques

exemples de la classe i et N exemples représentant toutes les autres classes, par contre les V_i ne contiennent que N éléments. $\frac{N}{2}$ exemples de la classe i et $\frac{N}{2}$ exemples représentant toutes les autres classes. Pour la base *MNIST* nous avons $i \in \{0, 1, 2, \dots, 9\}$ et $N = 1000$.

4.2 Résultats

Dans cette section nous montrons les résultats obtenus sur les cinq bases considérées en utilisant des classificateurs de différents types. La construction de l'ensemble initial des classificateurs simples est faite à l'aide d'un des classificateurs décrits dans la section 3.2. Le premier classificateur considéré est le classificateur CS_i qui est basé sur AdaBoost. D'une part il nous permet de trouver plusieurs seuils adaptés selon les exemples d'apprentissage, ceci constitue un avantage par rapport aux classificateurs basés sur un seul seuil, d'autre part la réponse de ce classificateur est numérique, le signe indiquant la classe et le module constituant une sorte de confiance comprise entre 0 et 1. Ce format permet de mettre en oeuvre différentes méthodes de combinaison ce qui n'est pas le cas pour les arbres de décision qui sont capables de trouver plusieurs seuils mais qui fournissent les étiquettes des classes comme réponse finale sans les accompagner d'une grandeur numérique, ce qui limite l'utilisation des méthodes de combinaison au vote majoritaire ou aux votes pondérés. Ensuite nous comparons les résultats obtenus avec les classificateurs CS_i et ceux obtenus avec les autres classificateurs. Nous commençons par le classificateur CS_i .

Après la construction de l'ensemble des classificateurs simples CS_i , et après avoir sélectionné le meilleur sous-ensemble des classificateurs pour les différentes bases, nous avons fait une étude expérimentale sur l'influence de la méthode de combinaison sur la qualité des sous-ensembles sélectionnés.

Le tableau 1 montre le nombre moyen de caractéristiques et le taux moyen de classification avant et après la sélection pour chaque descripteur sur les dix classes de chacune des bases. Le taux de classification pour évaluer la qualité des sous-ensembles trouvés par notre méthode, est calculé à l'aide d'un classificateur SVM entraîné sur les bases d'apprentissage A_i et testé sur les bases T_i . Nous remarquons que les sous-ensembles finaux sont en moyenne 65% plus petits que l'ensemble initial, les taux sont relativement proches et ceci quelque soit le descripteur. En conclusion, nous pouvons dire que nous avons réussi à sélectionner des sous-ensembles de caractéristiques 65% plus petits que la taille des ensembles initiaux mais ayant à peu près la même qualité.

		Zer	GFD1	GFD2	R-sig	Pix
Sans sélection	Nombre de caractéristiques	66	96	150	180	784
	Taux de classification	92.47	92.38	91.97	75.95	97.73
Avec sélection	Nombre de caractéristiques	25	30	46	42	245
	Taux de classification	92.25	92.34	92.08	79.19	97.6

TAB. 1 – Taux de classification et nombre de caractéristiques pour chaque descripteur avant et après la sélection

Nous avons comparé les résultats obtenus à partir de l'ensemble des classificateurs CS_i avec ceux basés sur les mêmes descripteurs mais des ensembles de classificateurs d'un autre

type. Pour les deux classificateurs simples "*Decision stump*" et "*Classif_Alamdari*" nous avons testé les différentes méthodes de combinaison. Pour les arbres de décisions, nous n'avons utilisé que le vote majoritaire et le vote majoritaire pondéré comme méthodes de combinaison. Le tableau 2 montre les meilleurs résultats obtenus pour chacun des ensembles de classifica-

	<i>Zernike</i>	<i>GFD_8×12</i>	<i>GFD_10×15</i>	<i>R-signature</i>	<i>Pixels</i>
<i>CS_i</i>	92.25	92.34	92.1	79.55	97.6
<i>classif_Alamdari</i>	91.5	90.15	90.85	74.15	93.85
<i>decison_stump</i>	92.05	92.05	91.95	79.25	97.45
<i>Arbre de décision</i>	91.95	92.17	91.85	79.35	97.5

TAB. 2 – Comparaison des résultats de plusieurs bases de classificateurs

teurs et chacune des bases. Les résultats montrent que la sélection à partir d'un ensemble de classificateurs *AdaBoost* est meilleure qu'à partir d'autres ensembles de classificateurs. Les résultats sur un ensemble de classificateurs de type "*arbres de décisions*" sont proches de ceux d'*AdaBoost* mais la possibilité d'utiliser plusieurs méthodes de combinaisons avec *AdaBoost* rend ces classificateurs plus avantageux.

Finalement nous avons comparé les résultats de sélection en utilisant les différentes méthodes de combinaison. Le tableau 3 montre les résultats de comparaison en faisant la moyenne sur les dix classes. Dans ce tableau, nous remarquons que les résultats pour les trois méthodes de combinaison *AWFO*, *Moyenne* et *Moyenne pondérée* sont proches pour les différentes bases utilisées et que celles-la sont plus performantes que les méthodes de combinaison par vote majoritaire, vote majoritaire pondérée ou le médian.

	<i>Zer</i>	<i>GFD1</i>	<i>GFD2</i>	<i>R-sig</i>	<i>Pix</i>	<i>Moyenne</i>
<i>AWFO</i>	92.25	92.55	92.08	79.19	97.60	90.74
<i>Moyenne</i>	92.42	91.90	91.95	79.04	97.40	90.54
<i>Moyenne pondérée</i>	92.28	92.34	92.10	79.55	97.55	90.76
<i>Vote majoritaire</i>	91.71	90.55	91.65	77.38	96.80	89.61
<i>Vote pondérée</i>	91.95	91.95	90.98	79.05	97.20	90.22
<i>Médian</i>	92.14	91.06	92.05	78.25	97.5	90.20
<i>Sans sélection</i>	92.47	92.38	91.97	75.95	97.73	90.10

TAB. 3 – Comparaison de différentes méthodes de combinaison

4.3 Comparaison avec d'autres méthodes

Nous avons comparé notre méthode de sélection avec trois autres méthodes existantes. Ces méthodes reposent sur différentes approches d'évaluation (*filter* et *wrapper*). Les trois méthodes considérées sont *Relief*(Kira et Rendell (1992)), *SAC*(Kachouri et al. (2010)) et la troisième est une méthode de *wrapper* classique basée sur une recherche aléatoire en utilisant le même algorithme génétique que notre méthode avec les mêmes paramètres mais la fonction de *fitness* est définie par l'erreur d'un classificateur *SVM*. Le tableau 4 montre les résultats de comparaisons entre notre méthode et les autres méthodes de sélection. Les résultats sont

Combinaison de classificateurs simples pour une sélection rapide de caractéristiques

calculés sur la moyenne des dix classes de la base *MNIST*. Nous pouvons remarquer que notre

	<i>Relief</i>	<i>Wrapper</i>	<i>SAC</i>	<i>Notre méthode</i>
<i>Zernike</i>	89.85	92.61	91.11	92.25
<i>GFD_8×12</i>	90.05	92.55	91.15	92.34
<i>GFD_10×15</i>	90.15	92.01	91.45	92.08
<i>R-signature</i>	73.55	80.05	75.88	79.98
<i>Pixels</i>	95.85	97.68	96.35	97.6

TAB. 4 – Comparaison avec d’autres méthodes pour chaque descripteur

méthode est nettement meilleure que les méthodes *Relief* et *SAC*. Par contre, nos résultats sont très proches de ceux obtenus par la méthode *Wrapper* et cela pour toutes les bases (tableau 4). Par ailleurs, nous avons calculé les temps de sélection de notre méthode et de cette dernière méthode *wrapper_SVM* et cela pour toutes les bases. Le tableau 5 montre les résultats et nous pouvons remarquer que notre méthode est 125 fois plus rapide, dans les pires des cas et 250 fois dans le meilleur cas, pour la base *Pixels*. Elle est aussi capable de sélectionner des sous-ensembles de taille de 6% plus petite dans les pires des cas et 15% dans le meilleur des cas.

		<i>Zer</i>	<i>GFD1</i>	<i>GFD2</i>	<i>R-sig</i>	<i>Pix</i>
<i>Notre méthode</i>	Nb de carac	25	30	46	42	245
	Temps	0.001	0.0015	0.0022	0.0026	0.004
<i>Wrapper_SVM</i>	Temps	0.13	0.22	0.28	0.36	1
	Nb de carac	36	52	65	79	299

TAB. 5 – Comparaison de temps relatifs, de sélection de notre méthode avec la méthode *wrapper*

5 Conclusion et perspectives

Dans cet article, une combinaison de classificateurs simples et un algorithmes génétiques sont utilisés pour définir une nouvelle méthode rapide de sélection. La fonction de *fitness* est basée sur la combinaison des classificateurs simples, chacun étant associé à une seule caractéristique. Plusieurs méthodes de combinaison ainsi que plusieurs classificateurs sont testés et évalués. Nos expérimentations sur différentes bases de données construites à partir de la base *MNIST* montrent que notre méthode a réussi à réduire de 69% le nombre de caractéristiques tout en conservant un bon taux de classification. Le choix de bases associée aux mêmes données du monde réel nous permettent de comparer le comportement des méthodes sur des problèmes de difficulté comparable. Par ailleurs, la méthode proposée est plus rapide de 125 fois dans les pires des cas qu’une méthode de *wrapper* classique. Finalement, la robustesse de l’approche proposée est confirmée. Les futurs travaux seront consacrées à l’étude de la diversité entre les classificateurs à sélectionner afin de minimiser la redondance. Ainsi, une approche multi-objectifs peut être utilisé pour intégrer ce nouvel objectif. Une autre perspective est de

sélectionner les caractéristiques en utilisant une approche hiérarchique appliquée sur plusieurs niveaux (caractéristique et descripteur).

Références

- Alamdari, A. (2006). Variable selection using correlation and single variable classifier methods : Applications. In *Feature Extraction*, Volume 207 of *Studies in Fuzziness and Soft Computing*, pp. 343–358. Springer Berlin / Heidelberg.
- Breiman, L. et al. (1984). *Classification and Regression Trees*. New York : Chapman and Hall.
- Dujet, C. et N. Vincent (1998). Feature selection for classification. *International journal of intelligent system 13*, 131–156.
- Duval, B. et J.-K. Hao (2010). Advances in metaheuristics for gene selection and classification of microarray data. *Briefings in Bioinformatics 11*(1), 127–141.
- Freund, Y. et R. E. Schapire (1995). A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of the Second European Conference on Computational Learning Theory*, London, UK, pp. 23–37. Springer-Verlag.
- Guyon, I. et A. Elisseeff (2003). An introduction to variable and feature selection. *J. Mach. Learn. Res.* 3, 1157–1182.
- Iba, W. et P. Langley (1992). Induction of one-level decision trees. In *Proceedings of the ninth international workshop on Machine learning*, ML92, pp. 233–240.
- John, G. H., R. Kohavi, et K. Pfleger (1994). Irrelevant features and the subset selection problem. In *Machine Learning : Proceedings of the Eleventh International Conference*, pp. 121–129. Morgan Kaufmann.
- Kachouri, R., K. Djemal, et H. Maaref (2010). Adaptive feature selection for heterogeneous image databases. In K. Djemal et M. Deriche (Eds.), *Second IEEE International Conference on Image Processing Theory, Tools 38 ; Applications, 10*, Paris, France.
- Kim, H., J. Kim, D. Sim, et D. Oh (2000). A modified zernike moment shape descriptor invariant to translation rotation and scale for similarity-based image retrieval. In *ICME00*.
- Kira, K. et L. A. Rendell (1992). The feature selection problem : Traditional methods and a new algorithm. In *AAAI*, Cambridge, MA, USA, pp. 129–134. AAAI Press and MIT Press.
- Kitoogo, F. E. et V. Baryamureeba (2007). A methodology for feature selection in named entity recognition. *International Journal of Computing and ICT*, 18–26.
- Kohavi, R. et G. H. John (1997). Wrappers for feature subset selection. *Artif. Intell.* 97, 273–324.
- Lecun, Y., L. Bottou, Y. Bengio, et P. Haffner (1998). Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pp. 2278–2324.
- Liu, H. et L. Yu (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering 17*, 491–502.
- Oliveira, L. S., R. Sabourin, F. Bortolozzi, et C. Y. Suen (2002). Feature selection using multi-objective genetic algorithms for handwritten digit recognition. In *16 th International Conference on Pattern Recognition Volume 1 - Volume 1*, ICPR '02, USA.

Combinaison de classificateurs simples pour une sélection rapide de caractéristiques

Tabbone, S. et L. Wendling (2003). Binary shape normalization using the Radon transform. In *11th International Conference on Discrete Geometry for Computer Imagery - DGCI'2003*, Volume 2886 of *Lecture Notes in Computer Science*, Naples, Italy. Springer.

Yang, J. et V. Honavar (1998). Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems and their Applications* 13(2), 44–49.

Zhang, D. et G. Lu (2002). Shape based image retrieval using generic fourier descriptors. In *Signal Processing : Image Communication* 17, pp. 825–848.

Summary

Feature selection happens to be an important step in any classification process. Its aim is to reduce the number of features and at the same time to try to maintain or even improve the performance of the used classifier. The selection methods described in the literature present some limitations at different levels. For instance, some are too complex to be operated in reasonable time or too dependent on the classifier used for evaluation. Others overlook interactions between features. In this paper, in order to limit these drawbacks, we propose a fast selection method based on a genetic algorithm. Each feature is closely associated with a single feature classifier. The weak classifiers we considered have several degrees of freedom and are optimized on the training dataset. Within the genetic algorithm, the individuals who are classifier subsets are evaluated by a fitness function based on a combination of single feature classifiers. Several combination operators are compared. The whole method is implemented and extensive trials are performed on the MNIST handwritten digits database. Results show how robust is our approach and how efficient is the method. On average, the number of selected features is more than 69.9% smaller than the initial set; nevertheless the recognition rate has not been decreased too much.