Apprentissage multiclasse en environnement incertain

Simon Bernard*, Clément Chatelain*, Sébastien Adam*, Robert Sabourin**

*Université de Rouen, LITIS EA 4108, BP 12 - 76801 Saint-Étienne du Rouvray, France
**Laboratoire d'Imagerie, de Vision et d'Intelligence Artificielle
École de Technologie Supérieure, Université du Québec, Montreal, Canada

Résumé. Dans cet article, nous abordons le problème de la classification multiclasses dans le contexte particulier où les coûts de mauvaise classification sont déséquilibrés den fonction des classes et sont inconnus lors de l'apprentissage mais disponibles en prédiction. La méthode proposée s'appuie sur des ensembles de classifieurs, chacun spécialisé à des contextes de coûts particuliers. Pour cela, elle combine une procédure d'optimisation multiobjectifs avec une décomposition par paires de classes, afin de réduire la complexité computationnelle. Les prédictions sont ensuite obtenues via la sélection du classifieur le plus adapté aux coûts, une fois que ceux-ci sont connus. Les premiers résultats obtenus montrent que cette méthode est efficace et qu'elle permet de traiter des problèmes avec un grand nombre de classes.

1 Introduction

Pour de nombreux problèmes de classification, tels que le diagnostic médical par exemple, les erreurs de classification ne sont pas toutes aussi coûteuses les unes que les autres. Prédire qu'un patient est sain alors qu'il présente en réalité une pathologie est plus préjudiciable que l'erreur inverse. Dans de telles situations, il est important d'adapter l'apprentissage du classifieur à ces coûts de mauvaise classification — que l'on appelle aussi *l'environnement* — déséquilibrés pour que les bonnes décisions sur certaines classes soitent privilégiées, parfois au détriment des autres (Zhou et Liu (2010)). Cependant, il est nécessaire pour cela de connaître ces coûts dès la phase d'apprentissage, ce qui est rarement le cas avec les problèmes réels. Et lorsqu'ils sont connus, ces coûts sont bien souvent amenés à évoluer au cours du temps, auquel cas il est souhaitable de pouvoir adapter le classifieur sans avoir à reprendre entièrement l'apprentissage. Prendre en considération cette incertitude dans le processus d'apprentissage est donc un enjeu majeur.

La plupart des méthodes de la littérature répondent à ce problème en adaptant les paramètres de décision. Supposons un problème à K classes $\{\omega_i\}_{i=[1..K]}$, associé à une matrice de coûts de mauvaise classification $\mathcal{C} = \{c_{ij}\}, \forall i,j = \{1,...,K\}$. Typiquement, un classifieur entrainé sur un tel problème fournit en sortie K scores d'appartenance $h(\omega_i, \mathbf{x})$, pour toute donnée à classer $\mathbf{x}$. La prédiction est alors obtenue à l'aide de la fonction de décision suivante : $\arg\max_{i=1,...,K} \; w_i h(\omega_i|\mathbf{x})$, où $\mathbf{w} = [w_1, w_2, \ldots, w_K]$ sont des paramètres de décision qui permettent d'adapter la décision à l'environnement, quel que soit le classifieur utilisé en amont. L'idée est donc d'effectuer un apprentissage de façon traditionnelle et d'optimiser les valeurs

de $\mathbf{w}$ une fois que les c_{ij} sont disponibles en phase de validation. Nous voyons trois principaux inconvénients à ce type d'approches : (i) l'évaluation des combinaisons possibles de $w_i, \forall i = \{1, \ldots, K\}$ devient très vite complexe et fastidieuse pour des valeurs de K relativement grandes, (ii) les efforts de calculs pour l'optimisation de ces paramètres sont réalisés en prédiction, quand les c_{ij} sont connus ; cela implique de recommencer toute la procédure d'optimisation chaque fois que les coûts changent et (iii) seuls les w_i permettent d'adapter les prédictions à l'environnement alors qu'il nous paraît plus pertinent d'adapter le classifieur, à l'aide d'hyperparamètres par exemple.

Pour pallier ces inconvénients, un autre type de méthode utilise des ensembles de classifieurs, chacun optimisé pour des coûts particuliers, afin de concentrer les efforts de calcul en phase d'apprentissage et de pouvoir choisir ensuite le classifieur le plus adapté une fois que les coûts sont connus (Chatelain et al. (2010); Levesque et al. (2012)). Pour ce faire, les approches basent l'apprentissage de ces classifieurs sur des objectifs multiples et concurrents, chacun d'eux représentant un type d'erreur de classification. Dans le cas de la classification à 2 classes par exemple, le *taux de faux positifs* (TFP) et le *taux de faux négatifs* (TFN) sont les deux critères à optimiser simultanément. Ce sont les objectifs utilisés par la méthode du Front ROC proposée dans Chatelain et al. (2010), pour laquelle l'optimisation est réalisée à l'aide d'un algorithme évolutionnaire multiobjectif. A l'issue de cette optimisation, on obtient une population de classifieurs, chacun proposant un compromis TFP/TFN qui lui est propre, et donc une solution proche de l'optimalité pour des coûts donnés.

Ces approches se sont montrées efficaces pour résoudre les problèmes de classification binaire en environnement incertain. Cependant, leur adaptation à la classification multiclasse est délicate, car le nombre d'objectifs devient vite important pour des K raisonnablement grands ($K \times (K-1)$ objectifs). De sorte qu'aujourd'hui, il n'existe pas d'applications de ces approches à la classification multiclasse avec $K > 3$. Nous proposons dans cet article une méthode qui permet de résoudre des problèmes de classification multiclasse avec coûts de mauvaise classification inconnus ou évolutifs, quel que soit le nombre de classes.

2 Fronts ROC pour la classification multiclasse

L'idée clé de la méthode que nous proposons est de combiner le principe de Front ROC avec une décomposition en paires de classes. De cette façon, le problème multiclasse est transformé en $\frac{K(K-1)}{2}$ sous-problèmes binaires, chacun associé à deux objectifs uniquement. La procédure que nous expliquons ci-dessous est illustrée par la figure 2, qui reprend l'ensemble des étapes des phases d'apprentissage et de prédiction.

Comme dans Chatelain et al. (2010), nous utilisons pour ces travaux des classifieurs de types SVM avec noyau RBF. La raison à cela est que, outre le fait qu'ils soient parmi les classifieurs les plus efficaces de la littérature, ils proposent des hyperparamètres, notés C_+ et C_-, qui permettent de contrôler l'influence de chacune des deux classes sur l'apprentissage, et donc de naturellement prendre en compte des coûts de mauvaise classification déséquilibrés.

L'algorithme d'optimisation est utilisé pour faire évoluer une population de SVM, chacun d'eux étant représenté par ses valeurs d'hyperparamètres et évalués via ses TFP et TFN. Pour des raisons de fiabilité, ces deux valeurs sont évaluées via une validation croisée avec 5 découpages. À la fin de cette phase d'évolution, l'algorithme fournit un ensemble de SVM uniformément répartis le long du Front ROC tel que représenté sur la partie droite de la figure 1.

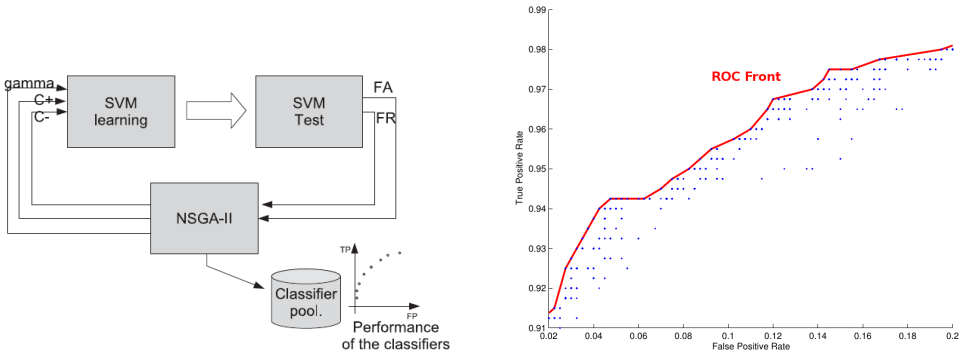

FIG. 1: Front ROC en classification binaire. Gauche : illustration de la méthode, utilisée avec des classifieurs de types SVM (tiré de Chatelain et al. (2010)) ; Droite : représentation du Front dans l'espace ROC qui permet de visualiser chaque classifieur via ses TVP (*i.e.* 1-TFN) et TFP.

Chaque point de ce front sur cette figure est un SVM entraîné avec des hyperparamètres qui lui sont propres et représenté par ses valeurs d'objectifs. On constate alors que l'on dispose d'une large gamme de compromis TFP/TFN, répondant à autant de situations de coûts différentes.

Dans notre schéma multiclasse, cette procédure d'optimisation multiobjectifs est appliquée à chacun des sous-problèmes binaires. Par conséquent, à l'issue de la phase d'apprentissage, la méthode fournit donc $\frac{K(K-1)}{2}$ Fronts ROC (*cf.* étape 2 sur la figure 2), c'est-à-dire $\frac{K(K-1)}{2}$ populations de classifieurs. À ce stade, il est nécessaire de connaître les coûts pour pouvoir fournir des prédictions. Une fois que ces coûts sont connus, l'enjeu est de sélectionner pour chaque sous-problème binaire le SVM le plus adapté à la paire de coûts concernée (*cf.* étape 3 sur la figure 2). Cette sélection est faite de façon traditionnelle, en minimisant la fonction de perte incluant les coûts de mauvaise classification : $\mathcal{L}(\mathcal{C}) = p(\omega_1).TFN.c_{12} + p(\omega_2).TFP.c_{21}$. Le classifieur permettant d'obtenir la plus petite valeur de $\mathcal{L}$ est celui qui est retenu pour la prédiction. De fait, après les sélections de classifieurs pour chaque sous-problème binaire, on dispose de $\frac{K(K-1)}{2}$ SVM binaires, comme c'est le cas avec les versions multiclasses de ces classifieurs. La prédiction finale peut alors être obtenue de façon classique, en utilisant une procédure de combinaison de la litérature, comme par exemple celle de Wu et al. (2003) (*cf.* étape 4 sur la figure 2)). Les algorithmes 1 et 2 détaillent les procédures complètes d'apprentissage et de prédiction de la méthode proposée pour chaque matrice de coûts.

Les premiers résultats obtenus avec cette procédure ont été comparés à des SVM multiclasses traditionnels, pour lesquels les hyperparamètres ont été optimisés comme conseillé dans Bergstra et Bengio (2012), sans prise en compte de l'environnement. Les performances des deux méthodes ont été évaluées à l'aide d'une mesure appelée *Mean Subjective Utility* (MSU) (McDonald (2006)), qui est une variante mise à l'échelle de la fonction de perte en multiclasse. Contrairement à cette dernière, les valeurs de MSU sont toujours comprises entre 0 et 1 et les valeurs les plus élevées correspondent aux meilleurs performances. Le tableau 1 récapitule les résultats de nos premières expérimentations sur 5 bases de données, 3 d'entre elles étant issues de *l'UCI repository* (Bache et Lichman (2013)), une quatrième, *Synth8*, étant une

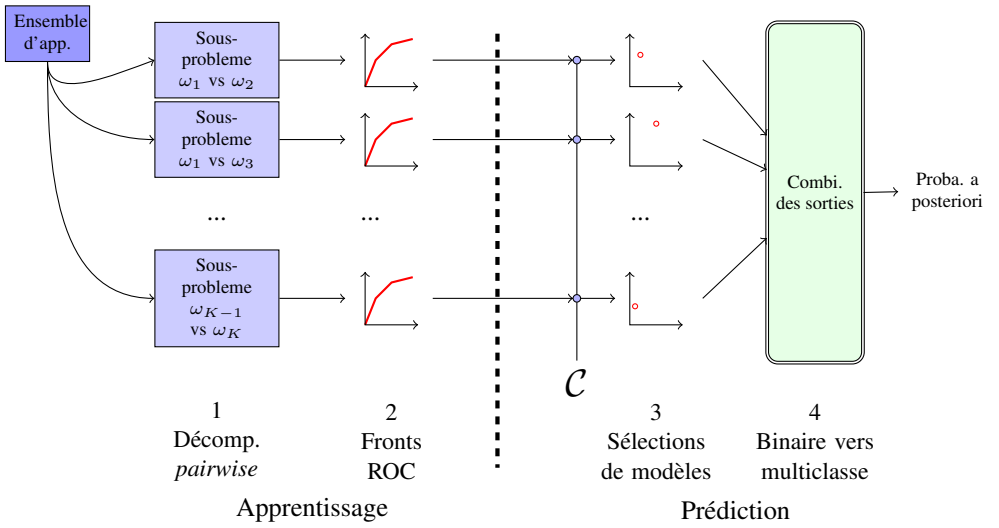

FIG. 2: Illustration de la méthode du Front ROC pour la classification multiclasse en environnement incertain. Le processus Apprentissage/Prédiction est composé de 4 étapes

Algorithm 1 *Apprentissage*

Require: D un ensemble d'apprentissage
Require: V un ensemble de validation
Require: K le nombre de classes
Require: $\mathcal{L}_{SVM}$ un algorithme d'apprentissage de SVM
Ensure: $(H_{ij})_{i,j\in[1..K]}$ une famille de $\frac{K(K-1)}{2}$ ensembles de classifieurs
1: $(d_{ij})_{i,j\in[1..K]} \leftarrow$ pairwiseDecomposition(D,K)
2: $(v_{ij})_{i,j\in[1..K]} \leftarrow$ pairwiseDecomposition(V,K)
3: **for** i from 1 to $K-1$ **do**
4: **for all** j from $i+1$ to K **do**
5: $H_{ij} \leftarrow ROCF(\mathcal{L}_{SVM}, d_{ij}, v_{ij})$
6: **end for**
7: **end for**
8: **return** $(H_{ij})_{i,j\in[1..K]}$

base synthétique générée avec la librairie Matlab *PRTools* (Duin et al. (2004)) [1], et la dernière étant la base de données MNIST pour laquelle les caractéristiques sont issues d'une pyramide multi-résolution de niveaux de gris. Ces résultats ont été obtenus à l'aide de 30 matrices de coûts générées aléatoirement à partir desquelles les valeurs de MSU ont été moyennées.

On constate que la méthode du Front ROC en multiclasse permet de s'adapter convenablement aux coûts de mauvaise classification, en comparaison avec une méthode de l'état de l'art qui ne les intégre pas. De plus, cela se vérifie également avec des problèmes allant jusqu'à

1. la fonction Matlab utilisée pour cela est *gendatm* avec le même nombres d'instances pour chacune des 8 classes.

Algorithm 2 *Test*

Require: $\mathcal{C}$ une matrice de coûts de mauvaises classification
Require: T un ensemble de test
Require: V un ensemble de validation
Require: K le nombre de classes
Ensure: $\hat{Y}$ un vecteur de $|T|$ prédictions
1: $(v_{ij})_{i,j\in[1..K]} \leftarrow$ pairwiseDecomposition(V,K)
2: **for** i from 1 to $K-1$ **do**
3: **for all** j from $i+1$ to K **do**
4: **for all** classifiers $h \in H_{ij}$ **do**
5: $\mathcal{J}(h) \leftarrow$ loss(v_{ij},h,c_{ij},c_{ji})
6: **end for**
7: $h_{ij} \leftarrow \arg\min_h \mathcal{J}(h)$ {*Sélection du SVM qui minimise* $\mathcal{J}$}
8: $P(i,j) \leftarrow h_{ij}(T)$ {*Scores du SVM pour les données de* T}
9: **end for**
10: **end for**
11: $Scores \leftarrow$ pairwiseToMulticlass(P)
12: **for all** k from 1 to $|T|$ **do**
13: $\hat{Y}(k) \leftarrow \arg\max_i Scores(k,\omega_i)$
14: **end for**
15: **return** $\hat{Y}$

Datasets	SVM_{opt}	$ROCF$	K
MFeat	0.832	**0.843**	10
MNIST	0.818	**0.852**	10
Segment	0.922	**0.954**	7
Synth8	0.892	**0.924**	8
Vehicle	0.800	**0.834**	4

TAB. 1: Résultats de MSU

10 classes, problèmes qui engendrent sans décomposition en paires de classes un problème d'optimisation de $\frac{10\times9}{2} = 45$ objectifs concurrents.

3 Conclusion

Nous avons présenté dans cet article une méthode de classification multiclasse qui s'adapte à des coûts de mauvaise classification déséquilibrés, sans nécessiter de les connaître en phase d'apprentissage. L'idée est d'utiliser une procédure d'optimisation multiobjective pour entrainer un ensemble de classifieurs, chacun étant spécialisé dans des coûts particuliers. Lorsque les coûts sont donnés, il suffit de sélectionner le classifieurs adéquat pour obtenir la prédiction. Ce type de méthode a déjà été appliqué avec succés à la classification binaire, mais n'avait jusque là jamais été adapté à la classification multiclasse. Les premiers résultats obtenus avec

cette méthode sont prometteurs et montrent que la méthode s'adapte bien aux coûts une fois qu'ils sont fournis, tout en déployant les efforts de calcul en phase d'apprentissage. C'est à notre connaissance la première fois qu'une telle méthode est appliquée à des problèmes dont le nombre de classes dépasse 3.

Références

Bache, K. et M. Lichman (2013). UCI Machine Learning Repository.

Bergstra, J. et Y. Bengio (2012). Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research 13*, 281–305.

Chatelain, C., S. Adam, Y. Lecourtier, L. Heutte, et T. Paquet (2010). A multi-model selection framework for unknown and/or evolutive misclassification cost problems. *Pattern Recognition 43*(3), 815–823.

Duin, R. P. W., P. Juszczak, D. de Ridder, P. Paclik, E. Pekalska, et D. M. Tax (2004). PRTools, a Matlab toolbox for pattern recognition.

Levesque, J.-C., A. Durand, C. Gagne, et R. Sabourin (2012). Multi-objective evolutionary optimization for generating ensembles of classifiers in the ROC space. In *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference*, pp. 879–886.

McDonald, R. A. (2006). The mean subjective utility score, a novel metric for cost-sensitive classifier evaluation. *Pattern Recognition Letters 27*(13), 1472 – 1477.

Wu, T.-F., C.-J. Lin, et R. Weng (2003). Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research 5*, 975–1005.

Zhou, Z.-H. et X.-Y. Liu (2010). On multi-class cost-sensitive Learning. *Computational Intelligence 26*(3), 232–257.

Summary

This paper adresses the problem of multiclass cost-sensitive classification. The proposed method does not need the knowledge of the misclassification costs during the learning stage and supposes that they are known only for prediction. It relies for that purpose on a multi-model selection strategy that learns a pool of classifiers, each one being specialized to particular misclassification costs, through a multiobjective ROC optimization procedure. The computational barriers involved by the multiclass context is broken using a pairwise decomposition. Experiments have been conducted on several multiclass datasets using a cost-sensitive evaluation protocol. The results confirm that efficient and tractable multi-class cost-sensitive learning can be achieved.