Vote de majorité *a priori* contraint pour la classification binaire : spécification au cas des plus proches voisins

Aurélien Bellet[1], Amaury Habrard[2], Emilie Morvant[3], et Marc Sebban[2]

[1]University of Southern California, Department of Computer Science, Los Angeles, CA 90089
[2]Université Jean Monnet de Saint-Étienne, Laboratoire Hubert Curien, UMR CNRS 5516
[3]Aix-Marseille Univ., LIF-QARMA, UMR CNRS 7279

27 mai 2013

Résumé

Pour combiner différents classifieurs/votants, une solution naturelle vise à construire un vote de majorité. Un algorithme récemment introduit, MinCq, apprend un tel vote en optimisant les poids associés aux votants. Son principe repose sur la minimisation du risque du vote de majorité (la C-borne), dans le cadre de la théorie PAC-Bayes. Une limite de MinCq vient du fait qu'il ne peut tirer avantage d'une connaissance *a priori* sur la performance des votants (comme cela peut être le cas des classifieurs de type plus proches voisins (PPV)). Dans cet article, nous introduisons P-MinCq, une extension de MinCq, afin de considérer une contrainte *a priori* sur la distribution des poids des votants. Cette contrainte pouvant dépendre des exemples d'apprentissage, nous généralisons les preuves de convergence aux schémas de compression. Appliqué à un vote de majorité sur un ensemble de classifieurs PPV et évalué sur vingt jeux de données, nous montrons que P-MinCq est significativement plus performant qu'un PPV classique, un PPV symétrique et MinCq lui-même. Nous montrons finalement que combiné avec LMNN, un algorithme d'apprentissage de métrique, P-MinCq permet d'obtenir des résultats encore meilleurs.

Mots-clef : Vote de majorité, Plus Proches Voisins, PAC-Bayes, Schéma de compression

1 Introduction

Un vote de majorité est une méthode de combinaison de classifieurs où l'on cherche à optimiser la pondération des votants. L'objectif est alors de construire un vote final plus performant et plus robuste que les votants individuels. Cependant, choisir des poids pertinents est une tâche parfois complexe. Dans ce contexte, un algorithme récent (MinCq) a été proposé [LMR11] pour apprendre les poids selon des principes inhérents à la théorie PAC-Bayésienne [McA99]. MinCq optimise les poids sur un ensemble de votants $\mathcal{H}$ en minimisant une borne sur le risque du vote de majorité – la C-borne [LLM$^+$07] – mettant en jeu les deux premiers moments statistiques de la marge du vote. Les auteurs ont montré, par une borne en généralisation PAC-Bayésienne, que l'optimisation empirique de cette borne permet de minimiser le risque réel du vote de majorité et revient à résoudre un programme quadratique simple. La sortie de l'algorithme MinCq est une distribution posterior sur $\mathcal{H}$ pondérant les votants. MinCq a montré de bons résultats sur des votes de stumps et de noyaux RBF [LMR11].

Il est important de noter que MinCq semble offrir un cadre naturel au contexte des plus proches voisins (PPV), pour lesquels deux principales stratégies d'amélioration ont été proposées dans la littérature. La première repose sur le fait qu'avec les PPV, les probabilités conditionelles par classe sont supposées localement régulières. Or, plus la dimension de l'espace augmente, moins cette hypothèse est satisfaite. En conséquence, une solution est d'adapter localement les voisinages [HT96, NSB03]. D'autre part, la performance des PPV étant dépendante de la distance intervenant dans le calcul des voisinages, une seconde stratégie fait appel à l'apprentissage de métriques [1]. Par exemple, LMNN (*Large Margin Nearest Neighbor*) apprend une distance de Mahalanobis qui minimise l'erreur empirique d'apprentissage d'un k-PPV avec une marge minimale [WS09]. Quelle que soit la stratégie, le nombre de voi-

1. Voir [YJ06] pour un état de l'art.

sins k reste à être convenablement choisi et la règle de décision se base sur ces voisinages locaux pouvant aboutir à des phénomènes de sur-apprentissage (notamment en grande dimension). A priori, MinCq semblerait être pertinent pour contourner ce problème, en optimisant un vote pondéré de PPV, pour lequel l'ensemble $\mathcal{H}$ serait constitué de classifieurs k-PPV (pour $k = \{1, 2, \ldots\}$). Cependant, comme nous le verrons dans des expériences préliminaires, MinCq s'avère être moins performant qu'un simple k-PPV. La raison de ce comportement vient du fait que la contrainte de quasi-uniformité nécessaire à MinCq suppose que les votants ont *a priori* la même importance, ce qui n'est clairement pas le cas des k-PPV, notamment dans un contexte d'échantillon d'apprentissage de taille finie. De plus, les garanties en généralisation de MinCq ne sont plus valides pour les PPV car, les votants étant construits à partir d'exemples d'apprentissage [GHST05], le contexte relève désormais des schémas de compression [LMR11].

Dans ce papier, nous proposons une généralisation de MinCq de deux manières. Tout d'abord, nous reformulons le problème pour contraindre la distribution posterior à être **P**-alignée, où **P** modélise un prior sur la distribution des poids des votants. **P** permet d'incorporer ainsi une connaissance *a priori* sur la performance de chaque votant. Nous montrons que toute distribution sur les poids peut être exprimée comme une distribution **P**-alignée et que ce nouveau problème, appelé P-MinCq, s'exprime lui aussi comme un programme quadratique. Ensuite, nous étendons les preuves de convergence [LMR11] aux schémas de compression. À l'aide de ces deux contributions originales, nous instancions P-MinCq au cas d'un vote de majorité sur un ensemble de classifieurs k-PPV. Pour ce faire, nous définissons une contrainte **P** modélisant la plus grande confiance que l'on donne aux voisinages locaux. Ceci offre une approche théoriquement fondée pour apprendre une combinaison de k-PPV à la fois performante et robuste. Nous confirmons expérimentalement ces résultats sur 20 jeux de données de référence : P-MinCq améliore significativement l'approche k-PPV classique, sa version symétrisée [NSB03], ainsi qu'un MinCq sur les mêmes votants. De plus, pour les problèmes en grande dimension, P-MinCq s'avère être plus robuste au sur-apprentissage. En outre, il se montre compétitif avec l'algorithme d'apprentissage de métriques LMMN [WS09] et ses performances peuvent encore être améliorées en le combinant avec la distance apprise par LMMN. Finalement, nous mettons en évidence les bonnes performances de P-MinCq sur une tâche de reconnaissance d'objets.

Le papier s'organise comme suit. La sec. 2 présente MinCq [LMR11] et sa théorie. Nous mettons en évidence les limites de MinCq pour les k-PPV en sec. 3. En sec. 4, nous décrivons P-MinCq, l'extension de MinCq aux distributions **P**-alignées. La borne en généralisation pour les schémas de compression est dérivée en sec. 5. La sec. 6 définit un **P**-alignement spécifique aux PPV. Nous expérimentons P-MinCq en sec. 7 et concluons en sec. 8.

2 MinCq : notations et théorie

Nous nous plaçons ici dans le cadre de l'algorithme MinCq [LMR11] qui vise à apprendre un vote de majorité sur un ensemble de votants à valeurs réelles en classification binaire.

Soit $X \subseteq \mathbb{R}^d$ l'espace d'entrée de dimension d et $Y = \{-1, +1\}$ l'ensemble de sortie. S est un échantillon d'apprentissage composé de m exemples $(\mathbf{x}, y)$ tirés *i.i.d.* selon une distribution (fixée, inconnue) D sur $X \times Y$. $(D)^m$ correspond à la distribution de S. MinCq a été conçu dans le cadre de la théorie PAC-Bayésienne [McA99]. Étant donné un ensemble de votants $\mathcal{H}$, cette théorie se base sur une distribution prior P et une distribution posterior Q sur $\mathcal{H}$. P modélise l'information *a priori* sur la pertinence des votants : ceux a priori plus performants ont un poids plus élevé selon P. En considérant l'information fournie par S, le but de l'apprenant est alors d'adapter P afin d'obtenir un posterior Q impliquant un vote de majorité Q-pondéré de risque réel faible, défini comme suit.

Définition 1. *Soit $\mathcal{H} = \{h_1, \ldots, h_n\}$ un ensemble de n votants de X vers $\mathbb{R}$. Soit Q une distribution sur $\mathcal{H}$. Un vote de majorité Q-pondéré B_Q est défini par :*

$$\forall \mathbf{x} \in X, B_Q(\mathbf{x}) = \text{sign}\left[\mathop{\mathbf{E}}_{h \sim Q} h(\mathbf{x})\right] = \text{sign}\left[\sum_{h \in \mathcal{H}} Q(h) h(\mathbf{x})\right].$$

Son risque réel (ou erreur réelle) $R_D(B_Q)$ sur les paires $(\mathbf{x}, y)$ i.i.d. selon la distribution D est :

$$R_D(B_Q) = \mathop{\mathbf{E}}_{(\mathbf{x},y) \sim D} \mathbf{I}\left[B_Q(\mathbf{x}) \neq y\right],$$

où $\mathbf{I}[.]$ est une fonction indicatrice.

Dans [LMR11, LLM+07], les auteurs font le lien entre le risque $R_D(B_Q)$ et la notion suivante de Q-marge qui modélise la confiance de $B_Q(\cdot)$ en son étiquetage.

Définition 2 ([LMR11])**.** *La Q-marge de $(\mathbf{x}, y)$ est :*

$$\mathcal{M}_Q(\mathbf{x}, y) = y \mathop{\mathbf{E}}_{h \sim Q} h(\mathbf{x}).$$

Le premier moment $\mathcal{M}_Q^D$ de la Q-marge est :

$$\mathcal{M}_Q^D = \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} \mathcal{M}_Q(\mathbf{x},y) = \mathop{\mathbf{E}}_{h\sim Q} \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} yh(\mathbf{x}),$$

alors que le second moment $\mathcal{M}_{Q^2}^D$ est défini ainsi :

$$\mathcal{M}_{Q^2}^D = \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} (\mathcal{M}_Q(\mathbf{x},y))^2 = \mathop{\mathbf{E}}_{(h,h')\sim Q^2} \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} h(\mathbf{x})h'(\mathbf{x}).$$

$B_Q(\cdot)$ classe correctement un exemple $(\mathbf{x},y)$ lorsque sa Q-marge est strictement positive. On a alors :

$$R_D(B_Q) = \mathop{\mathbf{Pr}}_{(\mathbf{x},y)\sim D} (\mathcal{M}_Q(\mathbf{x},y) \leq 0)\,. \qquad (1)$$

Pour finir, introduisons les notations suivantes :

$$\mathcal{M}_h^D = \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} yh(\mathbf{x}) \;\text{ et }\; \mathcal{M}_{h,h'}^D = \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} h(\mathbf{x})h'(\mathbf{x}). \quad (2)$$

Si l'on considère $S\sim(D)^m$ plutôt que la distribution D, nous obtenons le risque empirique $R_S(B_Q)$, les premier et second moments empiriques de la Q-marge $\mathcal{M}_Q^S$ et $\mathcal{M}_{Q^2}^S$, et les quantités associées $\mathcal{M}_h^S$ et $\mathcal{M}_{h,h'}^S$.

Nous rappelons dans ce qui suit trois résultats issus de [LMR11, LLM+07] et qui constituent les fondements de nos contributions. Le premier majore $R_D(B_Q)$ par la C-borne (théo. 1). Celle-ci relie le risque réel aux premier et second moments de la Q-marge. Le deuxième résultat garantit la consistence de la minimisation empirique de la C-borne pour apprendre Q, *i.e.* un vote de majorité Q-pondéré (théo. 2). Le troisième montre que cet apprentissage peut se faire de manière optimale en résolvant un programme quadratique simple, MinCq.

La C-borne s'obtient à l'aide de l'éq. (1), en appliquant l'inégalité de Cantelli-Chebychev [DGL96] à la variable aléatoire $\mathcal{M}_Q(\mathbf{x},y)$.

Théorème 1 (C-borne [LMR11])**.** *Pour toute distribution Q sur une classe $\mathcal{H}$ et pour toute distribution D sur $X\times Y$, si $\mathcal{M}_Q^D > 0$ alors $R_D(B_Q) \leq C_Q^D$ où :*

$$C_Q^D = \frac{\mathbf{Var}_{(\mathbf{x},y)\sim D}(\mathcal{M}_Q(\mathbf{x},y))}{\mathbf{E}_{(\mathbf{x},y)\sim D}(\mathcal{M}_Q(\mathbf{x},y))^2} = 1 - \frac{(\mathcal{M}_Q^D)^2}{\mathcal{M}_{Q^2}^D}.$$

Nous notons $C_Q^S = 1 - \frac{(\mathcal{M}_Q^S)}{\mathcal{M}_{Q^2}^S}$ son estimation sur S.

L'intérêt de la C-borne est que sa minimisation permet d'apprendre Q afin d'obtenir un vote de majorité B_Q de risque réel faible. Pour justifier cette stratégie, les auteurs ont dérivé une borne en généralisation PAC-Bayésienne sur C_Q^D. L'idée est de se focaliser sur des distributions quasi-uniformes Q sur un ensemble auto-complémenté de $2n$ votants $\mathcal{H} = \{h_1,\ldots,h_n,h_{n+1},\ldots,h_{2n}\}$, où pour tout $k\in\{1,\ldots,n\}, h_{k+n} = -h_k$ (auto-complementation) et $Q(h_k) + Q(h_{k+n}) = \frac{1}{n}$ (quasi-uniformité) [2]. Les auteurs affirment que cette hypothèse n'est pas une trop forte restriction et permet de caractériser les situations pour lesquelles on donne le même *a priori* à chaque votant (*i.e.* le prior est non-informatif). De plus, les distributions quasi-uniformes ont deux avantages. D'une part, elles permettent de s'affranchir du terme classique relié à la complexité [3] de $\mathcal{H}$ (ce terme est parfois difficile à optimiser et peut amener à une mauvaise régularisation) [LMR11]. D'autre part, cette contrainte joue le rôle d'une régularisation donnant le même *a priori* à chaque votant et apparaît être une solution concrète et naturelle au sur-apprentissage.

La borne en généralisation est obtenue en prenant la minoration (*resp.* majoration) de $\mathcal{M}_Q^D$ et la majoration (*resp.* minoration) de $\mathcal{M}_{Q^2}^D$ du théorème suivant.

Théorème 2 ([LMR11])**.** *Pour toute distribution D sur $X\times Y$, pour tout $m\geq 8$, pour tout ensemble auto-complémenté $\mathcal{H}$ de votants réels bornés par B, pour tout $\delta\in(0,1]$, avec une proba. d'au moins $1-\delta$ sur le choix de $S\sim(D)^m$, pour tout Q quasi-uniforme sur $\mathcal{H}$ on a :*

$$\left|\mathcal{M}_Q^D - \mathcal{M}_Q^S\right| \leq \frac{2B\sqrt{\ln\frac{2\sqrt{m}}{\delta}}}{\sqrt{2m}} \;\text{ et }\; \left|\mathcal{M}_{Q^2}^D - \mathcal{M}_{Q^2}^S\right| \leq \frac{2B^2\sqrt{\ln\frac{2\sqrt{m}}{\delta}}}{\sqrt{2m}}.$$

Sans perte de généralité, on obtient la proposition suivante :

Proposition 1 ([LMR11])**.** *Pour tout $\mu\in(0,1]$ et pour toute distribution Q sur $\mathcal{H}$ associée à une Q-marge empirique $\mathcal{M}_Q^S \geq \mu$, il existe une distribution quasi-uniforme Q' sur $\mathcal{H}$ de Q-marge empirique égale à μ, telle que Q et Q' induisent le même vote de majorité et la même valeur empirique de la C-borne, i.e. :*

$$\mathcal{M}_{Q'}^S = \mu,\; B_{Q'} = B_Q,\; C_{Q'}^S = C_Q^S,\; \text{et } C_{Q'}^D = C_Q^D.$$

En se basant sur les théos. 1, 2 et sur la prop. 1, les auteurs suggèrent de minimiser la C-borne empirique sous la contrainte $\mathcal{M}_Q^S \geq \mu$. Grâce à l'hypothèse de quasi-uniformité, ils montrent que ceci est équivalent à résoudre un programme quadratique simple mettant en jeu uniquement les n premiers votants de $\mathcal{H}$. MinCq est décrit dans l'algo. 1. Il minimise le dénominateur $\mathcal{M}_{Q^2}^S$, *i.e.* le second moment de la Q-marge (éq. (3)), sous les contraintes $\mathcal{M}_Q^S = \mu$ (éq. (4)), *i.e.* le premier moment est fixé, et Q est quasi-uniforme (éq. (5)).
Finalement le vote de majorité Q-pondéré appris est :

$$B_Q(\mathbf{x}) = \text{sign}\left[\sum_{k=1}^{n}\left(2Q_k - \frac{1}{n}\right)h_k(\mathbf{x})\right].$$

2. Pour simplifier, nous noterons $Q(h_k)$ par Q_k.

3. En PAC-Bayes, ce terme est relié à la divergence entre la distribution Q et la distribution prior P, mesurée via la divergence de Kullback-Leibler. Voir [LMR11] pour plus de détails.

Algorithme 1 MinCq : programme quadratique apprenant un vote de majorité Q-pondéré sur $\mathcal{H}$ auto-complémenté, sous une contrainte de quasi-uniformité.

entrée Un échantillon S, les n premiers votants d'un ensemble $\mathcal{H}$ auto-complémenté, une marge $\mu > 0$

sortie Un vecteur "posterior" $\mathbf{Q} = (Q_1, \ldots, Q_n)^T$

$$\textbf{Résoudre} \quad \operatorname*{argmin}_{\mathbf{Q}} \ \mathbf{Q}^T \mathbf{M}_S \mathbf{Q} - \mathbf{A}_S^T \mathbf{Q}, \tag{3}$$

$$\textbf{s.c.} \quad \mathbf{m}_S^T \mathbf{Q} = \frac{\mu}{2} + \frac{1}{2n} \sum_{k=1}^{n} \mathcal{M}_{h_k}^S, \tag{4}$$

$$\forall k \in \{1, \ldots, n\}, \quad 0 \leq Q_k \leq 1/n, \tag{5}$$

où $\mathbf{Q} = (Q_1, \ldots, Q_n)^T$ est le vecteur des n premiers poids Q_k, $\mathbf{M}_S$ la matrice $n \times n$ composée de $\mathcal{M}_{h_k,h_{k'}}^S$ pour $k, k' \in \{1, \ldots, n\}$ (*c.f.* éq. (2)) et :

$\mathbf{m}_S = \left(\mathcal{M}_{h_1}^S, \ldots, \mathcal{M}_{h_n}^S \right)^T$,

$\mathbf{A}_S = \left(\frac{1}{nm} \sum_{k=1}^{n} \mathcal{M}_{h_1,h_k}^S, \ldots, \frac{1}{nm} \sum_{k=1}^{n} \mathcal{M}_{h_n,h_k}^S \right)^T$.

3 MinCq et PPV : les limites

Au premier abord, MinCq semble être une solution intéressante pour contrer les limitations liées aux k-PPV. D'une part, la théorie stipule que plus k est élevé, meilleure est la convergence vers le risque bayésien optimal. Or, ceci n'est vrai qu'asymptotiquement et, en pratique, le choix de k requiert une attention particulière[4]. Optimiser un vote de majorité de classifieurs k-PPV[5] ($k = \{1,2,\ldots\}$) permettrait donc de s'affranchir de régler k. D'autre part, en faisant usage du cadre PAC-Bayésien, la minimisation de la C-borne semble fournir des garanties en généralisation qui ne pourraient pas être obtenues avec un algorithme standard des k-PPV avec un échantillon de taille finie. Nous avons réalisé une étude expérimentale préliminaire pour comparer un classifieur k-PPV classique (pour lequel, k a été choisi par validation croisée) avec MinCq (*c.f.* sec. 7 pour le protocole). Sur 20 jeux de données, MinCq atteint une erreur moyenne de 18.18% contre 17.88% pour k-PPV (*c.f.* tab. 2). Un test de Student ne distingue pas les deux approches. Ceci se confirme par un test de signe avec un résultat *win/loss/tie* égal à 7/6/7 et une p-valeur d'environ 0.5 (*c.f.* fig. 1). Cette série d'expériences montre clairement que MinCq n'améliore pas l'approche k-PPV. Ce résultat a priori surprenant peut en fait être expliqué par l'hypothèse non-informative de quasi-uniformité sur Q qui n'est pas pertinente quand les votants n'ont pas la même importance. Or, pour des raisons évidentes, un voisinage proche (*i.e.* un petit k) offre naturellement une meilleure information discriminative, dans le cas d'échantillons finis, qu'une règle de décision des k-PPV avec k très grand. En outre, d'un point de vue théorique, les bornes présentées dans la section précédente ne sont pas vérifiables si les votants dépendent de l'échantillon d'apprentissage (on parle de schéma de compression).

4. Voir sec. 6.2 pour plus de détails sur le choix de k.

5. D'autres ensembles de votants pourraient être considérés. *e.g.* $n^{\text{ème}}$ voisin pourrait correspondre au $n^{\text{ième}}$ votant.

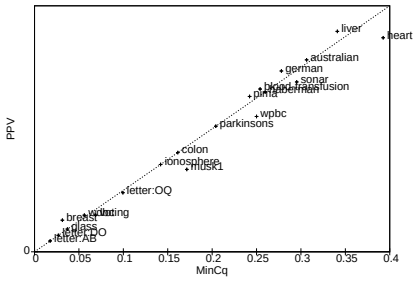


FIGURE 1 – **MinCq** *vs* **PPV**. Un point correspond au taux d'erreur sur un jeu de données. Un point au-dessus de la bissectrice est en faveur de MinCq.

Pour contrer ces limitations, nous proposons d'étendre MinCq. En sec. 4, nous généralisons la contrainte de quasi-uniformité à une contrainte de **P**-alignement : étant donnée une contrainte *a priori* $\mathbf{P}$, Q doit être une distribution $\mathbf{P}$-alignée (*i.e.* proche de $\mathbf{P}$). $\mathbf{P}$ va ainsi nous permettre de modéliser la connaissance sur la pertinence des votants. En sec.5 nous généralisons les preuves de convergence de [LMR11] aux schémas de compression PAC-Bayésien [GHST05].

4 De quasi-uniforme à P-alignée

Soit $\mathcal{H} = \{h_1, \ldots, h_{2n}\}$ un ensemble de votants auto-complémentés ($\forall i \in \{1, \ldots, n\}, h_{i+n} = -h_i$). Plutôt que de restreindre la distribution Q sur $\mathcal{H}$ à la quasi-uniformité ($Q_i + Q_{i+n} = \frac{1}{n}$), nous la contraignons au $\mathbf{P}$-alignement : $\forall i \in \{1, \ldots, n\}, Q_i + Q_{i+n} = P_i$, où $(P_1, \ldots, P_n)^T = \mathbf{P}$ tel que $\sum_{i=1}^{n} P_i = 1$ et $\forall i \in \{1, \ldots, n\}$, $0 \leq P_i \leq 1$. $\mathbf{P}$ joue donc le rôle d'*a priori* sur les votants. Notons que la quasi-uniformité est un cas particulier du $\mathbf{P}$-alignement où : $\forall i \in \{1, \ldots, n\}, P_i = \frac{1}{n}$. Par la suite, nous montrons que le $\mathbf{P}$-alignement ne restreint pas l'ensemble des votes de majorité possibles. Puis, nous introduisons P-MinCq qui optimise la C-borne dans ce contexte.

4.1 Expressivité du P-alignement

D'après [LMR11], la quasi-uniformité ne restreint pas l'ensemble des votes de majorité possibles. Nous généralisons ce résultat à toute distribution **P**-alignée sur un ensemble auto-complémenté $\mathcal{H}$ de votants.

Proposition 2. *Pour tout Q sur $\mathcal{H}$, il existe une distribution $\mathbf{P}$-alignée Q' sur $\mathcal{H}$ auto-complémenté telle que : $B_{Q'} = B_Q$, $C^S_{Q'} = C^S_Q$, et $C^D_{Q'} = C^D_Q$.*

Démonstration. En annexe. □

Puisque le **P**-alignement est une généralisation de la quasi-uniformité, la prop. 1 reste valide : sous la contrainte $\mathcal{M}^S_Q{=}\mu$, la C-borne est optimisable en minimisant le second moment $\mathcal{M}^S_{Q^2}$ de la Q-marge via le programme quadratique P-MinCq énoncé ci-dessous.

4.2 L'algorithme P-MinCq

Le théo. 2 reste vrai pour toute distribution **P**-alignée sur des votants indépendants des données. En effet, la preuve fait simplement appel à la contrainte de **P**-alignement [6] : $Q_i + Q_{i+n} = P_i + P_{i+n}$. Notre extension P-MinCq est décrite dans l'algo. 2 (et sa dérivation en annexe). Le **P**-alignement permet, comme pour MinCq, d'uniquement faire intervenir les n premiers votants de $\mathcal{H}$ lors de la résolution du programme.

Algorithme 2 P-MinCq : programme quadratique apprenant un vote de majorité Q-pondéré sur $\mathcal{H}$ auto-complémenté, sous une contrainte de **P**-alignement.

entrée Un échantillon S, les n premiers votants d'un ensemble $\mathcal{H}$ auto-complémenté, une marge $\mu > 0$, un vecteur "prior" $\mathbf{P} = (P_1, \ldots, P_n)^T$, une matrice $\mathbf{M}_S$ de taille $n{\times}n$ composée des éléments $\mathcal{M}^S_{h_i,h_{i'}}$(éq. (2))

sortie Un vecteur "posterior" $\mathbf{Q} = (Q_1, \ldots, Q_n)^T$

$$\textbf{Résoudre} \quad \underset{\mathbf{Q}}{\text{argmin}} \ (\mathbf{Q}-\mathbf{P})^T \mathbf{M}_S \mathbf{Q}, \tag{6}$$

$$\textbf{s.c.} \quad \mathbf{m}_S{}^T(2\mathbf{Q}-\mathbf{P}) = \mu, \tag{7}$$

$$\forall i \in \{1,\ldots,n\},\ 0 \le Q_i \le P_i, \tag{8}$$

où $\mathbf{m}_S{}^T = (\mathcal{M}_{h_1}, \ldots, \mathcal{M}_{h_n})^T$.

La fonction objectif (éq. (6)) minimise le second moment de la Q-marge alors que l'éq. (7) contraint le premier moment à μ. La partie gauche de l'éq. (7) est une moyenne pondérée des marges individuelles $\mathcal{M}_{h_i}$ (les poids valent $2Q_i{-}P_i$). Étant donnée **P**, la dernière éq. (8) impose de considérer uniquement des distributions **P**-alignées. Le vote de majorité appris est alors :

$$B_Q(\mathbf{x}) = \text{sign}\left[\sum_{i=1}^{n} (2Q_i - P_i)\, h_i(\mathbf{x})\right].$$

Concernant les capacités en généralisation de P-MinCq, le théo. 2 n'est pas valide lorsque les votants dépendent des données d'apprentissage, *i.e.* si l'on se place dans le cadre des schémas de compression. Nous généralisons le théo. 2 à ce cadre en utilisant des techniques issues de [LM07] comme intuité dans [LMR11]

5 Borne en généralisation pour les schémas de compression

Nous dérivons ici une preuve de consistence lorsque les votants sont définis à partir d'exemples de l'échantillon d'apprentissage S. Ce résultat est à la fois valide pour MinCq et pour son extension P-MinCq.

5.1 Cadre général

Un schéma de compression [FW95] est un algorithme d'apprentissage travaillant sur un ensemble de classifieurs dépendant des données. Un classifieur est alors représenté par deux éléments : une séquence d'exemples, appelée séquence de compression, et un message représentant l'information supplémentaire nécessaire pour trouver le classifieur à partir de la séquence. On définit ainsi une fonction de reconstruction renvoyant un classifieur à partir d'une séquence de compression et d'un message. La définition d'un schéma de compression est la suivante.

Définition 3. *Soit $S \sim (D)^m$ un échantillon. $\mathbf{J}_m{=}\bigcup_{i=1}^m \{(j_1,\ldots,j_i){\in}\{1,\ldots,m\}^i\}$ est l'ensemble des vecteurs d'indices possibles. Étant donnés une famille d'hypothèses $\mathcal{H}^S$ et un vecteur d'indices $\mathbf{j} \in \mathbf{J}_m$, la séquence de compression $S_{\mathbf{j}} = \{(\mathbf{x}_j, y_j)\}_{j\in\mathbf{j}}$ est définie comme la sous-séquence indicée par $\mathbf{j}$. Un algorithme $\mathcal{A} : (X\times Y)^{(\infty)} \mapsto \mathcal{H}^S$ est un schéma de compression si et seulement s'il existe un triplet $(\mathcal{C}, \mathcal{R}, \omega)$ tel que pour tout échantillon S, on ait : $\mathcal{A}(S) = \mathcal{R}\left(S_{\mathcal{C}(S)}, \omega\right)$, où $\mathcal{C} : (X\times Y)^{(\infty)} \mapsto \bigcup_{m=1}^{\infty} \mathbf{J}_m$ est la fonction de compression, $\mathcal{R} : (X\times Y)^{(\infty)} \times \Omega_{S_{\mathcal{C}(S)}} \mapsto \mathcal{H}^S$ la fonction de reconstruction et ω un message choisi dans l'ensemble $\Omega_{S_{\mathcal{C}(S)}}$ (défini a priori) de tous les messages pouvant être fournis avec la séquence $S_{\mathcal{C}(S)}$.*

En d'autres termes, un schéma de compression est une fonction de reconstruction $\mathcal{R}$ associant une séquence

6. Voir [LMR11] pour plus de détails.

de compression $\mathcal{C}(S)=S_{\mathbf{j}}$ à un ensemble $\mathcal{H}^S$ de fonctions $h^{\omega}_{S_{\mathbf{j}}}$ telle que $\mathcal{A}(S)=\mathcal{R}\left(S_{\mathbf{j}},\omega\right)=h^{\omega}_{S_{\mathbf{j}}}$. Par exemple les classifieurs PPV sont reconstructibles uniquement à partir d'une séquence de compression encodant les PPV [FW95, GHST05], alors que d'autres classifieurs, comme les *decision list machines* [MS05], requièrent une séquence de compression ainsi qu'un message.

5.2 Borne en généralisation

Soit $S_{\mathbf{j}}$ une séquence de compression composée de $|\mathbf{j}|$ exemples issus de S. Dans le contexte d'un schéma de compression PAC-Bayésien, les erreurs $R_P(\cdot)$ et $R_S(\cdot)$ peuvent être biaisées par ces éléments : il est donc préférable de calculer le risque empirique $R_S(\cdot)$ à partir de $S\backslash S_{\mathbf{j}}$ [LM07]. Cependant, la stratégie proposée par [GLL$^+$11], pour dériver une borne, prend en compte le biais. En suivant cette stratégie, étant donné un échantillon S, nous considérons $\mathcal{H}^S$ l'ensemble de tous les classifieurs possibles $h^{\omega}_{S_{\mathbf{j}}}=\mathcal{R}(S_{\mathbf{j}},\omega)$ tel que $\omega\in\Omega_{S_{\mathbf{j}}}$. Nous notons $Q_{\mathbf{J}_m}(\mathbf{j})$ la probabilité qu'une séquence de compression $S_{\mathbf{j}}$ soit choisie par Q, et $Q_{S_{\mathbf{j}}}(\omega)$ la probabilité de choisir un message ω sachant $S_{\mathbf{j}}$. Alors :

$$Q_{\mathbf{J}_m}(\mathbf{j})=\int_{\omega\in\Omega_{S_{\mathbf{j}}}} Q(h^{\omega}_{S_{\mathbf{j}}})d\omega,\quad\text{et}\quad Q_{S_{\mathbf{j}}}(\omega)=Q(h^{\omega}_{S_{\mathbf{j}}}|S_{\mathbf{j}}).$$

En PAC-Bayes, une borne en généralisation dépend de la distribution prior P sur $\mathcal{H}^S$. Ce prior est supposé connu avant d'observer S, impliquant que P et S sont indépendants. Or, les votants de $\mathcal{H}^S$ dépendent de S et empêchent une telle connaissance *a priori*. Ce problème peut être contré, selon le principe de [LM07, GLL$^+$11], en considérant une distribution prior définie par le couple : $\left(P_{\mathbf{J}_m},(P_{S_{\mathbf{j}}})_{\mathbf{j}\in\mathbf{J}_m}\right)$, où $P_{\mathbf{J}_m}$ est une distribution sur $\mathbf{J}_m$ et $P_{S_{\mathbf{j}}}$ est une distribution sur $\Omega_{S_{\mathbf{j}}}$, pour toutes les séquences $S_{\mathbf{j}}$. Ainsi la distribution P indépendante de S correspond à la distribution sur $\mathcal{H}^S$ associée au prior $\left(P_{\mathbf{J}_m},(P_{S_{\mathbf{j}}})_{\mathbf{j}\in\mathbf{J}_m}\right)$ et est définie par : $P(h^{\omega}_{S_{\mathbf{j}}})=P_{\mathbf{J}_m}P_{S_{\mathbf{j}}}(\omega)$.

Définition 4. *Pour un schéma de compression, la Q-marge d'un exemple $(\mathbf{x},y)$ est donnée par :*

$$\mathcal{M}_Q(\mathbf{x},y)=y\mathop{\mathbf{E}}_{h^{\omega}_{S_{\mathbf{j}}}\sim Q}\ h^{\omega}_{S_{\mathbf{j}}}(\mathbf{x}).$$

Le premier moment $\mathcal{M}^D_Q$ et le second moment $\mathcal{M}^D_{Q^2}$ de la Q-marge sont définis comme précédemment par :

$$\mathcal{M}^D_Q=\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D}\mathcal{M}_Q(\mathbf{x},y),\ \mathcal{M}^D_{Q^2}=\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D}\left(\mathcal{M}_Q(\mathbf{x},y)\right)^2.$$

$\mathcal{H}^S$ est un ensemble auto-complémenté de votants : le complémentaire de $h^{\omega}_S\in\mathcal{H}^S$ est $-h^{\omega}_S$. Ainsi, l'ensemble des messages associé est $\Omega_S\times\{+,-\}$ et : $\forall\sigma\in\Omega_S, h^{(\sigma,+)}_S=-h^{(\sigma,-)}_S$. Le résultat principal de cette section est :

Théorème 3. *Pour toute distribution D sur $X\times Y$, pout tout $m\geq 8$, pour tout $\delta\in(0,1]$, avec une probabilité d'au moins $1-\delta$ sur le choix de $S\sim(D)^m$, pour tout $\mathcal{H}^S$ auto-complémenté de votants réels bornés par B et de taille de séquence de compression au plus $|\mathbf{j}^{\max}|<\frac{m}{2}$ et pour toute distribution $\mathbf{P}$-alignée Q sur $\mathcal{H}^S$, on a :*

$$\left|\mathcal{M}^D_Q-\mathcal{M}^S_Q\right|\leq\frac{2B}{\sqrt{2(m-|\mathbf{j}^{\max}|)}}\sqrt{\frac{|\mathbf{j}^{\max}|}{B\delta}+\ln\left(\frac{2\sqrt{m}}{\delta}\right)},$$

$$\left|\mathcal{M}^D_{Q^2}-\mathcal{M}^S_{Q^2}\right|\leq\frac{2B^2}{\sqrt{2(m-2|\mathbf{j}^{\max}|)}}\sqrt{\frac{2|\mathbf{j}^{\max}|}{B^2\delta}+\ln\left(\frac{2\sqrt{m}}{\delta}\right)}.$$

Démonstration. En annexe, inspirée de la preuve du théo. 2 (voir [LMR11]). □

Si les votants sont indépendants des données, *i.e.* si $|\mathbf{j}^{\max}|=0$, on retrouve le théo. 2. Comme attendu, plus les séquences de compression sont grandes, plus $|\mathbf{j}^{\max}|$ est élevé et moins la borne est précise. Ainsi, pour préserver la consistence du processus d'apprentissage, cette valeur ne doit pas être trop élevée.

6 P-MinCq et les k-PPV

6.1 Sur les capacités en généralisation

Rappelons que les classifieurs k-PPV dépendent des exemples d'apprentissage. Cependant, quelle que soit la valeur de k, pour un échantillon d'apprentissage de taille m, la séquence de compression est de taille m rendant invalides les bornes (si la taille maximale des séquences de compression vaut $|\mathbf{j}^{\max}|>=\frac{m}{2}$ alors le dénominateur des bornes vaut 0). Pour éviter cette forme indéterminée, $|\mathbf{j}^{\max}|$ peut considérablement être réduit par des techniques de sélection de prototypes ou de réduction de bases [DHS01], qui permettent de supprimer de la séquence les exemples n'intervenant jamais dans la décision sur les données de test. Dans ce cas, chaque k-PPV utilise donc sa propre séquence de compression : un sous-ensemble de l'échantillon d'apprentissage. Finalement, les garanties en généralisation valides sont celles de notre théo. 3 (ici $B=1$).

6.2 Le choix de k pour les k-PPV

La théorie classique des k-PPV stipule que plus k augmente, plus la convergence vers le risque de Bayes optimal est forte. Cependant, cette propriété n'est vraie qu'asymptotiquement, quand $m\to+\infty$. En pratique, m est limité et k doit être choisi avec précaution. D'une part, plus k est grand, plus l'estimation de densité est pertinente. Mais d'autre part, seuls les voisinages les plus proches, avec un petit k, amènent à

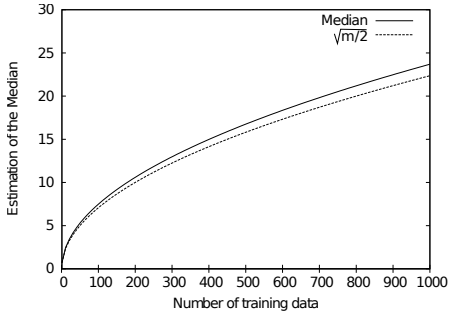


FIGURE 2 – Comparaison de la médiane de $\mathbf{P}$ et $\sqrt{m/2}$.

une règle de classification correcte. Dans la littérature, différentes études théoriques et empiriques ont été menées pour analyser ce compromis entre grande et petite valeur de k. Une solution [DHS01, CM11] consiste à faire appel à une faible fraction d'exemples environ égale à $\sqrt{m/|Y|}$ de voisins.

6.3 Une contrainte fondée

P-MinCq offre un contexte original pour les k-PPV, en combinant différents k-PPV ($\forall k = \{1, \ldots, m\}$). Au lieu de régler k, nous définissons une contrainte *a priori* de $\mathbf{P}$-alignement sur les votants. Comme mentionné dans [DGL96], les voisinages les plus proches apportent plus d'information dans une combinaison de k-PPV. Suivant cette recommandation, nous proposons la contrainte $\mathbf{P}$ suivante (à normaliser) :

$$\forall k \geq 1, \quad P_k = 1/k\,. \tag{9}$$

$\mathbf{P}$ concentre ses poids sur les votants définis à partir d'une faible fraction d'exemples, mais prend aussi en compte (dans une moindre mesure) l'information portée par l'ensemble des voisinages. Nous justifions, dans ce qui suit, ce choix en établissant une relation entre l'éq. (9) et sa médiane M (le nombre de voisins impliqués dans les votants accumulant la moitié de la densité). Alors que dans le cas d'une distribution continue, le calcul de M est aisé, le cas discret (qui nous intéresse, *i.e.* où $x \in \{1, \ldots, m\}$) requiert une approximation. On note : $H_M = \sum_{x=1}^{M} \frac{1}{x}$ et $H_m = \sum_{x=1}^{m} \frac{1}{x}$, la somme des termes d'une série harmonique n'admettant aucune solution analytique. Or, des sommes partielles de séries nous permettent de définir H_n par :

$$\forall n,\ H_n = \sum_{x=1}^{n} \frac{1}{x} = \ln(n) + \gamma + \epsilon_n,$$

où $\gamma \simeq 0.5772156$ est la constante d'Euler-Mascheroni [7]

7. La constante d'Euler-Mascheroni est la limite de la différence entre la série harmonique et le log. naturel.

TABLE 1 – Propriétés des 20 jeux de données.

Nom	# Attributs	Taille de S	Taille en test
australian	14	345	345
blood	4	374	374
breast	9	349	350
colon	2000	31	31
german	24	500	500
glass	9	107	107
haberman	3	153	153
heart	13	135	135
ionosphere	34	175	176
letterAB	16	297	1192
letterDO	16	297	1193
letterOQ	16	291	1166
liver	6	172	173
musk1	166	238	238
parkinsons	22	97	98
pima	8	384	384
sonar	60	104	104
voting	16	217	218
wdbc	30	284	285
wpbc	33	99	99

et $\epsilon_n \simeq \frac{1}{2n}$. Par conséquent, on a :

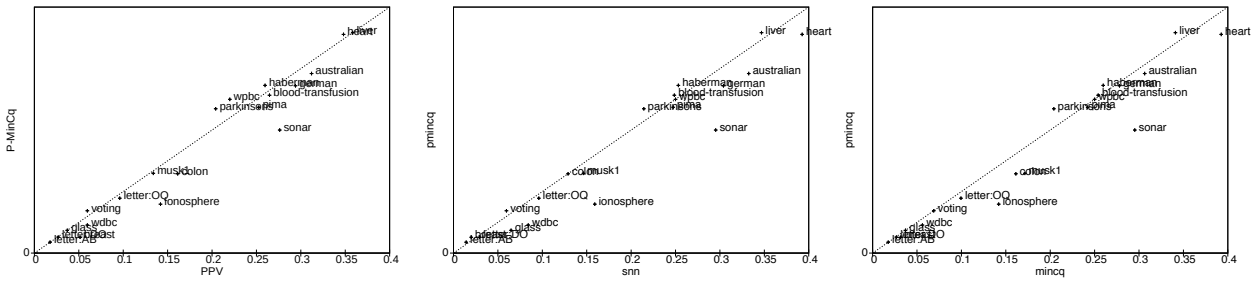
$$\begin{aligned}
H_M = \tfrac{1}{2} H_m &\Leftrightarrow \textstyle\sum_{x=1}^{M} \frac{1}{x} = \frac{1}{2} \sum_{x=1}^{m} \frac{1}{x} \\
&\Leftrightarrow \ln(M) + \gamma + \tfrac{1}{2M} = \tfrac{1}{2}(\ln(m) + \gamma) + \tfrac{1}{4m} \\
&\Leftrightarrow \ln(M) = \ln(\sqrt{m}) - \tfrac{\gamma}{2} + \tfrac{1}{4m} - \tfrac{1}{2M} \\
&\Leftrightarrow \ln(M) \leq \ln(\sqrt{m}) - \tfrac{\gamma}{2} - \tfrac{1}{4m} \\
&\quad \text{(puisque l'éq. (9) implique } M \leq m/2) \\
&\Leftrightarrow M \leq \sqrt{m\ e^{(-\gamma)} e^{(-1/4m)}} \simeq \sqrt{m/2}. \quad (10)
\end{aligned}$$

Dans le cas discret, l'éq. (10) indique que l'approximation de la médiane de $\mathbf{P}$ est très proche de $\sqrt{m/2}$ (fig. 2), qui n'est autre que la valeur suggérée pour k dans la règle des k-PPV en classification binaire. Nous avons donc établi ici une relation étroite entre la sélection classique de k et notre contrainte $\mathbf{P}$ (dans le cas d'un vote de majorité sur un ensemble de classifieurs k-PPV). La section suivante présente une étude expérimentale permettant de valider ce choix.

7 Expérimentations

Dans cette section nous réalisons une étude de P-MinCq pour les k-PPV, comme décrit dans la sec. 6. Nous nous comparons à quatre méthodes.

- L'algorithme standard des k-PPV (PPV).
- Un k-PPV symétrique (SNN : *Symmetric Nearest Neighbor*) [NSB03], pour lequel un exemple $\mathbf{x}$ est étiqueté par la classe majoritaire parmi les exemples appartenant au k-voisinage de $\mathbf{x}$ (comme pour PPV) et ceux incluant $\mathbf{x}$ dans leur propre k-voisinage.
- L'algorithme LMNN (*Large Margin Nearest Neighbor*) [WS09] qui apprend essentiellement une distance de Mahalanobis en optimisant l'erreur du k-PPV sur



(a) **P-MinCq** *vs* **PPV** (b) **P-MinCq** *vs* **SNN** (c) **P-MinCq** *vs* **MinCq**

FIGURE 3 – Comparaison de P-MinCq, PPV, SNN et MinCq. Chaque point sur le graphique correspond au taux d'erreur des deux algorithmes comparés. Un point en-dessous de la bissectrice est en faveur de P-MinCq.

TABLE 2 – Taux d'erreurs sur les 20 jeux de données.

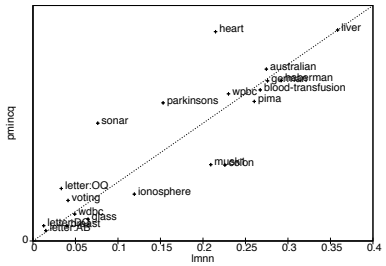
Jeu de données	PPV	SNN	LMNN	MinCq	P-MinCq
australian	0.3121	0.3324	0.2746	0.3064	0.2919
blood	0.2647	0.2487	0.2674	0.2540	0.2567
breast	0.0514	0.0200	0.0400	0.0314	0.0257
colon	0.1613	0.1290	0.2258	0.1613	0.1290
german	0.2940	0.3040	0.2760	0.2780	0.2720
glass	0.0370	0.0648	0.0648	0.0370	0.0370
haberman	0.2597	0.2532	0.2922	0.2597	0.2727
heart	0.3481	0.3926	0.2148	0.3926	0.3556
ionosphere	0.1420	0.1591	0.1193	0.1420	0.0795
letter :AB	0.0176	0.0143	0.0151	0.0176	0.0176
letter :DO	0.0268	0.0293	0.0126	0.0268	0.0260
letter :OQ	0.0961	0.0961	0.0334	0.0995	0.0892
liver	0.3584	0.3468	0.3584	0.3410	0.3584
musk1	0.1339	0.1464	0.2092	0.1715	0.1297
parkinsons	0.2041	0.2143	0.1531	0.2041	0.2347
pima	0.2526	0.2474	0.2604	0.2422	0.2370
sonar	0.2762	0.2952	0.0762	0.2952	0.2000
voting	0.0596	0.0596	0.0413	0.0688	0.0688
wdbc	0.0596	0.0842	0.0491	0.0561	0.0456
wpbc	0.2200	0.2500	0.2300	0.2500	0.2500
Erreur Moyenne	0.1788	0.1844	0.1607	0.1818	0.1689
Rang Moyen	2.9	3.1	2.65	2.9	2.25

l'ensemble d'apprentissage (avec une marge de sureté). Puis, un k-PPV est appliqué avec la distance apprise.
• MinCq, considèrant une distribution quasi-uniforme.
Nous évaluons tout d'abord ces approches sur 20 jeux de données (variés) de références. Puis, nous nous attaquons à une tâche de reconnaissance d'objets.

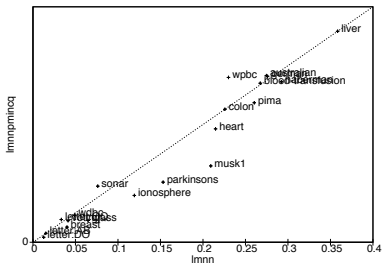
Jeux de données de références (UCI)

Protocole. Les 20 jeux de données (tab. 1 pour leurs propriétés) sont issus du *UCI Machine Learning Repository* (`archive.ics.uci.edu/ml/`). La distance euclidienne est utilisée pour calculer les voisinages. Les données sont découpées en 50% d'apprentissage et 50% de test, sauf *letterAB, letterDO* et *letterOQ* que nous divisons en 20%/80%. Les paramètres sont sélectionnés par validation croisée sur 10 sous-ensembles de l'échantillon d'apprentissage : la marge μ de MinCq et P-MinCq (parmi 14 valeurs dans $[10^{-4}, 0.5]$), le k des k-PPV et LMNN (parmi $\{1, \ldots, 10\}$). Le paramètre de compromis de LMNN est fixé à 0.5 comme dans [WS09].

Resultats. Les résultats obtenus sur les ensembles de test sont reportés dans la tab 2. P-MinCq s'avère plus performant qu'un PPV classique. En moyenne, P-MinCq atteint un taux d'erreur de 16.89% contre 17.88% pour PPV. Avec un test de Student apparié, cette différence est statistiquement significative avec une p-valeur de 0.06. Ce résultat est conforté par un test de signe renvoyant un résultat *win/loss/tie* égal à 12/5/3 avec une p-valeur de 0.07 (fig. 3(a)). De plus, la fig. 3(b) montre que P-MinCq est meilleur que SNN (qui est plus performant sur quelques jeux de données) : une p-valeur de 0.01 en faveur de P-MinCq avec un test de Student et de 0.24 avec un test de signe. En outre, P-MinCq améliore MinCq : un test de Student mène à une p-valeur de 0.02, un test de signe à une p-valeur $\simeq 0.03$ avec un *win/loss/tie* de 12/4/4 (fig. 3(c)). Cette étude montre, d'une part, l'intérêt de la généralisation de MinCq aux distributions **P**-alignées et, d'autre part, la pertinence de la contrainte $P_k = \frac{1}{k}$ (normalisée) pour les PPV. Même s'il n'est pas un algorithme d'apprentissage de métrique, P-MinCq s'avère compétitif avec LMNN (0.1689 contre 0.1607 d'erreur moyenne avec une p-valeur d'environ 0.10 pour un test de Student). Un test de signe amène à une p-valeur de 0.5, indiquant qu'aucune des méthodes n'est meilleure que l'autre. La fig. 4(a) montre que P-MinCq et LMNN sont plutôt complémentaires. D'une part, LMNN apprend une métrique qui ajuste les voisinages (parfois avec un grand succès, *e.g. heart, parkinsons*, *sonar*), mais est parfois moins performant que PPV car une dimensionnalité élevée implique un fort risque de sur-



(a) **P-MinCq** *vs* **LMNN**



(b) **P-MinCq+LMNN** *vs* **LMNN**

FIGURE 4 – Comparaison de P-MinCq et LMNN et de P-MinCq+LMNN et LMNN.

apprentissage (*e.g. colon*, *musk1*). D'autre part, P-MinCq combine différentes règles de PPV et n'intervient donc pas sur les voisinages : cette combinaison de votants apparaît plus stable (tab. 2 avec le meilleur rang moyen) et plus robuste au sur-apprentissage. Afin de mesurer cette complémentarité, nous réalisons une série d'expériences complémentaire dont le but est de combiner LMNN et P-MinCq (lorsque cela paraît pertinent). Concrètement, si LMNN est plus pertinent que P-MinCq sur l'ensemble de validation, la distance apprise par LMNN sera privilégiée par P-MinCq (sinon la distance euclidienne est conservée). Nous reportons alors les résultats dans la tab. 3. La combinaison LMNN+P-MinCq bat clairement toutes les autres méthodes, y compris LMNN seul (un test de Student avec une p-valeur de 0.05 et un test de signe à 0.17), comme illustré par la fig. 4(b). Notons que sur les jeux de données sur lesquels LMNN était le plus performant (e.g. *heart*, *parkinson*, *voting*), LMNN+P-MinCq est capable d'améliorer encore plus ces résultats.

Reconnaissance d'objets (Graz-01)

Protocole. Nous réalisons une expérience sur Graz-01 [OFPA04], un jeu de données où deux objets-classes sont à identifier (*bike*, *person*), ainsi qu'une classe d'arrière plan. Graz-01 est connu pour sa grande variation intra-classes et ses arrières plans très parasités (fig. 5). Les tâches de classification sont : *bike*/non-*bike* et *person*/non-*person*. Nous suivons le protocole de [OFPA04] : pour chaque objet, 100 images positives et 100 négatives sont choisies aléatoirement (50 issues de l'autre objet et 50 de l'arrière plan). Les images sont décrites par un histogramme de fréquences de 200 mots visuels construits à partir des points d'intérêts SIFT [8]. Les voisinages sont calculés via deux distances d'histogrammes : la distance χ^2 et la distance d'intersection.

Resultats. Nous reportons dans la tab. 4 les résultats moyennés sur 10 tirages aléatoires. P-MinCq est encore une fois le plus stable et le meilleur en moyenne. Il est plus performant que PPV et MinCq (une p-valeur <0.01 avec un test de Student) et que SSN dans une moindre mesure (une p-valeur de 0.13). Notons que SNN améliore significativement PPV sur ces données : la variation intra-classe semble rendre payante l'extension du voisinage. Cependant, alors que l'heuristique de symétrie de SNN n'est pas toujours pertinente (comme pour les jeux de données de référence), P-MinCq propose une alternative robuste fondée théoriquement.

TABLE 3 – Taux d'erreurs sur les 20 jeux de données.

Jeu de données	LMNN	LMNN+P-MinCq
australian	**0.2746**	0.2832
blood	**0.2674**	0.2701
breast	0.0400	**0.0257**
colon	0.2258	0.2258
german	**0.2760**	0.2820
glass	0.0648	**0.0370**
haberman	0.2922	**0.2727**
heart	0.2148	**0.1926**
ionosphere	0.1193	**0.0795**
letter :AB	0.0151	0.0151
letter :DO	0.0126	**0.0084**
letter :OQ	**0.0334**	0.0386
liver	0.3584	0.3584
musk1	0.2092	**0.1297**
parkinsons	0.1531	**0.1020**
pima	0.2604	**0.2370**
sonar	**0.0762**	0.0952
voting	0.0413	**0.0367**
wdbc	0.0491	**0.0456**
wpbc	**0.2300**	0.2800
Erreur Moyenne	0.1607	**0.1508**

FIGURE 5 – Exemples de *bikes* (gauche), *persons* (centre), d'arrière plans (droite) issus de Graz-01. Uniquement des parties d'objets peuvent être visibles. Identifier un arrière plan est dur (*e.g.* vélo *vs* moto).

8. *Scale-invariant feature transform* introduit dans [Low99]

TABLE 4 – Taux d'erreurs moyens sur Graz-01.

Distance	Tâche	PPV	SNN	MinCq	P-MinCq
χ^2	bike	0.2310	0.2090	0.2160	0.2095
χ^2	person	0.2385	0.2305	0.2730	0.2250
Intersection	bike	0.2260	0.2185	0.2130	0.2055
Intersection	person	0.2350	0.2370	0.3180	0.2255
Erreur moyenne		0.2326	0.2238	0.2550	0.2164

8 Conclusion

Nous avons proposé un nouvel algorithme, P-MinCq, pour apprendre un vote de majorité pondéré sur un ensemble de classifieurs k-PPV. Il se base sur une généralisation de l'algorithme MinCq [LMR11] (qui trouve sa source dans la théorie PAC-Bayésienne) en permettant l'introduction d'une connaissance *a priori* lors du processus d'apprentissage. Tandis que la distribution sur les votants utilisée par MinCq est non-informative (quasi-uniforme), la connaissance prise en compte par P-MinCq prend la forme d'une distribution $\mathbf{P}$-alignée. D'une part, notre approche n'implique aucune perte d'expressivité et, d'autre part, nous en avons démontré les garanties en généralisation lorsque les votants dépendent des données (comme c'est le cas des classifieurs k-PPV). De plus, nous avons défini un $\mathbf{P}$-alignement spécifique et adapté aux k-PPV et nous avons illustré son intérêt lors des expériences menées. Ceci ouvre des perspectives pour définir des $\mathbf{P}$-alignements pour différents types de classifieurs, mais aussi proposer des méthodes visant à estimer $\mathbf{P}$. Une autre piste prometteuse est de voir P-MinCq comme un algorithme de fusion de classifieurs appris à partir de différentes vues/modalités : $\mathbf{P}$ peut être un *a priori* sur la pertinence des descriptions des données. Dans le cadre particulier des k-PPV, il serait intéressant de combiner P-MinCq avec une méthode d'apprentissage de distance χ^2 proposée très récemment [KTW$^+$12].

Remerciements. Travail financé par les projets VideoSense ANR-09-CORD-026 et LAMPADA ANR-09-EMER-007-02.

Références

[CM11] A. Cornuéjols and L. Miclet. *Apprentissage artificiel*. Eyrolles, 2011.

[DGL96] L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer, 1996.

[DHS01] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern classification*. Wiley, 2001.

[FW95] S. Floyd and M. K. Warmuth. Sample compression, learnability, and the vapnik-chervonenkis dimension. *Mach. Lear.*, 21(3), 1995.

[GHST05] T. Graepel, R. Herbrich, and J. Shawe-Taylor. PAC-Bayesian compression bounds on the prediction error of learning algorithms for classification. *Mach. Lear.*, 59(1-2), 2005.

[GLL$^+$11] P. Germain, A. Lacoste, F. Laviolette, M. Marchand, and S. Shanian. A PAC-Bayes Sample Compression Approach to Kernel Methods. In *ICML*, 2011.

[HT96] T. Hastie and R. Tibshirani. Discriminant adaptive nearest neighbor classification. *IEEE TPAMI*, 18(6), 1996.

[KTW$^+$12] D. Kedem, S. Tyree, K. Weinberger, F. Sha, and G. Lanckriet. Non-linear metric learning. In *NIPS*, 2012.

[LLM$^+$07] A. Lacasse, F. Laviolette, M. Marchand, P. Germain, and N. Usunier. PAC-bayes bounds for the risk of the majority vote and the variance of the Gibbs classifier. In *NIPS*, 2007.

[LM07] F. Laviolette and M. Marchand. Pac-bayes risk bounds for stochastic averages and majority votes of sample-compressed classifiers. *JMLR*, 8, 2007.

[LMR11] F. Laviolette, M. Marchand, and J.-F. Roy. From PAC-Bayes bounds to quadratic programs for majority votes. In *ICML*, 2011.

[Low99] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999.

[McA99] D.A. McAllester. PAC-bayesian model averaging. In *COLT*, 1999.

[MS05] M. Marchand and M. Sokolova. Learning with decision lists of data-dependent features. *JMLR*, 6, 2005.

[NSB03] R. Nock, M. Sebban, and D. Bernard. A simple locally adaptive nearest neighbor rule with application to pollution forecasting. *IJPRAI*, 17(8), 2003.

[OFPA04] A. Opelt, M. Fussenegger, A. Pinz, and P. Auer. Weak hypotheses and boosting for generic object detection and recognition. In *ECCV*, 2004.

[WS09] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *JMLR*, 10, 2009.

[YJ06] L. Yang and R. Jin. Distance Metric Learning : A Comprehensive Survey. Technical report, 2006.