ALEXANDRE LACASSE

Bornes PAC-Bayes et algorithmes d'apprentissage

Thèse présentée
à la Faculté des études supérieures de l'Université Laval
dans le cadre du programme de doctorat en informatique
pour l'obtention du grade de Philosophiae doctor (Ph.D.)

FACULTÉ DES SCIENCES ET DE GÉNIE
UNIVERSITÉ LAVAL
QUÉBEC

2010

Résumé

L'objet principale de cette thèse est l'étude théorique et la conception d'algorithmes d'apprentissage concevant des classificateurs par vote de majorité. En particulier, nous présentons un théorème PAC-Bayes s'appliquant pour borner, entre autres, la variance de la perte de Gibbs (en plus de son espérance). Nous déduisons de ce théorème une borne du risque du vote de majorité plus serrée que la fameuse borne basée sur le risque de Gibbs. Nous présentons également un théorème permettant de borner le risque associé à des fonctions de perte générale. À partir de ce théorème, nous concevons des algorithmes d'apprentissage construisant des classificateurs par vote de majorité pondérés par une distribution minimisant une borne sur les risques associés aux fonctions de perte linéaire, quadratique, exponentielle, ainsi qu'à la fonction de perte du classificateur de Gibbs à piges multiples. Certains de ces algorithmes se comparent favorablement avec AdaBoost.

Abstract

The main purpose of this thesis is the theoretical study and the design of learning algorithms returning majority-vote classifiers. In particular, we present a PAC-Bayes theorem allowing us to bound the variance of the Gibbs' loss (not only its expectation). We deduce from this theorem a bound on the risk of a majority vote tighter than the famous bound based on the Gibbs' risk. We also present a theorem that allows to bound the risk associated with general loss functions. From this theorem, we design learning algorithms building weighted majority vote classifiers minimizing a bound on the risk associated with the following loss functions : linear, quadratic and exponential. Also, we present algorithms based on the randomized majority vote. Some of these algorithms compare favorably with AdaBoost.

Avant-propos

C'est un peu en explorateur que j'ai suivi, à l'automne 2004, le cours d'apprentissage automatique donné par le Professeur Mario Marchand. Je me suis retrouvé, un an plus tard, à débuter un doctorat dans ce domaine avec ce même professeur. Je remercie Mario, pour le projet qu'il m'a donné, pour la confiance qu'il m'a accordée à maintes reprises, pour ce que j'ai pu apprendre durant ces années. Il est très agréable de travailler avec Mario, et même après plusieurs années, on reste impressionné par son professionnalisme.

Je remercie François Laviolette, mon codirecteur de recherche, pour les discussions enrichissantes, pour son enthousiasme contagieux.

Je remercie toutes les personnes qui m'ont permis d'améliorer cette thèse, ce qui inclut en particulier mes directeurs de recherche et les autres membres du jury de la soutenance : Pascal Lang, Liva Ralaivola et Claude-Guy Quimper.

Je remercie mes collègues, que je m'épargne de tous nommer ici. Celui qui a joué le rôle le plus important est sans doute Pascal, qui a fait son entrée dans notre groupe de recherche en tant que stagiaire. Ses habiletés de programmeur ont été un facteur important de mes premiers succès de chercheur. Durant la dernière année, Francis est embarqué dans mon code, un chamboulement s'en est suivi, puis un article et une conférence. Une suite d'événements ponctuée de bons souvenirs.

Finalement, je remercie mes proches et les membres de ma famille, et en particulier Gabrielle, Émile et Mireille, mes principales sources de bonheur.

Durant ma soutenance, j'ai fait un clin d'oeil à Benoît Mandelbrot, certains comprendront avec cette citation : «Ne confondez pas *randoniser* avec le franglais *randomiser* !»

Ce travail a été supporté financièrement par le CRSNG (Conseil de recherches en sciences naturelles et en génie du Canada).

Table des matières

Liste des tableaux

Table des figures

Liste des Algorithmes

Chapitre 1

Introduction

Des applications résultant des recherches en intelligence artificielle sont aujourd'hui omniprésentes dans notre vie ; sans que l'on en ait toujours pleinement conscience, elles se retrouvent dans des logiciels que l'on utilise dans nos ordinateurs personnels, dans les téléphones dits «intelligents», dans les robots industriels, dans les voitures...

L'intelligence artificielle se subdivise en plusieurs domaines dont l'apprentissage automatique, qui consiste en l'étude des différentes façons d'automatiser le processus d'apprentissage d'une tâche, dans l'objectif de concevoir des machines capables de déduire des règles à partir de l'observation d'un environnement. À son tour, l'apprentissage automatique se divise en différents champs de recherche, dont la classification, qui est le sous-domaine de recherche dans lequel se situe cette thèse.

La classification consiste en le problème suivant : considérons des objets $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m$ appartenant tous à une classe parmi un ensemble prédéfini $\mathcal{Y} = \{y_1, y_2, \ldots, y_\ell\}$. On désire, à la vue d'un nouvel objet $\mathbf{x}$, déterminer à quelle classe il appartient. Un exemple concret est la reconnaissance de caractères, c'est-à-dire associer à une image donnée une lettre de l'alphabet ou encore un chiffre.

Pour rendre les objets accessibles pour un algorithme, ceux-ci sont d'abord traités par un observateur et transformés en vecteurs de caractéristiques. L'observateur peut prendre plusieurs formes : il peut par exemple s'agir d'une simple caméra qui prend une capture visuelle d'un objet, ou d'un être humain qui entre manuellement des données. Le vecteur de caractéristiques sera généralement vu comme un vecteur de $\mathbf{R}^n$, ou d'un quelconque espace vectoriel. Par exemple, pour le problème de la reconnaissance de caractères, les objets, qui consistent en des lettres manuscrites, seront transformés par l'observateur en une image de, disons, $m \times n$ pixels, qui peut être interprétée comme

un vecteur de $\mathbf{R}^{n \times m}$.

En notant $\mathcal{X}$ la complétion de l'ensemble des vecteurs de caractéristiques reliés à un problème de classification donné, on peut alors définir un classificateur comme étant simplement une fonction $h(\mathbf{x})$ de la forme

$$h : \mathcal{X} \to \mathcal{Y} \,.$$

L'objectif principal en apprentissage automatique est de concevoir des processus, appelés algorithmes d'apprentissage, capables de construire des «bons» classificateurs pour des problèmes donnés. Dans un contexte d'apprentissage supervisé, l'algorithme a accès à une banque d'exemples classifiés, $(\mathbf{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, dont la création nécessite l'intervention d'un expert, généralement un être humain, qui observe des objets et leur attribue une classe. Par exemple, pour l'identification de caractères, cette étape est banale, puisque la personne qui écrit une lettre sait laquelle elle a écrite (en supposant qu'elle effectue elle-même l'observation). Dans d'autres cas, pour l'étude du rôle des gênes dans le développement d'un cancer par exemple, cette étape nécessitera de couteuses expériences en laboratoire.

En notant $U^* = \bigcup_{i=0}^{\infty} (\mathcal{X} \times \mathcal{Y})^i$, c'est-à-dire que U^* correspond à l'espace duquel proviennent les ensembles de données classifiées, dont l'ensemble d'entrainement S, nous définissons un algorithme d'apprentissage comme étant une fonction $A(S)$ de la forme

$$A : U^* \to (\mathcal{X} \to \mathcal{Y}) \,,$$

c'est-à-dire une fonction prenant en entrée un certain nombre d'exemples classifiés (qui forment l'ensemble d'entrainement) et retournant en sortie un classificateur.

1.1 Recherche de bornes théoriques sur le risque des classificateurs

Une fois que nous disposons d'un algorithme d'apprentissage, une question qui se pose est à savoir si celui-ci construit des «bons» classificateurs. Nous nous intéressons alors au risque d'un classificateur retourné par un algorithme d'apprentissage, c'est-à-dire à la probabilité qu'il se trompe en classifiant des données futures. Nous désirons obtenir des résultats théoriques applicables pour borner le risque d'un classificateur retourné par un algorithme d'apprentissage. Nous sommes également intéressés

à développer des bornes pouvant être utilisées par un algorithme d'apprentissage pour l'orienter durant son exécution.

Une façon de procéder pour borner le risque d'un classificateur est de recourir à un ensemble test. On utilise alors un sous-ensemble des exemples disponibles, appelé ensemble d'entrainement, pour construire le classificateur. Puis on utilise le sous-ensemble des données restantes, appelé ensemble test, pour borner le risque du classificateur. Il est ainsi possible d'obtenir une borne très serrée du risque. Cependant, pour que la borne soit valide, l'ensemble test doit être indépendant du classificateur, ce qui empêche de recourir plusieurs fois à la borne sur l'ensemble test pour concevoir un classificateur.

Une autre approche consiste à borner le risque d'un classificateur directement à partir des informations provenant de l'ensemble d'apprentissage, dont en particulier son risque empirique, c'est-à-dire le taux d'erreur de classification que fait le classificateur sur l'ensemble d'apprentissage. Le théorème PAC-Bayes, dont nous discutons abondamment dans ce document, fournit une borne de ce type. Il permet en fait, à partir d'une classe de classificateurs $\mathcal{H}$ et d'un ensemble d'apprentissage S, d'obtenir une borne théorique sur le risque qui est valide simultanément pour tous les classificateurs de $\mathcal{H}$. Les bornes ainsi obtenues sont naturellement moins précises qu'une borne obtenue à l'aide d'un ensemble test, cependant elles permettent, entre autres, d'effectuer de la sélection de modèle, et elles ne nous contraignent pas à réserver une partie de nos exemples pour calculer une borne sur l'ensemble test.

Plusieurs des algorithmes utilisés dans la pratique (réseaux de neurones, AdaBoost, SVM, ...) construisent des classificateurs qui peuvent être interprétés comme des votes de majorité pondérés de classificateurs rudimentaires. Intuitivement, un vote de majorité peut grandement améliorer un algorithme d'apprentissage. Considérons par exemple le cas simpliste où nous possédons n classificateurs indépendants, tous de risque inférieur ou égal à $\frac{1}{2} - \epsilon$ avec $\epsilon > 0$. Dans cette situation, le risque du vote de majorité démocratique (où tous les votants ont le même poids) tend exponentiellement vite vers zéro si le nombre de classificateurs augmente. Dans la pratique, nous pouvons difficilement concevoir un ensemble de n classificateurs indépendants, et il n'est pas toujours garanti que la combinaison de plusieurs classificateurs donne lieu à un meilleur classificateur ; il peut même arriver que le classificateur par vote de majorité ait un risque supérieur au risque moyen des classificateurs formant le vote.

La borne fournie par le théorème PAC-Bayes classique n'est pas bien adaptée pour borner le risque d'un vote de majorité. En fait, ce théorème permet de borner le risque moyen des classificateurs formant le vote. Un raisonnement assez simple permet alors d'obtenir une borne sur le risque du vote de majorité, cette borne est en fait égale à

deux fois celle sur le risque moyen des classificateurs. Cependant, cette borne n'est pas intéressante puisqu'elle n'aide en rien à caractériser les bons votes de majorité, soit ceux ayant un risque grandement inférieur à celui de la moyenne des classificateurs formant le vote.

1.2 Conception d'algorithmes d'apprentissage inspirés par des bornes théoriques

Les algorithmes d'apprentissage utilisés dans la pratique sont principalement basés sur des heuristiques, des idées intuitives de ce qui doit être optimisé pour concevoir un bon algorithme. Une fois celui-ci conçu, on cherche alors à démontrer des propriétés intéressantes que possède l'algorithme : bornes théoriques sur les classificateurs construits, propriétés de convergence, ... En somme, l'algorithme est proposé et la théorie arrive ensuite. L'objet principal de cette thèse est d'obtenir de bons algorithmes d'apprentissage en faisant le chemin inverse, c'est-à-dire, rechercher d'abord un théorème applicable pour borner efficacement le risque des classificateurs, puis utiliser ce théorème pour déterminer quelles propriétés doit avoir un classificateur pour être prometteur.

Les bornes de type PAC-Bayes nous permettent, à l'aide d'un ensemble d'entrainement, de borner simultanément le risque de tout classificateur consistant en un vote de majorité de classificateurs d'un ensemble prédéfini $\mathcal{H}$ (possiblement infini). En supposant que le théorème PAC-Bayes, ou un de ses dérivés, fournisse une borne représentative du vrai risque des classificateurs, une question naturelle se pose alors : pouvons-nous utiliser la borne PAC-Bayes pour concevoir un algorithme d'apprentissage ? En d'autres termes, l'on se demande si, pour un problème donné, le classificateur possédant la plus faible borne PAC-Bayes est un «bon» classificateur. Si oui, alors des algorithmes d'apprentissage construisant des classificateurs minimisant des bornes de type PAC-Bayes devraient être prometteurs.

1.3 Contributions de la thèse

Dans cette thèse,

– Nous définissons la quantité C_Q, qui représente mieux le risque d'un classificateur par vote de majorité que le risque de Gibbs habituellement utilisé dans le théorème

PAC-Bayes ;
- Nous démontrons un théorème fournissant des bornes sur diversesf réalisations d'un classificateur par vote de majorité sur un ensemble d'entrainement (dont en particulier son risque de Gibbs et aussi la variance de celui-ci) ;
- Nous démontrons des bornes théoriques sur la quantité C_Q qui permettent d'obtenir une borne du risque d'un classificateur par vote de majorité souvent plus serrée que la borne classique déduite du risque de Gibbs ;
- Nous démontrons un théorème fournissant des bornes pour des risques associés à des fonctions de perte générales ;
- Nous présentons des algorithmes d'apprentissage, tous conçus dans l'objectif de minimiser la borne du théorème sur les fonctions de pertes générales pour une fonction de perte donnée ;
- Nous définissons le concept de distribution quasi-uniforme et nous adaptons nos algorithmes d'apprentissage dans ce cadre de travail.

1.4 Plan de cette thèse

Cette thèse est divisée en deux parties. Dans la première partie de la thèse, qui est constituée des chapitres 2 à 6 nous détaillons le cadre théorique dans lequel nous nous situons, puis nous présentons des nouvelles bornes de type PAC-Bayes. Dans la seconde partie, qui est constituée des chapitres 7 à 9, nous présentons quelques algorithmes d'apprentissage qui sont directement inspirés de nos bornes.

Dans les chapitres 2 et 3, nous présentons quelques définitions et résultats préliminaires servant de base pour la suite de la thèse. Dans le chapitre 2, nous définissons formellement notre cadre théorique d'apprentissage, ainsi que quelques notions importantes, dont ce qu'on entend exactement par «risque d'un classificateur» et par «classificateur par vote de majorité». Nous présentons également dans ce chapitre le théorème PAC-Bayes dans sa version classique. Dans le chapitre 3, nous poursuivons avec quelques définitions de quantités statistiques associées à un ensemble de classificateurs et un ensemble d'exemples, dont le taux d'erreur moyen et sa variance, ainsi que la quantité C_Q (apparue dans Lacasse *et al.* (2007)). Nous discutons également d'une variante de l'inégalité de Tchebychev et nous l'appliquons pour obtenir une borne (non calculable) du risque d'un classificateur par vote de majorité possiblement plus serrée que celle découlant de l'inégalité de Markov.

Le chapitre 4 présente un prolongement des travaux publiés dans Lacasse *et al.* (2007). Nous y présentons de nouvelles bornes du type PAC-Bayes permettant de borner

des quantités autres que le risque de Gibbs, dont la variance du taux moyen d'erreur et la quantité C_Q. Nous présentons en particulier une borne directe de C_Q (non présente dans Lacasse *et al.* (2007)), ainsi que quelques résultats mathématiques concernant C_Q permettant de calculer efficacement les bornes théoriques présentées.

Le chapitre 5 reprend pour sa part les travaux publiés dans Germain *et al.* (2007), concernant un nouveau théorème PAC-Bayes destiné à borner le risque associé à des fonctions perte générales (soit des fonctions plus complexes que la fonction de perte linéaire aboutissant au risque de Gibbs). L'objectif premier de ces travaux était d'obtenir une borne serrée du risque d'un classificateur par vote de majorité en approchant la fonction de perte $0-1$ par des séries de Taylor. Nous n'avons pu atteindre cet objectif, la borne devenant trop lâche pour des fonctions s'approchant de la perte $0-1$, cependant des algorithmes d'apprentissage intéressants (présentés dans la deuxième partie de la thèse) découlerons de ces bornes.

Suite à un parcours rapide de différentes approches que nous avons envisagées pour concevoir des algorithmes d'apprentissage inspirés de bornes de type PAC-Bayes (chapitre 7), nous consacrons le reste de cette thèse au développement de certains algorithmes et à la comparaison de ceux-ci avec des algorithmes existants (en l'occurrence AdaBoost et la régression ridge).

Dans le chapitre 8, nous présentons un algorithme générique permettant de construire un classificateur par vote de majorité en minimisant la borne d'une version du théorème sur les fonctions de perte générales présenté au chapitre 5. Nous implémentons ensuite cet algorithme avec deux versions du théorème et quatre fonctions de perte différentes : la perte linéaire (seulement présentée comme indicateur), la perte quadratique, la perte exponentielle et la perte du classificateur de Gibbs à piges multiples. La version de l'algorithme basée sur la perte du classificateur de Gibbs à piges multiples fait également l'objet d'une publication (voir Lacasse *et al.*, 2010). Nous montrons que nous obtenons, avec la version dite hyperparamétrée du théorème, des algorithmes à la fois compétitifs avec AdaBoost et avec la régression ridge, lorsque des souches de décision sont utilisées comme classificateurs de base.

Finalement dans le chapitre 9, qui reprend également des travaux publiés (voir Germain *et al.*, 2009b), nous présentons le concept de distribution quasi-uniforme ainsi qu'une version du théorème PAC-Bayes adaptée pour ce type de distribution. Nous adaptons les algorithmes du chapitre 8 en les contraignant à construire des classificateurs par vote de majorité pondéré par une distribution quasi-uniforme. Nous verrons qu'il est ainsi possible de réduire la complexité des problèmes d'optimisation que les algorithmes attaquent sans nuire à la puissance des classificateurs produits. Cependant, les bornes

obtenues dans le cadre des distributions quasi-uniformes s'avèrent moins serrées que celles du cadre avec distributions non contraintes.

Première partie

Bornes de type PAC-Bayes

Chapitre 2

Classification, vote de majorité et théorèmes PAC-Bayes.

Dans ce chapitre, nous présentons le contexte d'apprentissage que nous adoptons dans cette thèse. Nous présentons également quelques résultats précurseurs de nos travaux, dont le théorème PAC-Bayes.

2.1 Définitions préliminaires

Notons $\mathcal{X}$ l'ensemble des observations possibles et $\mathcal{Y}$ l'ensemble des classes. Nous appelons *ensemble d'exemples* ou encore *ensemble de données étiquetées* un sous-ensemble donné de $\mathcal{X} \times \mathcal{Y}$, et *classificateur* une fonction de $\mathcal{X} \to \mathcal{Y}$. Dans cet ouvrage, nous considérons uniquement des problèmes de classification binaire, nous avons donc $|\mathcal{Y}| = 2$ et nous identifions les classes par l'ensemble $\{-1, +1\}$. Nous supposons que les données (ou exemples) sont générées indépendamment suivant une distribution D sur $\mathcal{X} \times \mathcal{Y}$ que nous ne connaissons pas (mais que nous supposons exister). On s'intéresse au risque du classificateur h généré par un algorithme d'apprentissage, c'est-à-dire à la probabilité que celui-ci assigne à une entrée $\mathbf{x}$ (appelée observation ou vecteur de caractéristiques) une mauvaise classe, soit une classe autre que celle à laquelle appartient l'objet ayant produit l'observation $\mathbf{x}$. On entend ici par algorithme d'apprentissage, une fonction prenant en entrée un ensemble de données étiquetées, que l'on appelle *ensemble d'entrainement*, et retournant un classificateur.

Définition 2.1.1 (Risque)**.** *Soit h un classificateur, le risque de h (aussi appelé vrai*

risque) est défini par la fonction $R(h)$ suivante :

$$R(h) \stackrel{\text{déf}}{=} \Pr_{(\mathbf{x},y)\sim D} \left(h(\mathbf{x}) \neq y\right).$$

Puisque la distribution ayant généré les données est inconnue, il n'est pas possible d'évaluer le risque d'un classificateur avec exactitude. À noter également que si cette distribution était connue, alors le problème de trouver le classificateur optimal se réduirait à assigner à chaque vecteur de caractéristiques $\mathbf{x}$ la classe y la plus probable selon la distribution D.

Définition 2.1.2 (Risque empirique)**.** *Soit $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$ un échantillon et h un classificateur, le risque empirique de h sur S, noté $R_S(h)$ et correspondant au taux d'erreur de h sur S, est défini par*

$$R_S(h) \stackrel{\text{déf}}{=} \Pr_{(\mathbf{x},y)\sim S} \left(h(\mathbf{x}) \neq y\right) = \frac{1}{m}\sum_{i=1}^{m} I(h(\mathbf{x}_i) \neq y_i),$$

où $I(a)$ est la fonction indicatrice qui vaut 1 si la proposition a est vraie et 0 sinon.

Il est possible d'avoir une idée du risque d'un classificateur en calculant son risque empirique. Le risque empirique d'un classificateur procure habituellement une très bonne approximation de son vrai risque lorsque l'ensemble servant à l'évaluer est indépendant de celui ayant servi à l'apprentissage du classificateur. Il est également possible de calculer le risque empirique en utilisant l'ensemble ayant été utilisé lors de l'apprentissage. Dans ce cas par contre, la garantie que le résultat obtenu soit représentatif du vrai risque est beaucoup plus faible. Par exemple, considérons $S \in (\mathcal{X} \times \mathcal{Y})^m$, un ensemble d'entrainement. Si les classes sont bien séparées, c'est-à-dire si $(\mathbf{x}, y_1) \in \mathcal{X} \times \mathcal{Y}$ et $(\mathbf{x}, y_2) \in \mathcal{X} \times \mathcal{Y} \Rightarrow y_1 = y_2$, alors la fonction h définie par

$$h(\mathbf{x}) = \begin{cases} -1 & \text{si } (\mathbf{x}, -1) \in S \\ +1 & \text{sinon} \end{cases}$$

sera toujours un classificateur de risque empirique nul sur l'ensemble d'entrainement, mais il aura généralement un vrai risque très grand (dans certaines situations, le vrai risque de ce classificateur peut même valoir 1).

Il existe cependant des techniques pour obtenir des bornes serrées du risque d'un classificateur à partir de son risque empirique sur l'ensemble d'apprentissage. Par exemple, le théorème PAC-Bayes, que nous présentons à la section 2.3, permet de borner simultanément le risque d'une classe complète de classificateurs à partir de leur risque empirique sur l'ensemble d'entrainement. Un tel résultat permet théoriquement de faire de la sélection de paramètres lors de la phase d'apprentissage, ou de choisir un classificateur prometteur en fonction des résultats obtenus sur la borne.

2.2 Vote de majorité

Étant donné un ensemble de classificateurs $\mathcal{H}$, il est possible d'augmenter la puissance de classification des éléments de $\mathcal{H}$ en combinant ceux-ci de sorte à obtenir un classificateur plus complexe. Par exemple, si $\mathcal{H}$ est discret, disons $\mathcal{H} = \{h_1, h_2, \cdots, h_n\}$, nous pouvons construire un classificateur de la forme

$$c(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{n} \alpha_i h_i(\mathbf{x})\right),$$

où les α_i sont des nombres réels. Lorsque nous souhaitons que les α_i forment une distribution de probabilité, nous imposons de plus $\alpha_i \geq 0 \, \forall i$ et $\sum_{i=1}^{n} \alpha_i = 1$. Le classificateur $c(\mathbf{x})$ constitue un vote de majorité pondéré de classificateurs de $\mathcal{H}$. En somme, si la majorité des classificateurs de $\mathcal{H}$ attribue à $\mathbf{x}$ la classe y, alors c attribuera à $\mathbf{x}$ cette même classe.

De façon plus générale, considérons Q une distribution de probabilité sur un ensemble (pas nécessairement discret ni même dénombrable) de classificateurs $\mathcal{H}$. Nous appelons classificateur par vote de majorité le classificateur résultant d'un vote de majorité pondéré par Q.

Définition 2.2.1. *Pour Q une distribution sur un ensemble de classificateurs $\mathcal{H}$, le classificateur par vote de majorité pondéré par Q, noté $B_Q(\mathbf{x})$, est donné par*

$$B_Q(\mathbf{x}) \stackrel{\text{déf}}{=} \operatorname{sgn}\left(\int_{\mathcal{H}} Q(h) h(\mathbf{x}) dh\right).$$

L'expression «risque du vote de majorité» est utilisée pour désigner la quantité $R(B_Q)$, soit le risque du classificateur par vote de majorité.

On s'attend généralement à ce que la combinaison de classificateurs en un vote de majorité améliore la classification. Cependant, ceci n'est pas toujours le cas. Considérons par exemple un ensemble $\mathcal{H}$ formé de deux classificateurs h_1 et h_2 tels que h_1 possède un risque de 1 et h_2 un risque de zéro. Posons $Q(h_1) = \frac{1}{2} + \epsilon$ et $Q(h_2) = \frac{1}{2} - \epsilon$ pour $0 < \epsilon < \frac{1}{2}$. Le classificateur $B_Q(\mathbf{x})$ possède alors un risque de 1, ce qui est supérieur au risque moyen des classificateurs formant le vote, qui est de $\frac{1}{2} + \epsilon$.

Dans un autre cas extrême, si $\mathcal{H}$ est constitué de n classificateurs indépendants et possédant tous un risque inférieur à $\frac{1}{2} - \epsilon$ pour $0 < \epsilon < \frac{1}{2}$, en attribuant le poids des classificateurs démocratiquement, donc $Q(h) = \frac{1}{n}$ pour tout $h \in \mathcal{H}$, nous aurons comme borne de $R(B_Q)$:

$$R(B_Q) \leq \Pr_{X \sim \text{Bin}(n, \frac{1}{2} - \epsilon)} \left(X \geq \left\lceil \frac{n}{2} \right\rceil\right),$$

où $\text{Bin}(n, p)$ représente une loi binomiale à n épreuves et d'espérance p. Cette quantité tend vers zéro en fonction du nombre de classificateurs. Par exemple avec $\epsilon = 0.1$ et $n = 101$, on obtient une borne valant environ 0.02 (alors que chaque classificateur peut avoir un risque individuel de 0.4).

En somme, nous voyons que bien que la combinaison par vote de majorité de classificateurs puisse légèrement nuire à la classification dans certains cas, elle peut en revanche grandement l'améliorer dans plusieurs autres cas. Tout le problème de la construction d'un vote de majorité repose sur l'attribution du poids des classificateurs. Différents outils ont été développés dans le but de construire des classificateurs par vote de majorité les plus fiables possibles. Par exemple, le *boosting* (voir Freund et Schapire (1995, 1996), voir également Meir et Rätsch (2003)), attribue les poids de sorte à forcer une certaine décorrélation entre les classificateurs du vote. Un autre exemple est le *Bagging* (voir Breiman (1996); Quinlan (1996)), qui tente de créer un classificateur plus stable face aux données d'apprentissage, ou encore les *Random Forests* (voir Breiman (2001)) qui ont mené à une application commerciale.

Bien que plusieurs approches différentes aient été développées pour construire des classificateurs par vote de majorité, et bien que plusieurs de ces approches donnent souvent des résultats remarquables, nous ne possédons pas de bonnes garanties sur l'efficacité des votes de majorité, et nous ne savons pas exactement ce qu'il faut optimiser sur les données pour parvenir à construire un vote de majorité optimal.

2.3 Théorème PAC-Bayes

Le théorème PAC-Bayes, introduit par McAllester (voir McAllester (1999a,b), voir également Langford *et al.* (2001), Seeger (2002), Langford (2005), McAllester (2003), Meir et Zhang (2003), Laviolette et Marchand (2005), Banerjee (2006) et Germain *et al.* (2009a) pour des améliorations et des généralisations du résultat) ne permet pas directement de borner le risque du classificateur par vote de majorité. Il permet plutôt de borner le risque de la version stochastique de ce dernier, appelée classificateur de Gibbs. Le classificateur de Gibbs, noté G_Q, se définit à partir d'un ensemble de classificateurs $\mathcal{H}$ et d'une distribution Q sur ces classificateurs de la façon suivante : pour classifier une observation $\mathbf{x}$, l'on pige un classificateur h selon Q puis l'on assigne à $\mathbf{x}$ la classe $h(\mathbf{x})$. Le risque du classificateur de Gibbs, appelé plus simplement risque de Gibbs, noté $R(G_Q)$, correspond alors au risque moyen des classificateurs de $\mathcal{H}$.

Définition 2.3.1. *Pour Q, une distribution sur un ensemble de classificateurs $\mathcal{H}$, le*

risque de Gibbs (ou risque du classificateur de Gibbs), noté $R(G_Q)$, est donné par

$$R(G_Q) \stackrel{\text{déf}}{=} \mathop{\mathbf{E}}_{h\sim Q} R(h) = \mathop{\mathbf{E}}_{h\sim Q} \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} I(h(\mathbf{x}) \neq y)\,.$$

Théorème 2.3.2 (PAC-Bayes)**.** *Soit $\mathcal{H}$ un ensemble de classificateurs et $\delta \in (0,1]$. Pour toute distribution à priori P sur $\mathcal{H}$, nous avons :*

$$\Pr_{S\sim D^m}\left(\forall Q \text{ sur } \mathcal{H} : \mathrm{kl}\Big(R_S(G_Q)\|R(G_Q)\Big) \leq \frac{1}{m}\left[\mathrm{KL}(Q\|P) + \log\frac{m+1}{\delta}\right]\right) \geq 1-\delta,$$

où $\mathrm{KL}(Q\|P)$ correspond à la divergence de Kullback-Leibler entre deux distributions P et Q :

$$\mathrm{KL}(Q\|P) \stackrel{\text{déf}}{=} \mathop{\mathbf{E}}_{h\sim Q} \log\frac{Q(h)}{P(h)}$$

et où $kl(q\|p)$ correspond à la divergence de Kullback-Leibler entre deux distributions de Bernoulli avec probabilités de succès respectives q et p :

$$\mathrm{kl}(q\|p) \stackrel{\text{déf}}{=} q\log\frac{q}{p} + (1-q)\log\frac{1-q}{1-p}.$$

Démonstration : Voir le corollaire 6.1.3. ■

Dans l'énoncé du théorème PAC-Bayes, la première distribution, P, indique en quelque sorte notre degré de confiance à priori envers les différents classificateurs, elle est donc posée avant d'avoir observé les données. La seconde distribution, Q, dite distribution à postériori, indique notre degré de confiance envers les classificateurs après avoir observé les données. Le théorème PAC-Bayes indique donc que plus notre degré de confiance à priori des différents classificateurs est semblable à notre degré de confiance à postériori, moins le vrai risque du classificateur de Gibbs peut s'éloigner de son risque empirique (avec probabilité $1-\delta$).

Remarquons que le théorème PAC-Bayes fournit une borne de $R(G_Q)$ valide uniformément pour toute distribution Q, et donc pour tout classificateur G_Q que l'on peut construire à partir de $\mathcal{H}$. Nous pouvons alors utiliser la borne pour déterminer une distribution Q prometteuse, c'est-à-dire donnant un classificateur G_Q de plus petit risque possible.

Le théorème PAC-Bayes permet également de borner indirectement le risque du classificateur par vote de majorité. En effet, il n'est pas difficile de montrer (voir l'inégalité 3.4) que le risque du classificateur par vote de majorité est lié au risque du classificateur de Gibbs par la relation suivante :

$$R(B_Q) \leq 2R(G_Q)\,.$$

Une borne sur $R(G_Q)$ fournit alors une borne sur $R(B_Q)$. Cependant, la faiblesse de cette borne réside dans le fait qu'elle ne permet pas d'identifier les cas où le classificateur par vote de majorité est meilleur que les classificateurs formant le vote, c'est-à-dire les cas où $R(B_Q) < R(G_Q)$, qui sont les cas qui nous intéressent particulièrement : ceux où il est préférable d'effectuer un vote de majorité. Il est à remarquer également que le facteur 2 de la borne de $R(B_Q)$ n'est pas exagéré. En effet, il peut être approché lorsque des classificateurs possédant un grand risque sont également fortement corrélés et deviennent alors majoritaires dans le vote. Il peut également être approché dans des cas un peu pathologiques, par exemple, en faisant tendre ϵ vers zéro dans le raisonnement de la section 2.2, qui donne un exemple de cas où le risque de Gibbs peut être de $\frac{1}{2} + \epsilon$, alors que le risque du vote de majorité est de 1.

2.3.1 Représentation graphique de la borne du théorème PAC-Bayes

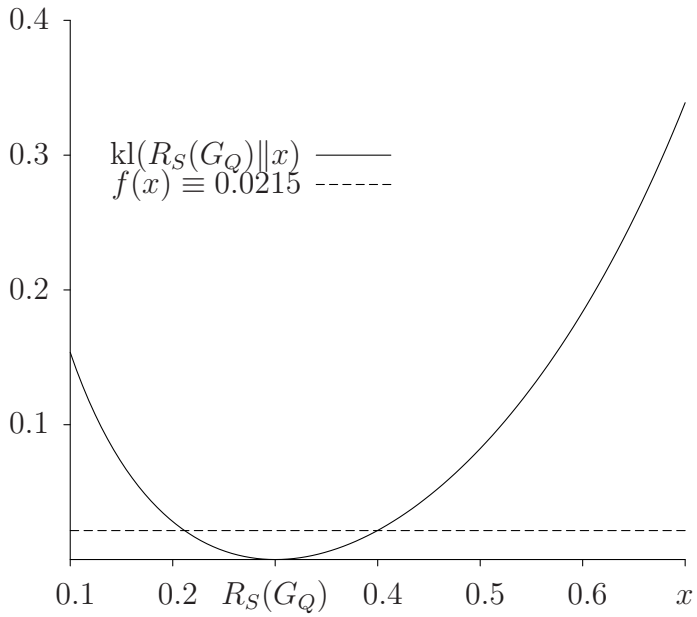

FIGURE 2.1 – Application du théorème PAC-Bayes

La figure 2.1 montre graphiquement une application du théorème PAC-Bayes. Dans cet exemple, nous avons supposé la quantité $\mathrm{KL}(Q\|P)$ égale à 10. Avec un échantillon d'apprentissage de 1000 exemples et un taux de confiance de 99%, on obtient alors dans le théorème $\frac{\mathrm{KL}(Q\|P)+\log((m+1)/\delta)}{m} \approx 0.0215$. En supposant $R_S(G_Q) = 0.3$, le théorème PAC-Bayes affirme qu'avec probabilité au moins 99%, le vrai risque empirique du classificateur de Gibbs est compris dans l'intervalle T ayant pour extrémités les valeurs

en x des deux points d'intersection entre les fonctions $\mathrm{kl}(R_S(G_Q)\|\cdot)$ et $f(\cdot) \equiv 0.0215$. Le théorème PAC-Bayes permet donc en une seule application d'obtenir à la fois une borne supérieure de $R(G_Q)$, donnée par le point maximal de l'intervalle T, et une borne inférieure, donnée par le point minimal de T. Dans cet exemple, on trouve donc respectivement comme bornes inférieure et supérieure les valeurs 0.21 et 0.4.

2.3.2 Théorème PAC-Bayes pour des fonctions de perte générales à valeurs dans {0,1}

Nous venons de voir l'énoncé du théorème PAC-Bayes sous sa forme la plus simple, celle s'appliquant seulement au risque standard, soit le risque de la définition 2.1.1 – qui, lui, découle de la fonction de perte $I(h(\mathbf{x}) \neq y)$, qui attribue une perte de 1 au classificateur h s'il attribue à $\mathbf{x}$ une classe autre que y et une perte de 0 sinon. C'est en fait la forme du théorème PAC-Bayes qui nous est la plus utile dans la pratique.

Nous aurons cependant besoin pour nos démonstrations des chapitres 4 et 5 d'une version un peu plus générale du théorème PAC-Bayes s'appliquant à des risques définis à partir d'une fonction de perte $\ell(h, \mathbf{x}, y)$ à valeur $\{0, 1\}$ pouvant différer de la fonction $I(h(\mathbf{x}) \neq y)$. À noter que bien que cette version du théorème soit plus générale (le théorème PAC-Bayes classique en étant un cas particulier avec $\ell(h, \mathbf{x}, y) = I(h(\mathbf{x}) \neq y)$) sa démonstration demeure identique.

Théorème 2.3.3 (PAC-Bayes)**.** *Soit $\mathcal{H}$ un ensemble de classificateurs et soit ℓ une fonction de la forme $\ell : \mathcal{H} \times \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$. Posons $R^\ell(h) = \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} \ell(h, \mathbf{x}, y)$ et posons $R^\ell_S(h) = \frac{1}{|S|} \sum_{i=1}^{|S|} \ell(h, \mathbf{x}_i, y_i)$. Posons $R^\ell(G_Q) = \mathop{\mathbf{E}}_{h\sim Q} R^\ell(h)$ et $R^\ell_S(G_Q) = \mathop{\mathbf{E}}_{h\sim Q} R^\ell_S(h)$. Soit $\delta \in (0, 1]$. Alors pour toute distribution à priori P sur $\mathcal{H}$ indépendante des données nous avons :*

$$\Pr_{S\sim D^m} \left(\forall Q \text{ sur } \mathcal{H} : \mathrm{kl}\Big(R^\ell_S(G_Q)\|R^\ell(G_Q)\Big) \leq \frac{1}{m} \left[\mathrm{KL}(Q\|P) + \log \frac{m+1}{\delta} \right] \right) \geq 1 - \delta \,.$$

Démonstration : Voir le corollaire 6.1.3 ainsi que la remarque 6.1.6. ■

Chapitre 3

Notions de probabilité

Dans ce chapitre, nous présentons quelques propositions concernant les différentes mesures (espérance, variance, taux de désaccord, ...) que l'on peut effectuer sur un ensemble de données en relation avec un ensemble de classificateurs et une distribution sur ceux-ci. Les quantités qui nous intéressent, et qui dépendent d'une distribution D sur les données et d'une distribution Q sur un ensemble de classificateurs $\mathcal{H}$, sont en fait toutes directement liées à la variable aléatoire correspondant au taux de classificateurs qui se trompent lors de la classification d'un exemple.

3.1 Définitions de base

Nous définissons dans cette section une variable aléatoire rattachée à la distribution Q, que nous notons $W_Q(\mathbf{x}, y)$, ainsi que quelques quantités qui lui sont associées — qui sont en fait liées à son espérance et sa variance. Plusieurs résultats présentés soit dans ce chapitre soit dans les chapitres 4 et 5 proviennent de l'étude de $W_Q(\mathbf{x}, y)$.

Définition 3.1.1. *Pour $(\mathbf{x}, y)$ un exemple, notons $W_Q(\mathbf{x}, y)$ le poids des classificateurs pondérés par la distribution Q classifiant incorrectement $(\mathbf{x}, y)$, c'est-à-dire*

$$W_Q(\mathbf{x}, y) = \mathop{\mathbf{E}}_{h \sim Q} I(h(\mathbf{x}) \neq y)\,.$$

Les mesures qui nous intéressent sont l'espérance et la variance de W_Q, ainsi que le taux moyen de désaccord entre les classificateurs de $\mathcal{H}$ et les taux moyens d'erreurs conjointes et de succès conjoints. Nous notons respectivement ces dernières valeurs d_Q, e_Q et s_Q.

Avant de définir formellement ces quantités, remarquons d'abord le lien direct qui existe entre le risque de Gibbs, $R(G_Q)$, et le taux moyen de classificateurs qui se trompent en classifiant des données (générées par une distribution D). En effet, nous pouvons écrire $R(G_Q)$ en fonction de W_Q comme suit

$$\begin{aligned} R(G_Q) &\stackrel{\text{déf}}{=} \mathop{\mathbf{E}}_{h\sim Q} \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} I(h(\mathbf{x}) \neq y) \\ &= \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} \mathop{\mathbf{E}}_{h\sim Q} I(h(\mathbf{x}) \neq y) \\ &= \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} W_Q(\mathbf{x},y)\,. \end{aligned}$$

Définition 3.1.2. *Les taux de désaccord d_Q, d'erreurs conjointes e_Q et de succès conjoints s_Q sont définis par*

$$\begin{aligned} d_Q &\stackrel{\text{déf}}{=} \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} \mathop{\mathbf{E}}_{h_1\sim Q} \mathop{\mathbf{E}}_{h_2\sim Q} I(h_1(\mathbf{x}) \neq h_2(\mathbf{x})) \\ e_Q &\stackrel{\text{déf}}{=} \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} \mathop{\mathbf{E}}_{h_1\sim Q} \mathop{\mathbf{E}}_{h_2\sim Q} I(h_1(\mathbf{x}) = h_2(\mathbf{x}) \neq y) \\ &= \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} \left(\mathop{\mathbf{E}}_{h\sim Q} I(h(\mathbf{x}) \neq y) \right)^2 \\ &= \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} \left(W_Q(\mathbf{x},y)\right)^2 \\ s_Q &\stackrel{\text{déf}}{=} \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} \mathop{\mathbf{E}}_{h_1\sim Q} \mathop{\mathbf{E}}_{h_2\sim Q} I(h_1(\mathbf{x}) = h_2(\mathbf{x}) = y)\,. \end{aligned}$$

Finalement, par définition de la variance d'une variable aléatoire, nous avons

$$\begin{aligned} \mathop{\mathbf{Var}}_{(\mathbf{x},y)\sim D} W_Q(\mathbf{x},y) &\stackrel{\text{déf}}{=} \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} \left(W_Q(\mathbf{x},y)\right)^2 - \left(\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} W_Q(\mathbf{x},y) \right)^2 \\ &= e_Q - \left(R(G_Q)\right)^2\,. \end{aligned} \tag{3.1}$$

Il suit directement de la définition 3.1.2 que l'on a l'égalité suivante :

$$1 = d_Q + e_Q + s_Q\,, \tag{3.2}$$

puisque, lors d'une classification, soit deux classificateurs donnés sont en désaccord, soit ils se trompent tous deux, soit ils ont tous les deux raison (ce qui épuise toutes les possibilités).

3.1.1 Relation entre $W_Q(\mathbf{x}, y)$ et le risque du vote de majorité

Le classificateur par vote de majorité pondéré par Q, que l'on note B_Q, fait une erreur de classification lorsque la majorité (selon la pondération donnée par Q) des

classificateurs se trompent. Il en découle l'inégalité suivante concernant le risque du vote de majorité

$$R(B_Q) \leq \Pr_{(\mathbf{x},y)\sim D}\left(W_Q(\mathbf{x},y) \geq \frac{1}{2}\right). \tag{3.3}$$

L'inégalité (au lieu d'une égalité) provient du fait que nous n'avons pas précisé ce que le classificateur de Bayes fait lorsque $W_Q(\mathbf{x},y) = \frac{1}{2}$. Elle devient une égalité lorsqu'il est impossible d'avoir $W_Q(\mathbf{x},y) = \frac{1}{2}$, par exemple lorsque l'on possède un nombre fini et impair de classificateurs et qu'ils ont tous la même pondération.

L'inégalité bien connue $R(B_Q) \leq 2R(G_Q)$ n'est alors qu'une simple conséquence de l'inégalité de Markov. En effet, nous avons

$$\begin{aligned} R(B_Q) &\leq \Pr_{(\mathbf{x},y)\sim D}\left(W_Q(\mathbf{x},y) \geq \frac{1}{2}\right) \\ &\leq \frac{\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} W_Q(\mathbf{x},y)}{1/2} \\ &= 2R(G_Q)\,. \end{aligned} \tag{3.4}$$

Cette relation entre le classificateur par vote de majorité et le classificateur de Gibbs est importante car elle permet d'utiliser le théorème PAC-Bayes (qui borne $R(G_Q)$) pour obtenir une borne de $R(B_Q)$. Cependant, cette approche a le défaut de toujours donner une borne de $R(B_Q)$ plus élevée que celle de $R(G_Q)$ (et même deux fois plus élevée), alors que nous savons bien que le classificateur par vote de majorité peut parfois être beaucoup plus performant que celui de Gibbs. Néanmoins, cela permet d'obtenir une borne assez faible de $R(B_Q)$ dans des situations idéales, c'est-à-dire lorsque tous les classificateurs formant le vote de majorité ont un faible risque, et donc que $R(G_Q)$ est faible. Cette approche donne entre autres de bons résultats pour borner le risque des SVM, mais elle échoue à donner des bornes intéressantes pour plusieurs algorithmes d'apprentissage qui procèdent par vote de majorité pondéré. Par exemple, on observe souvent pour le *boosting* des risques de Gibbs avoisinant 40% (même lorsque le vote de majorité ne produit aucune erreur de classification) et dans ces cas, l'approche PAC-Bayes classique fournit une borne de $R(B_Q)$ supérieure à 80%. Il est donc impossible d'utiliser cette approche pour caractériser les bons votes de majorité pour de tels algorithmes d'apprentissage.

3.2 Borne sur $R(B_Q)$ pouvant être plus petite que celle sur $R(G_Q)$

Langford et Shawe-Taylor (2003) ont proposé une façon d'améliorer la borne $R(B_Q) \leq 2R(G_Q)$ en une borne ayant la forme $R(B_Q) < (1+\varepsilon)R(G_Q)$ dans le cas de classificateurs ayant une grande marge de séparation. Cette inégalité possède cependant la même faiblesse que la précédente. En effet elle ne peut conduire à une borne de $R(B_Q)$ qui soit plus petite que celle de $R(G_Q)$. Il est cependant possible de faire mieux que cela.

Le corollaire 3.2.2, qui découle du théorème 3.2.1, conduit à une borne de $R(B_Q)$ pouvant être plus petite (et même beaucoup plus petite) que $R(G_Q)$; cette borne est expliquée en détails au chapitre 4. Notre méthode pour améliorer la borne du risque du classificateur par vote de majorité se base sur l'observation faite de la borne $R(B_Q) \leq 2R(G_Q)$ vue sous la forme d'une simple application de l'inégalité de Markov. L'inégalité de Tchebychev, présentée au théorème 3.2.1, généralise l'inégalité de Markov lorsqu'en plus de connaitre l'espérance d'une variable aléatoire, nous connaissons sa variance. Cette version de l'inégalité de Tchebychev (sans valeur absolue) ne se retrouve généralement pas dans les livres de probabilités, nous en fournissons donc une démonstration.

Théorème 3.2.1 (Inégalité de Tchebychev)**.** *Soit X une variable aléatoire d'espérance μ et de variance σ^2. Soit $a \geq 0$. Alors*

$$\Pr\left(X \geq a + \mu\right) \leq \frac{\sigma^2}{\sigma^2 + a^2}.$$

Démonstration :

$$\begin{aligned}
\Pr\left(X \geq a+\mu\right) &= \Pr\left(X-\mu+\frac{\sigma^2}{a} \geq a+\frac{\sigma^2}{a}\right) \\
&\leq \Pr\left(\left(X-\mu+\frac{\sigma^2}{a}\right)^2 \geq \left(a+\frac{\sigma^2}{a}\right)^2\right) \\
&\leq \frac{\mathbf{E}\left(X-\mu+\frac{\sigma^2}{a}\right)^2}{\left(a+\frac{\sigma^2}{a}\right)^2} \\
&= \frac{\mathbf{E}\,X^2+\mu^2+\frac{\sigma^4}{a^2}-2\mu\,\mathbf{E}\,X+2\frac{\sigma^2}{a}\,\mathbf{E}\,X-2\mu\frac{\sigma^2}{a}}{a^2+2\sigma^2+\frac{\sigma^4}{a^2}} \\
&= \frac{\mathbf{E}\,X^2-\mu^2+\frac{\sigma^4}{a^2}}{(1+\frac{\sigma^2}{a^2})(\sigma^2+a^2)} \\
&= \frac{\sigma^2+\frac{\sigma^4}{a^2}}{(1+\frac{\sigma^2}{a^2})(\sigma^2+a^2)} \\
&= \frac{\sigma^2(1+\frac{\sigma^2}{a^2})}{(1+\frac{\sigma^2}{a^2})(\sigma^2+a^2)} \\
&= \frac{\sigma^2}{\sigma^2+a^2}.
\end{aligned}$$

■

En appliquant l'inégalité de Tchebychev avec la variable aléatoire W_Q nous obtenons le corollaire suivant, qui fournit une nouvelle borne de $R(B_Q)$ en fonction de $R(G_Q)$.

Corollaire 3.2.2. *Soit $\mathcal{H}$ un ensemble de classificateurs et Q une distribution sur $\mathcal{H}$. Si $R(G_Q) < \frac{1}{2}$, alors*

$$R(B_Q) \leq \frac{\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y)}{\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y) + (1/2 - R(G_Q))^2}.$$

Démonstration :

$$\begin{aligned}
R(B_Q) &\leq \Pr_{(\mathbf{x},y)\sim D}\left(W_Q(\mathbf{x},y) \geq \frac{1}{2}\right) \\
&= \Pr_{(\mathbf{x},y)\sim D}\left(W_Q(\mathbf{x},y) - R(G_Q) \geq \frac{1}{2} - R(G_Q)\right)
\end{aligned}$$

Le résultat découle alors de l'application du théorème 3.2.1 avec $X = W_Q(\mathbf{x},y) - R(G_Q)$ et $a = 1/2 - R(G_Q)$. (Rappel : $R(G_Q) = \underset{(\mathbf{x},y)\sim D}{\mathbf{E}} W_Q(\mathbf{x},y)$.) ■

Définition 3.2.3. *Nous notons C_Q la quantité à droite de l'inégalité du corollaire 3.2.2, c'est-à-dire*

$$C_Q \stackrel{\text{déf}}{=} \frac{\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y)}{\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y) + (1/2 - R(G_Q))^2}.$$

La borne du corollaire 3.2.2 peut alors s'énoncer ainsi : $R(B_Q) \leq C_Q$.

3.2.1 Comparaison entre C_Q et $2R(G_Q)$

Un avantage important de la borne que fournit C_Q sur le risque du vote de majorité comparé à la borne $2R(G_Q)$ est que cette première peut être, dans de bonnes circonstances, très près de zéro, et ce même si chacun des votants présents dans le vote de majorité possède un grand risque, voire près de une demie.

Cependant, il n'est pas garanti que la quantité C_Q sera toujours plus petite que $2R(G_Q)$, et donc qu'elle mène à une borne plus serrée du risque du vote de majorité. La proposition suivante fournit une borne de C_Q en fonction de $R(G_Q)$.

Proposition 3.2.4. *Soit Q une distribution de probabilité sur un ensemble $\mathcal{H}$ de classificateurs binaires. Alors*

$$C_Q \leq 4\Big(R(G_Q) - R(G_Q)^2\Big).$$

Démonstration : En fixant la valeur de $R(G_Q)$, la quantité C_Q devient une fonction de $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y)$ strictement croissante. La variance, qui est donnée par $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y) = e_Q - R(G_Q)^2$, est maximisée lorsque la quantité e_Q est maximisée, et donc lorsque $e_Q = R(G_Q)$. On a ainsi

$$C_Q \leq \frac{R(G_Q) - R(G_Q)^2}{R(G_Q) - R(G_Q)^2 + \Big(\frac{1}{2} - R(G_Q)\Big)^2} = 4\Big(R(G_Q) - R(G_Q)^2\Big).$$

■

Puisque $4\Big(R(G_Q) - R(G_Q)^2\Big) \leq 4R(G_Q)$, la borne de $R(B_Q)$ fournie par la quantité C_Q sera dans le pire cas environ le double de celle donnée par $2R(G_Q)$. Dans l'autre sens cependant, la borne fournie par C_Q peut être «infiniment» plus petite que celle donnée par $2R(G_Q)$.

3.3 Relation entre d_Q, e_Q, s_Q, $R(G_Q)$ et $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x}, y)$

Une relation très étroite lie ensemble les cinq quantités données par le taux de désaccord, les taux d'erreurs et de succès conjoints et l'espérance et la variance de W_Q. En fait, seulement deux des cinq quantités d_Q, e_Q, s_Q, $R(G_Q)$ et $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x}, y)$ sont nécessaires pour les déduire toutes. Nous montrons d'abord que seulement deux des quatre quantités d_Q, e_Q, s_Q et $R(G_Q)$ sont nécessaires pour les déduire toutes, l'égalité 3.1 permettra alors d'ajouter la variance de W_Q à ce groupe. Cette affirmation découle de deux égalités reliant ces quatre quantités, la première étant l'égalité 3.2

$$d_Q + e_Q + s_Q = 1,$$

qui découle directement de la définition 3.1.2, et la seconde égalité est donnée par la proposition suivante.

Proposition 3.3.1. *Les quantités d_Q, e_Q et $R(G_Q)$ sont liées entre elles par l'égalité*

$$d_Q = 2(R(G_Q) - e_Q).$$

De façon équivalente, nous avons les deux égalités suivantes :

$$R(G_Q) = \frac{d_Q}{2} + e_Q$$
$$e_Q = R(G_Q) - \frac{d_Q}{2}.$$

Démonstration : Tout d'abord, remarquons que

$$I(h_1(\mathbf{x}) \neq y) - I(h_1(\mathbf{x}) \neq y)I(h_2(\mathbf{x}) \neq y) = \begin{cases} 0 & \text{si } h_1(\mathbf{x}) = y \\ 1 & \text{si } h_1(\mathbf{x}) \neq y \text{ et } h_2(\mathbf{x}) = y \\ 0 & \text{si } h_1(\mathbf{x}) \neq y \text{ et } h_2(\mathbf{x}) \neq y. \end{cases}$$

Donc,

$$\begin{aligned} I(h_1(\mathbf{x}) \neq h_2(\mathbf{x})) =& I(h_1(\mathbf{x}) \neq y) - I(h_1(\mathbf{x}) \neq y)I(h_2(\mathbf{x}) \neq y) \\ &+ I(h_2(\mathbf{x}) \neq y) - I(h_2(\mathbf{x}) \neq y)I(h_1(\mathbf{x}) \neq y). \end{aligned}$$

Il suit que

$$\begin{aligned} 2(R(G_Q) - e_Q) &= \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} \mathop{\mathbf{E}}_{h_1\sim\mathcal{H}} \mathop{\mathbf{E}}_{h_2\sim\mathcal{H}} I(h_1(\mathbf{x}) \neq y) - I(h_1(\mathbf{x}) \neq y)I(h_2(\mathbf{x}) \neq y) \\ &\quad + \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} \mathop{\mathbf{E}}_{h_1\sim\mathcal{H}} \mathop{\mathbf{E}}_{h_2\sim\mathcal{H}} I(h_2(\mathbf{x}) \neq y) - I(h_2(\mathbf{x}) \neq y)I(h_1(\mathbf{x}) \neq y) \\ &= \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} \mathop{\mathbf{E}}_{h_1\sim\mathcal{H}} \mathop{\mathbf{E}}_{h_2\sim\mathcal{H}} I(h_1(\mathbf{x}) \neq h_2(\mathbf{x})) \\ &= d_Q\,. \end{aligned}$$

■

Connaissant deux des trois quantités d_Q, e_Q et $R(G_Q)$, la proposition 3.3.1 nous fournit alors la troisième, et l'égalité 3.2 donnera la valeur de s_Q. De même, connaissant deux des trois valeurs e_Q, d_Q et s_Q, l'égalité 3.2 fournit alors la troisième et la proposition 3.3.1 donnera la valeur de $R(G_Q)$. Voyons maintenant que l'on peut également retrouver la valeur de d_Q (et donc également la valeur de e_Q) en connaissant seulement les valeurs $R(G_Q)$ et s_Q.

Corollaire 3.3.2. *Les quantités d_Q, s_Q et $R(G_Q)$ sont liées par l'égalité*

$$d_Q = 2(1 - R(G_Q) - s_Q).$$

Démonstration : De l'égalité 3.2 et de la proposition 3.3.1, on obtient les égalités $e_Q = 1 - d_Q - s_Q$ et $e_Q = R(G_Q) - d_Q/2$, en combinant ces deux égalités on trouve

$$1 - d_Q - s_Q = R(G_Q) - \frac{d_Q}{2},$$

et donc

$$d_Q = 2(1 - R(G_Q) - s_Q).$$

■

Le corollaire 3.2.2 permet d'obtenir une borne de $R(B_Q)$ en fonction des deux quantités $R(G_Q)$ et $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x}, y)$. Le corollaire 3.3.1, avec l'égalité 3.2 et le corollaire 3.3.2, nous indique qu'il est possible de récrire cette borne de $R(B_Q)$ en une borne dépendant de n'importe quelles deux valeurs parmi d_Q, e_Q, s_Q, $R(G_Q)$ et $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x}, y)$. L'écriture de la borne en fonction seulement de d_Q et $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x}, y)$, que nous donnons dans le corollaire 3.3.3, est particulièrement intéressante. En effet, la quantité d_Q peut s'évaluer par apprentissage non supervisé, c'est-à-dire qu'il n'est pas nécessaire de connaitre les étiquettes des exemples pour évaluer cette quantité. Pour simplifier, supposons que nous connaissons exactement d_Q, dans de telles situations, la borne fournie par l'inégalité de Tchebychev devient linéaire en la variance. C'est-à-dire que, par exemple, l'amélioration par un facteur 2 de la borne sur la variance améliore également d'un facteur 2 la borne sur $R(B_Q)$. L'utilisation de données non étiquetées ne pourra alors être avantageuse que si l'on possède une borne très serrée de la variance de W_Q.

Corollaire 3.3.3. *Soit $\mathcal{H}$ un ensemble de classificateurs binaires et Q une distribution sur $\mathcal{H}$. Si $R(G_Q) < \frac{1}{2}$, alors*

$$R(B_Q) \leq \frac{\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x}, y)}{\frac{1}{4} - \frac{d_Q}{2}}.$$

Démonstration : Selon le corollaire 3.2.2, il suffit de démontrer que

$$\frac{1}{4} - \frac{d_Q}{2} = \underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x}, y) + \left(\frac{1}{2} - R(G_Q)\right)^2.$$

Nous avons

$$\begin{aligned}\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x}, y) + \left(\frac{1}{2} - R(G_Q)\right)^2 &= \underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x}, y) + (R(G_Q))^2 + \frac{1}{4} - R(G_Q) \\ &= \frac{1}{4} + e_Q - R(G_Q) \\ &= \frac{1}{4} - \frac{d_Q}{2},\end{aligned}$$

où la dernière égalité provient de la proposition 3.3.1. ■

3.4 Optimalité de l'inégalité de Tchebychev

Sous certaines conditions, la borne obtenue de l'inégalité de Tchebychev (corollaire 3.2.2) est optimale. C'est-à-dire que, sous ces conditions, il n'est pas possible d'obtenir une borne supérieure du risque du vote de majorité qui soit inférieure à la quantité C_Q si les seules informations que nous avons concernant la distribution D générant les données sont l'espérance de $W_Q(\mathbf{x}, y)$ (soit $R(G_Q)$) ainsi que sa variance, et que nous ne faisons aucune hypothèse sur les valeurs prises par $W_Q(\mathbf{x}, y)$ (autre que $0 \leq W_Q(\mathbf{x}, y) \leq 1$).

Proposition 3.4.1. *La borne du corollaire 3.2.2 est atteinte, c'est-à-dire optimale, si et seulement si*

$$\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x}, y) \leq R(G_Q)\left(\frac{1}{2} - R(G_Q)\right).$$

Démonstration : Considérons que la distribution D générant les données soit telle que l'on ait $R(G_Q) = r$ et $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x}, y) = v$ et telle que $W_Q(\mathbf{x}, y)$ soit réparti en deux points : le point $\frac{1}{2}$ et un point s à déterminer. Nous avons donc

$$\begin{aligned}\Pr_{(\mathbf{x},y)\sim D}\left(W_Q(\mathbf{x}, y) = \frac{1}{2}\right) &= p \\ \Pr_{(\mathbf{x},y)\sim D}\left(W_Q(\mathbf{x}, y) = s\right) &= 1 - p,\end{aligned}$$

et nous sommes amenés à résoudre le système suivant :

$$\begin{aligned} \frac{p}{2} + (1-p)s &= r \\ \frac{p}{4} + (1-p)s^2 - r^2 &= v \\ s &\geq 0. \end{aligned}$$

De la première équation, nous trouvons

$$s = \frac{r - \frac{p}{2}}{1-p},$$

en portant cette valeur dans la deuxième équation, nous obtenons l'expression suivante pour p :

$$\frac{1}{4}p(1-p) + \left(r - \frac{p}{2}\right)^2 - r^2(1-p) - v(1-p) = 0,$$

donc

$$p = \frac{v}{v + \frac{1}{4} - r - r^2} = \frac{v}{v + \left(\frac{1}{2} - r\right)^2}.$$

Le système possèdera alors une solution si et seulement si la valeur pour s est positive, et donc si

$$\begin{aligned} 0 &\leq r - \frac{p}{2} \\ &= r - \frac{1}{2}\frac{v}{v + \left(\frac{1}{2} - r\right)^2} \\ &= rv + r\left(\frac{1}{2} - r\right)^2 - \frac{1}{2}v \\ &= \left(\frac{1}{2} - r\right)(-v) + r\left(\frac{1}{2} - r\right) \\ &= r\left(\frac{1}{2} - r\right) - v\,. \end{aligned}$$

Si $v \leq r\left(\frac{1}{2} - r\right)$, alors nous obtenons que le risque du vote de majorité pondéré par Q est donné par $\Pr_{(\mathbf{x},y)\sim D}\left(W_Q(\mathbf{x},y) \geq \frac{1}{2}\right) = p$, et donc égal à la borne donnée par le corollaire 3.2.2.

Donc lorsque $\mathbf{Var}_{(\mathbf{x},y)\sim D} W_Q(\mathbf{x},y) \leq R(G_Q)\left(\frac{1}{2} - R(G_Q)\right)$, la distribution D se confond avec une distribution possédant les mêmes propriétés (même risque de Gibbs et même variance) et engendrant un risque du vote de majorité égal à la borne du corollaire 3.2.2. Cette dernière est optimale.

Dans le cas contraire, si $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y) > R(G_Q)\left(\frac{1}{2} - R(G_Q)\right)$, nous avons

$$\begin{aligned} C_Q &= \frac{\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y)}{\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y) + \left(\frac{1}{2} - R(G_Q)\right)^2} \\ &> \frac{R(G_Q)\left(\frac{1}{2} - R(G_Q)\right)}{R(G_Q)\left(\frac{1}{2} - R(G_Q)\right) + \left(\frac{1}{2} - R(G_Q)\right)^2} \\ &= 2R(G_Q), \end{aligned}$$

et dans ce cas la borne du corollaire 3.2.2 est clairement non optimale, puisqu'elle est supérieure à celle découlant de l'inégalité 3.4 (qui se base sur l'inégalité de Markov au lieu de celle de Tchebychev). ■

3.5 Conclusion

Nous avons défini quelques quantités relatives à un problème de classification et nous avons démontré quelques propriétés que possèdent ces quantités. Nous avons de plus démontré un résultat (corollaire 3.2.2) permettant théoriquement de borner $R(B_Q)$ en fonction de $R(G_Q)$ et de $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y)$. Contrairement à la borne $R(B_Q) \leq 2R(G_Q)$, cette borne possède le grand avantage de pouvoir être voisine de zéro même dans des conditions où $R(G_Q)$ est aussi près que l'on veut de $\frac{1}{2}$ (pour autant qu'il soit inférieur à $\frac{1}{2}$).

La borne du corollaire 3.2.2 est cependant très sensible à l'erreur faite sur l'estimation de la variance de W_Q. En écrivant la borne sous la forme du corollaire 3.3.3, et en situation d'apprentissage semi-supervisé (où il est possible de bien estimer le taux de désaccord), le facteur d'erreur sur l'estimation de la variance devient même un facteur multiplicateur de la borne. Bien qu'il ne soit pas difficile d'obtenir une borne de $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y)$ à partir du théorème PAC-Bayes classique, la borne que l'on obtient n'est pas très serrée et donne lieu à des bornes de $R(B_Q)$ souvent supérieures à la simple borne $R(B_Q) \leq 2R(G_Q)$. Il est donc nécessaire, pour que le corollaire 3.2.2 fournisse une borne pratique, d'améliorer le théorème PAC-Bayes dans le but de borner plus efficacement la variance.

Chapitre 4

Bornes de la variance et nouvelles bornes de $R(B_Q)$

Dans ce chapitre, nous reprenons, en les améliorant et en les approfondissant, les résultats d'abord présentés dans Lacasse *et al.* (2007) permettant d'obtenir une borne de type PAC-Bayes de la quantité C_Q (ainsi que des autres quantités définies au chapitre 2, dont la variance de W_Q).

Le corollaire 3.2.2 fournit une borne de $R(B_Q)$ en fonction des quantités $R(G_Q)$ et $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y)$. Comme nous ne connaissons pas ces quantités, nous devons d'abord les borner pour ensuite en déduire une borne de $R(B_Q)$. En fait, nous pouvons récrire le corollaire sous cette forme plus précise.

Théorème 4.0.1. *Soit $\mathcal{H}$ un ensemble de classificateurs et Q une distribution sur $\mathcal{H}$. Soit $\overline{R} < \frac{1}{2}$ une borne supérieure de $R(G_Q)$ valide avec probabilité au moins $1-\delta_1$ et $\overline{V}$ une borne supérieure de $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y)$ valide avec probabilité au moins $1-\delta_2$. Alors, la borne suivante de $R(B_Q)$ est valide avec probabilité au moins $1-\delta_1-\delta_2$*

$$R(B_Q) \leq \frac{\overline{V}}{\overline{V} + \left(1/2 - \overline{R}\right)^2} .$$

Démonstration : La borne provient du corollaire 3.2.2, en effet, en écrivant la borne du corollaire sous la forme équivalente

$$R(B_Q) \leq \left(1 + \frac{(1/2 - R(G_Q))^2}{\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y)}\right)^{-1} ,$$

nous voyons qu'elle est une fonction croissante à la fois comme fonction de $R(G_Q)$ et comme fonction de $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y)$. L'inégalité sera conservée en remplaçant les vraies valeurs de $R(G_Q)$ et de $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y)$ par leurs bornes supérieures. La borne est alors valide avec probabilité au moins $1-\delta_1-\delta_2$ puisque les bornes sur $R(G_Q)$ et sur $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y)$ sont simultanément valides avec probabilité au moins $1-(\delta_1+\delta_2)$. ■

Pour appliquer le théorème 4.0.1, il nous faut alors être capable de borner supérieurement le risque de Gibbs et sa variance. Le théorème PAC-Bayes classique fournit déjà une borne serrée de $R(G_Q)$. Différentes approches peuvent être utilisées pour borner $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y)$, l'approche proposée à la section 4.6 se base sur l'égalité

$$\begin{aligned}\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y) &= \underset{(\mathbf{x},y)\sim D}{\mathbf{E}} (W_Q(\mathbf{x},y))^2 - \left(\underset{(\mathbf{x},y)\sim D}{\mathbf{E}} W_Q(\mathbf{x},y)\right)^2 \\ &= e_Q - (R(G_Q))^2.\end{aligned}$$

Il reste alors à trouver comment borner supérieurement la quantité e_Q.

4.1 Amélioration des bornes pour les petits votes de majorité

Dans certaines situations, il est possible d'améliorer les bornes déduites de l'inégalité

$$R(B_Q) \leq \Pr_{(\mathbf{x},y)\sim D}\left(W_Q(\mathbf{x},y) \geq \frac{1}{2}\right).$$

L'idée pour améliorer ces bornes est de constater que pour certains votes de majorité, il est impossible d'avoir $W_Q(\mathbf{x},y) = \frac{1}{2}$. En fait, pour une distribution Q donnée, la plus petite valeur supérieure ou égale à $\frac{1}{2}$ que peut prendre $W_Q(\mathbf{x},y)$ est donnée par $\frac{1}{2}+\eta_Q$, où η_Q est défini comme suit

$$\eta_Q \stackrel{\text{def}}{=} \inf_{\substack{\mathcal{V}\subseteq\mathcal{H}:\\ Q(\mathcal{V})\geq\frac{1}{2}}} \left(Q(\mathcal{V}) - \frac{1}{2}\right).$$

Nous avons alors l'inégalité suivante pour le risque de Bayes en fonction de η_Q

$$R(B_Q) \leq \Pr_{(\mathbf{x},y)\sim D}\left(W_Q(\mathbf{x},y) \geq \frac{1}{2}+\eta_Q\right). \tag{4.1}$$

Ainsi, l'inégalité habituelle $R(B_Q) \leq 2R(G_Q)$ devient (par l'inégalité de Markov appliquée à la probabilité de l'inégalité 4.1) :

$$R(B_Q) \leq \frac{2}{1+2\eta_Q} R(G_Q)\,.$$

Exemple 4.1.1. *Supposons que $\mathcal{H}$ soit de taille n où n est un entier impair, et que Q soit la distribution uniforme sur $\mathcal{H}$. Dans ces conditions, il n'est pas difficile de vérifier que l'on obtient alors $\eta_Q = \frac{s+1}{2s+1} - \frac{1}{2}$ où $s = \left\lfloor \frac{n}{2} \right\rfloor$. Par exemple, pour $n = 3$, on trouve $\eta_Q = \frac{1}{6}$, il suit que l'on obtient la borne $R(B_Q) \leq \frac{2}{1+\frac{1}{3}} = \frac{3}{2} R(G_Q)$.*

Comme le démontre l'exemple précédent, l'utilisation de la valeur η_Q dans le calcul des bornes peut réduire significativement celles-ci, dans l'exemple nous retrouvons même une diminution de 25% de la borne. Cependant, pour des distributions différentes de l'uniforme, l'évaluation de η_Q peut s'avérer très difficile. De plus, pour toute distribution proche de la distribution uniforme sur un ensemble possédant un grand nombre de classificateurs, la valeur de η_Q sera très proche de zéro. Comme la notation η_Q rend plus lourde l'écriture des bornes, et qu'il n'est pas clair qu'elle soit utilisable en pratique, nous ne l'avons pas utilisée dans l'écriture des théorèmes.

4.2 Comportement de la variance

Avant d'aller plus loin et de démontrer comment borner la variance, regardons pourquoi il semble être prometteur de se pencher sur ce problème. Nous savons, du corollaire 3.2.2 et des théorèmes qui en découlent (théorèmes 4.0.1 et 4.6.2), qu'une petite variance de W_Q implique un faible risque du vote de majorité (pour autant que $R(G_Q) < \frac{1}{2}$). Nous pouvons alors nous demander quelles sont les conditions pour que la variance soit faible, et si ces conditions sont souvent satisfaites.

4.2.1 Covariance des erreurs

La variance de W_Q peut être interprétée en termes de l'espérance de la covariance des erreurs entre paires de classificateurs de la façon suivante :

$$\begin{aligned}
\operatorname*{\mathbf{Var}}_{(\mathbf{x},y)\sim D} W_Q(\mathbf{x},y) &= \operatorname*{\mathbf{E}}_{(\mathbf{x},y)\sim D}\left(\Big(W_Q(\mathbf{x},y)\Big)^2 - \left(\operatorname*{\mathbf{E}}_{(\mathbf{x},y)\sim D} W_Q(\mathbf{x},y)\right)^2\right) \\
&= \operatorname*{\mathbf{E}}_{(\mathbf{x},y)\sim D}\left(\operatorname*{\mathbf{E}}_{h_1\sim Q} I(h_1(\mathbf{x})\neq y)\operatorname*{\mathbf{E}}_{h_2\sim Q} I(h_2(\mathbf{x})\neq y)\right) - (R(G_Q))^2 \\
&= \operatorname*{\mathbf{E}}_{(\mathbf{x},y)\sim D}\operatorname*{\mathbf{E}}_{h_1\sim Q}\operatorname*{\mathbf{E}}_{h_2\sim Q}\Big(I(h_1(\mathbf{x})\neq y)I(h_2(\mathbf{x})\neq y) - R(h_1)R(h_2)\Big) \\
&= \operatorname*{\mathbf{E}}_{h_1\sim Q}\operatorname*{\mathbf{E}}_{h_2\sim Q}\left(\operatorname*{\mathbf{E}}_{(\mathbf{x},y)\sim D} I(h_1(\mathbf{x})\neq y)I(h_2(\mathbf{x})\neq y) - R(h_1)R(h_2)\right) \\
&\overset{\text{déf}}{=} \operatorname*{\mathbf{E}}_{h_1\sim Q}\operatorname*{\mathbf{E}}_{h_2\sim Q}\mathrm{cov}_{\mathrm{err}}(h_1,h_2)\,. \qquad (4.2)
\end{aligned}$$

De cette écriture de la variance, nous pouvons déduire la proposition suivante, qui permet à son tour d'affirmer que lorsque le nombre de votants tend vers l'infini et que le poids de chaque votant tend vers zéro, une condition suffisante pour que la variance de W_Q tende également vers zéro est qu'en moyenne la covariance des erreurs entre paires de classificateurs distincts ne soit pas positive.

Proposition 4.2.1. *Soit $\mathcal{H}$ un ensemble dénombrable de classificateurs et soit Q une distribution sur $\mathcal{H}$. Alors, nous avons l'inégalité suivante concernant la variance de W_Q*

$$\operatorname*{\mathbf{Var}}_{(\mathbf{x},y)\sim D} W_Q(\mathbf{x},y) \;\leq\; \frac{1}{4}\sum_{h\in\mathcal{H}} Q^2(h) + \sum_{h_1\in\mathcal{H}}\sum_{\substack{h_2\in\mathcal{H}:\\ h_2\neq h_1}} Q(h_1)Q(h_2)\mathrm{cov}_{\mathrm{err}}(h_1,h_2)\,.$$

Démonstration : De l'égalité 4.2 on trouve directement

$$\operatorname*{\mathbf{Var}}_{(\mathbf{x},y)\sim D} W_Q(\mathbf{x},y) = \sum_{h\in\mathcal{H}} Q^2(h)\mathrm{cov}_{\mathrm{err}}(h,h) + \sum_{h_1\in\mathcal{H}}\sum_{\substack{h_2\in\mathcal{H}:\\ h_2\neq h_1}} Q(h_1)Q(h_2)\mathrm{cov}_{\mathrm{err}}(h_1,h_2)\,.$$

La covariance des erreurs d'un classificateur h avec lui-même est égale à

$$\begin{aligned}
\mathrm{cov}_{\mathrm{err}}(h,h) &= \operatorname*{\mathbf{E}}_{(\mathbf{x},y)\sim D} I(h(\mathbf{x})\neq y)I(h(\mathbf{x})\neq y) - R(h)R(h) \\
&= R(h) - (R(h))^2 = R(h)(1-R(h)),
\end{aligned}$$

où $R(h)$ correspond au risque de h, et donc $0\leq R(h)\leq 1$, il suit que $\mathrm{cov}_{\mathrm{err}}(h,h)\leq\frac{1}{4}$, ce qui implique le résultat. ■

La proposition 4.2.1 indique des situations dans lesquelles la variance de W_Q diminue lorsque l'on modifie la distribution Q de sorte à lui ajouter des classificateurs de masse non nulle. La diminution de la variance affectera directement la borne sur le risque du vote de majorité. Considérons par exemple une situation dans laquelle Q_n est une distribution uniforme sur n classificateurs deux à deux indépendants. Donc $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_{Q_n}(\mathbf{x},y) \leq \frac{1}{4}\sum_{h\in\mathcal{H}} Q_n^2(h) = \frac{1}{4n}$. Si de plus nous savons que pour tout n, $R(G_{Q_n}) < \frac{1}{2} - \epsilon$, pour une certaine constante $\epsilon > 0$, le théorème 4.0.1 permet alors de déduire la borne suivante sur le risque du vote de majorité :

$$R(B_{Q_n}) \leq \frac{1}{1+4n\epsilon^2}\,.$$

Le corollaire suivant, qui découle de la proposition 4.2.1 et du théorème 4.0.1, nous indique de manière plus précise comment ceci peut affecter la borne sur $R(B_Q)$.

Corollaire 4.2.2. *Soit $\epsilon > 0$ et soit Q une distribution sur un ensemble discret de classificateurs $\mathcal{H}$ telle que*

$$R(G_Q) \leq \frac{1}{2} - \epsilon \quad \textit{et} \quad \sum_{h_1\in\mathcal{H}} \sum_{\substack{h_2\in\mathcal{H}:\\ h_2\neq h_1}} Q(h_1)Q(h_2)\mathrm{cov}(h_1,h_2) \leq 0\,.$$

Alors

$$R(B_Q) \leq \frac{\zeta}{\zeta+4\epsilon^2} \qquad \left(< \frac{\zeta}{4\epsilon^2}\right)$$

où $\zeta = \sup_{h\in\mathcal{H}}(Q(h))$.

Démonstration : Par la proposition 4.2.1 et l'inégalité $\sum_{h\in\mathcal{H}} Q^2(h) \leq \sup_{h\in\mathcal{H}}(Q(h))$, on a

$$\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y) \leq \frac{\zeta}{4}\,.$$

Comme de plus nous avons par hypothèse $R(G_Q) < \frac{1}{2} - \epsilon$, le théorème 4.0.1 nous donne alors l'inégalité

$$R(B_Q) \leq \frac{\zeta/4}{\zeta/4 + (1/2 - (1/2-\epsilon))^2} = \frac{\zeta}{\zeta+4\epsilon^2}\,.$$

■

4.3 Observation empirique sur des ensembles test

Nous avons comparé, sur plusieurs problèmes de classificateurs binaires provenant des ensembles UCI (voir Blake et Merz, 1998), le comportement des quantités $R(G_Q)$,

$\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x}, y)$ et C_Q (définition 3.2.3) en fonction de $R(B_Q)$. Les résultats de la figure 4.1 ont été obtenus avec l'algorithme AdaBoost en utilisant des souches de décision (arbres de décision à une couche) comme classificateurs de base. Tous les ensembles de données ont été séparés en deux sous-ensembles distincs : un ensemble d'entrainement et un ensemble test. L'apprentissage de l'algorithme s'est fait à partir des ensembles d'entrainement et les résultats affichés dans les graphiques proviennent d'observation faites sur les ensembles test.

Nous observons dans ces tests empiriques (voir figure 4.1) qu'il n'y a pas de corrélation directe entre $R(G_Q)$ et $R(B_Q)$ ni entre $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x}, y)$ et $R(B_Q)$. Donc aucune des quantités $R(G_Q)$ et $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x}, y)$ ne semble être un bon indicateur de $R(B_Q)$. Par contre, la quantité C_Q, qui utilise à la fois les informations sur $R(G_Q)$ et sur $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x}, y)$, parait clairement liée à $R(B_Q)$. La quantité C_Q semble donc être un très bon indicateur de $R(B_Q)$, ce qui motive à développer des bornes serrées de cette quantité.

4.4 Conditions sur $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x}, y)$ pour obtenir une borne précise

Nous avons vu, à la section 4.2, des conditions pour que la variance soit faible dans un vote de majorité et, à la section 4.3, nous avons observé empiriquement, en calculant C_Q sur des ensembles test, que le corollaire 3.2.2 semble être un très bon indicateur de la qualité d'un vote de majorité.

Nous voyons dans cette section pourquoi il est important d'avoir une borne très serrée de $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x}, y)$ pour borner C_Q. Les deux prochaines propositions donnent les conditions nécessaires et suffisantes sur $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x}, y)$ pour que C_Q soit inférieur à $2R(G_Q)$ et pour qu'il soit inférieur à $R(G_Q)$. Cela nous indiquera entre autres dans quels types de vote de majorité la borne de C_Q sera une borne plus précise de $R(B_Q)$ que la borne classique $2R(G_Q)$.

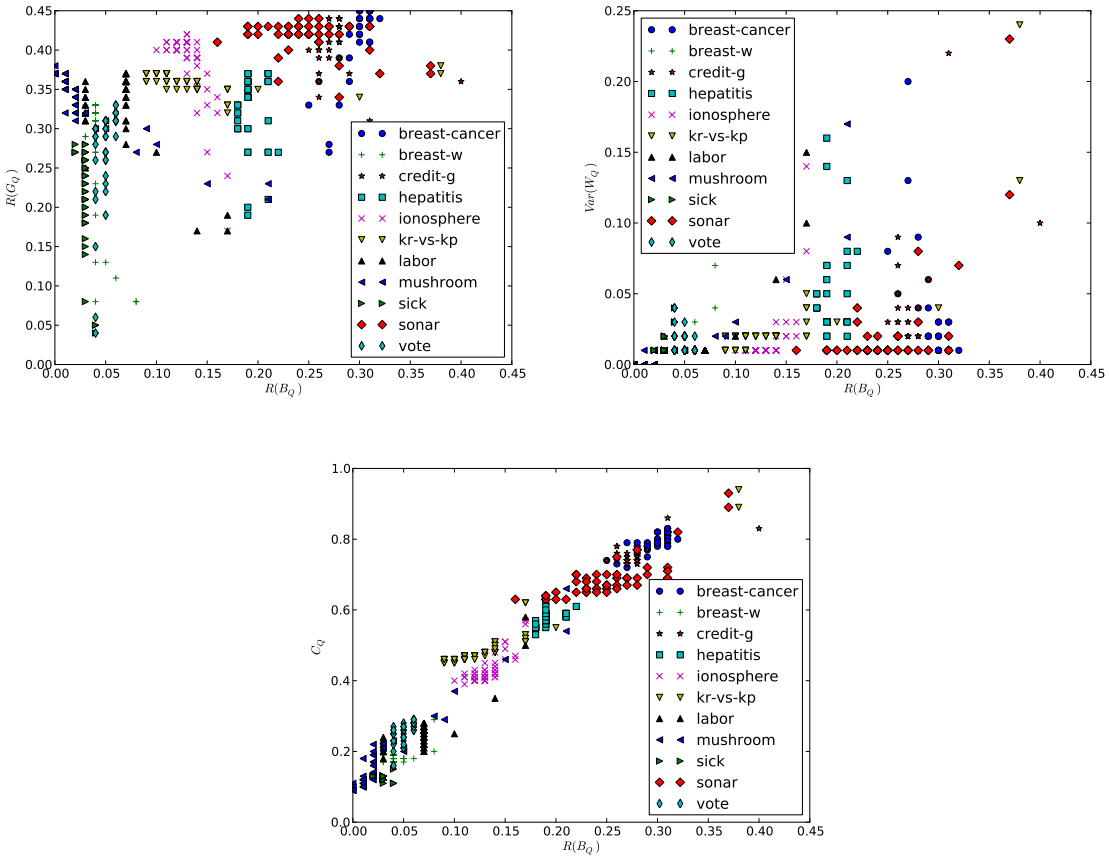

FIGURE 4.1 – Relation sur plusieurs ensembles de données entre $R(B_Q)$ et $R(G_Q)$ (en haut à gauche), entre $R(G_Q)$ et $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x}, y)$ (en haut à droite) et entre $R(B_Q)$ et C_Q (en bas).

4.4.1 Obtenir une borne meilleure que $2R(G_Q)$

Proposition 4.4.1. *La quantité C_Q est inférieure à $2R(G_Q)$ si et seulement si*

$$\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x}, y) < \frac{1}{2} R(G_Q)(1 - 2R(G_Q)) \,.$$

Démonstration :

$$
\begin{aligned}
& C_Q < 2R(G_Q) \\
\iff \quad & \underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y) < 2R(G_Q)\left(\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y) + (1/2 - R(G_Q))^2\right) \\
\iff \quad & \underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y)(1-2R(G_Q)) < 2R(G_Q)\,(1/2 - R(G_Q))^2 \\
\iff \quad & \underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y) < \frac{1}{2}R(G_Q)(1-2R(G_Q))
\end{aligned}
$$

■

La figure 4.2 donne le tracé de la fonction à droite de l'inégalité de la proposition 4.4.1. Cette fonction est concave et atteint son maximum au point 1/4. On en conclut que c'est lorsque $R(G_Q)$ est près de 1/4 que la borne de C_Q devrait le plus avantageusement se comparer à la borne $2R(G_Q)$. Par contre, si $R(G_Q)$ est très faible ou s'il approche 1/2, il devient difficile d'obtenir une valeur de C_Q inférieure à $2R(G_Q)$.

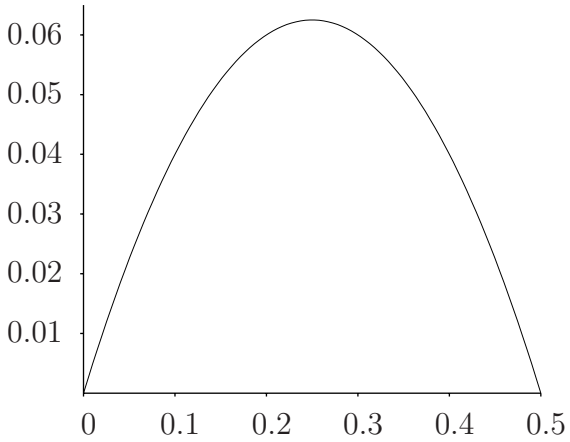

FIGURE 4.2 – Tracé de $\frac{1}{2}R(G_Q)(1-2R(G_Q))$ en fonction de $R(G_Q)$

4.4.2 Obtenir une borne meilleure que $R(G_Q)$

Nous voulons non seulement obtenir une borne plus serrée que $2R(G_Q)$, nous désirons également caractériser les bons votes de majorité, soit les votes tels que $R(B_Q) \ll R(G_Q)$. La proposition suivante nous donne une condition nécessaire et suffisante pour que la quantité C_Q soit plus petite que $R(G_Q)$.

Proposition 4.4.2. *La quantité C_Q est inférieure à $R(G_Q)$ si et seulement si*

$$\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x}, y) < \frac{R(G_Q)(\frac{1}{2} - R(G_Q))^2}{1 - R(G_Q)} .$$

Démonstration : Se fait en procédant comme pour la démonstration de la proposition 4.4.1. ■

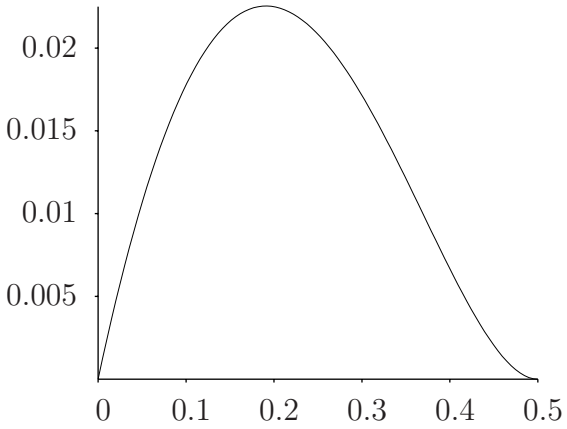

FIGURE 4.3 – Tracé de $\frac{R(G_Q)(\frac{1}{2}-R(G_Q))^2}{1-R(G_Q)}$ en fonction de $R(G_Q)$

La figure 4.3 donne le tracé de la fonction de la proposition 4.4.2. Nous voyons sur la figure que la variance de W_Q doit être très faible comparée à $R(G_Q)$ pour que C_Q soit inférieure à $R(G_Q)$. Cette fois, le point optimal se situe près de $R(G_Q) = 0,2$ et la dégradation se produit plus lentement en approchant de $R(G_Q) = 1/2$ que dans la fonction de la figure 4.2.

Les propositions 4.4.1 et 4.4.2 s'appliquent de la même manière lorsqu'on ne travaille pas avec des quantités exactes mais avec des bornes supérieures de $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x}, y)$ et $R(G_Q)$. Par exemple, en considérant $\overline{R}$ et $\overline{V}$ des bornes supérieures de $R(G_Q)$ et de $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x}, y)$ respectivement, pour $\overline{V} = 0.02$, nous obtenons une borne de C_Q (et donc une borne de $R(B_Q)$) inférieure à $\overline{R}$ seulement si $0.12 < \overline{R} < 0.26$. Cette condition est extrêmement restrictive pour plusieurs algorithmes d'apprentissage. Par exemple, avec le *boosting*, il est fréquent d'observer une valeur empirique de $R(G_Q)$ avoisinant 0.4.

4.5 Utiliser les moments supérieurs

Le théorème PAC-Bayes classique donne une borne de $R(G_Q)$ qui se traduit en une borne de $R(B_Q)$ grâce à la relation $R(B_Q) \leq 2R(G_Q)$ provenant de l'inégalité de Markov appliquée à $\underset{(\mathbf{x},y)\sim D}{\mathbf{E}} W_Q(\mathbf{x}, y) = R(G_Q)$, soit le premier moment de la variable aléatoire W_Q. Le théorème 4.0.1 utilise pour sa part les deux premiers moments de la variable aléatoire W_Q, soit $\underset{(\mathbf{x},y)\sim D}{\mathbf{E}} W_Q(\mathbf{x}, y)$ et $\underset{(\mathbf{x},y)\sim D}{\mathbf{E}} W_Q(\mathbf{x}, y)^2 = e_Q$, pour obtenir une borne plus précise de $R(B_Q)$. Rien n'empêche d'utiliser également des moments d'ordre supérieur à 2 pour obtenir une borne encore plus précise de $R(B_Q)$, ce que suggère le théorème suivant.

Théorème 4.5.1. *Soit $k \in \mathbb{N}$. Soit $\mathcal{H}$ un ensemble de classificateurs et Q une distribution sur $\mathcal{H}$. Si $(1/2 - R(G_Q))^k - \underset{(\mathbf{x},y)\sim D}{\mathbf{E}}((W_Q(\mathbf{x}, y) - R(G_Q))^k) = \tau > 0$ alors*

$$R(B_Q) \leq \frac{\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}}((W_Q(\mathbf{x}, y) - R(G_Q))^k)}{\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}}((W_Q(\mathbf{x}, y) - R(G_Q))^k) + \tau^2}\,.$$

À noter que ce théorème se réduit au théorème 4.0.1 lorsque $k = 1$ et $R(G_Q) < \frac{1}{2}$. En effet, dans ce cas, $\underset{(\mathbf{x},y)\sim D}{\mathbf{E}}(W_Q(\mathbf{x}, y) - R(G_Q)) = \underset{(\mathbf{x},y)\sim D}{\mathbf{E}} W_Q(\mathbf{x}, y) - R(G_Q) = R(G_Q) - R(G_Q) = 0$.

Démonstration : L'inégalité de Tchebychev peut être généralisée aux moments supérieurs en notant que $\forall k \in \mathbb{N}$, nous avons

$$\begin{aligned}
\Pr\,(X \geq a) &= \Pr\,(X - \mu \geq a - \mu) \\
&\leq \Pr\left((X - \mu)^k \geq (a - \mu)^k\right) \\
&= \Pr\left((X - \mu)^k \geq (a - \mu)^k - \mathbf{E}((X - \mu)^k) + \mathbf{E}((X - \mu)^k)\right)
\end{aligned}$$

où $\mu = \mathbf{E}\,[X]$. Si $(a - \mu)^k - \mathbf{E}((X - \mu)^k) > 0$, nous pouvons appliquer l'inégalité de Tchebychev pour obtenir

$$\Pr\,((X - \mu)^k \geq (a - \mu)^k) \leq \frac{\mathbf{Var}((X-\mu)^k)}{\mathbf{Var}((X-\mu)^k) + ((a-\mu)^k - \mathbf{E}((X-\mu)^k))^2}\,. \quad (4.3)$$

Nous pouvons alors borner $R(B_Q)$ en notant que

$$R(B_Q) \leq \underset{(\mathbf{x},y)\sim D}{\Pr}\left(W_Q(\mathbf{x}, y) \geq \frac{1}{2}\right) = \underset{(\mathbf{x},y)\sim D}{\Pr}\left(W_Q(\mathbf{x}, y) - R(G_Q) \geq \frac{1}{2} - R(G_Q)\right)$$

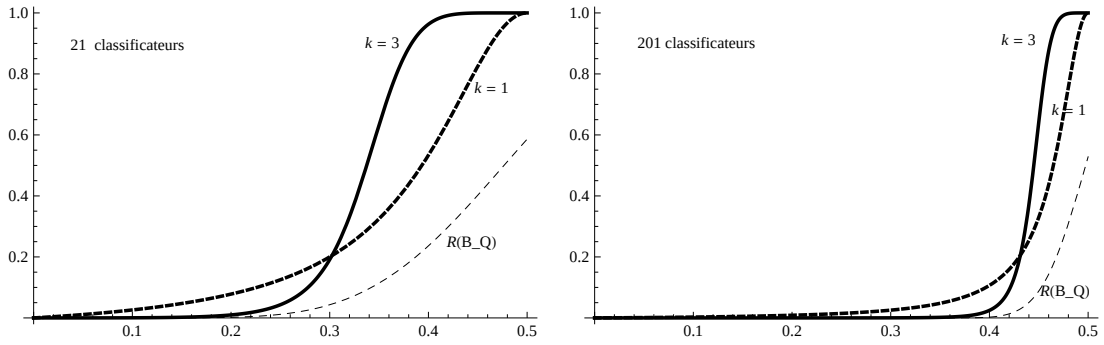

FIGURE 4.4 – Impact de k et $n = |\mathcal{H}|$ sur la borne de $R(B_Q)$ donnée par le théorème 4.5.1.

et en remplaçant a par $\frac{1}{2}$, X par $W_Q(\mathbf{x}, y)$ et μ par $R(G_Q)$ dans l'équation 4.3. ■

Dans certaines situations, le théorème 4.5.1 peut donner des bornes plus serrées de $R(B_Q)$ que le théorème 4.0.1. Ceci peut s'expliquer par le fait que $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}}((W_Q(\mathbf{x}, y) - R_Q(G)^k)$ tend plus rapidement vers zéro que $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x}, y)$. Mais pour utiliser le théorème 4.5.1, nous devons satisfaire $(1/2 - R(G_Q))^k - \underset{(\mathbf{x},y)\sim D}{\mathbf{E}} \left((W_Q(\mathbf{x}, y) - R(G_Q))^k\right) > 0$, et cette condition n'est pas toujours satisfaite.

La figure 4.4 illustre l'impact de k et de $|\mathcal{H}|$ sur la borne donnée par le théorème 4.5.1. Pour cet exemple, nous avons considéré un ensemble $\mathcal{H}$ de classificateurs ayant des probabilités indépendantes d'erreur en fonction de la distribution D. Dans ce cas, nous avons $\text{cov}(h, h') = 0$ pour tout $h, h' \in \mathcal{H}$, $h \neq h'$. De plus, nous avons supposé que chaque classificateur $h \in \mathcal{H}$ possédait le même risque $R(h) = R(G_Q) =: R$. Une telle situation ne peut se produire dans la pratique, cependant, cela permet de comparer les différentes bornes du théorème 4.5.1 en fonction de k, puisque nous pouvons calculer de façon exacte les différentes quantités en jeu.

Dans cette situation, pour un ensemble $\mathcal{H}$ de n classificateurs, la probabilité que k classificateurs sur les n classifient incorrectement un exemple choisi arbitrairement est donnée par $\binom{n}{k} R^k (1 - R)^{n-k}$, puisqu'elle suit une distribution binomiale. La fonction génératrice des moments, $M(t)$, associée à la variable aléatoire correspondant à $W_Q(\mathbf{x}, y)$ est donnée par

$$M(t) = \left(R \cdot \exp\left(\frac{t}{n}\right) + 1 - R\right)^n.$$

Il suit que nous avons

$$\underset{(\mathbf{x},y)\sim D}{\mathbf{E}} (W_Q(\mathbf{x}, y))^k = \frac{\partial^k}{\partial t^n} \left(R \cdot \exp\left(\frac{t}{n}\right) + 1 - R\right)^n \Bigg|_{t=0}.$$

Nous pouvons alors évaluer ces quantités à l'aide d'un logiciel de calculs mathématiques, tel que Mathematica, et ainsi évaluer la valeur exacte de la borne du théorème 4.5.1.

Nous observons que l'utilisation de moments supérieurs réduit la borne de $R(B_Q)$ lorsque la variance est faible (celle-ci diminuant avec le nombre de classificateurs). Dans la figure 4.4, nous observons que la borne obtenue avec $k = 3$ et $|\mathcal{H}| = 21$ est meilleure que la borne avec $k = 1$ pour les petites valeurs de $R(G_Q)$ (approximativement pour $R(G_Q) < 1/3$), de plus nous voyons que l'intervalle pour lequel $k = 3$ donne une meilleure borne que $k = 1$ s'agrandit lorsque $|\mathcal{H}|$ augmente, c'est-à-dire lorsque la diversité des classificateurs augmente.

4.5.1 Autre méthode pour utiliser les moments supérieurs

Le théorème 4.5.1 se base sur une généralisation de l'inégalité de Tchebychev exploitant les moments supérieurs de W_Q pour donner une borne plus serrée de $R(B_Q)$. Il est également possible de faire la même chose avec l'inégalité de Markov, ce qui mène à la borne suivante :

$$\begin{aligned} R(B_Q) \leq \Pr_{(\mathbf{x},y)\sim D}\left(W_Q(\mathbf{x},y) \geq \frac{1}{2}\right) &= \Pr_{(\mathbf{x},y)\sim D}\left((W_Q(\mathbf{x},y))^n \geq \left(\frac{1}{2}\right)^n\right) \\ &\leq \frac{\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D}(W_Q(\mathbf{x},y))^n}{\left(\frac{1}{2}\right)^n} \\ &= 2^n \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D}(W_Q(\mathbf{x},y))^n. \end{aligned}$$

Par exemple, pour $n = 2$, on obtient la borne

$$R(B_Q) \leq 4e_Q.$$

Cette approche pour exploiter les moments supérieurs de la variable aléatoire W_Q possède cependant les deux inconvénients suivants.

- Dès que le vote de majorité produit une erreur, la quantité $\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D}((W_Q(\mathbf{x},y))^n$ ne tend pas vers zéro et donc, la valeur de la borne tend vers l'infini avec n.
- Même lorsque le vote de majorité est parfait, la borne de $\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D}((W_Q(\mathbf{x},y))^n$ est toujours supérieure à zéro (et grandit avec n), et la borne de $R(B_Q)$ tendra également vers l'infini.

Le corollaire 4.5.1 permet pour sa part d'obtenir une valeur de borne toujours comprise dans l'intervalle $(0, 1]$.

Malgré ces inconvénients, pour de faibles valeurs de n, cette approche permet parfois d'obtenir une borne de $R(B_Q)$ inférieure à la borne obtenue de l'inégalité $R(B_Q) \leq$

$2R(G_Q)$.

4.5.2 Remarque sur les bornes des moments supérieurs

Le théorème 4.5.1 donne espoir de pouvoir obtenir une borne plus serrée de $R(B_Q)$ en utilisant les moments supérieurs de la variable aléatoire W_Q. Bien que les résultats affichés à la figure 4.4 soient encourageant, ceux-ci s'appuient sur des valeurs exactes des moments supérieurs et non pas sur une borne de ceux-ci. Comme nous le verrons plus loin, il est difficile d'obtenir une borne serrée des moments supérieurs. En pratique, l'utilisation des moments supérieurs ne permettra donc pas d'obtenir des bornes plus serrées de $R(B_Q)$.

4.6 Borner la quantité e_Q

Nous sommes intéressés à trouver une borne de $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y)$. Puisque par définition $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y) = e_Q - R(G_Q)^2$, nous pouvons borner supérieurement la variance en bornant inférieurement $R(G_Q)$ (ce que fait déjà le théorème PAC-Bayes classique) et en bornant supérieurement le taux moyen d'erreurs conjointes e_Q. Le théorème 4.6.1 permet de borner cette dernière quantité.

Théorème 4.6.1. *Soit $\delta \in (0,1]$. Soit $\mathcal{H}$ un ensemble de classificateurs binaires et P une distribution à priori sur $\mathcal{H}$. Alors nous avons*

$$\Pr_{S\sim D^m}\left(\forall Q \text{ sur } \mathcal{H} : \mathrm{kl}(\widehat{a_Q}\|a_Q) \leq \frac{1}{m}\left[2{\cdot}\mathrm{KL}(Q\|P) + \log\frac{(m+1)}{\delta}\right]\right) \geq 1-\delta$$

où a_Q est une valeur parmi d_Q, e_Q ou s_Q et $\widehat{a_Q}$ son estimation empirique sur S.

Démonstration :

Nous montrons d'abord le théorème pour $a_Q = e_Q$, soit le cas qui nous intéresse.

Pour $h_a, h_b \in \mathcal{H}$, considérons un classificateur $h_{a,b} \in \mathcal{H}\times\mathcal{H}$ de la forme $h_{a,b} : \mathcal{X} \to \mathcal{Y}$ auquel nous associons la perte $\ell(h_{a,b},\mathbf{x},y)$ définie comme suit :

$$\ell(h_{a,b},\mathbf{x},y) = \begin{cases} 1 & \text{si } h_a(\mathbf{x}) = h_b(\mathbf{x}) \neq y \\ 0 & \text{sinon}\,. \end{cases}$$

(Note : la définition exacte du classificateur $h_{a,b}$ n'est pas importante, seule la définition de la perte $\ell(h_{a,b}, \mathbf{x}, y)$ l'est.)

Nous notons $\mathcal{H}^{(2)}$ la classe des tels classificateurs. De plus, pour une distribution P donnée sur $\mathcal{H}$, nous notons $P^{(2)}$ la distribution sur $\mathcal{H}^{(2)}$ définie par

$$P^{(2)}(h_{a,b}) \stackrel{\text{déf}}{=} P(h_a)P(h_b)\,.$$

Le risque du classificateur $h_{a,b}$ est simplement égal au taux d'erreurs conjointes entre h_a et h_b. En effet

$$\begin{aligned} R(h_{a,b}) &= \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} \ell(h_{a,b}, \mathbf{x}, y) \\ &= \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} I(h_a(\mathbf{x}) = h_b(\mathbf{x}) \neq y) \\ &= \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} I(h_a(\mathbf{x}) \neq y \wedge h_b(\mathbf{x}) \neq y) \\ &= \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} I(h_a(\mathbf{x}) \neq y) I(h_b(\mathbf{x}) \neq y)\,. \end{aligned}$$

Comme pour la distribution $P^{(2)}$, notons $Q^{(2)}$ la distribution sur $\mathcal{H}^{(2)}$ donnée par $Q^{(2)}(h_{a,b}) = Q(h_a)Q(h_b)$. Notons $R(G_{Q^{(2)}})$ le classificateur de Gibbs associé à la distribution $Q^{(2)}$. Nous avons donc

$$\begin{aligned} R(G_{Q^{(2)}}) &= \mathop{\mathbf{E}}_{h_a\sim Q} \mathop{\mathbf{E}}_{h_b\sim Q} \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} I(h_a(\mathbf{x}) \neq y) I(h_b(\mathbf{x}) \neq y) \\ R_S(G_{Q^{(2)}}) &= \mathop{\mathbf{E}}_{h_a\sim Q} \mathop{\mathbf{E}}_{h_b\sim Q} \frac{1}{m} \sum_{i=1}^{m} I(h_a(\mathbf{x}_i) \neq y_i) I(h_b(\mathbf{x}_i) \neq y_i)\,. \end{aligned}$$

Ainsi, $R(G_{Q^{(2)}})$ est égal au taux moyen d'erreurs conjointes e_Q. Le théorème 2.3.3 s'applique directement pour borner le risque du classificateur $R(G_{Q^{(2)}})$. Par conséquent, pour toute distribution P sur $\mathcal{H}$ et pour tout $\delta \in (0, 1]$, nous avons

$$\Pr_{S\sim D^m}\left(\forall\, Q \text{ sur } \mathcal{H} :\ \mathrm{kl}(R_S(G_{Q^{(2)}}) \| R(G_{Q^{(2)}})) \leq \frac{1}{m}\left[\mathrm{KL}(Q^{(2)}\|P^{(2)}) + \ln\frac{m+1}{\delta}\right]\right) \geq 1-\delta\,.$$

Il reste finalement à démontrer que $\mathrm{KL}(Q^{(2)}\|P^{(2)}) = 2\mathrm{KL}(Q\|P)$, ce qui se fait comme suit :

$$\begin{aligned} \mathrm{KL}(Q^{(2)}\|P^{(2)}) &= \mathop{\mathbf{E}}_{(h_a,h_b)\sim Q^{(2)}} \log\frac{Q^{(2)}(h_a,h_b)}{P^{(2)}(h_a,h_b)} \\ &= \mathop{\mathbf{E}}_{(h_a,h_b)\sim Q^{(2)}} \log\frac{Q(h_a)Q(h_b)}{P(h_a)P(h_b)} \\ &= \mathop{\mathbf{E}}_{(h_a,h_b)\sim Q^{(2)}} \left[\log\frac{Q(h_a)}{P(h_a)} + \log\frac{Q(h_b)}{P(h_b)}\right] \\ &= 2\cdot\mathrm{KL}(Q\|P)\,. \end{aligned}$$

Pour les autres cas du théorème, soit les cas $a_Q = d_Q$ et $a_Q = s_Q$, la preuve demeure la même, excepté la définition de la perte $\ell(h_{a,b}, \mathbf{x}, y)$ qui doit être remplacée par

$$\ell(h_{a,b}, \mathbf{x}, y) = \begin{cases} 1 & \text{si } h_a(\mathbf{x}) \neq h_b(\mathbf{x}) \\ 0 & \text{sinon} \end{cases}$$

dans le cas $a_Q = d_Q$, et par

$$\ell(h_{a,b}, \mathbf{x}, y) = \begin{cases} 1 & \text{si } h_a(\mathbf{x}) = h_b(\mathbf{x}) = y \\ 0 & \text{sinon} \end{cases}$$

dans le cas $a_Q = s_Q$.

■

Nous pouvons maintenant récrire le théorème 4.0.1 en un théorème fournissant une borne de $R(B_Q)$ en fonction d'une borne supérieure de e_Q et de $R(G_Q)$ ainsi que d'une borne inférieure de $R(G_Q)$.

Théorème 4.6.2. *Soit $\mathcal{H}$ une classe de classificateurs binaires et Q une distribution sur $\mathcal{H}$. Soit $\overline{R}$ et $\underline{R}$ des bornes respectivement supérieure et inférieure de $R(G_Q)$ simultanément valides avec probabilité au moins $1 - \delta_1$. Soit $\overline{E}$ une borne supérieure de e_Q valide avec probabilité au moins $1 - \delta_2$. Alors, avec probabilité au moins $1 - \delta$, où $\delta = \delta_1 + \delta_2$, nous avons la borne suivante de $R(B_Q)$:*

$$R(B_Q) \leq \frac{\overline{E} - \underline{R}^2}{\overline{E} - \underline{R}^2 + \left(\frac{1}{2} - \overline{R}\right)^2} .$$

Démonstration : Conséquence directe du corollaire 3.2.2. Nous pouvons voir qu'il n'est pas nécessaire de posséder une borne inférieure de e_Q en écrivant l'inégalité sous la forme

$$R(B_Q) \leq \left(1 + \frac{\left(\frac{1}{2} - \overline{R}\right)^2}{\overline{E} - \underline{R}^2}\right)^{-1} .$$

De plus, l'inégalité tient avec probabilité au moins $1 - \delta$ car les bornes sur $R(G_Q)$ tiennent simultanément avec la borne sur e_Q avec probabilité au moins $1 - (\delta_1 + \delta_2) = 1 - \delta$.

■

4.7 Nouvelle borne PAC-Bayes

Le théorème 4.6.1 permet d'obtenir une borne de $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y)$ par l'intermédiaire d'une borne de e_Q. Le théorème suivant généralise plus profondément le théorème PAC-Bayes en donnant simultanément une borne de deux quantités choisies parmi d_Q, e_Q et s_Q, ce qui permettra d'obtenir directement une borne de $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y)$. En effet, de l'égalité 3.2 et de la proposition 3.3.1, il est facile de déduire les égalités suivantes entre $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y)$ et n'importe quelles paires parmi les quantités d_Q, e_Q et s_Q :

$$\begin{aligned}\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y) &= e_Q - \left(e_Q + \frac{d_Q}{2}\right)^2 \\ &= e_Q - \frac{1}{4}(1 + e_Q - s_Q)^2 \\ &= 1 - d_Q - s_Q - \left(1 - \frac{d_Q}{2} - s_Q\right)^2,\end{aligned} \tag{4.4}$$

ce qui permet de déduire (de trois manières différentes) une borne de $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y)$.

Théorème 4.7.1. *Soit $\mathcal{H}$ un ensemble de classificateurs et P une distribution sur $\mathcal{H}$ et soit $\delta \in (0,1]$. Alors nous avons*

$$\Pr_{S\sim D^m}\left(\forall Q : \mathrm{kl}(\widehat{a_Q}, \widehat{\beta_Q} \| a_Q, \beta_Q) \leq \frac{1}{m}\left[2{\cdot}\mathrm{KL}(Q\|P) + \ln\frac{(m+1)(m+2)}{\delta}\right]\right) \geq 1-\delta$$

où a_Q et β_Q sont deux valeurs parmi e_Q, s_Q et d_Q, $\widehat{a_Q}$ et $\widehat{\beta_Q}$ sont leurs valeurs empiriques observées sur l'ensemble S et

$$\mathrm{kl}(q_1, q_2 \| p_1, p_2) \stackrel{\text{déf}}{=} q_1 \log\frac{q_1}{p_1} + q_2 \log\frac{q_2}{p_2} + (1-q_1-q_2)\log\frac{1-q_1-q_2}{1-p_1-p_2}$$

est la divergence de Kullback-Leibler entre les distributions associées à deux variables aléatoires tri-valuées Y_q et Y_p avec $P(Y_q = a) = q_1$, $P(Y_q = b) = q_2$ et $P(Y_q = c) = 1 - q_1 - q_2$ (et similairement pour Y_p).

Démonstration : Voir corollaire 6.4.3. ■

4.8 Bornes théoriques de $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y)$ et de $R(B_Q)$

Nous présentons dans cette section trois bornes de $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y)$ et de $R(B_Q)$ que l'on peut déduire des résultats de ce chapitre. Les deux premières bornes se placent dans

le cadre d'un apprentissage supervisé et découlent des deux approches que nous avons présentées pour borner la quantité $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y)$. La troisième borne se situe pour sa part dans le cadre d'un apprentissage semi-supervisé, où nous supposons posséder un très grand échantillon de données non étiquetées.

Nous présentons à la section 4.8.2 des bornes de $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y)$ et de $R(B_Q)$ que l'on peut déduire directement du théorème 4.7.1 à l'aide de l'égalité 4.4. Nous comparons de plus ces bornes avec d'autres bornes déduites de résultats précédents, ainsi qu'avec une borne obtenue dans le cadre d'un apprentissage semi-supervisé, c'est-à-dire, en utilisant, en plus de données étiquetées, des données non étiquetées.

En définissant les ensembles $\mathcal{R}^{\delta}_{Q,S}$, $\mathcal{E}^{\delta}_{Q,S}$, $\mathcal{D}^{\delta}_{Q,S}$ et $\mathcal{A}^{\delta}_{Q,S}$ comme suit

$$\begin{aligned}
\mathcal{R}^{\delta}_{Q,S} &\stackrel{\text{déf}}{=} \left\{ r \,:\, \mathrm{kl}(R_S(G_Q)\|r) \leq \frac{1}{m}\left[\mathrm{KL}(Q\|P) + \log\frac{(m+1)}{\delta}\right]\right\},\\
\mathcal{E}^{\delta}_{Q,S} &\stackrel{\text{déf}}{=} \left\{ e \,:\, \mathrm{kl}(\widehat{e_Q}\|e) \leq \frac{1}{m}\left[2{\cdot}\mathrm{KL}(Q\|P) + \log\frac{(m+1)}{\delta}\right]\right\},\\
\mathcal{D}^{\delta}_{Q,S} &\stackrel{\text{déf}}{=} \left\{ d \,:\, \mathrm{kl}(\widehat{d_Q}\|d) \leq \frac{1}{m}\left[2{\cdot}\mathrm{KL}(Q\|P) + \log\frac{(m+1)}{\delta}\right]\right\},\\
\mathcal{A}^{\delta}_{Q,S} &\stackrel{\text{déf}}{=} \left\{ (d,e) \,:\, \mathrm{kl}(\widehat{d_Q},\widehat{e_Q}\|d,e) \leq \frac{1}{m}\left[2{\cdot}\mathrm{KL}(Q\|P) + \log\frac{(m+1)(m+2)}{2\delta}\right]\right\},
\end{aligned}$$

les théorèmes 4.6.1 et 4.6.2 impliquent le corollaire suivant, fournissant des bornes de $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y)$ et de $R(B_Q)$.

Corollaire 4.8.1. *Soit $\mathcal{H}$ un ensemble de classificateurs binaires et P une distribution sur $\mathcal{H}$. Soit $\delta \in (0,1]$. Alors nous avons*

$$\Pr_{S\sim D^m}\left(\forall\, Q \text{ sur} \mathcal{H}: \quad \underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y) \;\leq\; \sup\mathcal{E}^{\delta/2}_{Q,S} - \left(\inf\mathcal{R}^{\delta/2}_{Q,S}\right)^2\right) \geq 1-\delta\,,$$

$$\Pr_{S\sim D^m}\left(\forall\, Q \text{ sur} \mathcal{H}: \quad R(B_Q) \;\leq\; \frac{\sup\mathcal{E}^{\delta/2}_{Q,S} - \left(\inf\mathcal{R}^{\delta/2}_{Q,S}\right)^2}{\sup\mathcal{E}^{\delta/2}_{Q,S} - \left(\inf\mathcal{R}^{\delta/2}_{Q,S}\right)^2 \;+\; \left(\frac{1}{2} - \sup\mathcal{R}^{\delta/2}_{Q,S}\right)^2}\right) \geq 1-\delta\,.$$

Avec l'égalité $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y) = e_Q - \left(e_Q - \frac{d_Q}{2}\right)^2$, le théorème 4.7.1 permet de borner $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y)$ à l'aide des valeurs empiriques de d_Q et e_Q. Le théorème 4.0.1 fournit alors une borne de $R(B_Q)$. Ces deux bornes sont présentées dans le corollaire suivant.

Corollaire 4.8.2. *Soit $\mathcal{H}$ un ensemble de classificateurs binaires et P une distribution*

à priori sur $\mathcal{H}$. Soit $\delta \in (0,1]$. Alors nous avons

$$\Pr_{S\sim D^m}\left(\forall\, Q \ sur \mathcal{H}: \quad \underset{(\mathbf{x},y)\sim D}{\mathbf{Var}}\, W_Q(\mathbf{x},y) \ \leq \ \sup_{(d,e)\in\mathcal{A}^{\delta}_{Q,S}}\left\{e-\left(e-\frac{d}{2}\right)^2\right\}\right) \geq 1-\delta\,,$$

$$\Pr_{S\sim D^m}\left(\forall\, Q \ sur \mathcal{H}: \quad R(B_Q) \leq \frac{\sup\limits_{(e,s)\in\mathcal{A}^{\delta/2}_{Q,S}}\left\{\left(e-\frac{d}{2}\right)^2\right\}}{\sup\limits_{(d,e)\in\mathcal{A}^{\delta/2}_{Q,S}}\left\{e-\left(e-\frac{d}{2}\right)^2\right\}+\left(\frac{1}{2}-\sup\mathcal{R}^{\delta/2}_{Q,S}\right)^2}\right) \geq 1-\delta\,.$$

Dans le contexte d'un apprentissage semi-supervisé, il est possible de borner le taux de désaccord avec une très grande précision. En effet, pour savoir si deux classificateurs sont en désaccord sur la classification d'une données $\mathbf{x}$, nous n'avons pas besoin de connaitre la classe de $\mathbf{x}$. Il suit que le taux de désaccord peut s'évaluer à l'aide d'un échantillon de données non étiquetées, ce qui présente un avantage considérable dans les situations où il est facile d'obtenir un vecteur d'observation, mais difficile d'obtenir la classe d'une donnée. Dans ces situations, nous ne pourrons disposer que d'un petit ensemble d'apprentissage de données étiquetées, cependant, comme nous pourrons disposer d'un grand échantillon de données non étiquetées, il sera tout de même possible d'évaluer avec précision le taux de désaccord et ainsi obtenir une borne beaucoup plus serrée des quantités qui nous intéressent.

Corollaire 4.8.3 (borne semi-supervisée)**.** *Soit $\mathcal{H}$ un ensemble de classificateurs binaires et P une distribution sur $\mathcal{H}$. Soit $\delta \in (0,1]$. Alors nous avons*

$$\Pr_{\substack{S\sim D^m\\ S'\sim D^{m'}_{n.ét.}}}\left(\begin{array}{l}\forall Q \ sur \ \mathcal{H} \ : \\ \underset{(\mathbf{x},y)\sim D}{\mathbf{Var}}\, W_Q(\mathbf{x},y) \ \leq \ \sup\mathcal{E}^{\delta}_{Q,S} \ - \ \left(\sup\mathcal{E}^{\delta}_{Q,S} \ + \ \frac{1}{2}\cdot\inf\mathcal{D}^{\delta}_{Q,S'}\right)^2\end{array}\right) \geq 1-\delta$$

$$\Pr_{\substack{S\sim D^m\\ S'\sim D^{m'}_{n.ét.}}}\left(\begin{array}{l}\forall Q \ sur \ \mathcal{H} \ : \\ R(B_Q) \ \leq \ \dfrac{\sup\mathcal{E}^{\delta/2}_{Q,S} \ - \ \left(\sup\mathcal{E}^{\delta/2}_{Q,S} \ + \ \frac{1}{2}\cdot\inf\mathcal{D}^{\delta/2}_{Q,S_{\mathcal{U}}}\right)^2}{\frac{1}{4} \ - \ \frac{1}{2}\cdot\sup\mathcal{D}^{\delta/2}_{Q,S_{\mathcal{U}}}}\end{array}\right) \geq 1-\delta\,.$$

4.8.1 Note sur l'écriture des bornes

Dans l'article original présentant les principaux résultats de ce chapitre, dont le corollaire 4.8.2 (voir (Lacasse *et al.*, 2007)), nous avons écrit les bornes théoriques en

fonction des paramètres e_Q et s_Q, alors que dans cette thèse, nous avons choisi d'écrire ces mêmes bornes en fonction des paramètres d_Q et e_Q. La justification de ce changement de notation est simplement que l'écriture des bornes est légèrement moins lourde avec cette notation, par exemple le dénominateur présent dans l'écriture de C_Q en fonction de d_Q et e_Q est donné par $\frac{1}{4} - \frac{d_Q}{2}$, alors qu'il correspond à $\frac{e_Q}{2} + \frac{d_Q}{2} - \frac{1}{4}$ en fonction de e_Q et s_Q. Il est important de comprendre cependant que cela n'affecte en rien la valeur des bornes obtenues. En effet, la borne sur C_Q sera la même si l'on écrit C_Q avec d_Q et e_Q ou bien avec d_Q et s_Q, ou encore, comme dans l'article de (Lacasse *et al.*, 2007), avec e_Q et s_Q. En effet, on remarque que

$$\mathrm{kl}(\widehat{e_Q}, \widehat{s_Q} \| e, s) = \mathrm{kl}(\widehat{d_Q}, \widehat{e_Q} \| d, e) = \mathrm{kl}(\widehat{d_Q}, \widehat{s_Q} \| d, s)$$

si $d + e + s = 1$ (à noter que nous avons nécessairement l'égalité $\widehat{d_Q} + \widehat{e_Q} + \widehat{s_Q} = 1$, par définition de ces quantités). Il suit que nous avons les implications suivantes :

$$(d, e) \in \mathcal{A}^{\delta}_{Q,S} \iff (d, s) \in \mathcal{A}^{\delta,ds}_{Q,S} \iff (e, s) \in \mathcal{A}^{\delta,es}_{Q,S},$$

où $\mathcal{A}^{\delta,ds}_{Q,S}$ et $\mathcal{A}^{\delta,es}_{Q,S}$ sont les équivalents de $\mathcal{A}^{\delta}_{Q,S}$ respectivement définis avec $\widehat{d_Q}$, $\widehat{s_Q}$ et avec $\widehat{e_Q}$, $\widehat{s_Q}$ au lieu de $\widehat{d_Q}$, $\widehat{e_Q}$.

4.8.2 Résultats empiriques

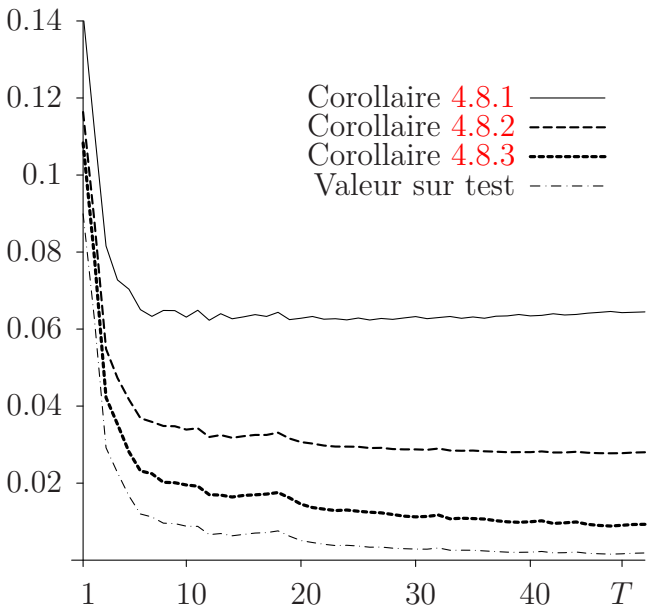

FIGURE 4.5 – Comparaison des différentes bornes de $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x}, y)$.

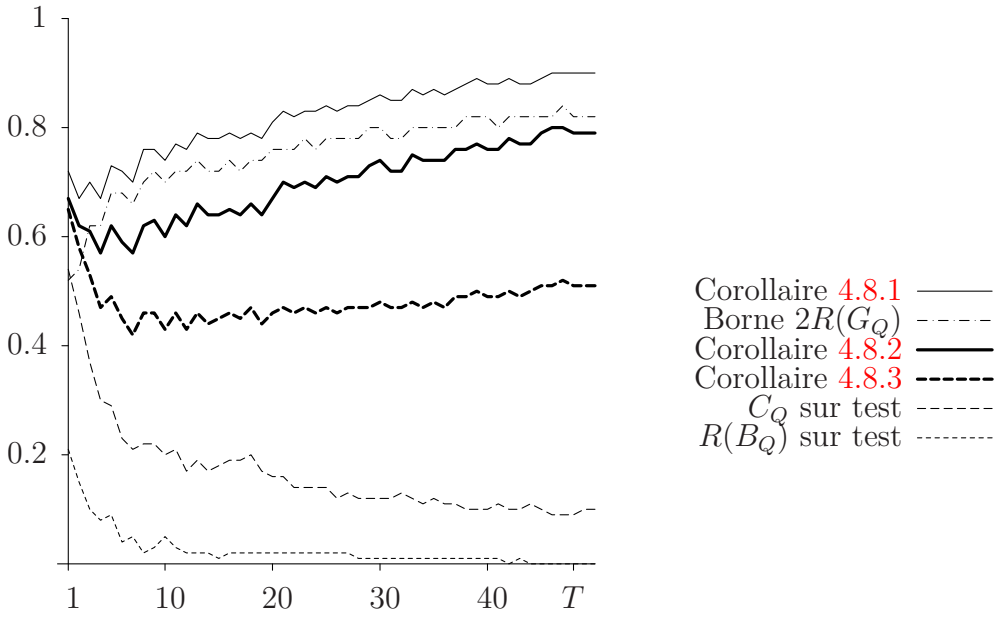

FIGURE 4.6 – Comparaison des différentes bornes de $R(B_Q)$.

Nous avons testé les trois bornes de $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x}, y)$ et de C_Q, ainsi que la borne $2R(B_Q)$ avec l'algorithme AdaBoost appliqué avec des souches de décision. Chaque souche de décision est un classificateur très simple qui ne regarde qu'un seul élément du vecteur de caractéristique pour effectuer une classification (voir la section 8.2 pour plus de détails). Il suit que ces classificateurs sont en moyenne très mauvais. Cependant, la combinaison de ces classificateurs en un vote de majorité bien pondéré peut donner des bons résultats. Par exemple, dans l'exemple que nous présentons ici, AdaBoost parvient à construire un vote de majorité ayant un risque nul à la fois sur l'ensemble d'entrainement et sur l'ensemble test.

La figure 4.5 illustre les résultats de nos expérimentations concernant les bornes de $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x}, y)$, et la figure 4.5 ceux concernant C_Q et $R(G_Q)$. Pour chacune de ces expérimentations, l'algorithme AdaBoost été exécuté avec l'ensemble *mushroom*, et à chacune des itérations (au total 50), les quantités statistiques nécessaires au calcul des bornes (d_Q, e_Q) ont été estimées sur l'ensemble d'entrainement. Les graphiques illutres donc l'évolution des bornes au fil des itérations.

4.9 Borner directement C_Q

Nous avons présenté, dans le corollaire 4.8.2, les bornes sur les deux quantités $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y)$ et $R(B_Q)$ parues dans (Lacasse *et al.*, 2007). La borne de $R(B_Q)$ s'obtient en bornant d'abord la variance (par le théorème 4.7.1) et le risque de Gibbs (par le théorème PAC-Bayes classique), puis en appliquant ces bornes dans le théorème 4.0.1.

En fait, il n'est pas nécessaire de procéder ainsi. Ce que le théorème 4.7.1 stipule, est qu'avec probabilité supérieure ou égale à $1-\delta$, les vraies valeurs d'un couple choisi parmi d_Q, e_Q et s_Q se trouveront à l'intérieur d'un certain domaine autour du couple formé par les valeurs empiriques. Par conséquent, nous pouvons borner par ce théorème n'importe quelle fonction de deux variables parmi d_Q, e_Q et s_Q. Par exemple, dans la section précédente, nous avons exprimé $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y)$ comme fonction de e_Q et s_Q, comme le théorème 4.7.1 précise qu'avec probabilité au moins $1-\delta$, les vraies valeurs de d_Q et e_Q se trouvent à l'intérieur du domaine que nous avons noté $\mathcal{A}^{\delta}_{Q,S}$, il suit que la quantité $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y)$ peut être bornée par la plus grande valeur que prend la fonction $e - \left(e - \frac{d}{2}\right)^2$ dans le domaine $\mathcal{A}^{\delta}_{Q,S}$.

Comme le risque de Gibbs peut également être exprimé comme une fonction de d_Q et e_Q (puisque la proposition 3.3.1 combinée à l'égalité 3.2 donne $R(G_Q) = e_Q - \frac{d_Q}{2}$), le théorème 4.7.1 permet également de borner celui-ci. Nous avons observé dans des tests empiriques que cette borne du risque de Gibbs peut même parfois être plus serrée que la borne PAC-Bayes classique. Dans ces cas, le gain sur la borne de C_Q est alors double, puisqu'en procédant ainsi, nous pouvons borner $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y)$ et $R(B_Q)$ avec une seule application du théorème 4.7.1. Il suit que des bornes de $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y)$ et $R(B_Q)$ chacune valide avec probabilité $1-\delta$ seront également simultanément valides avec cette même probabilité, alors que le corollaire 4.8.2 nécessite pour sa part des bornes de $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y)$ et $R(B_Q)$ chacune valide avec probabilité $1-\delta/2$ pour obtenir une borne de C_Q valide avec probabilité $1-\delta$.

Maintenant, si nous pouvons borner n'importe quelle fonction de d_Q et e_Q, il est alors possible de borner directement C_Q. Nous verrons que c'est en effet possible et que cela peut produire une grande amélioration sur la qualité de la borne. En effet, lorsque l'on borne C_Q en bornant séparément $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y)$ et $R(G_Q)$, nous utilisons deux points différents du domaine $\mathcal{A}^{\delta}_{Q,S}$, l'un représentant le pire cas de $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y)$ et l'autre le pire cas de $R(G_Q)$. En bornant directement C_Q, le maximum sera atteint en

un point qui devra faire un compromis entre la borne du risque de Gibbs et celle de la variance.

En procédant naïvement pour maximiser C_Q dans $\mathcal{A}^{\delta}_{Q,S}$, nous faisons cependant face à un problème : C_Q est un quotient de fonctions et peut diverger dans le domaine $\mathcal{A}^{\delta}_{Q,S}$. Cependant, nous savons que C_Q est une quantité inférieure ou égale à 1, les points pour lesquels C_Q diverge ne peuvent donc pas respecter toutes les contraintes du problème. En fait, il s'agit de points pour lesquels $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y) < 0$. Comme nous savons que la variance est une quantité positive, nous pouvons simplement soustraire ces points de $\mathcal{A}^{\delta}_{Q,S}$ pour obtenir un nouveau domaine.

On trouve :

$$\begin{aligned}
& \underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x},y) < 0 \\
\iff \quad & e_Q - \left(e_Q + \frac{d_Q}{2}\right)^2 < 0 \\
\iff \quad & e_Q < \left(e_Q + \frac{d_Q}{2}\right)^2 \\
\iff \quad & \sqrt{e_Q} < e_Q + \frac{d_Q}{2} \\
\iff \quad & d_Q > 2(\sqrt{e_Q} - e_Q)\,.
\end{aligned}$$

En notant

$$\widetilde{\mathcal{A}}^{\delta}_{Q,S} \stackrel{\text{déf}}{=} \mathcal{A}^{\delta}_{Q,S} \setminus \{(d,e) \in \mathcal{A}^{\delta}_{Q,S} \ : \ d > 2(\sqrt{e} - e)\},$$

on obtient alors le résultat suivant.

Théorème 4.9.1. *Soit $\mathcal{H}$ un ensemble de classificateurs binaires et P une distribution sur $\mathcal{H}$. Soit $\delta \in (0,1]$. Alors nous avons*

$$\Pr_{S\sim D^m}\left(\forall\, Q \ \textit{sur}\, \mathcal{H} : \quad R(B_Q) \leq \sup_{(d,e)\in\widetilde{\mathcal{A}}^{\delta}_{Q,S}} \frac{e - \left(e + \frac{d}{2}\right)^2}{\frac{1}{4} - \frac{d}{2}}\right) \geq 1 - \delta\,.$$

Démonstration : Avec probabilité $1-\delta$, les vraies valeurs de d_Q et e_Q se trouvent dans le domaine $\mathcal{A}^{\delta}_{Q,S}$, comme il est impossible que ces valeurs se trouvent dans l'ensemble $\{(d,e) \in \mathcal{A}^{\delta}_{Q,S} : d > 2(\sqrt{e} - e)\}$, il suit qu'avec probabilité $1-\delta$ elles se trouvent dans $\widetilde{\mathcal{A}}^{\delta}_{Q,S}$. Le résultat est alors une conséquence de l'inégalité $R(B_Q) \leq C_Q$ et de l'égalité

$$C_Q = \frac{e_Q - \left(e_Q + \frac{d_Q}{2}\right)^2}{\frac{1}{4} - \frac{d_Q}{2}}$$

qui découle du corollaire 3.3.3 et de la proposition 3.3.1. ■

4.9.1 Comparaison des bornes

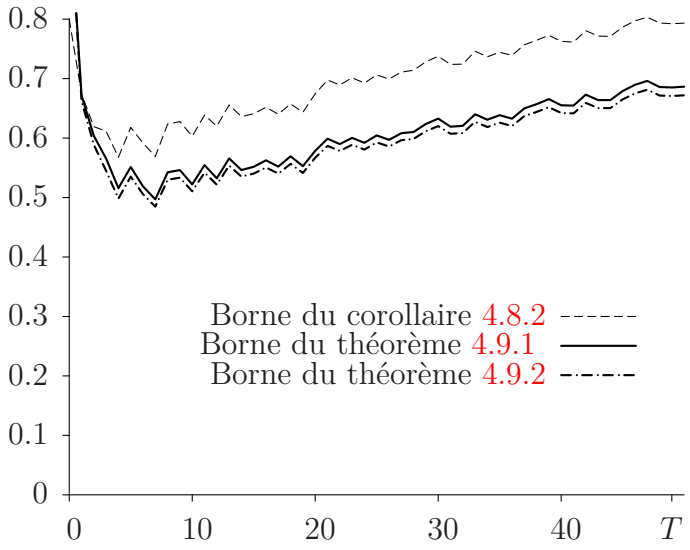

FIGURE 4.7 – Comparaison des différentes bornes de $R(B_Q)$.

La figure 4.7 compare la borne de $R(B_Q)$ obtenue avec le corollaire 4.8.2 et celle obtenue avec le théorème 4.9.1. Les calculs ont été fait en utilisant le même algorithme ainsi que le même ensemble d'apprentissage que dans la section 4.8.2. Nous remarquons que le fait de borner directement C_Q, à la manière du théorème 4.9.1, permet d'améliorer considérablement la qualité de la borne, dans cet exemple, la borne diminue de près de 10%. Cependant, elle reste moins précise que la borne obtenue dans un contexte d'apprentissage semi-supervisé avec un grand ensemble de données non étiquetées (borne du corollaire 4.8.3).

Comme l'illustre la figure, la borne de C_Q peut encore légèrement être améliorée. Une petite représentation graphique du domaine $\widetilde{\mathcal{A}}^{\delta}_{Q,S}$ nous suggérera d'elle-même le moyen d'améliorer la borne.

4.9.2 Représentation graphique de $\widetilde{\mathcal{A}}^{\delta}_{Q,S}$

La figure 4.8 donne une représentation graphique, avec e_Q en abscisse et d_Q en ordonnée, du domaine $\widetilde{\mathcal{A}}^{\delta}_{Q,S}$ ainsi que d'autres ensembles que nous avons étudiés. Dans le graphique, le domaine $\mathcal{A}^{\delta}_{Q,S}$ défini par le théorème 4.7.1 correspond à l'ovale centré sur le x blanc (qui correspond à une valeur empirique prise par le couple (e_Q, d_Q)). La partie grise de l'ovale correspond aux points de variance négative, la partie en bleue est donc $\widetilde{\mathcal{A}}^{\delta}_{Q,S}$.

Le graphique contient également quelques autres informations. La large bande diagonale correspond aux valeurs que peut prendre $R(G_Q)$ selon le théorème PAC-Bayes classique, alors que la large bande verticale qui traverse l'ovale correspond à l'ensemble $\mathcal{E}^{\delta}_{Q,S}$. Finalement, la mince bande horizontale qui traverse l'ovale correspond à l'ensemble $\mathcal{D}^{\delta}_{Q,S}$ (calculé à l'aide d'un très grand ensemble de données non étiquetées).

Le point en jaune à l'intérieur de $\mathcal{A}^{\delta}_{Q,S}$ et à l'extérieur de $\mathcal{E}^{\delta}_{Q,S}$ est le point maximisant C_Q dans $\widetilde{\mathcal{A}}^{\delta}_{Q,S}$. L'observation du fait que ce point est à l'extérieur de $\mathcal{E}^{\delta}_{Q,S}$ nous suggère qu'il est possible d'améliorer la borne en combinant le théorème 4.9.1 avec le théorème 4.6.1, ce qui donne lieu au théorème suivant.

Théorème 4.9.2. *Soit $\mathcal{H}$ un ensemble de classificateurs binaires et P une distribution sur $\mathcal{H}$. Soit $\delta \in (0, 1]$. Alors nous avons*

$$\Pr_{S\sim D^m}\left(\forall Q \text{ sur } \mathcal{H}: \quad R(B_Q) \leq \sup_{(d,e)\in \widehat{\mathcal{A}}^{\delta/2}_{Q,S}} \frac{e - \left(e + \frac{d}{2}\right)^2}{\frac{1}{4} - \frac{d}{2}}\right) \geq 1 - \delta$$

où

$$\widehat{\mathcal{A}}^{\delta/2}_{Q,S} = \widetilde{\mathcal{A}}^{\delta/2}_{Q,S} \setminus \{(d, e) : e > \overline{e}\} \quad \text{pour} \quad \overline{e} = \sup \mathcal{E}^{\delta/2}_{Q,S}.$$

Démonstration : Conséquence directe des théorèmes 4.9.1 et 4.6.1. ■

Le théorème 4.9.2 perd l'avantage que possède le théorème 4.9.1 de borner directement C_Q, puisqu'il nécessite une borne supplémentaire de e_Q. Cependant, comme le montre le graphique de la figure 4.7, le gain obtenu en contraignant le maximum à être choisi dans un point de $\mathcal{E}^{\delta/2}_{Q,S}$ permet d'obtenir une borne de $R(B_Q)$ légèrement inférieure à celle du théorème 4.9.1.

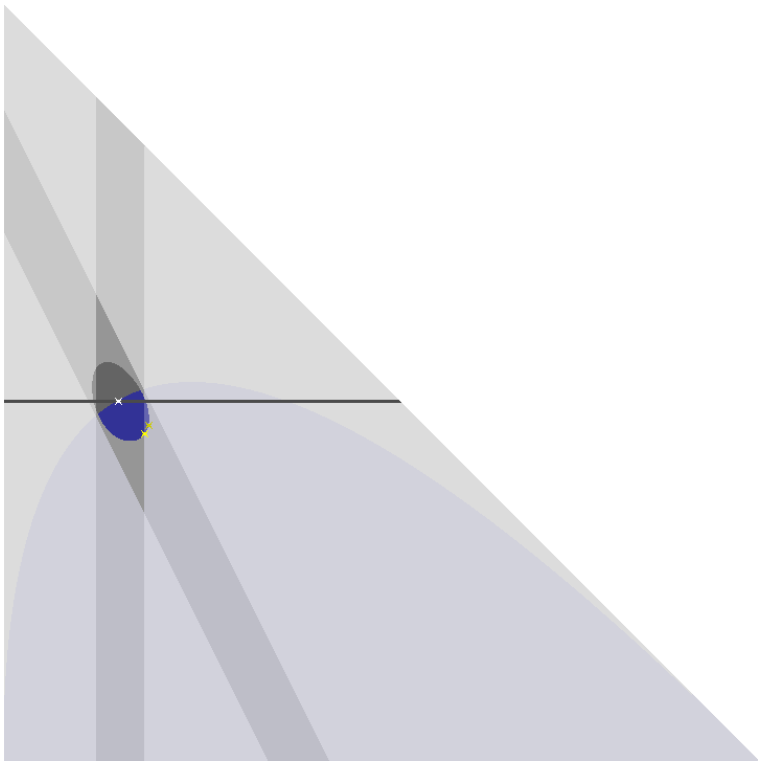

FIGURE 4.8 – Représentation graphique du domaine $\widetilde{\mathcal{A}}^{\delta}_{Q,S}$.

4.10 Calculabilité de C_Q

Considérons maintenant la quantité C_Q non comme une fonction de la distribution Q, mais simplement comme une fonction des quantités d_Q et e_Q, c'est-à-dire en posant

$$C_Q(d,e) \stackrel{\text{déf}}{=} \frac{e - \left(e - \frac{d}{2}\right)^2}{\frac{1}{4} - \frac{d}{2}}\,.$$

La proposition 4.10.1 montre que nous pouvons dans la pratique calculer efficacement la valeur de la borne du théorème 4.9.1 en démontrant que C_Q est une fonction concave. Pour obtenir la valeur de la borne, nous devons maximiser C_Q dans le domaine $\mathcal{A}^{\delta}_{Q,S}$, qui est convexe, ce problème revient à minimiser la fonction convexe $-C_Q$ dans un domaine borné et convexe de $\mathbf{R}^2$; diverses techniques d'optimisation convexe peuvent être utilisées pour résoudre ce problème.

Proposition 4.10.1 (Concavité de C_Q)**.** *La fonction $C_Q(d,e)$ est une fonction concave.*

Démonstration : Il suffit de montrer que la matrice hessienne associée à $C(d,e)$ est une

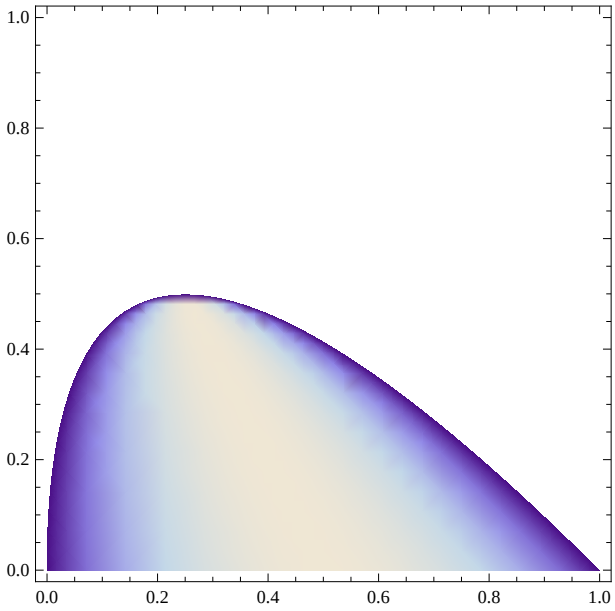

FIGURE 4.9 – Représentation de $C_Q(d,e)$ sous la forme d'un graphe de densité. La région pâle représente les points où la valeur de $C_Q(d,e)$ s'approche de 1, alors que le contour plus foncé correspond à des points où $C_Q(d,e)$ est presque nul.

matrice semi-définie négative, ce qui revient à montrer que

$$\frac{\partial^2 C_Q(d,e)}{\partial d^2} \leq 0\,; \quad \frac{\partial^2 C_Q(d,e)}{\partial e^2} \leq 0\,; \quad \frac{\partial^2 C_Q(d,e)}{\partial d^2}\frac{\partial^2 C_Q(d,e)}{\partial e^2} - \left(\frac{\partial^2 C_Q(d,e)}{\partial d \partial e}\right)^2 \geq 0\,.$$

On calcule

$$\begin{aligned}
\frac{\partial^2 C_Q(d,e)}{\partial d^2} &= \frac{2(1-4e)^2}{(2d-1)^3} \\
&\leq 0 \quad \forall e \in [0,1], d \in \left[0, \frac{1}{2}\right] \\
\frac{\partial^2 C_Q(d,e)}{\partial e^2} &= \frac{8}{2d-1} \\
&\leq 0 \quad \forall e \in [0,1], d \in \left[0, \frac{1}{2}\right] \\
\frac{\partial^2 C_Q(d,e)}{\partial d^2}\frac{\partial^2 C_Q(d,e)}{\partial e^2} - \left(\frac{\partial^2 C_Q(d,e)}{\partial d \partial e}\right)^2 &= \frac{2(1-4e)^2}{(2d-1)^3} \cdot \frac{8}{2d-1} - \left(\frac{4-16e}{(1-2d)^2}\right)^2 \\
&\equiv 0
\end{aligned}$$

■

4.11 Conclusion

Pour la plupart des algorithmes d'apprentissage construisant des classificateur par vote de majorité, la valeur de $R(G_Q)$ n'est pas un bon indicateur de la performance d'un classificateur retourné. Pour ces algorithmes, la borne $R(B_Q) \leq 2R(G_Q)$ n'est alors pas d'une grande utilité. Comme illustré à la figure 4.1, l'inégalité de Tchebychev conduit pour sa part à une borne de $R(B_Q)$ reflétant beaucoup mieux la performance du vote de majorité. Cette borne nécessite cependant la connaissance de la variance de W_Q pour être calculée. Comme cette quantité n'est pas connue, nous devons la borner. Pour ce faire, nous avons généralisé le théorème PAC-Bayes de sorte à pouvoir délimiter une région dans laquelle se trouve avec forte probabilité un couple (a_Q, b_Q) formé de deux valeurs prédéterminées parmi d_Q, e_Q et s_Q. Les égalités données en 4.4 permettent alors d'en conclure une borne de $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x}, y)$. De plus, comme d_Q peut se calculer à l'aide d'un ensemble de données non étiquetées, cette approche permet d'exploiter un contexte d'apprentissage semi-supervisé, permettant ainsi d'améliorer considérablement la borne. Finalement, dans un contexte d'apprentissage supervisé, nous constatons qu'il n'est pas nécessaire de borner individuellement $R(G_Q)$ et $\underset{(\mathbf{x},y)\sim D}{\mathbf{Var}} W_Q(\mathbf{x}, y)$ pour obtenir une borne de $R(B_Q)$. En effet, le théorème 4.7.1 permet de borner directement C_Q, ce qui permet d'obtenir une borne sensiblement plus serrée de $R(B_Q)$.

Chapitre 5

Fonctions de perte générales

Dans ce chapitre nous présentons un théorème permettant d'obtenir une borne sur des risques plus représentatifs du risque du vote de majorité que ne l'est le risque de Gibbs. Ce théorème permet de borner tous risques correspondant à l'espérance d'une fonction de perte pouvant s'exprimer comme une fonction de W_Q possédant un développement en série de Taylor défini dans l'intervalle $[0, 1]$.

5.1 Borner le risque associé à une fonction de perte générale

À la définition 2.1.1, nous avons défini le risque d'un classificateur comme étant l'espérance de la fonction de perte attribuant une perte de 1 au couple $(\mathbf{x}, y)$ si $h(\mathbf{x}) \neq y$ et 0 sinon, c'est-à-dire comme étant l'espérance de la fonction de perte, dite perte zéro-un, donnée par

$$\ell(\mathbf{x}, y) = I(h(\mathbf{x}) \neq y)\,.$$

Nous avons mentionné qu'il était possible, et parfois utile, de définir le risque à partir d'une fonction de perte de la forme $\ell(h, \mathbf{x}, y) : \mathcal{H} \times \mathcal{X} \times \{-1, 1\} \to \{0, 1\}$ autre que la fonction de perte zéro-un usuelle. Une telle fonction de perte intervient d'ailleurs dans la démonstration du théorème 4.6.1. Rien ne nous empêche de considérer également des risques correspondant à l'espérance de fonctions de perte plus complexes. Par exemple, à la définition 2.3.1, nous avons défini le risque de Gibbs comme étant le risque du classificateur de Gibbs, qui lui se trouve être une version stochastique du classificateur par vote de majorité, et par la suite (dans le texte suivant la définition 3.1.1), nous

avons montré l'égalité

$$R(G_Q) = \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} W_Q(\mathbf{x}, y)\,.$$

Nous pouvons donc interpréter le risque de Gibbs non pas comme le risque d'un certain classificateur stochastique apparenté au classificateur par vote de majorité, mais plutôt comme le risque du vote de majorité induit par la fonction de perte donnée par $W_Q(\mathbf{x}, y)$.

L'objectif que nous désirons atteindre est d'obtenir une borne de type PAC-Bayes du risque d'un classificateur par vote de majorité — on parle ici du risque induit par la fonction de perte $I(h(\mathbf{x}) \neq y)$. Comme cela s'avère difficile, nous proposons de borner le risque associé à des fonctions de perte différentes de la perte zéro-un (pouvant même possiblement être à valeurs dans tout $\mathbf{R}$), mais relié à celle-ci de sorte à pouvoir obtenir indirectement une borne du risque standard. Il est à remarquer que c'est en réalité précisément ce que fait le théorème PAC-Bayes classique. En effet, celui-ci permet d'obtenir une borne du risque de Gibbs, et ce dernier est induit par la perte $W_Q(\mathbf{x}, y)$, que l'on peut voir comme une approximation de la fonction de perte zéro-un (voir figure 5.1). Une borne du risque du vote de majorité s'obtient alors grâce à l'inégalité $R(B_Q) \leq 2R(G_Q)$. Nous avons appliqué dans le chapitre 3 (suite à l'inégalité 3.4) l'inégalité de Markov pour déduire cette relation entre ces deux risques, il est cependant possible de faire bien plus simple en constatant qu'elle est en fait une conséquence directe de l'inégalité $I(B_Q(\mathbf{x}) \neq y) \leq 2W_Q(\mathbf{x}, y)$:

$$R(B_Q) = \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} I(B_Q(\mathbf{x}) \neq y) \leq 2 \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} W_Q(\mathbf{x}, y) = 2R(G_Q)\,.$$

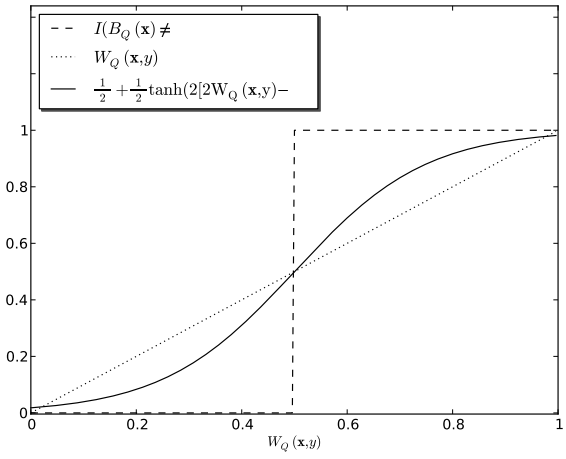

FIGURE 5.1 – Comparaison entre les fonctions de perte zéro-un, linéaire et tanh.

En résumé, le processus pour borner le risque du vote de majorité à l'aide du théorème PAC-Bayes classique se décrit comme suit : le théorème PAC-Bayes fournit une

borne du risque induit par la fonction de perte $W_Q(\mathbf{x}, y)$, soit le risque de Gibbs, et l'inégalité $I(B_Q(\mathbf{x}) \neq y) \leq 2W_Q(\mathbf{x}, y)$ permet de transformer cette borne en une borne du risque du vote de majorité.

Bien sûr, la fonction de perte $W_Q(\mathbf{x}, y)$ est une bien piètre approximation de la fonction de perte zéro-un, il n'est donc pas surprenant qu'elle mène à une borne du risque du vote de majorité qui soit souvent peu représentative de sa vraie valeur. Cette remarque mène à l'étude de fonctions de perte approchant davantage la fonction de perte zéro-un, un premier exemple étant la fonction $\frac{1}{2} + \frac{1}{2}\tanh(2[2W_Q(\mathbf{x}, y) - 1])$ illustrée à la figure 5.1.

5.1.1 Fonction de perte valant $\frac{1}{2}$ lorsque $W_Q(\mathbf{x}, y) = \frac{1}{2}$

Nous considérons des fonctions de perte pouvant s'écrire comme une série de Taylor centrée en $W_Q(\mathbf{x}, y) = \frac{1}{2}$, c'est-à-dire des fonctions de la forme

$$\begin{aligned} \zeta_Q(\mathbf{x}, y) &\stackrel{\text{déf}}{=} \frac{1}{2} + \frac{1}{2}\sum_{k=1}^{\infty} g(k)\,(2W_Q(\mathbf{x}, y) - 1)^k \qquad (5.1) \\ &= \frac{1}{2} + \frac{1}{2}\sum_{k=1}^{\infty} g(k)\left(\mathop{\mathbf{E}}_{h\sim Q} 2I(h(\mathbf{x}) \neq y) - 1\right)^k \\ &= \frac{1}{2} + \frac{1}{2}\sum_{k=1}^{\infty} g(k)\left(\mathop{\mathbf{E}}_{h\sim Q} -yh(\mathbf{x})\right)^k . \qquad (5.2) \end{aligned}$$

Le théorème suivant permet de borner le risque ζ_Q associé à la fonction de perte $\zeta_Q(\mathbf{x}, y)$ en fonction, entre autres, du risque empirique $\widehat{\zeta_Q}$ calculé à l'aide d'un ensemble d'exemples $S = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\}$ donné, on a donc

$$\zeta_Q \stackrel{\text{déf}}{=} \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} \zeta_Q(\mathbf{x}, y) \quad \text{et} \quad \widehat{\zeta_Q} \stackrel{\text{déf}}{=} \frac{1}{m}\sum_{i=1}^{m} \zeta_Q(\mathbf{x}_i, y_i)\ . \qquad (5.3)$$

Théorème 5.1.1. *Soit $\zeta_Q(\mathbf{x}, y)$ une fonction de perte de la forme de l'équation 5.1. Soit ζ_Q et $\widehat{\zeta_Q}$ respectivement le risque associé à $\zeta_Q(\mathbf{x}, y)$ et son estimé empirique sur un échantillon de m exemples. Alors, pour tout ensemble $\mathcal{H}$ de classificateurs binaires, pour toute distribution à priori P sur $\mathcal{H}$ et pour tout $\delta \in (0, 1]$, nous avons*

$$\Pr_{S\sim D^m}\left(\forall Q \text{ sur } \mathcal{H} : \mathrm{kl}\left(\frac{1}{c_a}\left[\widehat{\zeta_Q} - \frac{1}{2}\right] + \frac{1}{2}\middle\|\, \frac{1}{c_a}\left[\zeta_Q - \frac{1}{2}\right] + \frac{1}{2}\right) \leq \frac{1}{m}\left[k_a \cdot \mathrm{KL}(Q\|P) + \log\frac{m+1}{\delta}\right]\right) \geq 1 - \delta\,,$$

où

$$c_a \stackrel{\text{déf}}{=} \sum_{k=1}^{\infty} |g(k)| \quad et \quad k_a \stackrel{\text{déf}}{=} \frac{1}{c_a} \sum_{k=1}^{\infty} k \cdot |g(k)| \,.$$

Démonstration :

Nous obtenons une borne sur ζ_Q en mettant en relation le risque de ce classificateur avec le risque d'un classificateur de Gibbs particulier, que nous notons $G_{\overline{Q}}$, et qui est défini sur l'espace $\mathcal{H}^*$ de classificateurs où

$$\mathcal{H}^* = \cup_{k \in \mathbb{N}} \mathcal{H}^k.$$

Ainsi, la borne sur ζ_Q découlera de la borne obtenue en appliquant le théorème PAC-Bayes classique (théorème 2.3.3) avec le classificateur $G_{\overline{Q}}$.

Pour classifier un exemple $\mathbf{x}$, le classificateur $G_{\overline{Q}}$ pige d'abord un nombre $k \in \mathbb{N}$ suivant la distribution de probabilité $\Pr(k) = |g(k)|/c_a$, puis pige indépendamment k classificateurs $h_1, h_2, \ldots, h_k$ suivant la distribution Q ; la classe attribuée à $\mathbf{x}$ par $G_{\overline{Q}}$ est la classe donnée par le produit de classificateurs $h_1(\mathbf{x})h_2(\mathbf{x}) \cdots h_k(\mathbf{x})$. Le produit $h_1(\mathbf{x})h_2(\mathbf{x}) \cdots h_k(\mathbf{x})$ représente un classificateur binaire que nous notons $h_{1-k}(\mathbf{x})$, et nous définissons le risque de ce classificateur en utilisant la fonction de perte zéro-un donnée par

$$\ell(h_{1-k}, \mathbf{x}, y) = I\left((-y)^k h_{1-k}(\mathbf{x}) = \text{sgn}(g(k))\right) \,,$$

c'est-à-dire

$$\begin{aligned} R(h_{1-k}) &= \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} I\left((-y)^k h_{1-k}(\mathbf{x}) = \text{sgn}(g(k))\right) \\ &= \frac{1}{2} + \frac{1}{2} \cdot \text{sgn}(g(k)) \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} (-y)^k h_{1-k}(\mathbf{x}) \,. \end{aligned}$$

Alors

$$\begin{aligned}
R(G_{\overline{Q}}) &= \frac{1}{c_a}\sum_{k=1}^{\infty}|g(k)|\mathop{\mathbf{E}}_{h_{1-k}\sim Q^k} R(h_{1-k}) \\
&= \frac{1}{2}+\frac{1}{c_a}\sum_{k=1}^{\infty}|g(k)|\mathop{\mathbf{E}}_{h_{1-k}\sim Q^k}\left(R(h_{1-k})-\frac{1}{2}\right) \\
&= \frac{1}{2}+\frac{1}{c_a}\sum_{k=1}^{\infty}|g(k)|\cdot\frac{1}{2}\mathrm{sgn}(g(k))\mathop{\mathbf{E}}_{h_{1-k}\sim Q^k}\left(\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D}(-y)^k h_{1-k}(\mathbf{x})\right) \\
&= \frac{1}{2}+\frac{1}{c_a}\cdot\frac{1}{2}\sum_{k=1}^{\infty}|g(k)|\mathrm{sgn}(g(k))\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D}\mathop{\mathbf{E}}_{h_{1-k}\sim Q^k}\left((-y)^k h_{1-k}(\mathbf{x})\right) \\
&= \frac{1}{2}+\frac{1}{c_a}\cdot\frac{1}{2}\sum_{k=1}^{\infty}g(k)\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D}\left(\mathop{\mathbf{E}}_{h\sim Q}-yh(\mathbf{x})\right)^k \\
&= \frac{1}{2}+\frac{1}{c_a}\cdot\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D}\frac{1}{2}\sum_{k=1}^{\infty}g(k)\left(\mathop{\mathbf{E}}_{h\sim Q}-yh(\mathbf{x})\right)^k \\
&= \frac{1}{2}+\frac{1}{c_a}\cdot\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D}\left(\zeta_Q(\mathbf{x},y)-\frac{1}{2}\right) \\
&= \frac{1}{2}+\frac{1}{c_a}\left(\zeta_Q-\frac{1}{2}\right), \qquad (5.4)
\end{aligned}$$

où l'avant dernière égalité découle de l'égalité 5.2. Nous avons à présent fait un lien direct entre le risque ζ_Q et le risque d'un classificateur de Gibbs défini sur $\mathcal{H}^*$. Pour appliquer le théorème PAC-Bayes (théorème 2.3.3) et obtenir une borne de $R(G_{\overline{Q}})$, il nous reste maintenant à calculer la quantité $\mathrm{KL}(\overline{Q}\|\overline{P})$ où $\overline{Q}$ est la distribution sur $\mathcal{H}^*$ définie précédemment et $\overline{P}$ est une distribution à priori sur $\mathcal{H}^*$ que l'on définit à partir de la distribution P. Pour simplifier les calculs et restreindre la taille de la quantité $\mathrm{KL}(\overline{Q}\|\overline{P})$ (qu'il est préférable de garder petite), nous définissons $\overline{P}$ de la même façon que nous avons défini $\overline{Q}$. Tout comme pour $\overline{Q}$, une pige suivant $\overline{P}$ se fait en deux temps : dans un premier temps, un nombre $k \in \mathbb{N}$ est pigé suivant la distribution de probabilité $\mathrm{Pr(k)} = \mathrm{|g(k)|/c_a}$, puis k classificateurs sont pigés indépendamment suivant la distribution P. Avec cette définition de $\overline{P}$, il est aisé de calculer $\mathrm{KL}(\overline{Q}\|\overline{P})$:

$$\begin{aligned}
\mathrm{KL}(\overline{Q}\|\overline{P}) &= \frac{1}{c_a}\sum_{k=1}^{\infty}|g(k)|\ \cdot\mathop{\mathbf{E}}_{h_1\sim Q}\cdots\mathop{\mathbf{E}}_{h_k\sim Q}\log\frac{|g(k)|\prod_{i=1}^{k}Q(h_i)}{|g(k)|\prod_{i=1}^{k}P(h_i)} \\
&= \frac{1}{c_a}\sum_{k=1}^{\infty}|g(k)|\ \cdot\mathop{\mathbf{E}}_{h_1\sim Q}\cdots\mathop{\mathbf{E}}_{h_k\sim Q}\sum_{i=1}^{k}\log\frac{Q(h_i)}{P(h_i)} \\
&= \frac{1}{c_a}\sum_{k=1}^{\infty}|g(k)|\ k\cdot\mathop{\mathbf{E}}_{h\sim Q}\log\frac{Q(h)}{P(h)} \\
&= k_a\cdot\mathrm{KL}(Q\|P),
\end{aligned}$$

où

$$k_a = \frac{1}{c_a}\sum_{k=1}^{\infty}k\cdot|g(k)|.$$

En appliquant maintenant le théorème PAC-Bayes pour borner $R(G_{\overline{Q}})$, nous obtenons

$$\Pr_{S\sim D^m}\left(\forall Q \text{ sur } \mathcal{H} \,:\, \mathrm{kl}\Big(R_S(G_{\overline{Q}})\|R(G_{\overline{Q}})\Big) \leq \frac{1}{m}\left[k_a\cdot \mathrm{KL}(Q\|P)+\log\frac{m+1}{\delta}\right]\right) \;\geq\; 1-\delta\,,$$

et en remplaçant $R(G_{\overline{Q}})$ et $R_S(G_{\overline{Q}})$ respectivement par ζ_Q et $\widehat{\zeta_Q}$ (voir l'équation 5.4), nous obtenons le résultat voulu. ■

Le théorème 5.1.1 donne une borne de la quantité $\frac{1}{c_a}\left[\zeta_Q-\frac{1}{2}\right]+\frac{1}{2}$, ce qui mène à une borne du risque ζ_Q, puisque si $\frac{1}{c_a}\left[\zeta_Q-\frac{1}{2}\right]+\frac{1}{2}\leq M$, alors

$$\zeta_Q \leq c_a\left(M-\frac{1}{2}\right)+\frac{1}{2}\,. \tag{5.5}$$

Nous voyons de ce fait que toute perte de précision dans la borne de $\frac{1}{c_a}\left[\zeta_Q-\frac{1}{2}\right]+\frac{1}{2}$ sera multipliée par le facteur c_a dans la borne de ζ_Q. Le théorème ne pourra alors fournir une borne serrée de ζ_Q que si le facteur c_a est petit, et donc que si les coefficients de la série de Taylor 5.1 ne sont pas trop grands.

5.1.2 Fonction de perte plus générale

Le théorème 5.1.1, que nous avons écrit sous sa forme originale (voir Germain *et al.*, 2007), présente une borne de type PAC-Bayes valide pour toute fonction de perte s'écrivant comme une série de Taylor de $W_Q(\mathbf{x},y)$ définie dans l'intervalle $[0,1]$, centrée en $\frac{1}{2}$ et valant $\frac{1}{2}$ pour $W_Q(\mathbf{x},y)=\frac{1}{2}$. Cette dernière contrainte du théorème peut être un peu restrictive, par exemple, il est parfois plus naturel d'exprimer une fonction de perte sous la forme d'une fonction passant par le point 1 pour les exemples $(\mathbf{x},y)$ tels que $W_Q(\mathbf{x},y)=\frac{1}{2}$ (c'est ce que nous ferons par exemple plus loin pour la perte quadratique).

Nous présentons dans cette sous-section, une version du théorème sur les fonctions de perte générales applicable à des fonctions de perte non contraintes à valoir $\frac{1}{2}$ pour les exemples $(\mathbf{x},y)$ tels que $W_Q(\mathbf{x},y)=\frac{1}{2}$, c'est-à-dire, des fonctions de perte de la forme

$$\zeta_Q(\mathbf{x},y) \;=\; g(0)+\sum_{k=1}^{\infty} g(k)\left(2W_Q(\mathbf{x},y)-1\right)^k\,, \tag{5.6}$$

avec $g(k)\in\mathbf{R}$ pour tout $k=0,1,2,\ldots$

Théorème 5.1.2. *Soit $\zeta_Q(\mathbf{x}, y)$ une fonction de perte de la forme de l'équation 5.6. Soit ζ_Q et $\widehat{\zeta_Q}$ respectivement le risque associé à $\zeta_Q(\mathbf{x}, y)$ et son estimé empirique sur un échantillon de m exemples. Alors, pour tout ensemble $\mathcal{H}$ de classificateurs binaires, pour toute distribution à priori P sur $\mathcal{H}$ et pour tout $\delta \in (0, 1]$, nous avons*

$$\Pr_{S\sim D^m}\left(\forall Q \text{ sur } \mathcal{H} : \mathrm{kl}\left(\frac{1}{2c_a}\left[\widehat{\zeta_Q} - g(0)\right] + \frac{1}{2} \middle\| \frac{1}{2c_a}\left[\zeta_Q - g(0)\right] + \frac{1}{2}\right) \leq \frac{1}{m}\left[k_a \cdot \mathrm{KL}(Q\|P) + \log\frac{m+1}{\delta}\right]\right) \geq 1 - \delta\,,$$

où

$$c_a \overset{\text{déf}}{=} \sum_{k=1}^{\infty} |g(k)| \quad \textit{et} \quad k_a \overset{\text{déf}}{=} \frac{1}{c_a}\sum_{k=1}^{\infty} k \cdot |g(k)|\,.$$

Démonstration : Il suffit simplement d'appliquer le théorème 5.1.1 avec la fonction de perte $\zeta'_Q(\mathbf{x}, y) \overset{\text{déf}}{=} \dfrac{\zeta_Q(\mathbf{x}, y) - g(0) + 1}{2}$, qui a la forme de l'équation 5.1. ■

Borner le risque de Gibbs classique

Nous savons que $2 \cdot R(G_Q)$ constitue une borne supérieure du risque du vote de majorité, $R(B_Q)$, et comme le théorème PAC-Bayes classique permet d'obtenir une borne, notons-la $\overline{R}$, du risque de Gibbs, il est possible de déduire la borne $2 \cdot \overline{R}$ pour le risque du vote de majorité. Nous pouvons appliquer le théorème 5.1.1 pour borner directement la quantité $R(G_Q)$; cela mène précisément à la même borne de $R(B_Q)$.

En effet, nous avons

$$2 \cdot R_{(\mathbf{x},y)}(G_Q) = 2 \cdot W_Q(\mathbf{x}, y) = 1 + (2W_Q(\mathbf{x}, y) - 1),$$

ainsi le théorème 5.1.2 s'applique avec la fonction de perte $\zeta_Q = 2R(G_Q)$, nous avons alors pour les constantes c_a et k_a les valeurs $c_a = 1$ et $k_a = 1$. Le risque du classificateur de Gibbs intermédiaire intervenant dans la preuve du théorème 5.1.2, qui est donné par $R(G_{\overline{Q}}) = \frac{1}{2c_a}(\zeta_Q - 1) + \frac{1}{2} = \frac{1}{2}(2R(G_Q) - 1) + \frac{1}{2}$ se trouve donc être égal à $R(G_Q)$. Ainsi, comme $k_a = 1$, en appliquant le théorème 5.1.2 pour borner la quantité $2R(G_Q)$ nous nous trouvons à appliquer le théorème PAC-Bayes classique pour borner $R(G_Q)$ et à multiplier ensuite cette borne par 2.

5.2 Apprentissage semi-supervisé

Le théorème sur les fonctions de perte générales permet de borner le risque associé à des fonctions de perte de la forme

$$\zeta_Q(\mathbf{x}, y) \stackrel{\text{déf}}{=} g(0) + \sum_{k=1}^{\infty} g(k) \left(2W_Q(\mathbf{x}, y) - 1\right)^k = 1 + \sum_{k=1}^{\infty} g(k) \left(\mathop{\mathbf{E}}_{h \sim Q} -yh(\mathbf{x})\right)^k .$$

L'on remarque que si $g(k) = 0$ pour toute valeur de k impaire, c'est-à-dire, si ζ_Q peut s'écrire sous la forme

$$\zeta_Q(\mathbf{x}, y) = g(0) + \sum_{k=1}^{\infty} g(2k) \left(2W_Q(\mathbf{x}, y) - 1\right)^{2k} = g(0) + \sum_{k=1}^{\infty} g(2k) \left(\mathop{\mathbf{E}}_{h \sim Q} -h(\mathbf{x})\right)^{2k} ,$$

nous n'avons plus besoin de connaitre la classe y d'une donnée pour évaluer sa perte. C'est-à-dire que dans ce cas, l'évaluation du risque empirique $\widehat{\zeta}_Q$ peut s'effectuer avec un ensemble de données non étiquetées.

Dans un contexte d'apprentissage semi-supervisé, lorsque nous utilisons pour l'apprentissage en plus d'un ensemble de données étiquetées, un ensemble de données non étiquetées, il est possible de décomposer la fonction de perte ζ_Q en deux fonctions distinctes : l'une formée des termes impaires de la série de Taylor (que l'on peut évaluer seulement pour les exemples étiquetés) et l'autre formée des termes pairs (que l'on peut évaluer aussi bien pour les exemples étiquetés que pour les exemples non étiquetés). En somme, nous aurons

$$\zeta_Q(\mathbf{x}, y) = \zeta_Q^{impair}(\mathbf{x}, y) + \zeta_Q^{pair}(\mathbf{x}) ,$$

où

$$\zeta_Q^{impair}(\mathbf{x}, y) \stackrel{\text{déf}}{=} \sum_{k=1}^{\infty} g(2k-1) \left(2W_Q(\mathbf{x}, y) - 1\right)^{2k-1}$$

et

$$\zeta_Q^{pair}(\mathbf{x}, y) \stackrel{\text{déf}}{=} g(0) + \sum_{k=1}^{\infty} g(2k) \left(2W_Q(\mathbf{x}, y) - 1\right)^{2k} = g(0) + \sum_{k=1}^{\infty} g(2k) \left(\mathop{\mathbf{E}}_{h \sim Q} -h(\mathbf{x})\right)^{2k} .$$

Pour obtenir une borne du risque ζ_Q, il est alors possible d'additionner une borne du risque ζ_Q^{impair} (calculée en utilisant des données étiquetées seulement) et une borne du risque ζ_Q^{pair} (calculée en utilisant toutes les données disponibles, étiquetées et non étiquetés). L'utilisation des données non étiquetées permettra alors d'obtenir une borne plus serrée du risque associé aux termes pairs du développement de ζ_Q, et ainsi, possiblement obtenir une borne du risque global plus serrée.

5.3 Borner $R(B_Q)$ avec le théorème 5.1.1

Le risque du classificateur B_Q est borné par l'espérance sur D de la fonction de perte du vote de majorité suivante

$$P_{VM}(\mathbf{x}, y) \stackrel{\text{déf}}{=} \begin{cases} 0 & \text{si } W_Q(\mathbf{x}, y) < \frac{1}{2} \\ 1 & \text{sinon}\,. \end{cases}$$

Une borne du risque du vote de majorité sera obtenue en appliquant le théorème 5.1.1 pour borner le risque associé à une fonction de perte de la forme de l'équation 5.1 et majorant P_{VM}. Par exemple, les fonctions de perte exponentielle (notée $\mathcal{E}_Q(\mathbf{x}, y)$) et de perte sigmoïdale (notée $\mathcal{T}_Q(\mathbf{x}, y)$) satisfont ces conditions. Ces fonctions de perte se définissent comme suit :

$$\mathcal{E}_Q(\mathbf{x}, y) \stackrel{\text{déf}}{=} \frac{1}{2} \exp\Big(\beta[2W_Q(\mathbf{x}, y) - 1]\Big)$$

et

$$\mathcal{T}_Q(\mathbf{x}, y) \stackrel{\text{déf}}{=} \frac{1}{2} + \frac{1}{2} \tanh\Big(\beta[2W_Q(\mathbf{x}, y) - 1]\Big)\,,$$

où $\beta \in (0, \infty)$ est un paramètre d'ajustement (voir la figure 5.2). Ces fonctions sont bien de la forme de l'équation 5.1, de plus, elles sont positives et pour tout $(\mathbf{x}, y)$ tels que $W_Q(\mathbf{x}, y) \geq \frac{1}{2}$ on a $\mathcal{E}_Q(\mathbf{x}, y) \geq \frac{1}{2}$ et $\mathcal{T}_Q(\mathbf{x}, y) \geq \frac{1}{2}$, et donc

$$P_{VM}(\mathbf{x}, y) \leq 2\mathcal{E}_Q(\mathbf{x}, y) \quad \text{et} \quad P_{VM}(\mathbf{x}, y) \leq 2\mathcal{T}_Q(\mathbf{x}, y)\,.$$

Pour les valeurs de c_a et k_a nécessaires à l'application du théorème 5.1.1, nous trouvons les valeurs suivantes :

	$\mathcal{E}_Q(\mathbf{x}, y)$	$\mathcal{T}_Q(\mathbf{x}, y)$
c_a	$e^{\beta} - 1$	$\tan(\beta)$
k_a	$\dfrac{\beta}{1 - e^{-\beta}}$	$\dfrac{1}{\cos(\beta)\sin(\beta)}$

Notons $\mathcal{E}_Q$ et $\mathcal{T}_Q$ respectivement les risques associés aux fonctions de perte exponentielle et sigmoïdale, c'est-à-dire

$$\mathcal{E}_Q \stackrel{\text{déf}}{=} \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} \mathcal{E}_Q(\mathbf{x}, y) \quad \text{et} \quad \mathcal{T}_Q \stackrel{\text{déf}}{=} \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} \mathcal{T}_Q(\mathbf{x}, y)\,.$$

Nous avons alors les bornes suivantes pour $R(B_Q)$:

$$R(B_Q) \leq 2\mathcal{E}_Q \quad \text{et} \quad R(B_Q) \leq 2\mathcal{T}_Q\,.$$

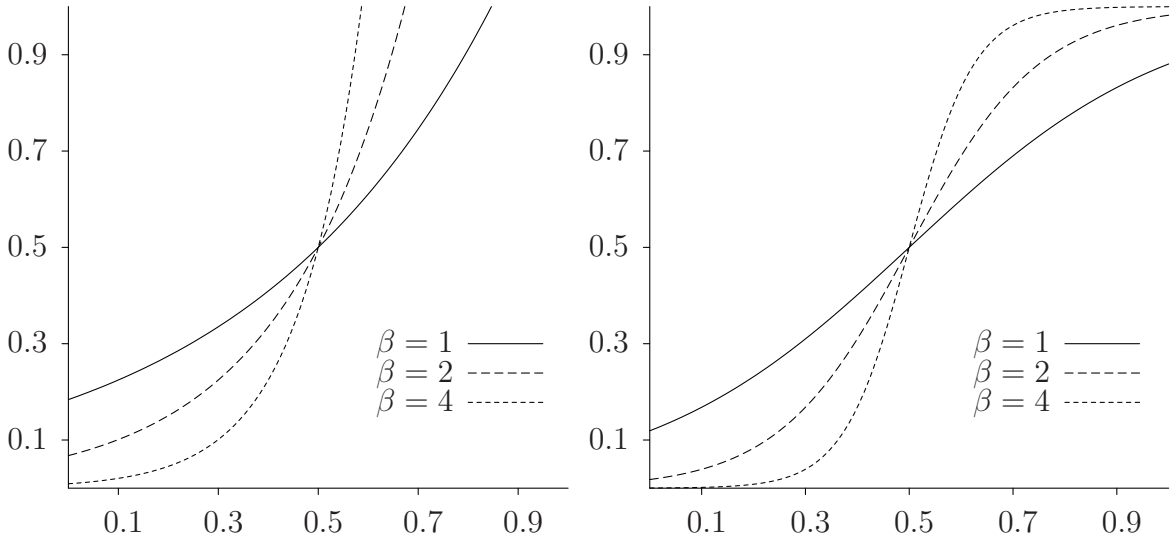

FIGURE 5.2 – Effet du paramètre β sur la fonction de perte exponentielle (à gauche) et sur la fonction de perte sigmoïdale (à droite).

La fonction de perte sigmoïdale semble particulièrement intéressante à étudier, car à la limite lorsque β tend vers l'infini, elle devient égale à la fonction P_{VM}. En effet,

$$\frac{1}{2} + \frac{1}{2} \lim_{\beta \to \infty} \tanh(\beta(2x-1)) = \begin{cases} 0 & \text{si } x < \frac{1}{2} \\ 1 & \text{si } x > \frac{1}{2} \end{cases}.$$

On obtient alors l'égalité

$$\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} I\left(W_Q(\mathbf{x},y) > \frac{1}{2}\right) \quad = \quad \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} \lim_{\beta\to\infty} \frac{1}{2} \tanh\left(\beta\left[2W_Q(\mathbf{x},y) - 1\right]\right) + \frac{1}{2}.$$

Malheureusement, comme le montre la figure 5.2, pour que la fonction de perte sigmoïdale approche convenablement la fonction de perte du vote de majorité (P_{VM}), il faut que le paramètre β soit grand, cependant, la série de Taylor centrée en 0 de la fonction $\tanh(x)$ converge seulement pour $x < \frac{\pi}{2}$. Par conséquent, puisque $W_Q(\mathbf{x}, y) \in [0, 1]$, la série de Taylor centrée en $\frac{1}{2}$ de la fonction $\tanh(\beta[2W_Q(\mathbf{x}, y) - 1])$ converge pour toute valeur de W_Q si et seulement si $\beta < \frac{\pi}{2} \approx 1.57$.

Il n'est donc pas possible d'obtenir, à partir de la fonction de perte sigmoïdale et du théorème 5.1.1, une borne intéressante de $R(B_Q)$ pour les algorithmes concentrant la masse de W_Q près de $\frac{1}{2}$ (comme AdaBoost), c'est-à-dire que cette approche ne donne pas une bonne borne pour les votes de majorité ayant un risque de Gibbs élevé. Par exemple, dans la situation où la masse de W_Q est entièrement centrée sur 0.4, en prenant $\beta = 1.57$, nous obtenons comme risque empirique sigmoïdal $\widehat{\zeta_Q} \approx 0.348$. Comme $R(B_Q) \leq 2\mathcal{T}_Q$, on obtient dans cet exemple une borne supérieure de $R(B_Q)$ dépassant 0.696.

5.3.1 La fonction de perte provenant de la fonction erf

La fonction de perte donnée par $\varphi(\mathbf{x}, y) = \Phi(\beta(2W_Q(\mathbf{x}, y) - 1))$, où $\Phi(a) = \Pr(X \leq a)$ pour $X \sim N(0, 1)$ (et donc $\Phi(a) = \frac{1}{2}(1 + \operatorname{erf}(a/\sqrt{2}))$), possède les mêmes propriétés que la fonction de perte sigmoïdale, c'est-à-dire qu'elle majore P_{VM} et qu'elle tend vers P_{VM} lorsque β tend vers l'infini. Elle a en plus l'avantage de posséder une série de Taylor qui converge pour toute valeur réelle. Nous pourrions donc utiliser cette fonction de perte pour borner le risque du vote de majorité. Cependant, un petit calcul nous permet de trouver comme valeur du coefficient c_a de cette fonction de perte la valeur suivante :

$$c_a = \frac{1}{\sqrt{2\pi}}\left[\beta + \frac{\beta^3}{2 \cdot 3} + \frac{\beta^5}{2 \cdot 3 \cdot 5} + \cdots\right].$$

Cette quantité grandit encore plus rapidement avec β que le coefficient c_a associé à la fonction de perte exponentielle, que nous voyons à la prochaine section, et également plus rapidement que celui associé à la fonction de perte sigmoïdale. Il suit que la dégradation de la borne que nous devons subir si nous voulons une valeur de β permettant d'approcher convenablement P_{VM} sera trop grande pour donner une borne intéressante.

5.4 Borner le risque exponentiel

Contrairement à celui de la fonction de perte sigmoïdale, le paramètre β de la fonction de perte exponentielle n'est aucunement contraint. Comme l'indique la figure 5.2, le fait d'augmenter β diminue $\mathcal{E}_Q(\mathbf{x}, y)$ à condition que $W_Q(\mathbf{x}, y)$ soit inférieur à $\frac{1}{2}$. Lorsque $W_Q(\mathbf{x}, y) > \frac{1}{2}$, $\mathcal{E}_Q(\mathbf{x}, y)$ tend vers l'infini lorsque β tend vers l'infini. Par conséquent, si le vote de majorité ne fait pas d'erreur, $\mathcal{E}_Q$ tend vers zéro lorsque β tend vers l'infini. Il y a donc espoir dans ces situations d'obtenir une borne serrée de $R(B_Q)$ par l'intermédiaire de la borne de $\mathcal{E}_Q$.

Pour la fonction de perte exponentielle, les valeurs c_a et k_a du théorème 5.1.1 se trouvent facilement. On obtient :

$$\begin{aligned} c_a &= \sum_{i=1}^{\infty} \frac{\beta^i}{i!} \\ &= e^{\beta} - 1 \end{aligned} \qquad \begin{aligned} k_a &= \frac{1}{c_a} \sum_{i=1}^{\infty} \frac{i\beta^i}{i!} \\ &= \frac{1}{e^{\beta} - 1} \cdot \beta \sum_{i=1}^{\infty} \frac{i\beta^{i-1}}{i!} = \frac{1}{e^{\beta} - 1} \cdot \beta \cdot e^{\beta} \\ &= \frac{\beta}{1 - e^{-\beta}}. \end{aligned}$$

5.4.1 Résultats d'expérimentation avec AdaBoost

Nous avons utilisé AdaBoost pour tester empiriquement la borne de $\mathcal{E}_Q$. Dans cette expérimentation, des souches de décision (arbres de décision à une couche) ont été utilisées comme classificateurs de base dans l'exécution de l'algorithme.

Lors de son exécution, AdaBoost construit un vote de majorité de la forme

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{\substack{i=1\ldots n\\ b=-1,1}} \alpha_b^i \cdot h_b^i(\mathbf{x})\right),$$

où n correspond au nombre total de valeurs prises par les différents attributs et où les α_b^i forment une distribution, c'est-à-dire qu'ils sont tous supérieurs ou égaux à zéro et qu'ils somment à 1. On peut alors noter Q la distribution sur $\mathcal{H}$ donnée par $Q(h_b^i) = \alpha_b^i$. Naturellement, on peut associer à ce vote de majorité le classificateur de Gibbs G_Q suivant : pour classifier un exemple $\mathbf{x}$, G_Q pige aléatoirement la souche de décision h_b^i suivant la distribution des α_b^i puis attribue à $\mathbf{x}$ la classe $h_b^i(\mathbf{x})$.

Ainsi, f correspond au classificateur par vote de majorité B_Q. Le théorème PAC-Bayes permet alors de borner le risque de f, et le théorème 5.1.1 permet de borner le risque exponentiel de f. Pour appliquer le théorème, il faut choisir une distribution à priori P sur l'ensemble $\mathcal{H}$ des souches de décision. Comme aucune souche de décision n'est à priori à privilégier, il est naturel de prendre la distribution uniforme sur $\mathcal{H}$, donc $P(h) = \frac{1}{2n}$ pour tout $h \in \mathcal{H}$. Nous pouvons maintenant calculer la divergence de Kullback-Leibler entre P et Q, ce qui donne

$$\mathrm{KL}(Q\|P) = \mathop{\mathbf{E}}_{h\sim Q} \log \frac{Q(h)}{P(h)} = \sum_{\substack{i=1\ldots n\\ b=-1,1}} \alpha_b^i \log(2n\alpha_b^i)\,.$$

La figure 5.3 montre les résultats obtenus avec deux ensembles de données distincts. Pour ces expérimentations, nous avons simplement utilisé $\beta = \log 2$, ce qui donne alors 1 pour valeur de c_a dans le calcul des bornes des risques exponentiels. On peut constater à la vue de cette figure que $\beta = \log 2$ est une valeur trop petite pour donner un risque exponentiel intéressant. Ainsi, la borne de $R(B_Q)$ que l'on obtient est approximativement $R(B_Q) \leq 0.9$ (après 200 itérations d'AdaBoost). Pour l'ensemble de données Sonar, le vote de majorité fait plusieurs erreurs de classification, le risque exponentiel ne diminuera donc pas en augmentant β. Par contre, passé 138 itérations d'AdaBoost, le vote de majorité ne fait aucune erreur avec l'ensemble de données Mushroom, c'est donc un ensemble idéal pour tenter d'obtenir une borne serrée du vote de majorité à

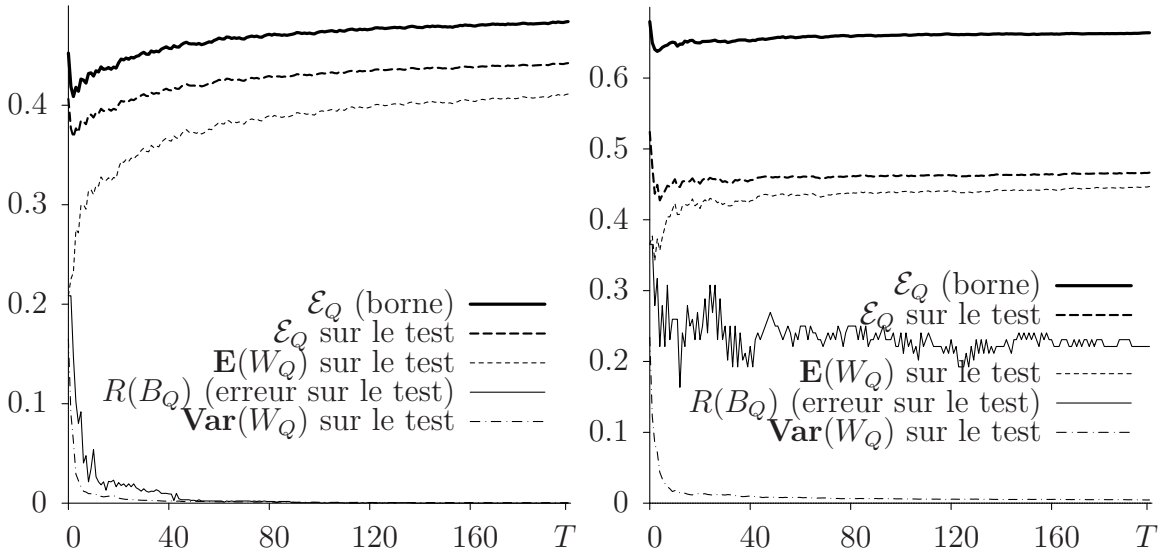

FIGURE 5.3 – Comportement de la borne du risque exponentiel ($\mathcal{E}_Q$ (borne)), du risque exponentiel empirique évalué sur l'ensemble test ($\mathcal{E}_Q$ sur le test), du risque de Gibbs ($\mathbf{E}(W_Q)$ sur le test), sa variance ($\mathbf{Var}(W_Q)$ sur le test), et l'erreur sur l'ensemble test du vote de majorité ($R(B_Q)$ (erreur sur le test)) en fonction des itérations d'AdaBoost, T, pour les ensembles de données Mushroom (à gauche) et Sonar (à droite). Les risques empiriques ainsi que les bornes ont été calculés avec $\beta = \log 2$.

partir du risque exponentiel, car celui-ci tend vers zéro en augmentant β (voir figure 5.2).

Pour la perte exponentielle, nous avons $c_a = e^{\beta} - 1$ pour l'application du théorème 5.1.1. Donc la valeur de c_a augmente exponentiellement vite avec β. Comme le montre la figure 5.4, il s'ensuit que la précision de la borne sur le risque exponentiel se dégrade très rapidement en augmentant légèrement β. En effet, nous voyons premièrement dans cette figure que, comme nous pouvions nous y attendre, la valeur du risque exponentiel calculée sur l'ensemble test diminue lorsque β augmente. Le risque exponentiel se comporte alors de plus en plus comme le risque du vote de majorité. Cependant, nous voyons sur la partie de droite de la figure que la borne sur le risque exponentiel augmente avec β.

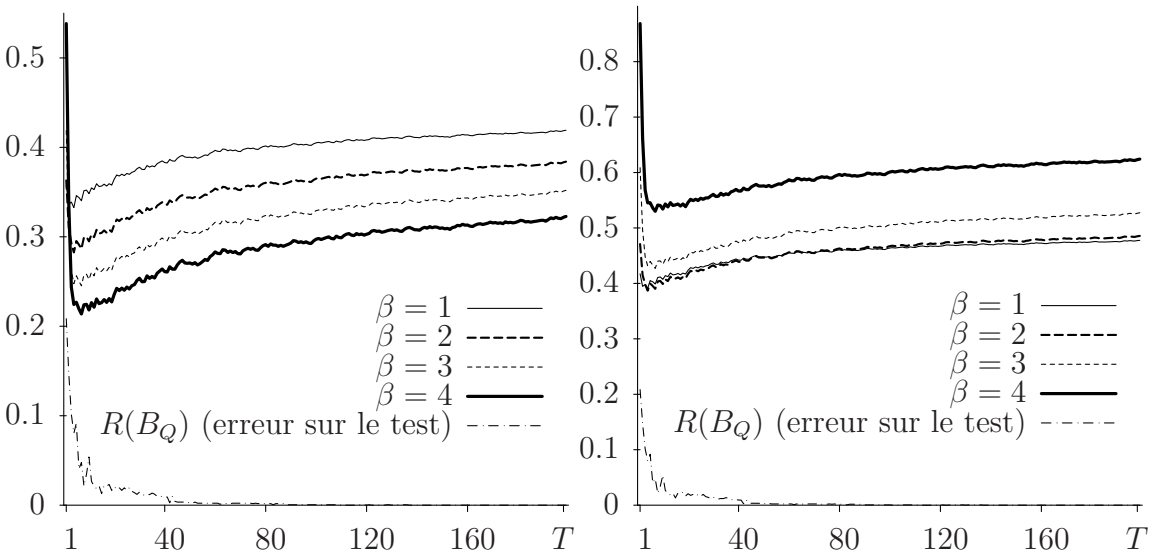

FIGURE 5.4 – Comportement du risque exponentiel empirique évalué sur l'ensemble test (à gauche) et de la borne du risque exponentiel (à droite) pour différentes valeurs de β sur l'ensemble de données Mushroom.

5.5 Risque quadratique

Partant de $f_q(\mathbf{x}, y) = (W_Q(\mathbf{x}, y) - q)^2$, on construit la fonction $\widetilde{f}$ suivante

$$\widetilde{f_q}(\mathbf{x}, y) = a f_q(\mathbf{x}, y) + b\,,$$

où les valeurs a et b sont choisies de sorte à obtenir une fonction de perte de la forme de l'équation 5.6 avec $g(0) = 1$. Donc pour $(\mathbf{x}, y)$ tel que $W_Q(\mathbf{x}, y) = \frac{1}{2}$, l'on doit avoir

$$\widetilde{f_q}(\mathbf{x}, y) = 1\,.$$

La valeur de b doit donc être donnée par $b = 1 - a(\frac{1}{2} - q)^2$. De l'égalité

$$a f_q(\mathbf{x}, y) + 1 - a\Big(\frac{1}{2} - q\Big)^2 = 1 + a\Big[\Big(\frac{1}{2} - q\Big)(2W_Q(\mathbf{x}, y) - 1) + a(2W_Q(\mathbf{x}, y) - 1)^2\Big]\,,$$

on trouve les valeurs suivantes de c_a et k_a pour l'application du théorème 5.1.1 avec la fonction de perte $\widetilde{f_q}(\mathbf{x}, y)$:

$$c_a = \frac{3}{4}a - aq \quad \text{et} \quad k_a = \frac{4 - 4q}{3 - 4q}\,.$$

Ainsi, pour toute valeur du paramètre a, le théorème sur les fonctions de perte générale borne $\underset{(\mathbf{x},y)\sim S}{\mathbf{E}} \widetilde{f}_q(\mathbf{x}, y)$ avec probabilité au moins $1 - \delta$ par la plus grande valeur B telle que l'inégalité suivante est respectée :

$$\mathrm{kl}\Big(\frac{2}{3 - 4q}\Big[\underset{(\mathbf{x},y)\sim S}{\mathbf{E}} f_q(W_Q(\mathbf{x}, y)) - \Big(\frac{1}{2} - q\Big)^2\Big] + \frac{1}{2}\|B\Big) \leq \frac{1}{m}\Big(\frac{4 - 4q}{3 - 4q}\mathrm{KL}(Q\|P) + \log\frac{m + 1}{\delta}\Big).$$

Finalement, on obtient comme borne du risque quadratique

$$\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D}(W_Q(\mathbf{x},y)-q)^2 \leq \left(\frac{3}{2}-2q\right)\left(B-\frac{1}{2}\right)+\left(\frac{1}{2}-q\right)^2 .$$

À noter que toute dépendance au paramètre a a disparu de la borne. Ainsi, peu importe comment la fonction de perte $f_q(\mathbf{x},y)$ est modifiée de sorte à être de la forme de l'équation 5.6, nous obtiendrons la même borne de la quantité $\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D}(W_Q(\mathbf{x},y)-q)^2$.

Risque quadratique en fonction de la marge

Pour des raisons de commodité, dans la suite du document nous définissons le risque quadratique en fonction de la marge réalisée sur les exemples, et non en fonction du taux de désaccord. La marge réalisée par le classificateur par vote de majorité sur un exemple $(\mathbf{x},y)$ est donnée par

$$y \mathop{\mathbf{E}}_{h\sim Q} \mathbf{h}(\mathbf{x}) .$$

Puisque pour tout $h \in \mathcal{H}$ et pour tout $(\mathbf{x},y) \in \mathcal{X}\times\{-1,1\}$ nous avons $yh(\mathbf{x}) = 1-2I(h(\mathbf{x})\neq y)$, la marge réalisée sur un exemple est liée au taux de désaccord par l'égalité

$$y \mathop{\mathbf{E}}_{h\sim Q} \mathbf{h}(\mathbf{x}) = 1-2W_Q(\mathbf{x},y) .$$

Définition 5.5.1. *Soit $\gamma \in (0,1]$. Soit $\mathcal{H}$ un ensemble de classificateurs binaires et Q une distribution sur $\mathcal{H}$, la perte quadratique centrée en γ du classificateur par vote majorité B_Q sur un exemple donné $(\mathbf{x},y)$, notée $\zeta_Q^\gamma(\mathbf{x},y)$, est donnée par*

$$\zeta_Q^\gamma(\mathbf{x},y) = \frac{1}{\gamma}\left(y \mathop{\mathbf{E}}_{h\sim Q} \mathbf{h}(\mathbf{x}) - \gamma\right)^2 .$$

Le risque quadratique, noté ζ_Q^γ, et son estimé empirique sur un ensemble d'exemples étiquetés S, noté $\widehat{\zeta_Q^\gamma}$, sont alors respectivement donnés par

$$\zeta_Q^\gamma = \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D}\zeta_Q^\gamma(\mathbf{x},y) \quad \textit{et} \quad \widehat{\zeta_Q^\gamma} = \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim S}\zeta_Q^\gamma(\mathbf{x},y) .$$

À noter que la fonction de perte quadratique définie ci-dessus correspond à une fonction de la forme de la fonction $\widetilde{f_q}$ (définie en début de section) avec les valeurs de a et de q données par

$$a = \frac{4}{(1-2q)^2} \quad \text{et} \quad q = \frac{1-\gamma}{2} ,$$

c'est-à-dire que nous avons l'égalité

$$\zeta_Q^\gamma(\mathbf{x},y) = \frac{4}{\gamma^2}\left(W_Q(\mathbf{x},y) - \frac{1-\gamma}{2}\right)^2 .$$

Nous avons ainsi la proposition suivante permettant de borner le risque quadratique ζ_Q^γ.

Proposition 5.5.2. *Soit $\gamma \in (0,1]$ et $\delta \in (0,1]$. Soit ζ_Q^γ et $\widehat{\zeta_Q^\gamma}$ respectivement le risque quadratique centré en γ et son estimé empirique sur un échantillon donné de m exemples. Alors, pour tout ensemble $\mathcal{H}$ de classificateurs binaires, pour toute distribution à priori P sur $\mathcal{H}$, nous avons*

$$\Pr_{S\sim D^m}\left(\mathrm{kl}\left(\frac{2(\widehat{\zeta_Q^\gamma}-1)}{1+2\gamma}+\frac{1}{2}\middle\|\frac{2(\zeta_Q^\gamma-1)}{1+2\gamma}+\frac{1}{2}\right)\leq \frac{1}{m}\left(\frac{2+2\gamma}{1+2\gamma}\cdot \mathrm{KL}(Q\|P)+\log\frac{m+1}{\delta}\right)\right)\geq 1-\delta\,.$$

■

5.5.1 Inapprochabilité du risque parabolique

Nous avons vu que la borne du risque exponentielle déduite du théorème sur les fonctions de perte générales se dégradait lorsque le paramètre β augmentait (même dans le cas où la vraie valeur du risque diminuait). Il en est de même pour le risque parabolique : plus le paramètre γ s'approche de zéro, donc plus la parabole décrite par la fonction de perte quadratique est flutée, plus la borne de ζ_Q^γ donnée par la proposition 5.5.2 est lâche.

Le théorème 5.5.5 affirme essentiellement que si l'on fait la supposition qu'une borne du risque de Gibbs découlant d'un théorème de type PAC-Bayes (c'est-à-dire se basant sur les mêmes hypothèses que celui-ci) et calculée à partir d'un ensemble d'apprentissage de taille m donnée ne peut être arbitrairement serrée, alors toute borne du risque quadratique obtenue à partir d'un théorème de type PAC-Bayes se dégradera si l'on fait tendre γ vers zéro (et ce même si la vraie valeur du risque diminue).

Ce théorème s'appuie sur la proposition suivante, qui donne des bornes inférieure et supérieure du risque de Gibbs en fonction du risque quadratique, ainsi que sur la proposition 5.5.4, qui donne une borne inférieure du risque quadratique en fonction du risque de Gibbs.

Proposition 5.5.3. *Soit $\gamma \in (0,1]$. Soit ζ_Q^γ et $R(G_Q)$ respectivement le risque quadratique centré en γ et le risque de Gibbs associé à une distribution Q sur un ensemble $\mathcal{H}$*

de classificateurs binaires. Alors nous avons les inégalités suivantes

$$\frac{1-\gamma}{2} - \frac{\gamma\sqrt{\zeta_Q^\gamma}}{2} \leq R(G_Q) \leq \frac{1-\gamma}{2} + \frac{\gamma\sqrt{\zeta_Q^\gamma}}{2}\,.$$

Démonstration : La perte quadratique est donnée par

$$\begin{aligned}\zeta_Q^\gamma(\mathbf{x}, y) &= \frac{1}{\gamma^2}\Big(y \mathop{\mathbf{E}}_{h\sim Q} h(\mathbf{x}) - \gamma\Big)^2 \\ &= \frac{1}{\gamma^2}\Big(1 - 2W_Q(\mathbf{x}, y) - \gamma\Big)^2 \\ &= \frac{4}{\gamma^2}\Big(W_Q(\mathbf{x}, y)\Big)^2 - \frac{4(1-\gamma)}{\gamma^2} W_Q(\mathbf{x}, y) + \frac{(1-\gamma)^2}{\gamma^2}\,,\end{aligned}$$

il suit que le risque quadratique ζ_Q^γ est égal à

$$\frac{4}{\gamma^2} e_Q - \frac{4(1-\gamma)}{\gamma^2} R(G_Q) + \frac{(1-\gamma)^2}{\gamma^2}\,.$$

Puisque $e_Q \geq (R(G_Q))^2$, nous obtenons l'inégalité

$$4(R(G_Q))^2 - 4(1-\gamma)R(G_Q) + (1-\gamma)^2 - \gamma^2\zeta_Q^\gamma \leq 0\,.$$

En trouvant les racines de cette parabole en $R(G_Q)$, nous obtenons les bornes inférieure et supérieure suivantes du risque de Gibbs :

$$\frac{1-\gamma}{2} - \frac{\gamma\sqrt{\zeta_Q^\gamma}}{2} \leq R(G_Q) \leq \frac{1-\gamma}{2} + \frac{\gamma\sqrt{\zeta_Q^\gamma}}{2}\,.$$

■

Proposition 5.5.4. *Soit $\gamma \in (0, 1]$. Soit ζ_Q^γ et $R(G_Q)$ respectivement le risque quadratique et le risque de Gibbs associés à une distribution Q sur un ensemble $\mathcal{H}$ de classificateurs binaires. Alors nous avons*

$$\zeta_Q^\gamma \geq \frac{4}{\gamma^2}\left[\frac{1-\gamma}{2} - R(G_Q)\right]^2\,.$$

Démonstration : Cette inégalité est une conséquence directe de l'inégalité de Jensen et de la définition de ζ_Q^γ, en effet, comme $(1 - 2W_Q(\mathbf{x}, y) - \gamma)^2$ représente une fonction de $W_Q(\mathbf{x}, y)$ convexe, on a

$$\zeta_Q^\gamma = \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} \zeta_Q^\gamma(\mathbf{x}, y) \geq \frac{1}{\gamma^2}(1 - 2 \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} W_Q(\mathbf{x}, y) - \gamma)^2 = \frac{1}{\gamma^2}(1 - 2R(G_Q) - \gamma)^2\,.$$

Elle peut également s'obtenir sans avoir recours à l'inégalité de Jensen en se servant de la proposition 5.5.3, qui nous fournit les inégalités

$$\frac{1-\gamma}{2} - R(G_Q) \leq \frac{\gamma}{2}\sqrt{\zeta_Q^\gamma} \quad \text{et} \quad R(G_Q) - \frac{1-\gamma}{2} \leq \frac{\gamma}{2}\sqrt{\zeta_Q^\gamma}\,.$$

On remarque alors que soit $\frac{1-\gamma}{2} - R(G_Q) \geq 0$, soit $R(G_Q) - \frac{1-\gamma}{2} \geq 0$, et dans les deux cas, l'une des deux inégalités ci-haut s'applique pour donner l'inégalité recherchée. ■

Théorème 5.5.5 (Inaprochabilité du risque quadratique)**.** *Soit $\mathcal{H}$ un ensemble de classificateurs, S un ensemble d'apprentissage et $\delta \in [0,1)$. Soit $Q(\gamma)$ une fonction retournant une distribution sur $\mathcal{H}$ telle que $R_S(G_{Q(\gamma)}) \in \left[0, \frac{1}{2}\right)$. Notons $A_{Q(\gamma)}$ une borne du risque quadratique $\zeta^\gamma_{Q(\lambda)}$ valide avec probabilité $1-\delta$ et basée sur des hypothèses au plus aussi fortes que celles du théorème PAC-Bayes. Alors, il existe $c > 0$ tel que*

$$A_{Q(\gamma)} \geq \frac{c}{\gamma^2} \quad \forall \gamma\,.$$

Démonstration :

Pour empêcher que l'on puisse obtenir à l'aide de la proposition 5.5.3 une borne supérieure arbitrairement serrée de $R(G_Q)$, il doit exister une valeur $\epsilon > 0$ et une valeur $\gamma_\epsilon > 0$ telles que pour tout $\gamma \in (0, \frac{\gamma_\epsilon}{2})$ satisfaisant $R(G_{Q(\gamma)}) \in \left[\frac{1-\gamma_\epsilon}{2}, \frac{1}{2}\right)$, l'on aie l'inégalité

$$\frac{1}{2} + \epsilon \leq \frac{1-\gamma}{2} + \frac{\gamma\sqrt{A_{Q(\gamma)}}}{2}\,.$$

On déduit de cette inégalité, pour $\gamma \in (0, \frac{\gamma_\epsilon}{2})$, la borne suivante du risque quadratique

$$\left(\frac{2\epsilon}{\gamma} + 1\right)^2 \leq A_{Q(\gamma)}\,.$$

Finalement, pour les $\gamma \in (0, \frac{\gamma_\epsilon}{2})$ telles que $R(G_{Q(\gamma)}) < \frac{1-\gamma_\epsilon}{2}$, la proposition 5.5.4 nous permet d'optenir

$$\begin{aligned} A_{Q(\gamma)} &\geq \frac{4}{\gamma^2}\left(\frac{1-\gamma}{2} - R(G_{Q(\gamma)})\right)^2 \\ &\geq \frac{4}{\gamma^2}\left(\frac{1-\frac{\gamma_\epsilon}{2}}{2} - \frac{1-\gamma_\epsilon}{2}\right)^2 \\ &= \frac{\gamma_\epsilon^2}{4\gamma^2}\,. \end{aligned}$$

Nous obtenons ainsi le résultat recherché en prenant $c = \min\left\{\frac{\gamma_\epsilon^2}{4\gamma^2}, \left(\frac{2\epsilon}{\gamma}+1\right)^2\right\}$.

■

5.6 Classificateur de Gibbs à piges multiples

Le classificateur de Gibbs à piges multiples a été utilisé par Schapire *et al.* (1998), puis par Langford *et al.* (2001), comme outil intermédiaire pour dériver une borne sur le risque d'un classificateur par vote de majorité.

Pour classifier un exemple $\mathbf{x}$, le classificateur de Gibbs à piges multiples choisit aléatoirement un nombre préfixé, N, de classificateurs dans l'ensemble $\mathcal{H}$ de façon indépendante et suivant une distribution de probabilité Q, puis attribue à $\mathbf{x}$ la classe déterminée par un vote démocratique des N classificateurs pigés, c'est-à-dire la classe donnée par

$$\operatorname{sgn}\left(\sum_{i=1}^{N} h_{k(i)}(\mathbf{x})\right),$$

où nous avons noté $h_{k(1)}, h_{k(2)}, \ldots, h_{k(N)}$ les N classificateurs pigés.

5.6.1 Risque du classificateur de Gibbs à piges multiples

La probabilité qu'un classificateur pigé suivant la distribution Q fasse une erreur de classification sur un exemple donné $(\mathbf{x}, y)$ correspond en fait au risque de Gibbs sur cet exemple, donc

$$R_{(\mathbf{x},y)}(G_Q) = W_Q(\mathbf{x}, y) = \mathop{\mathbf{E}}_{h\sim Q} I(h(\mathbf{x}) \neq y)\,.$$

Les N piges étant indépendantes, la probabilité que j classificateurs parmi N fassent une erreur de classification sur $(\mathbf{x}, y)$ est donnée par

$$\binom{N}{j}\Big(W_Q(\mathbf{x}, y)\Big)^j\Big(1 - W_Q(\mathbf{x}, y)\Big)^{N-j}.$$

Pour que le classificateur de Gibbs à N piges fasse une erreur de classification sur l'exemple $(\mathbf{x}, y)$, il faut qu'au moins $\lceil \frac{N}{2} \rceil$ classificateurs parmi les N pigés fassent une erreur de classification sur cet exemple. La fonction de perte associée au classificateur de Gibbs à N piges est donc donnée par

$$W_{Q^N}(\mathbf{x}, y) = \sum_{j=\lceil N/2 \rceil}^{N} \binom{N}{j}\Big(W_Q(\mathbf{x}, y)\Big)^j\Big(1 - W_Q(\mathbf{x}, y)\Big)^{N-j}.$$

La figure 5.5 illustre la fonction de perte $W_{Q^N}(\mathbf{x}, y)$ en fonction de $W_Q(\mathbf{x}, y)$ pour différentes valeurs de N.

En utilisant la fonction de perte $W_{Q^N}(\mathbf{x}, y)$ définie ci-dessus, le vrai risque du classificateur de Gibbs à N piges et son risque empirique sont respectivement donnés par

$$R(G_{Q^N}) \stackrel{\text{déf}}{=} \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} W_{Q^N}(\mathbf{x}, y)$$

et

$$R_S(G_{Q^N}) \stackrel{\text{déf}}{=} \frac{1}{m} \sum_{i=1}^{m} W_{Q^N}(\mathbf{x}_i, y_i)\,.$$

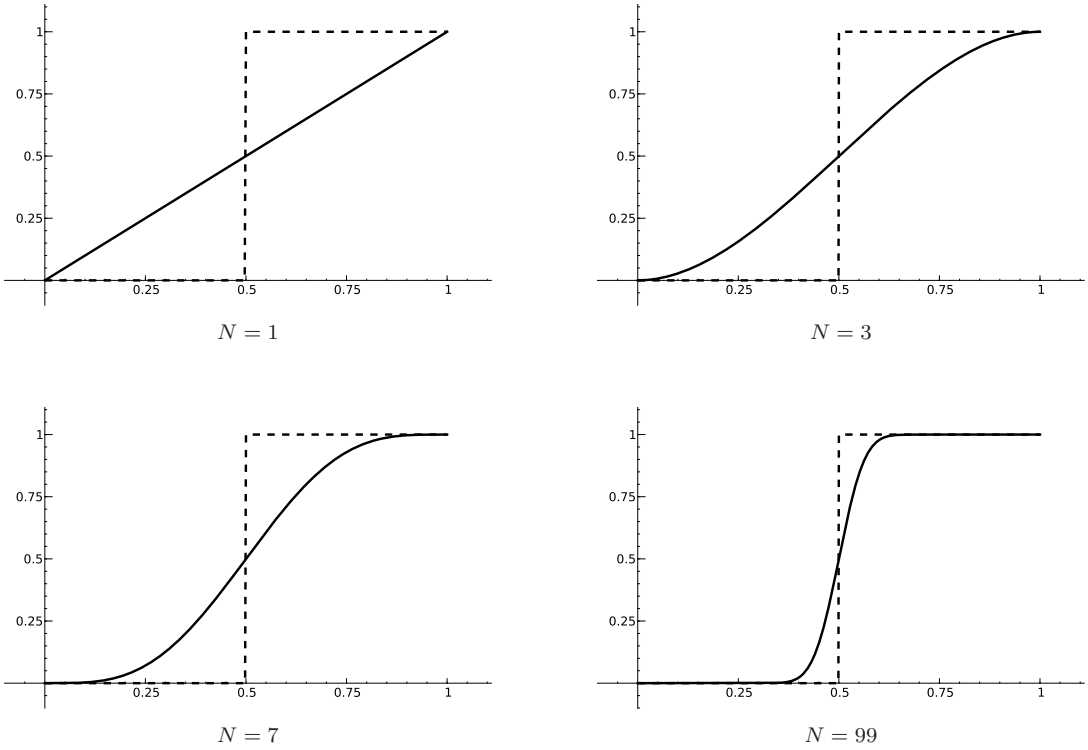

FIGURE 5.5 – Illustration de la fonction de perte du risque de Gibbs à N piges pour $N = 1, 3, 7, 99$ (ligne continue) en fonction de $W_Q(\mathbf{x}, y)$, comparée à la fonction de perte du vote de majorité (ligne pointillée).

5.6.2 Borner le risque du classificateur à piges multiples à l'aide du théorème sur les fonctions de perte générales

Toute fonction de perte infiniment différentiable et valant $\frac{1}{2}$ pour les exemples $(\mathbf{x}, y)$ tels que $W_Q(\mathbf{x}, y) = \frac{1}{2}$ peut s'écrire sous la forme de l'équation 5.1 dans un intervalle autour du point $W_Q(\mathbf{x}, y) = \frac{1}{2}$, et si la série de Taylor obtenue converge dans tout l'intervalle $[0, 1]$, le risque associé à la fonction de perte peut être bornée à l'aide du

théorème sur les fonctions de perte générales (théorème 5.1.1). Lorsque N est impair, la fonction de risque du classificateur de Gibbs à N piges satisfait ces contraintes (cependant, pour N pair, la fonction de risque du classificateur de Gibbs à N piges est strictement supérieure à $\frac{1}{2}$ pour les exemples $(\mathbf{x}, y)$ tels que $W_Q(\mathbf{x}, y) = \frac{1}{2}$, il faut alors utiliser la forme plus générale du théorème (énoncé au théorème 5.1.2) pour borner le risque associé).

Il suffit, pour appliquer le théorème à la fonction de perte du classificateur de Gibbs à N piges (avec N impair), de trouver les termes $g(k)$ permettant d'écrire la fonction de perte sous la forme

$$R_{(\mathbf{x},y)}(G_{Q^N}) = \frac{1}{2} + \frac{1}{2}\sum_{k=1}^{\infty} g(k)\left(2W_Q(\mathbf{x}, y) - 1\right)^k .$$

Comme le développement en série de Taylor d'une fonction $f(W)$ infiniment différentiable au point $\frac{1}{2}$ a la forme

$$f(W) = f\left(\frac{1}{2}\right) + \sum_{k=1}^{\infty} \frac{f^{(k)}\left(\frac{1}{2}\right)}{k!}\left(W - \frac{1}{2}\right)^k ,$$

on peut alors retrouver les termes $g(k)$ comme suit

$$g(k) = \frac{f^{(k)}\left(\frac{1}{2}\right)}{2^{k-1}k!} ,$$

où f correspond ici à la fonction

$$f(W) = \sum_{j=\lceil N/2 \rceil}^{N} \binom{N}{j} W^j (1 - W)^{N-j} .$$

Le tableau 5.6.2 présente les valeurs de c_a et k_a servant au théorème PAC-Bayes sur les fonctions de perte générales appliqué pour borner le risque du classificateur de Gibbs à piges multiples pour les valeurs impaires du nombre de piges allant de 1 à 15. Nous pouvons constater que la valeur de c_a semble presque doubler lorsque le nombre de piges est augmenté de 2, il en résulte que la borne du risque du classificateur de Gibbs à piges multiples devient rapidement très lâche en augmentant le nombre de piges.

5.6.3 Borne directe du risque du classificateur de Gibbs à piges multiples

Il est possible de borner directement le risque du classificateur de Gibbs à piges multiples en voyant ce classificateur comme un simple risque de Gibbs (à une pige) dans

N	c_a	k_a
1	1	1
3	2	$\frac{3}{2} = 1,5$
5	$\frac{7}{2} = 3,5$	$\frac{15}{7} \approx 2,14286$
7	6	$\frac{35}{12} \approx 2,91667$
9	$\frac{83}{8} = 10,375$	$\frac{315}{83} \approx 3,79518$
11	$\frac{73}{4} = 18,25$	$\frac{693}{146} \approx 4,74658$
13	$\frac{523}{16} = 32,6875$	$\frac{3003}{523} \approx 5,74187$
15	$\frac{119}{2} = 59,5$	$\frac{6435}{952} \approx 6,75945$

TABLE 5.1 – Valeurs de c_a et k_a dans le théorème PAC-Bayes sur les fonctions de perte générales appliqué au classificateur de Gibbs à piges multiples pour différentes valeurs du nombre de piges, N.

un ensemble augmenté de classificateurs. La borne que nous obtiendrons ainsi dégradera certes en fonction du nombre de piges, mais dans un ordre beaucoup moindre que la borne obtenue à la section précédente.

Partant d'un ensemble $\mathcal{H}$ contenant n classificateurs, nous créons l'ensemble $\mathcal{H}^N$ constitué des n^N classificateurs $h_{i_1,i_2,\ldots,i_N}$ avec $i_1, i_2, \ldots, i_N \in \{1, 2 \ldots, n\}$ définis comme suit

$$h_{i_1,i_2,\ldots,i_N}(\mathbf{x}) = \operatorname{sgn}\left(\sum_{k=1}^{N} h_{i_k}(\mathbf{x})\right) .$$

En somme, le classificateur $h_{i_1,i_2,\ldots,i_N}$ assigne la classe 1 à l'exemple $\mathbf{x}$ si au moins la moitié des classificateurs parmi $h_{i_1}, h_{i_2}, \ldots, h_{i_N}$ assignent cette classe à l'exemple $\mathbf{x}$.

Pour appliquer le théorème PAC-Bayes classique pour borner le risque du classificateur de Gibbs à N piges, il suffit alors de travailler avec l'ensemble $\mathcal{H}^N$, c'est-à-dire qu'au lieu de piger N classificateurs dans $\mathcal{H}$, l'on pige un seul classificateur dans $\mathcal{H}^N$. La distribution à priori sur $\mathcal{H}^N$ nous est donnée par P^N (où P est une distribution à priori sur $\mathcal{H}$), nous avons donc $P^N(h_{i_1,i_2,\ldots,i_N}) = P(h_1)P(h_2)\cdots P(h_N)$.

On obtient comme valeur de $\mathrm{KL}(P^N \| Q^N)$ dans l'application du théorème, la quantité suivante :

$$\mathrm{KL}(Q^N \| P^N) = N \cdot \mathrm{KL}(Q \| P) ,$$

il en résulte le théorème suivant.

Théorème 5.6.1. *Soit D une distribution, $\mathcal{H}$ un ensemble de classificateurs et P une distribution à priori sur $\mathcal{H}$ et soit $\delta \in (0, 1]$. Alors*

$$\Pr_{S\sim D^m}\left(\forall Q \text{ sur} \mathcal{H} \colon \ \mathrm{kl}(R_S(G_{Q^N}), R(G_{Q^N})) \leq \frac{1}{m}\left[N \cdot \mathrm{KL}(Q\|P) + \log\frac{m+1}{\delta}\right]\right) \geq 1 - \delta .$$

5.7 Conclusion

Nous avons présenté un résultat permettant de borner toute fonction de perte pouvant s'écrire sous la forme une série de Taylor en W_Q centrée en $\frac{1}{2}$ et définie dans l'intervalle $[0, 1]$. Bien que les théorèmes 5.1.1 et 5.1.2 ne permettent pas d'obtenir des bornes serrées du risque du vote de majorité, nos expérimentations démontrent qu'ils permettent d'obtenir des bornes très serrées sur le risque associé à des fonctions de perte complexes, telles que la perte exponentielle et la perte sigmoïdale calculées avec des petites valeurs de β ($\log 2$ dans nos tests).

Chapitre 6

Généralisation et amélioration du théorème PAC-Bayes classique

Nous présentons dans ce chapitre, les démonstrations de quelques théorèmes que nous avons omises dans les chapitres précédents. En particulier, nous y trouvons les démonstrations des théorèmes 2.3.2 et 2.3.3 (qui peuvent également se trouver dans d'autres ouvrages), ainsi que la démonstration du théorème 4.7.1, qui aurait due être publiée dans la version longue de l'article Lacasse *et al.* (2007), cependant, cette dernière n'a pas à ce jour été publiée.

Les démonstrations des théorèmes PAC-Bayes que nous donnons dans ce chapitre découlent de deux théorèmes PAC-Bayes dits généraux, qui correspondent aux théorèmes 6.1.1 et 6.4.1 ; plusieurs théorèmes de type PAC-Bayes se trouvant dans la littérature s'avèrent des corollaires du premier de ces deux théorèmes généraux, alors que le théorème 4.7.1 paru dans Lacasse *et al.* (2007) est un corollaire du second théorème général.

À noter que d'autres théorèmes PAC-Bayes généraux existent dans la littérature, dont celui publié par Catoni (voir Catoni, 2006, 2007).

6.1 Théorème PAC-Bayes général

Théorème 6.1.1. *Soit D une distribution, $\mathcal{H}$ un ensemble de classificateurs et P une distribution à priori sur $\mathcal{H}$ et soit $\delta \in (0, 1]$. Alors pour toute fonction $\mathcal{D}$ de la forme*

$\mathcal{D} : [0,1] \times [0,1] \to \mathbf{R}$*, satisfaisant l'inégalité*

$$\mathop{\mathbf{E}}_{h\sim Q} \mathcal{D}(R_S(h), R(h)) \geq \mathcal{D}\left(\mathop{\mathbf{E}}_{h\sim Q} R_S(h), \mathop{\mathbf{E}}_{h\sim Q} R(h)\right) ,$$

nous avons

$$\Pr_{S\sim D^m}\Bigg(\forall\, Q \text{ sur} \mathcal{H}\colon\ \mathcal{D}(R_S(G_Q), R(G_Q)) \leq \frac{1}{m}\left[\mathrm{KL}(Q\|P) + \log\left(\frac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{h\sim P} e^{m\mathcal{D}(R_S(h),R(h))}\right)\right]\Bigg) \geq 1-\delta\,,$$

où $\mathrm{KL}(Q\|P) \stackrel{\text{def}}{=} \mathop{\mathbf{E}}_{h\sim Q} \log\frac{Q(h)}{P(h)}$ *correspond à la divergence de Kullback-Leibler entre les distributions* Q *et* P*.*

Démonstration : Comme $\mathop{\mathbf{E}}_{h\sim P} e^{m\mathcal{D}(R_S(h),R(h))}$ correspond à une variable aléatoire positive, l'inégalité de Markov s'applique pour donner

$$\Pr_{S\sim D^m}\left(\mathop{\mathbf{E}}_{h\sim P} e^{m\mathcal{D}(R_S(h),R(h))} \leq \frac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{h\sim P} e^{m\mathcal{D}(R_S(h),R(h))}\right) \geq 1-\delta\,.$$

Nous transformons maintenant l'espérance sur P de la partie de gauche de l'inégalité en une espérance sur Q en appliquant le raisonnement suivant (valide pour toute fonction $f(h)$ positive) :

$$\begin{aligned}
\mathop{\mathbf{E}}_{h\sim P} f(h) &= \int_{\mathcal{H}} P(h) f(h) dh \\
&\geq \int_{\mathcal{H}\setminus\{h\,:\,Q(h)=0\}} P(h) f(h) dh \\
&= \int_{\mathcal{H}\setminus\{h\,:\,Q(h)=0\}} \frac{P(h)}{Q(h)} Q(h) f(h) dh \\
&= \mathop{\mathbf{E}}_{h\sim Q} \frac{P(h)}{Q(h)} f(h)\,.
\end{aligned} \tag{6.1}$$

Puis nous exploitons le fait que $\log(x)$ est une fonction monotone croissante pour obtenir l'expression suivante

$$\Pr_{S\sim D^m}\left(\forall Q : \log\left[\mathop{\mathbf{E}}_{h\sim Q}\frac{P(h)}{Q(h)} e^{m\mathcal{D}(R_S(h),R(h))}\right] \leq \log\left[\frac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{h\sim P} e^{m\mathcal{D}(R_S(h),R(h))}\right]\right) \geq 1-\delta\,. \tag{6.2}$$

La fonction $\log(x)$ étant concave, l'inégalité de Jensen s'applique pour donner

$$\begin{aligned}
\log\left[\mathop{\mathbf{E}}_{h\sim Q}\frac{P(h)}{Q(h)} e^{m\mathcal{D}(R_S(h),R(h))}\right] &\geq \mathop{\mathbf{E}}_{h\sim Q} \log\frac{P(h)}{Q(h)} e^{m\mathcal{D}(R_S(h),R(h))} \\
&= \mathop{\mathbf{E}}_{h\sim Q}\log\frac{P(h)}{Q(h)} + \mathop{\mathbf{E}}_{h\sim Q}\log e^{m\mathcal{D}(R_S(h),R(h))} \\
&= -\mathrm{KL}(Q\|P) + m\mathop{\mathbf{E}}_{h\sim Q}\mathcal{D}(R_S(h),R(h))\,.
\end{aligned}$$

Par hypothèse sur la fonction $\mathcal{D}(\cdot,\cdot)$ nous avons l'inégalité

$$\begin{aligned}\mathop{\mathbf{E}}_{h\sim Q}\mathcal{D}(R_S(h),R(h)) &\geq \mathcal{D}\left(\mathop{\mathbf{E}}_{h\sim Q}R_S(h),\mathop{\mathbf{E}}_{h\sim Q}R(h)\right)\\ &= \mathcal{D}\left(R_S(G_Q),R(G_Q)\right),\end{aligned}$$

en portant les deux dernières inégalités dans (6.2), nous obtenons finalement

$$\Pr_{S\sim D^m}\Bigg(\forall\, Q \text{ sur } \mathcal{H}\colon \quad -\mathrm{KL}(Q\|P)+m\mathcal{D}\left(R_S(G_Q),R(G_Q)\right)\leq \log\left[\frac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{h\sim P}e^{m\mathcal{D}(R_S(h),R(h))}\right]\Bigg)\geq 1-\delta\,.$$

■

Corollaire 6.1.2. *Soit D une distribution, $\mathcal{H}$ un ensemble de classificateurs et P une distribution à priori sur $\mathcal{H}$ et soit $\delta\in(0,1]$. Alors*

$$\Pr_{S\sim D^m}\left(\forall\, Q \text{ sur}\mathcal{H}\colon \ \mathrm{kl}(R_S(G_Q)\|R(G_Q))\leq\frac{1}{m}\left[\mathrm{KL}(Q\|P)+\log\frac{\xi(m)}{\delta}\right]\right)\geq 1-\delta\,,$$

où $\xi(m)\stackrel{\text{def}}{=}\sum_{k=0}^{m}\binom{m}{k}(k/m)^k(1-k/m)^{m-k}$.

Démonstration : Le lemme A.0.1 indique que nous pouvons recourir dans le théorème 6.1.1 à la fonction $\mathcal{D}(q,p)=\mathrm{kl}(q\|p)$. Avec ce choix nous trouvons

$$\begin{aligned}\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{h\sim P}e^{m\mathcal{D}(R_S(h),R(h))} &= \mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{h\sim P}e^{mR_S(h)\log\frac{R_S(h)}{R(h)}+m(1-R_S(h))\log\frac{1-R_S(h)}{1-R(h)}}\\ &= \mathop{\mathbf{E}}_{h\sim P}\mathop{\mathbf{E}}_{S\sim D^m}\left(\frac{R_S(h)}{R(h)}\right)^{mR_S(h)}\left(\frac{1-R_S(h)}{1-R(h)}\right)^{m(1-R_S(h))}\\ &= \mathop{\mathbf{E}}_{h\sim P}\sum_{k=0}^{m}\Pr_{S\sim D^m}\left(R_S(h)=\frac{k}{m}\right)\left(\frac{\frac{k}{m}}{R(h)}\right)^k\left(\frac{1-\frac{k}{m}}{1-R(h)}\right)^{m-k}\\ &= \mathop{\mathbf{E}}_{h\sim P}\sum_{k=0}^{m}\binom{m}{k}\Big(R(h)\Big)^k\Big(1-R(h)\Big)^{m-k}\cdot\\ &\qquad\qquad\left(\frac{\frac{k}{m}}{R(h)}\right)^k\left(\frac{1-\frac{k}{m}}{1-R(h)}\right)^{m-k}\\ &= \sum_{k=0}^{m}\binom{m}{k}\left(\frac{k}{m}\right)^k\left(1-\frac{k}{m}\right)^{m-k},\end{aligned}$$

où l'avant dernière égalité provient du fait que $R_S(h)$ correspond à une variable aléatoire suivant une loi binomiale de moyenne $R(h)$. ■

Corollaire 6.1.3 (Théorème PAC-Bayes classique)**.** *Soit D une distribution, $\mathcal{H}$ un ensemble de classificateurs et P une distribution à priori sur $\mathcal{H}$ et soit $\delta \in (0,1]$. Alors*

$$\Pr_{S\sim D^m}\left(\forall Q \text{ sur} \mathcal{H}\colon \ \mathrm{kl}(R_S(G_Q)\|R(G_Q)) \leq \frac{1}{m}\left[\mathrm{KL}(Q\|P) + \log\frac{m+1}{\delta}\right]\right) \geq 1-\delta\,,$$

Démonstration : Se déduit du corollaire 6.1.2 et de l'inégalité $\xi(m) \leq m+1$, cette dernière découle pour sa part de l'inégalité $\binom{m}{k}\left(\frac{k}{m}\right)^k\left(1-\frac{k}{m}\right)^{m-k} \leq 1$, qui est valide pour tout $k, m \in \mathbf{N}$ avec $k \leq m$. ■

La proposition suivante quantifie le gain obtenu avec la borne du corollaire 6.1.2 par rapport à celle de la version plus classique du théorème PAC-Bayes (formulée dans le corollaire 6.1.3). Puisque la différence entre les deux versions du théorème est une transformation du terme $\log(m+1)$ en le terme $\log(\xi(m))$, nous devons mesurer la différence entre ces deux quantités. À la lumière de la proposition suivante, nous pouvons affirmer que

$$\log(\xi(m)) \sim \frac{1}{2}\log(m+1)\,.$$

Proposition 6.1.4. *Soit la fonction $\xi(m)$ définie dans le corollaire 6.1.2, c'est-à-dire $\xi(m) \stackrel{\text{déf}}{=} \sum_{k=0}^{m}\binom{m}{k}(k/m)^k(1-k/m)^{m-k}$. Alors*

$$\xi(m) \in \Theta(\sqrt{m})\,.$$

Démonstration : Des inégalités

$$\sqrt{2\pi n}\left(\frac{n}{e}\right)^n \leq n! \leq \frac{12}{11}\sqrt{2\pi n}\left(\frac{n}{e}\right)^n \quad \forall\ n \geq 1\,,$$

qui découlent du développement en série de Taylor de la fonction Γ de Euler, nous avons que pour $1 \leq k \leq m/2$

$$\begin{aligned}\binom{m}{k}\left(\frac{k}{m}\right)^k\left(1-\frac{k}{m}\right)^{m-k} &\leq \frac{12}{11}\frac{\sqrt{2\pi m}\left(\frac{m}{e}\right)^m k^k(m-k)^{m-k}}{\sqrt{2\pi k}\left(\frac{k}{e}\right)^k\sqrt{2\pi(m-k)}\left(\frac{m-k}{e}\right)^{m-k}m^m}\\ &= \frac{12}{11\sqrt{2\pi}}\frac{\sqrt{m}}{\sqrt{k}\sqrt{m-k}}\\ &\leq \frac{12}{11\sqrt{\pi}}\frac{1}{\sqrt{k}}.\end{aligned}$$

Donc

$$\begin{aligned}\xi(m) &\leq 2 + \frac{24}{11\sqrt{\pi}}\sum_{k=1}^{m/2}k^{-1/2}\\ &\in O(\sqrt{m})\,.\end{aligned}$$

De plus,

$$\begin{aligned}
&\binom{m}{\lfloor m/4\rfloor}\left(\frac{\lfloor m/4\rfloor}{m}\right)^{\lfloor m/4\rfloor}\left(1-\frac{\lfloor m/4\rfloor}{m}\right)^{m-\lfloor m/4\rfloor} \\
&\qquad\sim \frac{\sqrt{2\pi m}\left(\frac{m}{e}\right)^m\left(\frac{\lfloor m/4\rfloor}{m}\right)^{\lfloor m/4\rfloor}\left(\frac{m-\lfloor m/4\rfloor}{m}\right)^{m-\lfloor m/4\rfloor}}{\sqrt{2\pi\lfloor m/4\rfloor}\left(\frac{\lfloor m/4\rfloor}{e}\right)^{\lfloor m/4\rfloor}\sqrt{2\pi(m-\lfloor m/4\rfloor)}\left(\frac{m-\lfloor m/4\rfloor}{e}\right)^{m-\lfloor m/4\rfloor}} \\
&\qquad= \frac{\sqrt{2\pi m}}{\sqrt{2\pi\lfloor m/4\rfloor}\sqrt{2\pi(m-\lfloor m/4\rfloor)}} \\
&\qquad\sim \frac{\sqrt{2\pi m}}{\sqrt{\pi m/2}\sqrt{3\pi m/2}} \\
&\qquad= \frac{2\sqrt{2}}{\sqrt{3\pi m}}\,.
\end{aligned}$$

Comme $\binom{m}{k}\left(\frac{k}{m}\right)^k\left(1-\frac{k}{m}\right)^{m-k} \geq \binom{m}{\lfloor m/4\rfloor}\left(\frac{\lfloor m/4\rfloor}{m}\right)^{\lfloor m/4\rfloor}\left(1-\frac{\lfloor m/4\rfloor}{m}\right)^{m-\lfloor m/4\rfloor}$ pour tout k tel que $\lfloor m/4\rfloor \leq k \leq m-\lfloor m/4\rfloor$, il suit qu'asymptotiquement, nous avons

$$\begin{aligned}
\xi(m) &= \sum_{k=0}^{m}\binom{m}{k}\left(\frac{k}{m}\right)^k\left(1-\frac{k}{m}\right)^{m-k} \\
&\geq \frac{m}{2}\cdot\frac{2\sqrt{2}}{\sqrt{3\pi m}} \\
&= \frac{\sqrt{2}}{\sqrt{3\pi}}\sqrt{m} \\
&\in \Omega(\sqrt{m})\,.
\end{aligned}$$

■

Remarquons que nous pouvons facilement donner plus de précision au résultat de la proposition 6.1.4 en bornant la somme $\sum_{k=1}^{m/2} k^{-1/2}$ par l'intégrale $\int_0^{m/2} x^{-1/2}dx$. En effet, comme $\sum_{k=1}^{m/2} k^{-1/2} \leq \sqrt{2m}$, nous obtenons l'inégalité

$$\begin{aligned}
\xi(m) &\leq 2 + \frac{24\sqrt{2}}{11\sqrt{\pi}}\sqrt{m} \\
&< 2 + \frac{7}{4}\sqrt{m}\,.
\end{aligned}$$

Pour m supérieur à 64, cette borne supérieure de $\xi(m)$ est plus serrée que celle de Maurer (2004), qui correspond à $\xi(m) \leq 2\sqrt{m}$. Dans le papier de Maurer, on retrouve

également la borne inférieure $\sqrt{m} \leq \xi(m)$. Cette dernière ne peut pas être déduite de la proposition 6.1.4.

Comme dernier exemple de théorème que l'on peut retrouver à partir du théorème PAC-Bayes général, nous voyons qu'en optant pour la fonction de pseudo-distance donnée par $\mathcal{D}(q,p) = 2(q-p)^2$, nous obtenons une version du théorème PAC-Bayes comparable au théorème 1 de McAllester (1999b).

Corollaire 6.1.5 (Théorème PAC-Bayes version McAllester)**.** *Soit D une distribution, $\mathcal{H}$ un ensemble de classificateurs et P une distribution à priori sur $\mathcal{H}$ et soit $\delta \in (0,1]$. Alors*

$$\Pr_{S\sim D^m}\left(\forall\, Q \text{ sur} \mathcal{H}\colon \; R(G_Q) \leq R_S(G_Q) + \sqrt{\frac{1}{2m}\left[\mathrm{KL}(Q\|P) + \log\frac{m+1}{\delta}\right]}\right) \geq 1-\delta\,,$$

Démonstration : Il suffit d'appliquer le théorème 6.1.1 en utilisant la fonction convexe $\mathcal{D}(q,p) = 2(q-p)^2$. En effet, nous obtenons alors

$$\begin{aligned}\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{h\sim P} e^{m2(R_S(h)-R(h))^2} &\leq \mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{h\sim P} e^{m\mathrm{kl}(R_S(h),R(h))}\\ &\leq m+1\,,\end{aligned}$$

où nous avons utilisé l'inégalité $2(q-p)^2 \leq \mathrm{kl}(q,p)$ ainsi que les calculs des corollaires 6.1.2 et 6.1.3. En portant cette inégalité dans le théorème 6.1.1, nous trouvons

$$\Pr_{S\sim D^m}\left(\forall\, Q \text{ sur } \mathcal{H}\colon \; 2(R(G_Q) - R_S(G_Q))^2 \leq \frac{1}{m}\left[\mathrm{KL}(Q\|P) + \log\frac{m+1}{\delta}\right]\right) \geq 1-\delta\,.$$

■

Nous terminons cette section avec une remarque permettant de justifier le théorème 2.3.3, qui est utilisé dans les démonstrations des théorèmes 4.6.1 et 6.1.1.

Remarque 6.1.6. *Dans la démonstration du théorème 6.1.1, la seule contrainte que doivent satisfaire les fonctions $R(h)$ et $R_S(h)$ est d'être à valeurs dans l'intervalle $[0,1]$ (parce que le domaine de définition de la fonction $\mathcal{D}$ est l'ensemble $[0,1]\times[0,1]$). Les contraintes additionnelles que doivent satisfaire ces fonctions dans la démonstration du corollaire 6.1.2 proviennent du fait que $R_S(h)$ doit représenter une variable aléatoire suivant une distribution binomiale d'espérance $R(h)$. Par conséquent, le corollaire 6.1.2, tout comme les corollaires 6.1.3 et 6.1.5, demeure valide en remplaçant les fonctions $R(h)$ et $R_S(h)$ par n'importe quelles fonctions $R^\ell(h)$ et $R^\ell_S(h)$ telles que*

$$R^\ell(h) = \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D}\ell(h,\mathbf{x},y) \qquad \textit{et} \qquad R^\ell_S(h) = \frac{1}{|S|}\sum_{i=1}^{|S|}\ell(h,\mathbf{x},y)\,,$$

où $\ell(h, \mathbf{x}, y)$ est une fonction de perte de la forme

$$\ell : \mathcal{H} \times \mathcal{X} \times \mathcal{Y} \to \{0, 1\}\,.$$

6.2 Théorème PAC-Bayes dépendant d'un hyperparamètre

Le prochain corollaire, qui découle également directement du théorème 6.1.1, énonce une nouvelle version du théorème PAC-Bayes utilisant un hyperparamètre que nous notons C ; ce résultat a d'abord été trouvé par Catoni, voir Catoni (2006) (théorème 1.5) et Catoni (2007) (théorème 1.2.1). Nous fournissons ici une preuve de ce résultat que l'on pourrait qualifier d'élémentaire, c'est-à-dire ne faisant pas intervenir toute la mécanique de la preuve originale.

Corollaire 6.2.1 (Borne PAC-Bayes hyperparamétrée, Catoni 2006)**.** *Soit D une distribution sur $\mathcal{X} \times \mathcal{Y}$. Soit P une distribution sur un ensemble $\mathcal{H}$ de classificateurs $h : \mathcal{X} \to \mathcal{Y}$. Soit $\delta \in (0, 1]$ et $C > 0$. Nous avons*

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\forall\, Q \text{ sur } \mathcal{H}: \\ R(G_Q) \;\le\; \frac{1}{1-e^{-C}}\left\{1 - \exp\left[-\left(C\cdot R_S(G_Q) + \frac{1}{m}\left[\mathrm{KL}(Q\|P) + \log\frac{1}{\delta}\right]\right)\right]\right\}\end{array}\right) \ge 1-\delta.$$

Démonstration : Il suffit d'appliquer le théorème 6.1.1 en utilisant la fonction convexe $\mathcal{D}(q, p) = -\log\left(1 - p(1 - e^{-C})\right) - C\cdot q$. En effet, dans ce cas nous obtenons

$$\begin{aligned}
&\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{h\sim P} e^{m\mathcal{D}(R_S(h),R(h))} \\
&\qquad= \mathop{\mathbf{E}}_{h\sim P}\mathop{\mathbf{E}}_{S\sim D^m} e^{-m\log\left(1-R(h)(1-e^{-C})\right)-CmR_S(h)} \\
&\qquad= \mathop{\mathbf{E}}_{h\sim P} e^{-m\log\left(1-R(h)(1-e^{-C})\right)}\sum_{k=0}^{m}\Pr_{S\sim D^m}\left(R_S(h)=\frac{k}{m}\right)e^{-Ck} \\
&\qquad= \mathop{\mathbf{E}}_{h\sim P}\left(1-R(h)(1-e^{-C})\right)^{-m}\sum_{k=0}^{m}\binom{m}{k}R(h)^k(1-R(h))^{m-k}e^{-Ck} \\
&\qquad= \mathop{\mathbf{E}}_{h\sim P}\left(1-R(h)(1-e^{-C})\right)^{-m}\left(R(h)e^{-C}+(1-R(h))\right)^m \\
&\qquad= 1\,,
\end{aligned}$$

en portant dans le théorème 6.1.1, nous obtenons le résultat

$$\Pr_{S\sim D^m}\Bigg(\forall\, Q \text{ sur } \mathcal{H}:$$
$$-\log\Big(1-R(G_Q)(1-e^{-C})\Big)-C\cdot R_S(G_Q)\leq\frac{1}{m}\left[\mathrm{KL}(Q\|P)+\log\frac{1}{\delta}\right]\Bigg)\geq 1-\delta\,,$$

duquel le corollaire découle directement. ■

La proposition suivante indique qu'il existe toujours une valeur C qui soit telle que la borne du théorème 6.2.1, appliquée avec cette valeur, soit plus serrée que la borne du théorème PAC-Bayes classique (même dans sa version optimisée, corollaire 6.1.2).

Proposition 6.2.2. *Soit $R, R_S \in [0,1]$ tels que $R_S \leq R < 1$. Alors*

$$\max_{c\geq 0}\left\{-\log\Big(1-R(1-e^{-c})\Big)-cR_S\right\}=\mathrm{kl}(R_S\|R).$$

Démonstration : Posons $g(c)=-\log\Big(1-R(1-e^{-c})\Big)-cR_S$. On a

$$\frac{\partial g}{\partial c}=\frac{R}{R-e^c(R-1)}-R_S$$

et donc $\dfrac{\partial g}{\partial c}(c_{opt})=0$ si et seulement si

$$R_S=\frac{R}{R-e^{c_{opt}}(R-1)},$$

ce qui donne pour valeur de c_{opt}

$$c_{opt}=\log\left(\frac{RR_S-R}{RR_S-R_S}\right).$$

En portant la valeur de c_{opt} dans g, on trouve

$$\begin{aligned}
g(c_{opt}) &= -\log\left(1-R\Big(1-\frac{RR_S-R_S}{RR_S-R}\Big)\right)-R_S\log\frac{RR_S-R}{RR_S-R_S}\\
&= -\log\left(\frac{(1-R)(R_S-1)+R_S(R-1)}{R_S-1}\right)-R_S\log\frac{R}{R_S}-R_S\log\frac{R_S-1}{R-1}\\
&= -\log\frac{R-1}{R_S-1}+R_S\log\frac{R_S}{R}-R_S\log\frac{1-R_S}{1-R}\\
&= (1-R_S)\log\frac{1-R_S}{1-R}+R_S\log\frac{R_S}{R}\\
&= \mathrm{kl}(R_S\|R)\,.
\end{aligned}$$

■

La proposition 6.2.2 mène aux deux observations suivantes :

1. Si l'on néglige le terme $\log(\xi(m))$ apparaissant dans le corollaire 6.1.2, l'on voit que la borne (non valide) qui en découle est plus faible ou égale à celle du corollaire 6.2.1 peu importe la valeur de l'hyperparamètre C utilisé dans cette dernière (il en est également de même pour le théorème PAC-Bayes classique en négligeant le terme $\log(m+1)$). Ainsi, le gain en précision de la borne que nous pourrons potentiellement obtenir en passant du corollaire 6.1.2 au corollaire 6.2.1 ne pourra être très grand.

2. Il existe toujours une valeur de C telle que la borne du théorème 6.2.1 devient équivalente à la borne suivante :

$$\Pr_{S\sim D^m}\left(\forall\ Q \text{ sur } \mathcal{H}\text{:}\mathrm{kl}(R(h)\|R_S(h)) \leq \frac{1}{m}\left[\mathrm{KL}(Q\|P) + \log\frac{1}{\delta}\right]\right) \geq 1-\delta\,,$$

   c'est-à-dire, équivalente aux bornes des corollaires 6.1.3 et 6.1.2 si l'on remplace les fonctions respectives $m+1$ et $\xi(m)$ intervenant dans ces bornes par la constante 1. La borne ainsi obtenue sera strictement inférieure à celles de ces corollaires.

   Plus encore, notons C_{max} la quantité

$$\underset{c\geq 0}{\operatorname{argmax}}\left\{-\log\left(1-R(1-e^{-c})\right) - cR_S\right\},$$

   la fonction $f(c) = -\log\left(1-R(1-e^{-c})\right) - cR_S$ étant continue, il existe un intervalle $I = [C_{max}-\epsilon_1, C_{max}+\epsilon_2]$, où $\epsilon_1, \epsilon_2 > 0$, pour lequel la borne du corollaire 6.2.1 appliquée avec n'importe quel $C \in I$ sera plus serrée que celle du corollaire 6.1.2.

   Cependant, la quantité C_{max} ne peut être déterminée sans la connaissance de $R_S(G_Q)$, c'est-à-dire sans avoir regardé les données d'apprentissage. Pour optimiser la borne du corollaire 6.2.1 il peut donc s'avérer nécessaire d'appliquer le corollaire avec plusieurs valeurs de C (disons T valeurs distinctes), et de choisir la plus faible des bornes obtenues. En faisant cela, il est par contre nécessaire, pour garantir la validité de la borne, d'appliquer à chaque fois le corollaire avec un taux d'incertitude δ/T (au lieu de δ) si l'on désire que la borne résultante soit valide avec un taux de confiance de $1-\delta$.

6.3 Réécriture du théorème bornant les risques associés à des fonctions de perte générale

Au chapitre 5, nous avons présenté un résultat (théorème 5.1.2) permettant de borner le risque de toute fonction de perte pouvant s'exprimer sous la forme d'une série de Taylor fonction de $W_Q(\mathbf{x}, y)$, définie dans l'intervalle $[0, 1]$ et centrée en $W_Q(\mathbf{x}, y) = \frac{1}{2}$. La démonstration de ce théorème se trouve n'être qu'une simple application du théorème PAC-Bayes classique (corollaire 6.1.3 dans ce chapitre, ou bien 2.3.2 dans le chapitre 2), en appliquant le corollaire 6.2.1 au lieu du théorème PAC-Bayes classique dans la démonstration, nous obtenons le résultat suivant.

Théorème 6.3.1. *Soit $\zeta_Q(\mathbf{x}, y)$ une fonction de perte de la forme de l'équation (5.6). Soit ζ_Q et $\widehat{\zeta_Q}$ respectivement le risque associé à $\zeta_Q(\mathbf{x}, y)$ et son estimé empirique sur un échantillon de m exemples. Alors, pour toute distribution à priori P sur l'ensemble de classificateurs binaires $\mathcal{H}$ et pour tout $\delta \in (0, 1]$, nous avons*

$$\Pr_{S\sim D^m}\left(\forall Q \text{ sur } \mathcal{H} : \zeta_Q \leq 2c_a\left(M - \frac{1}{2}\right) + g(0)\right) \geq 1 - \delta\,,$$

où

$$M = \frac{1}{1-e^{-C}}\left\{1 - \exp\left[-\left(C\cdot\left[\frac{1}{2c_a}\left(\widehat{\zeta_Q} - g(0)\right) + \frac{1}{2}\right] + \frac{1}{m}\left[k_a\cdot \mathrm{KL}(Q\|P) + \log\frac{1}{\delta}\right]\right)\right]\right\},$$

$$c_a \overset{\text{déf}}{=} \sum_{k=1}^{\infty}|g(k)| \quad \textit{et} \quad k_a \overset{\text{déf}}{=} \frac{1}{c_a}\sum_{k=1}^{\infty} k\cdot|g(k)|\,.$$

Démonstration : Identique à la preuve du théorème 5.1.1 mis à part le fait que l'on fait appel au corollaire 6.2.1 à la place du corollaire 6.1.3. ■

6.4 Amélioration de la borne sur C_Q

Nous pouvons également obtenir une forme générale pour le théorème 4.7.1, ce qu'énonce le théorème suivant. Le théorème 4.7.1 se trouve être un corollaire de ce théorème général, voir le corollaire 6.4.3. En prime, tout comme cela a été fait pour le théorème PAC-Bayes original, nous pouvons déduire de ce théorème général un énoncé plus précis du théorème 4.7.1 (c'est-à-dire fournissant une borne plus serrée), voir le corollaire 6.4.2.

Théorème 6.4.1. *Soit D une distribution sur un ensemble $\mathcal{X}$, $\mathcal{H}$ un ensemble de classificateurs définis sur $\mathcal{X}$ et P une distribution à priori sur $\mathcal{H}$ et soit $\delta \in (0,1]$. Alors*

$$\Pr_{S\sim D^m}\Bigg(\forall Q \text{ sur} \mathcal{H}\colon\ \mathcal{D}(\widehat{a_Q},\widehat{\beta_Q}\|a_Q,\beta_Q) \leq \frac{1}{m}\left[2\cdot \mathrm{KL}(Q\|P) + \log\left(\frac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{(h_1,h_2)\sim P^{(2)}} e^{m\mathcal{D}\left(\widehat{a_{12}},\widehat{\beta_{12}}\|a_{12},\beta_{12}\right)}\right)\right]\Bigg) \geq 1-\delta\,,$$

où $\mathrm{KL}(Q\|P) \stackrel{\text{def}}{=} \mathop{\mathbf{E}}_{h\sim Q}\log\frac{Q(h)}{P(h)}$ *correspond à la divergence de Kullback-Leibler entre Q et P et a_Q et β_Q sont deux valeurs quelconques parmi les trois quantités e_Q, s_Q et d_Q, $\widehat{a_Q}$ et $\widehat{\beta_Q}$ sont leurs valeurs empiriques observées sur l'ensemble S et $\mathcal{D}(q_1,q_1\|p_1,p_2)$ est une fonction de la forme $\mathcal{D}:[0,1]^2\times[0,1]^2\to\mathbf{R}$ qui doit respecter l'inégalité suivante :*

$$\mathcal{D}(\widehat{a_Q},\widehat{\beta_Q}\|a_Q,\beta_Q) \leq \mathop{\mathbf{E}}_{(h_1,h_2)\sim Q^{(2)}}\mathcal{D}(\widehat{a_{12}},\widehat{\beta_{12}}\|a_{12},\beta_{12})\,. \tag{6.3}$$

Démonstration : Comme $\mathop{\mathbf{E}}_{(h_1,h_2)\sim P^{(2)}} e^{m\mathcal{D}(\widehat{a_{12}},\widehat{\beta_{12}}\|a_{12},\beta_{12})}$ correspond à une variable aléatoire positive, l'inégalité de Markov s'applique pour donner

$$\Pr_{S\sim D^m}\left(\mathop{\mathbf{E}}_{(h_1,h_2)\sim P^{(2)}} e^{m\mathcal{D}(\widehat{a_{12}},\widehat{\beta_{12}}\|a_{12},\beta_{12})} \leq \frac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{(h_1,h_2)\sim P^{(2)}} e^{m\mathcal{D}(\widehat{a_{12}},\widehat{\beta_{12}}\|a_{12},\beta_{12})}\right) \geq 1-\delta\,.$$

Nous transformons maintenant l'espérance sur P de la partie de gauche de l'inégalité en une espérance sur Q, c'est-à-dire que nous appliquons l'inégalité

$$\mathop{\mathbf{E}}_{(h_1,h_2)\sim P^{(2)}} f(h_1,h_2) \geq \mathop{\mathbf{E}}_{(h_1,h_2)\sim Q^{(2)}}\frac{P^{(2)}(h_1,h_2)}{Q^{(2)}(h_1,h_2)}f(h_1,h_2)\,,$$

(voir l'inégalité 6.1) puis nous exploitons le fait que $\log(x)$ est une fonction monotone croissante pour obtenir l'expression suivante

$$\Pr_{S\sim D^m}\Bigg(\forall Q : \log\left[\mathop{\mathbf{E}}_{(h_1,h_2)\sim Q2^{(2)}}\frac{P^{(2)}(h_1,h_2)}{Q^{(2)}(h_1,h_2)}e^{m\mathcal{D}(\widehat{a_{12}},\widehat{\beta_{12}}\|a_{12},\beta_{12})}\right] \leq \log\left[\frac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{(h_1,h_2)\sim P^{(2)}} e^{m\mathcal{D}(\widehat{a_{12}},\widehat{\beta_{12}}\|a_{12},\beta_{12})}\right]\Bigg) \geq 1-\delta\,. \tag{6.4}$$

La fonction $\log(x)$ étant concave, l'inégalité de Jensen s'applique pour donner

$$\begin{aligned}
\log \left[\mathop{\mathbf{E}}_{(h_1,h_2)\sim Q^{(2)}} \frac{P^{(2)}(h_1,h_2)}{Q^{(2)}(h_1,h_2)} e^{m\mathcal{D}(\widehat{a_{12}},\widehat{\beta_{12}}\|a_{12},\beta_{12})} \right] & \\
&\geq \mathop{\mathbf{E}}_{(h_1,h_2)\sim Q^{(2)}} \log \frac{P^{(2)}(h_1,h_2)}{Q^{(2)}(h_1,h_2)} e^{m\mathcal{D}(\widehat{a_{12}},\widehat{\beta_{12}}\|a_{12},\beta_{12})} \\
&= \mathop{\mathbf{E}}_{(h_1,h_2)\sim Q^{(2)}} \log \frac{P^{(2)}(h_1,h_2)}{Q^{(2)}(h_1,h_2)} + \mathop{\mathbf{E}}_{(h_1,h_2)\sim Q^{(2)}} \log e^{m\mathcal{D}(\widehat{a_{12}},\widehat{\beta_{12}}\|a_{12},\beta_{12})} \\
&= \mathrm{KL}(Q^{(2)}\|P^{(2)}) + m \mathop{\mathbf{E}}_{(h_1,h_2)\sim Q^{(2)}} \mathcal{D}(\widehat{a_{12}},\widehat{\beta_{12}}\|a_{12},\beta_{12}) \\
&= 2\cdot \mathrm{KL}(Q\|P) + m \mathop{\mathbf{E}}_{(h_1,h_2)\sim Q^{(2)}} \mathcal{D}(\widehat{a_{12}},\widehat{\beta_{12}}\|a_{12},\beta_{12}) \\
&\geq 2\cdot \mathrm{KL}(Q\|P) + m\mathcal{D}(\widehat{a_Q},\widehat{\beta_Q}\|a_Q,\beta_Q)
\end{aligned}$$

où la dernière inégalité est une conséquence de l'inégalité (6.3) que l'on suppose valide pour $\mathcal{D}$.

■

Le lemme A.0.2 permet d'appliquer le théorème 6.4.1 avec la fonction de pseudo-distance $\mathcal{D}(\widehat{a_Q},\widehat{\beta_Q}\|a_Q,\beta_Q) = \mathrm{kl}(\widehat{a_Q},\widehat{\beta_Q}\|a_Q,\beta_Q)$. Ceci mène aux corollaires 6.4.2 et 6.4.3, ce dernier corollaire correspond en fait au théorème 4.7.1, nous obtenons ainsi une démonstration de celui-ci (laquelle avait été omise au chapitre 4).

Corollaire 6.4.2. *Soit $\mathcal{H}$ un ensemble de classificateurs et P une distribution sur $\mathcal{H}$ et soit $\delta \in (0,1]$. Alors nous avons*

$$\Pr_{S\sim D^m} \left(\forall Q : \mathrm{kl}(\widehat{a_Q},\widehat{\beta_Q}\|a_Q,\beta_Q) \leq \frac{1}{m} \left[2\cdot\mathrm{KL}(Q\|P) + \log \frac{\xi_2(m)}{\delta} \right] \right) \geq 1-\delta$$

où a_Q et β_Q sont deux valeurs quelconques parmi e_Q, s_Q et d_Q, $\widehat{a_Q}$ et $\widehat{\beta_Q}$ sont leurs valeurs empiriques observées sur l'ensemble S et $\xi_2(m)$ est donnée par

$$\xi_2(m) \stackrel{\text{déf}}{=} \sum_{j=0}^{m} \sum_{k=0}^{m-j} \binom{m}{j} \binom{m-j}{k} \left(\frac{j}{m}\right)^j \left(\frac{k}{m}\right)^k \left(1-\frac{j}{m}-\frac{k}{m}\right)^{m-j-k} .$$

Démonstration : Conséquence du théorème 6.4.1 en utilisant

$$\mathcal{D}(\widehat{a_{12}},\widehat{\beta_{12}}\|a_{12},\beta_{12}) = \mathrm{kl}(\widehat{a_{12}},\widehat{\beta_{12}}\|a_{12},\beta_{12}) ,$$

qui respecte bien l'inégalité 6.3 (selon le lemme A.0.2). Le corollaire découle directement de cette application du théorème, en effet, dans ce cas nous avons

$$
\begin{aligned}
&\mathop{\mathbf{E}}_{S\sim D^m}\ \mathop{\mathbf{E}}_{(h_1,h_2)\sim P^{(2)}} e^{\mathrm{kl}(\widehat{a_{12}},\widehat{\beta_{12}}\|a_{12},\beta_{12})} \\
&\quad= \mathop{\mathbf{E}}_{(h_1,h_2)\sim P^{(2)}}\ \mathop{\mathbf{E}}_{S\sim D^m} \left(\frac{\widehat{a_{12}}}{a_{12}}\right)^{m\widehat{a_{12}}} \left(\frac{\widehat{\beta_{12}}}{\beta_{12}}\right)^{m\widehat{\beta_{12}}} \left(\frac{1-\widehat{a_{12}}-\widehat{\beta_{12}}i}{1a_{12}-\beta_{12}}\right)^{m(1-\widehat{a_{12}}-\widehat{\beta_{12}})} \\
&\quad= \mathop{\mathbf{E}}_{(h_1,h_2)\sim P^{(2)}} \sum_{j=0}^{m}\sum_{k=0}^{m-j}\Bigg[\Pr_{S\sim D^m}\left(\widehat{a_{12}}=\frac{j}{m}\wedge\widehat{\beta_{12}}=\frac{k}{m}\right) \\
&\qquad\qquad \cdot\left(\frac{\widehat{a_{12}}}{a_{12}}\right)^{m\widehat{a_{12}}} \left(\frac{\widehat{\beta_{12}}}{\beta_{12}}\right)^{m\widehat{\beta_{12}}} \left(\frac{1-\widehat{a_{12}}-\widehat{\beta_{12}}}{1-a_{12}-\beta_{12}}\right)^{m(1-\widehat{a_{12}}-\widehat{\beta_{12}})}\Bigg] \\
&\quad= \mathop{\mathbf{E}}_{(h_1,h_2)\sim P^{(2)}} \sum_{j=0}^{m}\sum_{k=0}^{m-j}\Bigg[\binom{m}{j}\binom{m-j}{k}(a_{12})^j(\beta_{12})^k(1-a_{12}-\beta_{12})^{m-j-k} \\
&\qquad\qquad \cdot\left(\frac{j/m}{a_{12}}\right)^j \left(\frac{k/m}{\beta_{12}}\right)^k \left(\frac{1-j/m-k/m}{1-a_{12}-\beta_{12}}\right)^{m-j-k}\Bigg] \\
&\quad= \sum_{j=0}^{m}\sum_{k=0}^{m-j}\binom{m}{j}\binom{m-j}{k}\left(\frac{j}{m}\right)^j\left(\frac{k}{m}\right)^k\left(1-\frac{j}{m}-\frac{k}{m}\right)^{m-j-k}
\end{aligned}
$$

■

À partir du corollaire 6.4.2 nous pouvons facilement retrouvé le théorème 4.7.1, qui est en fait le théorème 1 de l'article Lacasse *et al.* (2007).

Corollaire 6.4.3. *Soit $\mathcal{H}$ un ensemble de classificateurs et P une distribution sur $\mathcal{H}$ et soit $\delta \in (0,1]$. Alors nous avons*

$$\Pr_{S\sim D^m}\left(\forall Q : \mathrm{kl}(\widehat{a_Q},\widehat{\beta_Q}\|a_Q,\beta_Q) \leq \frac{1}{m}\left[2\cdot\mathrm{KL}(Q\|P)+\log\frac{(m+1)(m+2)}{2\delta}\right]\right)\geq 1-\delta$$

où a_Q et β_Q sont deux valeurs quelconque parmi e_Q, s_Q et d_Q, $\widehat{a_Q}$ et $\widehat{\beta_Q}$ sont leurs valeurs empiriques observées sur l'ensemble S.

Démonstration : L'inégalité $\binom{m}{j}\binom{m-j}{k}\left(\frac{j}{m}\right)^j\left(\frac{k}{m}\right)^k\left(1-\frac{j}{m}-\frac{k}{m}\right)^{m-j-k} \leq 1$ est valide

pour tout $m, j, k \in \mathbf{N}$ tels que $k \leq m - j$. Nous trouvons alors

$$\begin{aligned}\sum_{j=0}^{m}\sum_{k=0}^{m-j}\binom{m}{j}\binom{m-j}{k}\left(\frac{j}{m}\right)^{j}\left(\frac{k}{m}\right)^{k}\left(1-\frac{j}{m}-\frac{k}{m}\right)^{m-j-k} &\leq \sum_{j=0}^{m}\sum_{k=0}^{m-j}1\\ &= \sum_{j=0}^{m}(m-j+1)\\ &= \sum_{j=1}^{m+1}j\\ &= \frac{(m+1)(m+2)}{2}.\end{aligned}$$

Le corollaire 6.4.2 demeure ainsi valide en remplaçant $\xi_2(m)$ par $\frac{(m+1)(m+2)}{2}$, ce qui donne le présent corollaire. ■

La fonction $\xi_2(m)$ définie dans le corollaire 6.4.2 semble beaucoup plus complexe à calculer que la fonction $\xi(m)$ définie dans le corollaire 6.1.2, étant définie à partir d'une double sommation au lieu d'une simple sommation. Nous avons cependant observé expérimentalement une égalité très simple entre ces deux fonctions qui rend le calcul de $\xi_2(m)$ aussi simple que celui de $\xi(m)$. Cependant, nous ne possédons pas présentement de preuve analytique de cette égalité, nous la formulons donc sous la forme d'une conjecture.

Conjecture 6.4.4. *Les fonctions $\xi(m)$ et $\xi_2(m)$ sont liées entre elles par l'égalité suivante*

$$\xi_2(m) = m + \xi(m)\,.$$

Deuxième partie

Conception d'algorithmes d'apprentissage

Chapitre 7

Applications pratiques de la borne PAC-Bayes

La borne PAC-Bayes a déjà été utilisée pour justifier à postériori la stratégie de certains algorithmes d'apprentissage. Par exemple, les SVM (voir Cortes et Vapnik (1995); Vapnik (1998), voir également Platt (1999); Cristianini et Shawe-Taylor (2000)) ont été conçus comme un algorithme construisant le classificateur linéaire ayant la plus grande marge normalisée sur les exemples d'entrainement. En voyant un classificateur linéaire $\mathbf{w}$ comme étant un classificateur dit Bayes-équivalent au classificateur de Gibbs associé à une distribution normale centrée en un point dans la direction de $\mathbf{w}$, il devient possible d'utiliser le théorème PAC-Bayes pour obtenir une borne sur le risque d'un classificateur retourné par un SVM (voir Herbrich et Graepel (2001) ainsi que Langford (2005)). Les résultats de ces applications théoriques de la borne PAC-Bayes permettent de justifier partiellement l'idée de maximiser la marge de classification.

Une autre idée d'utilisation pratique de la borne PAC-Bayes a été d'utiliser celle-ci dans le but d'effectuer de la sélection de paramètres (voir Ambroladze *et al.*, 2004, 2006). L'idée se fonde sur l'observation suivante : certains algorithmes d'apprentissage demandent pour leur exécution qu'on leur fournisse un jeu d'hyperparamètres (par exemple, deux valeurs réelles, C et γ, pour le SVM avec noyau RBF), la validation croisée est couramment utilisée pour choisir, lors d'une application, le «meilleur» jeu d'hyperparamètres. Or, la validation croisée est très couteuse en termes de temps de calculs. Une autre approche est de calculer une borne sur le risque des classificateurs obtenus avec les différents jeux d'hyperparamètre testés : on choisit alors pour construire le classificateur final, le jeu d'hyperparamètres ayant permis d'obtenir la plus faible borne.

Nous présentons dans cette partie de la thèse, des algorithmes d'apprentissage construisant des classificateurs par vote de majorité de classificateurs de base, provenant d'un ensemble $\mathcal{H}$, ces algorithmes choisirons pour pondérer le vote de majorité la distribution sur $\mathcal{H}$ minimisant une borne de type PAC-Bayes. Ainsi, la borne PAC-Bayes n'est pas ici utilisée pour partiellement justifier un choix de conception, ou pour effectuer de la sélection de paramètres sur un algorithme quelconque : elle est au coeur même de la conception de nos algorithmes. Avant d'aborder nos algorithmes, qui sont présentés dans les chapitres 8 et 9, nous présentons brièvement quelques pistes que nous avons explorées pour concevoir des algorithme inspirés de la borne PAC-Bayes.

7.1 Minimisation de la borne du corollaire 6.2.1

Il est théoriquement possible de trouver la distribution Q minimisant la borne du corollaire 6.2.1. Le théorème suivant donne une expression de cette distribution.

Théorème 7.1.1. *Soit $\mathcal{H}$ un ensemble de classificateurs binaires et P une distribution à priori sur $\mathcal{H}$. Alors, la distribution Q minimisant la borne du corollaire 6.2.1 est donnée par*

$$Q(h) = \frac{1}{Z} P(h) e^{-C \cdot m R_S(h)} .$$

Démonstration : Nous donnons la démonstration dans le cas discret. Dans ce cas, nous pouvons identifier l'ensemble $\mathcal{H}$ par $\mathcal{H} = \{h_1, h_2, \ldots\}$. Pour alléger un peu l'écriture, nous notons respectivement les quantités $P(h_i)$ et $Q(h_i)$ par P_i et Q_i.

La distribution Q minimisant la borne du corollaire 6.2.1 est la même que celle minimisant la quantité $B(Q)$ donnée par

$$B(Q) = C \cdot \sum_{i=1}^{\infty} Q_i R_S(h_i) + \frac{\mathrm{KL}(Q\|P)}{m}$$

sous la contrainte

$$\sum_{i=1}^{\infty} Q_i = 1.$$

À l'optimal, Q doit satisfaire les conditions de Lagrange, c'est-à-dire qu'il doit exister $\lambda \in \mathbf{R}$ tel que pour tout $i \geq 1$

$$\begin{aligned} \lambda &= \frac{\partial B}{\partial Q_i} \\ &= C \cdot R_S(h_i) + \frac{1}{m}\left(1 + \log \frac{Q_i}{P_i}\right). \end{aligned}$$

Il suit que

$$\begin{aligned} Q_i &= P_i e^{m(\lambda - C\cdot R_S(h_i))-1} \\ &= P_i e^{m\lambda-1} e^{-C\cdot m R_S(h_i)} \\ &= \frac{1}{Z} P_i e^{-C\cdot m R_S(h_i)}\,, \end{aligned}$$

où $Z = \sum_{i=1}^{\infty} P_i e^{-C\cdot m R_S(h_i)}$ est une constante de normalisation. ■

La distribution Q s'obtient donc en fonction de la distribution P et du risque empirique individuel de chacun des classificateurs formant le vote. Pour un ensemble $\mathcal{H}$ fini (et de taille raisonnable) de classificateurs , il est alors facile de trouver la distribution Q minimisant la borne du corollaire 6.2.1. Cependant, dans le cas plus général où nous voulons minimiser non pas directement le risque de Gibbs, mais le risque associé à une fonction de perte générale (théorème 6.3.1), il devient nécessaire également de considérer les classificateurs de $\mathcal{H}^k$ pour certaines valeurs de k (dépendant de la fonction de perte). Le nombre de classificateurs à considérer devient alors possiblement très grand, voire infini, comme dans le cas de la perte exponentielle (et ce même si la taille de $\mathcal{H}$ est petite).

Il demeure toutefois possible d'utiliser des méthodes telles que l'algorithme du *Metropolis-Hastings* (présenté initialement dans Metropolis *et al.*, 1953; Hastings, 1970) pour approcher la distribution Q ; de cette façon, il n'y a alors pas de problème à étendre la méthode à l'utilisation de noyaux. Cette idée, qui demeure potentiellement prometteuse, n'a pas apporté à ce jour de résultats probants.

7.2 Minimisation de bornes utilisant des distributions paramétrées

La première approche que nous avons explorée pour concevoir un algorithme d'apprentissage construisant des votes de majorité pondérés par une distribution minimisant une borne de type PAC-Bayes, fut de restreindre les distributions à priori P et à postériori Q à des distributions normales univariées. Cette approche possède les avantages suivants :

– il est possible d'exprimer le risque de Gibbs empirique associé à la distribution Q par une expression analytique simple (voir par exemple Langford, 2005) ;

– il est facile d'intégrer à l'algorithme l'utilisation de noyaux (polynomial, rbf, ...), tout comme pour les SVM ;
– le problème d'optimisation que doit résoudre l'algorithme (qui consiste simplement à trouver le vecteur $\mathbf{w}$ étant le meilleur candidat pour définir le centre de la distribution normale Q) est un problème d'optimisation sans contrainte (le vecteur $\mathbf{w}$ pouvant être n'importe quel vecteur de $\mathbf{R}^n$).

Cette approche a conduit à la publication d'un article dans la conférence ICML (voir Germain *et al.*, 2009a) et se trouve au coeur du mémoire de maitrise de Pascal Germain (voir Germain, 2009). Nous référons à ces publications le lecteur intéressé par cette approche.

7.3 Minimisation de bornes sur des votes de majorité de classificateurs simples

L'idée de combiner plusieurs classificateurs de sorte à former un classificateur par vote de majorité a déjà été étudiée par plusieurs chercheurs (voir par exemple Chen *et al.*, 1997; Kittler *et al.*, 1998) ; et parmi ces études, plusieurs ont mené à l'élaboration de nouveaux algorithmes d'apprentissage, certains ayant eu des répercussions scientifiques importantes ou ayant conduit à des applications commerciales, on peut citer par exemple :

Bagging : cet algorithme, développé par Breiman (voir Breiman, 1996; Quinlan, 1996), utilise la méthode du *Bootstrap* pour simuler la distribution D ayant généré les données d'apprentissage, puis construit un classificateur par vote de majorité formé des différents classificateurs obtenus en tentant de minimiser le risque sur chacune des simulations.

Random Forests : cet algorithme, également développé par Breiman (voir Breiman, 2001), construit un vote de majorité d'arbres de décision. Cette méthode a conduit au développement d'une application commerciale.

Adaboost : cet algorithme, développé par Freund et Shapire (voir Freund et Schapire (1995, 1996), voir également Meir et Rätsch (2003)), construit un classificateur par vote de majorité en sélectionnant dans un premier temps le classificateur de l'ensemble de base ayant le plus petit risque empirique, ensuite, l'importance des données dans le calcul du risque empirique est modifiée de sorte à donner plus de poids aux données mal classifiées et le classificateur minimisant ce nouveau risque empirique est sélectionné. Ce processus est répété jusqu'à l'atteinte d'un critère d'arrêt. Différentes fonctions de perte peuvent être utilisées pour calculer le risque empirique, chacune menant à une variante de l'algorithme, par exemple

l'utilisation de la perte exponentielle mène à la variante du *boosting* nommée *AdaBoost*.

L'idée que nous détaillons dans les chapitres suivants s'inspire de ces différents algorithmes d'apprentissage et de la théorie PAC-Bayes. À partir d'un ensemble fini $\mathcal{H}$ de classificateurs de base (dans nos applications il s'agira de souches de décision, qui consistent simplement en des arbres de décision à une seule couche), nos algorithmes construisent un classificateur par vote de majorité en pondérant les classificateurs de $\mathcal{H}$ (qui forment le vote) par une distribution Q choisie de sorte à minimiser une borne de type PAC-Bayes. Il s'agit donc de la même idée que celle brièvement présentée à la section 7.2, mais cette fois aucune contrainte n'est imposée à la distribution Q (chapitre 8), cependant, alors que la précédente approche s'applique à un ensemble continu de classificateurs, nous sommes maintenant contraint à un ensemble fini. Par la suite, nous explorons une variante de cette approche (chapitre 9) dans laquelle nous contraignons la distribution Q à être une distribution *quasi-uniforme* (terme que nous définissons au chapitre 9), cette approche nous permet de réduire la complexité du problème sans pour autant réduire la puissance des classificateurs générés.

Chapitre 8

Vote de majorité sur des ensembles finis de classificateurs

Nous présentons, dans ce chapitre, un algorithme d'apprentissage générique permettant de construire des classificateurs par vote de majorité pondéré par des distributions minimisant des bornes de type PAC-Bayes sur des fonctions de perte générales. Nous présentons deux versions de l'algorithme, l'une basée sur le théorème 5.1.2 et l'autre sur le théorème 6.3.1, et nous implémentons ces algorithmes avec les fonctions de perte linéaire (à des fins démonstratives), quadratique, exponentielle, ainsi qu'avec la fonction de perte du classificateur de Gibbs à piges multiples — cette dernière version de l'algorithme est l'objet d'une publication (voir Lacasse *et al.*, 2010). Nous comparons les résultats de nos algorithmes avec ceux obtenus par AdaBoost et par la régression ridge sur des ensembles d'apprentissage provenant principalement de la banque UCI (voir Blake et Merz, 1998).

8.1 Algorithme générique

Dans ce qui suit, $\mathcal{H} = \{h_1, h_2, \ldots, h_n\}$ dénote un ensemble de n classificateurs binaires, alors que Q correspond à une distribution sur cet ensemble $\mathcal{H}$. Nous associons, à une distribution Q donnée, le vecteur $\mathbf{Q}$ de dimension n défini par $\mathbf{Q} = \langle Q_1, Q_2, \ldots, Q_n \rangle$ avec $Q_j = Q(h_j)$ pour $j = 1, 2, \ldots, n$. Le vecteur $\mathbf{Q}$ a donc les propriétés suivantes : $\|\mathbf{Q}\|_1 = 1$ et $Q_j \geq 0$ pour tout $j \in \{1, 2, \ldots, n\}$.

Les algorithmes que nous présentons dans ce chapitre construisent des classificateurs par vote de majorité en choisissant, pour pondérer le vote, la distribution Q étant la plus susceptible, suivant les principes à la base de chacun des algorithmes, de définir le meilleur classificateur par vote de majorité. Ces algorithmes se basent sur des théorèmes de type PAC-Bayes applicables à des fonctions de perte générales. À partir d'un tel théorème et d'une fonction de perte, nous pouvons obtenir une borne sur un risque pouvant être utilisée comme fonction objectif pour définir un problème d'optimisation. Ceci suggère l'algorithme d'apprentissage suivant : trouver la distribution Q minimisant la fonction objectif, notée F_Q, associée à une borne donnée, puis retourner le classificateur par vote de majorité pondéré par Q, c'est-à-dire le classificateur f_Q défini par

$$f_Q(\mathbf{x}) = \operatorname{sgn}\left(\sum_{j=1}^{n} Q_j h_j(\mathbf{x})\right) .$$

En notant $\mathbf{h}(\mathbf{x})$ le vecteur $\langle h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_n(\mathbf{x})\rangle$, nous avons, de façon plus concise

$$f_Q(\mathbf{x}) = \operatorname{sgn}\left(\mathbf{Q} \cdot \mathbf{h}(\mathbf{x})\right) .$$

Pour chacun des algorithmes présentés dans ce chapitre, le principe de minimisation de la fonction objectif F_Q (propre à chaque algorithme) est le même et consiste en un processus itératif travaillant par paires de composantes. À chaque itération, deux classificateurs h_j et h_k sont pigés et un échange de poids optimal d'un classificateur à l'autre est déterminé (soit l'échange permettant de minimiser la fonction objectif). L'algorithme procède ensuite à cet échange de poids et le processus est recommencé avec une autre paire de classificateurs. Le tout s'exécute jusqu'à l'atteinte d'un critère d'arrêt (si la fonction objectif cesse de diminuer ou qu'un nombre maximal d'itérations a été atteint). Il s'agit donc d'algorithmes devant résoudre un problème d'optimisation sous contraintes linéaires (car Q doit demeurer une distribution), le processus ici utilisé, procédant par paires de composantes, est garanti de converger vers le minimum global de la fonction objectif si et seulement si cette fonction est convexe.

Définition 8.1.1. *Pour Q une distribution sur un ensemble $\mathcal{H} = \{h_1, h_2, \ldots, h_n\}$ de classificateurs et pour $\lambda \in [-\min(Q_j, 1-Q_k), \min(1-Q_j, Q_k)]$, on dénote par $Q_\lambda^{j,k}$ la distribution donnée par*

$$Q_\lambda^{j,k}(h_t) = \begin{cases} Q(h_t) + \lambda & \textit{si } t = j \\ Q(h_t) - \lambda & \textit{si } t = k \\ Q(h_t) & \textit{sinon}. \end{cases}$$

À partir de la distribution paramétrée $Q_\lambda^{j,k}$ de la définition 8.1.1, nous pouvons définir des sous-fonctions objectif $F_Q^{j,k}(\lambda)$ comme suit :

$$F_Q^{j,k}(\lambda) \stackrel{\text{déf}}{=} F_{Q_\lambda^{j,k}} .$$

Pour une fonction objectif F_Q donnée, les algorithmes d'apprentissage de ce chapitre procèdent en résolvant une succession de problèmes d'optimisation convexe consistant à trouver le minimum d'une fonction convexe $F_Q^{j,k}(\lambda)$ à une variable réelle. Ces algorithmes correspondent tous à une spécialisation de l'algorithme générique 1.

Algorithme 1 : Algorithme générique

Entrées : $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$, $\mathcal{H} = \{h_1, h_2, \ldots, h_n\}$

Initialiser : $Q_j = 1/n$ pour $j = 1, \ldots, n$

Exécuter

Piger j et k aléatoirement dans l'ensemble $\{1, 2, \ldots, n\}$.

$$\lambda_{opt} \leftarrow \underset{\lambda \in [-\min(Q_j, 1-Q_k), \min(Q_k, 1-Q_j)]}{\operatorname{argmin}} \{F_Q^{j,k}(\lambda)\}$$

$Q_j \leftarrow Q_j + \lambda_{opt}$

$Q_k \leftarrow Q_j - \lambda_{opt}$

Répéter tant que *critère d'arrêt non atteint* ;

Sortie : $f_{\mathbf{Q}}(\mathbf{x}) = \operatorname{sgn}\left(\sum_{j=1}^{n} Q_j h_j(\mathbf{x})\right)$

Fonction objectif découlant d'un théorème PAC-Bayes

Lors des échanges de poids λ entre des classificateurs h_j et h_k dans le déroulement de l'algorithme générique 1, la distribution Q se trouve être modifiée en une nouvelle distribution, que l'on note $Q_\lambda^{j,k}$, dans laquelle tous les poids des classificateurs de $\mathcal{H}$ restent inchangés par rapport à la distribution Q, exceptés $Q_\lambda^{j,k}(h_j)$ et $Q_\lambda^{j,k}(h_k)$ qui sont respectivement donnés par $Q(h_j) + \lambda$ et $Q(h_k) - \lambda$ (voir la définition 8.1.1). Comme chaque distribution $Q_\lambda^{j,k}$ (pour j, k et λ donnés) correspond à une distribution à postériori sur $\mathcal{H}$ et comme le théorème PAC-Bayes (ou l'un de ses dérivés) s'applique pour borner uniformément le risque (de Gibbs) associé à toute distribution à postériori sur $\mathcal{H}$, il suit que la borne PAC-Bayes utilisée dans les implémentations de l'algorithme générique 1 demeurent valide tout au long de leurs exécutions.

Pour chaque théorème de type PAC-Bayes, il est possible de dériver une fonction $F_Q^{j,k}(\lambda)$ pouvant servir de fonction objectif dans l'algorithme générique 1 et ainsi obtenir un nouvel algorithme d'apprentissage. Dans les sections 8.1.1 et 8.1.2, nous présentons deux approches pour construire les fonctions objectif $F_Q^{j,k}(\lambda)$ qui sont basées sur deux versions différentes du théorème PAC-Bayes sur les fonctions de perte générales. Le théorème sur les fonctions de perte générales permet de borner des risques associés à différentes fonctions de perte ; de chacune de ces deux méthodes que nous abordons pour construire $F_Q^{j,k}(\lambda)$, il est possible de concevoir une infinité d'algorithmes d'apprentissage

différents (un pour chaque fonction de perte ayant la forme acceptée par le théorème utilisé).

Dans les chapitres qui suivent, nous analysons quelques algorithmes, chacun basé sur une fonction de perte donnée. Il a fallut faire un choix pour ces fonctions de perte, ce choix s'est arrêté sur les fonctions de perte linéaire (à des fins démonstratives, voir section 8.3), quadratique et exponentielle (étudiée respectivement dans les sections 8.4 et 8.5), et sur la fonction de perte du classificateur de Gibbs à piges multiples (section 8.6).

8.1.1 Minimisation de la borne du théorème 5.1.2

Le théorème PAC-Bayes sur les fonctions de perte générales qui est déduit du théorème PAC-Bayes classique, lorsqu'appliqué aux problèmes d'optimisation que doit résoudre une implémentation de l'algorithme générique 1, prend la forme

$$\Pr_{S\sim D^m}\Bigg(\forall Q \text{ sur } \mathcal{H}\,, \forall \lambda \in I_Q^{j,k}\colon \\ \mathrm{kl}\Big(\widehat{A_Q^{j,k}}(\lambda)\,\Big\|\, A_Q^{j,k}(\lambda)\Big) \leq \frac{1}{m}\Big[k_a \cdot \mathrm{KL}_Q^{j,k}(\lambda) + \log\frac{\xi(m)}{\delta}\Big]\Bigg) \geq\ 1-\delta\,, \quad (8.1)$$

où $I_Q^{j,k}$ est l'intervalle de valeurs possibles pour λ, c'est-à-dire

$$I_Q^{j,k} = [-\min(Q_j, 1-Q_k), \min(Q_k, 1-Q_j)]\ ,$$

$\widehat{A_Q^{j,k}}(\lambda)$ correspond à un risque empirique et $A_Q^{j,k}(\lambda)$ correspond à un vrai risque (chacun défini à partir d'une fonction de perte donnée). Plus précisément, dans l'application du théorème sur les fonctions de perte générales (théorème 5.1.2), $A_Q^{j,k}(\lambda)$ et $\widehat{A_Q^{j,k}}(\lambda)$ prennent la forme

$$A_Q^{j,k}(\lambda) = \frac{1}{2c}\left[\zeta_{Q_\lambda^{j,k}} - g(0)\right] + \frac{1}{2} \quad ; \quad \widehat{A_Q^{j,k}}(\lambda) = \frac{1}{2c}\left[\widehat{\zeta_{Q_\lambda^{j,k}}} - g(0)\right] + \frac{1}{2}\,,$$

pour $\zeta_{Q_\lambda^{j,k}}$ une fonction de risque définie à partir d'une fonction de perte de la forme de l'équation 5.6, avec $Q_\lambda^{j,k}$ la distribution sur $\mathcal{H}$ donnée à la définition 8.1.1.

À noter que $Q_\lambda^{j,k}$ n'est rien d'autre qu'une distribution à postériori sur $\mathcal{H}$, le théorème PAC-Bayes sur les fonctions de perte générales s'applique uniformément pour toute distribution de cette forme, la borne est donc bien valide simultanément pour toute valeur que peuvent prendre les paramètres j, k et λ.

Pour Q donnée et j, k fixés, une borne supérieure de $A_Q^{j,k}(\lambda)$ peut être obtenue à partir de la valeur maximale B telle que

$$\mathrm{kl}\Big(\widehat{A_Q^{j,k}}(\lambda) \,\Big\|\, B\Big) \leq \frac{1}{m}\Big[k_a \cdot \mathrm{KL}_Q^{j,k}(\lambda) + \log\frac{\xi(m)}{\delta}\Big]\,,$$

où $\mathrm{KL}_Q^{j,k}(\lambda)$ correspond à la divergence de Kullback-Leibler entre $Q_\lambda^{j,k}$ et P (la distribution à priori), c'est-à-dire

$$\begin{aligned}\mathrm{KL}_Q^{j,k}(\lambda) &\stackrel{\text{déf}}{=} \mathrm{KL}(Q_\lambda^{j,k}\|P) \\ &= \mathrm{KL}(Q\|P) + (Q_j+\lambda)\log(Q_j+\lambda) + (Q_k-\lambda)\log(Q_k-\lambda) \\ &\quad -Q_j\log Q_j - Q_k\log Q_k\,.\end{aligned} \tag{8.2}$$

Cette application du théorème PAC-Bayes mène à la fonction objectif $F_Q^{j,k}$ suivante dans l'algorithme générique 1

$$F_Q^{j,k}(\lambda) = \max\left\{B : \mathrm{kl}\Big(\widehat{A_Q^{j,k}}(\lambda) \,\Big\|\, B\Big) \leq \frac{1}{m}\Big[k_a \cdot \mathrm{KL}_Q^{j,k}(\lambda) + \log\frac{\xi(m)}{\delta}\Big]\right\}\,.$$

Comme l'égalité dans le max ci-haut est atteinte, nous avons

$$\mathrm{kl}\Big(\widehat{A_Q^{j,k}}(\lambda) \,\Big\|\, F_Q^{j,k}(\lambda)\Big) - \frac{1}{m}\Big[k_a \cdot \mathrm{KL}_Q^{j,k}(\lambda) + \log\frac{\xi(m)}{\delta}\Big] \equiv 0\,.$$

Pour une fonction $\widehat{A_Q^{j,k}}(\lambda)$ dérivable et convexe, nous pouvons transformer le problème de recherche du minimum de $F_Q^{j,k}(\lambda)$ en un problème de recherche de l'unique zéro de sa dérivée. En calculant les dérivées, on trouve

$$\begin{aligned}\frac{\partial}{\partial\lambda}\mathrm{kl}\left(\widehat{A_Q^{j,k}}(\lambda)\|F_Q^{j,k}(\lambda)\right) &= \frac{\partial\widehat{A_Q^{j,k}}(\lambda)}{\partial\lambda}\cdot\left(\frac{\widehat{A_Q^{j,k}}(\lambda)(1-F_Q^{j,k}(\lambda))}{F_Q^{j,k}(\lambda)(1-\widehat{A_Q^{j,k}}(\lambda))}\right) \\ &\quad + \frac{\partial F_Q^{j,k}(\lambda)}{\partial\lambda}\cdot\left(\frac{F_Q^{j,k}(\lambda)-\widehat{A_Q^{j,k}}(\lambda)}{F_Q^{j,k}(\lambda)(1-F_Q^{j,k}(\lambda))}\right) \\ \frac{\partial}{\partial\lambda}\mathrm{KL}_Q^{j,k}(\lambda) &= \log\left(\frac{Q_j+\lambda}{Q_k-\lambda}\right)\,.\end{aligned}$$

Il suit que l'on doit avoir l'égalité

$$\begin{aligned}0 \equiv \frac{\partial\widehat{A_Q^{j,k}}(\lambda)}{\partial\lambda}\cdot\log\left(\frac{\widehat{A_Q^{j,k}}(\lambda)(1-F_Q^{j,k}(\lambda))}{F_Q^{j,k}(\lambda)(1-\widehat{A_Q^{j,k}}(\lambda))}\right) \\ + \frac{\partial F_Q^{j,k}(\lambda)}{\partial\lambda}\cdot\left(\frac{F_Q^{j,k}(\lambda)-\widehat{A_Q^{j,k}}(\lambda)}{F_Q^{j,k}(\lambda)(1-F_Q^{j,k}(\lambda))}\right) - \frac{k_a}{m}\log\left(\frac{Q_j+\lambda}{Q_k-\lambda}\right)\,,\end{aligned}$$

ce qui donne pour $\dfrac{\partial F_Q^{j,k}(\lambda)}{\partial \lambda}$:

$$\frac{\partial F_Q^{j,k}(\lambda)}{\partial \lambda} = \frac{\frac{k_a}{m}\log\left(\frac{Q_j+\lambda}{Q_k-\lambda}\right) - \frac{\partial \widehat{A_Q^{j,k}}(\lambda)}{\partial \lambda}\cdot\log\left(\frac{\widehat{A_Q^{j,k}}(\lambda)(1-F_Q^{j,k}(\lambda))}{F_Q^{j,k}(\lambda)(1-\widehat{A_Q^{j,k}}(\lambda))}\right)}{\left(\frac{F_Q^{j,k}(\lambda)-\widehat{A_Q^{j,k}}(\lambda)}{F_Q^{j,k}(\lambda)(1-F_Q^{j,k}(\lambda))}\right)}\,.$$

L'expression de droite est égale à zéro si et seulement si son numérateur est égal à zéro, c'est-à-dire si et seulement si la valeur de λ est telle que

$$\frac{\partial \widehat{A_Q^{j,k}}(\lambda)}{\partial \lambda}\cdot\log\left(\frac{\widehat{A_Q^{j,k}}(\lambda)(1-F_Q^{j,k}(\lambda))}{F_Q^{j,k}(\lambda)(1-\widehat{A_Q^{j,k}}(\lambda))}\right) - \frac{k_a}{m}\log\left(\frac{Q_j+\lambda}{Q_k-\lambda}\right) = 0\,. \tag{8.3}$$

À noter que l'on a l'inégalité $F_Q^{j,k}(\lambda) > 0$ et si $\widehat{A_Q^{j,k}}(\lambda) < 1$, alors on a également $F_Q^{j,k}(\lambda) > \widehat{A_Q^{j,k}}(\lambda)$ et $F_Q^{j,k}(\lambda) < 1$. Donc, le dénominateur principal dans l'expression de droite ne peut être égal à zéro que dans le cas extrême où $\widehat{A_Q^{j,k}}(\lambda) = 1$.

Ainsi, le problème d'optimisation consistant à minimiser $F_Q^{j,k}(\lambda)$ pour des valeurs données de j et k est équivalent à résoudre l'équation 8.3.

Théorème PAC-Bayes pour les fonctions de perte valant $\frac{1}{2}$ en $W_Q = \frac{1}{2}$

Dans ce cas, nous voulons minimiser le risque associé à une fonction de perte ζ_Q de la forme de l'équation 5.1. Pour mettre l'emphase sur le fait qu'à chaque itération de l'algorithme (c'est-à-dire avec j, k et Q fixés) nous minimisons le risque en fonction de la variable λ, nous notons $\zeta_Q^{j,k}(\lambda)$ le risque pour la distribution $Q_\lambda^{j,k}$ (voir la définition 8.1.1), nous avons donc

$$\zeta_Q^{j,k}(\lambda) \stackrel{\text{déf}}{=} \zeta_{Q_\lambda^{j,k}}\,. \tag{8.4}$$

Nous trouvons comme valeurs de $\widehat{A_Q^{j,k}}(\lambda)$ et de sa dérivée (voir théorème 5.1.1)

$$\widehat{A_Q^{j,k}}(\lambda) = \frac{1}{c_a}\left[\widehat{\zeta_Q^{j,k}}(\lambda) - \frac{1}{2}\right] + \frac{1}{2}$$
$$\frac{\partial \widehat{A_Q^{j,k}}(\lambda)}{\partial \lambda} = \frac{1}{c_a}\frac{\partial \widehat{\zeta_Q^{j,k}}(\lambda)}{\partial \lambda}\,.$$

En portant ces deux quantités dans l'équation 8.3, l'équation à résoudre devient maintenant

$$\frac{1}{c_a}\frac{\partial\widehat{\zeta_Q^{j,k}}(\lambda)}{\partial\lambda}\cdot\log\left(\frac{\left(\frac{1}{c_a}\left[\widehat{\zeta_Q^{j,k}}(\lambda)-\frac{1}{2}\right]+\frac{1}{2}\right)\left(1-F_Q^{j,k}(\lambda)\right)}{F_Q^{j,k}(\lambda)\left(\frac{1}{2}-\frac{1}{c_a}\left[\widehat{\zeta_Q^{j,k}}(\lambda)-\frac{1}{2}\right]\right)}\right)-\frac{k_a}{m}\log\left(\frac{Q_j+\lambda}{Q_k-\lambda}\right)=0\,,$$

où

$$F_Q^{j,k}(\lambda)\stackrel{\text{déf}}{=}\max\left\{B:\mathrm{kl}\left(\frac{1}{c_a}\left[\widehat{\zeta_Q^{j,k}}(\lambda)-\frac{1}{2}\right]+\frac{1}{2}\middle\|B\right)\leq\frac{1}{m}\left[k_a\cdot\mathrm{KL}_Q^{j,k}(\lambda)+\log\frac{\xi(m)}{\delta}\right]\right\}.$$

En effectuant quelques simplifications, nous obtenons l'expression

$$\frac{\partial\widehat{\zeta_Q^{j,k}}(\lambda)}{\partial\lambda}\cdot\log\left(\frac{\left(2\widehat{\zeta_Q^{j,k}}(\lambda)-1+c_a\right)\left(1-F_Q^{j,k}(\lambda)\right)}{F_Q^{j,k}(\lambda)\left(c_a-2\widehat{\zeta_Q^{j,k}}(\lambda)+1\right)}\right)-\frac{c_a\cdot k_a}{m}\log\left(\frac{Q_j+\lambda}{Q_k-\lambda}\right)=0\,.$$

Théorème PAC-Bayes pour les fonctions de perte valant 1 en $W_Q=\frac{1}{2}$

Dans ce cas, nous voulons minimiser le risque associé à une fonction de perte ζ_Q de la forme de l'équation 5.6 avec $g(0)=1$. Nous trouvons comme valeurs de $\widehat{A_Q^{j,k}}(\lambda)$ et de sa dérivée (voir théorème 5.1.2)

$$\widehat{A_Q^{j,k}}(\lambda)=\frac{\widehat{\zeta_Q^{j,k}}(\lambda)-1}{2c_a}+\frac{1}{2} \tag{8.5}$$

$$\frac{\partial\widehat{A_Q^{j,k}}(\lambda)}{\partial\lambda}=\frac{1}{2c_a}\frac{\partial\widehat{\zeta_Q^{j,k}}(\lambda)}{\partial\lambda} \tag{8.6}$$

En plaçant ces valeurs dans l'équation 8.3, nous trouvons comme équation à résoudre

$$\frac{1}{2c_a}\frac{\partial\widehat{\zeta_Q^{j,k}}(\lambda)}{\partial\lambda}\cdot\log\left(\frac{\left(\frac{\widehat{\zeta_Q^{j,k}}(\lambda)-1}{2c_a}-\frac{1}{2}\right)\left(1-F_Q^{j,k}(\lambda)\right)}{F_Q^{j,k}(\lambda)\left(\frac{1}{2}-\frac{\widehat{\zeta_Q^{j,k}}(\lambda)-1}{2c_a}\right)}\right)-\frac{k_a}{m}\log\left(\frac{Q_j+\lambda}{Q_k-\lambda}\right)=0\,,$$

où

$$F_Q^{j,k}(\lambda)\stackrel{\text{déf}}{=}\max\left\{B:\mathrm{kl}\left(\frac{1}{2c_a}\left[\widehat{\zeta_Q^{j,k}}(\lambda)-1\right]+\frac{1}{2}\middle\|B\right)\leq\frac{1}{m}\left[k_a\cdot\mathrm{KL}_Q^{j,k}(\lambda)+\log\frac{\xi(m)}{\delta}\right]\right\}.$$

En effectuant quelques simplifications, nous obtenons l'expression

$$\frac{\partial\widehat{\zeta_Q^{j,k}}(\lambda)}{\partial\lambda}\cdot\log\left(\frac{\left(\widehat{\zeta_Q^{j,k}}(\lambda)-1+c_a\right)\left(1-F_Q^{j,k}(\lambda)\right)}{F_Q^{j,k}(\lambda)\left(c_a-\widehat{\zeta_Q^{j,k}}(\lambda)+1\right)}\right)-\frac{2c_a\cdot k_a}{m}\log\left(\frac{Q_j+\lambda}{Q_k-\lambda}\right)=0\,. \tag{8.7}$$

8.1.2 Fonction de perte générale (Théorème PAC-Bayes version Catoni)

Le théorème PAC-Bayes sur les fonctions de perte générales de la forme de l'équation 5.6, c'est-à-dire de la forme

$$\zeta_Q(\mathbf{x}, y) = g(0) + \sum_{k=1}^{\infty} g(k) \left(2W_Q(\mathbf{x}, y) - 1\right)^k ,$$

qui est déduit du théorème PAC-Bayes version Catoni (théorème 5.1.2), lorsqu'appliqué aux problèmes d'optimisation que doit résoudre une implémentation de l'algorithme générique 1, prend la forme suivante (pour C' et δ des quantités fixées à priori) :

$$\Pr_{S\sim D^m}\left(\forall Q \text{ sur } \mathcal{H}\,, \forall \lambda \in I_Q^{j,k} \colon \zeta_Q^{j,k}(\lambda) \leq 2c_a\left(M_Q^{j,k}(\lambda) - \frac{1}{2}\right) + g(0)\right) \;\geq\; 1-\delta\,, \quad (8.8)$$

où

$$M_Q^{j,k}(\lambda) \;=\; \frac{1}{1-e^{-C'}}\left\{1 - \exp\left[-\left(C' \cdot \left[\frac{1}{2c_a}\left(\widehat{\zeta_Q^{j,k}}(\lambda) - g(0)\right) + \frac{1}{2}\right] + \frac{1}{m}\left(k_a \cdot \mathrm{KL}_Q^{j,k}(\lambda) + \log\frac{1}{\delta}\right)\right)\right]\right\},$$

avec

$$c_a \stackrel{\text{déf}}{=} \sum_{k=1}^{\infty} |g(k)| \quad \text{et} \quad k_a \stackrel{\text{déf}}{=} \frac{1}{c_a}\sum_{k=1}^{\infty} k \cdot |g(k)|\,.$$

Dans le cas des fonctions de perte ζ_Q telles que $\zeta_Q(\mathbf{x}, y) = 1$ lorsque $W_Q(\mathbf{x}, y) = \frac{1}{2}$, soit le cas qui nous a intéressé dans la plupart de nos applications de cette version du théorème, la quantité $g(0)$ est simplement égale à 1. Pour rappel, $\zeta_Q^{j,k}(\lambda)$ et $\mathrm{KL}_Q^{j,k}(\lambda)$ sont respectivement définis aux équations 8.4 et 8.2, et finalement, $I_Q^{j,k}$ est l'intervalle de valeurs possibles pour λ, c'est-à-dire

$$I_Q^{j,k} = [-\min(Q_j, 1-Q_k), \min(Q_k, 1-Q_j)]\ .$$

Pour minimiser la fonction $\zeta_Q^{j,k}(\lambda)$, nous devons trouver la valeur λ permettant de minimiser la quantité

$$2c_a\left(M_Q^{j,k}(\lambda) - \frac{1}{2}\right) + g(0)\,.$$

Minimiser cette fonction est équivalent à simplement minimiser $M_Q^{j,k}(\lambda)$, et une petite analyse de cette dernière révèle que la valeur λ recherchée est celle minimisant la quantité

$$F_Q^{j,k}(\lambda) = C \cdot m \cdot \widehat{\zeta_Q^{j,k}}(\lambda) + \mathrm{KL}_Q^{j,k}(\lambda)\,, \quad (8.9)$$

où $C = \dfrac{C'}{2 \cdot c_a \cdot k_a}$.

Pour une fonction $\widehat{\zeta_Q^{j,k}}(\lambda)$ dérivable et convexe, nous pouvons transformer le problème de trouver le point λ optimal en un problème de recherche du zéro de la dérivée de $\widehat{\zeta_Q^{j,k}}(\lambda)$. En effet, dans ces conditions, trouver la valeur λ minimisant la quantité donnée à l'équation 8.9 est équivalent à trouver l'unique solution de l'équation $\partial F_Q^{j,k}/\partial\lambda = 0$, et donc l'unique solution de l'équation

$$C \cdot m \cdot \frac{\partial \widehat{\zeta_Q^{j,k}}(\lambda)}{\partial \lambda} + \log\left(\frac{Q_j + \lambda}{Q_k - \lambda}\right) = 0\,. \tag{8.10}$$

Fonctions de perte passant par $\frac{1}{2}$

Le théorème 5.1.1 donne une formulation particulière pour les fonctions de perte ζ_Q valant $\frac{1}{2}$ pour les exemples $(\mathbf{x}, y)$ tels que $W_Q(\mathbf{x}, y) = \frac{1}{2}$ et représentées sous la forme de l'équation (5.1), c'est-à-dire sous la forme

$$\zeta_Q(\mathbf{x}, y) = \frac{1}{2} + \frac{1}{2}\sum_{k=1}^{\infty} g(k)\,(2W_Q(\mathbf{x}, y) - 1)^k\,.$$

Cette version du théorème sur les fonctions de perte générales amène exactement la même fonction objectif que la version utilisée précédemment, c'est-à-dire

$$F_Q^{j,k}(\lambda) = C \cdot m \cdot \widehat{\zeta_Q^{j,k}}(\lambda) + \mathrm{KL}_Q^{j,k}(\lambda)\,,$$

où la constante C doit être préalablement fixée. La borne reliée à cette application du théorème prend la forme suivante (pour δ préalablement fixé)

$$\Pr_{S\sim D^m}\left(\forall Q \text{ sur } \mathcal{H}\,, \forall \lambda \in I_Q^{j,k} \colon \zeta_Q^{j,k}(\lambda) \leq c_a\left(M_Q^{j,k}(\lambda) - \frac{1}{2}\right) + \frac{1}{2}\right) \;\geq\; 1 - \delta\,, \tag{8.11}$$

où

$$M_Q^{j,k}(\lambda) \;=\; \frac{1}{1 - e^{-C'}}\left\{1 - \exp\left[-\left(C' \cdot \left[\frac{1}{c_a}\left(\widehat{\zeta_Q^{j,k}}(\lambda) - \frac{1}{2}\right) + \frac{1}{2}\right] + \frac{1}{m}\left(k_a \cdot \mathrm{KL}_Q^{j,k}(\lambda) + \log\frac{1}{\delta}\right)\right)\right]\right\},$$

avec

$$c_a \overset{\text{déf}}{=} \sum_{k=1}^{\infty} |g(k)| \quad \text{et} \quad \overline{k} \overset{\text{déf}}{=} \frac{1}{c_a}\sum_{k=1}^{\infty} k \cdot |g(k)|$$

et les constantes C et C' sont reliées par l'égalité $C = \dfrac{C'}{c_a \cdot k_a}$.

8.1.3 Méthode de Newton

Lorsque la fonction de risque ζ_Q est appropriée, par exemple si elle est convexe et que sa dérivée est facilement calculable (ce qui est le cas pour la majorité des fonctions que nous avons étudiées), il est possible d'utiliser la méthode de Newton pour résoudre les équations des sections précédentes, c'est-à-dire la résolution de l'équation $\widetilde{F}_Q^{j,k}(\lambda) = 0$ où, pour le cas du théorème version Langford-Seeger (voir l'équation 8.3) :

$$\widetilde{F}_Q^{j,k}(\lambda) \stackrel{\text{déf}}{=} \frac{\partial \widehat{A_Q^{j,k}}(\lambda)}{\partial \lambda} \cdot \log \left(\frac{\widehat{A_Q^{j,k}}(\lambda)(1 - F_Q^{j,k}(\lambda)}{F_Q^{j,k}(\lambda)(1 - \widehat{A_Q^{j,k}}(\lambda))} \right) - \frac{k_a}{m} \log \left(\frac{Q_j + \lambda}{Q_k - \lambda} \right) , \tag{8.12}$$

où $\widehat{A_Q^{j,k}}(\lambda)$ dépend de la fonction de perte et de la version du théorème général utilisées (voir section 8.1.1). Pour le cas du théorème version Catoni (voir l'équation 8.10) nous avons

$$\widetilde{F}_Q^{j,k}(\lambda) \stackrel{\text{déf}}{=} C \cdot m \cdot \frac{\partial \widehat{\zeta_Q^{j,k}}(\lambda)}{\partial \lambda} + \log \left(\frac{Q_j + \lambda}{Q_k - \lambda} \right) . \tag{8.13}$$

Pour l'application de la méthode de Newton, il est nécessaire de calculer la dérivée première de $\widetilde{F}_Q^{j,k}(\lambda)$. On trouve, pour la version Langford-Seeger :

$$\begin{aligned} \frac{\partial \widetilde{F}_Q^{j,k}(\lambda)}{\partial \lambda} \quad = \quad & \frac{k_a}{m} \cdot \frac{(Q_j + Q_k)}{(Q_j + \lambda)(\lambda - Q_k)} \\ & + \frac{\partial \widehat{A_Q^{j,k}}(\lambda)}{\partial \lambda} \left(\frac{\partial \widehat{A_Q^{j,k}}(\lambda)}{\partial \lambda} \cdot \frac{1}{\widehat{A_Q^{j,k}}(\lambda) - \left(\widehat{A_Q^{j,k}}(\lambda) \right)^2} \right. \\ & \qquad \left. + \frac{\partial F_Q^{j,k}(\lambda)}{\partial \lambda} \cdot \frac{1}{F_Q^{j,k}(\lambda)(F_Q^{j,k}(\lambda) - 1)} \right) \\ & + \frac{\partial^2 \widehat{A_Q^{j,k}}(\lambda)}{\partial \lambda^2} \log \left(\frac{\widehat{A_Q^{j,k}}(\lambda)(F_Q^{j,k}(\lambda) - 1)}{F_Q^{j,k}(\lambda)(\widehat{A_Q^{j,k}}(\lambda) - 1)} \right) \end{aligned} \tag{8.14}$$

et pour la version Catoni :

$$\frac{\partial \widetilde{F}_Q^{j,k}(\lambda)}{\partial \lambda} = C \cdot m \cdot \frac{\partial^2 \widehat{\zeta_Q^{j,k}}(\lambda)}{\partial \lambda^2} + \frac{(Q_j + Q_k)}{(Q_j + \lambda)(\lambda - Q_k)} . \tag{8.15}$$

L'algorithme 2 présente une version de l'algorithme générique 1 dans laquelle la méthode de Newton est utilisée pour minimiser la fonction objectif. Dans cet algorithme, la fonction $\widetilde{F}_Q^{j,k}$ peut soit correspondre à une fonction du type des fonctions

Algorithme 2 : Algorithme générique implémentant la méthode de Newton

Entrées : $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$, $\mathcal{H} = \{h_1, h_2, \ldots, h_n\}$

Initialiser : $Q_j = 1/n$ pour $j = 1, 2, \ldots, n$

Exécuter

Piger j et k aléatoirement dans l'ensemble $\{1, 2, \ldots, n\}$.

Poser $\lambda_{opt} = 0$.

Exécuter

$$\lambda_{tmp} \leftarrow \lambda_{opt}$$

$$\lambda_{opt} \leftarrow \lambda_{opt} - \frac{\widetilde{F}_Q^{j,k}(\lambda_{opt})}{\frac{\partial \widetilde{F}_Q^{j,k}}{\partial \lambda}(\lambda_{opt})}$$

Répéter tant que $|\lambda_{opt} - \lambda_{tmp}| >$ *précision voulue* ;

Si $\lambda_{opt} < -\min(Q_j, 1 - Q_k)$ **alors**

$\lambda_{opt} \leftarrow -\min(Q_j, 1 - Q_k)$

FinSi

Si $\lambda_{opt} > \min(Q_k, 1 - Q_j)$ **alors**

$\lambda_{opt} \leftarrow \min(Q_k, 1 - Q_j)$

FinSi

$$Q_j \leftarrow Q_j + \lambda_{opt}$$

$$Q_k \leftarrow Q_j - \lambda_{opt}$$

Répéter tant que *critère d'arrêt non atteint* ;

des équations 8.3 et 8.10 (provenant du théorème sur les fonctions de perte générales en version Langford-Seeger ou Catoni), soit correspondre à la dérivée d'une quelconque fonction objectif associée à une fonction de risque (qui doit être de préférence dérivable et convexe).

À noter que le choix initial $\lambda_{opt} = 0$ de l'algorithme n'est pas arbitraire. En effet, puisqu'à chaque nouvelle itération de l'algorithme la fonction objectif s'approche davantage d'un minimum, les valeurs de λ_{opt} trouvées doivent tendre vers zéro. Donc, à chaque nouvelle itération de l'algorithme, la méthode de Newton commence sur un point s'approchant de plus en plus du zéro de la fonction $\widetilde{F}_Q^{j,k}$; à la limite, lorsque la convergence est atteinte avec la précision voulue, le choix initial $\lambda_{opt} = 0$ est optimal.

8.1.4 Échange de poids style AdaBoost

Dans les algorithmes 1 et 2 présentés dans ce chapitre, nous considérons seulement des échanges de poids pouvant se faire entre deux classificateurs formant une paire choisie aléatoirement. Il est cependant possible de procéder autrement, nous pouvons

par exemple rechercher un échange de poids simulant d'une certaine façon l'approche utilisée dans AdaBoost, c'est-à-dire en pigeant un seul classificateur h_k et en recherchant l'échange de poids optimal $\lambda \in [-Q_k, 1 - Q_k]$ qu'il est possible de faire entre ce classificateur h_k et l'ensemble des autres classificateurs de $\mathcal{H}$. Après avoir trouvé cette valeur λ, la distribution Q peut alors être mise à jour de la façon suivante :

$$\begin{aligned} Q_k &\leftarrow Q_k + \lambda \\ Q_j &\leftarrow Q_j - \lambda \frac{Q_j}{1 - Q_k} \quad \forall j \in \{1, 2, \ldots, n\}, j \neq k \,. \end{aligned}$$

Nous n'avons cependant pas expérimenté cette approche de conception de l'algorithme.

8.2 Méthodologie des expérimentations

Avant de présenter les différents algorithmes d'apprentissage que nous avons conçus en nous basant sur l'algorithme générique 1, nous présentons la méthodologie que nous avons utilisée pour comparer les algorithmes.

Les algorithmes d'apprentissage avec lesquels nous avons travaillé construisent des classificateurs par vote de majorité à partir d'un ensemble de classificateurs de base. Dans toutes nos expérimentations, nous avons utilisé des souches de décision comme classificateurs de base. Ces classificateurs dépendent chacun d'un seul attribut du vecteur de caractéristiques, ainsi que d'un seuil donné. Pour un vecteur de caractéristiques $\mathbf{x} = \langle x_1, x_2, \ldots, x_N \rangle$ donné, la souche de décision $h_{k,t,b}$ correspond au classificateur suivant

$$h_{k,t,b}(\mathbf{x}) = \begin{cases} +b & \text{si } x_k > t \\ -b & \text{sinon} \,, \end{cases}$$

où $b \in \{+1, -1\}$ et $t \in \mathbf{R}$.

Pour chaque jeu de données que nous avons testé, l'ensemble de souches de décision constituant l'ensemble de classificateurs de base a été conçu comme suit : pour chaque composante des vecteurs de caractéristiques, nous avons construit 10 souches de décision de la forme $h_{k,t,1}$, où t prend 10 valeurs uniformément réparties entre $x_k^{\min}$ et $x_k^{\max}$, où $x_k^{\min}$ est la plus petite valeur que peut prendre le k^{e} attribut et $x_k^{\max}$ la plus grande. Pour un jeu de données dont les vecteurs de caractéristiques possèdent N attributs, l'ensemble $\mathcal{H}$ contient donc $2 \cdot 10 \cdot N$ souches de décision.

Pour comparer les algorithmes entre eux, nous avons sélectionné 21 ensembles de données. La majorité de ces ensembles proviennent de la banque UCI (voir Blake et Merz, 1998). Nous avons séparé aléatoirement chaque ensemble de données en deux ensembles : l'un constituant un ensemble d'entrainement, noté S, et l'autre un ensemble test, noté T. Pour chaque jeu de données, tous les algorithmes sont entrainés avec le même ensemble S et ils utilisent tous le même ensemble de classificateurs de base. En utilisant la méthode dite de l'inversion de la queue de la binomiale (voir Langford (2005) théorèmes 3.3 et 3.9), nous calculons, à l'aide de l'ensemble test T, des bornes inférieures et supérieures sur le vrai risque de chaque classificateur obtenu. Deux classificateur sont considérés statistiquement significativement différents si la borne supérieure sur le vrai risque de l'un (calculée à l'aide de l'ensemble test) est inférieure à la borne inférieure sur le vrai risque de l'autre. Nous disons dans ce cas que le classificateur possédant le plus petit risque empirique est statistiquement significativement meilleur (SSM) que celui possédant le plus grand risque. Le taux de confiance pour le calcul des bornes a été fixé à 95%, ainsi les bornes inférieures et supérieures sont valides avec probabilité d'au moins 90%.

8.3 Risque linéaire

La première fonction de perte que nous examinons est la simple fonction

$$\zeta_Q(\mathbf{x}, y) \stackrel{\text{déf}}{=} \sum_{j=1}^{n} Q_j I(h_j(\mathbf{x}) \neq y)\,,$$

dont le risque associé correspond au risque de Gibbs, noté $R(G_Q)$. Nous avons les égalités suivantes permettant d'exprimer les risques de Gibbs réel $R(G_Q)$ et empirique $R_S(G_Q)$ en fonction des marges réalisées sur les exemples

$$\begin{aligned} R(G_Q) &= \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} \sum_{j=1}^{n} Q_j I(h_j(\mathbf{x}) \neq y) \\ &= \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} \sum_{j=1}^{n} Q_j \left(\frac{1}{2} - \frac{1}{2} y h_j(\mathbf{x})\right) \\ &= \frac{1}{2} - \frac{1}{2} \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} y\mathbf{Q}\cdot\mathbf{h}(\mathbf{x}) \\ R_S(G_Q) &= \frac{1}{2} - \frac{1}{2m} \sum_{i=1}^{m} y_i \mathbf{Q}\cdot\mathbf{h}(\mathbf{x}_i)\,. \end{aligned}$$

Notons $\widehat{R_Q^{j,k}}(\lambda)$ le risque empirique sur un ensemble S, préalablement choisi, associé à la distribution Q à laquelle un poids λ est transféré du classificateur k vers le classificateur

j. Donc $\widehat{R_Q^{j,k}}(\lambda)$ dénote le risque associé à la distribution $Q_\lambda^{j,k}$ de la définition 8.1.1. Nous avons alors

$$\begin{aligned}
\widehat{R_Q^{j,k}}(\lambda) &= \frac{1}{2} - \frac{1}{2m}\sum_{i=1}^{m}\left[y_i\mathbf{Q}\cdot\mathbf{h}(\mathbf{x}_i) + y_i\lambda h_j(\mathbf{x}_i) - y_i\lambda h_k(\mathbf{x}_i)\right] \\
&= \frac{1}{2} - \frac{1}{2m}\sum_{i=1}^{m} y_i\mathbf{Q}\cdot\mathbf{h}(\mathbf{x}_i) - \frac{1}{2m}\sum_{i=1}^{m} y_i\lambda h_j(\mathbf{x}_i) + \frac{1}{2m}\sum_{i=1}^{m} y_i\lambda h_k(\mathbf{x}_i) \\
&= R_S(G_Q) - \frac{\lambda}{2m}\sum_{i=1}^{m} y_i h_j(\mathbf{x}_i) + \frac{\lambda}{2m}\sum_{i=1}^{m} y_i h_k(\mathbf{x}_i) \\
&= R_S(G_Q) - \lambda\left(\frac{1}{2} - R_S(h_j)\right) + \lambda\left(\frac{1}{2} - R_S(h_k)\right) \\
&= R_S(G_Q) + \lambda\Big(R_S(h_j) - R_S(h_k)\Big)\,.
\end{aligned}$$

8.3.1 Borne de Langford-Seeger

Les méthodes décrites dans la section 8.1 peuvent être appliquées pour minimiser des bornes du risque de Gibbs. Pour obtenir une façon de minimiser la borne PAC-Bayes version Langford-Seeger, nous n'avons qu'à remplacer $\widehat{A_Q^{j,k}}(\lambda)$ par $\widehat{R_Q^{j,k}}(\lambda)$ et k_a par 1 dans l'inégalité 8.1. En transformant le problème de minimisation en un problème de recherche de la racine d'une fonction (voir l'équation 8.3), nous sommes amené à résoudre l'équation

$$\widetilde{F}_Q^{j,k}(\lambda) = 0$$

avec

$$\widetilde{F}_Q^{j,k}(\lambda) = \Big(R_S(h_j) - R_S(h_k)\Big)\cdot\log\left(\frac{\widehat{R_Q^{j,k}}(\lambda)(1-F_Q^{j,k}(\lambda))}{F_Q^{j,k}(\lambda)(1-\widehat{R_Q^{j,k}}(\lambda))}\right) - \frac{1}{m}\log\left(\frac{Q_j+\lambda}{Q_k-\lambda}\right),$$

et

$$F_Q^{j,k}(\lambda) = \max\left\{B : \mathrm{kl}\left(\widehat{R_Q^{j,k}}(\lambda)\,\middle\|\,B\right) \le \frac{1}{m}\left[\mathrm{KL}_Q^{j,k}(\lambda) + \log\frac{\xi(m)}{\delta}\right]\right\}.$$

En calculant la dérivée de $\widetilde{F}_Q^{j,k}$ (pour l'application de la méthode de Newton, voir section 8.1.3), nous trouvons

$$\begin{aligned}\frac{\partial \widetilde{F}_Q^{j,k}(\lambda)}{\partial \lambda} &= \frac{1}{m}\cdot\frac{(Q_j+Q_k)}{(Q_j+\lambda)(\lambda-Q_k)} \\ &\quad+\frac{\partial \widehat{R_Q^{j,k}}(\lambda)}{\partial\lambda}\left(\frac{\partial \widehat{R_Q^{j,k}}(\lambda)}{\partial\lambda}\cdot\frac{1}{\widehat{R_Q^{j,k}}(\lambda)-\left(\widehat{R_Q^{j,k}}(\lambda)\right)^2}\right. \\ &\qquad\left.+\frac{\partial F_Q^{j,k}(\lambda)}{\partial\lambda}\cdot\frac{1}{F_Q^{j,k}(\lambda)(F_Q^{j,k}(\lambda)-1)}\right) \\ &\quad+\Big(R_S(h_j)-R_S(h_k)\Big)\log\left(\frac{\widehat{R_Q^{j,k}}(\lambda)(F_Q^{j,k}(\lambda)-1)}{F_Q^{j,k}(\lambda)(\widehat{R_Q^{j,k}}(\lambda)-1)}\right).\end{aligned}$$

En portant les fonctions $\widetilde{F}_Q^{j,k}(\lambda)$ et $\frac{\partial \widetilde{F}_Q^{j,k}(\lambda)}{\partial\lambda}$ dans l'algorithme 2, nous obtenons un algorithme permettant de trouver la distribution Q minimisant la borne PAC-Bayes du risque de Gibbs (version Langford-Seeger).

8.3.2 Borne de Catoni

Pour la version Catoni du théorème PAC-Bayes sur les fonctions de perte générales, la fonction à minimiser, pour une paire j,k de classificateurs fixée, est donnée à l'équation 8.9. En remplaçant $\widehat{\zeta_Q^{j,k}}(\lambda)$ dans cette équation par $\widehat{R_Q^{j,k}}(\lambda)$, nous obtenons comme fonction à minimiser

$$\begin{aligned}F_Q^{j,k}(\lambda) &= C\cdot m\cdot\widehat{R_Q^{j,k}}(\lambda)+\mathrm{KL}_Q^{j,k}(\lambda)\\ &= C\cdot m\cdot\Big(R_S(G_Q)+\lambda R_S(h_j)-\lambda R_S(h_k)\Big)+\mathrm{KL}_Q^{j,k}(\lambda)\,,\end{aligned}$$

où $\mathrm{KL}_Q^{j,k}(\lambda)$ est défini à l'équation 8.2.

Pour minimiser cette fonction, on cherche le zéro de sa dérivée, c'est-à-dire que l'on résout

$$\widetilde{F}_Q^{j,k}(\lambda)=0 \tag{8.16}$$

avec

$$\widetilde{F}_Q^{j,k}(\lambda)\stackrel{\text{déf}}{=}\frac{\partial F_Q^{j,k}}{\partial\lambda}=C\cdot m\cdot\Big(R_S(h_j)-R_S(h_k)\Big)+\log\left(\frac{Q_j+\lambda}{Q_k-\lambda}\right).$$

On remarque que cette équation doit donner une solution assez simple. En effet, si les risques empiriques de h_j et de h_k sont égaux, le premier terme de la dérivée est nul et

le second s'annule seulement si $Q_j + \lambda = Q_k - \lambda$, c'est-à-dire seulement si la valeur de λ fait en sorte qu'après l'échange de poids, les poids respectifs de h_j et h_k deviennent égaux.

Pour ce qui est de la dérivée de $\widetilde{F}_Q^{j,k}(\lambda)$, puisque seul le terme en logarithme est fonction de λ, nous trouvons

$$\frac{\partial \widetilde{F}_Q^{j,k}}{\partial \lambda} = \frac{(Q_j + Q_k)}{(Q_j + \lambda)(Q_k - \lambda)} \,.$$

En portant les fonctions $\widetilde{F}_Q^{j,k}(\lambda)$ et $\frac{\partial \widetilde{F}_Q^{j,k}(\lambda)}{\partial \lambda}$ dans l'algorithme 2, nous obtenons un algorithme permettant de trouver la distribution Q minimisant la borne PAC-Bayes du risque de Gibbs (version Catoni).

8.3.3 Résultats empiriques

Ensemble				(1) G_Q-**kl**		(2) G_Q-**C-vc**		(3) G_Q-**C-bm**			**SSM**
Nom	$\lvert S\rvert$	$\lvert T\rvert$	n	R_T	Borne	R_T	C	R_T	C	Borne	
Adult	1809	10000	14	**0.205**	0.255	**0.205**	0.2	**0.205**	0.2	0.245	
BreastCancer	343	340	9	0.068	0.134	**0.059**	0.1	0.068	1	0.121	
Credit-A	353	300	15	**0.133**	0.241	**0.133**	0.1	**0.133**	0.5	0.224	
Glass	107	107	9	**0.224**	0.438	**0.224**	100	**0.224**	1	0.410	
Haberman	144	150	3	**0.273**	0.387	**0.273**	0.001	**0.273**	0.5	0.355	
Heart	150	147	13	0.231	0.415	**0.204**	0.2	0.231	0.5	0.392	
Ionosphere	176	175	34	0.194	0.349	**0.154**	0.2	0.194	1	0.331	
Letter:AB	500	1055	16	0.093	0.165	**0.049**	0.005	0.093	0.5	0.152	$(2) < (1,3)$
Letter:DO	500	1058	16	0.141	0.214	**0.104**	0.05	0.141	0.5	0.199	$(2) < (1,3)$
Letter:OQ	500	1036	16	0.257	0.328	0.208	0.1	**0.194**	0.5	0.327	$(2,3) < (1)$
Liver	170	175	6	0.406	0.576	0.400	0.001	**0.389**	0.5	0.545	
MNIST:0vs8	500	1916	784	0.046	0.111	**0.020**	0.02	0.046	0.5	0.107	$(2) < (1,3)$
MNIST:1vs7	500	1922	784	0.045	0.124	**0.035**	0.05	0.049	0.5	0.118	
MNIST:1vs8	500	1936	784	0.144	0.263	**0.143**	20	0.145	0.5	0.248	
MNIST:2vs3	500	1905	784	0.153	0.228	**0.093**	0.05	0.138	0.5	0.215	$(2) < (1,3)$
Mushroom	4062	4062	22	0.209	0.245	**0.132**	0.005	0.209	0.1	0.241	$(2) < (1,3)$
Ringnorm	3700	3700	20	0.398	0.420	0.407	20	**0.397**	0.1	0.413	
Sonar	104	104	60	0.356	0.515	**0.269**	100	0.385	1	0.488	
Usvotes	235	200	16	**0.055**	0.115	**0.055**	0.2	**0.055**	1	0.104	
Waveform	4000	4000	21	0.207	0.229	**0.124**	0.01	0.140	0.2	0.235	$(2,3) < (1)$
Wdbc	285	284	30	0.063	0.183	**0.053**	0.2	0.081	1	0.169	

TABLE 8.1 – Comparaison de trois algorithmes d'apprentissage basés sur la minimisation d'une borne de type PAC-Bayes sur le risque de Gibbs.

Le tableau 8.1 présente les résultats obtenus par trois algorithmes, chacun concevant un vote de majorité pondéré par une distribution Q optimisant une borne sur le risque de Gibbs : G_Q-kl dénote l'algorithme minimisant la borne du théorème 5.1.1, alors que G_Q-C-vc et G_Q-C-bm dénotent deux versions de l'algorithme minimisant la borne du théorème 6.3.1. Pour chacune de ces deux versions de l'algorithme, nous avons testé un jeu de 16 valeurs différentes du paramètre C (0.01, 0.02, 0,05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100, 200, 500 et 1000), dans la version G_Q-C-vc le paramètre C permettant de construire le classificateur final est choisi par validation croisée, alors que dans la version G_Q-C-bm, c'est le paramètre permettant d'obtenir la plus petite borne de $R(G_Q)$ qui est choisi.

En fait, comme nous pouvions nous y attendre, les résultats empiriques obtenus en testant ces algorithmes sur des données empiriques ne sont pas très bons (et cela ne vaut pas la peine de comparer les résultats avec ceux d'AdaBoost ou de la régression ridge). Ces mauvaises performances des algorithmes sont en soit un résultat positif pour nous, puisqu'ils illustrent l'importance de considérer des fonctions de perte plus complexes que la simple perte linéaire.

8.4 Risque quadratique

Nous étudions dans cette section deux algorithmes d'apprentissage basés sur la minimisation de bornes PAC-Bayes du risque associé à la perte quadratique, cette dernière étant définie, pour un exemple $(\mathbf{x}, y)$ donné, par

$$\zeta_Q^\gamma(\mathbf{x}, y) \stackrel{\text{déf}}{=} \left(\frac{y\mathbf{Q} \cdot \mathbf{h}(\mathbf{x})}{\gamma} - 1 \right)^2 ,$$

où γ est un hyperparamètre pris dans l'intervalle $[0, 1)$. Le risque quadratique, noté ζ_Q^γ, et le risque quadratique empirique évalué sur un ensemble $S = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\}$ (préalablement choisi), noté $\widehat{\zeta_Q^\gamma}$, sont donnés par

$$\zeta_Q^\gamma = \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} \left(\frac{y\mathbf{Q} \cdot \mathbf{h}(\mathbf{x})}{\gamma} - 1 \right)^2 \quad \text{et} \quad \widehat{\zeta_Q^\gamma} = \frac{1}{m} \sum_{i=1}^{m} \left(\frac{y_i\mathbf{Q} \cdot \mathbf{h}(\mathbf{x}_i)}{\gamma} - 1 \right)^2 .$$

Pour un hyperparamètre γ et une distribution Q donnés, notons $\widehat{\zeta_Q^{j,k}}(\lambda)$ le risque quadratique associé à la distribution $Q_\lambda^{j,k}$. Nous avons

$$\widehat{\zeta_Q^{j,k}}(\lambda) = \frac{1}{m} \sum_{i=1}^{m} \left(\frac{y_i\mathbf{Q} \cdot \mathbf{h}(\mathbf{x}_i) + y_i\lambda h_j(\mathbf{x}_i) - y_i\lambda h_k(\mathbf{x}_i)}{\gamma} - 1 \right)^2 .$$

En définissant

$$A_Q(j,k) = \frac{2}{m\gamma^2}\sum_{i=1}^{m} y_i\Big(y_i\mathbf{Q}\cdot\mathbf{h}(\mathbf{x}_i) - \gamma\Big)\Big(h_j(\mathbf{x}_i) - h_k(\mathbf{x}_i)\Big) \tag{8.17}$$

$$D_Q(j,k) = \frac{1}{m\gamma^2}\sum_{i=1}^{m}(h_j(\mathbf{x}_i) - h_k(\mathbf{x}_i))^2\,, \tag{8.18}$$

nous pouvons écrire de façon plus compacte

$$\widehat{\zeta_Q^{j,k}}(\lambda) = \widehat{\zeta_Q} + \lambda A_Q(j,k) + \lambda^2 D_Q(j,k)\,. \tag{8.19}$$

Les dérivées première et seconde de cette fonction par rapport λ sont données par

$$\frac{\partial}{\partial\lambda}\widehat{\zeta_Q^{j,k}}(\lambda) = A_Q(j,k) + 2\lambda D_Q(j,k)$$

et

$$\frac{\partial^2}{\partial\lambda^2}\widehat{\zeta_Q^{j,k}}(\lambda) = 2D_Q(j,k)\,.$$

Nous présentons, dans les sous-sections suivantes, deux algorithmes d'apprentissage construisant des classificateurs par vote de majorité pondéré par une distribution minimisant une borne sur le risque quadratique : un premier algorithme est basé sur le théorème sur les fonctions de perte générales version Langford-Seeger (théorème 5.1.1), et le second sur le même théorème en version Catoni (théorème 6.3.1).

8.4.1 Minimisation de la borne du théorème PAC-Bayes version Langford-Seeger

Nous nous basons à nouveau sur les sections 8.1.1 et 8.1.3 pour concevoir l'équation que doit résoudre l'algorithme d'apprentissage et pour formuler une méthode pour résoudre cette équation (soit en fait une implémentation de la méthode de Newton). Comme la perte quadratique vaut 1 sur les exemples $(\mathbf{x},y)$ tels que $W_Q(\mathbf{x},y) = \frac{1}{2}$ (en effet, lorsque $W_Q(\mathbf{x},y) = \frac{1}{2}$, l'égalité $y\mathbf{Q}\cdot\mathbf{h}(\mathbf{x}) = 1 - 2W_Q(\mathbf{x},y)$ nous indique que l'on doit avoir $y\mathbf{Q}\cdot\mathbf{h}(\mathbf{x}) = 0$), minimiser la borne du risque quadratique est équivalent à résoudre l'égalité 8.7.

$$\frac{\partial\widehat{\zeta_Q^{j,k}}(\lambda)}{\partial\lambda}\cdot\log\left(\frac{(\widehat{\zeta_Q^{j,k}}(\lambda) - 1 + c_a)(1 - F_Q^{j,k}(\lambda))}{F_Q^{j,k}(\lambda)(c_a - \widehat{\zeta_Q^{j,k}}(\lambda) + 1)}\right) - \frac{2c_a\cdot k_a}{m}\log\left(\frac{Q_j + \lambda}{Q_k - \lambda}\right) = 0\,,$$

où

$$F_Q^{j,k}(\lambda) \stackrel{\text{déf}}{=} \max\left\{B : \mathrm{kl}\left(\frac{1}{2c_a}\left[\widehat{\zeta_Q^{j,k}}(\lambda) - 1\right] + \frac{1}{2}\,\middle\|\, B\right) \le \frac{1}{m}\left[k_a\cdot\mathrm{KL}_Q^{j,k}(\lambda) + \log\frac{\xi(m)}{\delta}\right]\right\}\,.$$

avec

$$c_a = \frac{2}{\gamma} + \frac{1}{\gamma^2} \quad \text{et} \quad k_a = \frac{\gamma+1}{\gamma+\frac{1}{2}}\,.$$

En portant dans les équations (8.12) et (8.14) les quantités données par

$$A_Q^{j,k}(\lambda) = \frac{1}{2c_a}\left[\widehat{\zeta_Q^{j,k}}(\lambda) - 1\right] + \frac{1}{2}$$

et ses dérivées première et seconde, ainsi que les quantités c_a et k_a données ci-haut (où $\widehat{\zeta_Q^{j,k}}(\lambda)$ est donnée à l'équation 8.19), nous obtenons les fonctions $\widetilde{F}_Q^{j,k}(\lambda)$ et $\frac{\partial}{\partial\lambda}\widetilde{F}_Q^{j,k}(\lambda)$ qu'il faut placer dans l'algorithme 2 pour obtenir un algorithme d'apprentissage retournant un classificateur par vote de majorité pondéré par la distribution Q minimisant la borne du théorème 5.1.1 appliqué avec le risque quadratique.

8.4.2 Minimisation de la borne de Catoni

En portant dans les équations 8.13 et 8.15 les expressions que nous avons calculées pour les dérivées de $\widehat{\zeta_Q^{j,k}}(\lambda)$, nous obtenons les fonctions $\widetilde{F}_Q^{j,k}(\lambda)$ et $\frac{\partial}{\partial\lambda}\widetilde{F}_Q^{j,k}(\lambda)$ suivantes :

$$\widetilde{F}_Q^{j,k}(\lambda) \stackrel{\text{déf}}{=} C \cdot \Big(A_Q(j,k) + 2\lambda D_Q(j,k)\Big) + \log\left(\frac{Q_j+\lambda}{Q_k-\lambda}\right)\,;$$

$$\frac{\partial \widetilde{F}_Q^{j,k}(\lambda)}{\partial\lambda} = 2\cdot C\cdot D_Q(j,k) + \frac{Q_j+Q_k}{(Q_j+\lambda)(Q_k-\lambda)}\,.$$

En portant ces deux fonctions dans l'algorithme 2, nous obtenons un algorithme d'apprentissage retournant des classificateurs par vote de majorité pondéré par une distribution Q minimisant la borne du théorème 5.1.1 appliqué avec le risque quadratique.

8.4.3 Temps d'exécution

Les temps d'exécution des algorithmes donnés dans cette section dépendent principalement du nombre d'itérations effectuées durant l'exécution, c'est-à-dire du nombre de transferts de poids effectués et de temps nécessaire aux différentes itérations.

Chaque itération nécessite de calculer les valeurs $A_Q(j,k)$ et $D_Q(j,k)$. Le temps de calculs de $D_Q(j,k)$ est clairement de l'ordre de $O(m)$ (où m est le nombre d'exemples). À cause du produit scalaire présent dans la sommation, le temps de calcul de $A_Q(j,k)$ est apparemment de l'ordre de $O(m\cdot|\mathcal{H}|)$. Cependant, en maintenant à jour un vecteur des marges, c'est-à-dire un vecteur $M = (M_1, M_2, \ldots, M_m)$ tel que $M_i = y_i\mathbf{Q}\cdot\mathbf{h}(\mathbf{x}_i)\,\forall i$,

le calcul de $A_Q(j,k)$ se fait dans un temps de l'ordre de $O(m)$. À noter qu'il faut initialement poser $M_i = 0\,\forall i$.

Une fois les quantités $A_Q(j,k)$ et $D_Q(j,k)$ calculées, la recherche de λ_{opt} se fait dans un temps constant, c'est-à-dire qui peut être borné par une valeur qui ne dépend ni de m ni de $|\mathcal{H}|$. Il faut ensuite mettre à jour le distribution Q, ce qui se fait en temps constant, et mettre à jour le vecteur des marges, ce qui se fait dans un temps de l'ordre de $O(m)$, puisqu'il s'agit de poser

$$M_i \leftarrow M_i + y_i\lambda_{opt}h_j(\mathbf{x}_i) - y_i\lambda_{opt}h_k(\mathbf{x}_i) \quad \forall i\,.$$

Ainsi chaque itération de l'algorithme s'exécute dans un temps de l'ordre de $O(m)$. Le temps total d'exécution de l'algorithme est donc de l'ordre de $O(m \cdot T)$, où T est le nombre d'itérations effectuées lors de l'exécution.

8.4.4 Résultats

Ensemble				(1) **AB**	(2) **RR**		(3) **Quad-C**			(4) **Quad-kl**		**SSM**
Nom	$\|S\|$	$\|T\|$	n	R_T	R_T	C	R_T	γ	C	R_T	γ	
Adult	1809	10000	14	0.149	**0.148**	0.2	0.153	0.5	0.2	0.161	0.5	$(1,2) < (4)$
BreastCancer	343	340	9	0.053	0.050	10	**0.041**	0.7	0.1	0.050	0.9	
Credit-A	353	300	15	0.170	0.157	2	**0.133**	0.5	0.2	0.150	0.2	
Glass	107	107	9	**0.178**	0.206	5	0.187	0.1	2	0.196	0.0001	
Haberman	144	150	3	**0.260**	0.273	20	0.273	0.8	0.01	**0.260**	0.3	
Heart	150	147	13	0.252	0.197	1	**0.163**	0.5	0.01	0.184	0.0001	
Ionosphere	176	175	34	0.120	0.131	0.05	**0.103**	0.2	0.1	0.131	0.6	
Letter:AB	500	1055	16	0.010	**0.003**	0.2	0.004	0.1	0.01	0.027	0.2	$(1,2,3) < (4)$
Letter:DO	500	1058	16	0.036	**0.026**	0.05	**0.026**	0.1	0.1	0.045	0.0001	$(2,3) < (4)$
Letter:OQ	500	1036	16	**0.038**	0.044	0.2	0.049	0.05	0.02	0.062	0.0001	$(1) < (4)$
Liver	170	175	6	0.320	**0.309**	5	0.354	0.6	0.05	0.360	0.8	
MNIST:0vs8	500	1916	784	**0.008**	0.015	0.05	0.015	0.005	0.2	0.011	0.3	
MNIST:1vs7	500	1922	784	0.013	0.011	0.05	**0.010**	0.02	0.01	0.021	0.2	$(2,3) < (4)$
MNIST:1vs8	500	1936	784	0.025	0.024	0.2	**0.021**	0.1	0.05	0.046	0.3	$(1,2,3) < (4)$
MNIST:2vs3	500	1905	784	0.047	**0.033**	0.2	0.041	0.3	0.05	0.048	0.0001	
Mushroom	4062	4062	22	**0.000**	**0.000**	0.02	**0.000**	0.0001	0.01	0.021	0.5	$(1,2,3) < (4)$
Ringnorm	3700	3700	20	0.043	**0.037**	0.05	**0.037**	0.05	0.02	0.062	0.05	$(1,2,3) < (4)$
Sonar	104	104	60	0.231	0.192	0.05	0.202	0.2	1	**0.154**	0.0001	
Usvotes	235	200	16	**0.055**	0.060	2	**0.055**	0.3	0.05	**0.055**	0.9	
Waveform	4000	4000	21	0.085	**0.079**	10	**0.079**	0.5	0.5	0.083	0.4	
Wdbc	285	284	30	0.049	0.049	0.2	0.053	0.05	0.01	**0.032**	0.8	

TABLE 8.2 – Résultats d'expérimentations avec des algorithmes d'apprentissage basés sur une borne de type PAC-Bayes sur le risque quadratique.

Le tableau 8.2 présente des résultats d'expérimentations de deux algorithmes d'apprentissage construisant des votes de majorité pondérés par des distributions minimisant une borne de type PAC-Bayes du risque quadratique. L'algorithme identifié par Quad-C

est basé sur la version Catoni du théorème sur les fonctions de perte générales, alors que l'algorithme Quad-kl est basé sur la version Langford-Seeger.

Nous avons comparé nos algorithmes avec AdaBoost (AB) et la régression ridge (RR) (voir Hoerl et Kennard, 1970b,a) sur 21 ensembles de données (la méthodologie employée pour les comparaisons est décrite à la section 8.2). Les valeurs testées pour le paramètre γ correspondent à l'ensemble $\{0.0001, 0.0002,\ 0.0005, 0.001, 0.002, 0.005,\ 0.01, 0.02, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5,\ 0.6,\ 0.7,\ 0.8,\ 0.9, 1\}$ et celles du paramètre C à l'ensemble $\{0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10,\ 20,\ 50,\ 100,\ 200,\ 500, 1000\}$. La colonne SSM indique que l'algorithme Quad-kl est clairement moins performant que les autres algorithmes, puisqu'il est significativement moins bon qu'au moins un des autres algorithmes sur 8 ensemble de données, et même significativement moins bon que les trois autres algorithmes sur 4 ensembles de données. Aucune autre différence significative n'est présente, c'est-à-dire que Quad-C est essentiellement équivalent à AdaBoost et à la régression ridge.

8.5 Risque exponentiel

Dans cette section, nous nous intéressons à concevoir des algorithmes construisant un classificateur par vote de majorité pondéré par une distribution minimisant une borne du risque exponentiel. Ces algorithmes seront en fait des implémentations de l'algorithme 2, il nous suffit alors de définir les fonctions $\widetilde{F}_Q^{j,k}(\lambda)$ propres à chaque algorithme. Comme nous avons fait précédemment pour les risques linéaires et quadratique, nous présentons deux algorithmes, l'un basé sur la borne fournie par le théorème sur les fonctions de perte générales découlant du théorème PAC-Bayes version Langford-Seeger (théorème 5.1.2), et l'autre basé sur ce même théorème en version Catoni (théorème 6.3.1).

La perte exponentielle sur un exemple $(\mathbf{x}, y)$ est donnée par

$$\mathcal{E}_Q^\gamma(\mathbf{x}, y) = \exp\left(-\frac{y\mathbf{Q}\cdot\mathbf{h}(\mathbf{x})}{\gamma}\right),$$

où $\gamma \in (0, \infty)$ est un hyperparamètre. Le risque exponentiel, noté $\mathcal{E}_Q^\gamma(\mathbf{x}, y)$, et son risque empirique sur un ensemble fixé S de m exemples, noté $\widehat{\mathcal{E}_Q^\gamma}$, sont donc donnés par

$$\mathcal{E}_Q^\gamma = \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} \exp\left(-\frac{y\mathbf{Q}\cdot\mathbf{h}(\mathbf{x})}{\gamma}\right)$$

et

$$\widehat{\mathcal{E}_Q^\gamma} = \frac{1}{m}\sum_{i=1}^{m} \exp\left(-\frac{y_i\mathbf{Q}\cdot\mathbf{h}(\mathbf{x}_i)}{\gamma}\right).$$

Pour γ, fixé posons

$$\begin{aligned}
\widehat{\mathcal{E}_Q^{j,k}}(\lambda) &\stackrel{\text{déf}}{=} \frac{1}{m}\sum_{i=1}^{m}\exp\left(-\frac{y_i\mathbf{Q}\cdot\mathbf{h}(\mathbf{x}_i)}{\gamma}-\frac{\lambda y_i h_j(\mathbf{x}_i)}{\gamma}+\frac{\lambda y_i h_k(\mathbf{x}_i)}{\gamma}\right)\\
&= \frac{1}{m}\sum_{i=1}^{m}\exp\left(-\frac{y_i\mathbf{Q}\cdot\mathbf{h}(\mathbf{x}_i)}{\gamma}\right)\exp\left(-\frac{\lambda y_i h_j(\mathbf{x}_i)}{\gamma}\right)\exp\left(\frac{\lambda y_i h_k(\mathbf{x}_i)}{\gamma}\right)\\
&= \frac{1}{m}\sum_{i=1}^{m}\exp\left(-\frac{y_i\mathbf{Q}\cdot\mathbf{h}(\mathbf{x}_i)}{\gamma}\right)I(h_j(\mathbf{x}_i)=h_k(\mathbf{x}_i))\\
&\quad+\frac{1}{m}\exp\left(-\frac{2\lambda}{\gamma}\right)\sum_{i=1}^{m}\exp\left(-\frac{y_i\mathbf{Q}\cdot\mathbf{h}(\mathbf{x}_i)}{\gamma}\right)I(h_j(\mathbf{x}_i)=-h_k(\mathbf{x}_i)=y_i)\\
&\quad+\frac{1}{m}\exp\left(\frac{2\lambda}{\gamma}\right)\sum_{i=1}^{m}\exp\left(-\frac{y_i\mathbf{Q}\cdot\mathbf{h}(\mathbf{x}_i)}{\gamma}\right)I(h_j(\mathbf{x}_i)=-h_k(\mathbf{x}_i)=-y_i)\,.
\end{aligned}$$

En posant

$$\begin{aligned}
D_Q^{=}(j,k) &= \frac{1}{m}\sum_{i=1}^{m}\exp\left(-\frac{y_i\mathbf{Q}\cdot\mathbf{h}(\mathbf{x}_i)}{\gamma}\right)I(h_j(\mathbf{x}_i)=h_k(\mathbf{x}_i))\,;\\
D_Q^{+}(j,k) &= \frac{1}{m}\sum_{i=1}^{m}\exp\left(-\frac{y_i\mathbf{Q}\cdot\mathbf{h}(\mathbf{x}_i)}{\gamma}\right)I(h_j(\mathbf{x}_i)=-h_k(\mathbf{x}_i)=y_i)\,;\\
D_Q^{-}(j,k) &= \frac{1}{m}\sum_{i=1}^{m}\exp\left(-\frac{y_i\mathbf{Q}\cdot\mathbf{h}(\mathbf{x}_i)}{\gamma}\right)I(h_j(\mathbf{x}_i)=-h_k(\mathbf{x}_i)=-y_i)\,,
\end{aligned}$$

nous pouvons écrire de façon plus concise

$$\widehat{\mathcal{E}_Q^{j,k}}(\lambda) = D_Q^{=}(j,k)+D_Q^{+}(j,k)\cdot\exp\left(-\frac{2\lambda}{\gamma}\right)+D_Q^{-}(j,k)\cdot\exp\left(\frac{2\lambda}{\gamma}\right). \tag{8.20}$$

Les dérivées première et seconde de $\widehat{\mathcal{E}_Q^{j,k}}(\lambda)$ sont alors données par

$$\frac{\partial}{\partial\lambda}\widehat{\mathcal{E}_Q^{j,k}}(\lambda) = \frac{-2D_Q^{+}(j,k)}{\gamma}\exp\left(-\frac{2\lambda}{\gamma}\right)+\frac{2D_Q^{-}(j,k)}{\gamma}\exp\left(\frac{2\lambda}{\gamma}\right) \tag{8.21}$$

et

$$\frac{\partial^2}{\partial\lambda^2}\widehat{\mathcal{E}_Q^{j,k}}(\lambda) = \frac{4D_Q^{+}(j,k)}{\gamma^2}\exp\left(-\frac{2\lambda}{\gamma}\right)+\frac{4D_Q^{-}(j,k)}{\gamma^2}\exp\left(\frac{2\lambda}{\gamma}\right). \tag{8.22}$$

8.5.1 Borne de Langford-Seeger

Pour minimiser la borne du théorème 5.1.2 appliqué avec le risque exponentiel, il suffit d'implémenter l'algorithme 2 en définissant de façon appropriée la fonction $\widetilde{F}_Q^{j,k}(\lambda)$.

La forme générale de cette fonction est donnée à l'équation (8.12), il nous faut alors définir la quantité $\widehat{A_Q^{j,k}}(\lambda)$. Selon l'équation (8.5), on a

$$\widehat{A_Q^{j,k}}(\lambda) = \frac{\widehat{\mathcal{E}_Q^{j,k}}(\lambda) - 1}{2c_a} + \frac{1}{2},$$

et puisque pour le risque exponentiel nous avons $c_a = e^{1/\gamma} - 1$, on obtient

$$\widehat{A_Q^{j,k}}(\lambda) = \frac{1}{2} \cdot \frac{\widehat{\mathcal{E}_Q^{j,k}}(\lambda) - 1}{e^{1/\gamma} - 1} + \frac{1}{2}.$$

Pour les dérivées première et seconde de cette fonction, on trouve

$$\frac{\partial \widehat{A_Q^{j,k}}(\lambda)}{\partial \lambda} = \frac{-1}{(e^{1/\gamma} - 1)\gamma} \left(D_Q^+(j,k) \exp\left(-\frac{2\lambda}{\gamma}\right) - D_Q^-(j,k) \exp\left(\frac{2\lambda}{\gamma}\right)\right)$$

et

$$\frac{\partial^2 \widehat{A_Q^{j,k}}(\lambda)}{\partial \lambda^2} = \frac{2}{(e^{1/\gamma} - 1)\gamma^2} \left(D_Q^+(j,k) \exp\left(-\frac{2\lambda}{\gamma}\right) + D_Q^-(j,k) \exp\left(\frac{2\lambda}{\gamma}\right) . \right)$$

En portant ces expressions de $\widehat{A_Q^{j,k}}(\lambda)$ et de $\frac{\partial \widehat{A_Q^{j,k}}(\lambda)}{\partial \lambda}$ dans l'équation (8.12) et en portant ces mêmes quantités ainsi que $\frac{\partial^2 \widehat{A_Q^{j,k}}(\lambda)}{\partial \lambda^2}$ dans l'équation (8.14), nous obtenons respectivement les fonctions $\widetilde{F}_Q^{j,k}(\lambda)$ et $\frac{\partial \widetilde{F}_Q^{j,k}(\lambda)}{\partial \lambda}$ que nous devons placer dans l'algorithme 2 pour obtenir un algorithme retournant la distribution Q sur $\mathcal{H}$ minimisant la borne du risque exponentiel fournie par le théorème 5.1.2.

8.5.2 Borne de Catoni

Pour minimiser la borne du théorème 6.3.1 appliqué avec le risque exponentiel, il suffit d'implémenter l'algorithme 2 en définissant de façon appropriée la fonction $\widetilde{F}_Q^{j,k}(\lambda)$. La forme générale de cette fonction est donnée à l'équation 8.13, en y portant la valeur de $\frac{\partial \widehat{\mathcal{E}_Q^{j,k}}(\lambda)}{\partial \lambda}$ donnée ci-haut nous trouvons

$$\widetilde{F}_Q^{j,k}(\lambda) = \frac{-2C}{\gamma} \left(D_Q^+(j,k) \cdot \exp\left(\frac{-2\lambda}{\gamma}\right) - D_Q^-(j,k) \cdot \exp\left(\frac{2\lambda}{\gamma}\right)\right) + \log\left(\frac{Q_j + \lambda}{Q_k - \lambda}\right),$$

on calcule alors pour la dérivée de cette fonction

$$\frac{\partial \widetilde{F}_Q^{j,k}(\lambda)}{\partial \lambda} = \frac{4C}{\gamma^2} \left(D_Q^+(j,k) \cdot \exp\left(\frac{-2\lambda}{\gamma}\right) + D_Q^-(j,k) \cdot \exp\left(\frac{2\lambda}{\gamma}\right)\right) + \frac{Q_j + Q_k}{(Q_j + \lambda)(Q_k - \lambda)}.$$

En portant ces expressions des fonctions $\widetilde{F}_Q^{j,k}(\lambda)$ et $\frac{\partial \widetilde{F}_Q^{j,k}(\lambda)}{\partial \lambda}$ dans l'algorithme 2, nous obtenons un algorithme retournant la distribution Q sur $\mathcal{H}$ minimisant la borne du risque exponentiel fournie par le théorème 6.3.1.

8.5.3 Temps d'exécution

En tenant à jour un vecteur des marges (pour plus de détails, voir l'analyse du temps d'exécution des algorithmes basés sur le risque quadratique, section 8.4.3), le calcul des quantités $D_Q^{=}(j,k)$, $D_Q^{+}(j,k)$ et $D_Q^{-}(j,k)$ s'effectue dans un temps de l'ordre de $O(m)$.

Une fois ces valeurs calculées, la recherche de λ_{opt} (le poids transférer entre les classificateurs h_j et h_k) se fait dans un temps indépendant de m (le nombre d'exemples) et de $|\mathcal{H}|$. Suite à cela, la mise à jour du vecteur $\mathbf{Q}$ se fait en temps constant et la mise à jour du vecteur des marges se fait dans un temps de l'ordre de $O(m)$. Ainsi, pour les deux algorithmes présentés dans cette section, chaque itération s'exécute dans un temps total de l'ordre de $O(m)$.

8.5.4 Résultats

Ensemble				(1) **AB**	(2) **RR**		(3) **Exp-C**			(4) **Exp-kl**		**SSM**
Nom	$\lvert S\rvert$	$\lvert T\rvert$	n	R_T	R_T	C	R_T	γ	C	R_T	γ	
Adult	1809	10000	14	0.149	**0.148**	0.2	0.150	0.1	0.2	0.166	0.7	$(1,2,3) < (4)$
BreastCancer	343	340	9	0.053	**0.050**	10	0.062	0.5	2	0.056	0.9	
Credit-A	353	300	15	0.170	0.157	2	0.153	0.2	0.2	**0.140**	0.8	
Glass	107	107	9	0.178	0.206	5	**0.168**	0.1	1	0.187	0.9	
Haberman	144	150	3	0.260	0.273	20	**0.253**	0.9	1	0.273	0.05	
Heart	150	147	13	0.252	0.197	1	**0.156**	0.6	1	0.163	0.5	
Ionosphere	176	175	34	0.120	0.131	0.05	**0.097**	0.02	2	0.154	0.7	
Letter:AB	500	1055	16	0.010	**0.003**	0.2	0.004	0.001	1000	0.039	0.6	$(1,2,3) < (4)$
Letter:DO	500	1058	16	0.036	**0.026**	0.05	0.036	0.01	0.5	0.073	0.3	$(1,2,3) < (4)$
Letter:OQ	500	1036	16	**0.038**	0.044	0.2	0.055	0.005	0.01	0.104	0.5	$(1,2,3) < (4)$
Liver	170	175	6	0.320	**0.309**	5	0.360	0.2	0.5	0.337	0.8	
MNIST:0vs8	500	1916	784	0.008	0.015	0.05	**0.007**	0.05	50	0.016	0.5	$(3) < (4)$
MNIST:1vs7	500	1922	784	0.013	**0.011**	0.05	0.012	0.01	5	0.034	0.5	$(1,2,3) < (4)$
MNIST:1vs8	500	1936	784	0.025	0.024	0.2	**0.021**	0.02	50	0.070	0.5	$(1,2,3) < (4)$
MNIST:2vs3	500	1905	784	0.047	**0.033**	0.2	0.046	0.05	2	0.069	0.5	$(1,2,3) < (4)$
Mushroom	4062	4062	22	**0.000**	**0.000**	0.02	**0.000**	0.001	0.5	0.070	0.4	$(1,2,3) < (4)$
Ringnorm	3700	3700	20	0.043	0.037	0.05	**0.029**	0.01	0.05	0.118	0.4	$(1,2) < (4)$,$(3) < (1,4)$
Sonar	104	104	60	0.231	**0.192**	0.05	0.240	0.001	0.5	0.202	0.7	
Usvotes	235	200	16	**0.055**	0.060	2	**0.055**	0.9	5	**0.055**	0.9	
Waveform	4000	4000	21	0.085	**0.079**	10	0.081	0.2	0.05	0.090	0.3	
Wdbc	285	284	30	0.049	0.049	0.2	**0.042**	0.001	200	0.046	0.6	

TABLE 8.3 – Résultats d'expérimentations avec des algorithmes d'apprentissage basés sur une borne de type PAC-Bayes sur le risque exponentiel.

Le tableau 8.5.4 présente des résultats d'expérimentations de deux algorithmes d'apprentissage construisant des votes de majorité pondéré par une distribution minimisant une borne de type PAC-Bayes du risque quadratique. L'algorithme identifié par Quad-C est basé sur la version Catoni du théorème sur les fonctions de perte générales, alors que l'algorithme Quad-kl est basé sur la version Langford-Seeger.

Nous avons comparé nos algorithmes avec AdaBoost (AB) et la régression ridge (RR) sur 21 ensembles de données (la méthodologie employée pour les comparaisons est décrite à la section 8.2). Les valeurs paramètres γ et C permettant d'obtenir les résultats présentés ont été choisis par validation croisée. Les valeurs testées pour le paramètre γ correspondent à l'ensemble $\{0.0001, 0.0002,\ 0.0005, 0.001, 0.002, 0.005,\ 0.01, 0.02, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5,\ 0.6,\ 0.7,\ 0.8,\ 0.9, 1\}$ et celles du paramètre C à l'ensemble $\{0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20,\ 50,\ 100,\ 200,\ 500, 1000\}$. La colonne SSM indique que l'algorithme Quad-kl est clairement moins performant que les autres algorithmes, puisqu'il est significativement moins bon qu'au moins un des autres algorithmes sur 10 ensemble de données, et même significativement moins bon que les trois autres algorithmes sur 9 ensembles de données.

On remarque également dans le tableau que l'algorithme Exp-C se démarque légèrement des autres algorithmes, en effet, il est le seul algorithme significativement meilleur que Exp-kl sur l'ensemble MNIST:0vs8 et il est même significativement meilleur qu'AdaBoost sur l'ensemble Ringnorm.

8.6 Classificateur de Gibbs à piges multiples

Dans cette section, nous nous intéressons à concevoir des algorithmes construisant un classificateur par vote de majorité pondéré par une distribution minimisant une borne du risque du classificateur de Gibbs à piges multiples. Ces algorithmes seront en fait des implémentations de l'algorithme 2, il nous suffit alors de définir les fonctions $\widetilde{F}_Q^{j,k}(\lambda)$ propres à chaque algorithme. Nous présentons ici deux algorithmes, l'un basé sur la borne fournie par le théorème sur les fonctions de perte générales découlant du théorème PAC-Bayes version Langford-Seeger (théorème 5.1.1), et l'autre basé sur ce même théorème en version Catoni (théorème 6.3.1).

8.6.1 Risque du classificateur de Gibbs à piges multiples

Notons $R_N(W)$ la fonction représentant la perte du classificateur de Gibbs à N piges associée à un taux d'erreur W. C'est-à-dire que $R_N(W)$ correspond à la perte du classificateur de Gibbs à N piges pour un exemple $(\mathbf{x}, y)$ tel que $W_Q(\mathbf{x}, y) = W$. Nous avons alors

$$R_N(W) = \sum_{\ell=\lceil N/2 \rceil}^{N} \binom{N}{\ell} (1-W)^\ell W^{N-\ell}\,.$$

Donc, la perte du classificateur de Gibbs à N piges sur un exemple donné $(\mathbf{x}, y)$ est simplement donné par $R_{(\mathbf{x},y)}(G_{Q^N}) = R_N(W_Q(\mathbf{x}, y))$, il suit que le vrai risque du classificateur de Gibbs à N piges est donné par

$$\begin{aligned} R(G_{Q^N}) &= \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} R_N(W_Q(\mathbf{x}, y)) \\ &= \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} R_N\Big(\frac{1}{2} - \frac{1}{2} y_i \mathbf{Q} \cdot \mathbf{h}(\mathbf{x}_i)\Big) \end{aligned}$$

et que son risque empirique est donné par

$$\begin{aligned} R_S(G_{Q^N}) &= \frac{1}{m}\sum_{i=1}^{m} R_N(W_Q(\mathbf{x}_i, y_i)) \\ &= \frac{1}{m}\sum_{i=1}^{m} \sum_{\ell=\lceil N/2\rceil}^{N} \binom{N}{\ell} \Big(\frac{1}{2} - \frac{1}{2} y_i \mathbf{Q} \cdot \mathbf{h}(\mathbf{x}_i)\Big)^{\ell} \Big(\frac{1}{2} + \frac{1}{2} y_i \mathbf{Q} \cdot \mathbf{h}(\mathbf{x}_i)\Big)^{N-\ell}. \end{aligned}$$

La fonction de perte associée au classificateur de Gibbs à piges multiples est une fonction non convexe, pour obtenir de meilleurs propriétés de convergence pour nos algorithmes, nous allons considérer la forme convexifiée de cette fonction de perte (voir figure 8.6.1). Notons, pour une valeur de N impaire, $\mathcal{R}_{(\mathbf{x},y)}(G_{Q^N})$ la perte convexifiée du classificateur de Gibbs à piges multiples, c'est-à-dire

$$\mathcal{R}_{(\mathbf{x},y)}(G_{Q^N}) = \begin{cases} R_{(\mathbf{x},y)}(G_{Q^N}) & \text{si } W_Q(\mathbf{x}, y) < \frac{1}{2} \\ \frac{1}{2} - \frac{1}{2} y_i \mathbf{Q} \cdot \mathbf{h}(\mathbf{x}_i) \cdot R'_N\Big(\frac{1}{2}\Big) & \text{si } W_Q(\mathbf{x}, y) \geq \frac{1}{2}, \end{cases}$$

où $R'_N(W)$ est la dérivée de $R_N(W)$ en fonction de W, on a donc

$$R'_N\Big(\frac{1}{2}\Big) = \frac{N! \cdot \Big(\frac{1}{2}\Big)^{N-1}}{\lfloor \frac{N}{2} \rfloor ! \Big(\lfloor \frac{N}{2} \rfloor - 1\Big)!}$$

(voir la proposition 8.6.1 et ce qui en découle pour les détails du calcul de la dérivée). Donc $\mathcal{R}_{(\mathbf{x},y)}(G_{Q^N})$ correspond à $R_N(W_Q((\mathbf{x}, y))$ pour les exemples $(\mathbf{x}, y)$ ayant un taux de désaccord inférieur à $\frac{1}{2}$, et correspond à la droite qui est tangente à $R_N(W_Q((\mathbf{x}, y))$ au point $\frac{1}{2}$ pour les exemples ayant un taux de désaccord supérieur à $\frac{1}{2}$.

Considérons l'ensemble d'entrainement $S = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\}$ comme étant fixé et définissons la fonction $\widehat{\mathcal{R}^{j,k}_{Q^N}}(\lambda)$ comme retournant, en fonction de λ, le risque empirique du classificateur de Gibbs à N piges associé à la distribution $Q^{j,k}_{\lambda}$, qui correspond à la distribution Q dans laquelle un poids λ est transféré du classificateur h_k vers le classificateur h_j (voir la définition 8.1.1). C'est-à-dire que l'on a

$$\widehat{\mathcal{R}^{j,k}_{Q^N}}(\lambda) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{R}_{(\mathbf{x}_i,y_i)}(G_{(Q^{j,k}_{\lambda})^N})$$

avec

$$\mathcal{R}_{(\mathbf{x}_i,y_i)}(G_{(Q_\lambda^{j,k})^N}) = \begin{cases} R_N\Big(\frac{1}{2} - \frac{y_i}{2}\Big(\mathbf{Q}\cdot\mathbf{h}(\mathbf{x}_i) + \lambda h_j(\mathbf{x}_i) - \lambda h_k(\mathbf{x}_i)\Big)\Big) & \text{si } W_Q(\mathbf{x},y) < \frac{1}{2} \\ \frac{1}{2} - \frac{y_i}{2}\Big(\mathbf{Q}\cdot\mathbf{h}(\mathbf{x}_i) + \lambda h_j(\mathbf{x}_i) - \lambda h_k(\mathbf{x}_i)\Big)\cdot R'_N\Big(\frac{1}{2}\Big) & \text{si } W_Q(\mathbf{x},y) \geq \frac{1}{2}\,. \end{cases}$$

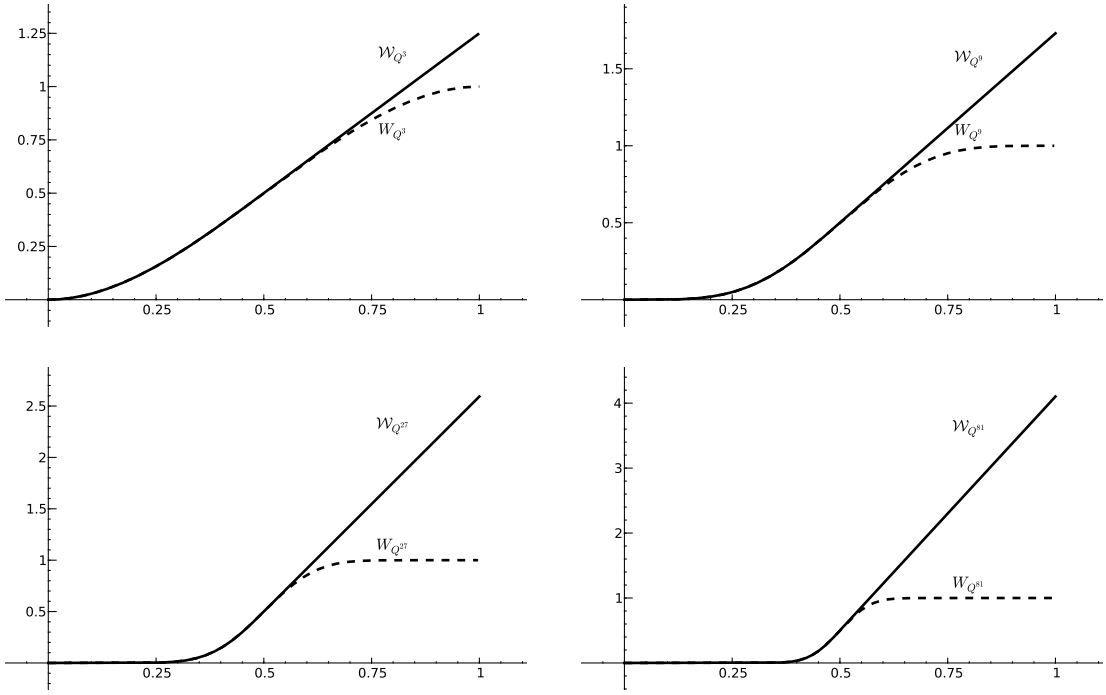

FIGURE 8.1 – Comparaison de la perte du classificateur de Gibbs à piges multiples (lignes pointillées) et de sa version convexifiée (lignes continues) pour 3, 9, 27 et 81 piges.

8.6.2 Calcul des dérivées

Un problème qui se pose pour l'implémentation d'un algorithme basé sur la minimisation du risque du classificateur de Gibbs à piges multiples, est le temps de calcul de celui-ci. En effet, pour évaluer $R_S(G_{Q^N})$, nous devons calculer une double sommation de la forme

$$\sum_{i=1}^{m}\sum_{\ell=\lceil N/2\rceil}^{N}(\dots)\,,$$

ce qui nécessite un temps de l'ordre de $\Theta(m\cdot N)$. De plus, il n'est pas aisé, comme cela l'était pour les risques linéaires, quadratique ou exponentiel (voir par exemple

l'égalité 8.20), de mettre d'un côté l'apport sur le risque empirique $\widehat{\mathcal{R}^{j,k}_{Q^N}}(\lambda)$ d'une distribution Q fixée, et celui des différentes valeurs possibles de λ. Par exemple, pour le risque exponentielle, pour Q fixée, après avoir évalué les quantités $D^{=}_{Q}, D^{+}_{Q}$ et D^{-}_{Q}, l'évaluation de $\widehat{\mathcal{E}^{j,k}_{Q}}(\lambda)$ se fait en temps constant. Or à cause de la complexité du risque du classificateurs de Gibbs à piges multiples, chaque évaluation de $\widehat{\mathcal{R}^{j,k}_{Q^N}}(\lambda)$ nécessite un temps de l'ordre de $\Theta(m \cdot N)$. Cependant, une égalité reliant la cumulative de la distribution binomiale et la fonction bêta (voir proposition 8.6.1), nous permet de grandement améliorer le calcul du risque du classificateur de Gibbs à piges multiples, ainsi que le calcul de ses dérivées (pour l'implémentation l'algorithme 2).

Proposition 8.6.1. *Nous avons l'égalité*

$$R_N(W) = \frac{B\left(W; \left\lceil \frac{N}{2} \right\rceil, \left\lfloor \frac{N}{2} \right\rfloor + 1\right)}{B\left(\left\lceil \frac{N}{2} \right\rceil, N\right)}$$

où

$$B(a,b) \stackrel{\text{déf}}{=} \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$$

correspond à la fonction bêta et

$$B(x;a,b) \stackrel{\text{déf}}{=} \int_0^x t^{a-1}(1-t)^{b-1}\,dt$$

correspond à la fonction bêta incomplète.

Démonstration : Découle de l'égalité connue (qui s'obtient, pour a et b entiers, en intégrant $b-1$ fois par parties la fonction $B(x;a,b)$)

$$\sum_{j=a}^{a+b-1} \binom{a+b-1}{j} x^j (1-x)^{a+b-1-j} = \frac{B(x;a,b)}{B(a,b)}$$

en posant $a = \left\lceil \frac{N}{2} \right\rceil$ et $b = \left\lfloor \frac{N}{2} \right\rfloor + 1$. ■

À partir de la proposition 8.6.1 il est facile de calculer la dérivée de $R_N(W)$, en effet, le théorème fondamental de l'analyse nous permet d'affirmer que

$$\frac{d}{dx}\int_0^x t^{a-1}(1-t)^{b-1}\,dt = x^{a-1}(1-x)^{b-1}\,,$$

il suit directement que nous avons

$$\frac{\partial}{\partial W} R_N(W) = \frac{N!}{\lfloor N/2 \rfloor!\,(\lceil N/2 \rceil - 1)!} W^{(\lceil N/2 \rceil - 1)}(1-W)^{\lfloor N/2 \rfloor}\,.$$

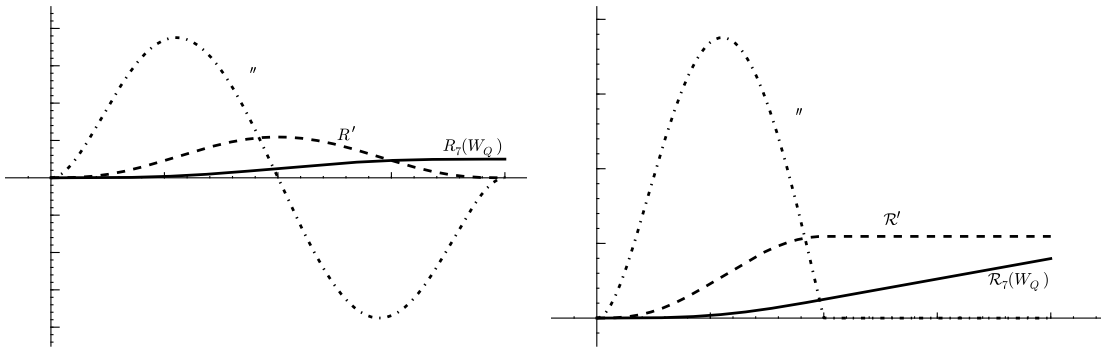

FIGURE 8.2 – Représentation de la fonction de perte associée au classificateur de Gibbs à 7 piges, $R_7(W_Q)$, et de ses dérivées première, $R_7'(W_Q)$, et seconde, $R_7''(W_Q)$ (figure de droite), et de leurs versions convexes (figure de gauche).

Une fois la dérivée première calculée, il n'est pas non plus difficile de calculer la dérivée seconde, on obtient

$$\frac{\partial^2}{\partial W^2} R_N(W) = -\frac{N!}{\lfloor N/2 \rfloor! \Big(\lceil N/2 \rceil - 1 \Big)!} W^{\lceil N/2 \rceil - 2} (1-W)^{\lfloor N/2 \rfloor - 1} \Big((N-1)W - \lceil N/2 \rceil + 1 \Big) .$$

La figure 8.6.2 illustre les différentes fonctions $R_N(W)$, $\frac{\partial}{\partial W} R_N(W)$ et $\frac{\partial^2}{\partial W^2} R_N(W)$ dans leur version originale et dans leur convexifiée pour une valeur de N égale à 7.

À partir de la dérivée de R_N par rapport à W, nous pouvons obtenir la dérivée de $\widehat{\mathcal{R}_{Q^N}^{j,k}}(\lambda)$ par rapport à λ. Comme

$$W_{Q_\lambda^{j,k}}(\mathbf{x}, y) = \frac{1}{2} - \frac{1}{2} y \mathbf{Q} \cdot \mathbf{h}(\mathbf{x}) - \lambda \frac{1}{2} y h_j(\mathbf{x}) + \lambda \frac{1}{2} y h_k(\mathbf{x}) ,$$

on a

$$\frac{\partial}{\partial \lambda} W_{Q_\lambda^{j,k}}(\mathbf{x}, y) = \frac{y}{2} \Big(h_k(\mathbf{x}) - h_j(\mathbf{x}) \Big) .$$

La règle sur la dérivation des fonctions composées nous donne alors

$$\begin{aligned}
&\frac{\partial}{\partial\lambda}\mathcal{R}_{(\mathbf{x},y)}(G_{(Q_\lambda^{j,k})^N}) \\
&\qquad = \frac{\partial}{\partial W_{Q_\lambda^{j,k}}(\mathbf{x},y)}\mathcal{R}_{(\mathbf{x},y)}(G_{(Q_\lambda^{j,k})^N})\frac{\partial}{\partial\lambda}W_{Q_\lambda^{j,k}}(\mathbf{x},y) \\
&\qquad = \frac{y}{2}\Big(h_k(\mathbf{x})-h_j(\mathbf{x})\Big)\cdot\frac{\partial}{\partial W_{Q_\lambda^{j,k}}(\mathbf{x},y)}\mathcal{R}_{(\mathbf{x},y)}(G_{(Q_\lambda^{j,k})^N}) \\
&\qquad = \begin{cases} \dfrac{y}{2}\Big(h_k(\mathbf{x})-h_j(\mathbf{x})\Big)\cdot\dfrac{\partial}{\partial W_{Q_\lambda^{j,k}}(\mathbf{x},y)}R_{(\mathbf{x},y)}(G_{(Q_\lambda^{j,k})^N}) & \text{si } W_Q(\mathbf{x},y)<\frac{1}{2} \\ \dfrac{y\Big(h_k(\mathbf{x})-h_j(\mathbf{x})\Big)N!\Big(\frac{1}{2}\Big)^{N-1}}{2\lfloor N/2\rfloor!\,(\lceil N/2\rceil]-1)!} & \text{sinon}\,, \end{cases}
\end{aligned}$$

avec

$$\frac{\partial}{\partial W_{Q_\lambda^{j,k}}(\mathbf{x},y)}R_{(\mathbf{x},y)}(G_{(Q_\lambda^{j,k})^N}) = \frac{N!}{\lfloor N/2\rfloor!\,(\lceil N/2\rceil]-1)!}(1-W_{Q_\lambda^{j,k}}(\mathbf{x},y))^{\lfloor N/2\rfloor}\Big(W_{Q_\lambda^{j,k}}(\mathbf{x},y)\Big)^{(\lceil N/2\rceil-1)}.$$

Pour la dérivée seconde, puisque $\frac{\partial}{\partial\lambda}W_{Q_\lambda^{j,k}}(\mathbf{x},y)$ correspond à une constante en fonction de λ, nous avons,

$$\begin{aligned}
&\frac{\partial^2}{\partial\lambda^2}\mathcal{R}_{(\mathbf{x},y)}(G_{(Q_\lambda^{j,k})^N}) \\
&\qquad = \frac{\partial^2}{\partial\Big(W_{Q_\lambda^{j,k}}(\mathbf{x},y)\Big)^2}\mathcal{R}_{(\mathbf{x},y)}(G_{(Q_\lambda^{j,k})^N})\cdot\Big(\frac{\partial}{\partial\lambda}W_{Q_\lambda^{j,k}}(\mathbf{x},y)\Big)^2 \\
&\qquad = \frac{1}{4}\cdot\Big(h_k(\mathbf{x})-h_j(\mathbf{x})\Big)^2\cdot\frac{\partial^2}{\partial\Big(W_{Q_\lambda^{j,k}}(\mathbf{x},y)\Big)^2}\mathcal{R}_{(\mathbf{x},y)}(G_{(Q_\lambda^{j,k})^N}) \\
&\qquad = \begin{cases} \dfrac{1}{4}\cdot\Big(h_k(\mathbf{x})-h_j(\mathbf{x})\Big)^2\cdot\dfrac{\partial^2}{\partial\Big(W_{Q_\lambda^{j,k}}(\mathbf{x},y)\Big)^2}R_{(\mathbf{x},y)}(G_{(Q_\lambda^{j,k})^N}) & \text{si } W_Q(\mathbf{x},y)<\frac{1}{2} \\ 0 & \text{sinon}\,, \end{cases}
\end{aligned}$$

avec

$$\begin{aligned}
&\frac{\partial^2}{\partial\Big(W_{Q_\lambda^{j,k}}(\mathbf{x},y)\Big)^2}R_{(\mathbf{x},y)}(G_{(Q_\lambda^{j,k})^N}) = \\
&\qquad -\frac{N!}{\lfloor N/2\rfloor!\,\Big(\lceil N/2\rceil-1\Big)!}\cdot\Big(1-W_{Q_\lambda^{j,k}}(\mathbf{x},y)\Big)^{\lfloor N/2\rfloor-1}\Big(W_{Q_\lambda^{j,k}}(\mathbf{x},y)\Big)^{\lceil N/2\rceil-2}\cdot \\
&\qquad\qquad \Big((N-1)W_{Q_\lambda^{j,k}}(\mathbf{x},y)-\lceil N/2\rceil+1\Big).
\end{aligned}$$

8.6.3 Borne du théorème 5.1.1

En appliquant le théorème PAC-Bayes pour borner le risque du classificateur de Gibbs à piges multiples, nous obtenons la borne suivante valide avec probabilité $1-\delta$ simultanément pour toute distribution à postériori Q sur $\mathcal{H}$

$$R(G_{Q^N}) \leq \max\left\{B : \mathrm{kl}\left(R_S(G_{Q^N}) \middle\| B\right) \leq \frac{1}{m}\left[N \cdot \mathrm{KL}(Q\|P) + \log\frac{\xi(m)}{\delta}\right]\right\}.$$

Puisque $R_{(\mathbf{x},y)}(G_{Q^N}) \leq \mathcal{R}_{(\mathbf{x},y)}(G_{Q^N})$ et puisque la fonction $f_\epsilon(x) \stackrel{\text{déf}}{=} \max\{y : \mathrm{kl}(x\|\, y) \leq \epsilon\}$ est monotone croissante, nous avons que la borne suivante de $R(G_{Q^N})$ est une borne valide avec probabilité au moins $1-\delta$ simultanément pour toute distribution à postériori Q sur $\mathcal{H}$, ainsi que pour tout $j,k \in \{1,2,\ldots n\}$ et pour tout λ pris dans l'intervalle $[-\min(Q_j, 1-Q_k), \min(Q_k, 1-Q_j)]$:

$$R(G_{Q^N}) \leq \max\left\{B : \mathrm{kl}\left(\widehat{\mathcal{R}^{j,k}_{Q^N}}(\lambda) \middle\| B\right) \leq \frac{1}{m}\left[N \cdot \mathrm{KL}^{j,k}_Q(\lambda) + \log\frac{\xi(m)}{\delta}\right]\right\},$$

où $\mathrm{KL}^{j,k}_Q(\lambda) \stackrel{\text{déf}}{=} \mathrm{KL}(Q^{j,k}_\lambda\|P)$.

Pour minimiser cette borne, il suffit d'implémenter l'algorithme 2 en définissant de façon appropriée la fonction $\widetilde{F}^{j,k}_Q(\lambda)$ définie à l'équation 8.12 :

$$\widetilde{F}^{j,k}_Q(\lambda) \stackrel{\text{déf}}{=} \frac{\partial\widehat{\mathcal{R}^{j,k}_{Q^N}}(\lambda)}{\partial\lambda} \cdot \log\left(\frac{\widehat{\mathcal{R}^{j,k}_{Q^N}}(\lambda)(1-F^{j,k}_Q(\lambda)}{F^{j,k}_Q(\lambda)(1-\widehat{\mathcal{R}^{j,k}_{Q^N}}(\lambda))}\right) - \frac{N}{m}\log\left(\frac{Q_j+\lambda}{Q_k-\lambda}\right),$$

où, pour j et k fixés, $F^{j,k}_Q(\lambda)$ retourne la borne de PAC-Bayes de $\mathcal{R}^{j,k}_{Q^N}(\lambda)$, c'est-à-dire

$$F^{j,k}_Q(\lambda) = \max\left\{B : \mathrm{kl}\left(\widehat{\mathcal{R}^{j,k}_{Q^N}}(\lambda) \middle\| B\right) \leq \frac{1}{m}\left[N \cdot \mathrm{KL}^{j,k}_Q(\lambda) + \log\frac{\xi(m)}{\delta}\right]\right\}.$$

8.6.4 Borne du théorème 6.3.1

En appliquant le théorème 6.3.1 pour borner le risque du classificateur de Gibbs à piges multiples, nous obtenons la borne suivante, qui est valide avec probabilité $1-\delta$ simultanément pour toute distribution à postériori Q sur $\mathcal{H}$:

$$R(G_{Q^N}) \leq \frac{1}{1-e^{-C}}\left\{1 - \exp\left[-\left(C \cdot R_S(G_{Q^N}) + \frac{1}{m}\left[N \cdot \mathrm{KL}(Q\|P) + \log\frac{1}{\delta}\right]\right)\right]\right\}.$$

Cette borne demeure valide en remplaçant le risque empirique par sa version convexifiée, nous avons donc que la borne suivante est une borne valide avec probabilité au moins

$1-\delta$ simultanément pour toute distribution à postériori Q sur $\mathcal{H}$, ainsi que pour tout $j,k \in \{1,2,\ldots n\}$ et $\lambda \in [-\min(Q_j, 1-Q_k), \min(Q_k, 1-Q_j)]$:

$$R(G_{Q^N}) \leq \frac{1}{1-e^{-C}}\left\{1-\exp\left[-\left(C\cdot\widehat{\mathcal{R}^{j,k}_{Q^N}}(\lambda)+\frac{1}{m}\left[N\cdot \mathrm{KL}^{j,k}_{Q}(\lambda)+\log\frac{1}{\delta}\right]\right)\right]\right\}.$$

Pour Q, j et k fixés, minimiser cette borne est équivalent à trouver la valeur λ minimisant la quantité suivante

$$C\cdot m\cdot\widehat{\mathcal{R}^{j,k}_{Q^N}}(\lambda)+N\cdot \mathrm{KL}^{j,k}_{Q}(\lambda)\,.$$

Pour minimiser la borne du théorème 6.3.1 appliqué avec le risque de Gibbs à piges multiples, il suffit d'implémenter l'algorithme 2 en définissant de façon appropriée la fonction $\widetilde{F}^{j,k}_{Q}(\lambda)$:

$$\widetilde{F}^{j,k}_{Q}(\lambda) = C\cdot m\cdot\frac{\partial\widehat{\mathcal{R}^{j,k}_{Q^N}}(\lambda)}{\partial\lambda}+\log\left(\frac{Q_j+\lambda}{Q_k-\lambda}\right).$$

8.6.5 Temps d'exécution

En maintenant à jour un vecteur des marges, chaque évaluation de la fonction $\widetilde{F}^{j,k}_{Q}(\lambda)$ se fait dans un temps de l'ordre de $O(m)$ (et non en temps constant comme pour les algorithmes associés aux pertes quadratique et exponentielle). Il suit que dans le cas des algorithmes de cette section, chaque recherche de λ_{opt} se fait dans un temps de l'ordre de $O(m\cdot k(\epsilon))$, où $k(\epsilon)$ est le nombre d'itérations de la méthode de Newton requis pour trouver λ_{opt} (pour les algorithmes des sections précédentes, ce temps était plutôt de l'ordre de $O(m+k(\epsilon))$, nous avions alors ignoré le terme $k(\epsilon)$ qui était moins pertinent).

Une fois λ_{opt} trouvé, la mise à jour du vecteur $\mathbf{Q}$ se fait dans un temps constant et la mise à jour du vecteur des marges se fait dans un temps de l'ordre $O(m)$. Le temps nécessaire à chaque itération des algorithmes de cette section est donc de l'ordre de $O(m\cdot k(\epsilon))$.

8.6.6 Résultats empiriques avec sélection de paramètres par validation croisée

Le tableau 8.6.6 présente des résultats d'expérimentations de nos deux algorithmes d'apprentissage construisant des votes de majorité pondéré par une distribution mini-

Ensemble				(1) **AB**	(2) **RR**		(3) **GibbsN-C**			(4) **GibbsN-kl**		**SSM**
Nom	$\|S\|$	$\|T\|$	n	R_T	R_T	C	R_T	N	C	R_T	N	
Adult	1809	10000	14	0.149	**0.148**	0.2	0.152	499	20	0.162	25	$(1,2) < (4)$
BreastCancer	343	340	9	0.053	0.050	10	0.041	7	1	**0.038**	7	
Credit-A	353	300	15	0.170	0.157	2	0.150	9999	2	**0.143**	99	
Glass	107	107	9	0.178	0.206	5	**0.131**	49	500	0.178	49	
Haberman	144	150	3	**0.260**	0.273	20	0.273	1	0.001	0.273	1	
Heart	150	147	13	0.252	0.197	1	**0.177**	75	1	**0.177**	75	
Ionosphere	176	175	34	0.120	0.131	0.05	**0.103**	499	200	0.137	5	
Letter:AB	500	1055	16	0.010	**0.003**	0.2	0.009	49	2	0.009	49	
Letter:DO	500	1058	16	0.036	**0.026**	0.05	0.027	999	50	0.043	9999	
Letter:OQ	500	1036	16	**0.038**	0.044	0.2	0.041	4999	200	0.056	99	
Liver	170	175	6	0.320	**0.309**	5	0.349	25	2	0.366	5	
MNIST:0vs8	500	1916	784	0.008	0.015	0.05	**0.007**	49	50	0.013	499	
MNIST:1vs7	500	1922	784	0.013	**0.011**	0.05	**0.011**	49999	100	0.016	75	
MNIST:1vs8	500	1936	784	0.025	0.024	0.2	**0.021**	499	500	0.040	25	$(1,2,3) < (4)$
MNIST:2vs3	500	1905	784	0.047	**0.033**	0.2	0.045	75	20	0.048	75	
Mushroom	4062	4062	22	**0.000**	**0.000**	0.02	**0.000**	999	100	0.012	499	$(1,2,3) < (4)$
Ringnorm	3700	3700	20	0.043	0.037	0.05	**0.026**	9999	200	0.047	49999	$(3) < (1,2,4)$
Sonar	104	104	60	0.231	**0.192**	0.05	**0.192**	25	20	0.231	3	
Usvotes	235	200	16	**0.055**	0.060	2	**0.055**	1	0.2	**0.055**	3	
Waveform	4000	4000	21	0.085	**0.079**	10	0.081	49	0.5	0.081	49	
Wdbc	285	284	30	0.049	0.049	0.2	**0.035**	499	20	**0.035**	7	

TABLE 8.4 – Résultats d'expérimentations avec des algorithmes d'apprentissage basés sur une borne de type PAC-Bayes sur le risque du classificateur de Gibbs à piges multiples.

misant une borne de type PAC-Bayes du risque associé à la perte convexifiée du classificateur de Gibbs à piges multiples. L'algorithme identifié par GibbsN-C est basé sur la version Catoni du théorème sur les fonctions de perte générales, alors que l'algorithme GibbsN-kl est basé sur la version Langford-Seeger.

Nous avons comparé nos algorithmes avec AdaBoost (AB) et la régression ridge (RR) sur 21 ensembles de données (la méthodologie employée pour les comparaisons est décrite à la section 8.2). Les valeurs paramètres N et C permettant d'obtenir les résultats présentés ont été choisis par validation croisée. Les valeurs testées pour le paramètre N correspondent à l'ensemble $\{1, 3, 5, 7, 9, 25, 49, 75, 99, 499, 999, 4999, 9999, 49999, 99999, 499999, 999999\}$ et celles du paramètre C à l'ensemble $\{0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100, 200, 500, 1000\}$. La colonne SSM indique que l'algorithme Quad-kl est clairement moins performant que les autres algorithmes, puisqu'il est significativement moins bon qu'au moins un des autres algorithmes sur 10 ensemble de données, et même significativement moins bon que les trois autres algorithmes sur 9 ensembles de données.

Nous déduisons de nos résultats empiriques que la version de l'algorithme basée sur le théorème de Catoni se compare avantageusement à AdaBoost ainsi qu'à la régression ridge. En effet la seule différence significative de performance entre notre algorithme et les deux algorithmes témoin s'observe avec l'ensemble Ringnorm, où l'algorithme

GibbsN-C est à la fois significativement meilleur que AdaBoost et la régression ridge.

Il n'en est pas de même pour la version basée sur le théorème PAC-Bayes version Seeger-Langford (GibbsN-kl), où l'on observe à trois reprises dans le tableau (pour Adult, MNIST:1vs8 et Mushroom) une différence significative de performance entre notre algorithme et les algorithmes témoin à l'avantage de ces derniers.

8.6.7 Résultats empiriques avec sélection de paramètres dictée par la borne

Ensemble				(1) **GibbsN-C**				(2) **GibbsN-kl**			**SSM**
Nom	$\|S\|$	$\|T\|$	n	R_T	N	C	Borne	R_T	N	Borne	
Adult	1809	10000	14	**0.205**	1	0.2	0.245	**0.205**	1	0.255	
BreastCancer	343	340	9	0.041	7	1	0.115	**0.038**	9	0.128	
Credit-A	353	300	15	**0.133**	1	0.5	0.224	**0.133**	1	0.241	
Glass	107	107	9	**0.224**	1	1	0.410	**0.224**	1	0.438	
Haberman	144	150	3	**0.273**	1	0.5	0.355	**0.273**	1	0.387	
Heart	150	147	13	**0.231**	1	0.5	0.392	**0.231**	1	0.415	
Ionosphere	176	175	34	**0.194**	1	1	0.331	**0.194**	1	0.349	
Letter:AB	500	1055	16	0.093	1	0.5	0.152	**0.014**	9	0.164	(2) < (1)
Letter:DO	500	1058	16	**0.141**	1	0.5	0.199	**0.141**	1	0.214	
Letter:OQ	500	1036	16	**0.092**	13	1	0.319	**0.092**	13	0.329	
Liver	170	175	6	**0.389**	1	0.5	0.545	0.406	1	0.576	
MNIST:0vs8	500	1916	784	0.046	1	0.5	0.107	**0.042**	1	0.120	
MNIST:1vs7	500	1922	784	0.049	1	0.5	0.118	**0.045**	1	0.130	
MNIST:1vs8	500	1936	784	0.046	15	1	0.233	**0.041**	23	0.241	
MNIST:2vs3	500	1905	784	**0.138**	1	0.5	0.215	0.151	1	0.228	
Mushroom	4062	4062	22	**0.019**	49	1	0.097	**0.019**	59	0.102	
Ringnorm	3700	3700	20	**0.046**	999999	1	0.252	0.048	999999	0.255	
Sonar	104	104	60	0.385	1	1	0.488	**0.356**	1	0.515	
Usvotes	235	200	16	**0.055**	1	1	0.104	**0.055**	1	0.123	
Waveform	4000	4000	21	**0.081**	29	0.5	0.172	**0.081**	29	0.179	
Wdbc	285	284	30	0.081	1	1	0.169	**0.077**	1	0.183	

TABLE 8.5 – Comparaison entre GibbsN-C et GibbsN-kl en sélectionnant les paramètres en fonction des bornes.

Bien que, pour des hyperparamètres sélectionnés par validation croisée, nos algorithmes d'apprentissage basés sur la minimisation du risque associé à des fonctions de perte générales offrent des performances comparables et même parfois supérieurs à celles d'algorithmes tels que AdaBoost, il n'en demeure pas ainsi lorsque nous demandons en

plus à la borne de sélectionner les hyperparamètres. En effet, dans tous les résultats que nous avons présentés, la borne sert uniquement à trouver une distribution Q optimale pour des hyperparamètres donnés, ensuite, on a recours à la validation croisée pour sélectionner les hyperparamètres. Idéalement nous aimerions utiliser nos bornes pour la sélection de paramètres, ce qui permettrait d'éviter la très couteuse étape (en termes de temps de calculs) de validation croisée ; l'idée est alors de choisir le jeu d'hyperparamètres permettant d'obtenir la plus petite borne sur le risque. Malheureusement dans la quasi-totalité de nos expérimentations, les bornes se sont montrées incapables de bien sélectionner les hyperparamètres. Par exemple, pour le risque parabolique, la borne choisit systématiquement $\gamma = 1$, et les résultats obtenus sont très loin d'être optimaux.

La situation est un peu différente dans le cas de la minimisation du risque du classificateur de Gibbs à piges multiples. En effet, on observe dans ce cas que la valeur de N permettant d'obtenir la plus petite borne n'est pas systématiquement la valeur 1, c'est-à-dire que parfois (dans 6 ou 7 de nos 21 expérimentations) il est possible d'obtenir une borne plus petite en choisissant une valeur de N supérieure à 1 (ce qui équivalent pour le risque quadratique ou pour le risque exponentiel à choisir une valeur de γ inférieure à 1, ce qui ne se produisait jamais).

8.6.8 Détails d'implémentation

Bien que les fonctions représentant les dérivées de $R_{(\mathbf{x},y)}(G_{(Q_\lambda^{j,k})^N})$ prennent des valeurs raisonnables (du point de vue de la précision numérique), leur évaluation numérique peut mener à des problèmes de précision. En effet, ces fonctions sont formées d'un produit d'une quantité considérable (attribuable à la fraction contenant les factorielles), et d'une quantité infime (attribuable aux exponentielles). Pour pallier ces problèmes de précision, nous récrivons les fonctions en nous basant sur l'égalité $f(x) = \exp(\log(f(x)))$, ainsi les produits de quantités infimes et titanesques deviennent des sommes de logarithmes de valeurs raisonnables. On obtient alors

$$\frac{\partial}{\partial W_{Q_\lambda^{j,k}}(\mathbf{x},y)} R_{(\mathbf{x},y)}(G_{(Q_\lambda^{j,k})^N}) = \exp\Big\{ \log\left(\Gamma(N+1)\right) - \log\left(\Gamma(\lfloor N/2 \rfloor + 1)\right) - \log\left(\Gamma(\lceil N/2 \rceil)\right) + \lfloor N/2 \rfloor \cdot \log\left(1 - W_{Q_\lambda^{j,k}}(\mathbf{x},y)\right) + (\lceil N/2 \rceil - 1) \cdot \log\left(W_{Q_\lambda^{j,k}}(\mathbf{x},y)\right) \Big\}$$

et

$$\frac{\partial^2}{\partial\Big(W_{Q_\lambda^{j,k}}(\mathbf{x},y)\Big)^2} R_{(\mathbf{x},y)}(G_{(Q_\lambda^{j,k})^N}) =$$
$$\exp\Big\{ -\log\left(\Gamma(N+1)\right) - \log\left(\Gamma(\lfloor N/2\rfloor+1)\right) - \log\left(\Gamma(\lceil N/2\rceil)\right)$$
$$+ (\lfloor N/2\rfloor - 1)\log\Big(1 - W_{Q_\lambda^{j,k}}(\mathbf{x},y)\Big) + (\lceil N/2\rceil - 2)\log\Big(W_{Q_\lambda^{j,k}}(\mathbf{x},y)\Big)$$
$$+ \log\Big((N-1)W_{Q_\lambda^{j,k}}(\mathbf{x},y) - \lceil N/2\rceil + 1\Big)\Big\}.$$

Dans cette écriture des fonctions dérivées, nous avons remplacer les factorielles (qui se trouvent maintenant à l'intérieur de logarithmes) par des fonctions gamma car nous n'avons pas en fait à calculer ces factorielles, puisque la fonction $\log(\Gamma(\cdot))$ se trouve directement implémentée dans certaines librairies de calculs scientifiques. Dans nos expérimentation, nous avons utilisé l'implémentation de GSL (gsl_sf_lngamma). À noter également que les appels à la fonction gsl_sf_lngamma sont de loin la partie la plus couteuse de l'évaluation des fonctions dérivées, mais, heureusement, ces appels dépendent uniquement de la valeur de N. Il est donc seulement nécessaire d'effectuer trois appels à la fonction gsl_sf_lngamma en tout dans l'exécution de l'algorithme (et non trois appels par évaluation des dérivées).

Chapitre 9

Distribution quasi-uniforme

Dans ce chapitre, nous reprenons, en les étendant un peu, les travaux parus dans Germain *et al.* (2009b) : nous définissons le concept de distribution quasi-uniforme, puis nous présentons des versions modifiées des algorithmes d'apprentissage du chapitre 8 de cette thèse dans lesquelles les distributions définissant les votes de majorité sont contraintes à être quasi-uniformes.

9.1 Introduction

Dans ce chapitre, nous concevons et analysons des algorithmes construisant des classificateurs par vote de majorité pondéré par une distribution quasi-uniforme. Avant de présenter ces algorithmes, nous fournissons, dans la section présente, quelques définitions et résultats concernant cette catégorie de distribution.

Définition 9.1.1 (Distribution quasi-uniforme)**.** *Pour* $\mathcal{H} = \{h_1, h_2, \ldots, h_n, h_{n+1}, h_{n+2}, \ldots, h_{2n}\}$ *un ensemble de classificateurs binaires tel que* $h_{i+n} = -h_i \ \forall i$ *et pour* $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_n)$ *tel que* $\epsilon_i \in [0, 1) \ \forall i$ *et* $\sum_{i=1}^{n} \epsilon_i = 1$*, on appelle* distribution $\boldsymbol{\epsilon}$-quasi-uniforme sur $\mathcal{H}$ *toute distribution* Q *telle que* $Q(h_i) + Q(h_{i+n}) = \epsilon_i \ \forall i$*. Pour* $\epsilon_i = \frac{1}{n} \ \forall i$*, nous disons simplement que* Q *est une distribution quasi-uniforme. À noter que la distribution uniforme est elle-même une distribution quasi-uniforme.*

Proposition 9.1.2. *Soit* Q *une distribution* $\boldsymbol{\epsilon}$*-quasi-uniforme sur un ensemble* $\mathcal{H}$ *de classificateurs et* P *une distribution sur* $\mathcal{H}$ *telle que* $P(h_i) = P(h_{i+n}) = \dfrac{\epsilon_i}{2}$*. Alors* $\mathrm{KL}(Q\|P) \leq \log 2$*.*

Démonstration : La contribution à la valeur de $\mathrm{KL}(Q\|P)$ d'une paire de classificateurs

complémentaires $(h_i,\, h_{i+n})$ est donnée par

$$\mathrm{KL}(w_i) = \frac{w_i + \epsilon_i}{2} \log \frac{w_i + \epsilon_i}{\epsilon_i} + \frac{\epsilon_i - w_i}{2} \log \frac{\epsilon_i - w_i}{\epsilon_i}$$

pour w_i appartenant à l'intervalle $[-\epsilon_i, \epsilon_i]$. Comme la fonction $\mathrm{KL}(w_i)$ est convexe (sa dérivée seconde étant donnée par la fonction positive $1/(2(\epsilon_i - w_i)) + 1/(2(\epsilon_i + w_i))$, elle atteint son maximum à l'une des extrémités de son intervalle de définition, c'est-à-dire au point $w_i = \epsilon_i$ ou $w_i = -\epsilon_i$. À chacun de ces points, la fonction vaut $\epsilon_i \log 2$, nous avons donc l'inégalité

$$\mathrm{KL}(w_i) \leq \epsilon_i \log 2\,.$$

Puisque cette inégalité est vraie pour chacune des paires de classificateurs complémentaires, on obtient

$$\begin{aligned} \mathrm{KL}(Q\|P) &= \sum_{i=1}^{n} \mathrm{KL}(w_i) \\ &\leq \sum_{i=1}^{n} \epsilon_i \log 2 \\ &= \log 2\,. \end{aligned}$$

■

À noter que la proposition 9.1.2 s'applique, entre autres, dans le cas où P est la distribution uniforme sur $\mathcal{H}$ et que Q est une distribution quasi-uniforme. Ainsi, dans ce cas, la valeur maximale que peut prendre $\mathrm{KL}(Q\|P)$ est donnée par une constante et ne dépend pas du nombre de classificateurs, contrairement au cas où la distribution est non contrainte (où l'on a plutôt $\mathrm{KL}(Q\|P) \leq \log 2n$ pour un ensemble $\mathcal{H}$ de $2n$ classificateurs).

Le lemme suivant nous permet d'obtenir une version améliorée du théorème PAC-Bayes (voir le théorème 9.1.4 ainsi que le corollaire 9.1.5) valide pour des distributions à priori et à postériori $\boldsymbol{\epsilon}$-quasi-uniformes.

Lemme 9.1.3. *Soit $\mathcal{H} = \{h_1, h_2, \ldots, h_n, h_{n+1}, h_{n+2}, \ldots, h_{2n}\}$ un ensemble de classificateurs binaires tel que $h_{i+n} = -h_i$ $\forall i$ et soit P et Q deux distributions $\boldsymbol{\epsilon}$-quasi-uniformes sur $\mathcal{H}$. Soit $\mathcal{D}(q\|p)$ une fonction telle que $\mathcal{D}(q\|p) = \mathcal{D}(1-q\|1-p)$. Alors*

$$\mathop{\mathbf{E}}_{h\sim P} e^{m\mathcal{D}(R_S(h)\|R(h))} = \mathop{\mathbf{E}}_{h\sim Q} e^{m\mathcal{D}(R_S(h)\|R(h))}\,.$$

Démonstration :

$$\begin{aligned}
\mathop{\mathbf{E}}_{h\sim P} e^{m\mathcal{D}(R_S(h)\|R(h))} &= \sum_{i=1}^{n} P(h_i)e^{m\mathcal{D}(R_S(h_i)\|R(h_i))} + P(h_{i+n})e^{m\mathcal{D}(R_S(h_{i+n})\|R(h_{i+n}))} \\
&= \sum_{i=1}^{n} (P(h_i) + P(h_{i+n}))e^{m\mathcal{D}(R_S(h_i)\|R(h_i))} \\
&= \sum_{i=1}^{n} \epsilon_i e^{m\mathcal{D}(R_S(h_i)\|R(h_i))} \\
&= \sum_{i=1}^{n} (Q(h_i) + Q(h_{i+n}))e^{m\mathcal{D}(R_S(h_i)\|R(h_i))} \\
&= \sum_{i=1}^{n} Q(h_i)e^{m\mathcal{D}(R_S(h_i)\|R(h_i))} + Q(h_{i+n})e^{m\mathcal{D}(R_S(h_{i+n})\|R(h_{i+n}))} \\
&= \mathop{\mathbf{E}}_{h\sim Q} e^{m\mathcal{D}(R_S(h)\|R(h))}
\end{aligned}$$

■

Théorème 9.1.4. *Soit D une distribution sur un ensemble $\mathcal{X}$ et $\mathcal{H}$ un ensemble de classificateurs binaires définis sur $\mathcal{X}$ tel que $h_{i+n} = -h_i$ $\forall i$. Soit P une distribution à priori sur $\mathcal{H}$ $\boldsymbol{\epsilon}$-quasi-uniforme et soit $\delta \in (0, 1]$. Alors pour toute fonction convexe $\mathcal{D} : [0, 1] \times [0, 1] \to \mathbf{R}$ telle que $\mathcal{D}(q\|p) = \mathcal{D}(1 - q\|1 - p)$, nous avons*

$$\Pr_{S\sim D^m}\Bigg(\forall Q \ \boldsymbol{\epsilon}\text{-}q.u.\ sur \mathcal{H}\colon\ \mathcal{D}(R_S(G_Q)\|R(G_Q)) \leq \frac{1}{m}\left[\log\left(\frac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{h\sim P} e^{m\mathcal{D}(R_S(h)\|R(h))}\right)\right]\Bigg) \geq 1 - \delta\,.$$

Démonstration : Comme $\mathop{\mathbf{E}}_{h\sim P} e^{m\mathcal{D}(R_S(h)\|R(h))}$ représente une variable aléatoire non négative, l'inégalité de Markov donne

$$\Pr_{S\sim D^m}\left(\mathop{\mathbf{E}}_{h\sim P} e^{m\mathcal{D}(R_S(h)\|R(h))} \leq \frac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{h\sim P} e^{m\mathcal{D}(R_S(h)\|R(h))}\right) \geq 1 - \delta\,.$$

Le lemme 9.1.3 nous permet de remplacer la première espérance en P en une espérance en Q. En exploitant le fait que la fonction logarithme est monotone croissante, nous obtenons alors

$$\Pr_{S\sim D^m}\left(\forall Q\ \boldsymbol{\epsilon}\text{-q.u. sur } \mathcal{H} : \log\left[\mathop{\mathbf{E}}_{h\sim Q} e^{m\mathcal{D}(R_S(h)\|R(h))}\right] \leq \log\left[\frac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{h\sim P} e^{m\mathcal{D}(R_S(h)\|R(h))}\right]\right) \geq 1 - \delta\,.$$

Le théorème suit alors de deux applications de l'inégalité de Jensen, l'une exploitant la concavité de la fonction $\log(x)$ et l'autre la convexité de $\mathcal{D}$. ■

Corollaire 9.1.5. *Soit D une distribution sur un ensemble $\mathcal{X}$, $\mathcal{H}$ un ensemble de classificateurs binaires définis sur $\mathcal{X}$ tel que $h_{i+n} = -h_i$ $\forall i$. Soit P une distribution à priori sur $\mathcal{H}$ $\boldsymbol{\epsilon}$-quasi-uniforme et soit $\delta \in (0, 1]$. Alors*

$$\Pr_{S\sim D^m}\left(\forall\, Q \text{ } \boldsymbol{\epsilon}\text{-q.u. sur } \mathcal{H}\colon \ \mathrm{kl}(R_S(G_Q)\|R(G_Q)) \leq \frac{1}{m}\log\frac{\xi(m)}{\delta}\right) \geq 1-\delta\,.$$

Démonstration : Découle du théorème 9.1.4 et de l'égalité

$$\begin{aligned}\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{h\sim P} e^{m\mathcal{D}(R_S(h)\|R(h))} &= \sum_{k=0}^{m}\binom{m}{k}(k/m)^k(1-k/m)^{m-k}\\ &=: \xi(m)\,,\end{aligned}$$

qui est valide lorsque $\mathcal{D}(q\|p)$ est la fonction $\mathrm{kl}(q\|p)$. ■

Nous venons de voir un théorème permettant d'obtenir une borne de type PAC-Bayes spécifique aux distributions quasi-uniformes permettant de se départir du régularisateur $\mathrm{KL}(Q\|P)$. Toutefois, toutes les distributions à postériori Q ne sont pas quasi-uniformes, il y a donc lieu de se demander si le fait de travailler avec des distributions quasi-uniformes ne restreint pas la puissance des classificateurs par vote de majorité que l'on peut construire. La proposition suivante indique que ce n'est pas le cas, c'est-à-dire que pour toute distribution Q satisfaisant une certaine contrainte non excessive, le classificateur par vote de majorité B_Q a un équivalent $B_{Q'}$ avec Q' quasi-uniforme.

Proposition 9.1.6. *Soit $\mathcal{H} = \{h_1, \ldots, h_n, h_{n+1}, \ldots, h_{2n}\}$ un ensemble de classificateurs binaires et Q une distribution sur $\mathcal{H}$ pour laquelle il existe $A > 0$ tel que*

$$A\cdot\Big|Q(h_i) - Q(h_{i+n})\Big| \leq \epsilon_i \quad \forall\, i \in [1, 2, \ldots, n]\,,$$

avec $\sum_{i=1}^{n}\epsilon_i = 1$. Alors il existe une distribution $\boldsymbol{\epsilon}$-quasi-uniforme Q' Bayes-équivalente à Q, c'est-à-dire telle que $B_{Q'}(\mathbf{x}) = B_Q(\mathbf{x})$ $\forall \mathbf{x}\in\mathcal{X}$.

Démonstration : Il suffit de prendre, pour $i \in [1, 2, \ldots, n]$

$$Q'(h_i) = \frac{1}{2}\Big(AQ(h_i) - AQ(h_{i+n}) + \epsilon_i\Big) \quad ; \quad Q'(h_{i+n}) = \frac{1}{2}\Big(\epsilon_i - AQ(h_i) + AQ(h_{i+n})\Big)\,.$$

En effet, on vérifie facilement que Q' est $\boldsymbol{\epsilon}$-quasi-uniforme, de plus, on a

$$\begin{aligned}
B_{Q'}(\mathbf{x}) &= \operatorname{sgn}\left(\sum_{h\in\mathcal{H}} Q'(h)h(\mathbf{x})\right)\\
&= \operatorname{sgn}\left(\sum_{i=1}^{n}\left[\frac{1}{2}\Big(AQ(h_i) - AQ(h_{i+n}) + \epsilon_i\Big) - \frac{1}{2}\Big(\epsilon_i - AQ(h_i) + AQ(h_{i+n})\Big)\right]h_i(\mathbf{x})\right)\\
&= \operatorname{sgn}\left(\sum_{i=1}^{n}\Big[AQ(h_i) - AQ(h_{i+n})\Big]h_i(\mathbf{x})\right)\\
&= \operatorname{sgn}\left(A\sum_{h\in\mathcal{H}} Q(h)h(\mathbf{x})\right)\\
&= B_Q(\mathbf{x})\,.
\end{aligned}$$

■

À noter que l'hypothèse relative à l'existence d'une certaine valeur $A > 0$ n'est aucunement contraignante (pour le développement de nos algorithmes). En effet, comme $\mathcal{H}$ est fini, l'hypothèse faite sur Q est moins forte que la condition $\forall i : P(h_i)+P(h_{i+n}) = 0 \Rightarrow Q(h_i) + Q(h_{i+n}) = 0$, qui revient à imposer que Q soit telle que $\mathrm{KL}(Q\|P) < \infty$ (condition nécessaire pour obtenir une borne PAC-Bayes classique inférieure à 1).

9.1.1 Théorème PAC-Bayes sur les fonctions de perte générales pour les distributions quasi-uniformes

Considérons $\mathcal{H} = \{h_1, h_2, \ldots, h_n, h_{n+1}, h_{n+2}, \ldots, h_{2n}\}$ un ensemble de classificateurs binaires complémentaires, c'est-à-dire tel que $h_{i+n} = -h_i\ \forall i$, et considérons Q une distribution quasi-uniforme sur $\mathcal{H}$. Pour $k \in \mathbf{N}$, l'ensemble $\mathcal{H}^k$ est un ensemble constitué de $2^k n^k$ classificateurs, cependant, pour $i_1, i_2, \ldots, i_k \in \{1, 2, \ldots, n\}$ donnés, tout classificateur de la forme $h_{i_1+b_1}h_{i_2+b_2}\cdots h_{i_k+b_k}$ avec $b_1, b_2, \ldots, b_k \in \{0, n\}$ se trouve être soit identique à $h_{i_1}h_{i_2}\cdots h_{i_k}$ soit identique à $h_{i_1+n}h_{i_2}\cdots h_{i_k} = -h_{i_1}h_{i_2}\cdots h_{i_k}$. En regroupant ensemble ces classificateurs identiques, nous pouvons voir l'ensemble $\mathcal{H}^k$ comme étant constitué de $2n^k$ classificateurs (plutôt que $2^k n^k$). Il n'est pas difficile de vérifier que la distribution Q^k est quasi-uniforme sur cet ensemble $\mathcal{H}^k$. En effet, pour s'en convaincre, il suffit de vérifier que

$$\sum_{b_1\in\{0,n\}}\sum_{b_2\in\{0,n\}}\cdots\sum_{b_k\in\{0,n\}} Q(h_{i_1+b_1})Q(h_{i_2+b_2})\cdots Q(h_{i_k+b_k}) = \frac{1}{n^k}\,.$$

Il découle de cette observation que le théorème PAC-Bayes pour les distributions quasi-uniformes s'applique dans le cas des fonctions de perte générales. Nous avons donc le théorème suivant.

Théorème 9.1.7. *Soit $\zeta_Q(\mathbf{x}, y)$ une fonction de perte de la forme de l'équation 5.6. Soit ζ_Q et $\widehat{\zeta_Q}$ respectivement le risque associé à $\zeta_Q(\mathbf{x}, y)$ et son estimé empirique sur un échantillon de m exemples. Alors, pour tout ensemble $\mathcal{H}$ de classificateurs binaires complémentaires, pour toute distribution quasi-uniforme à priori P sur $\mathcal{H}$ et pour tout $\delta \in (0, 1]$, nous avons*

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\forall\, Q \text{ q.u. sur} \mathcal{H}: \\ \mathrm{kl}\left(\dfrac{\widehat{\zeta_Q} - g(0)}{2c_a} + \dfrac{1}{2} \middle\| \dfrac{\zeta_Q - g(0)}{2c_a} + \dfrac{1}{2}\right) \leq \dfrac{1}{m}\log\dfrac{\xi(m)}{\delta}\end{array}\right) \geq 1 - \delta\,.$$

Nous remarquons que dans la borne du théorème 9.1.7, toute dépendance à la distribution Q se retrouve dans le calcul des risques empirique et réel. Il suit que la borne de ζ_Q déduite de ce théorème décroit lorsque le risque empirique $\widehat{\zeta_Q}$ décroit. Donc, pour trouver la distribution Q minimisant ce théorème, il suffit de trouver la distribution Q quasi-uniforme minimisant la valeur de $\widehat{\zeta_Q}$.

9.1.2 Algorithme d'optimisation

Nous associons à une distribution Q sur un ensemble $\mathcal{H}$ de classificateurs, le classificateur par vote de majorité défini par

$$f_Q(\mathbf{x}) = \mathrm{sgn}\left(\sum_{j=1}^{|\mathcal{H}|} Q(h_j)h_j(\mathbf{x})\right)\,.$$

Définition 9.1.8. *Soit Q une distribution quasi-uniforme sur $\mathcal{H} = \{h_1, \ldots, h_n, h_{n+1}, \ldots, h_{2n}\}$. Pour $j \in \{1, \ldots, n\}$ et $\lambda \in [-Q_j/2, \frac{1}{n} - Q_j/2]$ nous notons Q^j_λ la distribution sur $\mathcal{H}$ donnée par*

$$Q^j_\lambda(h_k) = \begin{cases} Q_j(h_k) & \text{si } k \neq j \text{ et } k \neq j+n \\ Q_j(h_k) + \frac{\lambda}{2} & \text{si } k = j \\ Q_j(h_k) - \frac{\lambda}{2} & \text{si } k = j+n\,. \end{cases}$$

Dans le contexte des distributions quasi-uniformes, nous supposons que l'ensemble $\mathcal{H}$ est de la forme $\mathcal{H} = \{h_1, h_2, \ldots, h_n, h_{1+n}, h_{2+n}, \ldots, h_{2n}\}$ avec $h_j = -h_{j+n}$ pour $j = 1, 2, \ldots, n$ et que la distribution Q satisfait $Q(h_j) + Q(h_{n+j}) = \frac{1}{n}$ pour $j = 1, 2, \ldots, n$.

Ainsi, nous pouvons associer à la distribution Q le vecteur de différences de poids complémentaires $\mathbf{w} = \{w_1, w_2, \ldots, w_n\}$ avec $w_j = Q(h_j) - Q(h_{j+n})\ \forall j$. En définissant maintenant $\mathbf{h}(\mathbf{x}) : \mathcal{X} \to \{-1, 1\}^n$ donnée par $\mathbf{h}(\mathbf{x}) = (h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_n(\mathbf{x}))$, nous pouvons redéfinir le classificateur par vote de majorité comme étant la fonction

$$f_{\mathbf{w}}(\mathbf{x}) = \operatorname{sgn}\left(\mathbf{w} \cdot \mathbf{h}(\mathbf{x})\right) .$$

Soit $F(Q)$ une fonction donnant une borne PAC-Bayes (du style de celle du théorème 9.1.7) d'un risque ζ_Q et définissons $F^j_{\mathbf{w}}(\lambda)$ comme étant la fonction

$$F^j_{\mathbf{w}}(\lambda) \stackrel{\text{déf}}{=} F_{Q^j_\lambda}$$

(voir la définition 9.1.8 pour un rappel de la définition de la distribution Q^j_λ). La fonction $F(Q)$ suggère l'algorithme d'apprentissage suivant : trouver la distribution quasi-uniforme Q sur $\mathcal{H}$ minimisant la borne sur le risque ζ_Q donnée par $F(Q)$. Pour une fonction F convexe, l'algorithme 3 minimise la fonction $F(Q)$ en procédant composante par composante, c'est-à-dire en travaillant sur les n fonctions $F^j_{\mathbf{w}}(\lambda)$, en les choisissant l'une après l'autre de façon aléatoire.

Algorithme 3 : Algorithme générique

Entrées : $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$, $\mathcal{H} = \{h_1, h_2, \ldots, h_n\}$

Initialiser : $w_j = 0$ pour $j = 1, 2, \ldots, n$

Exécuter

Piger j aléatoirement dans l'ensemble $\{1, 2, \ldots, n\}$.

$$\lambda_{opt} \leftarrow \operatorname*{argmin}_{\lambda \in [-\frac{1}{n} - w_j, \frac{1}{n} - wj]} \{F^j_{\mathbf{w}}(\lambda)\}$$

$$w_j \leftarrow w_j + \lambda_{opt}$$

Répéter tant que *critère d'arrêt non atteint* ;

Sortie : $f_{\mathbf{w}}(\mathbf{x}) = \operatorname{sgn}\left(\mathbf{w} \cdot \mathbf{h}(\mathbf{x})\right)$

Comme nous l'avons fait pour les distributions Q générales, nous présentons une version de l'algorithme générique (algorithme 3) valable uniquement pour des fonctions $F(Q)$ dérivables et utilisant la méthode de Newton pour trouver le minimum des différentes fonctions $F^j_{\mathbf{w}}(\lambda)$. Si $F^j_{\mathbf{w}}(\lambda)$ est convexe et différentiable, son minimum (s'il existe) est donné par la valeur λ_{opt} telle que

$$\left.\frac{\partial F^j_{\mathbf{w}}(\lambda)}{\partial \lambda}\right|_{\lambda = \lambda_{opt}} = 0 .$$

En notant $\widetilde{F}^j_{\mathbf{w}}(\lambda)$ la quantité $\frac{\partial F^j_{\mathbf{w}}(\lambda)}{\partial \lambda}$ et en utilisant la méthode de Newton pour trouver le zéro de cette fonction, nous sommes amené à l'algorithme 4.

Algorithme 4 : Algorithme générique implémentant la méthode de Newton

Entrées : $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$, $\mathcal{H} = \{h_1, h_2, \ldots, h_n\}$

Initialiser : $w_j = 0$ pour $j = 1, 2, \ldots, n$

Exécuter

Piger j aléatoirement dans l'ensemble $\{1, 2, \ldots, n\}$.

Poser $\lambda_{opt} = 0$.

Exécuter

$$\begin{array}{rcl} \lambda_{tmp} & \leftarrow & \lambda_{opt} \\ \lambda_{opt} & \leftarrow & \lambda_{opt} - \dfrac{\widetilde{F}^j_{\mathbf{w}}(\lambda_{opt})}{\frac{\partial \widetilde{F}^j_{\mathbf{w}}}{\partial \lambda}(\lambda_{opt})} \end{array}$$

Répéter tant que $|\lambda_{opt} - \lambda_{tmp}| >$ *précision voulue* ;

Si $\lambda_{opt} < -\frac{1}{n} - w_j$ **alors**

$\lambda_{opt} \leftarrow -\frac{1}{n} - w_j$

FinSi

Si $\lambda_{opt} > \frac{1}{n} - w_j$ **alors**

$\lambda_{opt} \leftarrow \frac{1}{n} - w_j$

FinSi

$w_j \leftarrow w_j + \lambda_{opt}$

Répéter tant que *critère d'arrêt non atteint* ;

Sortie : $f_{\mathbf{w}}(\mathbf{x}) = \operatorname{sgn}(\mathbf{w} \cdot \mathbf{h}(\mathbf{x}))$

Minimisation de la borne du théorème 9.1.7

Pour minimiser la borne du théorème 9.1.7 associée à une fonction de risque donnée ζ_Q, il suffit d'appliquer l'algorithme 3 en posant

$$F^j_{\mathbf{w}}(\lambda) \stackrel{\text{déf}}{=} \widehat{\zeta_{Q^j_\lambda}} \,.$$

Nous pouvons aussi appliquer l'algorithme 4, il suffit alors de calculer $\widetilde{F}^j_{\mathbf{w}}(\lambda)$ et $\frac{\partial}{\partial \lambda}\widetilde{F}^j_{\mathbf{w}}(\lambda)$, où

$$\widetilde{F}^j_{\mathbf{w}}(\lambda) \stackrel{\text{déf}}{=} \frac{\partial}{\partial \lambda} F^j_{\mathbf{w}}(\lambda) \,. \tag{9.1}$$

On remarque qu'un algorithme d'apprentissage basé sur la minimisation de la borne du théorème 9.1.7 est en fait entièrement basé sur la minimisation du risque empirique. Il semble alors qu'un tel algorithme perd toute forme de régularisation que devrait lui procurer le théorème PAC-Bayes, cependant, nous sommes ici placé dans le cadre des distributions quasi-uniformes, cadre qui confère une certaine forme de régularisation.

Minimisation de la borne du théorème 6.3.1

Minimiser la borne du théorème 6.3.1 associée à une fonction de risque donnée ζ_Q lorsque la distribution Q est contrainte à être quasi-uniforme, revient à trouver la distribution quasi-uniforme Q minimisant la fonction

$$C \cdot m \cdot \zeta_Q + \mathrm{KL}(Q\|P)\,.$$

Pour minimiser cette fonction en utilisant l'algorithme 3, il suffit de choisir la fonction $F^j_{\mathbf{w}}(\lambda)$ définie par

$$F^j_{\mathbf{w}}(\lambda) \stackrel{\text{déf}}{=} C \cdot m \cdot \widehat{\zeta_{Q^j_\lambda}} + \mathrm{KL}^j_Q(\lambda)\,,$$

où

$$\begin{aligned}
\mathrm{KL}^j_Q(\lambda) &\stackrel{\text{déf}}{=} \mathrm{KL}(Q^j_\lambda\|P) && (9.2)\\
&= \mathrm{KL}(Q\|P) + \left(Q_j + \frac{\lambda}{2}\right)\log\left(Q_j + \frac{\lambda}{2}\right) + \left(Q_{j+n} - \frac{\lambda}{2}\right)\log\left(Q_{j+n} - \frac{\lambda}{2}\right)\\
&\quad -Q_j \log Q_j - Q_{j+n}\log Q_{j+n} && (9.3)
\end{aligned}$$

En calculant les dérivées première et seconde de cette fonction $F^j_{\mathbf{w}}(\lambda)$ par rapport à λ, nous obtenons la fonction $\widetilde{F}^j_{\mathbf{w}}(\lambda)$ permettant de minimiser le théorème 6.3.1 par le biais de l'algorithme 4. Nous trouvons

$$\begin{aligned}
\widetilde{F}^j_{\mathbf{w}}(\lambda) &= C \cdot m \cdot \frac{\partial}{\partial\lambda}\widehat{\zeta_{Q^j_\lambda}} + \frac{1}{2}\log\frac{2Q_j + \lambda}{2Q_{j+n} + \lambda}\\
&= C \cdot m \cdot \frac{\partial}{\partial\lambda}\widehat{\zeta_{Q^j_\lambda}} + \frac{1}{2}\log\frac{\frac{1}{n} + w_j + \lambda}{\frac{1}{n} - w_j - \lambda} && (9.4)
\end{aligned}$$

et

$$\begin{aligned}
\frac{\partial}{\partial\lambda}\widetilde{F}^j_{\mathbf{w}}(\lambda) &= C \cdot m \cdot \frac{\partial^2}{\partial\lambda^2}\widehat{\zeta_{Q^j_\lambda}} + \frac{1}{n(2Q_j + \lambda)(2Q_{j+n} - \lambda)}\\
&= C \cdot m \cdot \frac{\partial^2}{\partial\lambda^2}\widehat{\zeta_{Q^j_\lambda}} + \frac{n}{1 - n^2(w_j + \lambda)^2}\,. && (9.5)
\end{aligned}$$

9.2 Risque linéaire

La première fonction de perte que nous examinons est la simple fonction

$$\begin{aligned}\zeta_Q(\mathbf{x}, y) &\stackrel{\text{déf}}{=} \sum_{j=1}^{2n} Q_j I(h_j(\mathbf{x}) \neq y) \\ &= \frac{1}{2} - \frac{1}{2}\sum_{j=1}^{2n} y Q_j(h_j(\mathbf{x}) \\ &= \frac{1}{2} - \frac{1}{2}\sum_{j=1}^{n} y(Q_j - Q_{j+n})(h_j(\mathbf{x}) \\ &= \frac{1}{2} - \frac{1}{2} y\mathbf{w}\cdot\mathbf{h}(\mathbf{x})\end{aligned}$$

dont le risque associé correspond au risque de Gibbs, noté $R(G_Q)$. Nous avons les égalités suivantes permettant d'exprimer les risques de Gibbs réel $R(G_Q)$ et empirique $R_S(G_Q)$ en fonction de la marge moyenne des exemples

$$\begin{aligned}R(G_Q) &= \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} \sum_{j=1}^{2n} Q_j I(h_j(\mathbf{x}) \neq y) \\ &= \frac{1}{2} - \frac{1}{2} \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} y\mathbf{w}\cdot\mathbf{h}(\mathbf{x}) \\ R_S(G_Q) &= \frac{1}{m}\sum_{j=1}^{2n} Q_j I(h_j(\mathbf{x}) \neq y) \\ &= \frac{1}{2} - \frac{1}{2m}\sum_{i=1}^{m} y_i \mathbf{w}\cdot\mathbf{h}(\mathbf{x}_i)\end{aligned}$$

Notons $\widehat{R^j_Q}(\lambda)$ le risque empirique sur l'ensemble S (préalablement choisi) associé à la distribution Q à laquelle un poids λ est transféré du classificateur $j+n$ vers le classificateur j. C'est-à-dire, $\widehat{R^j_Q}(\lambda)$ est le risque associé à la distribution Q^j_λ de la définition 9.1.8. Nous avons alors

$$\begin{aligned}\widehat{R^j_Q}(\lambda) &= \frac{1}{2} - \frac{1}{2m}\sum_{i=1}^{m}\left[y_i \mathbf{w}\cdot\mathbf{h}(\mathbf{x}_i) + y_i\lambda h_j(\mathbf{x}_i)\right] \\ &= \frac{1}{2} - \frac{1}{2m}\sum_{i=1}^{m} y_i \mathbf{w}\cdot\mathbf{h}(\mathbf{x}_i) - \frac{1}{2m}\sum_{i=1}^{m} y_i\lambda h_j(\mathbf{x}_i) \\ &= R_S(G_Q) - \frac{\lambda}{2m}\sum_{i=1}^{m} y_i h_j(\mathbf{x}_i) \\ &= R_S(G_Q) - \lambda\left(\frac{1}{2} - R_S(h_j)\right) \\ &= R_S(G_Q) + \lambda R_S(h_j) - \frac{\lambda}{2}\,.\end{aligned}$$

9.2.1 Minimisation de la borne du théorème 9.1.7

La dérivée par rapport à λ de la fonction $\widehat{R^j_Q}(\lambda)$ correspond à une constante négative si $R(h_j) < \frac{1}{2}$ et positive si $R(h_j) > \frac{1}{2}$. En effet, on calcule

$$\frac{\partial}{\partial\lambda}\widehat{R^j_Q}(\lambda) = R(h_j) - \frac{1}{2}\,.$$

Il suit que les composantes du vecteur $\mathbf{w}$ associé à la distribution Q quasi-uniforme minimisant la borne du théorème 9.1.7 appliquée au risque de Gibbs sont simplement données par

$$w_j = \begin{cases} 0 & \text{si } R(h_j) \geq \frac{1}{2} \\ \frac{1}{n} & \text{sinon } . \end{cases}$$

9.2.2 Minimisation de la borne du théorème 6.3.1

En portant les valeurs de $\frac{\partial}{\partial\lambda}\widehat{\zeta_{Q^j_\lambda}} = R(h_j) - \frac{1}{2}$ et $\frac{\partial^2}{\partial\lambda^2}\widehat{\zeta_{Q^j_\lambda}} = 0$ respectivement dans les équations (9.4) et (9.5), nous obtenons les fonctions $\widetilde{F}^j_{\mathbf{w}}(\lambda)$ et $\frac{\partial}{\partial\lambda}\widetilde{F}^j_{\mathbf{w}}(\lambda)$ qu'il faut porter dans l'algorithme 4 pour obtenir un algorithme de minimisation de la borne donnée par le théorème 6.3.1 sur le risque de Gibbs (pour des distributions quasi-uniformes).

Plus précisément, les fonctions à porter dans l'algorithme sont données par

$$\widetilde{F}^j_{\mathbf{w}}(\lambda) = C \cdot m \cdot \left(R_S(h_j) - \frac{1}{2}\right) + \frac{1}{2}\log\frac{\frac{1}{n} + w_j + \lambda}{\frac{1}{n} - w_j - \lambda}$$

et

$$\frac{\partial}{\partial\lambda}\widetilde{F}^j_{\mathbf{w}}(\lambda) = \frac{n}{1 - n(w_j + \lambda)^2}\,.$$

9.2.3 Résultats

Le tableau 9.2.3 présente les résultats obtenus par trois algorithmes, chacun concevant un vote de majorité pondéré par une distribution Q quasi-uniforme optimisant une borne sur le risque de Gibbs : G_Q-kl dénote l'algorithme minimisant la borne du théorème 9.1.7, alors que G_Q-C-vc et G_Q-C-bm dénote deux versions de l'algorithme minimisant la borne du théorème 6.3.1. Pour chacune de ces deux versions de l'algorithme, nous avons testé un jeu de 16 valeurs différentes du paramètre C (0.01, 0.02, 0,05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100, 200, 500 et 1000), dans la version G_Q-C-vc le paramètre C permettant de construire le classificateur final est choisi par validation

Ensemble				(1) G_Q-**kl**		(2) G_Q-**C-vc**		(3) G_Q-**C-bm**			**SSM**
Nom	$\|S\|$	$\|T\|$	n	R_T	Borne	R_T	C	R_T	C	Borne	
Adult	1809	10000	14	**0.248**	0.374	**0.248**	0.001	**0.248**	0.1	0.363	
BreastCancer	343	340	9	0.171	0.269	**0.126**	0.001	0.171	0.5	0.253	
Credit-A	353	300	15	0.417	0.487	**0.347**	0.001	0.420	0.2	0.471	
Glass	107	107	9	**0.411**	0.586	**0.411**	0.001	**0.411**	0.5	0.556	
Haberman	144	150	3	**0.273**	0.460	**0.273**	0.001	**0.273**	0.5	0.431	
Heart	150	147	13	0.422	0.551	**0.265**	0.001	0.435	0.5	0.524	$(2) < (1,3)$
Ionosphere	176	175	34	**0.337**	0.492	**0.337**	0.001	**0.337**	0.5	0.466	
Letter:AB	500	1055	16	0.449	0.483	**0.050**	0.001	0.423	0.2	0.465	$(2) < (1,3)$
Letter:DO	500	1058	16	0.490	0.535	**0.257**	0.001	0.490	0.2	0.516	$(2) < (1,3)$
Letter:OQ	500	1036	16	**0.156**	0.534	0.208	0.001	0.182	0.2	0.515	$(1) < (2)$
Liver	170	175	6	0.411	0.581	**0.400**	0.001	**0.400**	0.5	0.556	
MNIST:0vs8	500	1916	784	0.257	0.484	**0.034**	0.001	0.252	0.2	0.466	$(2) < (1,3)$
MNIST:1vs7	500	1922	784	0.499	0.475	**0.267**	0.001	0.499	0.2	0.457	$(2) < (1,3)$
MNIST:1vs8	500	1936	784	0.320	0.495	**0.270**	0.001	0.367	0.2	0.476	$(1) < (3)$,$(2) < (1,3)$
MNIST:2vs3	500	1905	784	0.491	0.507	**0.148**	0.001	0.483	0.2	0.489	$(2) < (1,3)$
Mushroom	4062	4062	22	0.145	0.442	0.142	0.001	**0.141**	0.1	0.433	
Ringnorm	3700	3700	20	**0.511**	0.499	**0.511**	0.5	**0.511**	0.1	0.489	
Sonar	104	104	60	**0.490**	0.605	**0.490**	0.001	**0.490**	0.5	0.574	
Usvotes	235	200	16	0.160	0.364	**0.130**	0.001	0.135	0.5	0.342	
Waveform	4000	4000	21	0.450	0.473	**0.182**	0.001	0.403	0.1	0.464	$(2) < (1,3)$,$(3) < (1)$
Wdbc	285	284	30	0.359	0.447	**0.324**	0.001	0.359	0.5	0.430	

TABLE 9.1 – Comparaison de trois algorithmes d'apprentissage basés sur la minimisation d'une borne sur le risque de Gibbs avec contrainte de distribution quasi-uniforme.

croisée, alors que dans la version G_Q-C-bm, c'est le paramètre permettant d'obtenir la plus petite borne de $R(G_Q)$ qui a été choisi.

Sans surprise, tous ces algorithmes d'apprentissage offrent de piètres performances, en fait, sur bien des ensembles (dont Adult,Ringmorn et Wdbc), les résultats ici obtenus sont même significativement moins bons que ceux obtenus par les algorithmes non contraints à concevoir une distribution Q quasi-uniforme (voir section 8.3). Les risques sur l'ensemble test obtenus sur Ringnorm sont mêmes supérieurs à 50%. Il ne s'agit pas d'une erreur dans le tableau, c'est plutôt un bon exemple du fait que le risque de Gibbs peut être inférieur au risque du vote de majorité. On remarque aussi que sur ce même exemple, les bornes présentées sont plus faibles que le risque sur l'ensemble test, les bornes ne sont toutefois pas violées, car il s'agit de bornes sur le risque de Gibbs ; il faut multiplier par 2 ces bornes pour obtenir des bornes sur le vote de majorité.

9.3 Risque quadratique

Pour un hyperparamètre γ fixé, la perte quadratique dans le contexte des distributions quasi-uniformes est donnée par

$$\zeta_{\mathbf{w}}(\mathbf{x}, y) = \left(\frac{y\mathbf{w}\cdot\mathbf{h}(\mathbf{x})}{\gamma} - 1\right)^2 ,$$

il suit que nous avons

$$\begin{aligned}\widehat{\zeta^j_{\mathbf{w}}}(\lambda) &= \frac{1}{m}\sum_{i=1}^{m}\left(\frac{y_i\mathbf{w}\cdot\mathbf{h}(\mathbf{x}_i) + \lambda y_i h_j(\mathbf{x}_i)}{\gamma} - 1\right)^2 \\ &= \widehat{\zeta_{\mathbf{w}}} + \frac{2\lambda}{m\gamma^2}\left(\sum_{i=1}^{m} y_i h_j(\mathbf{x}_i)(y_i\mathbf{w}\cdot\mathbf{h}(\mathbf{x}_i) - \gamma)\right) + \frac{\lambda^2}{\gamma^2}\,.\end{aligned}$$

En posant $D^j_{\mathbf{w}} = \frac{1}{m}\sum_{i=1}^{m} y_i h_j(\mathbf{x}_i)(y_i\mathbf{w}\cdot\mathbf{h}(\mathbf{x}_i) - \gamma)$, nous pouvons écrire de façon plus succincte

$$\widehat{\zeta^j_{\mathbf{w}}}(\lambda) = \widehat{\zeta_{\mathbf{w}}} + \frac{2\lambda}{\gamma^2}D^j_{\mathbf{w}} + \frac{\lambda^2}{\gamma^2}\,.$$

Les dérivées première et seconde de cette fonction par rapport à λ sont données par

$$\begin{aligned}\frac{\partial}{\partial\lambda}\widehat{\zeta^j_{\mathbf{w}}}(\lambda) &= \frac{2}{\gamma^2}D^j_{\mathbf{w}} + \frac{2\lambda}{\gamma^2} \\ \frac{\partial^2}{\partial\lambda^2}\widehat{\zeta^j_{\mathbf{w}}}(\lambda) &= \frac{2}{\gamma^2}\,.\end{aligned}$$

9.3.1 Algorithmes d'apprentissage

Minimisation de la borne du théorème 9.1.7

Pour minimiser la borne du théorème 9.1.7 nous devons trouver la distribution Q quasi-uniforme minimisant le risque quadratique, en procédant composante par composante, nous devons donc minimiser des quantités $\widehat{\zeta^j_{\mathbf{w}}}(\lambda)$, ou de façon similaire (puisque cette fonction est convexe) nous devons résoudre

$$\frac{\partial}{\partial\lambda}\widehat{\zeta^j_{\mathbf{w}}}(\lambda) = 0\,,$$

ce qui se fait analytiquement. On obtient

$$\frac{\partial}{\partial\lambda}\widehat{\zeta^j_{\mathbf{w}}}(\lambda)\bigg|_{\lambda=\lambda_{opt}} = 0 \iff \lambda_{opt} = -D^j_{\mathbf{w}}.$$

Ainsi, nous pouvons concevoir, pour résoudre ce problème d'optimisation, un algorithme semblable à l'algorithme 4, mais dans lequel l'application de la méthode de Newton est remplacée par une expression directe du zéro de la fonction $\widehat{\zeta^j_{\mathbf{w}}}(\lambda)$. Voir l'algorithme 5 pour plus de détails.

Algorithme 5 : Minimisation de la borne du théorème 9.1.7

Entrées : $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$, $\mathcal{H} = \{h_1, h_2, \ldots, h_n\}$,$\gamma \in (0, 1]$

Initialiser : $w_j = 0$ pour $j = 1, 2, \ldots, n$ et $M_i = 0$ pour $i = 1, 2, \ldots, m$

Exécuter

Piger j aléatoirement dans l'ensemble $\{1, 2, \ldots, n\}$.

Poser $\lambda_{opt} = -\frac{1}{m}\sum_{i=1}^{m} y_i h_j(\mathbf{x}_i)(M_i - \gamma)$.

$\lambda_{opt} \leftarrow \min(\lambda_{opt}, \frac{1}{n} - w_j)$

$\lambda_{opt} \leftarrow \max(\lambda_{opt}, -\frac{1}{n} - w_j)$

$w_j \leftarrow w_j + \lambda_{opt}$

$\forall i : M_i \leftarrow M_i + y_i \lambda_{opt} h_j(\mathbf{x}_i)$

Répéter tant que *critère d'arrêt non atteint* ;

Sortie : $f_{\mathbf{w}}(\mathbf{x}) = \operatorname{sgn}(\mathbf{w} \cdot \mathbf{h}(\mathbf{x}))$

Minimisation de la borne du théorème 6.3.1

En portant dans les équations (9.4) et (9.5) les expressions que nous avons calculées pour les dérivées de $\widehat{\zeta^j_{\mathbf{w}}}(\lambda)$, nous obtenons les fonctions qu'il faut porter dans l'algorithme 4 pour obtenir un algorithme d'apprentissage retournant un classificateur par vote de majorité pondéré par la distribution Q quasi-uniforme minimisant la borne du théorème 6.3.1 appliqué avec le risque quadratique :

$$\widetilde{F}^j_{\mathbf{w}}(\lambda) \quad = \quad \frac{2Cm}{\gamma^2}\lambda + \frac{2Cm}{\gamma^2} D^j_{\mathbf{w}} + \frac{1}{2}\log\frac{\frac{1}{n} + w_j + \lambda}{\frac{1}{n} - w_j - \lambda}$$

et

$$\frac{\partial}{\partial\lambda}\widetilde{F}^j_{\mathbf{w}}(\lambda) \quad = \quad \frac{2Cm}{\gamma^2} + \frac{n}{1 - n^2(w_j + \lambda)^2}\,.$$

Temps d'exécution

Chaque étape d'optimisation (pour une composante j donnée) nécessite de calculer la quantité $D^j_{\mathbf{w}}$, ce qui nécessite un temps de calcul de l'ordre de $\Theta(m)$ si nous gardons en mémoire un vecteur contenant la marge des différents exemples (ce vecteur correspond à

M dans l'algorithme 5). Une fois ce calcul fait, les évaluations des différentes fonctions se font dans un temps constant, c'est-à-dire qui ne dépend ni de m (le nombre d'exemples), ni de n (le nombre de classificateurs). Une fois le transfert de poids optimal (λ_{opt}) trouvé, il est nécessaire de mettre à jour le vecteur de marges, ce qui se fait dans un temps de l'ordre de $\Theta(m)$. Ainsi, pour les deux versions de l'algorithme, chaque étape d'optimisation a un cout total en temps de calcul de l'ordre de $\Theta(m)$. Cependant, dans la version de l'algorithme basée sur le théorème 5.1.1, à chaque itération de l'algorithme, la recherche du zéro se fait par une méthode analytique directe, alors que pour la version basée sur le théorème 6.3.1, la recherche du zéro se fait avec une méthode itérative, qui est la méthode de Newton. Ainsi, en notant N le nombre maximal d'itérations exécuté par la méthode de Newton, nous pouvons dire que chaque itération de la deuxième version de l'algorithme s'exécute dans un temps de l'ordre $O(N \cdot m)$. Dans la pratique, la valeur maximale de N étant un nombre entier fixé (en fonction de la précision voulue dans l'implémentation de l'algorithme), chaque itération de l'algorithme s'exécute tout de même dans temps de l'ordre de $O(m)$.

9.3.2 Résultats

Le tableau 9.2 présente des résultats d'expérimentations de deux algorithmes d'apprentissage construisant des votes de majorité pondérés par des distributions minimisant une borne de type PAC-Bayes du risque quadratique. L'algorithme identifié par Quad-C est basé sur la version Catoni du théorème sur les fonctions de perte générales, alors que l'algorithme Quad-kl est basé sur la version Langford-Seeger.

Nous avons comparé nos algorithmes avec AdaBoost (AB) et la régression ridge (RR) sur 21 ensembles de données (la méthodologie employée pour les comparaisons est décrite à la section 8.2). Les valeurs testées pour le paramètre γ correspondent à l'ensemble $\{0.0001, 0.0002,\ 0.0005, 0.001, 0.002, 0.005,\ 0.01, 0.02, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5,\ 0.6,\ 0.7,\ 0.8,\ 0.9, 1\}$ et celles du paramètre C à l'ensemble $\{0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10,\ 20,\ 50,\ 100,\ 200,\ 500, 1000\}$. La colonne SSM indique qu'il y a une différence statistiquement significative de performance entre deux algorithmes sur seulement 2 ensembles de données : sur MNIST :0vs8, AdaBoost est significativement meilleur que Quad-kl, alors que sur MNIST :2vs3, Quad-C est significativement meilleur que AdaBoost. La méthodologie employée pour les comparaisons est donnée à la section 8.2.

Ensemble				(1) **AB**	(2) **RR**		(3) **Quad-C**			(4) **Quad-kl**		**SSM**
Nom	$\|S\|$	$\|T\|$	n	R_T	R_T	C	R_T	γ	C	R_T	γ	
Adult	1809	10000	14	0.149	**0.148**	0.2	0.149	0.02	0.05	**0.148**	0.02	
BreastCancer	343	340	9	0.053	0.050	10	**0.044**	0.3	0.1	**0.044**	0.2	
Credit-A	353	300	15	0.170	0.157	2	**0.147**	0.02	0.01	**0.147**	0.02	
Glass	107	107	9	**0.178**	0.206	5	0.196	0.01	0.02	0.224	0.02	
Haberman	144	150	3	**0.260**	0.273	20	0.267	0.4	0.5	**0.260**	0.6	
Heart	150	147	13	0.252	0.197	1	**0.177**	0.2	0.05	0.190	0.2	
Ionosphere	176	175	34	0.120	0.131	0.05	**0.097**	0.1	0.2	0.114	0.1	
Letter:AB	500	1055	16	0.010	**0.003**	0.2	0.006	0.1	0.01	0.006	0.1	
Letter:DO	500	1058	16	0.036	0.026	0.05	**0.020**	0.05	0.02	0.021	0.05	
Letter:OQ	500	1036	16	**0.038**	0.044	0.2	0.042	0.05	0.01	0.047	0.05	
Liver	170	175	6	0.320	**0.309**	5	0.337	0.1	10	0.337	0.1	
MNIST:0vs8	500	1916	784	**0.008**	0.015	0.05	0.015	0.005	0.01	0.018	0.02	(1) < (4)
MNIST:1vs7	500	1922	784	0.013	0.011	0.05	**0.010**	0.05	0.01	0.014	0.1	
MNIST:1vs8	500	1936	784	0.025	**0.024**	0.2	**0.024**	0.05	0.01	0.029	0.05	
MNIST:2vs3	500	1905	784	0.047	0.033	0.2	**0.029**	0.05	0.01	0.035	0.1	(3) < (1)
Mushroom	4062	4062	22	**0.000**	**0.000**	0.02	**0.000**	0.0001	0.01	**0.000**	0.0001	
Ringnorm	3700	3700	20	0.043	**0.037**	0.05	0.039	0.05	0.05	0.038	0.02	
Sonar	104	104	60	0.231	0.192	0.05	**0.125**	0.2	1	**0.125**	0.2	
Usvotes	235	200	16	**0.055**	0.060	2	**0.055**	0.0001	0.01	**0.055**	0.0001	
Waveform	4000	4000	21	0.085	0.079	10	**0.078**	0.1	0.01	0.079	0.05	
Wdbc	285	284	30	0.049	0.049	0.2	**0.046**	0.1	0.1	**0.046**	0.1	

TABLE 9.2 – Résultats d'expérimentations avec des algorithmes d'apprentissage basés sur une borne de type PAC-Bayes sur le risque quadratique.

9.4 Risque exponentiel

La perte exponentielle sur un exemple $(\mathbf{x}, y)$ dans le contexte des distributions quasi-uniformes est donnée par

$$\mathcal{E}_{\mathbf{w}}(\mathbf{x}, y) = \exp\left(-\frac{y\mathbf{w}\cdot\mathbf{h}(\mathbf{x})}{\gamma}\right) .$$

Le risque exponentiel, noté $\mathcal{E}_{\mathbf{w}}$, et son risque empirique sur un ensemble S de taille m, noté $\widehat{\mathcal{E}_{\mathbf{w}}}$, sont donc donnés par

$$\mathcal{E}_{\mathbf{w}} = \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} \exp\left(-\frac{y\mathbf{w}\cdot\mathbf{h}(\mathbf{x})}{\gamma}\right) \quad \text{et} \quad \widehat{\mathcal{E}_{\mathbf{w}}} = \frac{1}{m}\sum_{i=1}^{m} \exp\left(-\frac{y_i\mathbf{w}\cdot\mathbf{h}(\mathbf{x}_i)}{\gamma}\right) .$$

Pour un vecteur de différences de poids complémentaires $\mathbf{w}$ donné (relié à une distribution quasi-uniforme Q), notons $\widehat{\mathcal{E}^j_{\mathbf{w}}}(\lambda)$ la fonction retournant le risque exponentiel

empirique associé à la distribution Q^j_λ, c'est-à-dire

$$\begin{aligned}\widehat{\mathcal{E}^j_{\mathbf{w}}}(\lambda) &= \frac{1}{m}\sum_{i=1}^{m}\exp\left(-\frac{y_i\mathbf{w}\cdot\mathbf{h}(\mathbf{x}_i)}{\gamma} - \frac{\lambda y_i h_j(\mathbf{x}_i)}{\gamma}\right)\\ &= \frac{1}{m}\sum_{i=1}^{m}\exp\left(-\frac{y_i\mathbf{w}\cdot\mathbf{h}(\mathbf{x}_i)}{\gamma}\right)\exp\left(-\frac{\lambda y_i h_j(\mathbf{x}_i)}{\gamma}\right)\\ &= \frac{1}{m}\exp\left(-\frac{\lambda}{\gamma}\right)\sum_{i=1}^{m}\exp\left(-\frac{y_i\mathbf{w}\cdot\mathbf{h}(\mathbf{x}_i)}{\gamma}\right)I(h_j(\mathbf{x}_i)=y_i)\\ &\quad+\frac{1}{m}\exp\left(\frac{\lambda}{\gamma}\right)\sum_{i=1}^{m}\exp\left(-\frac{y_i\mathbf{w}\cdot\mathbf{h}(\mathbf{x}_i)}{\gamma}\right)I(h_j(\mathbf{x}_i)=-y_i)\,.\end{aligned}$$

En posant maintenant

$$\begin{aligned}D^+_{\mathbf{w}} &= \frac{1}{m}\sum_{i=1}^{m}\exp\left(-\frac{y_i\mathbf{w}\cdot\mathbf{h}(\mathbf{x}_i)}{\gamma}\right)I(h_j(\mathbf{x}_i)=y_i)\\ D^-_{\mathbf{w}} &= \frac{1}{m}\sum_{i=1}^{m}\exp\left(-\frac{y_i\mathbf{w}\cdot\mathbf{h}(\mathbf{x}_i)}{\gamma}\right)I(h_j(\mathbf{x}_i)=-y_i)\,,\end{aligned}$$

nous pouvons écrire

$$\widehat{\mathcal{E}^j_{\mathbf{w}}}(\lambda) = D^+_{\mathbf{w}}\exp\left(-\frac{\lambda}{\gamma}\right) + D^-_{\mathbf{w}}\exp\left(\frac{\lambda}{\gamma}\right)\,. \tag{9.6}$$

Nous avons donc les expressions suivantes pour les dérivées première et seconde du risque exponentiel

$$\frac{\partial\widehat{\mathcal{E}^j_{\mathbf{w}}}(\lambda)}{\partial\lambda} = \frac{-D^+_{\mathbf{w}}}{\gamma}\exp\left(-\frac{\lambda}{\gamma}\right) + \frac{D^-_{\mathbf{w}}}{\gamma}\exp\left(\frac{\lambda}{\gamma}\right) \tag{9.7}$$

et

$$\frac{\partial^2\widehat{\mathcal{E}^j_{\mathbf{w}}}(\lambda)}{\partial\lambda^2} = \frac{D^+_{\mathbf{w}}}{\gamma^2}\exp\left(-\frac{\lambda}{\gamma}\right) + \frac{D^-_{\mathbf{w}}}{\gamma^2}\exp\left(\frac{\lambda}{\gamma}\right)\,. \tag{9.8}$$

9.4.1 Algorithmes d'optimisation

Minimisation de la borne du théorème 9.1.7

Pour obtenir un algorithme d'apprentissage retournant la distribution Q quasi-uniforme minimisant la borne du théorème 9.1.7 appliqué avec le risque exponentiel, il suffit de porter dans l'algorithme 4 les valeurs suivantes de $\widetilde{F}^j_{\mathbf{w}}(\lambda)$ et $\frac{\partial}{\partial\lambda}\widetilde{F}^j_{\mathbf{w}}(\lambda)$:

$$\widetilde{F}^j_{\mathbf{w}}(\lambda) = \frac{\partial\widehat{\mathcal{E}^j_{\mathbf{w}}}(\lambda)}{\partial\lambda}$$

et

$$\frac{\partial}{\partial\lambda}\widetilde{F}^j_{\mathbf{w}}(\lambda) = \frac{\partial^2\widehat{\mathcal{E}^j_{\mathbf{w}}}(\lambda)}{\partial\lambda^2}\,,$$

où $\frac{\partial\widehat{\mathcal{E}^j_{\mathbf{w}}}(\lambda)}{\partial\lambda}$ et $\frac{\partial^2\widehat{\mathcal{E}^j_{\mathbf{w}}}(\lambda)}{\partial\lambda^2}$ sont respectivement donnés aux équations 9.7 et 9.8.

Minimisation de la borne du théorème 6.3.1

Pour obtenir un algorithme d'apprentissage retournant la distribution Q quasi-uniforme minimisant la borne du théorème 6.3.1 appliqué avec le risque exponentiel, il suffit de porter dans l'algorithme 4 les valeurs suivantes de $\widetilde{F}^j_{\mathbf{w}}(\lambda)$ et $\frac{\partial}{\partial\lambda}\widetilde{F}^j_{\mathbf{w}}(\lambda)$:

$$\widetilde{F}^j_{\mathbf{w}}(\lambda) = C\cdot m\cdot\frac{\partial\widehat{\mathcal{E}^j_{\mathbf{w}}}(\lambda)}{\partial\lambda} + \frac{1}{2}\log\frac{\frac{1}{n}+w_j+\lambda}{\frac{1}{n}-w_j-\lambda}$$

et

$$\frac{\partial}{\partial\lambda}\widetilde{F}^j_{\mathbf{w}}(\lambda) = C\cdot m\cdot\frac{\partial^2\widehat{\mathcal{E}^j_{\mathbf{w}}}(\lambda)}{\partial\lambda^2} + \frac{n}{1-n^2(w_j+\lambda)^2}\,,$$

où $\frac{\partial\widehat{\mathcal{E}^j_{\mathbf{w}}}(\lambda)}{\partial\lambda}$ et $\frac{\partial^2\widehat{\mathcal{E}^j_{\mathbf{w}}}(\lambda)}{\partial\lambda^2}$ sont respectivement donnés aux équations 9.7 et 9.8.

Temps d'exécution

Pour une composante j donnée, une fois les calculs de $D^+_{\mathbf{w}}$ et $D^-_{\mathbf{w}}$ effectués, ce qui se fait en temps $\Theta(m)$ en supposant que nous tenons à jour un vecteur de marges, c'est-à-dire un vecteur contenant les m valeurs de $y_i\mathbf{w}\cdot\mathbf{h}(\mathbf{x}_i)$, les évaluations de $\widetilde{F}^j_{\mathbf{w}}(\lambda)$ et $\frac{\partial}{\partial\lambda}\widetilde{F}^j_{\mathbf{w}}(\lambda)$ se font en temps constant. Une fois le transfert de poids optimal (λ_{opt}) trouvé, il est nécessaire de mettre à jour le vecteur de marges, ce qui se fait dans un temps de l'ordre de $\Theta(m)$. Ainsi, pour les deux versions de l'algorithme, chaque étape d'optimisation a un cout total en temps de calcul de l'ordre de $\Theta(m)$. Le temps d'exécution, par itération de l'algorithme, est donc le même que celui de l'algorithme basé sur la perte quadratique.

9.4.2 Résultats

Le tableau 9.3 présente des résultats d'expérimentations de deux algorithmes d'apprentissage construisant des votes de majorité pondérés par des distributions quasi-uniformes minimisant une borne de type PAC-Bayes du risque exponentiel. L'algorithme

identifié par Exp-C est basé sur la version Catoni du théorème sur les fonctions de perte générales, alors que l'algorithme Exp-kl est basé sur la version Langford-Seeger.

Nous avons comparé nos algorithmes avec AdaBoost (AB) et la régression ridge (RR) sur 21 ensembles de données (la méthodologie employée pour les comparaisons est décrite à la section 8.2). Les valeurs paramètres γ et C permettant d'obtenir les résultats présentés ont été choisis par validation croisée. Les valeurs testées pour le paramètre γ correspondent à l'ensemble $\{0.0001, 0.0002,\ 0.0005, 0.001, 0.002, 0.005,\ 0.01, 0.02, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5,\ 0.6,\ 0.7,\ 0.8,\ 0.9, 1\}$ et celles du paramètre C à l'ensemble $\{0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20,\ 50,\ 100,\ 200,\ 500, 1000\}$. La colonne SSM indique qu'il n'y a aucune différence statistiquement significative entre les différents algorithmes, excepté sur l'ensemble Ringnorm où les deux versions de nos algorithmes sont significativement meilleurs qu'AdaBoost et également significativement meilleurs que la régression ridge.

Ensemble				(1) **AB**	(2) **RR**		(3) **Exp-C**			(4) **Exp-kl**		**SSM**
Nom	$\lvert S\rvert$	$\lvert T\rvert$	n	R_T	R_T	C	R_T	γ	C	R_T	γ	
Adult	1809	10000	14	0.149	**0.148**	0.2	0.149	0.02	0.02	0.150	0.02	
BreastCancer	343	340	9	0.053	0.050	10	**0.044**	0.1	0.02	0.050	0.2	
Credit-A	353	300	15	0.170	**0.157**	2	0.227	0.1	0.02	0.203	0.1	
Glass	107	107	9	**0.178**	0.206	5	**0.178**	0.01	0.01	0.187	0.01	
Haberman	144	150	3	**0.260**	0.273	20	0.267	0.5	0.5	**0.260**	0.5	
Heart	150	147	13	0.252	**0.197**	1	0.204	0.1	0.2	0.211	0.1	
Ionosphere	176	175	34	0.120	0.131	0.05	**0.109**	0.01	0.05	0.114	0.05	
Letter:AB	500	1055	16	0.010	0.003	0.2	**0.002**	0.002	20	0.004	0.0001	
Letter:DO	500	1058	16	0.036	0.026	0.05	0.024	0.005	0.01	**0.019**	0.01	
Letter:OQ	500	1036	16	**0.038**	0.044	0.2	0.042	0.0001	0.1	0.052	0.0001	
Liver	170	175	6	0.320	**0.309**	5	0.360	0.02	0.5	0.320	0.05	
MNIST:0vs8	500	1916	784	**0.008**	0.015	0.05	0.016	0.005	10	0.015	0.0002	
MNIST:1vs7	500	1922	784	0.013	0.011	0.05	**0.010**	0.005	1000	0.014	0.001	
MNIST:1vs8	500	1936	784	0.025	0.024	0.2	**0.022**	0.005	1	0.023	0.0002	
MNIST:2vs3	500	1905	784	0.047	**0.033**	0.2	0.037	0.002	100	0.036	0.005	
Mushroom	4062	4062	22	**0.000**	**0.000**	0.02	**0.000**	0.0001	0.01	**0.000**	0.0001	
Ringnorm	3700	3700	20	0.043	0.037	0.05	**0.025**	0.01	0.02	0.026	0.005	$(3,4) < (1,2)$
Sonar	104	104	60	0.231	0.192	0.05	**0.135**	0.005	0.2	0.163	0.0001	
Usvotes	235	200	16	**0.055**	0.060	2	0.085	0.02	0.01	0.060	0.05	
Waveform	4000	4000	21	0.085	**0.079**	10	0.082	0.05	0.01	0.080	0.05	
Wdbc	285	284	30	0.049	0.049	0.2	0.046	0.02	0.2	**0.042**	0.02	

TABLE 9.3 – Résultats d'expérimentations avec des algorithmes d'apprentissage basés sur une borne de type PAC-Bayes sur le risque exponentiel.

9.5 Classificateur de Gibbs à piges multiples

Dans cette section, nous nous intéressons à concevoir des algorithmes construisant un classificateur par vote de majorité pondéré par une distribution quasi-uniforme minimisant une borne du risque du classificateur de Gibbs à piges multiples. Ces algorithmes

seront en fait des implémentations de l'algorithme 4, il nous suffit alors de définir les fonctions $\widetilde{F}_Q^{j,k}(\lambda)$ propres à chaque algorithme. Nous présentons ici deux algorithmes, l'un basé sur la borne fournie par le théorème 9.1.7, et l'autre basé sur le théorème 6.3.1.

Comme nous l'avons fait dans la section 8.6, nous avons en fait minimisé non pas directement le risque associé à la fonction de perte du classificateur de Gibbs à piges multiples, mais plutôt le risque associé à une version convexifiée de celle ci, soit la fonction de perte suivante

$$\mathcal{R}_{(\mathbf{x}_i,y_i)}(G_{(Q_\lambda^j)^N}) = \begin{cases} R_N\left(\frac{1}{2} - \frac{y_i}{2}\left(\mathbf{w}\cdot\mathbf{h}(\mathbf{x}_i) + \lambda h_j(\mathbf{x}_i)\right)\right) & \text{si } y_i\mathbf{w}\cdot\mathbf{h}(\mathbf{x}_i) < 0 \\ \frac{1}{2} - \frac{y_i}{2}\left(\mathbf{w}\cdot\mathbf{h}(\mathbf{x}_i) + \lambda h_j(\mathbf{x}_i)\right)\cdot R'_N\left(\frac{1}{2}\right) & \text{sinon}\,. \end{cases}$$

Nous avons choisi de ne pas détailler les calculs nécessaires à l'adaptation des algorithmes de la section 8.6 pour les distributions quasi-uniformes car ceux-ci sont un peu long et se font sans trop de difficultés. Nous présentons alors pour ces algorithmes seulement les résultats expérimentaux.

9.5.1 Résultats

Le tableau 9.4 présente des résultats d'expérimentations de deux algorithmes d'apprentissage construisant des votes de majorité pondérés par des distributions quasi-uniformes minimisant une borne de type PAC-Bayes du risque exponentiel. L'algorithme identifié par GibbsN-C est basé sur la version Catoni du théorème sur les fonctions de perte générales, alors que l'algorithme GibbsN-kl est basé sur la version Langford-Seeger.

Nous avons comparé nos algorithmes avec AdaBoost (AB) et la régression ridge (RR) sur 21 ensembles de données (la méthodologie employée pour les comparaisons est décrite à la section 8.2). Les valeurs paramètres N et C permettant d'obtenir les résultats présentés ont été choisis par validation croisée. Les valeurs testées pour le paramètre N correspondent à l'ensemble $\{1, 3, 5, 7, 9, 25, 49, 75, 99, 499, 999, 4999, 9999, 49999, 99999, 499999, 999999\}$ et celles du paramètre C à l'ensemble $\{0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100, 200, 500, 1000\}$. La colonne SSM indique qu'il n'y a aucune différence statistiquement significative entre les différents algorithmes, excepté sur l'ensemble Ringnorm où les deux versions de nos algorithmes sont significativement meilleurs qu'AdaBoost et également significativement meilleurs que la régression ridge.

Ensemble				(1) **AB**	(2) **RR**		(3) **GibbsN-C**			(4) **GibbsN-kl**		**SSM**
Nom	$\|S\|$	$\|T\|$	n	R_T	R_T	C	R_T	N	C	R_T	N	
Adult	1809	10000	14	0.149	**0.148**	0.2	0.152	9999	20	0.153	9999	
BreastCancer	343	340	9	0.053	0.050	10	0.041	75	10	**0.038**	75	
Credit-A	353	300	15	0.170	0.157	2	**0.153**	99999	2	0.207	99999	
Glass	107	107	9	**0.178**	0.206	5	**0.178**	49999	500	**0.178**	49999	
Haberman	144	150	3	**0.260**	0.273	20	0.273	1	0.001	0.273	1	
Heart	150	147	13	0.252	0.197	1	**0.184**	999	2	**0.184**	99	
Ionosphere	176	175	34	0.120	0.131	0.05	**0.097**	4999	50	**0.097**	499	
Letter:AB	500	1055	16	0.010	**0.003**	0.2	0.005	499	5	0.009	499	
Letter:DO	500	1058	16	0.036	**0.026**	0.05	**0.026**	999	10	**0.026**	499999	
Letter:OQ	500	1036	16	**0.038**	0.044	0.2	0.039	49999	100	0.054	499999	
Liver	170	175	6	0.320	**0.309**	5	0.314	999	2	0.331	4999	
MNIST:0vs8	500	1916	784	**0.008**	0.015	0.05	0.010	4999	50	0.013	999999	
MNIST:1vs7	500	1922	784	0.013	**0.011**	0.05	0.018	499	50	0.017	499	
MNIST:1vs8	500	1936	784	0.025	0.024	0.2	**0.016**	49999	500	0.023	49999	
MNIST:2vs3	500	1905	784	0.047	**0.033**	0.2	0.035	499999	1000	0.036	4999	
Mushroom	4062	4062	22	**0.000**	**0.000**	0.02	**0.000**	49999	100	**0.000**	99999	
Ringnorm	3700	3700	20	0.043	0.037	0.05	0.027	9999	200	**0.026**	9999	$(3,4) < (1,2)$
Sonar	104	104	60	0.231	0.192	0.05	0.144	4999	100	**0.125**	99	
Usvotes	235	200	16	**0.055**	0.060	2	0.070	4999	2	0.080	99	
Waveform	4000	4000	21	0.085	**0.079**	10	0.081	999	0.5	**0.079**	999	
Wdbc	285	284	30	0.049	0.049	0.2	**0.046**	999	200	**0.046**	999	

TABLE 9.4 – Résultats d'expérimentations avec des algorithmes d'apprentissage basés sur une borne de type PAC-Bayes sur le risque du classificateur de Gibbs à piges multiples.

Conclusion

Dans cette thèse, nous avons présenté deux nouveaux théorèmes de type PAC-Bayes ainsi que différents résultats théoriques leur étant liés. Puis, nous avons développé des algorithmes d'apprentissage construisant des classificateurs par vote de majorité pondérés par une distribution Q minimisant une borne de type PAC-Bayes.

Le théorème 4.7.1 permet de borner, en plus du risque de Gibbs, diverses quantités statistiques reliées à un vote de majorité, telles que la variance du taux d'erreur des votants (alors que le théorème PAC-Bayes classique permet seulement de borne le risque de Gibbs). En particulier, ce théorème s'applique pour borner une quantité (C_Q) plus représentative du risque d'un classificateur par vote de majorité que ne l'est le risque de Gibbs (qui ne correspond en fait qu'à la moyenne des risques de votants).

Le théorème 5.1.1 (comme les théorèmes 5.1.2 et 6.3.1) permet de borner le risque associé à toute fonction de perte s'appliquant à un classificateur par vote de majorité et possédant un développement en série de Taylor centré en $W_Q(\mathbf{x}, y) = \frac{1}{2}$ et défini sur tout l'intervalle $[0, 1]$. En particulier, ce théorème s'applique pour obtenir des bornes des risques linéaire (qui coïncide avec le risque de Gibbs), quadratique et exponentielle, ainsi que du risque du classificateur de Gibbs à piges multiples (ce classificateur permet d'approcher, en augmentant le nombre de piges, le classificateur par vote de majorité).

Pour un ensemble d'entrainement donné, et pour une fonction de perte convenable donnée $\zeta_Q(\mathbf{x}, y)$, les théorèmes 5.1.2 et 6.3.1 permettent de borner les risques ζ_Q simultanément pour toute distribution Q. À partir de cette observation vient l'idée de concevoir des algorithmes d'apprentissage basé sur ces théorèmes. Ces algorithmes, que nous présentons dans la deuxième partie de la thèse, conçoivent des classificateurs par votes de majorité, B_Q, en choisissant la distribution Q de sorte à minimiser le risque associé à une fonction de perte préalablement choisie. À partir des fonctions de perte linéaire, quadratique, exponentielle ainsi qu'à partir de la fonction de perte du classificateur de Gibbs à piges multiples, nous avons conçu huit algorithmes d'apprentissage (un algorithme basé sur le théorème 5.1.2 et un autre sur le théorème 6.3.1 pour chaque

fonction de perte). Certains de ces algorithmes se comparent favorablement avec une implémentation d'AdaBoost.

Questions ouvertes

La différence entre l'algorithme Exp-kl de la section 8.5 et celui de la section 9.4 réside dans le fait que la première version de l'algorithme se base sur une borne utilisant la fonction de pseudo-distance $\mathrm{KL}(Q\|P)$ pour réguler la distribution Q, alors que la seconde utilise seulement le fait que la distribution Q est contrainte à être une distribution quasi-uniforme. Par cette seule modification, nous observons une nette amélioration de performance des classificateurs produits (voir tableaux 8.5.4 et 9.3). Nous observons également que les versions des algorithmes basées sur le théorème 6.3.1 performent généralement mieux que celles basées sur le théorème 5.1.2. Ces améliorations s'expliquent par la présence de l'hyperparamètre C qui permet de modifier l'importance du régularisateur. Il résulte de ces observations que le régularisateur $\mathrm{KL}(Q\|P)$ n'est pas optimal.

De plus, nous observons que la sélection par la borne des hyperparamètres mène généralement à de piètres résultats (que nous n'avons d'ailleurs pas présentés). En fait, les version des algorithmes permettant d'obtenir les meilleurs classificateurs fournissent généralement des bornes sur le risque très lâches.

La question naturelle qui se pose alors est la suivante : pouvons-nous concevoir un meilleur théorème PAC-Bayes ? Pour répondre à cette question, il faut d'abord se pencher sur le fameux théorème 5.1.2.

La différence entre le risque empirique associé à une fonction de perte générale et la valeur de la borne fournie par la théorème 5.1.2 (ou le théorème 5.1.1) augmente essentiellement linéairement avec la valeur de la somme des coefficients du développement en série de Taylor centré en $W_Q(\mathbf{x}, y) = \frac{1}{2}$ de la fonction de perte (voir l'inégalité 5.5). Elle augmente donc au moins linéairement avec la valeur de la pente de la fonction de perte évaluée au point $W_Q(\mathbf{x}, y) = \frac{1}{2}$.

Il suit que la borne du risque associé à des fonctions de perte à valeurs dans $[0, 1]$ peut être arbitrairement grande. Par exemple, la fonction de perte

$$\varphi(\mathbf{x}, y) = \frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{\beta}{\sqrt{2}}\Big(2W_Q(\mathbf{x}, y) - 1\Big)\right)\right)$$

est à valeurs comprises dans $[0, 1]$ et satisfait les contraintes du théorème 5.1.1 pour

toute valeur $\beta > 0$. De plus, elle s'approche de la fonction de perte $I(W_Q(\mathbf{x}, y) > \frac{1}{2})$ lorsque $\beta \to \infty$. Cette fonction est donc la candidate idéale pour approcher le risque du classificateur par vote de majorité. Malheureusement, pour des valeurs de β intéressantes, le théorème sur les fonctions de perte générales ne permet pas d'obtenir une borne serrée du risque. Même que, malgré le fait que le risque soit à valeur dans $[0, 1]$, pour un ensemble S donné, la borne sur le risque tendra vers l'infini en faisant tendre β vers l'infini.

Un résultat équivalent au théorème 5.1.2 mais fournissant une borne dont la dégradation serait moins sévère pour des fonctions de perte approchant la perte zéro-un (et idéalement à valeurs $[0, 1]$ pour une fonction de perte à valeurs $[0, 1]$) serait un point de départ intéressant pour la conception de meilleurs algorithmes d'apprentissage basés sur des bornes de type PAC-Bayes.

Index

Bibliographie

Amiran Ambroladze, Emilio Parrado-Hernández et John Shawe-Taylor : Learning the prior for the pac-bayes bound. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.95.7955, 2004.

Amiran Ambroladze, Emilio Parrado-Hernández et John Shawe-Taylor : Tighter PAC-Bayes bounds. *In Proceedings of the 2006 conference on Neural Information Processing Systems (NIPS-06)*, pages 9–16, 2006.

Arindam Banerjee : On bayesian bounds. *In ICML '06 : Proceedings of the 23rd international conference on Machine learning*, pages 81–88, New York, NY, USA, 2006. ACM. ISBN 1-59593-383-2.

C.L. Blake et C.J. Merz : *UCI Repository of machine learning databases*. Department of Information and Computer Science, Irvine, CA : University of California, 1998. URL http://archive.ics.uci.edu/ml/.

Leo Breiman : Bagging predictors. *Machine Learning*, 24(2):123–140, 1996. URL http://citeseer.ist.psu.edu/breiman96bagging.html.

Leo Breiman : Random forests. *Machine Learning*, 45(1):5–32, 2001. ISSN 0885-6125. URL http://citeseer.ist.psu.edu/breiman01random.html.

Olivier Catoni : Pac-bayesian inductive and transductive learning, 2006. URL http://arxiv.org/abs/math/0605793.

Olivier Catoni : *PAC-Bayesian surpevised classification : the thermodynamics of statistical learning*. Monograph series of the Institute of Mathematical Statistics, 2007. URL http://arxiv.org/abs/0712.0248.

Ke Chen, Lan Wang et Huisheng Chi : Methods of combining multiple classifiers with different features and their applications to text-independent speaker identification. *International Journal of Pattern Recognition and Artificial Intelligence*, 11:417–445, 1997.

Corinna Cortes et Vladimir Vapnik : Support-vector networks. *In Machine Learning*, pages 273–297, 1995. URL citeseer.ist.psu.edu/cortes95supportvector.html.

Thomas M. Cover et Joy A. Thomas : *Elements of Information Theory*, chapitre 12. Wiley, 1991.

Nello Cristianini et John Shawe-Taylor : *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press, Cambridge, U.K., 2000.

Yoav Freund et Robert E. Schapire : A decision-theoretic generalization of on-line learning and an application to boosting. *In European Conference on Computational Learning Theory*, pages 23–37, 1995. voir http://citeseer.ist.psu.edu/freund95decisiontheoretic.html.

Yoav Freund et Robert E. Schapire : Experiments with a new boosting algorithm. *In International Conference on Machine Learning*, pages 148–156, 1996. voir http://citeseer.ist.psu.edu/freund96experiments.html.

Pascal Germain : Algorithmes d'apprentissage automatique inspirés de la théorie pac-bayes. Mémoire de D.E.A., Université Laval, 2009. http://archimede.bibl.ulaval.ca/archimede/uid/555802bc-3351-4c1e-8b74-b1e2bf14b508.

Pascal Germain, Alexandre Lacasse, François Laviolette et Mario Marchand : PAC-Bayesian learning of linear classifiers. *In* Léon Bottou et Michael Littman, éditeurs : *Proceedings of the 26th International Conference on Machine Learning*, pages 353–360, Montreal, June 2009a. Omnipress.

Pascal Germain, Alexandre Lacasse, Francois Laviolette, Mario Marchand et Sara Shanian : From PAC-Bayes bounds to KL regularization. *In* Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams et A. Culotta, éditeurs : *Advances in Neural Information Processing Systems 22*, pages 603–610. 2009b.

Pascal Germain, Alexandre Lacasse, François Laviolette et Mario Marchand : A pac-bayes risk bound for general loss functions. *In Advances in Neural Information Processing Systems 19*, pages 449–456. MIT Press, Cambridge, MA, 2007. voir http://books.nips.cc/papers/files/nips19/NIPS2006_0494.pdf.

W. K. Hastings : Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, April 1970. URL http://dx.doi.org/10.1093/biomet/57.1.97.

Ralf Herbrich et Thore Graepel : A pac-bayesian margin bound for linear classifiers : Why svms work. *In Advances in neural information processing systems*, volume 13, pages 224–230, 2001.

A. E. Hoerl et R. Kennard : Ridge regression : Application to non orthogonal problems. *Technometrics*, 12, 1970a.

A. E. Hoerl et R. Kennard : Ridge regression : biased estimation for non orthogonal problems. *Technometrics*, 12, 1970b.

Josef Kittler, Mohamad Hatef, Robert P.W. Duin et Jiri Matas : On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:226–239, 1998. ISSN 0162-8828.

Alexandre Lacasse, François Laviolette, Mario Marchand, Pascal Germain et Nicolas Usunier : Pac-bayes bounds for the risk of the majority vote and the variance of the gibbs classifier. *In Advances in Neural Information Processing Systems 19*, pages 769–776. MIT Press, Cambridge, MA, 2007. voir http://books.nips.cc/papers/files/nips19/NIPS2006_0872.pdf.

Alexandre Lacasse, François Laviolette, Mario Marchand et Francis Turgeon-Boutin : Learning with randomized majority vote. *In Proceedings of the European Conference on Machine Learning (ECML)*, 2010. À paraître.

John Langford : Tutorial on practical prediction theory for classification, 2005. voir http://citeseer.ist.psu.edu/620731.html.

John Langford, Matthias Seeger et Nimrod Megiddo : An improved predictive accuracy bound for averaging classifiers. *In In Proceeding of the Eighteenth International Conference on Machine Learning*, pages 290–297, 2001.

John Langford et John Shawe-Taylor : PAC-Bayes & margins. *In* S. Thrun S. Becker et K. Obermayer, éditeurs : *Advances in Neural Information Processing Systems 15*, pages 423–430. MIT Press, Cambridge, MA, 2003. URL http://citeseer.ist.psu.edu/594035.html.

François Laviolette et Mario Marchand : PAC-Bayes risk bounds for sample-compressed Gibbs classifiers. *Proceedings of the 22nth International Conference on Machine Learning (ICML 2005)*, pages 481–488, 2005.

Andreas Maurer : A note on the pac bayesian theorem. *CoRR*, cs.LG/0411099, 2004. URL http://arxiv.org/abs/cs.LG/0411099.

David McAllester : Some PAC-Bayesian theorems. *Machine Learning*, 37:355–363, 1999a.

David McAllester : PAC-Bayesian stochastic model selection. *Machine Learning*, 51:5–21, 2003.

David A. McAllester : Pac-bayesian model averaging. *In COLT*, pages 164–170, 1999b.

Ron Meir et Gunnar Rätsch : An introduction to boosting and leveraging. pages 118–183, 2003. voir http://www.boosting.org/papers/MeiRae03.pdf.

Ron Meir et Tong Zhang : Data-dependent bounds for bayesian mixture methods, 2003.

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller et Edward Teller : Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953. URL http://dx.doi.org/10.1063/1.1699114.

John C. Platt : Fast training of support vector machines using sequential minimal optimization. pages 185–208, 1999. URL citeseer.ist.psu.edu/92261.html.

J. R. Quinlan : Bagging, boosting, and c4.5. *In In Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 725–730. AAAI Press, 1996. URL citeseer.ist.psu.edu/quinlan96bagging.html.

Robert E. Schapire, Yoav Freund, Peter Bartlett et Wee Sun Lee : Boosting the margin : A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26:1651–1686, 1998.

Matthias Seeger : PAC-Bayesian generalization bounds for gaussian processes. *Journal of Machine Learning Research*, 3:233–269, 2002.

Vladimir N. Vapnik : *Statistical Learning Theory*. Wiley, New York, NY, 1998.

Annexe A

Démonstrations

Lemme A.0.1. *Soit Q une distribution sur un ensemble de classificateurs binaires $\mathcal{H}$. Soit $f : \mathcal{H} \to [0,1]$ et $g : \mathcal{H} \to [0,1]$ deux fonctions. Alors nous avons*

$$\mathop{\mathbf{E}}_{h\sim Q} \mathrm{kl}\left(f(h) \| g(h)\right) \;\geq\; \mathrm{kl}\left(\mathop{\mathbf{E}}_{h\sim Q} f(h) \Big\| \mathop{\mathbf{E}}_{h\sim Q} g(h)\right) .$$

Démonstration : Nous présentons ici une preuve supposant la dénombrabilité de l'ensemble $\mathcal{H}$, cependant, de par la convexité de $\mathrm{kl}(\cdot\|\cdot)$, le lemme tient également dans le

cas indénombrable. Nous avons

$$\begin{aligned}
\mathop{\mathbf{E}}_{h\sim\mathcal{H}} \mathrm{kl}\left(f(h)\|g(h)\right) &= \mathop{\mathbf{E}}_{h\sim\mathcal{H}} f(h)\log\frac{f(h)}{g(h)} + (1-f(h))\log\frac{1-f(h)}{1-g(h)} \\
&= \sum_{h\sim\mathcal{H}} Q(h)f(h)\log\frac{f(h)}{g(h)} + \sum_{h\sim\mathcal{H}} Q(h)(1-f(h))\log\frac{1-f(h)}{1-g(h)} \\
&= \sum_{h\sim\mathcal{H}} Q(h)f(h)\log\frac{Q(h)f(h)}{Q(h)g(h)} \\
&\qquad + \sum_{h\sim\mathcal{H}} Q(h)(1-f(h))\log\frac{Q(h)(1-f(h))}{Q(h)(1-g(h))} \\
&\geq \left(\sum_{h\in\mathcal{H}} Q(h)f(h)\right)\log\frac{\sum_{h\in\mathcal{H}} Q(h)f(h)}{\sum_{h\in\mathcal{H}} Q(h)g(h)} \\
&\qquad + \left(1-\sum_{h\in\mathcal{H}} Q(h)f(h)\right)\log\frac{1-\sum_{h\in\mathcal{H}} Q(h)f(h)}{1-\sum_{h\in\mathcal{H}} Q(h)g(h)} \\
&= \left(\mathop{\mathbf{E}}_{h\sim\mathcal{H}} f(h)\right)\log\frac{\mathop{\mathbf{E}}_{h\sim\mathcal{H}} f(h)}{\mathop{\mathbf{E}}_{h\sim\mathcal{H}} g(h)} + \left(1-\mathop{\mathbf{E}}_{h\sim\mathcal{H}} f(h)\right)\log\frac{1-\mathop{\mathbf{E}}_{h\sim\mathcal{H}} f(h)}{1-\mathop{\mathbf{E}}_{h\sim\mathcal{H}} g(h)} \\
&= \mathrm{kl}\left(\mathop{\mathbf{E}}_{h\sim Q} f(h)\Big\|\mathop{\mathbf{E}}_{h\sim Q} g(h)\right) .
\end{aligned}$$

Suite à la troisième égalité, pour faire apparaitre une inégalité, nous avons fait intervenir l'inégalité log-somme (voir Cover et Thomas, 1991, chapitre 12, section 1). ■

Lemme A.0.2. *Soit Q une distribution sur un ensemble de classificateurs binaires $\mathcal{H}$. Soit $\alpha : \mathcal{H}\times\mathcal{H}\to[0,1]$, $\widehat{\alpha} : \mathcal{H}\times\mathcal{H}\to[0,1]$, $\beta : \mathcal{H}\times\mathcal{H}\to[0,1]$ et $\widehat{\beta} : \mathcal{H}\times\mathcal{H}\to[0,1]$ des fonctions. Alors nous avons*

$$\mathop{\mathbf{E}}_{(h_1,h_2)\sim Q^{(2)}} \mathrm{kl}(\widehat{\alpha_{12}},\widehat{\beta_{12}}\|\alpha_{12},\beta_{12}) \quad\geq\quad \mathrm{kl}(\widehat{\alpha_Q},\widehat{\beta_Q}\|\alpha_Q,\beta_Q)\,,$$

où α_{12} est une formulation abrégée pour $\alpha(h_1,h_2)$ et α_Q est une formulation abrégée pour $\mathop{\mathbf{E}}_{(h_1,h_2)\sim Q^{(2)}} \alpha(h_1,h_2)$ et similairement pour $\widehat{\alpha}$, β et $\widehat{\beta}$.

Démonstration : Nous présentons ici une preuve supposant la dénombrabilité de l'ensemble $\mathcal{H}$, cependant, de par la convexité de $\mathrm{kl}(\cdot\|\cdot)$, le lemme tient également dans le

cas indénombrable. Nous avons

$$
\begin{aligned}
&\mathop{\mathbf{E}}_{(h_1,h_2)\sim Q^{(2)}} \mathrm{kl}(\widehat{\alpha_{12}}, \widehat{\beta_{12}} \| \alpha_{12}, \beta_{12}) \\
&\quad = \mathop{\mathbf{E}}_{(h_1,h_2)\sim Q^{(2)}} \left[\widehat{\alpha_{12}} \log \frac{\widehat{\alpha_{12}}}{\alpha_{12}} + \widehat{\beta_{12}} \log \frac{\widehat{\beta_{12}}}{\beta_{12}} + \ (1 - \widehat{\alpha_{12}} - \widehat{\beta_{12}}) \log \frac{1 - \widehat{\alpha_{12}} - \widehat{\beta_{12}}}{1 - \alpha_{12} - \beta_{12}} \right] \\
&\quad = \sum_{(h_1,h_2)\in\mathcal{H}^{(2)}} Q^{(2)}(h_1,h_2) \Bigg[\widehat{\alpha_{12}} \log \frac{\widehat{\alpha_{12}}}{\alpha_{12}} \\
&\qquad\qquad + \widehat{\beta_{12}} \log \frac{\widehat{\beta_{12}}}{\beta_{12}} + \ (1 - \widehat{\alpha_{12}} - \widehat{\beta_{12}}) \log \frac{1 - \widehat{\alpha_{12}} - \widehat{\beta_{12}}}{1 - \alpha_{12} - \beta_{12}} \Bigg] \\
&\quad = \sum_{(h_1,h_2)\in\mathcal{H}^{(2)}} Q^{(2)}(h_1,h_2) \Bigg[\widehat{\alpha_{12}} \log \frac{Q^{(2)}(h_1,h_2)\widehat{\alpha_{12}}}{Q^{(2)}(h_1,h_2)\alpha_{12}} + \widehat{\beta_{12}} \log \frac{Q^{(2)}(h_1,h_2)\widehat{\beta_{12}}}{Q^{(2)}(h_1,h_2)\beta_{12}} \\
&\qquad\qquad + (1 - \widehat{\alpha_{12}} - \widehat{\beta_{12}}) \log \frac{Q^{(2)}(h_1,h_2)(1 - \widehat{\alpha_{12}} - \widehat{\beta_{12}})}{Q^{(2)}(h_1,h_2)(1 - \alpha_{12} - \beta_{12})} \Bigg] \\
&\quad = \sum_{(h_1,h_2)\in\mathcal{H}^{(2)}} Q^{(2)}(h_1,h_2)\widehat{\alpha_{12}} \log \frac{Q^{(2)}(h_1,h_2)\widehat{\alpha_{12}}}{Q^{(2)}(h_1,h_2)\alpha_{12}} \\
&\qquad + \sum_{(h_1,h_2)\in\mathcal{H}^{(2)}} Q^{(2)}(h_1,h_2)\widehat{\beta_{12}} \log \frac{Q^{(2)}(h_1,h_2)\widehat{\beta_{12}}}{Q^{(2)}(h_1,h_2)\beta_{12}} \\
&\qquad + \sum_{(h_1,h_2)\in\mathcal{H}^{(2)}} Q^{(2)}(h_1,h_2)(1 - \widehat{\alpha_{12}} - \widehat{\beta_{12}}) \log \frac{Q^{(2)}(h_1,h_2)(1 - \widehat{\alpha_{12}} - \widehat{\beta_{12}})}{Q^{(2)}(h_1,h_2)(1 - \alpha_{12} - \beta_{12})} \,.
\end{aligned}
$$

L'inégalité log-somme (voir Cover et Thomas, 1991, chapitre 12, section 1) appliquée à

chacune des trois sommations permet alors d'obtenir

$$
\begin{aligned}
&\mathop{\mathbf{E}}_{(h_1,h_2)\sim Q^{(2)}} \mathrm{kl}(\widehat{\alpha_{12}}, \widehat{\beta_{12}} \| \alpha_{12}, \beta_{12}) \\
&\qquad \geq \left(\sum_{(h_1,h_2)\in\mathcal{H}^{(2)}} Q^{(2)}(h_1,h_2)\widehat{\alpha_{12}} \right) \cdot \log \frac{\sum\limits_{(h_1,h_2)\in\mathcal{H}^{(2)}} Q^{(2)}(h_1,h_2)\widehat{\alpha_{12}}}{\sum\limits_{(h_1,h_2)\in\mathcal{H}^{(2)}} Q^{(2)}(h_1,h_2)\alpha_{12}} \\
&\qquad\quad + \left(\sum_{(h_1,h_2)\in\mathcal{H}^{(2)}} Q^{(2)}(h_1,h_2)\widehat{\beta_{12}} \right) \cdot \log \frac{\sum\limits_{(h_1,h_2)\in\mathcal{H}^{(2)}} Q^{(2)}(h_1,h_2)\widehat{\beta_{12}}}{\sum\limits_{(h_1,h_2)\in\mathcal{H}^{(2)}} Q^{(2)}(h_1,h_2)\beta_{12}} \\
&\qquad\quad + \left(\sum_{(h_1,h_2)\in\mathcal{H}^{(2)}} Q^{(2)}(h_1,h_2)(1-\widehat{\alpha_{12}}-\widehat{\beta_{12}}) \right) \\
&\qquad\qquad\qquad \cdot \log \frac{\sum\limits_{(h_1,h_2)\in\mathcal{H}^{(2)}} Q^{(2)}(h_1,h_2)(1-\widehat{\alpha_{12}}-\widehat{\beta_{12}})}{\sum\limits_{(h_1,h_2)\in\mathcal{H}^{(2)}} Q^{(2)}(h_1,h_2)(1-\alpha_{12}-\beta_{12})} \\
&\qquad = \widehat{\alpha_Q} \log \frac{\widehat{\alpha_Q}}{\alpha_Q} \;+\; \widehat{\beta_Q} \log \frac{\widehat{\beta_Q}}{\beta_Q} \;+\; (1-\widehat{\alpha_Q}-\widehat{\beta_Q}) \log \frac{1-\widehat{\alpha_Q}-\widehat{\beta_Q}}{1-\alpha_Q-\beta_Q} \\
&\qquad = \mathrm{kl}(\widehat{\alpha_Q}, \widehat{\beta_Q} \| \alpha_Q, \beta_Q)\,.
\end{aligned}
$$

■