Stratégies multi-classes

Sommaire

5.1 Combinaisons de SVM 70
5.1.1 Estimation des probabilités a posteriori 70
5.1.2 Approche *Un contre tous* (OVA) 71
5.1.3 Approche *Un contre un* (OVO) 71
5.1.4 Codes Correcteurs d'Erreur (ECOC) 73
5.1.5 Classification hiérarchique 73
5.1.5.1 Graphe Acyclique Direct (DAGSVM) 73
5.1.5.2 Dendogrammes (DSVM) 74
5.1.5.3 Dendogrammes hybrides 74
5.1.5.4 Probabilités a posteriori par pondérations successives 75
5.2 Reformulation des SVM 76
5.3 Discussion et Conclusion 77

Nous abordons maintenant la question de l'adaptation, pour un problème multi-classes, des Machines à Vecteurs de Support, originellement conçues sur un paradigme de discrimination binaire. Nous emploierons par la suite le terme *multi-classes* pour désigner une classification impliquant plus de deux classes. Ce problème est antérieur à la création des SVM puisqu'il est légitimement posé par toute méthode discriminative, en particulier par la séparation par hyperplan linéaire. Ainsi, de nombreuses méthodes ont été proposées qui permettent de combiner les résultats de classifieurs binaires pour formuler une réponse multi-classes. Nous présenterons celles-ci dans la section 5.1. De nombreuses propositions ont également été faites pour reformuler les Machines à Vecteurs de Support dans un cadre multi-classes. Nous présenterons dans la section 5.2 quelques une des plus citées, qui aboutissent chacune à un nouveau problème d'optimisation. Nous discuterons par la suite, en section 5.3, des mérites comparatifs de ces deux paradigmes (combinaison et reformulation) et justifierons notre choix de nous restreindre au premier. Le lecteur intéressé par cette question particulière pourra consulter les références [95], [197] et [107] pour une comparaison détaillée des différentes approches.

Cette étude nous permettra ainsi de distinguer les méthodes permettant l'évaluation de probabilités a posteriori, qui nous seront utiles par la suite. Nous proposerons une méthodologie de classification basée sur le parcours d'un arbre de classification hybride combinant les approches «Un contre un» et par dendogramme.

On considérera dans la suite de ce chapitre que le label y_i associé à l'exemple $\boldsymbol{x}_i$ prend ses valeurs dans $[1, \ldots, C]$, où C est le nombre de classes impliquées dans le problème.

5.1 Combinaisons de SVM

5.1.1 Estimation des probabilités a posteriori

Nous présentons dans les sections suivantes les différentes stratégies proposées pour combiner les résultats de plusieurs machines bi-classes sur un problème multi-classes.

Toutefois il nous faut avant tout pour cela appliquer sur chaque SVM binaire un traitement destiné à en tirer un résultat probabiliste. On peut se restreindre à la mise en place d'un algorithme n'évaluant que la classe optimale associée à un exemple donné, mais nous verrons au chapitre 8, que l'on apporte un gain significatif aux performances des SVM en appliquant un post-traitement sur leurs résultats. Les post-traitements que nous présenterons exploitent les probabilités a posteriori associées aux classes du problème :

$$p_c(\boldsymbol{x}_i) = p(y_i = c \,|\, \boldsymbol{x}_i).$$

On construira donc, dans la mesure du possible, des algorithmes de combinaison dont le résultat est un vecteur contenant les probabilités a posteriori estimées.

Cependant, les SVM sont construites sur la séparation et non sur l'estimation de probabilités. En effet, la fonction de décision

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i y_i k(\boldsymbol{x}_i, \boldsymbol{x})$$

fournit des valeurs non-bornées et non-calibrées sur la droite réelle, et n'est construite que pour opérer une prise de décision sur le signe de la valeur de sortie : $\hat{y} = \text{sign}(f(\boldsymbol{x}))$. On peut cependant raisonnablement avancer l'hypothèse que plus la valeur est éloignée de 0, plus la classe estimée est fiable. Une des premières méthodes proposées [103] pour transformer la sortie des SVM en valeur probabiliste consiste à modéliser les valeurs de sortie de chaque classe par une gaussienne normalisée de manière à obtenir $P(y = c|f(\boldsymbol{x}) = 0) = 0.5$ pour chacune des classes $c = +1$ et $c = -1$. Néanmoins cette méthode est affaiblie par le fait que l'hypothèse de gaussianité sur les densités de probabilités est rarement respectée.

Partant du constat empirique que les densités de probabilités conditionnelles de chaque classe (pour les valeurs $f(\boldsymbol{x})$) sont exponentielles dans la marge, Platt propose [185] de modéliser la probabilité de la classe positive par une forme sigmoïdale :

$$P\left(y = 1 \,|\, f(\boldsymbol{x})\right) = \frac{1}{1 + \exp\left(A\, f(\boldsymbol{x}) + B\right)}.$$

La probabilité de la classe négative étant implicite :

$$\begin{aligned} P\left(y = -1 \,|\, f(\boldsymbol{x})\right) &= 1 - P\left(y = 1 \,|\, f(\boldsymbol{x})\right) \\ &= \frac{\exp\left(A\, f(\boldsymbol{x}) + B\right)}{1 + \exp\left(A\, f(\boldsymbol{x}) + B\right)}. \end{aligned}$$

Les paramètres A et B sont fixés en maximisant l'estimation de vraisemblance sur les exemples (f_i, y_i) de l'ensemble d'apprentissage (avec $f_i = f(\boldsymbol{x}_i)$) :

$$\min_{A,B} \quad -\sum_{i=1}^{n} t_i \log(p_i) + (1 - t_i) \log(1 - p_i),$$

où l'on a défini les valeurs $t_i = \dfrac{y_i + 1}{2}$ et $p_i = \dfrac{1}{1 + \exp\left(A\, f_i + B\right)}$.
Le problème de minimisation peut être résolu à l'aide de n'importe quelle méthode d'optimisation.

Par la suite on désignera par f^* la composition de la fonction de décision et du post-traitement sigmoïdal, soit :

$$f^*(\boldsymbol{x}) = \frac{1}{1 + \exp\left(A\, f(\boldsymbol{x}) + B\right)}.$$

5.1.2 Approche *Un contre tous* (OVA)

Le premier algorithme multi-classes employé pour les SVM [208][236] est également le plus simple. Il consiste à utiliser un classifieur binaire pour chaque classe. Celui-ci est appris pour discriminer les exemples de la classe des exemples de l'ensemble des autres classes, d'où son nom de *Un contre tous* (ou *One versus All*, OVA). Si l'on désigne par f_c la fonction de décision du classifieur concernant la classe c, l'algorithme OVA choisit donc la classe maximisant les valeurs prises par les fonctions de décisions :

$$\hat{y} = \arg\max_{1 \leq c \leq C} f_c(\boldsymbol{x}).$$

On remarque que le vecteur $[f_1(\boldsymbol{x}), \ldots, f_C(\boldsymbol{x})]$ ne saurait constituer un vecteur de probabilités a posteriori puisque leur somme n'est pas unitaire. On pourra, en normalisant celles-ci, fournir l'estimation suivante des probabilités a posteriori :

$$\hat{p}_c(\boldsymbol{x}) = \frac{f_c(\boldsymbol{x})}{\sum_{k=1}^{C} f_k(\boldsymbol{x})}.$$

Cependant, les fonctions de décisions étant indépendantes, ces probabilités peuvent être très biaisées et n'ont pas de fondement statistique solide.

En pratique, l'algorithme OVA, se révèle très efficace pour la prise de décision, malgré sa simplicité. Cette question, largement débattue dans la littérature, sera discutée en section 5.3. Les défauts majeurs de cet algorithme restent son inadéquation pour l'estimation de probabilités a posteriori, et le fait que des fonctions de décision peuvent être peu fiables si certaines classes disposent de beaucoup moins d'exemples d'apprentissage que les autres.

5.1.3 Approche *Un contre un* (OVO)

Lorsque le nombre de classes est trop élevé, le problème de séparation OVA peut devenir trop complexe, engendrant ainsi des classifieurs mal calibrés. On peut donc espérer mieux contrôler la complexité des surfaces de décision en se restreignant à l'usage de classifieurs appris sur des couples de classes. Les figures 5.1 et 5.2 illustrent cette différence sur un problème simple n'impliquant que 3 classes. On peut ainsi constater que le formalisme *un contre tous* (à gauche) peine à déterminer un plan de séparation adéquat pour la discrimination 1 *vs* 2&3 tandis le problème ne se pose pas dans une approche par paires (à droite).

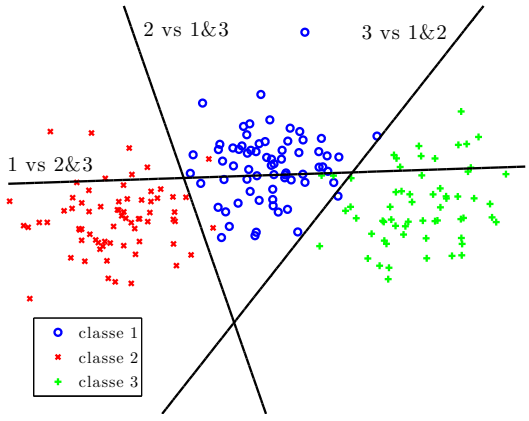

FIGURE 5.1 – Hyperplans de séparation par approche *un contre tous* (OVA) sur 3 classes.

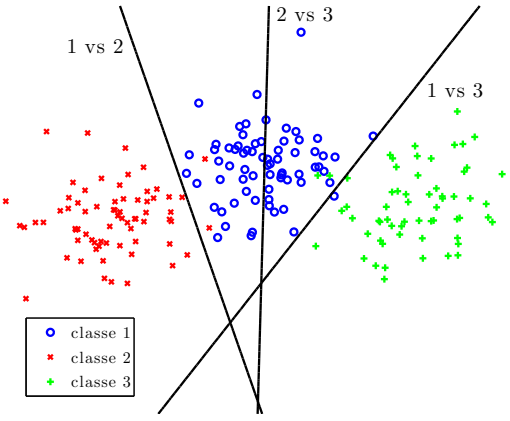

FIGURE 5.2 – Hyperplans de séparation par approche *un contre un* (OVO) sur 3 classes.

L'approche *un contre un*, généralement attribuée à Knerr et al. [126] et Friedman [81], se base sur les résultats des classifieurs séparant chacune des $\frac{C(C-1)}{2}$ paires sur les C classes.

On désigne par f_{kl} la fonction de décision du classifieur appris pour discriminer la classe positive k et la classe négative l ; on peut donc également considérer le classifieur $f_{lk} = -f_{kl}$, pour simplifier

les notations à venir. Friedman propose la règle de décision multi-classes suivante, basée sur un vote majoritaire sur l'ensemble des classes :

$$\hat{y} = \underset{1\leq k\leq C}{\arg\max} \sum_{l=1}^{C} H\left(f_{kl}(\boldsymbol{x})\right), \tag{5.1}$$

où H est la fonction de Heaviside. La stratégie du vote majoritaire associée aux classifieurs par paires est employée pour la première fois sur les SVM par Kressel [127], et est depuis largement reprise dans la littérature.
Toutefois ce paradigme présente deux défauts majeurs :

- De nombreuses régions de l'espace de décision (où évolue le vecteur $[f_{kl}(\boldsymbol{x})]_{k>l}$) sont indécidables, parculièrement lorsque le nombre de classes est réduit. Par exemple, dans un problème à 3 classes, si $f_{12}(\boldsymbol{x}) > 0$, $f_{23}(\boldsymbol{x}) > 0$ et $f_{31}(\boldsymbol{x}) > 0$, toutes les classes maximisent le critère. Le cas d'un problème à 4 classes où 2 classes maximisent le critère, sans configuration cyclique, est également plausible et problématique.
- La présence de la fonction de Heaviside introduit une forte discontinuité dans le critère. Il est donc impossible d'espèrer en tirer une estimation fiable des probabilités a posteriori.

Hastie et Tibshirani ont proposé [103] un algorithme permettant d'estimer les probabilités a posteriori sur une approche OVO, qui résout ainsi ces deux problèmes. Celui-ci se base sur une procédure d'optimisation sur les probabilités a posteriori p_i, minimisant la distance entre les résultats probabilisés $f^*_{kl}(\boldsymbol{x})$ (ici notés r_{kl}) des classifieurs et une estimation μ_{kl} de ces valeurs, calculée à partir des $\boldsymbol{p} = [p_i]_{1\leq i\leq C}$. Ainsi si on définit μ_{kl} comme l'estimée de la probabilité conditionnelle de la classe k, sachant que la classe est k ou l :

$$\mu_{kl} = \mathrm{E}(r_{kl}) = \frac{p_k}{p_k + p_l},$$

soit,

$$\log \mu_{kl} = \log(p_k) - \log(p_k + p_l).$$

On souhaite mettre à jour les p_i de manière à minimiser la distance entre les r_{kl} (pour rappel $r_{kl} = f^*_{kl}(\boldsymbol{x})$) et les μ_{kl}. La mesure choisie par Hastie et Tibshirani est la distance moyenne de Kullback-Leibler entre les deux vecteurs, soit :

$$\ell(\boldsymbol{p}) = \sum_{k<l} n_{kl} \left[r_{kl} \log \frac{r_{kl}}{\mu_{kl}} + (1 - r_{kl}) \log \frac{1 - r_{kl}}{1 - \mu_{kl}} \right],$$

celle-ci est pondérée par les valeurs n_{kl} qui représentent le nombre d'exemples utilisés pour l'apprentissage du classifieur f_{kl}. Ce terme permet de compenser les effets d'une disproportion entre les données disponibles pour les différentes classes, et peut être uniforme sur l'ensemble des classes, dans le cas de classes à peu près équilibrées, sans nuire aux performances de l'algorithme.

Du calcul du gradient du critère $\ell(\boldsymbol{p})$, on déduit la méthode d'optimisation PWC (*Pair-Wise Classification*) synthétisée par l'algorithme 2. En pratique on utilise pour l'initialisation des probabilités $\hat{p}_k$ les valeurs normalisées de la procédure de Friedman (équation 5.1), soit $\hat{p}_k = \frac{\sum_{l=1}^{C} H\left(f_{kl}(\boldsymbol{x})\right)}{\sum_{k=1}^{C}\sum_{l=1}^{C} H\left(f_{kl}(\boldsymbol{x})\right)}$.
La convergence est en général très rapide. Une alternative a cependant été proposée [101] qui permet la détermination des probabilités a posteriori sans itération. On pourra toutefois noter que Hastie et Tibshirani précisent qu'une seule itération est suffisante si l'on ne s'intéresse qu'à la probabilité majoritaire (c'est-à-dire la seule décision multi-classes).

Plusieurs auteurs formulent la critique que l'approche OVO prend en compte les résultats de tous les classifieurs, y compris ceux pour lesquels la classe d'un exemple donné n'est pas concernée, ce qui introduit une part importante d'informations non-pertinentes qui peut pénaliser le processus. Ainsi, l'algorithme O-PWC se propose [136] de combiner les résultats des C algorithmes PWC pour chacun desquels seuls les couples impliquant une classe donnée sont pris en compte (au travers des poids n_{kl}). Garcia-Pedrajas et Ortiz-Boyer [84] proposent également de combiner les approches OVO et OVA, sans toutefois réellement justifier leur démarche d'un point de vue théorique, ni mettre en valeur dans leurs résultats un réel gain de performances.

Algorithme 2 PWC, *PairWise Classification*

$r_{kl} = f^*_{kl}(\boldsymbol{x})$
Initialiser les estimées $\hat{p}_k$.
Initialiser les $\hat{\mu}_{kl}$:
$$\hat{\mu}_{kl} = \frac{p_k}{p_k + p_l}$$
répéter
Mise à jour des $\hat{p}_k$:
$$\hat{p}_k \leftarrow \hat{p}_k \frac{\sum_{k \neq l} n_{kl} r_{kl}}{\sum_{k \neq l} n_{kl} \hat{\mu}_{kl}}.$$
Normalisation des probabilités :
$$\hat{\boldsymbol{p}} \leftarrow \hat{\boldsymbol{p}} / \sum \hat{p}_k$$
Mise à jour des $\hat{\mu}_{kl}$:
$$\hat{\mu}_{kl} = \frac{p_k}{p_k + p_l}$$
jusqu'à convergence des $\hat{p}_k$

5.1.4 Codes Correcteurs d'Erreur (ECOC)

Dietterich et Bakiri développent dans [65] un cadre plus général pour la combinaison de classifieurs binaires. Sans imposer de contrainte a priori sur la collection de classifieurs exploités, ils proposent de baser la fusion de résultats sur un principe inspiré de la théorie des codes correcteurs d'erreurs. On définit ainsi une matrice binaire $\boldsymbol{M} = [m_{cn}] \in \mathcal{M}_{C,N}(\{-1, 1\})$ contenant C lignes (où C est le nombre de classes) et N colonnes (N étant le nombre de classifieurs binaires f_n impliqués). Chaque vecteur ligne $\boldsymbol{M}_c$ de la matrice $\boldsymbol{M}$ est un *mot-code* pour une classe donnée. Chaque exemple $\boldsymbol{x}$, après classification par les N classifieurs, se voit attribuer un vecteur $\boldsymbol{f} = [f_1(\boldsymbol{x}), \ldots, f_N(\boldsymbol{x})]$ regroupant les résultats des fonctions de décision.

Le principe de la méthode ECOC (*Error Correcting Output Codes*) consiste à attribuer à l'exemple $\boldsymbol{x}$ la classe minimisant la distance L^1 (distance de Manhattan) entre son mot-code $\boldsymbol{M}_c$ et le mot-code $\boldsymbol{f}$ de l'exemple :

$$\hat{y} = \underset{1 \leq c \leq C}{\arg\min}\, L1(\boldsymbol{f}, \boldsymbol{M}_c) = \underset{1 \leq c \leq C}{\arg\min} \sum_{n=1}^{N} |f_n(\boldsymbol{x}) - m_{cn}|.$$

Allwein et al. [14] apportent un regard intéressant sur les ECOC en étendant l'espace des matrices de codes aux matrices "ternaires", soit $m_{cn} \in \{-1, 0, 1\}$, où la valeur $m_{cn} = 0$ introduite représente le fait que le classifieur n n'apporte aucune information sur la classe c. Cette extension permet d'inclure les méthodes OVO et OVA dans le cadre théorique des ECOC.

Le point central de la méthode ECOC est la définition de la matrice de codes $\boldsymbol{M}$, qui doit être définie avec soin pour représenter le problème posé. Cependant, en définitive, l'approche ECOC apporte peu par rapport aux approches OVA et OVO, dans un cas comme le notre, où le nombre de classes est assez restreint. On n'entreprendra donc pas d'étude expérimentale sur celle-ci.

5.1.5 Classification hiérarchique

Les méthodes présentées jusqu'ici sont basées sur des combinaisons de classifieurs indépendants qui peuvent être traités en parallèle. Nous présentons ici quelques méthodes multi-classes où l'ordre des classifieurs est déterminant. Celles-ci sont basées sur une structure d'arbre (graphes connexes acycliques) dont les nœuds représentent les différents classifieurs ; on parlera également de *classification hiérarchique*.

5.1.5.1 Graphe Acyclique Direct (DAGSVM)

La première contribution basée sur les graphes de décision s'appuie sur une structure particulière proposée par Platt et al. [186], qui réduit le nombre de classifications dans le processus de décision. On associe au nœud racine l'ensemble des classes. Chaque nœud décrit un classifieur portant sur la première et la dernière des classes associées. La règle de décision implique deux branches, associées

chacune à la négation d'une classe, qui est ôtée de la liste du noeud fils. La figure 5.3 clarifie cette structure pour un exemple à 4 classes.

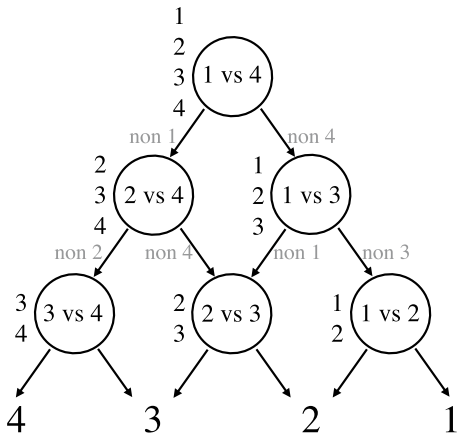

FIGURE 5.3 – Structure du Graphe Acyclique Direct (DAGSVM) défini pour un problème à 4 classes. À chaque nœud est associé un discriminateur (n vs m) et la liste des classes restantes (agencée verticalement à gauche du nœud).

Les auteurs nomment ce modèle DAGSVM (pour *Direct Acyclic Graph SVM*). Celui-ci repose sur l'élimination d'une classe à chaque nœud de discrimination. La feuille terminale de l'arbre indique l'unique classe restante, résultat de la classification. On voit que cette structure implique exactement les mêmes classifieurs par paires que l'approche OVO. Si l'ordre des éliminations successives a théoriquement une influence sur les résultats, les auteurs avancent, d'après leurs observations empiriques, que celle-ci est très modérée et non systématique.

Cette approche a le mérite de se baser sur les classifieurs par paires, supposés plus robustes que les classifieurs OVA, tout en n'impliquant que C classifications pour la prise de décision, au lieu des $C(C-1)/2$ nécessaires pour l'approche OVO. Néanmoins, comme la méthode OVO avec vote majoritaire, elle ne fournit aucune estimation des probabilités a posteriori.

5.1.5.2 Dendogrammes (DSVM)

La structure d'arbre de classification permet également la mise en place d'un algorithme [21] par discriminations successives sur des unions de classes, à l'aide d'un *dendogramme*. Les auteurs du DSVM (*Dendogram SVM*) regroupent itérativement les classes par *clustering bottom-up* sur les exemples d'apprentissage. Le processus, basé sur une mesure de proximité entre les groupes de classes, permet ainsi la construction d'un dendogramme décrivant le processus *top-down* de classification, illustré par l'exemple figure 5.4.

La structure du dendogramme renseigne sur les classifieurs nécessaires à la classification. Ceux-ci associent aux nœuds les plus profonds des paires de classes, et aux autres nœuds des paires impliquant des unions de classes (qui regroupent les exemples de plusieurs classes). On trouve une formulation équivalente sous le nom de « *Half against Half* » [130], où le dendogramme est construit sur un paradigme *top-down*, où chaque nœud de classification est choisi de manière à équilibrer les deux groupes de classes discriminées.

Cette approche implique un très net gain en complexité puisque la classification n'implique que $\lceil \log_2 C \rceil$ discriminations successives. Toutefois elle ne fournit qu'un indice de classe estimée et ne permet pas d'estimer les probabilités a posteriori.

5.1.5.3 Dendogrammes hybrides

Nous proposons une extension du cadre de l'approche par dendogramme en combinant ce dernier à l'approche *One-vs-One*. On peut en effet étendre l'arbre de classification à des arbres non-binaires en traitant d'éventuels nœuds non-binaires (c'est-à-dire à plus de deux branches filles) par une approche multi-classes non hiérarchisée. Le cadre *One-vs-One* est ici le mieux indiqué puisqu'il nous permet, contrairement à l'approche *One-vs-All*, d'estimer les probabilités a posteriori des classes

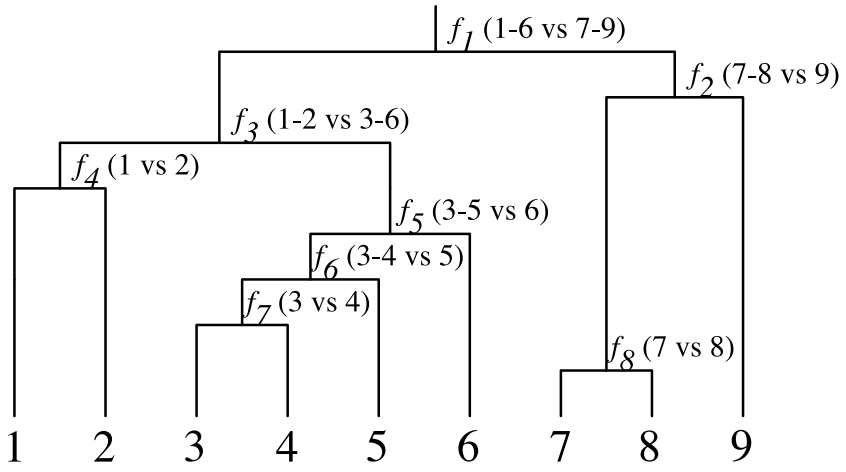

FIGURE 5.4 – Dendogramme de classification *top-down* sur un exemple de problème à 9 classes. Chaque noeud du dendogramme est associé à un classifieur binaire f_n.

impliquées.

Cette approche constitue ainsi ce que nous appelons le *dendogramme hybride*.

5.1.5.4 Probabilités a posteriori par pondérations successives

La plupart des méthodes multi-classes présentées jusque-là se focalisent sur le problème de l'estimation de la classe des exemples, laissant de côté la question des probabilités a posteriori. Nous proposons une méthode simple permettant d'estimer ces probabilités sur l'approche par dendogramme.

Celle-ci est basée sur un parcours récursif de l'arbre de classification. On supposera qu'à chaque nœud est associé un index n unique et un classifieur f_n. Le résultat probabilisé du classifieur f_n sur l'exemple $\boldsymbol{x}$ est noté $f_n^*(\boldsymbol{x})$. Le nœud racine est associé à l'index 1. La figure 5.5 fournit un exemple de dendogramme indexé pour un problème à 6 classes (les index de nœuds sont en gris clair et les index de classes en bleu).

Le déroulement de notre méthode sur un exemple $\boldsymbol{x}$ est le suivant :

- Le nœud racine (d'indice 1) reçoit une probabilité d'entrée égale à $p_{in}^1 = 1$.
- Tout nœud fils n reçoit une probabilité p_{in}^n de son nœud parent.
- Le nœud n produit une probabilité de sortie $p_{out,\pm}^n$ pour les 2 classes traitées par le classifieur f_n, pondérée par la probabilité d'entrée :

$$\begin{aligned} p_{out,+}^n &= p_{in}^n \, f_n^*(\boldsymbol{x}) \\ p_{out,-}^n &= p_{in}^n \, (1 - f_n^*(\boldsymbol{x})) \, . \end{aligned}$$

- Si le nœud n a deux nœuds fils m_+ et m_-, ses probabilités de sorties sont transmises en entrée des nœuds fils :

$$\begin{aligned} p_{in}^{m_+} &= p_{out,+}^n \\ p_{in}^{m_-} &= p_{out,-}^n . \end{aligned}$$

La concaténation des probabilités de sortie de toutes les feuilles constitue l'estimée des probabilités a posteriori. Ainsi on obtient sur l'exemple de la figure 5.5 :

$$\begin{aligned} \hat{\boldsymbol{p}}(\boldsymbol{x}) &= \left[p_{out,+}^4, p_{out,-}^4, p_{out,+}^5, p_{out,-}^5, p_{out,+}^3, p_{out,-}^3 \right] \\ &= \left[f_1 f_2 f_4, f_1 f_2 (1 - f_4), f_1 (1 - f_2) f_5, f_1 (1 - f_2)(1 - f_5), (1 - f_1) f_3, (1 - f_1)(1 - f_3) \right] , \end{aligned}$$

où l'on utilise la notation allégée $f_n = f_n^*(\boldsymbol{x})$.

Cet algorithme consiste tout simplement en une mise à jour récursive des probabilités par pondérations successives. Le processus garantit par ailleurs que le vecteur estimé a bien une somme

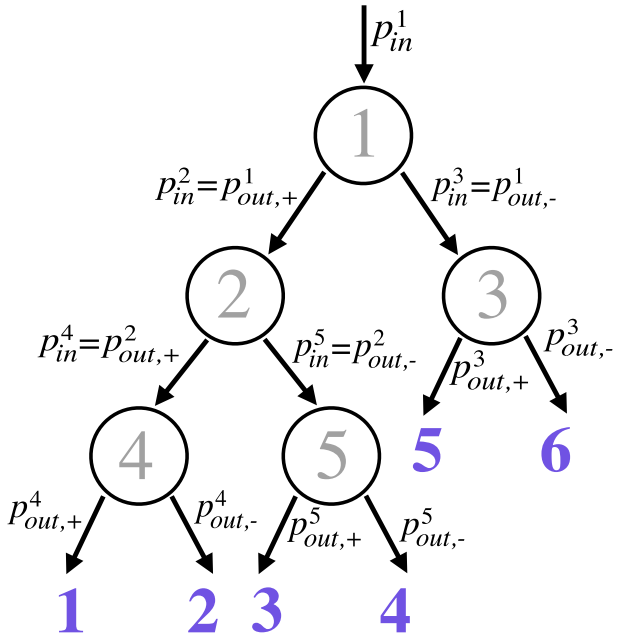

FIGURE 5.5 – Exemple d'arbre de classification pour l'estimation de probabilités a posteriori. La figure illustre la transmission des probabilités de sorties d'un nœud vers la probabilité d'entrée des nœuds fils. Les index de nœuds sont en gris clair, les index de classes en bleu aux feuilles de l'arbre.

unitaire. Il est par ailleurs applicable sur les dendogrammes hybrides puisque l'approche *one-vs-one* produit également des probabilités a posteriori.

Etude de l'application aux DAGSVM

L'application à l'approche DAGSVM est moins cohérente. En effet, le graphe n'ayant pas une structure d'arbre (puisqu'un nœud peut avoir plusieurs parents), il existe plusieurs estimations possibles pour certaines classes. Ainsi, en se basant sur l'exemple de la figure 5.3, on obtient les estimées suivantes :

$$\hat{p_1} = (1-f_{14})(1-f_{24})(1-f_{34}) \qquad \hat{p_4} = f_{14}f_{13}f_{12}$$

$$\begin{aligned}\hat{p_2} &= (1-f_{14})(1-f_{24})f_{34} \\ &= (1-f_{14})f_{24}(1-f_{23}) \\ &= f_{14}(1-f_{13})(1-f_{23})\end{aligned} \qquad \begin{aligned}\hat{p_3} &= (1-f_{14})f_{24}f_{23} \\ &= f_{14}(1-f_{13})f_{23} \\ &= f_{14}f_{13}(1-f_{12})\end{aligned}$$

De plus les probabilités estimées ne sont pas de somme unitaire. Il est bien sûr possible de les normaliser mais ce constat affaibli la pertinence de l'estimation pour l'approche DAGSVM.

5.2 Reformulation des SVM

Nous avons décrit plusieurs méthodes pour combiner les résultats de différents classifieurs bi-classes pour une tâche multi-classes. Plusieurs auteurs ont cependant tenté de reformuler directement les Machines à Vecteurs de Support sur ce type de problèmes. Cette tâche est loin d'être évidente puisque, comme nous l'avons vu, les SVM reposent sur le principe de la maximisation de la marge, dont la définition même repose sur le principe de la séparation linéaire. On trouve ainsi plusieurs alternatives dans la littérature, dont les références [197] et [95] couvrent un large éventail. Nous présentons ici brièvement les principales d'entre elles.

La première approche est introduite en 1998 par Weston et Watkins [246]. Elle consiste à déterminer simultanément les C fonctions de décision « un contre tous » f_c :

$$f_c(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_{c,i}\, y_i\, k(\boldsymbol{x}, \boldsymbol{x}_i) + b_c.$$

L'idée de base est de contraindre les variables d'écart, non plus indépendamment pour chaque fonction, mais relativement aux résultats des autres. Ainsi, le problème comprend $n \times (C-1)$ variables d'écart ξ_{ic} (où n est le nombre d'exemples) relatives à chaque classe $c \neq y_i$, où y_i est la classe de l'exemple $\boldsymbol{x}_i$, avec les contraintes suivantes :

$$f_{y_i}(\boldsymbol{x}_i) \geq f_c(\boldsymbol{x}_i) + 1 - \xi_{ic} \qquad \text{et} \qquad \xi_{ic} \geq 0$$

Si la formulation du problème est simple, elle pose plusieurs difficultés pour sa mise en application. En premier lieu la formulation du problème dual (non présentée ici) ne permet pas de se débarrasser des constantes introduites dans le problème primal, comme c'est le cas pour les SVM bi-classes. De plus, les auteurs ne proposent aucune implémentation efficace de l'algorithme d'optimisation, qui ne peut tirer partie des techniques permettant de rendre efficace l'apprentissage SVM traditionnel. Le problème posé est donc beaucoup plus complexe, du fait de la multiplication du nombre de variables d'écart, et donc difficilement applicable sur des données réelles.

Bredensteiner et Bennett ont proposé, sans lien avec Weston et Watkins, une approche [37] basée sur les mêmes contraintes liant les fonctions de décisions entre elles :

$$f_{y_i}(\boldsymbol{x}_i) \geq f_c(\boldsymbol{x}_i) + 1 - \xi_{ic}.$$

Soit, dans le cas linéaire :

$$(\boldsymbol{w}_{y_i} - \boldsymbol{w}_c)^T\, \boldsymbol{x}_i \geq (b_c - b_{y_i}) + 1 - \xi_{ic}.$$

Cette relation les mène à proposer la valeur $\frac{2}{\|\boldsymbol{w}_c - \boldsymbol{w}_d\|}$ comme mesure de séparabilité entre les classes c et d. Ils suggèrent donc la minimisation du terme $\|\boldsymbol{w}_c - \boldsymbol{w}_d\|$ sur toutes les paires (c,d), regularisée par le terme $\sum_{c=1}^{C} \|\boldsymbol{w}_c\|^2$.L'expression duale du problème permet de substituer aux produits scalaires la fonction noyau. L'équivalence formelle avec l'approche de Weston et Watkins a par la suite été démontrée par différents auteurs [95][107].

Crammer et Singer proposent [55] par la suite une approche simplifiant considérablement le modèle de Weston et Watkins. Au lieu de définir une pénalité ξ_{ic} pour chaque couple (i,c), une seule pénalité est définie pour la valeur $f_c(\boldsymbol{x})$ maximale. Ainsi on retrouve une unique variable d'écart pour chaque exemple. Le problème d'optimisation s'exprime alors de la manière suivante :

$$\begin{aligned} \min_{f_1,\ldots,f_C} \quad & \tfrac{1}{2} \textstyle\sum_{c=1}^{C} \|\boldsymbol{w}_c\|^2 + C_{pen} \sum_{i=1}^{n} \xi_i \\ \text{sous les contraintes} \quad & f_{y_i}(\boldsymbol{x}_i) \geq f_c(\boldsymbol{x}_i) + 1 - \xi_i. \\ & \xi_i \geq 0 \end{aligned}$$

Les auteurs proposent en outre un algorithme de décomposition du problème d'optimisation, permettant une implémentation efficace de leur approche multi-classes.

5.3 Discussion et Conclusion

Le bilan essentiel de la courte présentation précédente est qu'il n'existe pas à l'heure actuelle de formulation multi-classes des SVM qui fasse consensus au sein de la communauté. Celle-ci se limite encore essentiellement aux méthodes par combinaisons présentées dans la section 5.1, beaucoup plus simples à mettre en place et bien moins coûteuses en temps de calcul.

En vérité, on peine à trouver dans la littérature des résultats qui justifient l'usage des SVM reformulées plutôt que celui des méthodes par combinaisons. Les résultats expérimentaux de Weston et Watkins [246] ne montrent pas d'amélioration des performances par rapport à une simple approche OVA ou OVO, et se focalisent surtout sur la réduction du nombre de vecteurs de support.

On trouvera de plus dans [107] un test comparatif impliquant les méthodes de Weston & Watkins, de Crammer & Singer, OVO, OVA et les DAGSVM, duquel il ne ressort pas d'avantage particulier pour les méthodes reformulées.

Rifkin et Klautau [197] ont également écrit un plaidoyer très didactique en faveur de la méthode un contre tous (OVA), souvent dédaignée dans la littérature. Reproduisant avec rigueur les expériences de nombreux articles, ils montrent que cette méthode, de loin la plus simple de toutes, est tout à fait comparable en performances aux alternatives considérées (OVA, ECOC avec diverses configurations, ainsi que les méthodes reformulées présentées précédemment). En définitive, leur constat est que les méthodes se valent globalement, si l'on prend la peine d'affiner avec soin les paramètres des classifieurs SVM impliqués dans le processus. Cette conclusion justifie donc l'attention particulière que nous portons à cette question dans le chapitre 4.

Les implémentations libres de méthodes par SVM réformulées sont pour l'instant rares et très coûteuses en temps de calcul. Aussi, au vue des résultats expérimentaux énoncés ci-dessus, nous n'avons porté notre attention que sur les méthodes multi-classes par combinaisons de SVM.

Il est difficile d'évaluer les méthodes multi-classes en dehors d'un contexte applicatif, aussi nous reportons l'évaluation comparée de ces dernières au chapitre d'évaluation 10, dont la section 10.3 présente une comparaison de différentes taxonomies de classification sur des corpus audio.

Toutefois, le résultat théorique essentiel de ce chapitre réside dans notre proposition d'une méthode de classification hybride combinant l'approche *one-vs-one* aux arbres de classification hiérarchiques, couplé à une procédure d'estimation des probabilités a posteriori. Cette dernière contribution démarque l'approche proposée de la plupart des méthodes de l'état de l'art (comme l'approche *one-vs-all* ou les DAGSVM), qui ne permettent pas d'estimer ces probabilités.