



HAL
open science

Sélection de caractéristiques à partir d'un algorithme génétique et d'une combinaison de classifieurs Adaboost

H. Chouaib, Salvatore Tabbone, Oriol Ramos Terrades, F. Cloppet, N. Vincent

► To cite this version:

H. Chouaib, Salvatore Tabbone, Oriol Ramos Terrades, F. Cloppet, N. Vincent. Sélection de caractéristiques à partir d'un algorithme génétique et d'une combinaison de classifieurs Adaboost. Colloque International Francophone sur l'Écrit et le Document - CIFED 08, Oct 2008, Rouen, France. pp.181-186. hal-00334416

HAL Id: hal-00334416

<https://hal.science/hal-00334416>

Submitted on 26 Oct 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Sélection de caractéristiques à partir d'un algorithme génétique et d'une combinaison de classifieurs Adaboost

H.Chouaib¹ – S.Tabbone² – O.Ramos Terrades² – F.Cloppet¹ – N.Vincent¹

¹ Laboratoire CRIP5 (EA 2517), Université Paris Descartes
45, rue des Saints Pères 75006 - Paris

² LORIA-Université Nancy 2, Campus scientifique BP 239
54506 Vandoeuvre-les-Nancy CEDEX, France

{hassan.chouaib, florence.cloppet, nicole.vincent} @math-info.univ-paris5.fr
{ramos, tabbone} @loria.fr

Résumé :

Cet article se situe dans la problématique de la sélection de caractéristiques. Nous proposons une méthode rapide basée sur un algorithme génétique et utilisant la combinaison de classifieurs Adaboost. L'évaluation des individus dans l'algorithme génétique se fait par une fonction de "fitness" basée sur la combinaison de classifieurs entraînés par Adaboost pour chacune des caractéristiques. Cette méthode est implémentée et testée sur la base des images de chiffres manuscrits MNIST et les résultats montrent la robustesse de notre approche ainsi que ses performances. En moyenne le nombre de caractéristiques est divisé par deux sans pour autant diminuer les taux de reconnaissance des images.

Mots-clés : sélection de caractéristiques, algorithme génétique, Adaboost, combinaison de classifieurs.

1 Introduction

La thématique de la sélection de caractéristiques utilisées pour la reconnaissance de forme est un domaine de recherche actif depuis plusieurs décennies. Elle consiste à extraire de l'ensemble des variables explicatives disponibles un ensemble optimal de caractéristiques les plus appropriées pour un système donné.

Les algorithmes génétiques constituent, à notre avis, des méthodes originales qu'on peut utiliser pour la sélection des caractéristiques. Souvent, les méthodes de sélection qui utilisent les algorithmes génétiques sont basées sur la méthode de Wrapper [JOH 94] pour l'évaluation des individus. Ces méthodes [ALT 07, TAN 07, OLI 03] s'appuient sur de multiples apprentissages de classifieurs (réseau de neurones, SVM,...) pour évaluer les individus de la population qui représentent les sous-ensembles des caractéristiques à sélectionner et cet apprentissage nécessite une mise en oeuvre lourde en complexité temporelle.

A la différence de ces méthodes nous proposons une méthode d'évaluation des individus basée sur la combinaison de classifieurs Adaboost entraînés pour chacune des caractéristiques. Plus précisément, à chaque primitive nous associons un classifieur Adaboost et nous mettons en oeuvre une sélection des classifieurs. Dans la mé-

thode que nous proposons, l'apprentissage¹ des classifieurs, pour chaque primitive, se fait en une seule étape avant de commencer l'algorithme génétique. Plus précisément, nous nous basons sur la combinaison de ces classifieurs pour calculer la fonction de fitness dans l'algorithme génétique. Ainsi, nous réduisons considérablement le temps d'entraînement des classifieurs pour chacun des individus. L'approche de sélection proposée présente ainsi des temps d'exécution réduits par rapport à une méthode classique de Wrapper.

La suite de cet article est organisée comme suit : dans la section 2 nous motivons notre choix pour la sélection de caractéristiques. Ensuite nous donnons les concepts de base sur les approches génétiques et la méthode Adaboost pour la combinaison de classifieurs (section 3 et 4). Dans la section 5 nous présentons notre méthode de sélection et nous menons dans la section 6 une étude expérimentale approfondie. Enfin, nous concluons sur l'approche et donnons des perspectives à notre travail (section 7).

2 Sélection de caractéristiques

La sélection de caractéristiques est une technique permettant de choisir les caractéristiques, variables ou mesures les plus intéressantes, pertinentes ou informantes, d'un système donné, pour la réalisation de la tâche pour laquelle il a été conçu. Cette phase est généralement un module important d'un système complexe. Les domaines d'application des techniques de sélection de caractéristiques sont variés par exemple la modélisation, la classification, l'apprentissage automatique (Machine Learning) et l'analyse exploratoire de données (Data Mining).

Dans [KUD 00] les auteurs énoncent une liste de trois objectifs pour réaliser une sélection de caractéristiques pour la classification :

- Réduire la tâche d'extraction de caractéristiques,
- Améliorer la précision du module de classification,
- Améliorer la fiabilité de l'estimation de la performance.

¹Nous utiliserons indifféremment, dans cet article, le terme apprentissage ou entraînement

En fonction du critère d'évaluation utilisé, les algorithmes de sélection de caractéristiques sont classés en deux groupes par John et al [JOH 94] : Filter Model et Wrapper Model. L'approche Filter est réalisée comme un pré-traitement. C'est-à-dire que la sélection se fait sans tenir compte de son influence sur les performances du système. Par opposition, l'approche Wrapper considère cette influence en utilisant le système pour évaluer la qualité du sous-ensemble de caractéristiques sélectionné.

3 Sélection de caractéristiques utilisant des algorithmes génétiques

Les algorithmes génétiques (AG) constituent une des techniques les plus récentes dans le domaine de la sélection de caractéristiques. 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 issus d'une population formée de points de candidats. Il s'agit de trouver une fonction sélective permettant une bonne discrimination entre les chromosomes et de définir les opérateurs génétiques qui seront utilisés.

Des travaux récents dans le domaine de la reconnaissance de formes utilisent les algorithmes évolutionnaires pour la sélection des caractéristiques. Les algorithmes génétiques constituent un type d'algorithmes évolutionnaires qu'on peut utiliser pour résoudre ce genre de problème[SIE 89, ALT 07, TAN 07, OLI 03].

3.1 Concepts de base

Les AG[HOL 92] sont des algorithmes d'optimisation stochastique fondés sur les mécanismes de la sélection naturelle et de la génétique. Leur fonctionnement est extrêmement simple. On part d'une population de chromosomes initiale arbitrairement choisis et on évalue la performance (fitness) relative de chaque chromosome.

Un algorithme génétique est un algorithme itératif de recherche d'*optimum*, il manipule une population de taille constante. La taille constante de la population entraîne un phénomène de compétition entre les chromosomes. Chaque chromosome représente le codage d'une solution potentielle au problème à résoudre, il est constitué d'un ensemble d'éléments appelés gènes, pouvant prendre plusieurs valeurs appartenant à un alphabet non forcément numérique.

A chaque itération, appelée génération, est créée une nouvelle population avec le même nombre de chromosomes. Cette génération consiste en des chromosomes mieux "adaptés" à leur environnement tel qu'il est représenté par la fonction sélective. Au fur et à mesure des générations, les chromosomes vont tendre vers l'*optimum* de la fonction sélective. La création d'une nouvelle population à partir de la précédente se fait par application des opérateurs génétiques que sont : la sélection, le croisement et la mutation.

3.2 Opérations génétiques

Sélection : La sélection est un procédé par lequel un chromosome est recopié dans la nouvelle population en fonction des valeurs de la fonction à optimiser pour ce chromosome. Cela revient à donner aux chromosomes dont

```

Base d'apprentissage de  $p$  échantillons :
 $S = \{ (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n) \}$ 
Initialise les poids de chaque échantillon :
 $w_i^t = \frac{1}{p}$  pour  $i = 1, \dots, P$ 
Faire pour  $t=1, \dots, T$ 
  Entraîner un classifieur faible  $h_t : X \rightarrow \{-1, 1\}$ 
  Calculer l'erreur pondérée  $\epsilon_t$  de  $h_t$ 
 $\epsilon_t = \sum_{i=1}^p w_i^t I(y_i \neq h_t(x_i))$ 
  Calculer les coefficients  $\alpha_t$ 
 $\alpha_t = \frac{1}{2} \log \frac{1-\epsilon_t}{\epsilon_t}$ 
  Mettre à jour les poids
 $w_i^{t+1} = \frac{w_i^t}{Z} \exp\{-\alpha_t y_i h_t(x_i)\}$ 
  Où  $Z$  est une constante de normalisation
  Arrêt si  $\epsilon_t = 0$  ou  $\epsilon_t > 0.5$ 
Fin pour
Résultat :  $h(x) = \sum_{i=1}^t \alpha_i h_t$ 

```

TAB. 1 – Algorithme : Adaboost.

la valeur de fonction de fitness est grande une probabilité plus élevée de contribuer à la génération suivante. Elle peut être mise en oeuvre sous forme algorithmique de différentes façons, la plus connue étant désignée par la méthode de "sélection de la roulette biaisée".

Croisement : Le croisement simple ou à un point de coupe pure consiste premièrement à choisir un couple de chromosomes avec une probabilité p puis dans une deuxième étape, on coupe les chaînes représentatives en une position aléatoire identique chez les deux parents. Cela produit alors deux "segments tête" et deux "segments queue" enfin on permute les deux segments queue des parents pour obtenir ainsi deux enfants qui héritent de quelques caractéristiques de leurs parents.

Mutation : Une mutation est définie comme étant l'inversion d'un bit dans un chromosome. Cela revient à modifier aléatoirement la valeur d'un paramètre. Les mutations jouent le rôle de bruit et empêchent l'évolution de se figer. Elles permettent d'assurer une recherche aussi bien globale que locale, selon le poids et le nombre de bits mutés. De plus, elles garantissent mathématiquement que l'*optimum* global puisse être atteint.

4 Adaboost

Les méthodes de boosting constituent une famille d'algorithmes d'apprentissage automatique qui construisent des modèles (de classification ou de régression) fondés sur la combinaison d'apprenants dits "faibles".

L'algorithme de boosting le plus utilisé s'appelle AdaBoost [FRE 95]. L'idée principale est de définir à chacune de ses étapes, une nouvelle distribution de probabilité *a priori* sur les exemples d'apprentissages en fonction des résultats de l'algorithme à l'étape précédente.

A chaque étape t , l'apprenant cherche une hypothèse h_t qui minimise l'erreur de classification et à la fin de cet algorithme, chaque règle de classification h_t est pondérée par une valeur α_t calculée en cours de route. La classification

d'un nouvel exemple se fait en utilisant la règle :

$$h(x) = \begin{cases} 1 & \text{Si } \sum_{i=1}^t \alpha_i h_i \geq \text{Seuil} \\ 0 & \text{Sinon} \end{cases}$$

Le principe de la méthode Adaboost est rappelé dans le tableau 1.

5 Notre méthode

Nous explicitons dans cette partie notre choix de la fonction de fitness qui constitue l'originalité de notre approche. Nous donnons ensuite les différentes étapes de notre algorithme.

5.1 Choix de la fonction de fitness

L'étape la plus sensible dans un système par évolution génétique est la définition de la fonction de fitness. La réalisation d'un classifieur pour chaque ensemble de caractéristiques rassemblées dans les chromosomes suppose un apprentissage et un développement long qui rendent difficile cette étape de sélection. Nous avons donc tenté de limiter les apprentissages en les limitant à des systèmes simples désignés par des caractéristiques. L'étape de l'AG sera alors appliquée aux classifieurs optimisés pour chaque primitive. La fonction de fitness est alors constituée à partir des qualités de l'ensemble des classifieurs. Donc le classifieur associé à un chromosome est la moyenne de tous les classifieurs Adaboost du chromosome qui sont à 1 :

$$H = \text{Moyenne}(h_i),$$

où h_i est le classifieur Adaboost entraîné pour la i ème primitive sélectionnée. Par exemple si un individu est défini par $X = 1101001$, $H(X)$ sera la moyenne des classifieurs Adaboost h_1, h_2, h_4, h_7 .

Finalement nous définissons la fonction de fitness comme l'erreur de classification de ce classifieur :

$$\text{fitness} = \text{Erreur}(H).$$

L'objectif de notre algorithme génétique est de minimiser cette fonction. Nous n'avons pas pour objectif de construire le meilleur algorithme de reconnaissance mais seulement de maintenir, avec notre technique de sélection de caractéristiques, la qualité d'un système de reconnaissance en se basant sur un nombre réduit de caractéristiques.

5.2 Algorithme

Notre approche de sélection est divisée en deux parties :

1. Entraînement des classifieurs Adaboost pour chaque primitive,
2. Algorithme génétique pour la sélection des caractéristiques.

Nous commençons l'algorithme de sélection par entraîner un simple classifieur Adaboost pour chacune des caractéristiques sur la base d'apprentissage dans une phase indépendante de l'AG. Après la construction de la base des classifieurs nous passons à la phase de sélection en utilisant un algorithme génétique simple. Le but de l'algorithme génétique est de minimiser la fonction de fitness définie dans

la section 5.1 pour trouver un meilleur individu qui représente le sous-ensemble de caractéristiques sélectionnées.

La population initiale est générée aléatoirement ; un chromosome dans cette population est un vecteur binaire de dimension n où n est le nombre de caractéristiques initiales. Le i ème bit du chromosome est à 1 si la i ème primitive est incluse dans la combinaison et à 0 sinon. Une population de 200 individus est générée aléatoirement au début de l'AG. A chaque génération de l'AG une évaluation de chacun des individus de la population est faite en utilisant notre fonction de fitness. La fonction de fitness est la phase la plus importante pour le processus de sélection pour la nouvelle génération qui est déterminée à partir de la valeur de fitness de la génération courante.

Après l'évaluation de la population, plusieurs opérateurs génétiques sont appliqués : le croisement, la mutation et la sélection.

En général un AG se termine quand le critère d'arrêt est satisfait. Dans notre cas le critère de terminaison est lié au nombre de générations (fixé expérimentalement à 50).

Finalement l'ensemble des caractéristiques sélectionnées est donné par le meilleur individu de la dernière génération de l'AG.

6 Expérimentations et discussion

Dans cette section, nous décrivons notre protocole d'évaluation. Cette section est divisée en deux parties. Dans un premier temps nous décrivons notre base de test ainsi que les descripteurs sélectionnés. Ensuite nous donnons les résultats de l'évaluation qui porte sur le nombre de caractéristiques sélectionnées, les taux de reconnaissances et les temps d'exécution.

6.1 Base de données et descripteurs

Pour nos expérimentations nous avons utilisé la base de données des chiffres manuscrits MNIST. Cette base de données contient des images de chiffres (0 ... 9) manuscrits. La taille des images est de 28 x 28 pixels, en niveaux de gris et les chiffres sont centrés au milieu des images. Cette base contient 60000 images.

Avec cette base de données, nous avons défini trois sous-ensembles A, B et C où chaque sous-ensemble contient 1000 images. Nous avons utilisé un sous-ensemble pour l'entraînement des classifieurs Adaboost pour chacune des caractéristiques et les deux autres sous-ensembles comme des bases d'apprentissage et de test pour l'évaluation de l'ensemble des caractéristiques sélectionnées. Dans ce travail, nous nous sommes intéressés aux problèmes de la classification bi-classes. Ainsi pour la phase d'apprentissage, nous avons utilisé 1000 exemples positifs de la classe à apprendre et 1000 exemples, choisis aléatoirement, de toutes les autres classes. Les trois sous-ensembles A, B et C sont définis de la même manière.

Comme descripteurs, nous avons utilisé trois types de descripteurs. Le premier est le descripteur de Zernike (noté ZER)[PRO 92] le deuxième descripteur est la \mathcal{R} -Signature (RS)[TAB 04] et le troisième l'image toute entière en niveaux de gris (pixels). Nous avons généré trois autres descripteurs en combinant ces descripteurs deux à deux i.e

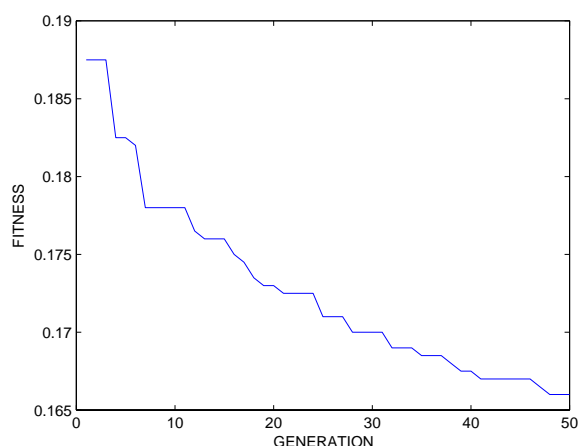


FIG. 1 – Évolution de la valeur fitness du meilleur individu en fonction de la génération.

RS+ZER, ZER+Pixels et RS+Pixels. Le tableau 2 montre le nombre de caractéristiques initiales (première colonne du tableau) pour chacun de ces descripteurs.

6.2 Résultats

Pour tester la performance de l'approche proposée nous avons utilisé les trois sous-ensembles définis dans la section précédente. Nous avons utilisé le sous ensemble A comme une base d'apprentissage pour entraîner les classifieurs Adaboost pour chaque primitive afin de l'utiliser dans l'algorithme génétique. Les deux autres sous ensembles B et C sont utilisés comme base d'apprentissage et test pour évaluer les sous ensembles des caractéristiques sélectionnées par l'algorithme génétique.

Nous avons sélectionné un sous-ensemble de caractéristiques en utilisant notre AG pour chacune des classes de la base MNIST et pour chacun des descripteurs. Les résultats du tableau 2 montrent le taux moyen de caractéristiques sélectionnées pour les dix classes.

| Descripteur | Initiales | Sélectionnées |
|--------------|-----------|----------------|
| RS | 180 | 79 |
| ZER | 47 | 36 |
| Pixels | 1024 | 514 |
| RS + ZER | 227 | 73 + 28 = 101 |
| RS + Pixels | 1204 | 87 + 498 = 585 |
| ZER + Pixels | 1071 | 32 + 492 = 524 |

TAB. 2 – Nombre de caractéristiques avant et après la sélection.

Suivant les résultats de ce tableau nous pouvons remarquer que le nombre des caractéristiques pour la plupart des descripteurs est réduit de plus de 50% tout en conservant les taux de reconnaissance avant et après sélection (cf tableaux 3-5).

D'autre part, nous pouvons remarquer d'après le tableau 2 l'efficacité de notre méthode à sélectionner les caractéristiques les plus pertinentes même après la concaté-

nation de plusieurs descripteurs. Par exemple, pour le descripteur RS+ZER nous pouvons remarquer que l'AG a sélectionné 73 caractéristiques pour RS et 28 caractéristiques pour Zernike alors que pris séparément le nombre de caractéristiques sélectionnées était, respectivement, de 79 et 36.

Une étude expérimentale a été faite sur l'influence des deux paramètres, le nombre de générations et la taille de la population de l'algorithme génétique. En prenant les valeurs 100, 200 et 500 individus pour la taille de population et 30, 50 et 100 comme nombre de génération, nous avons remarqué que l'erreur minimale du meilleur individu est atteinte toujours avant la cinquantième génération et que 200 individus sont suffisants pour trouver cet individu.

La figure 1 montre l'évolution de la valeur de *fitness* du meilleur chromosome de la population courante au fur et à mesure des générations. Cette courbe montre que l'erreur de classification du meilleur individu qui représente le sous-ensemble de caractéristiques à sélectionner diminue avec l'évolution de la population à chaque génération.

Pour évaluer les sous-ensembles de caractéristiques sélectionnées par notre algorithme génétique nous avons utilisé trois classifieurs : SVM, Adaboost et le KPPV (avec $K=5$) et comparer le taux de reconnaissance du système original qui utilise toutes les caractéristiques et avec les sous-ensembles sélectionnés.

Les tableaux 3 à 5 montrent les taux de reconnaissance en utilisant chacun des classifieurs sur chaque descripteur. La comparaison montre que le taux de reconnaissance est resté quasiment le même mais avec un nombre de caractéristiques diminué de 50 %.

| Descripteur | Sans sélection | Après sélection |
|-------------|----------------|-----------------|
| RS | 75.055 | 75.16 |
| ZER | 84 | 82.24 |
| Pixels | 97.72 | 97.712 |
| RS+ZER | 86.315 | 86.7 |
| RS+Pixels | 97.68 | 97.695 |
| ZER+Pixels | 97.77 | 97.77 |

TAB. 3 – Taux de reconnaissance utilisant un classifieur SVM.

| Descripteur | Sans sélection | Après sélection |
|-------------|----------------|-----------------|
| RS | 81.62 | 81.555 |
| ZER | 81.11 | 80.8 |
| Pixels | 94.35 | 94.47 |
| RS+ZER | 81.02 | 82.5 |
| RS+Pixels | 94.59 | 94.72 |
| ZER+Pixels | 94.32 | 94.615 |

TAB. 4 – Taux de reconnaissance utilisant un classifieur Adaboost.

Pour valider la performance de la méthode proposée nous avons fait une validation croisée pour les trois sous-

| Descripteur | Sans sélection | Après sélection |
|-------------|----------------|-----------------|
| RS | 84.55 | 84.555 |
| ZER | 88.28 | 87.56 |
| Pixels | 97.255 | 97.315 |
| RS+ZER | 88.28 | 87.75 |
| RS+Pixels | 97.255 | 97.325 |
| ZER+Pixels | 88.325 | 87.95 |

TAB. 5 – Taux de reconnaissance utilisant un classifieur KPPV ($K = 5$).

ensembles A, B et C en utilisant chaque sous-ensemble à tour de rôle comme base d'apprentissage pour les classifieurs Adaboost, comme base d'apprentissage pour évaluer le sous-ensemble de caractéristiques et comme une base de test. Ces différentes validations croisées ont donné des résultats similaires qui confirment la qualité de notre méthode pour sélectionner les caractéristiques.

Nous avons comparé notre approche de sélection avec une autre qui s'appelle MRMR [LON 05] (Maximum relevant Minimum redundancy). Nous avons choisi cette méthode parce que le nombre de caractéristiques à sélectionner est précisé en avance et nous pouvons ainsi sélectionner le même nombre de caractéristiques que celui de notre méthode.

| Descripteur | MrMr | Notre méthode |
|-------------|-------|---------------|
| RS | 71.78 | 75.16 |
| ZER | 81 | 82.24 |
| Pixels | 96.95 | 97.712 |

TAB. 6 – Comparaison du taux de reconnaissance entre la méthode MrMr et la nôtre.

| Descripteur | AG+Adaboost | Notre méthode |
|-------------|-------------|---------------|
| RS | 94 | 79 |
| ZER | 39 | 36 |
| Pixels | 525 | 514 |

TAB. 7 – Nombre de caractéristiques sélectionnées par un AG+Adaboost (méthode Wrapper) et notre méthode.

Le tableau 6 montre la comparaison entre notre méthode et la méthode MrMr en utilisant le même nombre de caractéristiques et en utilisant un classifieur SVM pour l'évaluation. Cette comparaison montre que les taux de reconnaissance en utilisant les caractéristiques sélectionnées par notre méthode sont meilleurs que ceux qui utilisent la méthode MrMr.

Une autre comparaison est faite entre notre méthode de sélection et une méthode classique de sélection qui utilise un algorithme génétique simple avec une fonction de fitness basée sur la méthode Wrapper et utilisant le classifieur Adaboost pour évaluer chacun des individus de la population. Les résultats sont présentés dans les tableaux 7 et 8 où nous indiquons respectivement le nombre de caractéris-

| Descripteur | AG+Adaboost | Notre méthode |
|-------------|-------------|---------------|
| RS | 75.1 | 75.16 |
| ZER | 83.035 | 82.24 |
| Pixels | 97.95 | 97.712 |

TAB. 8 – Comparaison du taux de reconnaissance entre la méthode Wrapper (AG+Adaboost) et notre méthode avec un classifieur SVM.

| Descripteur | AG+Adaboost | Notre méthode |
|-------------|-------------|---------------|
| RS | 78.125 | 2.656 min |
| ZER | 51.875 | 1 |
| Pixels | 260 | 11.75 |

TAB. 9 – Comparaison du temps d'exécution relatif entre la méthode Wrapper et notre méthode.

tiques sélectionnées dans chacun des cas ainsi que les taux de reconnaissance à partir d'un classifieur SVM.

Finalement nous avons comparé le temps d'exécution de notre méthode et cette méthode classique de Wrapper (cf tableau 9). Nous constatons que l'exécution de notre méthode s'effectue rapidement et le gain varie en fonction des descripteurs d'un facteur 20 à 50.

7 Conclusion

Dans ce papier une nouvelle approche de sélection de primitive est présentée. Cette méthode de sélection est basée sur les algorithmes génétiques et la combinaison de classifieurs. Le calcul de la fonction de fitness se fait par la combinaison des plusieurs classifieurs Adaboost entraînés pour chaque primitive.

A la différence des méthodes de sélection, utilisant les AG avec la méthode de Wrapper pour l'évaluation des individus, qui s'appuient sur un apprentissage de classifieurs pour les individus de la population, notre méthode propose une combinaison des classifieurs entraînés dans une phase indépendante de l'AG.

Cette méthode a été testée sur la base des chiffres manuscrits MNIST en utilisant plusieurs types de descripteur et différents types de classifieurs. Les résultats montrent la robustesse de l'approche proposée qui affiche des taux de reconnaissance quasi similaires mais avec moins de 50 % des caractéristiques et avec des temps d'exécution réduits.

Références

- [ALT 07] ALTUN A. A., ALLAHVERDI N., Neural Network Based Recognition by Using Genetic Algorithm for Feature Selection of Enhanced Fingerprints, *Lecture Notes in Computer Science*, vol 4432, 2007, pp. 467-476.
- [FRE 95] FREUND Y., SCHAPIRE R. E., A decision-theoretic generalization of on-line learning and an application to boosting, *European Conference on Computational Learning Theory*, 1995, pp. 23-37.

- [HOL 92] HOLLAND J. H., *Adaptation in Natural and Artificial Systems*, Ann Arbor, MI, University of Michigan Press, 1992.
- [JOH 94] JOHN G. H., KOHAVI R., , PFLEGER K., Irrelevant Features and the Subset Selection Problem, *International Conference on Machine Learning*, 1994, pp. 121-129.
- [KUD 00] KUDO M., SOMOL P., PUDIL P., SHIMBO M., , SKLANSKY J., Comparison of Classifier-Specific Feature Selection Algorithms, *Lecture Notes in Computer Science*, vol 1876, 2000, pp. 677-686.
- [LON 05] LONG F., DING C., Feature Selection Based on Mutual Information : Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, n° 8, 2005, pp. 1226-1238.
- [OLI 03] OLIVEIRA L., SABOURIN R., BORTOLOZZI F., , SUEN C., A methodology for feature selection using multi-objective genetic algorithms for handwritten digit string recognition, *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 17, n° 6, 2003.
- [PRO 92] PROKOP R., REEVES A., A survey of moment-based techniques for unoccluded object representation and recognition, *Computer Vision and Graphics Image Processing*, vol. 54, n° 5, 1992, pp. 438-460.
- [SIE 89] SIEDLECKI W., SKLANSKY J., A note on genetic algorithms for large-scale feature selection, *Pattern Recognition Letters*, vol. 10, n° 5, 1989, pp. 335-347.
- [TAB 04] TABBONE S., WENDLING L., Recognition of symbols in grey level line drawings from an adaptation of the Radon transform, *In Proceedings of 17th International Conference on Pattern Recognition, Cambridge (UK)*, 2004, pp. 570-573.
- [TAN 07] TAN F., FU X., ZHANG Y., , BOURGEOIS A. G., A genetic algorithm-based method for feature subset selection, *Soft Comput.*, vol. 12, n° 2, 2007, pp. 111-120.