Optimisation multi-objectif pour la sélection de modèles SVM

C. Chatelain, S. Adam, Y. Lecourtier, L. Heutte, Y. Oufella, T. Paquet
Laboratoire LITIS, Université de Rouen, Avenue de l'université,
76800 Saint Etienne du Rouvray, FRANCE

2 juillet 2007

Résumé

Dans cet article, nous proposons une méthode d'optimisation multi-objectif pour la sélection de modèle SVM, en utilisant l'algorithme NSGA-II. Le faux rejet et la fausse acceptation sont les deux critères employés pour trouver les hyperparametres optimaux d'un ensemble de classifieurs SVM. La stratégie proposée est appliquée à une tâche de discrimination chiffre/rejet embarquée dans un système plus global d'extraction d'information dans des documents manuscrits. Nos expérimentations sur une base réelle de chiffres/rejet montrent que notre méthode fournit de meilleurs résultats que les techniques d'optimisation mono-objectif plus classiques telles que l'apprentissage basé sur un critère d'aire sous la courbe ROC. Ces résultats sont dans un second temps validés sur des bases de l'UCI et montrent la supériorité de l'approche proposée.

Mots Clef

Optimisation multiobjectif, SVM, courbe ROC, algorithmes évolutionnaires.

Abstract

In this paper, we propose a multi-objective optimization method for SVM model selection using the well known NSGA-II algorithm. FA and FR rates are the two criteria used to find the optimal hyperparameters of a set of SVM classifiers. The proposed strategy is applied to a digit/outlier discrimination task embedded in a more global information extraction system that aims at locating and recognizing numerical fields in handwritten incoming mail documents. Experiments conducted on a large database of digits and outliers show clearly that our method compares favorably with the results obtained by a state-of-the-art mono-objective optimization technique using the classical Area Under ROC Curve criterion (AUC). A validation of these results on several UCI databases is also proposed, which show the superiority of the proposed approach.

Keywords

Multiobjective optimization, SVM, ROC curve, evolutionnary algorithms.

1 Introduction

Le réglage des hyperparamètres d'un classifieur SVM est une étape cruciale afin d'établir un système de classification efficace. Généralement, au moins deux paramètres doivent être soigneusement choisis : un paramètre relatif au noyau utilisé (γ dans le cas d'un noyau RBF par exemple), et le paramètre de régularisation (habituellement appelé C), qui permet d'intervenir sur le compromis entre l'erreur sur la base d'apprentissage et la complexité du modèle. La recherche de paramètres adaptés est appelée *sélection de modèle* dans la littérature, et ses résultats influent fortement sur les performances du classifieur.

Pendant longtemps, la sélection de modèle a été effectuée par une méthode de type "grid search", où une recherche systématique est mise en œuvre en discrétisant l'espace des paramètres à l'aide d'un pas fixe plus ou moins grand. Il a été montré que ces approches fonctionnaient mal et qu'elles étaient très gourmandes en temps de calcul [19, 23].

Plus récemment, la sélection de modèle a été vue comme une tâche d'optimisation. Dans ce contexte, un algorithme d'optimisation est mis en œuvre afin de trouver l'ensemble d'hyperparamètres qui permettra d'obtenir les meilleures performances en classification. Parmi les algorithmes d'optimisation existants, la méthode de descente de gradient a été souvent employée pour la sélection de modèle SVM (voir [5, 8] par exemple). Cependant, il est bien connu que les méthodes à gradient imposent une dérivabilité du critère d'apprentissage et du noyau SVM par rapport aux paramètres à optimiser, ce qui n'est pas toujours le cas. De plus, les performances des méthodes à descente de gradient dépendent fortement de l'initialisation et peuvent se stabiliser dans des extrema locaux.

Les algorithmes évolutionnaires ont également été employés pour la sélection de modèle SVM afin de surmonter les problèmes mentionnés ci-dessus. On peut citer par exemple les travaux décrits dans [20] ou dans [31] basés sur l'utilisation d'un algorithme génétique (AG), ou l'approche proposée par Friedrichs [16] basée sur l'utilisation de stratégies évolutionnaires. Dans les deux cas, l'algorithme d'optimisation est employé dans le but de maximiser le taux de bonne classification du système.

Cependant, le fait d'utiliser un critère unique en tant qu'objectif pendant le processus d'optimisation constitue selon

nous une limitation. En effet, un critère unique ne suffit pas toujours à décrire les performances d'un système, en particulier dans le cas d'un problème comportant des effectifs de classes déséquilibrés ou des coûts de mauvaise classification asymétriques. Dans ces situations très fréquentes dans des problèmes réels, les probabilités *a priori* des classes et les coûts de mauvaise classification doivent idéalement être considérés pour évaluer les performances du classifieur. Or il est souvent difficile d'estimer ces coûts de mauvaise classification, par exemple quand le classifieur est inclus dans un système plus complexe. Dans le contexte d'un problème à deux classes sans connaissance des coûts, la courbe "Receiver Operating Characteristic" (ROC) introduite dans [3] propose un meilleur critère d'évaluation des performances : elle représente le compromis entre le Faux Rejet (FR) et la Fausse Acceptation (FA), parfois aussi appelé compromis sensibilité/spécificité. Ainsi, pour l'optimisation d'un problème de classification à deux classes, deux critères doivent être minimisés à la place du critère unique et réducteur de bonne classification.

Dans cet article, nous considérons la sélection de modèle SVM comme un problème d'optimisation multi-objectif. L'algorithme d'optimisation évolutionnaire multi-objectif "Non dominated Sorting Genetic Algorithm II" (NSGA-II, voir [11]) est appliqué pour optimiser les hyperparametres d'un SVM en utilisant FA et FR comme critères. Une telle stratégie permet d'obtenir en une seule génération un ensemble de classifieurs proposant chacun un compromis FA/FR optimal. Une fois cet ensemble de classifieurs entrainés, il sera possible de choisir le meilleur du point de vue des contraintes de l'application, à l'aide d'une étape de validation sur une base dédiée.

La stratégie proposée est appliquée à un problème de discrimination chiffre/rejet qui s'inscrit dans un système d'extraction de champs numériques dans des documents manuscrits [7]. Le terme rejet désigne ici tout ce qui n'est pas chiffre : lettre, mot ou fragment de mots, bruit, etc. Comme ce processus de discrimination chiffre/rejet est embarqué dans ce système plus complexe, les coûts de mauvaise classification ne peuvent pas être estimés *a priori*, et le meilleur compromis FA/FR du point de vue des performances globales du système (c'est à dire du point de vue du compromis rappel/précision en extraction des champs numériques) est inconnu. La stratégie proposée permet ainsi de surmonter ce problème en apprenant automatiquement plusieurs classifieurs proposant des compromis intéressants.

La suite de l'article est organisée de la manière suivante : dans la section 2, nous discutons du problème de la sélection de modèle SVM. Nous proposons une brève introduction aux machines à vecteur de support pour en expliciter les paramètres critiques, avant de discuter du choix des critères à optimiser pour la sélection de modèle. La section 3 décrit l'application de l'algorithme NSGA-II à la sélection de modèle SVM. Puis, dans la section 4, nous présentons l'application de la méthode proposée au problème d'extraction de champs numériques et nous la comparons à une méthode d'optimisation mono-objectif reconnue employant comme critère l'aire sous la courbe de ROC (AUC pour Area Under Curve). Afin de valider l'intérêt de notre approche, nous l'appliquons dans un second temps aux bases de l'UCI, et comparons les résultats aux approches de la littérature.

2 Position du problème

2.1 Les classifieurs SVM et leurs hyperparametres pour la sélection de modèle

Comme décrit dans [26], les problèmes de classification avec des coûts de mauvaise classification asymétriques et inconnus peuvent être pris en charge par les SVM en introduisant deux paramètres de pénalités différents C_- et C_+. Dans ce cas, étant donné un ensemble de m exemples d'apprentissage $x_i \in \Re^n$ appartenant à la classe y_i :

$$(x_1, y_1) \dots (x_m, y_m), x_i \in \Re^n, y_i \in \{-1, +1\}$$

la maximisation du lagrangien dual par rapport aux α_i devient :

$$Max_\alpha \Big\{ \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \Big\}$$

$$\text{sous les contraintes : } \begin{cases} 0 \leq \alpha_i \leq C_+ \ \ pour \ \ y_i = -1 \\ 0 \leq \alpha_i \leq C_- \ \ pour \ \ y_i = +1 \\ \sum_{i=1}^{m} \alpha_i y_i = 0 \end{cases}$$

où les α_i représentent les multiplicateurs de Lagrange et $K(.)$ représente la fonction noyau. Dans le cas d'un noyau gaussien (RBF-SVM), $K(.)$ est défini par :

$$K(x_i, x_j) = exp\left(-\gamma \times \|x_i - x_j\|^2\right)$$

Ainsi, dans le cas de coûts de mauvaise classification asymétriques, trois paramètres doivent être déterminés pour réaliser un apprentissage optimal de SVM :
– Le paramètre du noyau, γ pour un SVM-RBF.
– Les paramètres de pénalité introduits ci dessus : C_- et C_+.

Dans la suite de cet article, un "ensemble d'hyperparamètres" désigne donc un ensemble de valeurs données pour γ, C_- et C_+.

2.2 Critères pour la sélection de modèle SVM

Considérer la sélection de modèle comme un processus d'optimisation nécessite le choix d'un ou plusieurs critère(s) à optimiser. Comme indiqué dans l'introduction, la courbe ROC d'un classifieur donné est un meilleur indicateur de performance que le simple taux de bonne classification, particulièrement dans le cas d'un problème de classification à deux classes où les coûts de mauvaise classification sont inconnus. Cependant, employer la courbe ROC

comme indicateur de performance plutôt qu'un critère unique implique l'optimisation de deux critères antagonistes : FA et FR. Ce problème multiobjectif est *a priori* plus difficile que l'optimisation d'un critère unique.

Plusieurs approches ont été proposées dans la littérature pour obtenir la "meilleure courbe ROC possible", en réglant les paramètres intrinsèques d'un classifieur (en l'occurence la position et la valeur des α_i des vecteurs de support dans le cas des SVM). Ce type d'approche est généralement basé sur la réduction des deux critères FA et FR en un seul, tel que l'aire sous la courbe ROC (AUC : Area Under the ROC Curve) ou la F-mesure (FM). C'est le cas des travaux présentés dans [27], où un critère d'AUC est utilisé pour entrainer un classifieur SVM. Dans ces travaux, les supports vecteurs et les α_i associés sont déterminés par minimisation du critère AUC. Cette approche ayant donné de bons résultats, nous les comparons avec les nôtres dans la partie 4, et referrons à [27] pour les détails concernant le calcul de l'AUC et le processus d'optimisation de la méthode. Signalons que les approches reposant sur un critère AUC ont également été proposées dans [13] et [24] dans le cas d'autres classifieurs, et qu'une approche similaire basée sur la F-mesure est proposée dans [25].

Dans tous ces travaux, le but est de réaliser un classifieur optimal au sens du critère de performance choisi (AUC ou FM). Cependant, ces critères de performance ne sont que des indicateurs réducteurs de la courbe ROC. Ainsi, pour une valeur de FA donnée (respectivement FR), les classifieurs entrainés avec ce type de critère ne sont pas capables de produire le classifieur avec la valeur de FR optimale (respectivement FA). Ce qui signifie qu'un classifieur optimisant l'aire sous la courbe ROC ne garantit pas d'être le classifieur optimal pour une valeur donnée de FA (respectivement FR). Cette remarque est illustrée sur la figure 1.

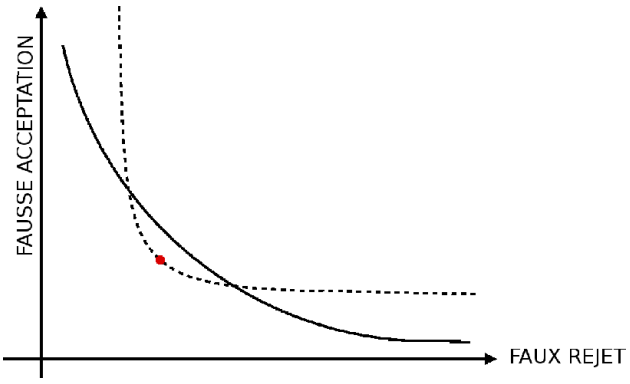

FIG. 1 – La courbe en trait plein minimise l'aire sous la courbe ROC, mais en certains points la courbe en pointillés donne un meilleur compromis FA/FR.

Plutôt que de rechercher un seul classifieur optimal en tous les points de la courbe, nous proposons de rechercher l'ensemble des classifieurs qui proposent les meilleurs points de fonctionnement, c'est à dire un ensemble d'ensembles d'hyperparamètres (γ, C_+, C_-). Ainsi, l'ensemble des points de fonctionnement optimaux de l'ensemble de classifieurs peut être vu comme un "front ROC" (voir figure 2).

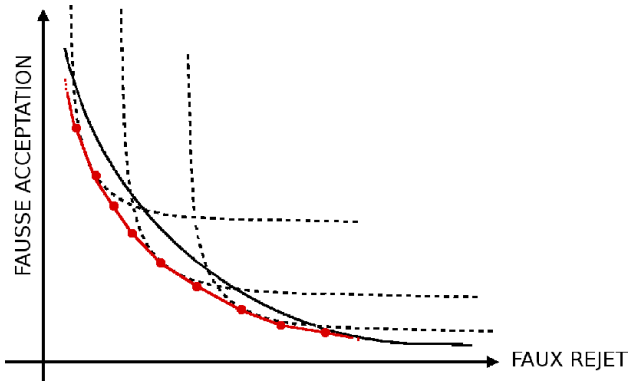

FIG. 2 – Ensemble de compromis FA/FR optimaux obtenus par une population de classifieurs.

La méthode que nous proposons ici est basée sur l'optimisation des compromis FA/FR d'un ensemble de classifieurs à l'aide d'une véritable optimisation multicritères. Cela implique la mise en œuvre d'un algorithme d'optimisation multiobjectif pour la recherche des ensembles d'hyperparamètres, chaque ensemble d'hyperparamètres optimisant un compromis FA/FR. La dimension de l'espace des objectifs étant supérieur à 1, le concept de dominance employé dans le domaine de l'optimisation multiobjectif doit être introduit pour comparer les performances de deux classifieurs.

Le concept de dominance a été proposé par Vilfredo Pareto au 19ème siècle. On dit qu'un vecteur $\overrightarrow{u}$ (dans notre cas, un ensemble donné (C_+, C_-, γ))) domine un autre vecteur $\overrightarrow{v}$ si $\overrightarrow{u}$ n'est pas pire que $\overrightarrow{v}$ pour n'importe lequel des objectifs (FA et FR) et si $\overrightarrow{u}$ est meilleur que $\overrightarrow{v}$ pour au moins un objectif. La notation est la suivante : $\overrightarrow{u} \prec \overrightarrow{v}$. Plus formellement, un vecteur $\overrightarrow{u} = (u_1, u_2, \ldots, u_k)$ domine un vecteur $\overrightarrow{v} = (v_1, v_2, \ldots, v_k)$ si et seulement si :

$$\forall i \in \{1, \ldots, k\}, u_i \leq v_i \wedge \exists j \in \{1, \ldots, k\} : u_j < v_j$$

Étant donné le concept de dominance, l'objectif d'un algorithme d'optimisation multiobjectif est de chercher l'*ensemble de Pareto*, défini comme l'ensemble des solutions dans l'espace des paramètres engendrant des solutions non dominées dans l'espace des objectifs :

$$\text{Ensemble de Pareto} = \left\{ \overrightarrow{u} \in \vartheta / \neg\exists \overrightarrow{v} \in \vartheta, \overrightarrow{f(v)} \prec \overrightarrow{f(u)} \right\}$$

où ϑ désigne l'espace des paramètres où les contraintes sont satisfaites, et $\overrightarrow{f}$ désigne le vecteur d'objectifs. Du point de vue de la sélection de modèle SVM, l'ensemble

de Pareto correspond à la population d'ensembles d'hyperparamètres produisant tous les compromis FA/FR optimaux. Dans l'espace des objectifs, cet ensemble de compromis optimaux est appelé *front de Pareto*. Remarquons que dans le cadre de la sélection de modèles SVM, le front de Pareto pourrait être comparé à la courbe ROC qui décrirait le meilleur ensemble de compromis FA/FR. Dans notre cas, le front de Pareto correspond toutefois aux compromis FA/FR obtenus à l'aide d'un ensemble de classifieurs, alors que la courbe ROC est obtenue à l'aide d'un seul classifieur. Si la comparaison entre notre "front ROC" et une courbe ROC n'est pas théoriquement valide, elle permet toutefois de bien saisir le concept proposé dans cette article.
L'approche proposée cherche donc à approximer l'ensemble optimal de Pareto d'un classifieur SVM à deux classes à l'aide d'une optimisation multiobjectif évolutionnaire. Nous dressons maitenant un bref panorama des méthodes d'optimisation multiobjectif évolutionnaire, et décrivons l'algorithme choisi ainsi que son application à la sélection de modèles SVM.

3 Optimisation multiobjectif évolutionnaire

Nous recherchons l'ensemble de classifieurs SVM décrivant l'ensemble des compromis FA/FR optimaux. Les classifieurs sont paramétrés par les hyperparamètres (C_+, C_-, γ). Du point de vue de l'optimisation multiobjectif, cet ensemble peut être vu comme un ensemble de Pareto. L'ensemble des compromis FA/FR associés à ces classifieurs forme le front que nous recherchons. Les algorithmes évolutionnaires sont bien adaptés à la recherche de ce front car ils sont capables grâce à leur parallélisme implicite de dégager des solutions optimales de façon plus efficace qu'une méthode exhaustive.

3.1 Panorama des approches existantes

Depuis les premiers travaux de [28] au milieu des années 80, un certain nombre d'approches d'optimisation multiobjectif évolutionnaire a été proposé : MOGA [15], NSGA [30], NPGA [18], SPEA [33], NSGA II [11], PESA [9] ou encore SPEA2 [32]. Dans une étude comparative, [21] compare les performances des trois algorithmes les plus populaires : SPEA2, PESA et NSGA-II. Ces trois approches sont élitistes, c'est-à-dire que les meilleures solutions non dominées trouvées sont sauvegardées dans une archive afin d'assurer la préservation de bonnes solutions. Cette étude comparative a été menée sur différents problèmes, avec pour mesure de qualité les deux critères importants pour un algorithme multiobjectif : se rapprocher le plus possible du front de Pareto et obtenir une bonne dispersion des solutions sur ce front. Les résultats de cette étude (qui ont été confirmés dans [32] et [4]) montrent qu'aucun des algorithmes ne domine les autres au sens de Pareto. SPEA2 et NSGA-II offrent des performances similaires en terme de convergence et de diversité. Leur convergence est inférieure à celle de PESA mais la diversité des solutions est meilleure. L'étude montre également que NSGA-II est plus rapide que SPEA2.
Dans le contexte de la sélection de modèles SVM, le calcul des fonctions objectifs prend beaucoup de temps puisqu'il faut entrainer puis évaluer le classifieur pour chaque ensemble d'hyperparamètres. De plus, une bonne diversité des solutions est nécessaire puisqu'on ne connait pas le point de fonctionnement sur le front de Pareto. Nous avons donc choisi l'algorithme NSGA-II. Nous donnons dans la partie suivante une description de cet algorithme.

3.2 NSGA-II

NSGA-II est une version modifiée de l'algorithme NSGA [30]. C'est une approche rapide, élitiste et sans paramètres qui manipule une population de solutions et utilise un mécanisme explicite de préservation de la diversité.
Initialement, une population parent P_0 de N solutions (ou individus) est créée aléatoirement. Cette population est triée sur une base de non-dominance à l'aide d'un algorithme rapide. Ce tri associe un rang de dominance à chaque individu. Les individus non dominés ont un rang de 1 et constituent le front $\mathcal{F}_1$. Les autres fronts $\mathcal{F}_i$ sont ensuite définis récursivement en ignorant les solutions des fronts précédemment détectés. Ce tri est illustré sur la figure 3 (à gauche) dans le cas d'un problème à deux objectifs (f_1,f_2), où pour une population de 16 individus, 3 fronts sont détectés.

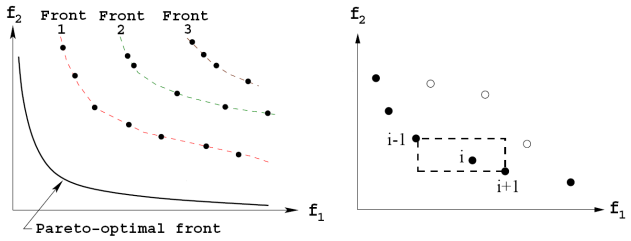

FIG. 3 – Illustration du concept $\mathcal{F}_i$. Sur la figure de droite, les points noirs sont les vecteurs dominants, les points blancs sont dominés.

Les opérateurs de croisement, de recombinaison et de mutation (voir [17] et [11] pour plus de détails) sont ensuite utilisés pour créer une population fille Q_0 de même taille que P_0. À l'issue de cette première étape, l'algorithme est itéré durant M générations. À chaque itération, t désigne le numéro de génération courante, $\mathcal{F}$ désigne le résultat de la procédure de tri, $\mathcal{F}_i$ désigne le i^{eme} front de $\mathcal{F}$, P_t et Q_t désignent respectivement la population et la progéniture à la génération t, et R_t est une population temporaire.
Remarquons que chaque itération de l'algorithme débute avec une fusion des populations parent P_t et fille Q_t pour construire R_t. Cette population de $2N$ solutions est triée à l'aide de la procédure de tri de non-dominance pour construire la population P_{t+1}. Durant cette étape, un autre critère de tri est appliqué pour conserver l'effectif de P_{t+1}

Algorithm 1 Algorithme NSGA-II

$t \leftarrow 0$
while $t < M$ **do**
  $R_t \leftarrow P_t \cup Q_t$
  $\mathcal{F} \leftarrow$ tri-selon-non-dominance (R_t)
  $P_{t+1} \leftarrow \emptyset$
  $i \leftarrow 0$
  while $|P_{t+1}| + |\mathcal{F}_i| \geq N$ **do**
    $P_{t+1} \leftarrow P_{t+1} \cup \mathcal{F}_i$
    assigner-crowding-distance $(\mathcal{F}_i)$
    $i \leftarrow i + 1$
  end while
  Trier $(\mathcal{F}_i, \prec_n)$
  $P_{t+1} \leftarrow P_{t+1} \cup \mathcal{F}_i[1 : (N - |P_{t+1}|)]$
  $Q_{t+1} \leftarrow$ creer-nouvelle-population (P_{t+1})
  $t \leftarrow t + 1$
end while

à une taille constante durant l'intégration des $\mathcal{F}_i$ successifs. Son but est de prendre en compte la contribution des solutions pour la diversité de la fonction objectif dans la population. Ce tri des individus de dominance équivalente est effectué selon une mesure de dispersion appelée *crowding distance* [11]. Cette mesure est basée sur le calcul de la distance moyenne aux deux points de part et d'autre de l'individu considéré selon les deux objectifs (voir figure 3 droite). Plus la surface (resp. volume pour 3 objectifs, hypervolume au delà de 3) autour de l'individu considéré est grande, plus la solution est bonne du point de vue de la diversité. Les solutions de R_t contribuant le plus à la diversité sont ainsi favorisées dans la construction de P_{t+1}. Cette étape est désignée dans l'algorithme 1 par : trier($\mathcal{F}_i$,$\prec_n$), où $\prec_n$ désigne une relation d'ordre partiel basée à la fois sur la dominance et sur la *crowding distance*. Selon cette relation, une solution i est meilleure qu'une solution j si $(i_{rank} < j_{rank})$ ou si $(i_{rank} = j_{rank})$ et $(i_{distance} > j_{distance})$.
Grâce à cet algorithme, la population P_t converge nécessairement vers un ensemble de points du front de Pareto puisque les solutions non dominées sont préservées à travers les générations. De plus, le critère de dispersion (*crowding distance*) garantit une bonne diversité dans la population [11].

3.3 Application de NSGA-II à la sélection de modèles SVM

Dans cette section, nous présentons l'application de l'algorithme NSGA-II au problème de sélection de modèle SVM. Pour cela, deux points particuliers doivent être précisés :

– **Le codage des individus** : rappelons que trois paramètres sont impliqués dans l'apprentissage des classifieurs SVM avec des coûts de mauvaise classification déséquilibrés : C_+, C_- et γ. Ces trois paramètres constituent l'espace des paramètres de notre problème d'optimisation. Chaque individu de la population doit donc coder ces trois valeurs réelles. Nous avons choisi un codage réel des paramètres afin d'être le plus précis possible.
– **La procédure d'évaluation** : chaque individu de la population correspond à un ensemble de trois hyperparamètres. Afin d'évaluer la qualité de cet individu, un apprentissage SVM classique piloté par l'ensemble d'hyperparamètres encodé est lancé. Ce classifieur SVM est ensuite évalué sur une base de validation à l'aide des critères FA et FR.

4 Applications et résultats

Dans cette section, nous appliquons notre approche dans un premier temps sur un problème réel - la discrimination chiffres/rejet d'un système plus important, puis sur les bases de l'UCI afin de valider notre approche et de les comparer sur les mêmes données.

4.1 Discrimination chiffres/rejets

Le travail décrit dans cet article a été appliqué à un système complexe qui vise à extraire les champs numériques (numéro de téléphone, code postal, code client, etc.) dans des images de document manuscrits (voir figure 4). Nous renvoyons à [6, 7] pour une description précise du système.

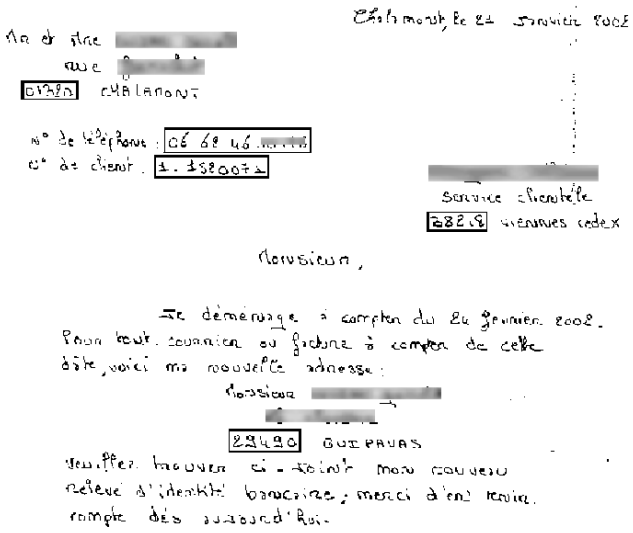

FIG. 4 – Exemple de courrier manuscrit où les champs numériques ont été mis en évidence.

L'approche proposée dans cet article est appliquée à une étape particulière : la discrimination entre chiffres et rejets réalisée en amont d'un classifieur chiffre traditionnel. L'idée est de filtrer un maximum de "rejets évidents" (voir figure 5 a) des autres formes (voir figure 5 b et c) afin d'éviter une identification coûteuse de chiffre quand elle n'est pas nécessaire. Le choix du SVM a été motivé par son efficacité dans un contexte de classification à deux classes. L'objectif est donc de rejeter le plus de rejets possible, tout en acceptant un maximum de chiffres, sachant

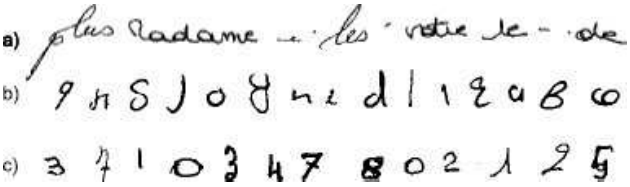

FIG. 5 – Exemples de rejets évidents (a), de rejets “ambigüs” (b), et de chiffres (c).

qu’une autre étape du système concerne le traitement des rejets ambigüs restants. Il est donc évident que le rejet d’un chiffre est beaucoup plus grave que l’acceptation d’un rejet. Cependant, il est difficile de mesurer les conséquences d’une fausse acceptation ou d’un faux rejet sur les résultats finaux du système. Ce problème est donc un bon exemple d’un processus de classification où les coûts de mauvaise classification sont déséquilibrés et inconnus. Nous évaluons maintenant le système proposé.

Le système est entrainé sur une base d’apprentissage de 7129 formes contenant 1/3 de chiffres et 2/3 de rejets, testés sur une base de test de 7149 formes, et évalués sur une base de validation de 5000 formes [1].

Les plages de valeurs des hyperparamètres sont données dans la table 1. Une précision de 10^{-6} est utilisée pour ces paramètres (précision de la machine sur le type flottant). En ce qui concerne les paramètres de NSGA-II, nous avons employé les valeurs classiques proposées dans [11]. Parmi celles-là, notons que la taille de la population a été fixée à 40 afin d’obtenir suffisamment de points sur l’estimation du front de Pareto.

γ	C_-	C_+
$0-1$	$0-500$	$0-5000$

TAB. 1 – Plages de valeurs pour γ, C_- et C_+.

A l’issue de l’optimisation par l’algorithme NSGA-II, on obtient donc une population de classifieurs. La figure 6 montre quelques courbes FA/FR des classifieurs de la population fournie par le système. On peut remarquer que chacune des courbes est localement optimale sur une partie du front. Ces courbes sont la validation expérimentale des courbes théoriques de la figure 2.

Afin d’évaluer notre approche, nous comparons nos résultats à ceux de l’algorithme présenté dans [27], où un classifieur SVM unique est entrainé avec un critère d’aire sous la courbe ROC (voir figure 7). Au vu de ces résultats, remarquons que tous les points obtenus avec le SVM unique entrainé avec un critère AUC sont dominés par au moins un point de la courbe FA/FR obtenue avec NSGA-II. Ce résultat s’explique par le fait qu’avec l’algorithme d’optimisation multiobjectif évolutionnaire, chaque classifieur SVM

[1] Les trois bases respectent les mêmes proportions de chiffres (1/3) et de rejets (2/3).

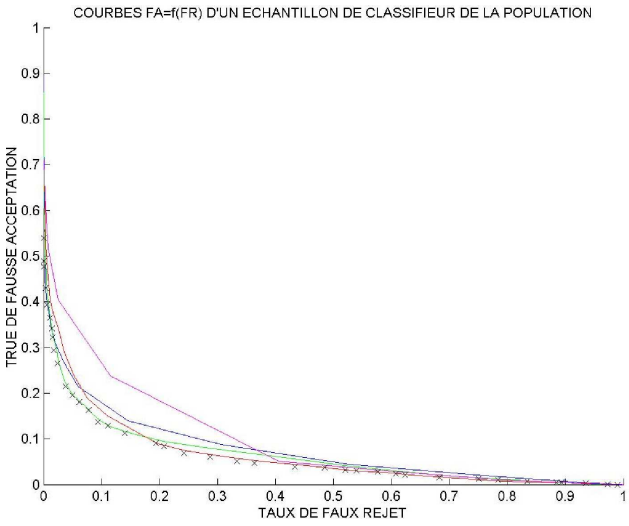

FIG. 6 – Courbes FA/FR de quelques classifieurs SVM de la population. Les croix désigne le front, c’est à dire l’ensemble des meilleurs compromis FA/FR de toutes les courbes.

est spécifique à un point de fonctionnement FA/FR. Le compromis FA/FR est ainsi forcément meilleur que celui obtenu par un classifieur unique entrainé pour optimiser tous les compromis FA/FR possibles.

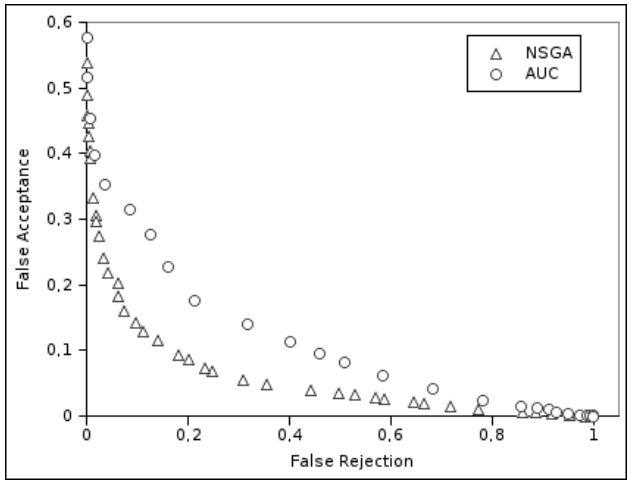

FIG. 7 – Courbe FA/FR obtenue par un classifieur unique entraîné avec un classifieur AUC (AUC), et front obtenu avec notre approche basée sur NSGA-II (NSGA).

Notons que pour un ensemble donné d’hyperparamètres, les paramètres intrinsèques des classifieurs SVM (les positions et poids des vecteurs de support) sont déterminés à l’aide d’une optimisation mono-objectif adaptée à cette tâche. Ainsi, l’algorithme évolutionnaire se concentre sur le choix des hyperparamètres. Cette approche diffère donc des autres travaux mettant en œuvre des algorithmes évolu-

tionnaires pour régler à la fois les paramètres intrinsèques et les hyperparamètres. Nous pouvons en particulier mentionner les travaux de [22, 1, 14, 12]. Tous ces travaux sont limités à des classifieurs très simples (c'est-à-dire possédant un faible nombre de paramètres intrinsèques) à cause de l'impossibilité pour un algorithme évolutionnaire de traiter un nombre élevé de paramètres. Dans un contexte mono-objectif, une telle limitation a été contournée en développant des méthodes spécifiques telles que la maximisation du lagrangien pour les SVM ou la rétropropagation du gradient pour les MLP. Dans un contexte multiobjectif, l'utilisation de la maximisation du lagrangien pour le réglage des paramètres intrinsèques couplée à l'algorithme évolutionnaire en charge des hyperparamètres en nombre plus réduit constitue ainsi une solution intéressante.

4.2 Validation de l'approche sur les bases de l'UCI

Afin de valider notre approche, nous présentons maintenant des résultats sur plusieurs bases disponibles de l'UCI repository. Nous nous sommes limités aux problèmes à deux classes pour lesquels nous avons trouvés des résultats de référence. Les caractéristiques de ces problèmes sont données dans le tableau 2.

problème	# exemples	# attributs
autralian	690	14
wdbc	569	30
breast cancer	699	10
ionosphere	351	34
heart	270	13
pima	768	8

TAB. 2 – Description de quelques problèmes de l'UCI à deux classes.

L'idée est de comparer l'aire sous la courbe ROC obtenue par un classifieur unique avec l'aire sous notre front ROC. Nous insistons à nouveau sur le fait que cette comparaison n'est théoriquement pas correcte[2], mais qu'elle permet de visualiser ce qu'apporte notre approche par rapport aux approches classiques. Nous avons reporté dans le tableau 2 les meilleurs résultats parmi les travaux de Boström [2], Cortes [10], Ferri [13], Rakotomamonjy [27] et Wu [29], que nous comparons à l'aire sous notre front ("AUF" pour Area Under the Front). Notons qu'il existe plusieurs représentations équivalentes de la courbe ROC : $FA = f(FR)$, sensibilité $= f($spécificité$)$, etc. Lorsqu'on parle d'aire sous la courbe, on représente la courbe ROC sous la forme sensibilité en fonction de la spécificité. L'équivalence des deux représentations est obtenue par les relations sensibilité $= 1 - FR$ et spécificité $= FA$.
Remarquons que l'aire sous le front est nettement supérieure à l'aire sous la courbe, quel que soit le problème. Ces résultats montrent clairement que notre approche permet d'atteindre des points de fonctionnement localement beaucoup plus intéressants que les points d'une courbe globalement optimisée. Dans la mesure où dans la plupart des systèmes, un seul point de fonctionnement de la courbe est utilisé, notre approche se révèle particulièrement intéressante.

[2]En effet, il n'est pas correct de comparer un ensemble discret des meilleurs points de plusieurs courbes avec une courbe continue obtenue en faisant varier le seuil de décision d'un seul classifieur.

problème	AUC littérature	ref.	AUF
australian	90.25 ± 0.6	[29]	$\mathbf{96.22 \pm 1.7}$
wdbc	94.7 ± 4.6	[13]	$\mathbf{99.59 \pm 0.4}$
breast cancer	99.13	[2]	$\mathbf{99.78 \pm 0.2}$
ionosphere	$98{,}7 \pm 3.3$	[27]	$\mathbf{99.00 \pm 1.4}$
heart	92.60 ± 0.7	[29]	$\mathbf{94.74 \pm 1.9}$
pima	84.80 ± 6.5	[10]	$\mathbf{87.42 \pm 1.2}$

TAB. 3 – Comparaison de l'aire sous la courbe ROC obtenues dans la littérature et de l'aire sous le front (AUF pour Area Under the Front) obtenue par notre approche.

5 Conclusion

Dans cet article, nous avons présenté une stratégie pour la sélection de modèle SVM pour des problèmes où les coûts de mauvaise classification sont déséquilibrés et inconnus. Pour cela, nous avons proposé une méthode d'apprentissage pour entrainer automatiquement une population de classifieurs proposant chacun des points de fonctionnement localement optimaux. L'approche est basée sur un algorithme évolutionnaire multiobjectif permettant d'optimiser les hyperparamètres des classifieurs. Le système produit ainsi un front ROC dans lequel il est possible de choisir le classifieur convenant le mieux aux contraintes de l'application visée.
Nous avons montré sur un problème réel et sur des bases de référence que cette approche fournissait des résultats intéressants. Soulignons que cette approche simple et générique peut être utilisée avec n'importe quel classifieur comportant des hyperparamètres (KPPV, réseaux de neurones, etc.). Concernant l'application aux SVM, d'autres paramètres (type de noyau, ...) et d'autres objectifs (nombre de vecteurs de support, temps de décision) peuvent également être intégrés dans le processus d'optimisation.
Nos futurs travaux concerneront l'extension de cette approche aux problèmes à plus de deux classes : choix des paramètres à optimiser, choix des objectifs, et méthode de comparaison avec les méthodes traditionnelles.

Références

[1] M. Anastasio, M. Kupinski, and R. Nishikawa. Optimization and froc analysis of rule-based detection schemes using a multiobjective approach. *IEEE Trans. Med. Imaging*, 17 :1089–1093, 1998.

[2] H. Boström. Maximizing the area under the roc curve using incremental reduced error pruning. *ROCML 2005*, 2005.

[3] A. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30 :1145–1159, 1997.

[4] L. Bui, D. Essam, H. Abbass, and D. Green. Performance analysis of multiobjective evolutionary methods in noisy environnments. *APS 2004*, pages 29–39, 2004.

[5] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1) :131–159, 2002.

[6] C. Chatelain. *Extraction de séquences numériques dans des documents manuscrits quelconques*. PhD thesis, december 2006.

[7] C. Chatelain, L. Heutte, and T. Paquet. Segmentation-driven recognition applied to numerical field extraction from handwritten incoming mail documents. *Document Analysis System, Nelson, New Zealand, LNCS 3872, Springer*, pages 564–575, 2006.

[8] K. Chung, W. Kao, C. Sun, and C. Lin. Radius margin bounds for support vector machines with rbf kernel. *Neural comput.*, 15(11) :2643–2681, 2003.

[9] D. Corne, J. Knowles, and M. Oates. The pareto envelope-based selection algorithm for multiobjective optimization. *Parallel problem solving from nature*, pages 839–848, 2000.

[10] C. Cortes and M. Mohri. Auc optimization vs. error rate minimization, 2004.

[11] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist nondominated sorting genetic algorithm for multiobjective optimization : Nsga-ii. In *Parallel problem solving from nature*, pages 849–858. 2000.

[12] R. Everson and J. Fieldsend. Multi-class roc analysis from a multi-objective optimisation perspective. *Pattern Recognition Letters*, page in press, 2006.

[13] C. Ferri, P. Flach, and J. Hernandez-Orallo. Learning decision trees using the area under the roc curve. *Proceedings of the 19th International Conference on Machine Learning*, pages 139–146, 2002.

[14] J. Fieldsend and R. Everson. Roc optimisation of safety related systems. *Proceedings of ROCAI 2004*, pages 37–44, 2004.

[15] C. Fonseca and P. Flemming. Genetic algorithm for multiobjective optimization : formulation, discussion and generalization. *Proceedings of ICGA 1993*, pages 416–423, 1993.

[16] F. Friedrichs and C. Igel. Evolutionnary tuning of multiple svm parameters. *Neurocomputing*, 64 :107–117, 2005.

[17] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[18] J. Horn, N. Nafpliotis, and G. D.E. A niched pareto genetic algorithm for multiobjective optimization. *Proceedings of IEEE-WCCC*, pages 82–87, 1994.

[19] C. Hsu and C. Lin. A simple decomposition method for support vector machine. *Machine Learning*, 46 :219–314, 2002.

[20] C.-L. Huang and C.-J. Wang. A ga-based feature selection and parameters optimization for support vector machine. *Expert systems with application*, 31 :231–240, 2006.

[21] V. Khare, X. Yao, and K. Deb. Performance scaling of multiobjective evolutionary algorithm. *Technical report - SCS, University of Birmingham*, pages 1–70, 2002.

[22] M. Kupinski and M. Anastasio. Multiobjective genetic optimization of diagnostic classifiers with implications for generating receiver operating characterisitic curves. *IEEE Trans. Med. Imaging*, 8 :675–685, 1999.

[23] S. Lavalle and M. Branicky. On the relationship between classical grid search and probabilistic roadmaps. *International Journal of Robotics research*, 23 :673–692, 2002.

[24] M. C. Mozer, R. Dodier, M. D. Colagrosso, C. Guerra-Salcedo, and R. Wolniewicz. Prodding the roc curve : Constrained optimization of classifier performance. *NIPS*, pages 1409–1415, 2002.

[25] D. Musicant, V. Kumar, and A. Ozgur. Optimizing f-measure with support vector machines. *FLAIRS Conference*, pages 356–360, 2003.

[26] E. Osuna, R. Freund, and F. Girosi. *Support vector machines : Training and applications*. 1997.

[27] A. Rakotomamonjy. Optimizing auc with support vector machine. *European Conference on Artificial Intelligence Workshop on ROC Curve and AI*, pages 469–478, 2004.

[28] J. Schaffer and J. Grefenstette. Multiobjective learning via genetic algorithms. *IJCAI*, pages 593–595, 1985.

[29] S. W. Shaomin. A scored auc metric for classifier evaluation and selection. *ROCML 2005*, 2005.

[30] N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionnary Computation*, pages 221–248, 1994.

[31] C.-H. Wu, G.-H. Tzeng, Y.-J. Goo, and W.-C. Fang. A real-valued genetic algorithm to optimize the parameters of support vector machine for predicting bankruptcy. *Expert systems with applications Article in Press*, 2006.

[32] E. Zitzler, M. Laumanns, and L. Thiele. Spea2 : Improving the strength pareto evolutionary algorithm. *Technical report - Swiss Federal Institute of Technology*, 2001.

[33] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms : A comparison case study and the strength pareto approach. *Evolutionary Computation*, pages 257–271, 1999.