SARA SHANIAN

Sample Compressed PAC-Bayesian Bounds and Learning Algorithms

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

DÉPARTEMENT D'INFORMATIQUE ET DE GÉNIE LOGICIEL
FACULTÉ DES SCIENCES ET DE GÉNIE
UNIVERSITÉ LAVAL
QUÉBEC

2012

Résumé

Dans le domaine de la classification, les algorithmes d'apprentissage par compression d'échantillons sont des algorithmes qui utilisent les données d'apprentissage disponibles pour construire l'ensemble de classificateurs possibles. Si les données appartiennent seulement à un petit sous-espace de l'espace de toutes les données possibles, ces algorithmes possédent l'intéressante capacité de ne considérer que les classificateurs qui permettent de distinguer les exemples qui appartiennent à notre domaine d'intérêt. Ceci contraste avec d'autres algorithmes qui doivent considérer l'ensemble des classificateurs avant d'examiner les données d'entraînement. La machine à vecteurs de support (le SVM) est un algorithme d'apprentissage trés performant qui peut être considéré comme un algorithme d'apprentissage par compression d'échantillons. Malgré son succés, le SVM est actuellement limité par le fait que sa fonction de similarité doit être un noyau symétrique semi-défini positif. Cette limitation rend le SVM difficilement applicable au cas où on désire utiliser une mesure de similarité quelconque.

Il a été montré que la théorie PAC-Bayes est un bon point de départ pour concevoir des algorithmes d'apprentissage. Dans cette thése, nous proposons une approche aux méthodes à noyaux qui permet d'utiliser n'importe quelle fonction de similarité bornée. Cette approche est basée à la fois sur la théorie de la compression d'échantillons et la théorie PAC-Bayes. Nous montrons que le SVM est un cas particulier de classificateur par compression d'échantillons que l'on nomme les votes de majorité de classificateurs comprimés. Nous proposons deux catégories différentes de bornes PAC-Bayes sur le risque des votes de majorité de classificateurs comprimés. La premiére catégorie de bornes proposée dépend de la KL-divergence entre la distribution a priori et la distribution a posteriori sur l'ensemble des classificateurs comprimés. La deuxiéme catégorie de bornes proposée posséde la propriété inhabituelle de ne pas utiliser la KL-divergence lorsque la distribution a posteriori est alignée avec la distribution a priori d'une façon précise que nous définissons plus loin dans cette thése. Enfin, pour chaque borne proposée, nous fournissons un nouvel algorithme d'apprentissage qui consiste à trouver le classificateur qui minimise la borne. Les temps de calcul de ces algorithmes sont comparables à des algorithmes comme le SVM. De plus, nous montrons empiriquement que les algorithmes proposés sont trés compétitifs avec le SVM.

Abstract

In classification, sample compression algorithms are the algorithms that make use of the available training data to construct the set of possible predictors. If the data belongs to only a small subspace of the space of all ”possible” data, such algorithms have the interesting ability of considering only the predictors that distinguish examples in our areas of interest. This is in contrast with non sample compressed algorithms which have to consider the set of predictors before seeing the training data. The Support Vector Machine (SVM) is a very successful learning algorithm that can be considered as a sample-compression learning algorithm. Despite its success, the SVM is currently limited by the fact that its similarity function must be a symmetric positive semi-definite kernel. This limitation by design makes SVM hardly applicable for the cases where one would like to be able to use any similarity measure of input example.

PAC-Bayesian theory has been shown to be a good starting point for designing learning algorithms. In this thesis, we propose a PAC-Bayes sample-compression approach to kernel methods that can accommodate any bounded similarity function. We show that the support vector classifier is actually a particular case of sample-compressed classifiers known as majority votes of sample-compressed classifiers. We propose two different groups of PAC-Bayesian risk bounds for majority votes of sample-compressed classifiers. The first group of proposed bounds depends on the KL divergence between the prior and the posterior over the set of sample-compressed classifiers. The second group of proposed bounds has the unusual property of having no KL divergence when the posterior is aligned with the prior in some precise way that we define later in this thesis. Finally, for each bound, we provide a new learning algorithm that consists of finding the predictor that minimizes the bound. The computation times of these algorithms are comparable with algorithms like the SVM. We also empirically show that the proposed algorithms are very competitive with the SVM.

Acknowledgments

It would not have been possible to write this dissertation without the help and support of the kind people around me and I want to extend my sincerest thanks to them all.

I am particularly indebted to my supervisor Dr. François Laviolette for kindly providing guidance, encouragement, and enormous help throughout this study. He has always been available whenever I needed help or advice. He has given me extremely helpful reviews and comments on everything I have written in this dissertation. I would also like to thank my co-supervisor Dr. Mario Marchand for his insightful suggestions and his help at the various stages of the present work.

I would like to thank the rest of the members of my dissertation committee: Dr. Hugo Larochelle, Dr. Claude-Guy Quimper, and Dr. Pascal Tesson for taking the time to read this dissertation and for their helpful advices and suggestions.

I would like to acknowledge the financial support for my research which was provided by *Fond Québécois de la Recherche sur la Nature et les Technologies* (FQRNT) through a postgraduate scholarship.

I would like to thank everybody at the Machine Learning Group of Laval University (GRAAL). In particular, I like to thank Pascal Germain for his friendship and all his assistance in running experiments and also in translating the abstract of this dissertation to French. Thank you Pascal, it was all fun working with you! I also like to thank Alexandre Lacoste for his friendship and the many times he connected me to our group meetings when I was away. I would also like to thank Alexandre Lacasse, Jean-Francis Roy, Francis Turgeon-Boutin and Sébastien Giguère for their friendship, assistance,

support, and encouragement.

I would also like to thank all of my friends and colleagues whose collaborative and personal support was essential for the completion of this research. Mohak Shah may know as much about the research work presented in my dissertation as the members of my dissertation committee. I cannot thank him enough for his willingness to listen and comprehend challenges I faced in my studies and for his invaluable support and friendship over the years. I would like to thank Atousa Reyhani for her support and kindness throughout my time of studying in Quebec. I think of her as a big sister. I would also like to thank Akanksha and Maher for all the fun times, laughters, conversations, and their friendship.

I like to thank my parents, Fatemeh and Mohammad Taghi for their love, hard work and countless sacrifices to give me a chance at a better life. I thank my sister Solmaz and my brother Sasan (Ali) for their support and encouragement.

Finally, most important thanks here go to my husband, Amir (Heidar), for his endless love, constant support and encouragement all while he was pursuing his own PhD. It is so great to have you beside me. Thank you for being such a wonderful friend and husband. Without you I could not have done this. I love you... always.

To my husband Amir, my best friend and the love of my life, who has been by my side throughout every step of this journey:

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Machine learning is a subfield of artificial intelligence in which different approaches are studied to automate the process of learning a task. There are many examples of machine learning problems such as optical character recognition, face recognition, spam filtering, etc. In this thesis, we focus on classification problems in which the goal is to categorize objects into a fixed set of categories. We are interested in designing machine learning processes (learning algorithms) capable of constructing classifiers of good quality when given a set of already classified examples. The probability of a classifier making mistakes in classifying future data is a factor that can show its quality. This probability is known as the risk of the classifier. A classifier of low risk is considered to “generalize well”, meaning that it will make few errors on new (not yet classified) examples. A risk bound, therefore, can work as a guarantee on the future performance of the classifier. Interestingly, the same bound can be also exploited by the learning algorithm to guide it during its execution, to construct classifiers of better quality.

One of the most frequently used approaches to find risk bounds is the test set approach. In this approach, the data is divided into a training set and a test set. First, the training data is used to build a classifier. Then, the test set data is used to bound the risk of the classifier. However, for the bound to be valid, the test set must be independent of the classifier, which prevents us from using the bound in building a better classifier [49].

Another approach is to bound the risk of a classifier directly from the information obtained from the training set, including the empirical risk, which is the rate of misclassification of the classifier on the training set. This type of bounds has to be simultaneously valid for all possible classifiers of a given class. This is why from such type of bounds one can always derive a learning algorithm, the algorithm that outputs among all possible classifiers, the one having the smallest bound value.

The PAC-Bayes theorem initiated by [42] provides this type of bounds. This type of bounds allows us to perform model selection while not forcing us to reserve a part of our examples to calculate the bound on the test set. Knowing that the PAC-Bayes theorem provides a bound that is capable of guiding model selection, we would like to find out if it would be possible to design new PAC-Bayes bounds that in turn will propose learning algorithms. That is, we would like to know if the classifier with the lowest PAC-Bayes bound is a classifier that generalizes well. If so, learning algorithms that construct classifiers which can minimize the PAC-Bayes type of bounds will be promising.

Many learning algorithms use a fixed-size set of training examples to construct a classifier. On the other hand, there are certain types of algorithms relying on the substantially small subset of training examples to construct the classifier. These algorithms fall under the category of Sample Compression based algorithms. Within the sample-compression framework [14, 33], each sample-compressed classifier is partly represented by a subset of the training examples, called the *compression set*. This compression set along with possibly some additional informations called the *message* are enough to reconstruct the classifier. Algorithms such as Nearest Neighbor (NN), classical Perceptron, Support Vectors Machine (SVM) [22], and Set Covering Machine (SCM)) [38] can be considered as sample compression based algorithms. In the perceptron, the compression set consists of the examples used to update the weight vector and the threshold of the separating hyperplane. In the SVM, the compression set consists of the examples that belong to the set of support vectors. In the case of SCM, the compression set is not enough to reconstruct the classifier; a message has to be added to the compression set. This particular sample compressed algorithm will be addressed in Chapter 4.

Sample Compression is an important class of learning algorithms since many well known learning algorithms, such as the SVM and the perceptron learning rule, can be considered as sample-compression learning algorithms. Considering the capability of PAC-Bayes bounds (as a type of training set bounds) in model selection and guiding strategy for learning algorithms, it is worthwhile to strive to derive novel PAC-Bayes type of bounds in the sample compression framework. This can provide us with insights and intuitions towards, if not concrete algorithms, designing practical algorithms.

1.1 Thesis Contributions

Note that, all along the thesis, the publications for which the author of this thesis is a co-author are emphasized by double brackets: "[[ ]]" instead of the usual single bracket form "[ ]".

The most important contribution of this thesis has been published as "A PAC-Bayes Sample Compression Approach to Kernel Methods" [[20]], for which the author of this thesis is a principal author. In [[20]], we proposed two different groups of PAC-Bayesian risk bounds for majority votes of sample-compressed classifiers, and their associated new learning algorithms. These algorithms are very competitive in comparison with the state of the art. The approaches proposed in [[20]], were highly inspired by preliminary work presented in two other publications, "From PAC-Bayes Bounds to KL Regularization" [[19]] and "Learning the set covering machine by bound minimization and margin-sparsity trade-off" [[31]]. In [[19]], we investigated how the PAC-Bayes theory can be adapted to a more general notion of loss. [[19]] was itself based on [18], a pioneer work on that topic. On the other hand, [[31]] explores learning algorithms based on minimizing bounds for the SCM. [[31]] was a continuation of two previous publications "Margin-Sparsity Trade-off for the Set Covering Machine" [34], and "A PAC-Bayes Approach to the Set Covering Machine" [35]. The new proposed bounds related to [[20]] are presented in Chapter 6, their associated learning algorithms are presented in Chapter 7, and finally, experimental results about those algorithms can be found in Chapter 8.

In this thesis:

- Inspired by the work in [[19]] on general loss bounds for stochastic classifiers and also by [[31]] on proposing a learning algorithm based on minimizing a bound for SCM, we propose two new groups of PAC-Bayesian risk bounds for majority votes of sample-compressed classifiers [[20]].
- We propose a PAC-Bayes sample-compression approach to kernel methods that can accommodate any bounded similarity function.

- We show that the SVM classifier is actually a particular case of a (weighted) majority votes of sample-compressed classifiers where the compression sequence of each classifier consists of at most a single training example.
- For each proposed risk bound, we provide a learning algorithm that minimizes that bound. The first group of proposed PAC-Bayes risk bounds depends on the KL divergence between the prior and the posterior over the set of sample-compressed classifiers. We show that the corresponding bound-minimizing learning algorithm is KL-regularized. The second group of proposed PAC-Bayes risk bounds have the unusual property of having no KL divergence when the posterior is *aligned* with the prior in some precise way that we define later in this thesis. We show that minimizing these risk upper bounds just amounts to minimizing the proposed empirical loss under the constraint that the posterior is kept aligned with the prior.
- We present an empirical study of applying the test set risk bound approach and compare it against the Gaussian confidence interval approach to evaluate the performance of machine learning algorithms. We also discuss the drawbacks of test set risk bounds and the advantages of using the training set bounds. This part of the thesis has been published as “Hold-out Risk Bounds for Classifier Performance Evaluation” [[49]].

1.2 Thesis Organization

The rest of the thesis is organized as below:

In Chapter 2, we present some definitions that form the basis for the rest of the thesis. We start by defining the classification problem and the risk of the classifier. We also present the notion of majority votes of classifiers and Gibbs and Bayes classifiers.

In Chapter 3, we present the test set bound approach as one of the most frequently used approaches to find a bound for the risk of a classifier. We empirically study the use of the test set approach and the Gaussian confidence interval approach to see how they compare in evaluating the performance of machine learning algorithms as explained in [[49]].

In Chapter 4, we define the sample compression setting and show how we can compute a risk of the sample compression classifier. we also present the SCM algorithm as a sample compression algorithm.

In Chapter 5, we first define the classical PAC-Bayes bound along with some proposed versions of the PAC-Bayes bounds in the data-independent setting. We then present the PAC-Bayes bound in the sample compression setting of [32].

In Chapter 6, we present a number of new PAC-Bayes sample compressed bounds for majority votes of sample-compressed classifiers which are valid for any similarity measure of input examples [[20]].

In Chapter 7, we specialize the proposed risk bounds in Chapter 6 to the majority vote of sample compressed classifiers having compression set of size at most one. We also show that the SVM classifier can be considered as a particular case of a (weighted) majority votes of sample compressed classifiers [[20]]. In this particular case, the compression sequence of each classifier consists of at most a single training example. We also provide a learning algorithm that minimizes the bounds proposed in chapter 6.

In Chapter 8, we present some empirical results of applying the proposed algorithms on a number of data sets [[20]].

Finally, in Chapter 9, we conclude the thesis and discuss some future directions.

Chapter 2

Basic Notions

2.1 Classification

Classification is the task of assigning objects to one of many predefined classes. As an example, consider the task of categorizing bank clients (objects) who are demanding credit cards into eligible or not eligible (classes). In this task, the credit records of each client such as his income, other bank accounts, age, and other personal information (attributes) are needed to be analyzed to categorize each client into eligible or not eligible. In this thesis, we focus on binary classification problems where the input space $\mathcal{X}$ consists of an arbitrary subset of R^n and the output space $\mathcal{Y} = \{-1, +1\}$. An object (also known as an example) is characterized by a tuple $(\mathbf{x}, y)$ where $\mathbf{x} \in \mathcal{X}$ consists of a vector of n real valued attributes and $y \in \mathcal{Y}$ is the class label.

The goal of a classification task is to construct a function h (known as classifier) that maps each example $\mathbf{x} \in \mathcal{X}$ to one of the predefined class labels $y \in \mathcal{Y}$ ($h : \mathcal{X} \rightarrow \mathcal{Y}$). This goal is fulfilled by a learning algorithm that receives training set S as its input and outputs a classifier h. The *training set* $S = \{z_1 = (\mathbf{x}_1, y_1), ..., z_m = (\mathbf{x}_m, y_m)\}$ consists of m examples whose class labels are known. This way, classifier h is then used to classify a new unseen example which is drawn according to a fixed but unknown distribution D.

To measure the accuracy of classifier h, we need to measure its *risk*. The risk of classifier h is the degree of disagreement between the label y of example $\mathbf{x}$ and the label assigned to $\mathbf{x}$ by the classifier, denoted as $h(\mathbf{x})$. Given a training set S, the goal is to construct a classifier with minimum risk without any information about D. If the example $\mathbf{x}$ is drawn according to an unknown distribution D, the risk, in this case referred to as *true risk*, is the probability that a classifier mis-classifies $\mathbf{x}$ and is represented as:

$$R_D(h) = Pr_{(\mathbf{x},y)\sim D}(h(\mathbf{x}) \neq y)).$$

Since computing the exact value of the true risk of a classifier as given above is not possible (the distribution D is unknown in our model), we compute the *empirical risk* which is the risk of the classifier with respect to the training set. The empirical risk $R_S(h)$ is shown as:

$$R_S(h) = \frac{1}{m}\sum_{i=1}^{m} I(y_i \neq h(\mathbf{x}_i)).$$

Where $I(a) = 1$ if predicate a is true and 0 otherwise.

2.2 Majority Vote

When solving a classification problem, a learning algorithm is trained on a training set S and outputs the best classifier according to some criteria (e.g, empirical risk). The best output classifier is not necessarily the ideal choice for the following reasons:

- When we have an insufficient number of examples in the training set, more than one classifier may have the same empirical risk and it is not clear which of these classifier is the best one.
- Less successful classifiers (e.g, classifiers with higher empirical risk) may also correctly classify some examples. By discarding these classifiers some valuable information may be lost.

Combining a number of trained classifier can lead to a better performance than a single classifier [23]. There is a variety of methods to combine classifiers (see [5, 25, 55, 26]). Boosting [15] and Bagging [4] are two popular combining strategies. They both modify the training set, build classifiers on these modified training sets and then combine them into a final classifier by a *simple majority vote* or a *weighted majority vote*. However, each of these strategies (i.e., Boosting and Bagging) build their classifiers in their own specific way.

In this thesis, we are interested in weighted majority vote. Given a set of classifiers $\mathcal{H}$, a weighted majority vote is a classifier which is constructed form the combination of a number of classifiers $h_i \in \mathcal{H}$. We denote Q as a weighted distribution on the set of classifiers $\mathcal{H}$. This way, in voting, each classifier $h_i \in \mathcal{H}$ has a weight $Q(h_i)$ which reflects how confident it is about its outputs. Simple majority vote is a special case of weighted majority vote assigning an equal weight of $1/k$ to each classifier $h_i \in \mathcal{H}$ where k is the number of classifiers in $\mathcal{H}$.

Definition 2.2.1. *For any distribution Q over $\mathcal{H}$, the Q-weighted majority vote classifier B_Q, denoted as Bayes classifier, on any example $\mathbf{x}$ is given by:*

$$B_Q(\mathbf{x}) \overset{def}{=} \operatorname{sgn}\left[\mathbf{E}_{h\sim Q} h(\mathbf{x})\right]$$

where $\operatorname{sgn}(s) = +1$ *if real number* $s > 0$ *and* $\operatorname{sgn}(s) = -1$ *otherwise.*

The majority vote classifier B_Q is related to the output of a stochastic classifier called the *Gibbs classifier*. To classify an input example $\mathbf{x}$, the Gibbs classifier G_Q chooses randomly a (deterministic) classifier h according to Q. The true risk $R_D(G_Q)$ and the empirical risk $R_S(G_Q)$ of the Gibbs classifier are thus given by:

$$R_D(G_Q) \overset{\text{def}}{=} \mathbf{E}_{h\sim Q} R_D(h)) = \mathbf{E}_{h\sim Q}\mathbf{E}_{(\mathbf{x},y)\sim D} I(h(\mathbf{x}) \neq y).$$

$$R_S(G_Q) \overset{\text{def}}{=} \mathbf{E}_{h\sim Q} R_S(h) = \mathbf{E}_{h\sim Q}\frac{1}{m}\sum_{i=1}^{m} I(y_i \neq h(\mathbf{x}_i)).$$

Note that whenever B_Q mis-classifies an example $\mathbf{x}$, at least half of the classifiers (under measure Q), mis classify $\mathbf{x}$. It follows that the error rate of G_Q is at least half of the error rate of B_Q. Hence $R(B_Q) \leq 2R(G_Q)$. More formally we have:

$$B_Q(\mathbf{x}) \neq y \Rightarrow \mathbf{E}_{h\sim Q} I(h(\mathbf{x}) \neq y) > 1/2.$$

2.3 Support Vector Machine

The Support Vector Machine (SVM) is a state-of-the-art classification method proposed by [3]. SVM belongs to the group of methods that depend on the data only through dot-products. This group is known as *kernel methods*. In these methods the data are mapped into a higher dimensional space, known as *feature space*, and then the dot products are replaced by a *kernel function* which computes the dot products in the feature space [2].

The SVM is one of the commonly used learning algorithm in a class of kernel methods. We use SVM as a benchmark to evaluate the performance of our new suggested learning algorithms later in this thesis.

We consider the SVM in the binary classification setting in which the SVM constructs a hyperplane that separates the input space into two parts. This hyperplane can be described as:

$$\mathbf{w} \cdot \mathbf{x} + b = 0,$$

where the vector $\mathbf{w}$ is the normal vector perpendicular to the hyperplane, b is the bias that allows the hyperplane not to pass through the origin, and $\mathbf{w} \cdot \mathbf{x}$ is the dot product.

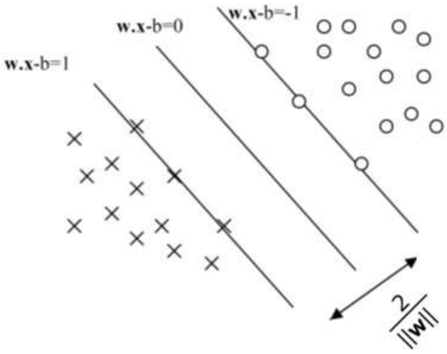

Figure 2.1: Maximum-margin hyperplane for SVM trained with samples from two classes. On both side of the optimal separating hyperplane the instances are at least $\frac{1}{\|\mathbf{w}\|}$ away and the total margin is $\frac{2}{\|\mathbf{w}\|}$. Support vectors are examples located on $\mathbf{w}{\cdot}x{+}b = \pm 1$.

The examples that lie closest to the hyperplane are called *support vectors*. Although there are many hyperplanes that can separate the data into two parts, the goal of the SVM is to choose a hyperplane with the maximum distance from the closest examples of both classes. (see Figure 2.1). This distance is known as *margin*. Formally, let $\mathbf{x}_+$

and $\mathbf{x}_-$ respectively be the closest points to the hyperplane among the examples with the class label $+1$ and -1. By using geometry, the margin of the hyperplane is given by:

$$M = \frac{1}{2}\hat{\mathbf{w}} \cdot (\mathbf{x}_+ - \mathbf{x}_-),$$

where $\hat{\mathbf{w}} = \frac{\mathbf{w}}{\|\mathbf{w}\|}$ is a unit vector in the direction of $\mathbf{w}$. By assuming that both $\mathbf{x}_+$ and $\mathbf{x}_-$ have equal distance from the hyperplane we have:

$$\mathbf{w} \cdot \mathbf{x}_+ + b = a,$$

$$\mathbf{w} \cdot \mathbf{x}_- + b = -a$$

for some constant $a > 0$. We can set $a = 1$ in order to just consider the examples that lie closest to the separating hyperplane and make the geometric margin meaningful. This way we obtain:

$$M = \frac{1}{2}\hat{\mathbf{w}} \cdot (\mathbf{x}_+ - \mathbf{x}_-) = \frac{1}{\|\mathbf{w}\|}.$$

Now maximizing the geometric margin $\frac{1}{\|\mathbf{w}\|}$ is equivalent to minimizing $\|\mathbf{w}\|^2$ which leads to the following constrained optimization problem:

$$\begin{aligned} &\text{Minimize:} && \tfrac{1}{2}\|\mathbf{w}\|^2 \\ &\text{subject to:} && y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \quad i = 1 \cdots m. \end{aligned} \tag{2.1}$$

The obtained maximum margin classifier from the above equation classifies each example correctly due to the constraints $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1$. However, in many cases when the examples are not linearly separable, greater margin can be obtained by allowing the classifier to mis-classify some examples. This can be done by replacing the inequality constraints in Equation (2.1) with $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i \quad i = 1 \cdots m$ where $\xi \geq 0$ are known as *slack variables* and when $\xi \geq 1$ the constraints $y_i(\mathbf{w}.\mathbf{x}_i + b) \geq 1 - \xi$ allow the examples to be misclassified. This leads to the following constrained optimization problem known as *soft-Margin SVM* which is introduced by [12].

$$\begin{aligned} &\text{Minimize:} && \tfrac{1}{2}\|\mathbf{w}\|^2 + C\textstyle\sum_{i=1}^{m} \xi_i \\ &\text{subject to:} && y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \end{aligned} \tag{2.2}$$

where the parameter C ($C \geq 0$) controls the trade off between the slack variables penalty and the size of the margin. By using the Lagrange multipliers method, we obtain:[12]

$$\begin{aligned} &\text{Minimize:} && \textstyle\sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} y_i y_j \alpha_i \alpha_j \mathbf{x}_i \cdot \mathbf{x}_j \\ &\text{subject to:} && \textstyle\sum_{i=1}^{m} y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq C. \end{aligned} \tag{2.3}$$

where $\mathbf{w} = \sum_{i=1}^{m} y_i \alpha_i \mathbf{x}_i$. The support vectors are the examples for which $\alpha_i \geq 0$.

There are many classification problems that are not linearly separable in input space while they are linearly separable in a higher dimensional space (feature space). In these cases, input space $\mathcal{X}$ can be mapped into a higher dimensional feature space $\mathcal{F}$ using a non-linear function $\boldsymbol{\phi} : \mathcal{X} \rightarrow \mathcal{F}$. Suppose that the weight vector $\mathbf{w}$ can be expressed as a linear combination of the training examples ($\mathbf{w} = \sum_{i=1}^{m} y_i \alpha_i \mathbf{x}_i$), then the separating hyperplane can be defined as:

$$\sum_{i=1}^{m} y_i \alpha_i \mathbf{x}_i \cdot \mathbf{x} + b = 0$$

In the feature space $\mathcal{F}$ this expression takes the following form:

$$\sum_{i=1}^{m} y_i \alpha_i \boldsymbol{\phi}(\mathbf{x}_i) \cdot \boldsymbol{\phi}(\mathbf{x}) + b = 0.$$

To calculate the value of the dot product in $\mathcal{F}$ we need to explicitly calculate the mapping $\boldsymbol{\phi}$. This can become impractical when the feature space $\mathcal{F}$ is of high dimension. In this case, a kernel function $k(\mathbf{x}_i, \mathbf{x}) = \boldsymbol{\phi}(\mathbf{x}) \cdot \boldsymbol{\phi}(\mathbf{x}_i)$ can be employed. The kernel $k(\mathbf{x}_i, \mathbf{x})$ can be computed without explicitly computing the mapping $\boldsymbol{\phi}$ [2].

By using a kernel function, the separating hyperplane takes the following form

$$\sum_{i=1}^{m} y_i \alpha_i k(\mathbf{x}_i, \mathbf{x}) + b = 0,$$

and the constrained optimization problem in Equation (2.4), is given by:

$$\begin{aligned} &\text{Minimize } \alpha\text{:} && \textstyle\sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} y_i y_j \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) \\ &\text{subject to:} && \textstyle\sum_{i=1}^{m} y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq C. \end{aligned} \tag{2.4}$$

Two commonly used kernels are the polynomial kernel given by $k(\mathbf{u}, \mathbf{v}) = (\mathbf{u}.\mathbf{v}+1)^p$ which induces polynomial boundaries of degree p in the input space $\mathcal{X}$, and the radial basis function kernel $k(\mathbf{u}, \mathbf{v}) = (e^{-\gamma(\mathbf{u}-\mathbf{v}).(\mathbf{u}-\mathbf{v})})$ which induces boundaries by placing weighted Gaussian upon key training examples.

Chapter 3

Test Set Risk Bound for Classifier Performance Evaluation

One of the most common techniques of evaluating the performance of a machine learning algorithm is its empirical evaluation on a separate set of test examples (not used for training the algorithm) [27]. This is generally referred to as hold-out testing. In this case, the full dataset is divided into a training and a hold-out set. A learning algorithm is trained on the training set and outputs the classifier. The output classifier by the learning algorithm after training is then tested on the hold out set of data. Furthermore, one aims to provide a confidence interval around the performance estimate of the learned classifier on the test set. Naturally, to do so, we assume that the test set is representative of the underlying distribution of the test data. Providing such confidence interval around the empirical risk estimate of the chosen classifier on the test data is the issue that we focus on in this chapter. The main aim of such an evaluation is to answer the following questions:

- Given the observed accuracy of a learning algorithm over a limited sample of data, what can we say about the behavior of the learning algorithm over future unseen examples?
- Given that one learning algorithm outperforms another over some sample data, how probable is it that this learning algorithm is more accurate, in general?

The estimates on the future performance of the empirical risk of the classifier, or more appropriately the degree of deviation of the empirical risk from the true risk is generally obtained using a confidence interval in which we believe the true risk of the classifier to lie. The most common method of obtaining such confidence interval relies on the assumption that the empirical risk of the classifier on the test data can be modeled as a Gaussian distribution. Based on this assumption, the necessary statistics are obtained from testing the classifier on the test data. That is, the mean classification error and its corresponding variance on the test examples are obtained. A confidence interval is then provided in terms of a Gaussian around the mean empirical risk with its tails removed at twice the standard deviation estimate on either sides. This provides both a lower and an upper bound on the true risk of the classifier [30, 27]. However, there is a strong caveat in this approach. The whole confidence interval bound strategy relies very significantly on the Gaussian assumption. But the basis of this Gaussian assumption generally comes from the Central Limit theorem in the statistics theory. This result implies that given a true estimate of the data statistic, the sampling distribution of this statistic approaches a Gaussian distribution as the number of samplings reaches infinity. That is, the Gaussian assumption holds on a fixed underlying statistic and that is too asymptotically. However, this might not, and indeed is not, generally the case. The risk in the case of classification is modeled as a zero-one loss. This is equivalent then to having an indicator function which is true when the classifier errs on an example. This would lead to a binomial distribution over a number of trials (tests of the classifier on a number of samples). Further, the aim of learning is to obtain as low an empirical risk as possible. That is, we are interested in modeling the empirical risk of the classifier for lower values (values closer to zero). However, for smaller values of empirical risk a binomial distribution cannot be approximated by a Gaussian. This observation was made by [30]. As a result, applying a Gaussian assumption results in estimates that are overly pessimistic when obtaining an upper bound and overly optimistic when obtaining a lower bound around the empirical risk. [30] also showed a comparison between the behavior of the two distributions with an empirical example of upper bounds on the risk of a decision tree classifier on test datasets.

[48] gave a qualitative analysis of this approach and discussed some important extension possibilities. In this chapter, we further the empirical validation of the test set bound approach [30] by looking at the behavior of both the upper and the lower bounds on the true risk of the classifiers. This is analogous to providing a confidence interval around a binomial distribution. We compare this against the traditional Gaussian confidence interval approach and show on a range of classifiers and datasets, how the

test set bound approach yields more realistic estimates as opposed to the Gaussian confidence intervals.

3.1 Test Set Bound

In this section, we present the test set bound on the true risk of the classifier. We saw earlier that the true risk $R(h)$ of any classifier h is defined as the probability that it misclassifies an example drawn according to D:

$$R_D(h) \overset{\text{def}}{=} \Pr_{(\mathbf{x},y)\sim D}\left(h(\mathbf{x}) \neq y\right) = \mathbf{E}_{(\mathbf{x},y)\sim D} I(h(\mathbf{x}) \neq y)$$

where $I(a) = 1$ if predicate a is true and 0 otherwise. Given a classifier h, and a test set $T = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{|T|}, y_{|T|})\}$ of $|T|$ examples, the *empirical risk* $R_T(h)$ on T of any classifier h, is defined according to:

$$R_T(h) \overset{\text{def}}{=} \frac{1}{|T|}\sum_{i=1}^{|T|} I(h(\mathbf{x}_i) \neq y_i) \overset{\text{def}}{=} \mathbf{E}_{(\mathbf{x},y)\sim T} I(h(\mathbf{x}) \neq y)$$

Now, we model $R_T(h)$ as a binomial distribution. The distribution is defined as the probability of λ errors on a set of $|T|$ examples ($\lambda = |T|R_T(h)$) with true risk of the classifier h being $R(h)$.

$$Pr_{T\sim D^{|T|}}(|T|R_T(h) = \lambda|\ R_D(h)) = \binom{|T|}{\lambda}(R_D(h))^{\lambda}(1 - R_D(h))^{|T|-\lambda} \tag{3.1}$$

We use the cumulative which is the probability of λ or fewer errors with $|T|$ examples.

$$\begin{aligned} \text{Bin}(|T|, \lambda, R(h)) &= Pr_{T\sim D^{|T|}}((|T|R_T(h) \leq \lambda\ |R_D(h)) \\ &= \sum_{i=0}^{\lambda}\binom{|T|}{i}(R_D(h))^{i}(1 - R_D(h))^{|T|-i} \end{aligned}$$

We define the binomial inversion tail [30] as:

$$\overline{\text{Bin}}(|T|, \lambda, \delta) = \max\{p : \text{Bin}(|T|, \lambda, p) \geq \delta)\} \tag{3.2}$$

which is the largest true error such that the probability of observing λ or fewer errors is at least δ.

Then, the risk bound on the true risk of the classifier is defined as [30]:

Theorem 3.1.1. *For all classifiers h, for all D, for all $\delta \in (0,1]$:*

$$Pr_{T\sim D^{|T|}}(R_D(h) \leq \overline{\text{Bin}}(|T|, |T|R_T(h), \delta)) \geq 1-\delta$$

From this result , it follows that $\overline{\text{Bin}}(m, |T|R_T(h), \delta)$ is the smallest upper bound which holds with probability at least $1-\delta$, on the true risk $R_D(h)$ of any classifier h with an observed empirical risk $R_T(h)$ on a set of $|T|$ examples.

In an analogous manner, a lower bound on $R_D(h)$ can be found to be:

Theorem 3.1.2. *For all classifiers h, for all D, for all $\delta \in (0,1]$:*

$$Pr_{T\sim D^{|T|}}(R_D(h) \geq \min_p\{p : 1 - \text{Bin}(|T|, |T|R_T(h), \delta) \geq \delta\}) \geq 1-\delta$$

3.2 Empirical Results

In this section, we present an empirical study of the generalization error bounds on the empirical risk of the classifier on a test set. We show how this approach, by modeling the empirical risk as a binomial[1], can be used to obtain realistic confidence intervals that lie strictly in the $[0,1]$ interval. More precisely, in this section, we examine how the empirical estimates of the risk bounds around the empirical risk fares compared to the traditionally utilized method of obtaining confidence intervals around the empirical risk based on the gaussian assumption. This work has been published in [[49]].

We compare six learning algorithms on 16 different datasets. The learning algorithms compared are the Support Vector Machine equipped with a radial basis function kernel, the Set Covering Machine for learning conjunctions of data-dependent balls [39], Adaboost with decision stumps, decision trees and the Naive Bayes algorithm. With the exception of the SCM for which an in-house implementation was used, the other algorithms were the ones implemented in the Weka machine learning toolkit [56].

Each data set was divided into two parts, a training set S and a test set T. The training set was used to train the learning algorithm and perform model selection to

[1] From Equation (3.1), one can easily show that $|T|R_T(h)$ is the number of errors on a set of $|T|$ of iid examples knowing that on each example, classifier h has a probability $R(h)$ of making an error. In other words, $|T|R_T(h)$ is a random variable that follows a binomial law with parameters $(|T|, R(h))$.

Data-Set	**A**	R_T	B_l	B_u	CI_l	CI_u
USvotes	SVM	0.05	0.027	0.096	0.019	0.080
	Ada	0.04	0.017	0.077	0.012	0.067
	DT	0.055	0.027	0.096	0.022	0.087
	DL	0.045	0.020	0.083	0.015	0.074
	NB	0.07	0.038	0.114	0.033	0.106
	SCM	0.105	0.066	0.156	0.061	0.148
Liver	SVM	0.352	0.235	0.376	0.279	0.424
	Ada	0.291	0.225	0.364	0.222	0.359
	DT	0.325	0.256	0.400	0.254	0.395
	DL	0.325	0.256	0.400	0.254	0.395
	NB	0.4	0.326	0.476	0.3251	0.474
	SCM	0.377	0.305	0.453	0.303	0.450
Credit	SVM	0.183	0.141	0.231	0.138	0.227
	Ada	0.17	0.129	0.217	0.126	0.213
	DT	0.13	0.094	0.173	0.091	0.168
	DL	0.193	0.150	0.242	0.147	0.238
	NB	0.2	0.156	0.249	0.153	0.246
	SCM	0.19	0.147	0.239	0.144	0.235
Glass	SVM	0.168	0.102	0.252	0.095	0.240
	Ada	0	0	0.033	0.0	0.0
	DT	0.186	0.118	0.273	0.110	0.261
	DL	0.065	0.026	0.130	0.017	0.112
	NB	0.299	0.214	0.395	0.210	0.387
	SCM	0.215	0.141	0.304	0.135	0.294

Table 3.1: Results of various classifiers on UCI Datasets

obtain the best parameters from a pre-defined set of parameter values. The learning parameters of all algorithms were determined from the training set only. To do the model selection, a 10 fold cross validation was used on the training set and the parameters with the best average cross-validation error were chosen for each of the learning algorithms on each dataset. The parameters included the C and the γ values in the case of the SVM, the penalty parameter p and the best number of features s for the SCM, the confidence parameter for pruning C and the minimum leaf nodes in the case of decision trees, and the number of iterations in the case of Adaboost. The algorithms were then trained with the chosen parameter values on the training set. The final classifier output by each algorithm was then tested on the test set.

The details of the datasets are provided in Appendix 9.2: Table A.1. The columns $|S|$ and $|T|$ refers to the number of examples in the training and the test sets respectively.

In this section, we present results of testing each of the classifier on the four data sets in Table 3.1. The results of testing each of the classifier on more data sets are presented in Tables B.1, B.2 and B.3 in Appendix 9.2. In these tables, the column labeled R_T denotes the empirical risk of the classifier on the test set, the columns CI denotes the confidence interval obtained using the Gaussian assumption on the sampling distribution of the empirical risk. The lower (CI_l) and the upper (CI_u) limits of the interval are the two standard deviation limits around the empirical risk.

$$CI_l = R_T(h) - 2\frac{\hat{\sigma}}{\sqrt{|T|}},\ CI_u = R_T(h) + 2\frac{\hat{\sigma}}{\sqrt{|T|}}.$$

The variance $\hat{\sigma}$ of the risk is obtained on the test set data samples with the empirical risk assumed as the mean $\hat{\mu}$ of the distribution (see [28, 30] for more detail).

$$\hat{\mu} = R_T(h),\ \hat{\sigma}^2 = \frac{1}{|T|-1}\sum_{i=1}^{|T|}(I(h(x_i) \neq y_i) - \hat{\mu})^2$$

Finally, the B_l and B_u columns denote, respectively, the lower and upper intervals generated from computing the lower and upper risk bounds of Theorems 3.1.2 and 3.1.1 of Section 3.1 with $\delta = 0.025$. This value of δ is chosen to obtain the intervals comparable to the two standard deviations intervals obtained with the Gaussian assumption approach.

As mentioned above, the risk bound technique can be considered as an alternate approach to obtain confidence intervals around the empirical risk of the classifiers. It is different from the traditional confidence interval technique in the sense that the empirical risk is modeled as a binomial distribution. In contrast, the classical approach to obtain confidence intervals makes an implicit use of the central limit theorem in imposing an asymptotic Gaussian assumption on the distribution of the empirical risk considering the true risk to be fixed. However, for lower values of the empirical risk (closer to zero), this assumption rarely, if ever, holds. As a result the limits of the confidence intervals obtained using the classical technique are either overly pessimistic (the upper limits) or overly optimistic (the lower limits). Moreover, the limits of these intervals are also not restricted to the [0, 1] intervals rendering them meaningless in most scenarios. For instance, upper limits of the confidence interval around the empirical risk exceeding unity can hardly be interpreted. Indeed, the empirical risk of the classifier, by definition, should always be constrained in the [0, 1], and so should be its true risk. Hence, obtaining confidence intervals that spill over this known interval do not make much sense. On the other hand, the risk bound approach is guaranteed to lie in the [0, 1] interval. Moreover, as we also saw in the results presented in Tables 3.1.1 (and also Tables B.1, B.2 and B.3 in Appendix 9.2) this technique allows us to obtain tight intervals in practice. The upper bound never results in an overly pessimistic estimate greater than 1 while the lower bound never becomes too optimistic. Furthermore, the confidence interval technique cannot yield a confidence interval in the case when the observed empirical risk is zero. This can be seen directly since the resulting Gaussian in this case has both a zero mean and a zero variance. Hence, in the case of zero empirical risk, the confidence interval technique becomes overly optimistic. The risk bound on the other hand, still yields a finite upper bound (of course very small since $R_T(h) = 0$). Hence, we show empirically how a test set approach yields more realistic estimates on the limits of the confidence intervals and make a case for its wider use.

However, one of the drawbacks to the test set approach is that the examples used for training cannot be used for testing while in some cases a few extra examples make output classifier more accurate. Another drawback is that splitting the original dataset into a test set and a training set might result in cases where there are insufficient examples in the training set. In those cases, the assumption that the behavior of the training set accuracy is close to the true error dose not hold anymore. Training set bounds including VC [53] and Sample compression bounds are alternative approaches to bound the future error rate of the learned classifier. Sample compression bounds [39] can result in practical realizable bounds on the true risk of the classifier. In these bounds, all the examples can be used for both learning and bounding. Indeed, training set based bounds can be used to design learning algorithms in the following way. Given any such bound, learning algorithms should find a classifier that minimizes the given risk bound. This way, the tighter the bound is, the better the performance the algorithm might achieve. In Chapter 6 such an approach will be proposed. Moreover, training set bounds provide insights into the learning problems [30]. In the following chapters, we present two training set bounds: the PAC-Bayes bound and the sample compression bound.

Chapter 4

Sample Compression Learning

In the previous chapter, we presented the test set bound to bound the future error rate of the learned classifier. We also discussed the drawbacks of using this approach. These issues motivate us to study training set bounds as an alternative approach. In this chapter, we first discuss the sample compression setting. In this setting, the set of classifiers are constructed using a subset of the examples of the training set. We then present the sample compression training set bounds in [[31]]. These bounds are obtained from test set bounds via a wise utilization of the union bound.

4.1 Sample Compression Setting

In the sample compression setting, the returned classifier (called here the sc-classifier) by the learning algorithm is encoded by a subset of the training set and a message. More formally, given a training sequence $S = \langle z_1, .., z_m \rangle = \langle (\mathbf{x}_1, y_1), .., (\mathbf{x}_m, y_m) \rangle$, each sc-classifier is described by a subsequence $S_\mathbf{i}$ of S called the *compression sequence*, and a *message* μ which represents the additional information needed to obtain a classifier from $S_\mathbf{i}$. Given S, the compression sequence $S_\mathbf{i}$ is defined by the following vector $\mathbf{i}$ of indices:

$$\mathbf{i} \quad \stackrel{\text{def}}{=} \quad \langle i_1, i_2, \ldots, i_{|\mathbf{i}|} \rangle \,,$$

with $1 \leq i_1 < i_2 < \ldots < i_{|\mathbf{i}|} \leq m$. The number of indices present in $\mathbf{i}$ is denoted by $|\mathbf{i}|$, and the vector of indices of a sc-classifier h by $\mathbf{i}_h$. The set of all the 2^m possible vectors of indices is denoted by $\mathcal{I}$. The fact that each sc-classifier is described by a compression sequence and a message implies that there exists a *reconstruction function* $\mathcal{R}$ that outputs a classifier

$$h^{\mu}_{S'} \overset{\text{def}}{=} \mathcal{R}(S', \mu), \tag{4.1}$$

when given an arbitrary compression sequence S' and a message μ chosen from the set $\mathcal{M}_{S'}$ of all messages that can be supplied with the compression sequence S'. $\mathcal{M}_{S'}$ must be defined a priori (before observing S) for all possible sequences S' of size at most m of elements of $\mathcal{X} \times \mathcal{Y}$. The messages can be strings or values taken from a continuous set. Given a training set S, we denote by $\mathcal{H}^S$ the set of all sc-classifiers $\mathcal{R}(S_{\mathbf{i}}, \mu)$ such that $\mu \in \mathcal{M}_{S_{\mathbf{i}}}$ and $\mathbf{i} \in \mathcal{I}$. Also, let us denote $\mathcal{R}(S_{\mathbf{i}}, \mu)$ by $h^{\mu}_{S_{\mathbf{i}}}$.

As examples of learning algorithms that output sc-classifier, let us point out the perceptron learning rule and the SVM algorithm where the final classifier can be reconstructed solely from a compression sequence [21]. In the case of the perceptron, the compression sequence is the set of all examples used to update the weight vector and threshold of the separating hyperplane (classifier). Indeed, applying the perceptron algorithm to the compression set will output the same classifier. Hence, the reconstruction function is the perceptron algorithm itself. For the SVM, the compression set is the set of all support vectors, and again, the reconstruction function is the algorithm itself. In contrast, the reconstruction function of the Set Covering Machine (SCM) [39] needs both a compression set and a message string. SCM constructs the smallest possible conjunction of *boolean-valued features*. Each feature is a *ball* identified by two training examples; the center and the border. The compression set is the set of examples used to construct the (features) balls. The message string specifies which examples of the compression set is a center of balls. Also, most of the time additional information is needed to determine for each center which example is the border example associated with it (see Section 4.4 for more details).

4.2 Calculating the Risk

The risk $R_D(h^{\mu}_{S_{\mathbf{i}}})$ (or generalization error) of any sc-classifier $h^{\mu}_{S_{\mathbf{i}}}$ is defined as:

$$R_D(h^{\mu}_{S_{\mathbf{i}}}) \overset{\text{def}}{=} \underset{(\mathbf{x},y)\sim D}{\mathbf{E}} I(h^{\mu}_{S_{\mathbf{i}}}(\mathbf{x}) \neq y) = \underset{(\mathbf{x},y)\sim D}{\Pr} (h^{\mu}_{S_{\mathbf{i}}}(\mathbf{x}) \neq y) \tag{4.2}$$

where $I(a) = 1$ if predicate a is true.

Depending on the learning algorithm the *empirical risk* $R_S(h^{\mu}_{S_{\mathbf{i}}})$ on S of any sc-classifier $h^{\mu}_{S_{\mathbf{i}}}$ is defined in one of the following ways:

$$(1) : R_S(h^{\mu}_{S_{\mathbf{i}}}) \overset{\text{def}}{=} \mathop{\mathbf{E}}_{(x,y)\sim S/S_{\mathbf{i}}} I(h^{\mu}_{S_{\mathbf{i}}}(\mathbf{x}) \neq y) \overset{\text{def}}{=} \frac{1}{m - |\mathbf{i}|} \sum_{i=1}^{m} I(h^{\mu}_{S_{\mathbf{i}}}(\mathbf{x}_i) \neq y_i) I((\mathbf{x}_i, y_i) \notin S_{\mathbf{i}}). \tag{4.3}$$

$$(2) : R_S(h^{\mu}_{S_{\mathbf{i}}}) \overset{\text{def}}{=} \mathop{\mathbf{E}}_{(x,y)\sim S} I(h^{\mu}_{S_{\mathbf{i}}}(\mathbf{x}) \neq y) \overset{\text{def}}{=} \frac{1}{m} \sum_{i=1}^{m} I(h^{\mu}_{S_{\mathbf{i}}}(\mathbf{x}_i) \neq y_i). \tag{4.4}$$

where $I(a) = 1$ if predicate a is true and 0 otherwise, and where $(\mathbf{x}, y) \sim S/S_{\mathbf{i}}$ means that (x, y) is drawn according to the uniform distribution on $S/S_{\mathbf{i}}$.

As we can see in Equation (4.4), the empirical risk is a biased estimate of the true risk. The bias comes from the elements of S that are in the compression sequence. This is not the case in Equation (4.3). To be more precise, note that with the i.i.d assumption, in the classical (non-sample compressed) setting, $mR_S(h)$ is a random variable that follows a binomial law with parameters $(m, R_D(h))$. As we said earlier, this is no longer the case in Equation (4.4) because the risk can then be biased by the elements of S that are in the compression sequence. However, if $a_h \overset{\text{def}}{=} \sum_{(\mathbf{x},y)\in S_{\mathbf{i}}} I(h^{\mu}_{S_{\mathbf{i}}}(\mathbf{x} \neq y)$, then $mR_S(h) - a_h$ is a binomial random variable with parameters $(m - |\mathbf{i}|, R_D(h))$. In the case of Equation (4.3), $mR_S(h)$ is a binomial random variable with parameters $(m - |\mathbf{i}|, R_D(h))$. On the other hand, the definition of Equation (4.4) can be of our interest if we are dealing with sample compressed algorithms that among other things try to minimize the empirical risk. This is especially true if the algorithm is designed to be very confident on classifying the examples of the compression set and on examples that are close to these examples. In this case, we want the algorithm to take into account its performance on the examples of the compression sequence. Using Equation (4.4) is one way to achieve that.

4.3 Sample Compression Risk Bound

In this section, we present the sample compressed bound presented in [[31]]. The proposed risk bound is a generalization of the sample-compression risk bound of [30] to the case where part of the data-compression information is given by the message. It also has the property to reduce to the Occam's Razor bound (see [30]) when the sample compression set vanishes.

We define priors over $\mathcal{I} \times \mathcal{M}_S$ for any possible $S \in D^m$. Moreover, for any given S, we will consider only the priors P_S that can be factorized as

$$P_S(h^{\mu}_{S_{\mathbf{i}}}) = P_{\mathcal{I}}(\mathbf{i})P_{S_{\mathbf{i}}}(\mu)$$

where $P_{\mathcal{I}}(\mathbf{i})$ is the prior probability of using the vector $\mathbf{i}$ of indices as defined above and where $P_{S_{\mathbf{i}}}(\mu)$ is the prior probability of using the message μ given that we use the compression set $S_{\mathbf{i}}$. The proposed bound in Theorem 4.3.1 applies to any compression set-dependent distribution of messages $P_{S_{\mathbf{i}}}(\mu)$ satisfying:

$$\sum_{\mu \in \mathcal{M}_{S_{\mathbf{i}}}} P_{S_{\mathbf{i}}}(\mu) \leq 1 \quad \forall S_{\mathbf{i}} \tag{4.5}$$

and any prior distribution $P_{\mathcal{I}}$ of vectors of indices satisfying:

$$\sum_{\mathbf{i} \in \mathcal{I}} P_{\mathcal{I}}(\mathbf{i}) \leq 1 \tag{4.6}$$

Theorem 4.3.1. *[[31]] For any sample compressed classifier $h^{\mu}_{S_{\mathbf{i}}}$, for any prior distribution $P_{\mathcal{I}}$ of vectors of indices, for any compression set-dependent distribution of messages $P_{S_{\mathbf{i}}}(\mu)$, and for any $\delta \in (0, 1]$, we have:*

$$\Pr_{S \sim D^m}\Bigg(\forall \mathbf{i} \in \mathcal{I}, \forall \mu \in \mathcal{M}_{S_{\mathbf{i}}} \colon R_D(h^{\mu}_{S_{\mathbf{i}}}) \leq \\ \overline{\mathrm{Bin}}\Big((m - |\mathbf{i}|)R_S(h^{\mu}_{S_{\mathbf{i}}}), (m - |\mathbf{i}|), P_{\mathcal{I}}(\mathbf{i})P_{S_{\mathbf{i}}}(\mu)\delta\Big)\Bigg) \geq 1 - \delta$$

where $\overline{\mathrm{Bin}}$ is the inversion of a binomial tail function defined in Equation (3.2).

Proof. We seek a tight risk bound for arbitrary reconstruction functions that holds uniformly for all compression sets and messages. To obtain the tightest possible risk bound, we fully exploit the fact that the distribution of classification errors is a binomial and make use of the Theorem 3.1.1.(see Chapter 3, Section 3.1).

Consider:

$$P' = \Pr_{S\sim D^m}\Bigg(\exists \mathbf{i} \in \mathcal{I} \colon \exists \mu \in \mathcal{M}_{S_{\mathbf{i}}} \colon R_D(h^{\mu}_{S_{\mathbf{i}}}) > \\ \overline{\mathrm{Bin}}\Big((m-|\mathbf{i}|)R_S(h^{\mu}_{S_{\mathbf{i}}}), (m-|\mathbf{i}|), P_{\mathcal{I}}(\mathbf{i})P_{S_{\mathbf{i}}}(\mu)\delta\Big)\Bigg)$$

To prove the theorem, we show that $P' \leq \delta$. Since $\Pr_{S\sim D^m}(\cdot) = \mathbf{E}_{S_{\mathbf{i}}}\ \Pr_{S/S_{\mathbf{i}}|S_{\mathbf{i}}}(\cdot)$, the union bound, Theorem 3.1.1 and Equations 4.5, and 4.6 imply that we have:

$$\begin{aligned} P' &\leq \sum_{\mathbf{i}\in\mathcal{I}} \mathbf{E}_{S_{\mathbf{i}}} \sum_{\mu\in\mathcal{M}_{S_{\mathbf{i}}}} \Pr_{S/S_{\mathbf{i}}|S_{\mathbf{i}}}\Big(R_D(h^{\mu}_{S_{\mathbf{i}}}) > \overline{\mathrm{Bin}}\Big((m-|\mathbf{i}|)R_D(h^{\mu}_{S_{\mathbf{i}}}), (m-|\mathbf{i}|), P_{\mathcal{I}}(\mathbf{i})P_{S_{\mathbf{i}}}(\mu)\delta\Big)\Big) \\ &\leq \sum_{\mathbf{i}\in\mathcal{I}} \mathbf{E}_{S_{\mathbf{i}}} \sum_{\mu\in\mathcal{M}_{S_{\mathbf{i}}}} P_{\mathcal{I}}(\mathbf{i})P_{S_{\mathbf{i}}}(\mu)\delta \\ &\leq \delta \end{aligned}$$

□

By inverting a standard approximation of the binomial tail using Lemma 4.3.2, the bound of Theorem 4.3.1 is rewritten into the bound of Theorem 4.3.3.

Lemma 4.3.2. *[[31]] For any integer $m \geq 1$ and $k \in \{0,\ldots,m\}$, we have:*

$$\overline{\mathrm{Bin}}\,(k,m,\delta) \leq 1 - \exp\left(\frac{-1}{m-k}\left[\ln\binom{m}{k} + \ln\left(\frac{1}{\delta}\right)\right]\right) \tag{4.7}$$

$$\leq \frac{1}{m-k}\left[\ln\binom{m}{k} + \ln\left(\frac{1}{\delta}\right)\right] \tag{4.8}$$

Proof. We first show that

$$\mathrm{Bin}\,(m,k,r) \stackrel{\mathrm{def}}{=} \sum_{i=0}^{k}\binom{m}{i}r^i(1-r)^{m-i} \leq \binom{m}{k}(1-r)^{m-k}$$

Let h be a classifier with risk $R_D(h) = r$. Recall that the binomial tail distribution $\mathrm{Bin}\,(m,k,r)$ associated with a classifier of (true) risk r is defined as the probability that this classifier makes at most k errors on a test set of m examples:

$$\begin{aligned}
\mathrm{Bin}\left(m,k,r\right) &\overset{\text{def}}{=} \sum_{i=0}^{k}\binom{m}{i}r^{i}\left(1-r\right)^{m-i}\\
&= \Pr\exists S'\subseteq\{1,2,..,m\}\ \textit{such that}\ |S'|=m-k\wedge R_{S'}(h)=0\\
&\leq \sum_{S'\subseteq\{1,\ldots,m\}:\ |S'|=m-k}\Pr\ \{R_{S'}(h)=0\}\quad\text{(the union bound)}\\
&= \binom{m}{m-k}(1-r)^{m-k} = \binom{m}{k}(1-r)^{m-k}\\
&\overset{\text{def}}{=} g\left(m,k,r\right)
\end{aligned}$$

Since the tail of the binomial is a decreasing function of r when k and m are fixed, it follows that:

$$\begin{aligned}
\overline{\mathrm{Bin}}\left(m,k,\delta\right) &\overset{\text{def}}{=} \sup\left\{r:\mathrm{Bin}\left(k,m,r\right)\geq\delta\right\}\\
&\leq \sup\{r\colon g\left(m,k,r\right)\geq\delta\}\\
&= \{r:g\left(m,k,r\right)=\delta\}
\end{aligned}$$

Now, note that the value of r that satisfies the equation $g\left(m,k,r\right)=\delta$ is precisely given by:

$$r = 1-\exp\left[-\frac{1}{m-k}\left(\ln\binom{m}{k}+\ln\frac{1}{\delta}\right)\right]$$

Hence,

$$\overline{\mathrm{Bin}}\left(m,k,\delta\right) \leq 1-\exp\left[-\frac{1}{m-k}\left(\ln\binom{m}{k}+\ln\frac{1}{\delta}\right)\right]$$

□

We, therefore, have the following relaxation of Theorem 4.3.1:

Theorem 4.3.3. *[[31]] For any sample compressed classifier $h^{\mu}_{S_{\mathrm{i}}}$, for any prior distribution $P_{\mathcal{I}}$ of vectors of indices, for any compression set-dependent distribution of* *messages $P_{S_\mathbf{i}}(\mu)$, and for any $\delta \in (0,1]$, we have:*

$$\Pr_{S\sim D^m}\Bigg(\forall \mathbf{i} \in \mathcal{I}, \forall \mu \in \mathcal{M}_{S_\mathbf{i}} \colon R_D(h^\mu_{S_\mathbf{i}}) \leq 1 - \exp\left(\frac{-1}{m-d-k}\left[\ln\binom{m-d}{k} + \ln\left(\frac{1}{P_\mathcal{I}(\mathbf{i})P_{S_\mathbf{i}}(\mu)\delta}\right)\right]\right)\Bigg) \geq 1-\delta \quad (4.9)$$

and, consequently:

$$\Pr_{S\sim D^m}\Bigg(\forall \mathbf{i} \in \mathcal{I}, \forall \mu \in \mathcal{M}_{S_\mathbf{i}} \colon R_D(h^\mu_{S_\mathbf{i}}) \leq \frac{1}{m-d-k}\left[\ln\binom{m-d}{k} + \ln\left(\frac{1}{P_\mathcal{I}(\mathbf{i}))P_{S_\mathbf{i}}(\mu)\delta}\right)\right]\Bigg) \geq 1-\delta \quad (4.10)$$

where $d \stackrel{\text{def}}{=} |\mathbf{i}|$ is the sample compression set size of classifier $h^\mu_{S_\mathbf{i}}$ and $k \stackrel{\text{def}}{=} (m-\mathbf{i})R_S(h^\mu_{S_\mathbf{i}})$ is the number of training errors that this classifier makes on the examples that are not in the compression set.

Proof. The proof is straightforward by using the following inequality given in Lemma 4.3.2 in Theorem 4.3.1:

$$\overline{\mathrm{Bin}}\Big((m-|\mathbf{i}|), (m-|\mathbf{i}|)R_S(h^\mu_{S_\mathbf{i}}), P_\mathcal{I}(\mathbf{i})P_{S_\mathbf{i}}(\mu)\delta\Big) \leq 1 - \exp\left(\frac{-1}{m-d-k}\left[\ln\binom{m-d}{k} + \ln\left(\frac{1}{P_\mathcal{I}(\mathbf{i})P_{S_\mathbf{i}}(\mu)\delta}\right)\right]\right)$$

where $d \stackrel{\text{def}}{=} |\mathbf{i}|$ is the sample compression set size of classifier $h^\mu_{S_\mathbf{i}}$ and $k \stackrel{\text{def}}{=} (m-\mathbf{i})R_S(h^\mu_{S_\mathbf{i}})$ is the number of training errors that this classifier makes on the examples that are not in the compression set. □

As we can see in Theorem 4.3.3, the risk bound of classifier $h^\mu_{S_\mathbf{i}}$ is small when its compression set size d and its number k of training errors are both much smaller than the number m of training examples. The bound of Equation (4.9) is very similar to, and slightly tighter than, the recent bound of [40] owing to the more efficient treatment of errors by the binomial tail inversion.

The bound of Equation (4.10) is similar to the bounds of [36] and [14] when the set $\mathcal{M}$ of all possible messages is independent of the compression set $S_\mathbf{i}$ and when we choose [[31]]:

$$P_{S_\mathbf{i}}(\mu) = 1/|\mathcal{M}| \quad \forall \mu \in \mathcal{M} \quad (4.11)$$

$$P_{\mathcal{I}}(\mathbf{i}) = \binom{m}{|\mathbf{i}|}^{-1}(m+1)^{-1} \quad \forall \mathbf{i} \in \mathcal{I} \tag{4.12}$$

However, other choices that give better bounds are clearly possible. For example, we can choose:

$$P_{\mathcal{I}}(\mathbf{i}) = \binom{m}{|\mathbf{i}|}^{-1}\zeta(|\mathbf{i}|) \text{ with } \zeta(a) \stackrel{\text{def}}{=} \frac{6}{\pi^2}(a+1)^{-2} \quad \forall a \in \mathbb{N} \tag{4.13}$$

which satisfies the constraint of Equation (4.6) since $\sum_{i=1}^{\infty} i^{-2} = \pi^2/6$. This choice for $P_{\mathcal{I}}$ has the advantage that the risk bounds do not deteriorate too rapidly when $|\mathbf{i}|$ increases. But clearly, since the number of compression sets of size $|\mathbf{i}|$ increases rapidly with $|\mathbf{i}|$, a good choice for $P_{\mathcal{I}}$ is the one that gives more weight to smaller compression sets. The bound of Theorem 4.3.3 then indicates that a good classifier should not only have a good performance on the training set (low empirical risk on $S/S_{\mathbf{i}}$), but also should have a small compression set. Thus, the bounds of Theorems 4.3.1 and 4.3.3 express the importance of looking for an empirical accuracy–sparsity trade off.

In the next section, we present the Set covering machine algorithm (SCM) [39] . This is a sample compression algorithms that expresses this empirical accuracy–sparsity trade-off. We also present the application of the presented sample compression bound to SCM [[31]].

4.4 Set Covering Machine (SCM)

The Set Covering Machine (SCM) was proposed by [39]. The Set Covering Machine algorithm is the generalized form of the two-step algorithm which was proposed by [52, 24]. The set covering Machine extends this algorithm for learning conjunctions [1] of boolean attributes over arbitrary sets of boolean features which are constructed from data (i.e., *Data dependent*). This learning algorithm also provides some learning parameters which controls the trade off between the accuracy and the size of the conjunction. In this section, we will give a brief explanation of the SCM algorithm that uses *data dependent balls* as its set of features.

We consider classification problems where the input space $\mathcal{X}$ consists of an arbitrary subset of $\mathbb{R}^n$ and the output space $\mathcal{Y} = \{-1, +1\}$. Let a training set $S =$

[1] In the set covering machine, we can also consider the disjunction of the boolean attributes. In this thesis, we just consider the conjunction case.

$\{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_m, y_m)\}$ consists of two parts ($S = P \cup N$), the set of positive examples P and the set of negative examples N. In this case for each $(\mathbf{x}_i, y_i) \in S$, we have:

$$y_i = \begin{cases} 1 & if\ \mathbf{x}_i \in P \\ -1 & if\ \mathbf{x}_i \in N \end{cases}$$

Let h_i be a *feature* where a feature is an arbitrary boolean valued function that maps $\mathcal{X}$ to $\{0, 1\}$. Let $\mathcal{F} = \{h_i\}_{i=1}^{|\mathcal{F}|}$ be any set of features . The learning algorithm then returns a small subset $F \subset \mathcal{F}$ of features when given any such set $\mathcal{F}$. Given this subset F and an arbitrary input vector $\mathbf{x} \in \mathcal{X}$, the output of SCM is defined to be[2]:

$$h_{SCM}(\mathbf{x}) = \wedge_{i \in F} h_i(\mathbf{x})$$

$h_{SCM}(x)$ outputs True (1) if all $h(x) \in F$ are True and False otherwise. Note that, here we use the value of -1 to denote the output of 0 (False) for the conjunction.

We describe the SCM for the case where the set of features $\mathcal{F}$ is constructed from the data (data-dependent features). The set of data dependent features that we use for the SCM is the set of data dependent balls which has been introduced by [38]. In the following, we present this set of features.

Let $d : \mathbf{x} \times \mathbf{x} \mapsto \mathbb{R}^+$ be a metric which defines the distance $d(\mathbf{x}, \mathbf{x}_i)$ between a pair of points of $\mathbf{x}_i$ and $\mathbf{x}$. Let $h_{\rho,i}$ be a feature identified by a center i and raduis ρ. We define $h_{\rho,i}$ be the following data dependent ball centered on $\mathbf{x}_i$:

$$h_{\rho,i}(\mathbf{x}) = \begin{cases} y_i & \text{if } d(\mathbf{x}, \mathbf{x}_i) \leq \rho \\ -y_i & \text{otherwise} \end{cases} \tag{4.14}$$

where $\rho = d(\mathbf{x}_i, \mathbf{x}_j) + \varepsilon$ if $\mathbf{x}_i \in P$, $\rho = d(\mathbf{x}_i, \mathbf{x}_j) - \varepsilon$ if $\mathbf{x}_i \in N$, $\mathbf{x}_j$ is the border point and ε is a small positive real number. Note that a center $\mathbf{x}_i$ is defined by every example in the training set, however, a ball border $\mathbf{x}_j$ is defined from the set of P (positive) examples. This way, the compression set $S_\mathbf{i}$ consists of examples denoting the centers and borders of the balls. Given a compression set $S_\mathbf{i}$, we need to specify the examples in $S_\mathbf{i}$ that are used for the border point without being used as a center. Recall that, each border point is defined from the set of P (positive) examples. Thus, each message $\mu \in \mathcal{M}_{S_\mathbf{i}}$ just needs to specify the positive examples that are the border point without being a center.

[2] In the case of disjunction, the output of the SCM is defined to be: $h(\mathbf{x}) = \vee_{i \in F} h_i(\mathbf{x})$

4.4.1 The SCM Learning Algorithm

In the following, we present the SCM learning algorithm. Note that, here, a ball (feature) $h_{\rho,i}$ is sometimes used interchangeably with h. The SCM algorithm is divided into two parts. The first part simply generates a set of SCMs. We will call this part the SCMs generator algorithm. The second part will select a single SCM among those SCMs that have been generated by the first part. We will call this second part the SCM model selector.

The SCM Generator Algorithm

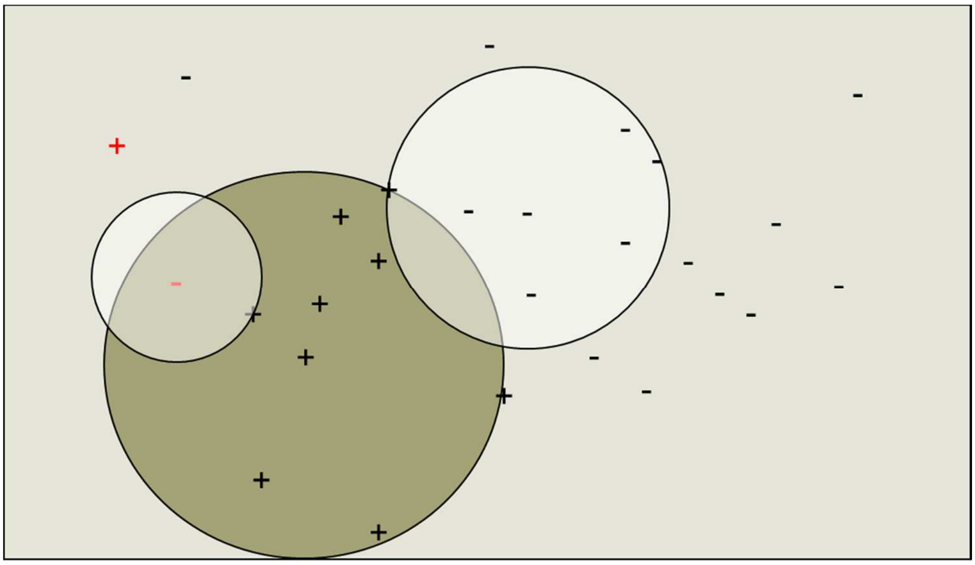

Figure 4.1: An example of SCM generator algorithm. One ball is added at a time, in this example, a classification error on an example "+" cannot be fixed by adding other balls. However, it is possible for a classification error on an example "-" to be fixed later.

Let Q_i be the subset of examples of N on which feature h_i makes no error. Thus, h_{SCM} makes no error on N iff $\cup_{i \in F} Q_i = N$. Hence, the problem of finding the smallest set F for which h_{SCM} makes no training error can be considered as the problem of finding the smallest collection of $Q_i s$ that covers all N, which is the well known *minimum set cover problem*[16]. Although this problem is NP-hard and it is hard to find the set cover of minimum size, the greedy algorithm of the minimum set covering problem will always find a cover of size $z \ln(|N|)$ if the smallest cover is of size z [11, 29]. The set covering greedy algorithm is a simple algorithm : first choose the set Q_i which covers the largest number of elements in N, remove from N and each Q_j the elements that are in Q_i, then repeat this process of finding the set Q_k of largest cardinality and updating N and each Q_j until there are no more elements in N.

As mentioned earlier, the Set Covering Machine algorithm provides learning parameters which control the trade-off between the accuracy and the size of the conjunction. Hence, in general we are not looking for a SCM with zero training error, but instead we are looking for a "small" SCM which makes a few errors on the training set. Theorems 4.3.1 and 4.3.3 point out that this might give a better generalization than a larger SCM with more features that makes zero training error. Hence, there is a sparsity-empirical accuracy trade-off here. One way to control this trade-off is to stop the set covering algorithm when there still exist some training examples to be covered. In this case, the SCM has fewer features and also makes errors on those training examples which are not covered (see Figure 4.1). According to the algorithm, the training examples which are not covered by SCM all belong to N and, since it is not suitable in general to make all the errors in N, early stopping is not sufficient.

Hence, to include the flexibility in choosing the proper trade off between complexity and accuracy, each greedy step will be modified as follows. Let Q_h be the set of examples in N on which a ball h makes no error and also let R_h be the set of examples in P on which a ball h makes an error. Given that, each example in P misclassified by h should decrease by some fixed *penalty* p the "usefulness" of ball (feature) h. The usefulness U_h of ball(feature) h is defined by the following equation:

$$U_h = |Q_h| - p.\,|R_h| \tag{4.15}$$

Note that parameter p in Equation (4.15) gives the trade-off that will be used in the greedy algorithm at each step of the construction (that is, a step of the algorithm consisting of choosing a new ball (feature)) . A value of $p = 1$ means that at each step of the construction, we consider that making an error on a positive example is as bad as making an error on a negative one. This might not be a good idea since as shown in Figure 4.1, an error on a positive example on a single ball implies an error in the conjunction. However, an error on a negative example is not so dramatic since it can be corrected later on during the construction. For this reason, it might be better to choose values of p that are greater than one. This way, an error on a positive example at a given step of the algorithm has a bigger long-run impact than an error on a negative example. The good value of p is a non trivial trade-off which is determined either by cross validation or by referring to the bounds of Theorems 4.3.1 and 4.3.3 (the later case is presented in Algorithm 2). Thus, the set covering greedy algorithm is modified as follows: instead of using the feature that covers the largest number of examples in N, the feature $h \in \mathcal{F}$ that has the highest usefulness value U_h is used. We remove from N and from each Q_g (for $g \neq h$) the elements that are in Q_h and we remove from each R_g (for $g \neq h$) the elements that are in R_h. Note that, we update each such set R_g because a feature g that makes an error on an example in P does not increase the error of the machine if another feature h is already making an error on that example. We repeat this process of finding the feature h of largest usefulness U_h and updating N, and each Q_g and R_g, until N is empty or until the early stopping criterion $|F| > s$ is reached (where s is some positive integer number). Therefore, SCM contains a stopping parameter s that stops the SCM early so that a smaller conjunction of features is being returned (see Algorithm 1).

Algorithm 1 :(SCM Generator Algorithm)

```
1: Initialize: $F \leftarrow \emptyset$, a set of data dependent balls $\mathcal{F} = \{h_i\}_{i=1}^{|\mathcal{F}|}$, a stopping parameter
   $s$ and penalty parameter $p$.
2: For each $h_i \in \mathcal{F}$ let $Q_i$ be the set of examples in $N$ on which $h_i$ makes no error and
   let $R_i$ be the set of examples in $P$ on which $h_i$ makes errors.
3: repeat
4:     Choose a new ball $h_k \in \mathcal{F}/F$ that maximizes Equation (4.15). $(|Q_k| - p.|R_k|)$.
5:     Update $F \leftarrow F \cup \{k\}$, $N \leftarrow (N - Q_k)$ and $P \leftarrow (P - R_k)$.
6:     for all $i \in \{1, \cdots |\mathcal{F}|\}$ do: $Q_i \leftarrow Q_i - Q_k$ and $R_i \leftarrow R_i - R_k$
7: until $(|F| > s$ or $N = \emptyset)$
8: Return $h_{SCM}(\mathbf{x}) = \wedge_{i \in F} h_i(\mathbf{x})$ and $F$
```

The SCM Selector Algorithm

As we can see in Algorithm 1, the penalty value p and early stopping point s give us the ability to control the proper trade off between the learning accuracy and the size of the conjunction. The penalty p and early stopping point s are two model-selection parameters. The values of these two parameters are determined by using k-fold cross-validation or by computing the objective function f that, given any SCM, outputs a real value. This objective function can be a training set bound similar to the bounds of Theorems 4.3.1 and 4.3.3. The goal is to find a SCM that minimizes this objective function f. To fulfill this goal, we use Algorithm 2. Note that, in the case of using cross validation, we can still use Algorithm 2 where the objective function f is the k-fold cross validation risk.

Algorithm 2 :(SCM Model Selector)

```
1: Initialize: Define a list of parameter $\mathcal{P} = \{p_1, p_2, \cdots, p_p\}$ as penalty values and a
   list of parameter $\mathcal{S} = \{1, 2, \cdots, s\}$ as early stopping points. Let $f$ be an objective
   function.
   $f_{min} \leftarrow \infty$.
2: For each pair of parameters $(p, s)$ where $p \in \mathcal{P}$ and $s \in \mathcal{S}$. (Do step 3 to 5)
3: Run Algorithm 1 for parameters $(p, s)$, and let $h_{SCM}$ be the SCM it outputs.
4: Compute $f_F$: the value of $f$ for $h_{SCM}$.
5: if $f_F \leq f_{min}$ then
         $f_{min} \leftarrow f_F$ and $h_{SCM}^{min} \leftarrow h_{SCM}$
6: return $h_{SCM}^{min}$
```

Chapter 5

Sample Compressed PAC-Bayes Theorems

In this chapter, we present another type of training set bounds, called the PAC-Bayes bounds. We also presents a PAC-Bayes bound proposed by [33] for the data-dependent setting (sample compression).

5.1 PAC-Bayes Bounds

The PAC-Bayes approach was initiated by [42]. It aims at providing PAC guarantees to "Bayesian" learning algorithms. Bayesian algorithms are generally specified in terms of a *prior distribution* P over a space of classifiers and a *posterior distribution* Q (over the same space of classifiers). The prior distribution characterizes our prior belief about good classifiers (before the observation of the data). On the other hand, the posterior distribution takes into account the additional information provided by the training data. The "PAC-Bayes theorem", provides a tight upper bound on the risk of a stochastic classifier called the *Gibbs classifier*.

Recall from Section 2.2 that given an input example $\mathbf{x}$, the label $G_Q(\mathbf{x})$ assigned to $\mathbf{x}$ by the Gibbs classifier is defined by the following process. We first choose a classifier h according to the posterior distribution Q and then use h to assign the label $h(\mathbf{x})$ to $\mathbf{x}$. As we have seen before (Section 2.2), the risk of G_Q is defined as the expected risk of classifiers drawn according to Q:

$$R_D(G_Q) \stackrel{\text{def}}{=} \mathbf{E}_{h\sim Q} R(h) = \mathbf{E}_{h\sim Q} \mathbf{E}_{(\mathbf{x},y)\sim D} I(h(\mathbf{x}) \neq y) \tag{5.1}$$

In the following, we present some PAC-Bayes bounds. The following quantities are part of these bounds:

- $\mathrm{KL}(Q\|P)$: is the Kullback-Leibler divergence between distributions Q and P:
  $$\mathrm{KL}(Q\|P) \stackrel{\text{def}}{=} \mathbf{E}_{h\sim Q} \ln \frac{Q(h)}{P(h)} \tag{5.2}$$
  Note that, the Kullback-Leibler is often intuitively presented as a measure of distance between two probability distributions.

- $\mathrm{kl}(q\|p)$: is the Kullback-Leibler divergence between the Bernoulli distributions with probabilities of success q and p ($p, q \in [0,1]$):
  $$\mathrm{kl}(q\|p) \stackrel{\text{def}}{=} q \ln \frac{q}{p} + (1-q) \ln \frac{1-q}{1-p}.$$

We also define:

$$\xi(m) := \sum_{k=0}^{m} \binom{m}{k} \left(\frac{k}{m}\right)^{k} \left(1 - \frac{k}{m}\right)^{m-k},$$

where $\xi(m) \in [\sqrt{m}, 2\sqrt{m}]$ (Please see [41] for the proof).

The PAC-Bayes theorem was first proposed by [43]. The first version presented here is due to [47, 30].

Theorem 5.1.1. (Classical PAC-Bayes bound) *Given any space $\mathcal{H}$ of classifiers, for any data-independent prior distribution P over $\mathcal{H}$ and for any (possibly data-dependent) posterior distribution Q over $\mathcal{H}$, we have:*

$$\Pr_{S\sim D^m}\left(\forall Q \text{ on} \mathcal{H}\colon\ \mathrm{kl}(R_S(G_Q), R_D(G_Q)) \ \le\ \frac{1}{m}\left[\mathrm{KL}(Q\|P) + \ln \frac{\xi(m)}{\delta}\right]\right) \ \ge\ 1 - \delta\,.$$

The proof of Theorem 5.1.1 is given in Section 5.1.1.

Note that, in Theorem 5.1.1, $\mathrm{kl}(R_S(G_Q), R_D(G_Q))$ quantifies the "distance" between the true Gibbs risk and the empirical Gibbs risk which is bounded by $\frac{1}{m}\left[\mathrm{KL}(Q\|P) + \ln \frac{\xi(m)}{\delta}\right]$.

Hence, we achieve a better guarantee as m grows, but the guarantee is looser when the posterior distribution Q is "far" from the prior P (in the sense that the KL-divergence between prior P and posterior Q is big). Moreover, Theorem 5.1.1 provides both an upper bound and a lower bound on the true risk $R(G_Q)$ based on its empirical risk $R_S(G_Q)$ [33]. With probability at least $1-\delta$ over the random draws of S, $R(G_Q)$ is upper-bounded by:

$$sup\left(B : \mathrm{kl}(R_S(G_Q), B) \le \frac{1}{m}\left[\mathrm{KL}(Q\|P) + \ln\frac{\xi(m)}{\delta}\right]\right)$$

and lower bounded by:

$$inf\left(B : \mathrm{kl}(R_S(G_Q), B) \le \frac{1}{m}\left[\mathrm{KL}(Q\|P) + \ln\frac{\xi(m)}{\delta}\right]\right)$$

.

There exists also the following version of PAC-Bayes bound proposed by McAllester [42, 44] (it should be mentioned that the version presented here is slightly different from the original McAllester PAC-bayes bounds in [42, 44]).

Theorem 5.1.2. (PAC-Bayes Theorem McAllester) *Given any space $\mathcal{H}$ of classifiers, for any data-independent prior distribution P over $\mathcal{H}$ and for any (possibly data-dependent) posterior distribution Q over $\mathcal{H}$, we have:*

$$\Pr_{S\sim D^m}\left(\forall Q \; on \mathcal{H}\colon \;\; R_D(G_Q) \;\; \le \;\; R_S(G_Q) \; + \; \sqrt{\frac{\mathrm{KL}(Q\|P) + \ln\frac{\xi(m)}{\delta}}{2m}}\right) \;\; \ge \;\; 1 \; - \; \delta\,.$$

Note that Theorem 5.1.2 can be straightforwardly retrieved from Theorem 5.1.1 using the following inequality which is known as Pinsker's inequality [13]:

$$2(R_D(G_Q) - R_S(G_Q))^2 \le \mathrm{kl}(R_S(G_Q)\|R_D(G_Q)).$$

The more precise proof of Theorem 5.1.2 is given in Section 5.1.1.

In [7], Catoni proposed the following PAC-Bayes bound:

Theorem 5.1.3. (PAC-Bayes Theorem Catoni) *Given any space $\mathcal{H}$ of classifiers, for any data-independent prior distribution P over $\mathcal{H}$ and for any (possibly data-*

dependent) posterior distribution Q over $\mathcal{H}$, we have:

$$\Pr_{S\sim D^m}\left(\forall\, Q \text{ on} \mathcal{H}\colon\ R_D(G_Q) \leq \frac{1}{1-e^{-C}}\cdot\left[C\cdot R_S(G_Q)+\frac{\mathrm{KL}(Q\|P)+\ln\frac{1}{\delta}}{m}\right]\right)\geq 1-\delta\,.$$

We give the proof of Theorem 5.1.3 in Section 5.1.1[1]. The bound of Theorem 5.1.3 is an interesting bound in for deriving learning algorithm since it has a hyper-parameter C that leads to a hyper-parameter for the algorithm itself. Many interesting learning algorithms have such a hyper-parameter (SVM, L_2-regularized adaboost, etc...).

Finally, in [1], Audibert proposed a version of PAC-Bayes bound which is tighter than Theorem 5.1.2 (Mcallester bound) when $R_D(G_Q)$ is small (see [1] for the proof) . In the following we present this bound:

Theorem 5.1.4. (PAC-Bayes Theorem Audibert) *Given any space $\mathcal{H}$ of classifiers, for any data-independent prior distribution P over $\mathcal{H}$ and for any (possibly data-dependent) posterior distribution Q over $\mathcal{H}$, we have:*

$$\Pr_{S\sim D^m}\left(\forall\, Q \text{ on} \mathcal{H}\colon\ |R_D(G_Q)-R_S(G_Q)|\leq\sqrt{\frac{2R_S(G_Q)[1-R_S(G_Q)][\mathrm{KL}(Q\|P)+\ln\frac{2\sqrt{m}}{\delta}]}{m}}\right.$$
$$\left.+\frac{4[\mathrm{KL}(Q\|P)+\ln\frac{2\sqrt{m}}{\delta}]}{3m}\right)\geq 1-\delta\ .$$

It should be mentioned that the bound given by the PAC-Bayes theorem for the risk of Gibbs classifiers can be turned into a bound for the risk of Bayes classifiers in the following way. Given a posterior distribution Q, the Bayes classifier B_Q performs a majority vote (under measure Q) of binary classifiers in $\mathcal{H}$. When B_Q misclassifies an example $\mathbf{x}$, at least half of the binary classifiers (under measure Q) misclassifies $\mathbf{x}$. It follows that the error rate of G_Q is at least half of the error rate of B_Q. Hence,

$$R_D(B_Q)\leq 2R_D(G_Q). \tag{5.3}$$

5.1.1 General PAC-Bayes Theorem

Now that we presented some PAC-Bayes theorems existing in the literature, we also present a general PAC-Bayes theorem proposed by [17] from which all presented PAC-Bayes risk bounds except Theorem 5.1.4 can be obtained as particular cases.

[1] This proof is inspired by the proof given by [17].

Before stating the general PAC-Bayes theorem (Theorem 5.1.7), we present some preliminaries containing a few lemmas and definitions which make the demonstration of the Theorem 5.1.7 more comprehensible.

Lemma 5.1.5. *(Markov's inequality) for all none negative random variable X with expected value μ and for all $t \in \Re$ we have:*

$$\Pr(X > t\mu) < \frac{1}{t}.$$

Lemma 5.1.6. *(Jensen's inequality) Let f be any convex function and X any none negative random variable then we have:*

$$\boldsymbol{E}\, f(X) \geq f(\boldsymbol{E}\, X).$$

An overview of the above lemmas are presented in [37]. note that if f is any concave function then we have:

$$\boldsymbol{E}\, f(X) \leq f(\boldsymbol{E}\, X).$$

Also, often in the demonstration of the theorems we use the following property of the probability in which for all distributions P and Q on $\mathcal{H}$ and for all function $f : \mathcal{H} \to \Re$ we have:

$$\mathbf{E}_{h\sim P} f(h) \leq \mathbf{E}_{h\sim Q} \frac{P(h)}{Q(h)} f(h) \tag{5.4}$$

Now that we have all the necessary lemmas and definitions we present the general PAC-Bayes theorem proposed by [17].

Theorem 5.1.7. (General PAC-Bayes bound) *Given any space $\mathcal{H}$ of classifiers, for any data-independent prior distribution P over $\mathcal{H}$, for any (possibly data-dependent) posterior distribution Q over $\mathcal{H}$ and for all convex function $\mathcal{D}:\ [0,1] \times [0,1] \to R$, we have:*

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

Proof. This proof is based on the proof presented in [17].

Since $e^{m\mathcal{D}(R_S(h),R_D(h))}$ is a non negative random variable, by using Markov's inequality (Lemma 5.1.5) we have:

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

By taking the logarithm on each side of the innermost inequality and by transforming expectation over P to the expectation over Q using Equation (5.4) we obtain:

$$\Pr_{S\sim D^m}\left(\forall\, Q \text{ on } \mathcal{H}\colon\ \log(\mathop{\mathbf{E}}_{h\sim Q}\frac{P(h)}{Q(h)}e^{m\mathcal{D}(R_S(h),R_D(h))}) \leq \log(\frac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{h\sim P} e^{m\mathcal{D}(R_S(h),R_D(h))})\right) \geq 1-\delta$$

Using Equation (6.14) together with Jensen's inequality (Lemma 5.1.6) applied to concave $\log(x)$ gives:

$$\log(\mathop{\mathbf{E}}_{h\sim Q}\frac{P(h)}{Q(h)}e^{m\mathcal{D}(R_S(h),R_D(h))}) \geq -\mathrm{KL}(Q\|P) + \mathop{\mathbf{E}}_{h\sim Q} m\mathcal{D}(R_S(h),R_D(h))$$

Again from using Jensen's inequality applied to the convex function $\mathcal{D}(R_S(h),R_D(h))$ together with the Equation (5.1) we obtain:

$$\log(\mathop{\mathbf{E}}_{h\sim Q}\frac{P(h)}{Q(h)}e^{m\mathcal{D}(R_S(h),R_D(h))}) \geq -\mathrm{KL}(Q\|P) + m\mathcal{D}(R_S(G_Q),R_D(G_Q))$$

Therefore,

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

□

Based on the results presented in [17] by using a certain convex function $\mathcal{D}:[0,1]\times[0,1]\to\mathbb{R}$ and by upper-bounding $\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{h\sim P} e^{m\mathcal{D}(R_S(h),R_D(h))}$. we can use Theorem 5.1.7 to derive the PAC-Bayesian risk bounds that we have already presented.

Proof of Theorem 5.1.1:

The PAC-Bayes theorem 5.1.1 proposed by [47, 30] can be obtained from Theorem 5.1.7 by using $\mathcal{D}(q,p) = \mathrm{kl}(q,p)$. Indeed, in this case we have :

$$\begin{aligned}
&\underset{S\sim D^m}{\mathbf{E}}\ \underset{h\sim P}{\mathbf{E}}\ e^{m\mathrm{kl}(R_S(h),R_D(h))}\\
&\quad= \underset{h\sim P}{\mathbf{E}}\ \underset{S\sim D^m}{\mathbf{E}}\ \left(\tfrac{R_S(h)}{R_D(h)}\right)^{mR_S(h)}\left(\tfrac{1-R_S(h)}{1-R_D(h)}\right)^{m(1-R_S(h))}\\
&\quad= \underset{h\sim P}{\mathbf{E}}\ \textstyle\sum_{k=0}^{m} \underset{S\sim D^m}{\Pr}\left(R_S(h)=\tfrac{k}{m}\right)\left(\tfrac{\frac{k}{m}}{R_D(h)}\right)^{k}\left(\tfrac{1-\frac{k}{m}}{1-R_D(h)}\right)^{m-k}\\
&\quad= \underset{h\sim P}{\mathbf{E}}\ \textstyle\sum_{k=0}^{m} \binom{m}{k}(R_D(h))^k(1-R_D(h))^{m-k}\left(\tfrac{\frac{k}{m}}{R_D(h)}\right)^{k}\left(\tfrac{1-\frac{k}{m}}{1-R_D(h)}\right)^{m-k},
\end{aligned}$$

where the last equality arises from the fact that $mR_S(h)$ is a binomial random variable that follows the binomial law with parameter $(m, R_D(h))$. Thus, we have:

$$\underset{S\sim D^m}{\Pr}\left(R_S(h) = \frac{k}{m}\right) = \binom{m}{k}(R_D(h))^k(1-R_D(h))^{m-k} \tag{5.5}$$

and simplifying it gives us:

$$\underset{S\sim D^m}{\mathbf{E}}\ \underset{h\sim P}{\mathbf{E}}\ e^{m\mathrm{kl}(R_S(h),R_D(h))} = \underset{h\sim P}{\mathbf{E}}\ \sum_{k=0}^{m}\binom{m}{k}\left(\frac{k}{m}\right)^k\left(1-\frac{k}{m}\right)^{m-k} = \xi(m)$$

Now, by using $\mathcal{D}(q,p) = \mathrm{kl}(q,p)$ and the upper bound of $\underset{S\sim D^m}{\mathbf{E}}\ \underset{h\sim P}{\mathbf{E}}\ e^{m\mathcal{D}(R_S(h),R_D(h))}$ in Theorem 5.1.7 we obtain Theorem 5.1.1.

Proof of Theorem 5.1.2:

The PAC-Bayes bound of Theorem 5.1.2 can also be obtained by using $\mathcal{D}(q,p) = 2(q-p)^2$.

By upper bounding $\underset{S\sim D^m}{\mathbf{E}}\ \underset{h\sim P}{\mathbf{E}}\, e^{m\cdot 2(R_S(h)-R_D(h))^2}$ using the property $2(q-p)^2 \le \mathrm{kl}(q,p)$

we have:

$$\begin{aligned}
&\mathop{\mathbf{E}}_{S\sim D^m}\ \mathop{\mathbf{E}}_{h\sim P}\ e^{m\cdot 2(R_S(h)-R_D(h))^2} \\
&\quad\le \mathop{\mathbf{E}}_{S\sim D^m}\ \mathop{\mathbf{E}}_{h\sim P}\ e^{m\mathrm{kl}(R_S(h),R_D(h))} \\
&\quad= \mathop{\mathbf{E}}_{h\sim P}\ \mathop{\mathbf{E}}_{S\sim D^m}\ \left(\frac{R_S(h)}{R_D(h)}\right)^{mR_S(h)} \left(\frac{1-R_S(h)}{1-R_D(h)}\right)^{m(1-R_S(h))} \\
&\quad= \mathop{\mathbf{E}}_{h\sim P}\ \sum_{k=0}^{m} \Pr_{S\sim D^m}\left(R_S(h)=\frac{k}{m}\right)\left(\frac{\frac{k}{m}}{R_D(h)}\right)^{k}\left(\frac{1-\frac{k}{m}}{1-R_D(h)}\right)^{m-k} \\
&\quad= \sum_{k=0}^{m}\binom{m}{k}(k/m)^k(1-k/m)^{m-k}=\xi(m)\,,
\end{aligned}$$

The last equality obtained by using Equation (5.5). Now by using $\mathcal{D}(q,p) = 2(q-p)^2$ and the upper bound of $\mathop{\mathbf{E}}_{S\sim D^m}\ \mathop{\mathbf{E}}_{h\sim P}\ e^{m\mathcal{D}(R_S(h),R_D(h))})$ in Theorem 5.1.7 we obtain Theorem 5.1.2.

Proof of Theorem 5.1.3:

The PAC-Bayes bound of Theorem 5.1.3 is obtained from Theorem 5.1.7 by using functions that are *linear* in the empirical risk, *i.e.*, functions of the form $\mathcal{D}(q,p) = \mathcal{F}(p) - C\cdot q$ for convex $\mathcal{F}$. This choice for $\mathcal{D}$ gives a PAC-Bayes bound whose minimum is obtained for Gibbs classifiers G_Q minimizing a simple linear combination of $R_S(G_Q)$ and $\mathrm{KL}(Q\|P)$.

Let $\mathcal{D}(q,p) = \mathcal{F}(p) - C\cdot q$ for some function $\mathcal{F}$ to be defined. Then

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

The third inequality obtained by using Equation (5.5). The result follows easily from Theorem 5.1.7 when $\mathcal{F}$ is the convex function $\mathcal{F}(p)=\ln\frac{1}{(1-p\,[1-e^{-C}])}$.

5.1.2 PAC-Bayes Bounds without Kullback-Leibler (KL) Term:

In the standard PAC-Bayes approach the divergence between prior and posterior forms part of the bound. This way, analysis of the bound is constrained by the choice of prior distribution. The choice of prior is likely to be generic and may not be suitable for the particular problem and results in a large Kullback-Leibler divergence term in the PAC-Bayes analysis. Therefore, removing the Kullback-Leibler (KL) term from the bound significantly reduces the complexity penalty. Moreover, removing the Kullback-Leibler (KL) term gives us a guarantee by which the empirical estimates of the Gibbs risk uniformly converges to the true risk (for all Q). This statement cannot be generally derived from the previous PAC-Bayes theorems since $\mathrm{KL}(Q\|P)$ can be arbitrary large.

Catoni was the first one who investigated a PAC-bayes bound with no KL term. In [6], Catoni achieved such a bound simply by bounding the KL-divergence by some inequalities and developed PAC-Bayes bounds that do not rely on the KL-divergence. In [[19]], we propose a new approach that produce PAC-Bayes risk bounds in which the KL-term simply disappears from the bounds by restricting ourselves to *aligned* posteriors. Before presenting these bounds we present some definitions and results relating to these categories of distributions.

Definition 5.1.8. *(aligned distribution) Let $\mathcal{H} = \{h_1, h_2, \cdots, h_{2n}\}$ be a set of binary classifiers such that it is* auto-complemented, *meaning that there exists a bijection $c : \mathcal{H} \to \mathcal{H}$ such that $c(h) = -h$ for any $h \in \mathcal{H}$. Moreover, a distribution Q on $\mathcal{H}$ will be called* aligned *on a prior P if for any $h \in \mathcal{H}$, we have*

$$Q(h) + Q(c(h)) \quad = \quad P(h) + P(c(h))\,,$$

where P is the uniform distribution on $\mathcal{H}$.

The following Lemma helps us obtain a version of PAC-Bayes Theorem 5.1.7 that has no KL divergence term.

Lemma 5.1.9. *Let $\mathcal{H} = \{h_1, h_2, \cdots, h_{2n}\}$ be a set of binary classifiers such that it is* auto-complemented, *Q be a distribution which is aligned on $\mathcal{H}$ and $\mathcal{D}$ be a function such that $\mathcal{D}(p, q) = \mathcal{D}(1-p, 1-q)$ then we have:*

$$\mathop{\mathbf{E}}_{h\sim P} e^{m\cdot\mathcal{D}(R_S(h),R_D(h))} = \mathop{\mathbf{E}}_{h\sim Q} e^{m\cdot\mathcal{D}(R_S(h),R_D(h))}$$

Proof.

$$\mathop{\mathbf{E}}_{h\sim P} e^{m\cdot\mathcal{D}(R_S(h),R_D(h))} = \int_{h\in\mathcal{H}} P(h)e^{m\cdot\mathcal{D}(R_S(h),R_D(h))} = \int_{h\in\mathcal{H}} P(c(h))e^{m\mathcal{D}(R_S(c(h)),R_D(c(h))} .$$

$$\begin{aligned}
& 2\mathop{\mathbf{E}}_{h\sim P} e^{m\cdot\mathcal{D}(R_S(h),R_D(h))} \\
&= \int_{h\in\mathcal{H}} P(h)e^{m\cdot\mathcal{D}(R_S(h),R_D(h))} + \int_{h\in\mathcal{H}} P(c(h))e^{m\cdot\mathcal{D}(R_S(c(h)),R_D(c(h)))} \\
&= \int_{h\in\mathcal{H}} P(h)e^{m\cdot\mathcal{D}(R_S(h),R_D(h))} + \int_{h\in\mathcal{H}} P(c(h))e^{m\cdot\mathcal{D}(1-R_S(h),1-R_D(h))} \\
&= \int_{h\in\mathcal{H}} \left(P(h)+P(c(h))\right) e^{m\cdot\mathcal{D}(R_S(h),R_D(h))} \\
&= \int_{h\in\mathcal{H}} \left(Q(h)+Q(c(h))\right) e^{m\cdot\mathcal{D}(R_S(h),R_D(h))} \\
&= \int_{h\in\mathcal{H}} Q(h)e^{m\cdot\mathcal{D}(R_S(h),R_D(h))} + \int_{h\in\mathcal{H}} Q(c(h))e^{m\cdot\mathcal{D}(R_S(c(h)),R_D(c(h)))} \\
&= 2\mathop{\mathbf{E}}_{h\sim Q} e^{m\cdot\mathcal{D}(R_S(h),R_D(h))} .
\end{aligned}$$

□

Now, let us show how Lemma 5.1.9 leads us to obtain PAC-Bayes bounds with no KL term. In the proof of PAC-Bayes Theorem 5.1.7, we had to bound random variable $\mathop{\mathbf{E}}_{h\sim P} e^{m\cdot\mathcal{D}(R_S(h)-R(h))}$. Basically, the term $\mathrm{KL}(Q\|P)$ arises when we are transforming the expectation over P into expectation over Q. In the case where the posterior Q is not aligned we have:

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

Recall from the proof of Theorem 5.1.7 that the above results are obtained from two applications of Jensen's inequality(Lemma 5.1.6): one exploiting the concavity of $\log(x)$ and the second the convexity of $\mathcal{D}$.

However, when the posterior Q is aligned, because of Lemma 5.1.9, we can instead follow this modified argument:

$$\begin{aligned}
& \log\left[\mathop{\mathbf{E}}_{h\sim P} e^{m\cdot\mathcal{D}(R_S(h),R_D(h))}\right] \\
=\ & \log\left[\mathop{\mathbf{E}}_{h\sim Q} e^{m\cdot\mathcal{D}(R_S(h),R_D(h))}\right] \\
\geq\ & \mathop{\mathbf{E}}_{h\sim Q} \log\left[e^{m\cdot\mathcal{D}(R_S(h),R_D(h))}\right] \\
=\ & m \mathop{\mathbf{E}}_{h\sim Q} \mathcal{D}\left(R_S(h),R_D(h)\right) \\
\geq\ & m\cdot\mathcal{D}\left(\mathop{\mathbf{E}}_{h\sim Q} R_S(h), \mathop{\mathbf{E}}_{h\sim Q} R_D(h)\right) \\
=\ & m\cdot\mathcal{D}\left(R_S(G_Q),R_D(G_Q)\right)\ . \qquad (5.6)
\end{aligned}$$

In the following, we present a general theorem similar to Theorem 5.1.7 for the cases where the posterior Q is aligned.

Theorem 5.1.10. (General PAC-Bayes bound with no KL term) *Given any space $\mathcal{H}$ of classifiers, for any data-independent prior distribution P over $\mathcal{H}$ and for any aligned posterior distribution Q over $\mathcal{H}$ and for all convex function $\mathcal{D}:\ [0,1]\times[0,1]\to R$, we have:*

$$\Pr_{S\sim D^m}\left(\forall\, Q \textit{ aligned on } P\colon\ \mathcal{D}(R_S(G_Q),R_D(G_Q)) \leq \frac{\log(\frac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{h\sim P} e^{m\mathcal{D}(R_S(h),R_D(h))})}{m}\right) \geq 1{-}\delta\,.$$

Proof. Since $e^{m\mathcal{D}(R_S(h),R(h))}$ is a non negative random variable by Markov's inequality we have:

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

From Equation (5.6) we have:

$$\Pr_{S\sim D^m}\left(\forall\, Q \text{ aligned on } P\colon\ m\mathcal{D}(R_S(h),R_D(h)) \leq \log(\frac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{h\sim P} e^{m\mathcal{D}(R_S(h),R_D(h))})\right) \geq 1{-}\delta$$

□

Using particular convex function $\mathcal{D}: \ [0,1]\times[0,1] \to R$ and by upper bounding $\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{h\sim P} e^{m\mathcal{D}(R_S(h),R_D(h))})$, we, therefore, can obtain risk bounds with no KL term.

By using $\mathcal{D}(q,p)=2(q-p)^2$ and the upper bound of $\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{h\sim P} e^{m\mathcal{D}(R_S(h),R_D(h))})$ (see the proof of Theorem 5.1.2) in the Theorem 5.1.10 we have the following corollary:

Corollary 5.1.11. (McAllester PAC-Bayes bound with no KL term) *Given any space $\mathcal{H}$ of classifiers, for any data-independent prior distribution P over $\mathcal{H}$ and for any aligned posterior distribution Q over $\mathcal{H}$ we have:*

$$\Pr_{S\sim D^m}\left(\forall Q \textit{ aligned on } P\colon\ \ R_D(G_Q) \ \ \leq \ \ R_S(G_Q) \ + \ \sqrt{\frac{\log(\frac{\xi(m)}{\delta})}{2m}}\right) \ \ \geq \ \ 1 \ - \ \delta\,.$$

Similarly, by using $\mathcal{D}(q,p)=\mathrm{kl}(q,p)$ and the upper bound of $\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{h\sim P} e^{m\mathcal{D}(R_S(h),R_D(h))})$ (see the proof of Theorem 5.1.2) in the Theorem 5.1.10 we have the following corollary:

Corollary 5.1.12. (Seeger PAC-Bayes bound with no KL term) *Given any space $\mathcal{H}$ of classifiers, for any data-independent prior distribution P over $\mathcal{H}$ and for any aligned posterior distribution Q over $\mathcal{H}$, we have:*

$$\Pr_{S\sim D^m}\left(\forall Q \textit{ aligned on } P\colon\ \ \mathrm{kl}(R_D(G_Q), R_S(G_Q)) \ \ \leq \ \ \frac{\log(\frac{1}{\delta}\xi(m))}{m}\right) \ \ \geq \ \ 1 \ - \ \delta\,.$$

5.2 Sample-Compression PAC-Bayes Theorem

As seen in Section 5.1, in PAC-Bayes theory, risk bounds are obtained by comparing a *posterior* distribution Q on $\mathcal{H}$ (the set of all classifiers) to a *prior* P defined before observing the training sequence S. In the sample-compression setting, this seems problematic since sc-classifiers are defined upon S.

Recall that in the sample compression setting, each sc-classifiers $h^{\mu}_{S_\mathbf{i}}$ is described by a compression sequence $S_\mathbf{i}$ and a message μ. given S, the compression sequence $S_\mathbf{i}$ is defined by the following vector $\mathbf{i}$ of indices:

$$\mathbf{i} \quad \stackrel{\text{def}}{=} \quad \langle i_1, i_2, \ldots, i_{|\mathbf{i}|}\rangle\,,$$

with $1 \leq i_1 < i_2 < \ldots < i_{|\mathbf{i}|} \leq m$. The number of indices present in $\mathbf{i}$ is denoted by $|\mathbf{i}|$. The set of all 2^m possible vectors of indices is denoted by $\mathcal{I}$. Given an arbitrary compression sequence $S_{\mathbf{i}}$, a message μ is chosen from the set $\mathcal{M}_{S_{\mathbf{i}}}$ of all messages that can be supplied with the compression sequence $S_{\mathbf{i}}$.

[33, 32] have extended the PAC-Bayes theorem to the sample-compression setting. Their proposed risk bound depends on a data-independent *prior distribution* P which is defined as a couple $(P_{\mathcal{I}},\, P_{S'})$, where $P_{\mathcal{I}}$ is a distribution on $\mathcal{I}$, and, for every possible compression sequence S', $P_{S'}$ is a distribution on $\mathcal{M}_{S'}$. For more details, see [33]. Given a training sequence S, P^S denotes the distribution on $\mathcal{H}^S$ associated with the prior P, i.e.,

$$P^S(h^{\mu}_{S_{\mathbf{i}}}) \quad = \quad P_{\mathcal{I}}(\mathbf{i})\, P_{_{S_{\mathbf{i}}}}(\mu)\,. \tag{5.7}$$

More precisely, we will only consider priors P^S on $\mathcal{I} \times \mathcal{M}_{S_{\mathbf{i}}}$ that can be written in the form of Equation (5.7). Note that $P_{\mathcal{I}}(\mathbf{i})$ does not depend on S at all and $P_{_{S_{\mathbf{i}}}}(\mu)$ can only depend on S through $\mathcal{M}_{S_{\mathbf{i}}}$. This implies that $P_I(\mathbf{i})$ must be defined before observing S and $P_{_{S_{\mathbf{i}}}}(\mu)$ defined for all possible values of $S_{\mathbf{i}} \subseteq \mathcal{X} \times \mathcal{Y}$ of size at most m. Since we do not allow any dependencies on S for $P_I(\mathbf{i})$, we can hardly consider any difference a priori between two vectors of indices $\mathbf{i}, \acute{\mathbf{i}} \subset \mathcal{I}$ that have the same size. Hence, we adopt the convention that a same prior probability is assigned to every vector $\mathbf{i}$ having the same size, that is we have:

$$P_I(\mathbf{i}) = p(|\mathbf{i}|) \cdot \binom{m}{|\mathbf{i}|}^{-1}$$

for any $p(\cdot)$ such that $\sum_{d=0}^{m} p(d) = 1$.

Given a training sequence S, we denote by $Q_{\mathcal{I}}(\mathbf{i})$, the probability that a compression sequence $S_{\mathbf{i}}$ is chosen by Q, and by $Q_{_{S_{\mathbf{i}}}}(\mu)$, the probability distribution of choosing μ given $S_{\mathbf{i}}$. More precisely,

$$Q_{\mathcal{I}}(\mathbf{i}) \;\stackrel{\text{def}}{=}\; \int_{\mu \in \mathcal{M}(S_{\mathbf{i}})} Q(h^{\mu}_{S_{\mathbf{i}}})\, d\mu \quad \text{and} \quad Q_{_{S_{\mathbf{i}}}}(\mu) \;\stackrel{\text{def}}{=}\; Q(h^{\mu}_{S_{\mathbf{i}}} \mid S_{\mathbf{i}})\,. \tag{5.8}$$

Under this convention, the posterior Q can be written similar to P^S as:

$$Q(h^{\mu}_{S_{\mathbf{i}}}) = Q_{\mathcal{I}}(\mathbf{i}) Q_{_{S_{\mathbf{i}}}}(\mu) \tag{5.9}$$

Both $Q_{\mathcal{I}}(\mathbf{i})$ and $Q_{_{S_{\mathbf{i}}}}(\mu)$ can be dependent on S and can be chosen after observing the training data S.

In the data-independent PAC-Bayes setting, a bound on $R_D(B_Q)$ is indirectly obtained by bounding the risk of an associated stochastic classifier known as the Gibbs classifier G_Q. To assign an output label to an input example $\mathbf{x}$, the Gibbs classifier G_Q randomly chooses a classifier h according to Q and uses h to assign the label $h(\mathbf{x})$ to $\mathbf{x}$. In the sample compressed (data-dependent) PAC-Bayes setting, given a training sequence S, G_Q randomly chooses $\mathbf{i}$ according to $Q_{\mathcal{I}}$, then chooses a message μ according to $Q_{S_\mathbf{i}}$, and then classifies $\mathbf{x}$ according to $h^{\mu}_{S_\mathbf{i}}(\mathbf{x})$. Given a distribution D and a training sequence S generated by D, the true risk $R_D(G_Q)$ is given by:

$$R_D(G_Q) \;=\; \mathop{\mathbf{E}}_{h^{\mu}_{S_\mathbf{i}}\sim Q}\;\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D}\; I(h^{\mu}_{S_\mathbf{i}}(\mathbf{x}) \neq y)$$

As in [33] and [32] the empirical risk of a sc-classifier is usually computed on examples of the training set S that are not in the compression set $S_\mathbf{i}$ ($S \setminus S_\mathbf{i}$). However, based on our discussion in Section 4.2 of Chapter 4, depending on the learning algorithm the empirical estimate $R_S(G_Q)$ on S can be given in one of the two following ways:

$$(1) : R_S(G_Q) \;=\; \mathop{\mathbf{E}}_{h^{\mu}_{S_\mathbf{i}}\sim Q} R_S(h^{\mu}_{S_\mathbf{i}}) = \mathop{\mathbf{E}}_{h^{\mu}_{S_\mathbf{i}}\sim Q}\;\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim S\setminus S_\mathbf{i}}\; I(h^{\mu}_{S_\mathbf{i}}(\mathbf{x}) \neq y)$$

or

$$(2) : R_S(G_Q) \;=\; \mathop{\mathbf{E}}_{h^{\mu}_{S_\mathbf{i}}\sim Q} R_D(h^{\mu}_{S_\mathbf{i}}) = \mathop{\mathbf{E}}_{h^{\mu}_{S_\mathbf{i}}\sim Q}\;\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim S}\; I(h^{\mu}_{S_\mathbf{i}}(\mathbf{x}) \neq y)$$

With these classes of posteriors Q and priors P, we present the PAC-Bayes Theorem 5.2.1 from [32] in which Q is restricted to have a non zero weight only on classifiers having a compression set size $|\mathbf{i}| \leq l$ for some $l \in \{0, \cdots, m\}$. Note that other versions of the PAC-Bayes theorems for sample compression setting exist(see [32, 33]). However, Theorem 5.2.1 is more of our interest since the new PAC-Bayes theorems proposed in this thesis are inspired by this theorem.

Theorem 5.2.1. *[32] Given all our previous definitions, for any prior P and for any $\delta \in (0, 1]$*

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\forall\, Q \textit{ on } \mathcal{I}\times\mathcal{M}_{S_\mathbf{i}} \quad \textit{such that} \quad Q_{s_\mathbf{i}}(\mu) = 0 \textit{ if } |\mathbf{i}| > l \;: \\ kl(R_S(G_Q)\|R_D(G_Q) \leq \frac{1}{m-l}\,\left[KL(Q\,\|P^S) + \ln\frac{m+1}{\delta}\right]\end{array}\right) \geq 1-\delta$$

Inspired by the work of [33], who have generalized the PAC-Bayes approach to the sample compression setting in which SCM is defined and the success of kernel methods such as SVM (that can also viewed as sc-classifiers) as state-of the-art machine learning algorithms, we decided to propose a PAC-Bayes sample compression approach to kernel methods.

In the next chapter, we first present our motivations that lead us in this direction in more details and then we present our new sample compression PAC-Bayes bounds.

Chapter 6

New Sample Compressed PAC-Bayes Theorems

In this chapter, we propose our new PAC-Bayes theorems for the sample compression setting. The next section presents the motivation that leads us to derive these bounds.

6.1 Motivation

Research works in [[31, 19]] showed that the PAC-Bayesian theory is a good starting point for designing learning algorithms. PAC-Bayes bounds provide an upper bound on the risk of stochastic classifiers (Gibbs classifiers) $R_D(G_Q)$. As we showed earlier (see Equation (5.3)), an upper-bound on $R_D(G_Q)$ also provides an upper bound on the risk of the majority vote classifier $R_D(B_Q)$. While Gibbs classifiers are rare in practice, the majority vote classifiers such as the SVM are more common. SVM is a state-of-the-art learning algorithm that belongs to kernel methods (see Section 2.3). We will see later in Chapter 7 that the SVM classifier is actually a particular case of a (weighted) majority vote of sample-compressed classifiers where the compression sequence of each classifier consists of at most a single training example. Inspired by [[31]], who specialized the risk bound of [33] to SCM and proposed a learning strategy for SCM based on the minimization of the mentioned bound, and also by the success of kernel methods such as SVM, we propose PAC-Bayes risk bounds for majority votes of sample-compressed classifiers.

6.2 Preliminaries

In this section, we present some background information that is necessary to derive our new sample compressed PAC-Bayes bounds.

Recall that the bound given by the PAC-Bayes theorem for the risk of Gibbs classifiers can be turned into a bound for the risk of Bayes classifiers using Equation (5.3). This way, Theorem 5.2.1 in Chapter 5 provides an upper bound for the true risk of majority vote B_Q. ($R_D(B_Q) \leq 2R_D(G_Q)$). Hence, an upper-bound on $R(G_Q)$ also provides an upper bound on $R(B_Q)$.

However, we focus in this thesis on majority votes of sc-classifiers having a small compression size. In this setting, $\mathcal{H}^S$ consists mostly of weak classifiers having large risk $R(h)$. Then, $R_D(G_Q)$ is (almost) always large (near 1/2) for any Q even if the majority vote B_Q has very low risk. Thus, the disparity between $R_D(B_Q)$ and $R_D(G_Q)$ is enormous. Consequently, trying to minimize an upper-bound on $R_D(G_Q)$ should not lead to a majority vote B_Q having low risk.

One way to obtain a more relevant bound on $R_D(B_Q)$ from the PAC-Bayes theory is to use a loss function for stochastic classifiers which is distinct from the zero-one loss used for the deterministic classifiers. In order to obtain a tractable optimization problem, we propose to use a convex loss function of the margin of the Q-convex combination of sc-classifiers where the margin on example $(\mathbf{x}, y)$ of the Q-convex combination is given by

$$M_Q(\mathbf{x}, y) \overset{\text{def}}{=} \mathbf{E}_{h^{\mu}_{S_{\mathbf{i}}} \sim Q}\; y h^{\mu}_{S_{\mathbf{i}}}(\mathbf{x}) \,. \tag{6.1}$$

Note that $R_D(G_Q) = \frac{1}{2} - \frac{1}{2}\mathbf{E}_{(\mathbf{x},y)\sim D} M_Q(\mathbf{x}, y)$ gives a relation between $R_D(G_Q)$ and $M_Q(\mathbf{x}, y)$.

As in [[19]], we restrict ourselves to losses that upper-bound the zero-one loss of B_Q. More precisely, we consider functions $\zeta : [-1, 1] \to \mathbb{R}$ of the form

$$\zeta(\alpha) \overset{\text{def}}{=} \sum_{k=0}^{\deg(\zeta)} a_k\, \alpha^k \quad \text{with } a_k \geq 0 \text{ and such that } \zeta(\alpha) \geq I(-\alpha \leq 0)\,,$$

and we will then look for PAC-Bayes bounds of the following expected loss

$$\zeta^Q_D \overset{\text{def}}{=} \mathbf{E}_{(\mathbf{x},y)\sim D} \zeta(-M_Q(\mathbf{x}, y)) = \underset{(\mathbf{x},y)\sim D}{\mathbf{E}} \sum_{k=0}^{\deg(\zeta)} a_k\, \left(-M_Q(\mathbf{x}, y)\right)^k \,, \tag{6.2}$$

based on its empirical (possibly biased) estimate we have:

$$\zeta_S^Q \overset{\text{def}}{=} \underset{(\mathbf{x},y)\sim S}{\mathbf{E}} \sum_{k=0}^{\deg(\zeta)} a_k \, \left(-M_Q(\mathbf{x},y)\right)^k . \tag{6.3}$$

Such a ζ is called a *convex margin loss function* (also called a *convex surrogate loss*).

Since $\zeta(\alpha) \geq I(-\alpha \leq 0)$ we have

$$\zeta_D^Q \quad \geq \quad \underset{(\mathbf{x},y)\sim D}{\mathbf{E}} I(M_Q(\mathbf{x},y) \leq 0\,) \quad \geq \quad R_D(B_Q)\,. \tag{6.4}$$

Thus, the convex loss function ζ_D^Q is always an upper bound of the true risk of the Bayes classifier $R_D(B_Q)$. In particular, the factor-of-two rule $R_D(B_Q) \leq 2R_D(G_Q)$ simply corresponds to the case where $a_0 = a_1 = 1$, and $a_j = 0$ for all $j > 1$, since for these values, $\zeta_D^Q = 1 - M_Q(\mathbf{x},y) = 2R_D(G_Q)$ (see Figure 6.1).

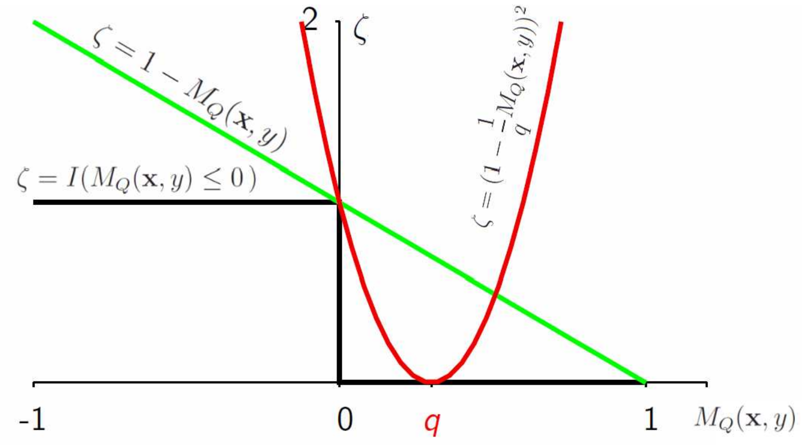

Figure 6.1: Three different loss functions. The curves of $1-M_Q(\mathbf{x},y)$ and $I(M_Q(\mathbf{x},y) \leq 0\,)$ illustrate: $R_D(B_Q) \leq 2R_D(G_Q)$.

We obtain a bound on ζ_D^Q by linking the risk of this classifier with the risk of a particular Gibbs classifier that we denote as $G_{\overline{Q}}$ which is defined on the space $\overline{\mathcal{H}^S}$ of classifiers where

$$\overline{\mathcal{H}^S} \overset{\text{def}}{=} \{\overline{h_1..h_k} \mid k \in \{0,\ldots,d\}, h_1,\ldots,h_k \in \mathcal{H}^S\}.$$

$\overline{h_1..h_k} = \bar{h}$ represents an "abstract" sc-classifier for each $k \in \{0,..,d\}$ and any k-tupple $(h_1,..,h_k)$ where $d = \deg(\zeta)$ and the size of the compression sequence of each

h_i ($\forall i \in \{0, \cdots, d\}$) is at most l. For each S, we define $\overline{\mathcal{H}^S}$ as the set of all such sc-classifiers, and for each distribution Q on $\mathcal{H}^S$, we denote by $\overline{Q}$ the following distribution on $\overline{\mathcal{H}^S}$:

$$\overline{Q}(\,\overline{h_1..h_k}\,) \overset{\text{def}}{=} \frac{a_k}{\zeta(1)}\, Q(h_1) \cdot \,\ldots\, \cdot Q(h_k)\,. \tag{6.5}$$

Note that since $\zeta(1) = \sum_{k=0}^{d} a_k$, $\overline{Q}$ is a probability distribution.

The true and empirical risks of $\overline{h_1..h_k}$ are respectively defined as:

$$\overline{R}_D(\,\overline{h_1..h_k}\,) \overset{\text{def}}{=} \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} I(\neg \underline{\bigvee}_{i=1}^{k} (h_i(\mathbf{x}) \neq y)) \tag{6.6}$$

$$\overline{R}_S(\,\overline{h_1..h_k}\,) \overset{\text{def}}{=} \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim S} I(\neg \underline{\bigvee}_{i=1}^{k} (h_i(\mathbf{x}) \neq y)) \;\;=\;\; \frac{1}{m} \sum_{j=1}^{m} I(\neg \underline{\bigvee}_{i=1}^{k} (h_i(\mathbf{x}_j) \neq y_j))\,, \tag{6.7}$$

where $\veebar$ denotes the exclusive or. (Observe that since the compression sequence size of each h_i's is at most l, for any $\bar{h} = \overline{h_1..h_k}$, we have $|\mathbf{i}_{\bar{h}}| \leq l \cdot k$. Moreover, for the case where $k = 0$, we have $\overline{R}_D(\overline{h_1..h_0}) = 1$, because the exclusive or over an empty sequence outputs *false*. Finally, note that $G_{\bar{Q}}$ is the Gibbs classifier related to the distribution $\bar{Q}$. Its empirical and true risks are calculated as usual, that is:

$$\begin{aligned} R_S(G_{\bar{Q}}) &= \mathop{\mathbf{E}}_{\bar{h}\sim\bar{Q}} \overline{R}_D(\,\bar{h}\,), \\ R_D(G_Q) &= \mathop{\mathbf{E}}_{\bar{h}\sim\bar{Q}} \overline{R}_S(\,\bar{h}). \end{aligned}$$

As we can see in Equation (6.7), the empirical risk of a sc-classifier is not computed on $S \setminus S_{\mathbf{i}}$ (recall our discussion in Section 4.2 of Chapter 4). This way, the empirical risk is a biased estimate of the true risk (the bias comes from the elements of S that are in the compression sequence). There is, therefore, a bias for the Gibbs classifiers. We take the performance of each sc-classifier on its compression set into consideration when we want to find the "best" posterior Q. Because of this consideration and also for simplicity, we decide to compute the empirical risk of a sc-classifier on S and deal with this bias directly in the elaboration of the proposed risk bounds. Indeed, otherwise, the performance of sc-classifiers on the compression set would have been totally absent in the resulting risk bounds. This way, we have to introduce $\tilde{R}_S$, *the unbiased abstract* *empirical risk*. $\tilde{R}_S$ is computed on the examples of S that *are not* in the compression sequence of $\bar{h}$. More formally,

$$\widetilde{R}_S(\,\overline{h_1..h_k}\,) \overset{\text{def}}{=} \frac{1}{m-|\mathbf{i}_{\overline{h_1..h_k}}|}\sum_{j=1}^{m} I\left(\neg\underline{\bigvee}{}_{i=1}^{k}(h_i(\mathbf{x}_j)\!\neq\! y_j)\right) I\left((\mathbf{x}_j,y_j)\notin \mathbf{i}_{\overline{h_1..h_k}}\right) \,. \tag{6.8}$$

Hence, contrarily to $m\cdot\overline{R}_S(\bar{h})$ which may contain some bias, $(m-|\mathbf{i}_{\bar{h}}|)\cdot\widetilde{R}_S(\bar{h})$ is an arithmetic mean of truly iid random variables. On the other hand, the two values are very close. Indeed, since

$$0 \;\;\leq\;\; m\cdot\overline{R}_S(\bar{h})-(m-|\mathbf{i}_{\bar{h}}|)\cdot\widetilde{R}_S(\bar{h}) \;\;\leq\;\; |\mathbf{i}_{\bar{h}}|\,, \tag{6.9}$$

we have

$$-|\mathbf{i}_{\bar{h}}| \;\;\leq\;\; -|\mathbf{i}_{\bar{h}}|\cdot\widetilde{R}_S(\bar{h}) \;\;\leq\;\; m\cdot\overline{R}_S(\bar{h})-m\cdot\widetilde{R}_S(\bar{h}) \;\;\leq\;\; |\mathbf{i}_{\bar{h}}|-|\mathbf{i}_{\bar{h}}|\cdot\widetilde{R}_S(\bar{h}) \;\;\leq\;\; |\mathbf{i}_{\bar{h}}|\,.$$

Therefore,

$$\left|\overline{R}_S(\bar{h})-\widetilde{R}_S(\bar{h})\right| \;\;\leq\;\; \frac{|\mathbf{i}_{\bar{h}}|}{m}\,. \tag{6.10}$$

Now, from Equation (6.6) we have:

$$\overline{R}_D(\,\overline{h_1..h_k}\,) = \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D} I(\neg\underline{\bigvee}{}_{i=1}^{k}(h_i(\mathbf{x})\!\neq\! y)) = \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D}\frac{1}{2}\left[1+\prod_{i=1}^{k}-yh_i(\mathbf{x})\right] \tag{6.11}$$

Thus for $U\!=\!D$ and $U\!=\!S$, we have:

$$\begin{aligned}
R_U(G_{\overline{Q}}) &= \mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}}\overline{R}_U(\bar{h})\\
&= \sum_{k=0}^{\deg(\zeta)}\frac{a_k}{\zeta(1)}\mathop{\mathbf{E}}_{h_1\sim Q}\cdots\mathop{\mathbf{E}}_{h_k\sim Q}\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D}\frac{1}{2}\left[1+\prod_{i=1}^{k}-yh_i(\mathbf{x})\right]\\
&= \sum_{k=0}^{\deg(\zeta)}\frac{a_k}{\zeta(1)}\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D}\frac{1}{2}\left[1+\prod_{i=1}^{k}\mathop{\mathbf{E}}_{h_i\sim Q}-yh_i(\mathbf{x})\right]\\
&= \frac{1}{2}\left[1+\frac{1}{\zeta(1)}\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D}\sum_{k=0}^{\deg(\zeta)}a_k(\mathop{\mathbf{E}}_{h\sim Q}-yh(\mathbf{x}))^k\right]\\
&= \frac{1}{2}\left[1+\frac{1}{\zeta(1)}\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D}\sum_{k=0}^{\deg(\zeta)}a_k(-M_Q(\mathbf{x},y))^k\right]\\
&= \frac{1}{2}\left[1+\frac{1}{\zeta(1)}\,\zeta_U^Q\right]
\end{aligned} \tag{6.12}$$

The last equality is obtained from Equations 6.2 or 6.3. We have now made a direct link between the ζ_D^Q and the risk of a Gibbs classifier defined on $\overline{\mathcal{H}^S}$.

Recall that, here we want PAC-Bayes bounds on $R(G_{\overline{Q}})$ that translate into a bound for ζ_D^Q. As usual, the upper bounds on ζ_D^Q depend on the value of $\mathrm{KL}(\overline{Q}||\overline{P^S})$ where

$$\mathrm{KL}(\overline{Q}||\overline{P^S}) = \mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} \ln \frac{\overline{Q}(\bar{h})}{\overline{P^S}(\bar{h})}$$

denotes the Kullback-Leibler divergence between distributions $\overline{Q}$ and $\overline{P^S}$ defined on $\overline{\mathcal{H}^S}$. Thus, to obtain PAC-Bayes bounds on ζ_D^Q, we now calculate the value of $\mathrm{KL}(\overline{Q}||\overline{P^S})$. To simplify the calculation and restrict the size of $\mathrm{KL}(\overline{Q}||\overline{P^S})$ which is best to keep it small, it is preferable to choose a prior $\overline{P^S}$ having properties similar to those of $\overline{Q}$ (see Equation (6.5)). Therefore, for any S, the prior $\overline{P^S}$ is given by:

$$\overline{P^S}(\,\overline{h_1..h_k}\,) \overset{\mathrm{def}}{=} \frac{a_k}{\zeta(1)}\, P^S(h_1)\cdot\,\ldots\,\cdot P^S(h_k)\,. \tag{6.13}$$

In this case, we have:

$$\begin{aligned}
\mathrm{KL}(\overline{Q}||\overline{P^S}) &= \sum_{k=0}^{\deg(\zeta)} \frac{a_k}{\zeta(1)} \mathop{\mathbf{E}}_{h_1\sim Q} \cdots \mathop{\mathbf{E}}_{h_k\sim Q} \ln \frac{a_k \prod_{i=1}^k Q(h_i)}{a_k \prod_{i=1}^k P(h_i)} \\
&= \sum_{k=0}^{\deg(\zeta)} \frac{a_k}{\zeta(1)} \mathop{\mathbf{E}}_{h_1\sim Q} \cdots \mathop{\mathbf{E}}_{h_k\sim Q} \sum_{i=1}^{k} \ln \frac{Q(h_i)}{P(h_i)} \\
&= \sum_{k=0}^{\deg(\zeta)} \frac{a_k}{\zeta(1)} k \cdot \mathop{\mathbf{E}}_{h\sim Q} \ln \frac{Q(h)}{P(h)} \\
&= \frac{\zeta'(1)}{\zeta(1)} \cdot KL(Q\|P^S)
\end{aligned} \tag{6.14}$$

where $\zeta'(1)=\sum_{k=1}^{d} k\cdot a_k$.

The theorems presented in this chapter give a bound on ζ_D^Q and, consequently, on $R_D(B_Q)$ (see Equation (6.4)). These bounds can be categorized into two groups. In the first group named *sample compressed Pac-Bayes bounds with KL*, the PAC-Bayes risk bounds depend on the KL divergence between the prior and the posterior over the set of sample-compressed classifiers. In the second group named *sample compressed Pac-Bayes bounds without KL*, the PAC-Bayes risk bound has the unusual property of having no KL divergence term when the posterior is *aligned* with the prior in some precise way defined later.

Before stating our PAC-Bayes sample compressed theorems, we present Lemma 6.2.1(see Maurer [41]) for the proof) which we often refer to in the demonstrations of some theorems in this chapter.

Lemma 6.2.1. *(Maurer [41]) Let $n \geq 8$, and suppose that $X = (X_1, \ldots, X_n)$ is a vector of iid random variables, $0 \leq X_i \leq 1$, $\mathbf{E}[X_i] = \nu$ and let $M(X) = \frac{1}{n}\sum_{j=1}^{n} X_i$ be the arithmetic mean of the random variables. Then*

$$\sqrt{n} \quad \leq \quad \mathbf{E}\, e^{n\,\mathrm{kl}(M(X)\|\nu)} \quad \leq \quad 2\sqrt{n}\,.$$

In the following section, we present the first group of PAC-Bayes theorems: PAC-Bayes Bounds with KL.

6.3 PAC-Bayes Bounds with KL Term

In this Section, we present the PAC-Bayes risk bounds which, as usual, depend on the KL divergence between the prior and the posterior of sample-compressed classifiers. These bounds are valid for any margin loss ζ and for any sample-compressed classifier.

Catoni bound with KL

The next theorem gives a bound on ζ_D^Q and, consequently, on $R_D(B_Q)$. It can be viewed as a generalization of Theorem 1.2.1 of [7].

Theorem 6.3.1. *For any D, any family $(\mathcal{H}^S)_{S\in D^m}$ of sets of sc-classifiers of size at most l, any prior $\mathcal{P}$, any $\delta \in (0,1]$, any positive real number C_1, and any margin loss function ζ such that $l \cdot \deg(\zeta) < m$, we have*

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\forall\, Q \ \ on\ \ \mathcal{H}^S\text{:}\\ \zeta_D^Q \ \leq \ \zeta(1)[C'-1] + C'\cdot\left(\zeta_S^Q + \frac{2}{m\cdot C_1}[\zeta'(1)\cdot \mathrm{KL}(Q\|P^S) + \zeta(1)\cdot \ln\frac{1}{\delta}]\right)\end{array}\right) \geq 1-\delta$$

where $\mathrm{KL}(\cdot\|\cdot)$ *is the Kullback-Leibler divergence, and where* $C' = \frac{C_1\cdot\frac{m}{m-l\cdot\deg\zeta}}{1-e^{-C_1\cdot\frac{m-l\cdot\deg\zeta}{m}}}$.

Proof. Let S be any training sequence, $d \overset{\mathrm{def}}{=} \deg\zeta$. Similarly as in the Section 6.2, we define $\bar{h} = \overline{h_1..h_k}$ (with $k\in\{0,..,d\}$), $\overline{R}_D(\bar{h})$, $\overline{R}_S(\bar{h})$, $\overline{\mathcal{H}^S}$, $\overline{P^S}$, $\overline{Q}$, ζ_D^Q, and ζ_S^Q. Let $\mathcal{F}$ be a convex function to be defined later, and $\mathcal{D}(q,p) \overset{\mathrm{def}}{=} \mathcal{F}(p) - C_1\cdot q$.

Now, let us consider the following random variable:

$$X_{\overline{P^S}} \overset{\text{def}}{=} \mathbf{E}_{\bar{h}\sim\overline{P^S}}\, e^{(m-|\mathbf{i}_{\bar{h}}|)\mathcal{D}(\overline{R}_S(\bar{h}),\overline{R}_D(\bar{h}))}\,.$$

Since $\mathbf{E}_{\bar{h}\sim\overline{P^S}}\, e^{(m-|\mathbf{i}_{\bar{h}}|)\mathcal{D}(\overline{R}_S(\bar{h}),\overline{R}_D(\bar{h}))}$ is a non negative random variable it then follows from Markov's inequality (Lemma 5.1.5) that

$$\Pr_{S\sim D^m}\left(X_{\overline{P^S}} \le \frac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m} X_{\overline{P^S}}\right) \ge 1-\delta\,.$$

By taking the logarithm on each side of the innermost inequality and by transforming the expectation over $\overline{P^S}$ into an expectation over $\overline{Q}$ using Equation (5.4), we obtain

$$\Pr_{S\sim D^m}\!\left(\forall Q\colon \ln\!\left[\mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} \frac{\overline{P^S}(\bar{h})}{\overline{Q}(\bar{h})} e^{(m-|\mathbf{i}_{\bar{h}}|)\mathcal{D}(\overline{R}_S(\bar{h}),\overline{R}_D(\bar{h}))}\right] \le \ln\!\left[\frac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{\bar{h}\sim\overline{P^S}} e^{(m-|\mathbf{i}_{\bar{h}}|)\mathcal{D}(\overline{R}_S(\bar{h}),\overline{R}_D(\bar{h}))}\right]\right) \ge 1-\delta\,. \quad (6.15)$$

Using Equation (6.14), together with Jensen's inequality(Lemma 5.1.6) applied to the concave $\ln(x)$ gives:

$$\ln\!\left[\mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} \frac{\overline{P^S}(\bar{h})}{\overline{Q}(\bar{h})}\, e^{(m-|\mathbf{i}_{\bar{h}}|)\mathcal{D}(\overline{R}_S(\bar{h}),\overline{R}_D(\bar{h}))}\right] \ge \mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} \ln\frac{\overline{P^S}(\bar{h})}{\overline{Q}(\bar{h})} + \mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} (m-|\mathbf{i}_{\bar{h}}|)\,\mathcal{D}(\overline{R}_S(\bar{h}),\overline{R}_D(\bar{h}))\,.$$

Now, it follows from Equation (6.12) and the definition of the Kullback-Leibler divergence that:

$$\mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} \ln\frac{\overline{P^S}(\bar{h})}{\overline{Q}(\bar{h})} = -\mathrm{KL}(\overline{Q}||\overline{P^S}) = -\frac{\zeta'(1)}{\zeta(1)}\cdot KL(Q\|P^S)\,.$$

So we have:

$$\ln\!\left[\mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} \frac{\overline{P^S}(\bar{h})}{\overline{Q}(\bar{h})}\, e^{(m-|\mathbf{i}_{\bar{h}}|)\mathcal{D}(\overline{R}_S(\bar{h}),\overline{R}_D(\bar{h}))}\right] \ge -\frac{\zeta'(1)}{\zeta(1)}\cdot \mathrm{KL}(Q\|P^S) + \mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} (m-|\mathbf{i}_{\bar{h}}|)\,\mathcal{D}(\overline{R}_S(\bar{h}),\overline{R}_D(\bar{h}))(6.16)$$

Again from the Jensen's inequality(Lemma 5.1.6), applied to the convex function $\mathcal{F}$ together with Equation (6.12) and the fact that $m - l \cdot d \leq (m - |\mathbf{i}_{\bar{h}}|) \leq m$, we obtain

$$\begin{aligned}\mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} (m - |\mathbf{i}_{\bar{h}}|)\, \mathcal{D}(\overline{R}_S(\bar{h}), \overline{R}_D(\bar{h})) \;\; &\geq \;\; (m - ld) \mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} \mathcal{F}(\overline{R}_D(\bar{h})) - m \mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} C_1 \cdot \overline{R}_S(\bar{h})) \\ &\geq (m - ld)\, \mathcal{F}\left(\frac{1}{2}\left[1 + \frac{1}{\zeta(1)}\, \zeta_D^Q\right]\right) - m\, C_1 \cdot \frac{1}{2}\left[1 + \frac{1}{\zeta(1)}\, \zeta_S^Q\right]\end{aligned} \tag{6.17}$$

Let us now analyze the value of $\mathbf{E}_{S\sim D^m} X_{\overline{P^S}}$, appearing in the right-hand side of the innermost inequality of Equation (6.32) . First, let us define $\mathbf{i}^c$ as the vector of indices of $\mathcal{I}$ that are not in the vector $\mathbf{i}$. Thus, $|\mathbf{i}^c| = m - |\mathbf{i}|$. Now, note that:

$$\mathop{\mathbf{E}}_{S\sim D^m} \mathop{\mathbf{E}}_{\bar{h}\sim\overline{P^S}} e^{(m-|\mathbf{i}_{\bar{h}}|)\, \mathcal{D}(\overline{R}_S(\bar{h}),\overline{R}_D(\bar{h}))} \;\; = \;\; \mathop{\mathbf{E}}_{\mathbf{i}\sim\overline{P}_{\mathcal{I}}} \mathop{\mathbf{E}}_{S_{\mathbf{i}}\sim D^{|\mathbf{i}|}} \mathop{\mathbf{E}}_{\bar{\mu}\sim\overline{P}_{S_{\mathbf{i}}}} \mathop{\mathbf{E}}_{S_{\mathbf{i}^c}\sim D^{m-|\mathbf{i}|}} e^{|\mathbf{i}^c|\, \mathcal{D}(\overline{R}_S(h^{\bar{\mu}}_{S_{\mathbf{i}}}),\overline{R}_D(h^{\bar{\mu}}_{S_{\mathbf{i}}}))}\,.$$

Recall that $\overline{P^S}$ denotes the distribution on $\overline{\mathcal{H}^S}$ and is given by $\overline{P^S}(\bar{h}) = \overline{P_{\mathcal{I}}}(\mathbf{i})\, \overline{P_{S_{\mathbf{i}}}}(\mu)$ (see Equation (5.7)).

Now, for each $h^{\bar{\mu}}_{S_{\mathbf{i}}} \in \overline{\mathcal{H}^S}$ define $a^{\bar{\mu}}_{S_{\mathbf{i}}} \overset{\text{def}}{=} \sum_{(x,y)\in S_{\mathbf{i}}} I(h^{\bar{\mu}}_{S_{\mathbf{i}}}(x) \neq y)$, and observe that $m \cdot \overline{R}_S(h^{\bar{\mu}}_{S_{\mathbf{i}}}) - a^{\bar{\mu}}_{S_{\mathbf{i}}}$ is then the number of errors made by $h^{\bar{\mu}}_{S_{\mathbf{i}}}$ on $S_{\mathbf{i}^c}$. Since the later is iid and disjoint from the compression sequence of $h^{\bar{\mu}}_{S_{\mathbf{i}}}$, we have that $m \cdot \overline{R}_S(h^{\bar{\mu}}_{S_{\mathbf{i}}}) - a^{\bar{\mu}}_{S_{\mathbf{i}}}$ is a random variable following a binomial law of parameters $(|\mathbf{i}^c|, \overline{R}_D(h^{\bar{\mu}}_{S_{\mathbf{i}}}))$. Since, $(m - ld) \cdot \frac{k}{m} \leq |\mathbf{i}^c| \cdot \frac{k+a^{\bar{\mu}}_{S_{\mathbf{i}}}}{m}$ for any $k \in \{0, .., |\mathbf{i}^c|\}$, we have:

$$\begin{aligned}&\mathbf{E}_{S_{\mathbf{i}^c}\sim D^{m-|\mathbf{i}|}}\, e^{|\mathbf{i}^c|\mathcal{D}(\overline{R}_S(h^{\bar{\mu}}_{S_{\mathbf{i}}}),\overline{R}_D(h^{\bar{\mu}}_{S_{\mathbf{i}}}))} \\ &\quad = \;\; \mathbf{E}_{S_{\mathbf{i}^c}\sim D^{m-|\mathbf{i}|}}\, e^{|\mathbf{i}^c|\mathcal{F}(\overline{R}_D(h^{\bar{\mu}}_{S_{\mathbf{i}}}))-C_1|\mathbf{i}^c|\overline{R}_S(h^{\bar{\mu}}_{S_{\mathbf{i}}})} \\ &\quad = \;\; e^{|\mathbf{i}^c|\mathcal{F}(\overline{R}_D(h^{\bar{\mu}}_{S_{\mathbf{i}}}))} \cdot \sum_{k=0}^{|\mathbf{i}^c|} \Pr_{S\sim D^m}\Big(m\cdot\overline{R}_S(h^{\bar{\mu}}_{S_{\mathbf{i}}}) - a^{\bar{\mu}}_{S_{\mathbf{i}}} = k\Big)\; e^{-C_1|\mathbf{i}^c|\cdot\frac{k+a^{\bar{\mu}}_{S_{\mathbf{i}}}}{m}} \\ &\quad \leq \;\; e^{|\mathbf{i}^c|\mathcal{F}(\overline{R}_D(h^{\bar{\mu}}_{S_{\mathbf{i}}}))} \cdot \sum_{k=0}^{|\mathbf{i}^c|} \Pr_{S\sim D^m}\Big(m\cdot\overline{R}_S(h^{\bar{\mu}}_{S_{\mathbf{i}}}) - a^{\bar{\mu}}_{S_{\mathbf{i}}} = k\Big)\; e^{-C_1\cdot(m-ld)\cdot\frac{k}{m}} \\ &\quad = \;\; e^{|\mathbf{i}^c|\mathcal{F}(\overline{R}_D(h^{\bar{\mu}}_{S_{\mathbf{i}}}))} \cdot \sum_{k=0}^{|\mathbf{i}^c|} \binom{|\mathbf{i}^c|}{k} \Big(\overline{R}_D(h^{\bar{\mu}}_{S_{\mathbf{i}}})\Big)^k \Big(1-\overline{R}_D(h^{\bar{\mu}}_{S_{\mathbf{i}}})\Big)^{|\mathbf{i}^c|-k}\; e^{-C_1\cdot\frac{m-ld}{m}\cdot k} \\ &\quad = \;\; e^{|\mathbf{i}^c|\mathcal{F}(\overline{R}_D(h^{\bar{\mu}}_{S_{\mathbf{i}}}))} \Big(1-\overline{R}_D(h^{\bar{\mu}}_{S_{\mathbf{i}}})\,[1-e^{-C_1\cdot\frac{m-ld}{m}}]\Big)^{|\mathbf{i}^c|}\,.\end{aligned} \tag{6.18}$$

The last equation being obtained from the Newton binomial: $\sum_{k=0}^{m} \binom{m}{k} x^k y^{m-k} = (x+y)^k$ with m replaced by $|\mathbf{i}^c|$, x by $\overline{R}_D(h^{\bar{\mu}}_{S_{\mathbf{i}}}))$ and y by $1 - \overline{R}_D(h^{\bar{\mu}}_{S_{\mathbf{i}}})$.

Let us now define $\mathcal{F}$ such that $1 = e^{|\mathbf{i}^c|\mathcal{F}(\overline{R}_D(h^{\bar{\mu}}_{S_\mathbf{i}}))}\left(1 - \overline{R}_D(h^{\bar{\mu}}_{S_\mathbf{i}})\,[1 - e^{-C_1 \cdot \frac{m-ld}{m}}]\right)^{|\mathbf{i}^c|}$, or equivalently, let

$$\mathcal{F}(\overline{R}_D(h^{\bar{\mu}}_{S_\mathbf{i}})))\overset{\text{def}}{=} -\ln\left(1 - \overline{R}_D(h^{\bar{\mu}}_{S_\mathbf{i}})\ \left[1 - e^{-C_1 \frac{m-ld}{m}}\right]\right) . \tag{6.19}$$

Note that $\mathcal{F}$ is convex since it is minus the logarithm of a linear function (recall that the logarithm is a concave function).

Therefore, with this choice and by replacing $\mathcal{F}$ in Equation (6.18) we get:

$$\mathop{\mathbf{E}}_{S\sim D^m}\ \mathop{\mathbf{E}}_{h^{\bar{\mu}}_{S_\mathbf{i}}\sim\overline{P^S}}\ e^{|\mathbf{i}^c|\mathcal{D}(\overline{R}_S(h^{\bar{\mu}}_{S_\mathbf{i}}),\overline{R}_D(h^{\bar{\mu}}_{S_\mathbf{i}}))} = 1\,.$$

In order to finish the proof, combine Equations (6.16), (6.17), (6.18) and (6.19) to rewrite the innermost inequality of Equation (6.15) as follows:

$$\begin{aligned}
&(m-ld)\cdot\mathcal{F}\left(\frac{1}{2}\left[1+\frac{1}{\zeta(1)}\ \zeta^Q_D\right]\right) - mC_1\cdot\frac{1}{2}\left[1+\frac{1}{\zeta(1)}\ \zeta^Q_S\right] - \frac{\zeta'(1)}{\zeta(1)}\cdot\mathrm{KL}(Q\|P^S) \le \ln\frac{1}{\delta}\\
&(m-ld)\left\{-\ln\left(1-\frac{1}{2}\left[1+\frac{1}{\zeta(1)}\ \zeta^Q_D\right]\left[1-e^{-C_1\frac{m-ld}{m}}\right]\right)\right\} \le mC_1\cdot\frac{1}{2}\left[1+\frac{1}{\zeta(1)}\ \zeta^Q_S\right] + \frac{\zeta'(1)}{\zeta(1)}\cdot\mathrm{KL}(Q\|P^S) + \ln\frac{1}{\delta}\\
&\frac{1}{2}\left[1+\frac{1}{\zeta(1)}\ \zeta^Q_D\right]\left[1-e^{-C_1\frac{m-ld}{m}}\right] \le 1 - \exp\left\{-\left(\frac{1}{m-ld}\right)\left(mC_1\cdot\frac{1}{2}\left[1+\frac{1}{\zeta(1)}\ \zeta^Q_S\right] + \frac{\zeta'(1)}{\zeta(1)}\cdot\mathrm{KL}(Q\|P^S) + \ln\frac{1}{\delta}\right)\right\}\\
&\frac{1}{2}\left[1+\frac{1}{\zeta(1)}\ \zeta^Q_D\right]\left[1-e^{-C_1\frac{m-ld}{m}}\right] \le \left(\frac{1}{m-ld}\right)\left(mC_1\cdot\frac{1}{2}\left[1+\frac{1}{\zeta(1)}\ \zeta^Q_S\right] + \frac{\zeta'(1)}{\zeta(1)}\cdot\mathrm{KL}(Q\|P^S) + \ln\frac{1}{\delta}\right)
\end{aligned}$$

The last transformation is an application of the inequality $1 - e^{-x} \le x$. We are now able to isolate ζ^Q_D:

$$\begin{aligned}
\zeta^Q_D &\le \left(\frac{2\cdot\zeta(1)}{1-e^{-C_1\frac{m-ld}{m}}}\right)\left(\frac{1}{m-ld}\right)\left(mC_1\cdot\frac{1}{2}\left[1+\frac{1}{\zeta(1)}\ \zeta^Q_S\right] + \frac{\zeta'(1)}{\zeta(1)}\cdot\mathrm{KL}(Q\|P^S) + \ln\frac{1}{\delta}\right) - \zeta(1)\\
&= \left(\frac{C_1\frac{m}{m-ld}}{1-e^{-C_1\frac{m-ld}{m}}}\right)\left(\zeta(1) + \zeta^Q_S + \frac{2}{mC_1}[\zeta'(1)\cdot\mathrm{KL}(Q\|P^S) + \zeta(1)\cdot\ln\frac{1}{\delta}]\right) - \zeta(1)\\
&= \zeta(1)[C'-1] + C'\cdot\left(\zeta^Q_S + \frac{2}{mC_1}[\zeta'(1)\cdot\mathrm{KL}(Q\|P^S) + \zeta(1)\cdot\ln\frac{1}{\delta}]\right)
\end{aligned}$$

where

$$C' = \frac{C_1\cdot\frac{m}{m-l\cdot\deg\zeta}}{1-e^{-C_1\cdot\frac{m-l\cdot\deg\zeta}{m}}}\,.$$

□

In the particular case of non-sample compressed classifiers (when $l = 0$) Theorem 6.3.1 reduces to the following corollary:

Corollary 6.3.2. *For any D, any $\mathcal{H}$ of sets of classifiers any prior $\mathcal{P}$, any $\delta \in (0,1]$, any positive real number C_1, and any margin loss function we have:*

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\forall\, Q \;\; on \;\; \mathcal{H}: \\ \zeta_D^Q \;\leq\; \zeta(1)[C'-1] + C' \cdot \left(\zeta_S^Q + \frac{2}{m\cdot C_1}[\zeta'(1)\cdot \mathrm{KL}(Q\|P^S) + \zeta(1)\cdot \ln\frac{1}{\delta}]\right)\end{array}\right) \geq 1-\delta$$

where $\mathrm{KL}(\cdot\|\cdot)$ *is the Kullback-Leibler divergence, and where* $C' = \frac{C_1}{1-e^{-C_1}}$.

McAllester bound with KL

Theorem 6.3.3. *For any D, for any $m \geq 8$, for any family $(\mathcal{H}^S)_{S\in D^m}$ of sets of sc-classifiers of size at most l, for any prior $\mathcal{P}$, for any margin loss function ζ such that $l \cdot \deg(\zeta) < m$, and for any $\delta \in (0,1]$, we have*

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\forall Q \in \mathcal{H}^S \quad : \\ \quad \zeta_D^Q \leq \zeta_S^Q + \frac{\zeta(1)}{\sqrt{\frac{1}{2}(m-l\deg(\zeta))}}\sqrt{\frac{\zeta'(1)}{\zeta(1)}\cdot \mathrm{KL}(Q\|P^S) + 4\,l\deg(\zeta) + \ln\frac{2\sqrt{m}}{\delta}}\end{array}\right) \geq 1-\delta$$

Proof. Let S be any training sequence, $d \overset{\mathrm{def}}{=} \deg\zeta$. Similarly as in the Section 6.2, we define $\bar{h} = \overline{h_1..h_k}$ (with $k\in\{0,..,d\}$), $\overline{R}_D(\bar{h})$, $\overline{R}_S(\bar{h})$, $\overline{\mathcal{H}^S}$, $\overline{P^S}$, $\overline{Q}$, ζ_D^Q, and ζ_S^Q.

We will consider the following random variable:

$$X_{\overline{P^S}} \overset{\mathrm{def}}{=} \mathop{\mathbf{E}}_{\bar{h}\sim\overline{P^S}} e^{m-|\mathbf{i}_{\bar{h}}|)2(\overline{R}_S(\bar{h})-\overline{R}_D(\bar{h}))^2} . \tag{6.20}$$

By using Markov's inequality (Lemma 5.1.5) we have

$$\Pr_{S\sim D^m}\left(X_{\overline{P^S}} \leq \frac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m} X_{\overline{P^S}}\right) \geq 1-\delta\,.$$

By taking the logarithm on each side of the innermost inequality and by transforming the expectation over $\overline{P^S}$ into an expectation over $\overline{Q}$ using Equation (5.4), we obtain

$$\Pr_{S\sim D^m}\left(\forall\, Q \;: \ln\left[\mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} \frac{\overline{P^S}(\bar{h})}{\overline{Q}(\bar{h})} e^{(m-|\mathbf{i}_{\bar{h}}|)2(\overline{R}_S(\bar{h})-\overline{R}_D(\bar{h}))^2}\right] \;\leq\; \ln\left[\frac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m} X_{\overline{P^S}}\right]\right) \;\geq\; 1-\delta\,. \tag{6.21}$$

Using Equation (6.14), together with Jensen's inequality(Lemma 5.1.6) applied to the concave $\ln(x)$ gives

$$\begin{aligned}\ln\left[\mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} \frac{\overline{P^S}(\bar{h})}{\overline{Q}(\bar{h})}\, e^{(m-|\mathbf{i}_{\bar{h}}|)2(\overline{R}_S(\bar{h})-\overline{R}_D(\bar{h}))^2}\right] \geq{}& -\frac{\zeta'(1)}{\zeta(1)}\cdot \mathrm{KL}(Q\|P^S) \\ &+ \mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} (m-|\mathbf{i}_{\bar{h}}|)2(\overline{R}_S(\bar{h})-\overline{R}_D(\bar{h}))^2 \end{aligned} \tag{6.22}$$

Again from the Jensen's inequality, applied to the convex function $\mathcal{D}(q,p)=(q-p)^2$, together with the definition of ζ_D^Q and ζ_S^Q (see Equation (6.12)) and the fact that $m-|\mathbf{i}_{\bar{h}}| \geq m-l\cdot d$, we obtain:

$$\mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} (m-|\mathbf{i}_{\bar{h}}|)2(\overline{R}_S(\bar{h})-\overline{R}_D(\bar{h}))^2 \;\; \geq \;\; (m-ld)2\left(\mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} \overline{R}_S(\bar{h}) - \mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} \overline{R}_D(\bar{h})\right)^2 \tag{6.23}$$

Thus, from Equations (6.21) to (6.23), we obtain:

$$\Pr_{S\sim D^m}\left(\forall\, Q : (m-ld)2\left(\mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} \overline{R}_S(\bar{h}) - \mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} \overline{R}_D(\bar{h})\right)^2 \leq \frac{\zeta'(1)}{\zeta(1)}\cdot \mathrm{KL}(Q\|P^S) + \ln\left[\frac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m} X_{\overline{P^S}}\right]\right) \geq 1-\delta\,. \tag{6.24}$$

Let us analyze $\mathbf{E}_{S\sim D^m} X_{\overline{P^S}}$. Let $\widetilde{R}_S(\bar{h})$ be the *abstract empirical risk* (Equation (6.8)).

Now, let us show that

$$\mathbf{E}_{S\sim D^m} X_{\overline{P^S}} \;\; \leq \;\; e^{4ld}\cdot 2\sqrt{m}\,. \tag{6.25}$$

First note that, since

$$\begin{aligned}\mathbf{E}_{S\sim D^m} X_{\overline{P^S}} \;\; &\overset{\text{def}}{=} \;\; \mathop{\mathbf{E}}_{S\sim D^m}\, \mathop{\mathbf{E}}_{\bar{h}\sim\overline{P^S}} e^{(m-|\mathbf{i}_{\bar{h}}|)2(\overline{R}_S(\bar{h})-\overline{R}_D(\bar{h}))^2} \\ &= \mathop{\mathbf{E}}_{\mathbf{i}\sim\overline{P}_{\mathcal{I}}}\, \mathop{\mathbf{E}}_{S_{\mathbf{i}}\sim D^{|\mathbf{i}|}}\, \mathop{\mathbf{E}}_{\bar{\mu}\sim\overline{P}_{S_{\mathbf{i}}}}\, \mathop{\mathbf{E}}_{S_{\mathbf{i}^c}\sim D^{m-|\mathbf{i}|}} e^{(m-|\mathbf{i}|)2\,(\overline{R}_S(h^{\bar{\mu}}_{S_{\mathbf{i}}})-\overline{R}_D(h^{\bar{\mu}}_{S_{\mathbf{i}}}))^2}\,,\end{aligned}$$

to prove Equation (6.25), it suffices to show that we have

$$\mathop{\mathbf{E}}_{S_{\mathbf{i}^c}\sim D^{m-|\mathbf{i}|}} e^{(m-|\mathbf{i}|)2\,(\overline{R}_S(h^{\bar{\mu}}_{S_{\mathbf{i}}})-\overline{R}_D(h^{\bar{\mu}}_{S_{\mathbf{i}}}))^2} \;\; \leq \;\; e^{4ld}\cdot 2\sqrt{m}\,. \tag{6.26}$$

for any $\mathbf{i}\in\mathcal{I}$, $S_{\mathbf{i}}\in(\mathcal{X}\times\mathcal{Y})^{|\mathbf{i}|}$, and $\bar{\mu}\in\mathcal{M}_{S_{\mathbf{i}}}$. Here is the sketch of the proof of

Equation (6.26). Justification for line (6.27) to (6.30) follows below.

$$
\begin{aligned}
&\mathop{\mathbf{E}}_{S_{\mathbf{i}^c}\sim D^{m-|\mathbf{i}|}} e^{(m-|\mathbf{i}|)2\,(\overline{R}_S(h^{\bar{\mu}}_{S_\mathbf{i}})-\overline{R}_D(h^{\bar{\mu}}_{S_\mathbf{i}}))^2} \\
&\quad= \mathop{\mathbf{E}}_{S_{\mathbf{i}^c}\sim D^{(m-|\mathbf{i}|)}} e^{(m-|\mathbf{i}|)2\left(\overline{R}_S(h^{\bar{\mu}}_{S_\mathbf{i}})-\widetilde{R}_S(h^{\bar{\mu}}_{S_\mathbf{i}})+\widetilde{R}_S(h^{\bar{\mu}}_{S_\mathbf{i}})-\overline{R}_D(h^{\bar{\mu}}_{S_\mathbf{i}})\right)^2} \\
&\quad\leq \mathop{\mathbf{E}}_{S_{\mathbf{i}^c}\sim D^{(m-|\mathbf{i}|)}} e^{(m-|\mathbf{i}|)2\left(\left[\overline{R}_S(h^{\bar{\mu}}_{S_\mathbf{i}})-\widetilde{R}_S(h^{\bar{\mu}}_{S_\mathbf{i}})\right]^2+2\left|\overline{R}_S(h^{\bar{\mu}}_{S_\mathbf{i}})-\widetilde{R}_S(h^{\bar{\mu}}_{S_\mathbf{i}})\right|\cdot\left|\widetilde{R}_S(h^{\bar{\mu}}_{S_\mathbf{i}})-\overline{R}_D(h^{\bar{\mu}}_{S_\mathbf{i}})\right|+\left[\widetilde{R}_S(h^{\bar{\mu}}_{S_\mathbf{i}})-\overline{R}_D(h^{\bar{\mu}}_{S_\mathbf{i}})\right]^2\right)} \\
&\quad\leq \mathop{\mathbf{E}}_{S_{\mathbf{i}^c}\sim D^{(m-|\mathbf{i}|)}} e^{(m-|\mathbf{i}|)2\left(\left[\frac{|\mathbf{i}|}{m}\right]^2+2\frac{|\mathbf{i}|}{m}\cdot 1+\left[\widetilde{R}_S(h^{\bar{\mu}}_{S_\mathbf{i}})-\overline{R}_D(h^{\bar{\mu}}_{S_\mathbf{i}})\right]^2\right)} && (6.27)\\
&\quad\leq \mathop{\mathbf{E}}_{S_{\mathbf{i}^c}\sim D^{(m-|\mathbf{i}|)}} e^{4ld\,+\,(m-|\mathbf{i}|)2\left[\widetilde{R}_S(h^{\bar{\mu}}_{S_\mathbf{i}})-\overline{R}_D(h^{\bar{\mu}}_{S_\mathbf{i}})\right]^2} && (6.28)\\
&\quad\leq \mathop{\mathbf{E}}_{S_{\mathbf{i}^c}\sim D^{(m-|\mathbf{i}|)}} e^{4ld\,+\,(m-|\mathbf{i}|)\cdot\mathrm{kl}\left(\widetilde{R}_S(h^{\bar{\mu}}_{S_\mathbf{i}})\|\overline{R}_D(h^{\bar{\mu}}_{S_\mathbf{i}})\right)} && (6.29)\\
&\quad= e^{4ld}\cdot \mathop{\mathbf{E}}_{S_{\mathbf{i}^c}\sim D^{(m-|\mathbf{i}|)}} e^{(m-|\mathbf{i}|)\cdot\mathrm{kl}\left(\widetilde{R}_S(h^{\bar{\mu}}_{S_\mathbf{i}})\|\overline{R}_D(h^{\bar{\mu}}_{S_\mathbf{i}})\right)} \\
&\quad\leq e^{4ld}\cdot 2\sqrt{m-|\mathbf{i}|} && (6.30)\\
&\quad\leq e^{4ld}\cdot 2\sqrt{m}
\end{aligned}
$$

Line (6.27) follows from Equation (6.10) and the fact that the exponential function is increasing. For Line (6.28), we have :

$$(m-|\mathbf{i}|)2\left(\frac{|\mathbf{i}|^2}{m^2}+\frac{2|\mathbf{i}|}{m}\right)=|\mathbf{i}|(m-|\mathbf{i}|)2\left(\frac{|\mathbf{i}|}{m^2}+\frac{2}{m}\right).$$

From $|\mathbf{i}|\leq ld\leq m$, it follows that $m-|\mathbf{i}|\leq m$ and $\frac{|\mathbf{i}|}{m}\leq 1$. Thus, we have:

$$(m-|\mathbf{i}|)2\left(\frac{|\mathbf{i}|}{m^2}+\frac{2}{m}\right)\leq m2\left(\frac{|\mathbf{i}|}{m^2}+\frac{2}{m}\right)\leq 2\frac{|\mathbf{i}|}{m}+2\leq 4.$$

Therefore, we have:

$$(m-|\mathbf{i}|)2\left(\frac{|\mathbf{i}|^2}{m^2}+\frac{2|\mathbf{i}|}{m}\right)=|\mathbf{i}|(m-|\mathbf{i}|)2\left(\frac{|\mathbf{i}|}{m^2}+\frac{2}{m}\right)\leq 4|\mathbf{i}|\leq 4ld.$$

Line (6.29) follows directly from the property : $2(q-p)^2\leq \mathrm{kl}(q\parallel p)$ (Pinsker's inequality [13]). Finally, for Line (6.30), first observe that $\widetilde{R}_S(h^{\bar{\mu}}_{S_\mathbf{i}})$ is an arithmetic mean of $(m-|\mathbf{i}|)$ iid random variables. Thus Line (6.30) is simply an application of Lemma 6.2.1

with $M(X)$ replaced by $\widetilde{R}_S(h^{\bar{\mu}}_{S_{\mathbf{i}}})$, n replaced by $m-|\mathbf{i}|$, and ν replaced by $\overline{R}_D(h^{\bar{\mu}}_{S_{\mathbf{i}}})$. Thus, Equation (6.25) is proved.

By substituting Equations (6.12) and (6.25) into Equation (6.24) we have:

$$(m-ld)2\left(\frac{1}{2}\left[1+\frac{1}{\zeta(1)}\;\zeta_S^Q\right]-\frac{1}{2}\left[1+\frac{1}{\zeta(1)}\;\zeta_D^Q\right]\right)^2 \leq \frac{\zeta'(1)}{\zeta(1)}\cdot\mathrm{KL}(Q\|P^S)+\ln\left[\frac{1}{\delta}\cdot e^{4ld}\cdot 2\sqrt{m}\right]$$

By rearranging the above equations we get:

$$\zeta_D^Q \leq \zeta_S^Q + \frac{\zeta(1)}{\sqrt{\frac{1}{2}(m-l\deg(\zeta))}}\sqrt{\frac{\zeta'(1)}{\zeta(1)}\cdot\mathrm{KL}(Q\|P^S)+4\,l\deg(\zeta)+\ln\frac{2\sqrt{m}}{\delta}}$$

□

In the particular case of non-sample compressed classifiers (when $l=0$) Theorem 6.3.3 reduces to the following corollary:

Corollary 6.3.4. *For any D, any $\mathcal{H}$ of sets of classifiers any prior $\mathcal{P}$, any $\delta\in(0,1]$, and any margin loss function we have:*

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\forall Q\in\mathcal{H}\quad :\\ \quad\zeta_D^Q\leq\zeta_S^Q+\frac{\zeta(1)}{\sqrt{\frac{1}{2}m}}\sqrt{\frac{\zeta'(1)}{\zeta(1)}\cdot\mathrm{KL}(Q\|P^S)+\ln\frac{2\sqrt{m}}{\delta}}\end{array}\right)\geq 1-\delta$$

Seeger bound with KL

The following theorem gives a bound on ζ_D^Q and, consequently, on $R_D(B_Q)$.

Theorem 6.3.5. *For any D, any family $(\mathcal{H}^S)_{S\in\mathcal{D}^m}$ of sets of sc-classifiers of size at most l, any prior $\mathcal{P}$, any $\delta\in(0,1]$, any margin loss function ζ of degree $<m/l$, we have*

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\forall Q\in\mathcal{H}^S :\\ \mathrm{kl}^+\left(\frac{m}{m-l\cdot\deg(\zeta)}\left(\frac{1}{2}\left[1+\frac{1}{\zeta(1)}\;\zeta_S^Q\right]+\frac{ld}{m}\right)\,\|\,\frac{1}{2}\left[1+\frac{1}{\zeta(1)}\;\zeta_D^Q\right]\right)\\ \qquad\leq\frac{1}{m-l\cdot\deg(\zeta)}\left(\frac{\zeta'(1)}{\zeta(1)}\cdot\mathrm{KL}(Q\|P^S)+\ln\frac{2\sqrt{m}}{\delta}\right)\end{array}\right)\geq 1-\delta$$

where $\mathrm{kl}(q\|p) \overset{def}{=} q\ln\frac{q}{p} + (1-q)\ln\frac{1-q}{1-p}$*, and where* $\mathrm{kl}^{+}(q\|p) = \mathrm{kl}(q\|p)$ *if* $q \leq p$ *and* 0 *otherwise. Moreover, if* $l = 0$*, the function* kl^{+} *can be replace by the function* kl *in the statement, giving rise to both a lower and an upper bound of* ζ_D^Q*.*

Proof. The first part of the proof is very similar to the one of Theorem 6.3.3. Similarly as in the Section 6.2, we again define d, $\bar{h} = \overline{h_1..h_k}$ (with $k \in \{0,..,d\}$), $\overline{R}_D(\bar{h})$, $\overline{R}_S(\bar{h})$, $\overline{\mathcal{H}^S}$, $\overline{P^S}$, $\overline{Q}$, ζ_D^Q, and ζ_S^Q. However, we will instead consider this quite different random variable that among other thing is not based on $\overline{R}_S(\bar{h})$ but on a slightly different value $\widetilde{R}_S(\bar{h})$ given by Equation (6.8). Note that in the case of the McAllester bound (Theorem 6.3.3) , we manage to deal with the bias that comes from the examples of the compression set in the calculation of the random variable $X_{\overline{P^S}}$ (see Equations (6.27) to (6.30)). In the case of Seeger bound, this is no more possible. In the presence of such bias, the random variable $X_{\overline{P^S}}$ using the kl as the divergence $\mathcal{D}$ can be huge. For this reason, contrarily to the preceding proof, in this proof, we consider the unbiased $\widetilde{R}_S(\bar{h})$ instead of $\overline{R}_S(\bar{h})$ in the definition of the random variable $X_{\overline{P^S}}$[1]. Therefore, we consider the following random variable which is based on the value $\widetilde{R}_S(\bar{h})$.

$$X_{\overline{P^S}} \overset{\mathrm{def}}{=} \mathop{\mathbf{E}}_{\bar{h}\sim\overline{p^S}} e^{(m-|\mathbf{i}_{\bar{h}}|)\mathrm{kl}(\widetilde{R}_S(\bar{h}),\overline{R}_D(\bar{h}))}\,, \tag{6.31}$$

By using Markov's inequality(Lemma 5.1.5) we have:

$$\Pr_{S\sim D^m}\left(X_{\overline{P^S}} \leq \frac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m} X_{\overline{P^S}}\right) \geq 1-\delta\,.$$

By taking the logarithm on each side of the innermost inequality and by transforming the expectation over $\overline{P^S}$ into an expectation over $\overline{Q}$ using Equation (5.4), we obtain

$$\Pr_{S\sim D^m}\!\left(\forall Q\colon \ln\!\left[\mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} \frac{\overline{P^S}(\bar{h})}{\overline{Q}(\bar{h})} e^{(m-|\mathbf{i}_{\bar{h}}|)\mathrm{kl}(\widetilde{R}_S(\bar{h}),\overline{R}_D(\bar{h}))}\right] \leq \ln\!\left[\frac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{\bar{h}\sim\overline{P^S}} e^{(m-|\mathbf{i}_{\bar{h}}|)\mathrm{kl}(\widetilde{R}_S(\bar{h}),\overline{R}_D(\bar{h}))}\right]\right) \geq 1-\delta\,. \tag{6.32}$$

Using Equation (6.14), together with Jensen's inequality(Lemma 5.1.6) applied to

[1] This is the main reason why the mechanism of the proof of Theorems 6.3.3 and 6.3.5 seems so different, although in the non sample compression case where $\widetilde{R}_S(\bar{h}) = \overline{R}_S(\bar{h})$, the proof of these theorems are very similar (See Chapter 5, Section 5.1.1).

the concave $\ln(x)$ gives

$$\ln\left[\mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} \frac{\overline{P^S}(\bar{h})}{\overline{Q}(\bar{h})}\; e^{(m-|\mathbf{i}_{\bar{h}}|)\mathrm{kl}(\widetilde{R}_S(\bar{h}),\overline{R}_D(\bar{h}))}\right] \geq -\frac{\zeta'(1)}{\zeta(1)}\cdot \mathrm{KL}(Q\|P^S) \\ + \mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}}\,(m-|\mathbf{i}_{\bar{h}}|)\,\mathrm{kl}(\widetilde{R}_S(\bar{h}),\overline{R}_D(\bar{h})) \quad (6.33)$$

where

$$\mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} \ln\frac{\overline{P^S}(\bar{h})}{\overline{Q}(\bar{h})} = -\mathrm{KL}(\overline{Q}||\overline{P^S}) = -\frac{\zeta'(1)}{\zeta(1)}\cdot KL(Q\|P^S)\,.$$

Again from the Jensen's inequality, applied to the convex function $\mathrm{kl}(\cdot\,\|\,\cdot)$, and the fact that $m-|\mathbf{i}_{\bar{h}}| \geq m-l\cdot d$, we obtain:

$$\mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}}\,(m-|\mathbf{i}_{\bar{h}}|)\;\mathrm{kl}\left(\widetilde{R}_S(\bar{h})\,\|\,\overline{R}_D(\bar{h})\right) \;\geq\; (m-ld)\;\mathrm{kl}\left(\mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}}\widetilde{R}_S(\bar{h})\,\|\,\mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}}\overline{R}_D(\bar{h})\right) \quad (6.34)$$

Let us now analyse the value of $\mathbf{E}_{S\sim D^m}X_{\overline{P^S}}$.

Let $\mathbf{i}^c$ be the vector of indices of $\mathcal{I}$ that are not in the vector $\mathbf{i}$, and note that

$$\mathop{\mathbf{E}}_{S\sim D^m}\;\mathop{\mathbf{E}}_{\bar{h}\sim\overline{P^S}} e^{(m-|\mathbf{i}_{\bar{h}}|)\,\mathrm{kl}(\widetilde{R}_S(\bar{h}),\overline{R}_D(\bar{h}))} \;=\; \mathop{\mathbf{E}}_{\mathbf{i}\sim\overline{P}_{\mathcal{I}}}\;\mathop{\mathbf{E}}_{S_{\mathbf{i}}\sim D^{|\mathbf{i}|}}\;\mathop{\mathbf{E}}_{\bar{\mu}\sim\overline{P}_{S_{\mathbf{i}}}}\;\mathop{\mathbf{E}}_{S_{\mathbf{i}^c}\sim D^{m-|\mathbf{i}|}} e^{|\mathbf{i}^c|\,\mathrm{kl}(\widetilde{R}_S(h^{\bar{\mu}}_{S_{\mathbf{i}}}),\overline{R}_D(h^{\bar{\mu}}_{S_{\mathbf{i}}}))}$$

Since $\widetilde{R}_S(h^{\bar{\mu}}_{S_{\mathbf{i}}})$ is an arithmetic mean of iid random variables, one can apply Lemma 6.2.1 with $M(X)$ replaced by $\widetilde{R}_S(h^{\bar{\mu}}_{S_{\mathbf{i}}})$, n replaced by $m-|\mathbf{i}|$, and ν replaced by $\overline{R}_D(h^{\bar{\mu}}_{S_{\mathbf{i}}})$ to obtain:

$$\mathbf{E}_{S_{\mathbf{i}^c}\sim D^{m-|\mathbf{i}|}}\; e^{(m-|\mathbf{i}|)\,\mathrm{kl}(\widetilde{R}_S(h^{\bar{\mu}}_{S_{\mathbf{i}}}),\overline{R}_D(h^{\bar{\mu}}_{S_{\mathbf{i}}}))} \;\leq\; 2\sqrt{m-|\mathbf{i}|} \;\leq\; 2\sqrt{m}\,. \quad (6.35)$$

By rearranging Equation (6.32) based on Equation 6.35 and using Equations 6.33 and 6.34, we have:

$$(m-ld)\;\;\mathrm{kl}\left(\mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}}\widetilde{R}_S(\bar{h})\,\|\,\mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}}\overline{R}_D(\bar{h})\right) \;\leq\; \ln\frac{2\sqrt{m}}{\delta} + \frac{\zeta'(1)}{\zeta(1)}\cdot \mathrm{KL}(Q\|P^S) \quad (6.36)$$

Finally, observe that for any classifier $\bar{h} \in \overline{\mathcal{H}^S}$, we have

$$\begin{aligned}
\widetilde{R}_S(\bar{h}) &\leq \left(\overline{R}_S(\bar{h}) + \frac{ld}{m}\right) \frac{m}{m - |\mathbf{i}|} \\
&\leq \left(\overline{R}_S(\bar{h}) + \frac{ld}{m}\right) \frac{m}{m - ld} \\
&\leq \frac{m}{m - ld} \overline{R}_S(\bar{h}) + \frac{ld}{m - ld} \\
&= \frac{m}{m - ld} \left(\overline{R}_S(\bar{h}) + \frac{ld}{m}\right), \qquad (6.37)
\end{aligned}$$

and consider the following two cases.

case 1 : $l = 0$. In that case we have that $\mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} \widetilde{R}_S(\bar{h}) = \mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} \overline{R}_S(\bar{h})$. Hence we have

$$\mathrm{kl}\left(\mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} \widetilde{R}_S(\bar{h}) \,\|\, \mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} \overline{R}_D(\bar{h})\right) = \mathrm{kl}\left(\frac{m}{m - ld} \mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} \overline{R}_S(\bar{h}) + \frac{ld}{m} \,\|\, \mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} \overline{R}_D(\bar{h})\right)$$

case 2 : $l > 0$. In that case, following Equation (6.37), we have:

$$\mathrm{kl}\left(\mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} \widetilde{R}_S(\bar{h}) \,\|\, \mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} \overline{R}_D(\bar{h})\right) \geq \mathrm{kl}^+\left(\frac{m}{m - ld} \mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} \overline{R}_S(\bar{h}) + \frac{ld}{m} \,\|\, \mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} \overline{R}_D(\bar{h})\right)$$

In each cases, the result then follows from Equation (6.36):

$$(m - ld)\ \mathrm{kl}^+\left(\frac{m}{m - ld} \mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} \overline{R}_S(\bar{h}) + \frac{ld}{m} \,\|\, \mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} \overline{R}_D(\bar{h})\right) \leq \ln \frac{2\sqrt{m}}{\delta} + \frac{\zeta'(1)}{\zeta(1)} \cdot \mathrm{KL}(Q\|P^S)$$

Now, by replacing $\mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} \overline{R}_U(\bar{h})$ by $\frac{1}{2}\left[1 + \frac{1}{\zeta(1)}\ \zeta_U^Q\right]$ from Equation (6.12) for $U = D$ and $U = S$ we get:

$$\begin{aligned}
&\mathrm{kl}^+\left(\frac{m}{m - l \cdot \deg(\zeta)} (\frac{1}{2}\left[1 + \frac{1}{\zeta(1)}\ \zeta_S^Q\right] + \frac{ld}{m}) \,\|\, \frac{1}{2}\left[1 + \frac{1}{\zeta(1)}\ \zeta_D^Q\right]\right) \\
&\qquad \leq \frac{1}{m - l \cdot \deg(\zeta)}\left(\frac{\zeta'(1)}{\zeta(1)} \cdot \mathrm{KL}(Q\|P^S) + \ln \frac{2\sqrt{m}}{\delta}\right)
\end{aligned}$$

□

In the particular case of non-sample compressed classifiers (when $l = 0$), Theorem 6.3.5 reduces to the following corollary:

Corollary 6.3.6. *For any D, any $\mathcal{H}$ of sets of classifiers any prior $\mathcal{P}$, any $\delta \in (0,1]$, any positive real number C_1, and any margin loss function we have:*

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\forall Q\in\mathcal{H}:\\ \mathrm{kl}\left(\frac{1}{2}\left[1+\frac{1}{\zeta(1)}\ \zeta_S^Q\right]\|\ \frac{1}{2}\left[1+\frac{1}{\zeta(1)}\ \zeta_D^Q\right]\right)\\ \qquad\leq \frac{1}{m}\left(\frac{\zeta'(1)}{\zeta(1)}\cdot \mathrm{KL}(Q\|P^S)+\ln\frac{2\sqrt{m}}{\delta}\right)\end{array}\right)\geq 1-\delta$$

where $\mathrm{kl}(q\|p) \overset{def}{=} q\ln\frac{q}{p}+(1-q)\ln\frac{1-q}{1-p}$.

6.4 PAC-Bayes Bounds without KL

In this section, we present the other PAC-Bayes risk bounds that have the unusual property of having no KL divergence when the posterior is aligned with the prior in some precise way defined in the following subsection.

6.4.1 The case of aligned posteriors

Recall the definition of the aligned posteriors in the non-sample compression setting (see Definition 5.1.8), here, we also present the notion of aligned in the sample compression setting in the similar way. To define the notion of aligned posteriors in the sample compression setting, we need to consider the boolean complement $-h^{\mu}_{S_{\mathbf{i}}}$ of any sc-classifier $h^{\mu}_{S_{\mathbf{i}}}$. Thus, we will now always suppose that all the message sets are of the form

$$\mathcal{M}_{S_{\mathbf{i}}} \;=\; \mathcal{M}^1_{S_{\mathbf{i}}}\times\{{\scriptstyle +},{\scriptstyle -}\}\,, \tag{6.38}$$

and that we will always have :

$$h^{(\sigma,+)}_{S_{\mathbf{i}}}(\mathbf{x}) = -h^{(\sigma,-)}_{S_{\mathbf{i}}}(\mathbf{x})\;\;\forall\mathbf{x},\;\sigma\in\mathcal{M}^1_{S_{\mathbf{i}}} \tag{6.39}$$

Definition 6.4.1. *Given a prior $\mathcal{P}$ and a training sequence S, we say that a posterior Q* is aligned on P^S *if*

$$Q(h^{(\sigma,+)}_{S_{\mathbf{i}}})+Q(h^{(\sigma,-)}_{S_{\mathbf{i}}}) \;=\; P^S(h^{(\sigma,+)}_{S_{\mathbf{i}}})+P^S(h^{(\sigma,-)}_{S_{\mathbf{i}}}) \qquad \forall(\mathbf{i},\sigma)\in\mathcal{I}\times\mathcal{M}^1_{S_{\mathbf{i}}}\,.$$

Note that an aligned posterior is totally defined by the values

$$w(\mathbf{i},\sigma) \overset{\text{def}}{=} Q(h_{S_\mathbf{i}}^{(\sigma,+)}) - Q(h_{S_\mathbf{i}}^{(\sigma,-)})\,, \tag{6.40}$$

under the constraints $$|w(\mathbf{i},\sigma)| \;\leq\; P^S(h_{S_\mathbf{i}}^{(\sigma,+)})+P^S(h_{S_\mathbf{i}}^{(\sigma,-)}) \qquad (\mathbf{i},\sigma)\in\mathcal{I}\times\mathcal{M}^1_{S_\mathbf{i}}\,. \tag{6.41}$$

Indeed, it immediately follows that Q can be retrieve from $\mathcal{P}$ and w because

$$Q(h_{S_\mathbf{i}}^{(\sigma,\pm)}) \;=\; \frac{1}{2}\left(P^S(h_{S_\mathbf{i}}^{(\sigma,+)})+P^S(h_{S_\mathbf{i}}^{(\sigma,-)}) \pm w(\mathbf{i},\sigma)\right)\,. \tag{6.42}$$

Moreover, given any function $w:\mathcal{I}\times\mathcal{M}^1_{S_\mathbf{i}}\to\mathbb{R}$ satisfying Equation (6.41), the function Q given by Equation (6.42) is a distribution aligned on P^S.

The next proposition follows directly from what precedes and points out that there is (almost) no loss of expressiveness if we restrict ourselves to aligned posterior.

Proposition 6.4.2. *Let $\mathcal{P}$ be a prior, S a training sequence, and Q a distribution on $\mathcal{H}^S$ for which there exists $A>0$ such that for all $\mathbf{i}$ and σ, $A\,|Q(h_{S_\mathbf{i}}^{(\sigma,+)}) - Q(h_{S_\mathbf{i}}^{(\sigma,-)})| \leq P^S(h_{S_\mathbf{i}}^{(\sigma,+)}) + P^S(h_{S_\mathbf{i}}^{(\sigma,-)})$. Let Q' be a distribution aligned on P^S such that $w'(\mathbf{i},\sigma) = A\,(Q(h_{S_\mathbf{i}}^{(\sigma,+)}) - Q(h_{S_\mathbf{i}}^{(\sigma,-)}))$. Then Q' is Bayes-equivalent to Q (i.e., $B_{Q'}(x)=B_Q(x)\ \forall x\in\mathcal{X}$).*

Proof. Let $Q'(h_{S_\mathbf{i}}^{(\sigma,+)}) = \frac{1}{2}(AQ(h_{S_\mathbf{i}}^{(\sigma,+)}) - AQ(h_{S_\mathbf{i}}^{(\sigma,-)}) + w'(\mathbf{i},\sigma))$, and let $Q'(h_{S_\mathbf{i}}^{(\sigma,-)}) = \frac{1}{2}(w'(\mathbf{i},\sigma)) - AQ(h_{S_\mathbf{i}}^{(\sigma,+)}) + AQ(h_{S_\mathbf{i}}^{(\sigma,-)}))$. This way, we have:

$$Q'(h_{S_\mathbf{i}}^{(\sigma,+)}) + Q'(h_{S_\mathbf{i}}^{(\sigma,-)}) = w'(\mathbf{i},\sigma) = A\,(Q(h_{S_\mathbf{i}}^{(\sigma,+)}) - Q(h_{S_\mathbf{i}}^{(\sigma,-)})) \leq P^S(h_{S_\mathbf{i}}^{(\sigma,+)}) + P^S(h_{S_\mathbf{i}}^{(\sigma,-)}),$$

which shows that Q' is aligned on P^S. Moreover, we have:

$$\begin{aligned} B_{Q'}(\mathbf{x}) &= \operatorname{sgn}\left(\sum_{\mathbf{i}\in\mathcal{I}}\sum_{s\in\{-,+\}}\int_{\mathcal{M}^1_{S_\mathbf{i}}} d\sigma Q'(h_{S_\mathbf{i}}^{(\sigma,s)})h_{S_\mathbf{i}}^{(\sigma,s)}(\mathbf{x})\right)\\ &= \operatorname{sgn}\left(\sum_{\mathbf{i}\in\mathcal{I}}\int_{\mathcal{M}^1_{S_\mathbf{i}}} d\sigma\left[Q'(h_{S_\mathbf{i}}^{(\sigma,+)}) - Q'(h_{S_\mathbf{i}}^{(\sigma,-)}\right]h_{S_\mathbf{i}}^{(\sigma,+)}(\mathbf{x}\right) \end{aligned}$$

The last equality is obtained from Equation (6.39). By incorporating the value of

$$\left[Q'(h_{S_\mathbf{i}}^{(\sigma,+)}) - Q'(h_{S_\mathbf{i}}^{(\sigma,-)}\right] = A\,(Q(h_{S_\mathbf{i}}^{(\sigma,+)}) + Q(h_{S_\mathbf{i}}^{(\sigma,-)})),$$

into above equation, we then have:

$$\begin{aligned} B_{Q'}(\mathbf{x}) &= \operatorname{sgn}\left(A \sum_{\mathbf{i}\in\mathcal{I}} \sum_{s\in\{-,+\}} \int_{\mathcal{M}^1} d\sigma Q(h_{S_{\mathbf{i}}}^{(\sigma,s)}) h_{S_{\mathbf{i}}}^{(\sigma,s)}(\mathbf{x}) \right) \\ &= \operatorname{sgn}\left(\sum_{\mathbf{i}\in\mathcal{I}} \sum_{s\in\{-,+\}} \int_{\mathcal{M}^1} d\sigma Q(h_{S_{\mathbf{i}}}^{(\sigma,s)}) h_{S_{\mathbf{i}}}^{(\sigma,s)}(\mathbf{x}) \right) = B_Q(\mathbf{x}). \end{aligned}$$

□

Now, by using the following Lemma we can obtain versions of Theorems 6.4.4 and 6.4.7 that have no KL divergence terms (when the posterior is aligned with the prior).

Lemma 6.4.3. *Let S be any training sequence, $d \overset{def}{=} \deg \zeta$ and posterior Q be aligned with prior P^S. Let $D(q,p)$ be a function such that $D(q,p) = D(1-q, 1-p)$. Similarly as in the Section 6.2, we define $\bar{h} = \overline{h_1..h_k}$ (with $k \in \{0,..,d\}$), $\overline{R}_D(\bar{h})$, $\overline{R}_S(\bar{h})$, $\overline{\mathcal{H}^S}$, $\overline{P^S}$, $\overline{Q}$, then we have:*

$$X_{\overline{P^S}} \overset{def}{=} \mathop{\mathbf{E}}_{\bar{h}\sim\overline{P^S}} e^{(m-|\mathbf{i}_{\bar{h}}|)\cdot D(\overline{R}_S(\bar{h})||\overline{R}_D(\bar{h}))} = \mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} e^{(m-|\mathbf{i}_{\bar{h}}|)\cdot D(\overline{R}_S(\bar{h})||\overline{R}_D(\bar{h}))}. \qquad (6.43)$$

Proof. For each $k \in \{0,..,d\}$, define $\overline{\mathcal{H}^S_{(k)}}$ as the set of abstract classifiers $\overline{h}$ that are k-tupples $\overline{h_1..h_k}$. Then, for each $\overline{h} \in \overline{\mathcal{H}^S_{(k)}}$ and each $j = 0,..,2^k-1$, let us define $\overline{h}^{[j]} \overset{\text{def}}{=} \overline{h_1^{(s_1)}..h_k^{(s_k)}}$, where $s_1s_2..s_k$ is the binary representation of the number j, and where $h^{(0)} = h$ and $h^{(1)} = -h$. For any $\overline{h} \in \overline{\mathcal{H}^S_{(k)}}$, let $\mathcal{G}(\overline{h})$ be the set of all $\overline{h}^{[j]}$s for the different choices of j. Note that, given any two $\overline{h}, \overline{h'} \in \overline{\mathcal{H}^S_{(k)}}$, both $\mathcal{G}(\overline{h})$ and $\mathcal{G}(\overline{h'})$ either coincide or are disjoint. They will coincide iff $\overline{h'} = \overline{h}^{[j]}$ for some j, and in that case they will have the same compression sequence, i.e., $\mathbf{i}_{\overline{h}^{[j]}} = \mathbf{i}_{\overline{h}}$. Moreover, if Q is aligned on

P, we have:

$$\begin{aligned}
\sum_{j=0}^{2^k-1} \overline{P}(\overline{h}^{[j]}) &= \frac{a_k}{\zeta(1)} \sum_{\mathbf{s}\in\{0,1\}^k} \prod_{i=1}^{k} P\left(h_i^{(s_i)}\right) \\
&= \frac{a_k}{\zeta(1)} \prod_{i=1}^{k} \left[P\left(h_i^{(0)}\right) + P\left(h_i^{(1)}\right)\right] \\
&= \frac{a_k}{\zeta(1)} \prod_{i=1}^{k} \left[Q\left(h_i^{(0)}\right) + Q\left(h_i^{(1)}\right)\right] \\
&= \frac{a_k}{\zeta(1)} \sum_{\mathbf{s}\in\{0,1\}^k} \prod_{i=1}^{k} Q\left(h_i^{(s_i)}\right) \\
&= \sum_{j=0}^{2^k-1} \overline{Q}(\overline{h}^{[j]})\,.
\end{aligned} \tag{6.44}$$

By the definition of abstract classifier from Equation (6.11) for all $\overline{h}^{[j]} \in \mathcal{G}(\overline{h})$ we have one of the two following cases:

$$(1) : \overline{R}_S(\overline{h}) = \overline{R}_S(\overline{h}^{[j]}) \textit{ and } \overline{R}_D(\overline{h}) = \overline{R}_D(\overline{h}^{[j]}).$$

$$(2) : \overline{R}_S(\overline{h}) = 1 - \overline{R}_S(\overline{h}^{[j]}) \textit{ and } \overline{R}_D(\overline{h}) = 1 - \overline{R}_D(\overline{h}^{[j]}).$$

Therefore, from the property $D(q,p) = D(1-q, 1-p)$, it follows that:

$$\mathcal{D}(\overline{R}_S(\overline{h}), \overline{R}_D(\overline{h})) = \mathcal{D}(\overline{R}_S(\overline{h}^{[j]}), \overline{R}_D(\overline{h}^{[j]})) \tag{6.45}$$

From Equations (6.44) and (6.45), we now have:

$$\begin{aligned}
&\int_{\overline{h}\in\overline{\mathcal{H}^S_{(k)}}} \overline{P}(\overline{h})e^{(m-|\mathbf{i}_{\overline{h}}|)\cdot\mathcal{D}(\overline{R}_S(\overline{h}),\overline{R}_D(\overline{h}))} \\
&\quad = \frac{1}{2^k}\sum_{j=0}^{2^k-1}\int_{\overline{h}\in\overline{\mathcal{H}^S_{(k)}}} \overline{P}(\overline{h})e^{(m-|\mathbf{i}_{\overline{h}}|)\cdot\mathcal{D}(\overline{R}_S(\overline{h}),\overline{R}_D(\overline{h}))} \\
&\quad = \frac{1}{2^k}\sum_{j=0}^{2^k-1}\int_{\overline{h}\in\overline{\mathcal{H}^S_{(k)}}} \overline{P}(\overline{h}^{[j]})e^{(m-|\mathbf{i}_{\overline{h}^{[j]}}|)\cdot\mathcal{D}(\overline{R}_S(\overline{h}^{[j]}),\overline{R}_D(\overline{h}^{[j]}))} \\
&\quad = \frac{1}{2^k}\sum_{j=0}^{2^k-1}\int_{\overline{h}\in\overline{\mathcal{H}^S_{(k)}}} \overline{P}(\overline{h}^{[j]})e^{(m-|\mathbf{i}_{\overline{h}}|)\cdot\mathcal{D}(\overline{R}_S(\overline{h}),\overline{R}_D(\overline{h}))} \\
&\quad = \frac{1}{2^k}\int_{\overline{h}\in\overline{\mathcal{H}^S_{(k)}}} \sum_{j=0}^{2^k-1}\overline{P}(\overline{h}^{[j]})e^{(m-|\mathbf{i}_{\overline{h}}|)\cdot\mathcal{D}(\overline{R}_S(\overline{h}),\overline{R}_D(\overline{h}))} \\
&\quad = \frac{1}{2^k}\int_{\overline{h}\in\overline{\mathcal{H}^S_{(k)}}} \sum_{j=0}^{2^k-1}\overline{Q}(\overline{h}^{[j]})\ e^{(m-|\mathbf{i}_{\overline{h}}|)\cdot\mathcal{D}(\overline{R}_S(\overline{h}),\overline{R}_D(\overline{h}))} \\
&\quad\ \vdots \\
&\quad = \int_{\overline{h}\in\overline{\mathcal{H}^S_{(k)}}} \overline{Q}(\overline{h})e^{(m-|\mathbf{i}_{\overline{h}}|)\cdot\mathcal{D}(\overline{R}_S(\overline{h}),\overline{R}_D(\overline{h}))}\,.
\end{aligned} \tag{6.46}$$

Thus,

$$\begin{aligned}
&\mathop{\mathbf{E}}_{\overline{h}\sim\overline{P}}\ e^{(m-|\mathbf{i}_{\overline{h}}|)\cdot\mathcal{D}(\overline{R}_S(\overline{h}),\overline{R}_D(\overline{h}))} \\
&\qquad = \sum_{k=0}^{\deg\zeta}\int_{\overline{h}\in\overline{\mathcal{H}^S_{(k)}}} \overline{P}(\overline{h})e^{(m-|\mathbf{i}_{\overline{h}}|)\cdot\mathcal{D}(\overline{R}_S(\overline{h}),\overline{R}_D(\overline{h}))} \\
&\qquad = \sum_{k=0}^{\deg\zeta}\int_{\overline{h}\in\overline{\mathcal{H}^S_{(k)}}} \overline{Q}(\overline{h})e^{(m-|\mathbf{i}_{\overline{h}}|)\cdot\mathcal{D}(\overline{R}_S(\overline{h}),\overline{R}_D(\overline{h}))} \\
&\qquad = \mathop{\mathbf{E}}_{\overline{h}\sim\overline{Q}}\ e^{(m-|\mathbf{i}_{\overline{h}}|)\cdot\mathcal{D}(\overline{R}_S(\overline{h}),\overline{R}_D(\overline{h}))}\,.
\end{aligned}$$

□

We now provide a PAC-Bayes bound for aligned posteriors which does not depend on how far they are from the prior.

McAllester bound without KL

Theorem 6.4.4. *For any D, for any $m \geq 8$, for any family $(\mathcal{H}^S)_{S \in D^m}$ of sets of sc-classifiers of size at most l, for any prior $\mathcal{P}$, for any margin loss function ζ such that $l \cdot \deg(\zeta) < m$, and for for any $\delta \in (0,1]$, we have*

$$\Pr_{S \sim D^m}\left(\forall Q \text{ aligned on } P^S \ : \quad \zeta_D^Q \leq \zeta_S^Q + \frac{\zeta(1)}{\sqrt{\frac{1}{2}(m - l\deg(\zeta))}} \sqrt{4\, l \deg(\zeta) + \ln \frac{2\sqrt{m}}{\delta}} \right) \geq 1 - \delta$$

Proof. Similarly as in the proof of Theorem 6.3.1, we define $d = \deg \zeta$, $\bar{h} = \overline{h_1..h_k}$ (with $k \in \{0, .., d\}$), $\overline{R}_D(\bar{h})$, $\overline{R}_S(\bar{h})$, $\overline{\mathcal{H}^S}$, $\overline{P^S}$, $\overline{Q}$, ζ_D^Q, and ζ_S^Q. However, we will consider the following random variable

$$X_{\overline{P^S}} \overset{\text{def}}{=} \mathop{\mathbf{E}}_{\bar{h} \sim \overline{P^S}} e^{(m - |\mathbf{i}_{\bar{h}}|) \cdot 2(\overline{R}_S(\bar{h}) - \overline{R}_D(\bar{h}))^2} . \tag{6.47}$$

By the same proof as in Lemma 6.4.3, except that $D(\overline{R}_S(\bar{h}) || \overline{R}_D(\bar{h}))$ is replaced by $(\overline{R}_S(\bar{h}) - \overline{R}_D(\bar{h}))^2$, one can show that : for any posterior Q aligned on P^S, we have:

$$X_{\overline{P^S}} = \mathop{\mathbf{E}}_{\bar{h} \sim \overline{Q}} e^{(m - |\mathbf{i}_{\bar{h}}|) \cdot 2(\overline{R}_S(\bar{h}) - \overline{R}_D(\bar{h}))^2} . \tag{6.48}$$

Now, again, as in the proof of Theorem 6.3.1, by Markov's inequality we have

$$\Pr_{S \sim D^m}\left(X_{\overline{P^S}} \leq \frac{1}{\delta} \mathop{\mathbf{E}}_{S \sim D^m} X_{\overline{P^S}} \right) \geq 1 - \delta \, .$$

Thus, by applying the claim and by taking the logarithm on each side of the innermost inequality, we obtain

$$\Pr_{S \sim D^m}\left(\begin{array}{l} \forall Q \textit{ aligned on } P^S : \\ \ln\left[\mathop{\mathbf{E}}_{\bar{h} \sim \overline{Q}} e^{(m - |\mathbf{i}_{\bar{h}}|) \cdot 2(\overline{R}_S(\bar{h}) - \overline{R}_D(\bar{h}))^2} \right] \ \leq \ \ln\left[\frac{1}{\delta} \mathop{\mathbf{E}}_{S \sim D^m} X_{\overline{P^S}} \right] \end{array} \right) \ \geq \ 1 - \delta \, . \tag{6.49}$$

Jensen's inequality applied to the concave $\ln(x)$ gives

$$\ln\left[\mathop{\mathbf{E}}_{\bar{h} \sim \overline{Q}} e^{(m - |\mathbf{i}_{\bar{h}}|) \cdot 2(\overline{R}_S(\bar{h}) - \overline{R}_D(\bar{h}))^2} \right] \geq \mathop{\mathbf{E}}_{\bar{h} \sim \overline{Q}} (m - |\mathbf{i}_{\bar{h}}|) \cdot 2(\overline{R}_S(\bar{h}) - \overline{R}_D(\bar{h}))^2 \, . \tag{6.50}$$

Again from the Jensen's inequality, applied to the convex function $\mathcal{D}(q,p) = (q-p)^2$, together with the definition of ζ_D^Q and ζ_S^Q (see Equation (6.12)) and the fact that $m - |\mathbf{i}_{\bar{h}}| \geq m - l \cdot d$, we obtain:

$$\mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} (m - |\mathbf{i}_{\bar{h}}|)2(\overline{R}_S(\bar{h}) - \overline{R}_D(\bar{h}))^2 \ \geq\ (m-ld)2\left(\mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} \overline{R}_S(\bar{h}) - \mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} \overline{R}_D(\bar{h})\right)^2 \tag{6.51}$$

Thus, from Equations (6.49) to (6.51), we obtain:

$$\Pr_{S\sim D^m}\left(\begin{array}{l} \forall Q \textit{ aligned on } P^S : \\ (m-ld)\cdot 2\left(\mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} \overline{R}_S(\bar{h}) - \mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} \overline{R}_D(\bar{h})\right)^2 \ \leq\ \ln\left[\frac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m} X_{\overline{P^S}}\right] \end{array}\right) \ \geq\ 1-\delta\,. \tag{6.52}$$

Let us now analyze the value of $\mathbf{E}_{S\sim D^m} X_{\overline{P^S}}$.

For this, we have to use the *abstract empirical risk* $\widetilde{R}_S(\bar{h})$ as given by Equation (6.8).

Now, let us show that,

$$\mathbf{E}_{S\sim D^m} X_{\overline{P^S}} \ \leq\ e^{4ld}\cdot 2\sqrt{m}\,. \tag{6.53}$$

First note that, since

$$\begin{aligned} \mathbf{E}_{S\sim D^m} X_{\overline{P^S}} \ &\stackrel{\text{def}}{=}\ \mathop{\mathbf{E}}_{S\sim D^m}\ \mathop{\mathbf{E}}_{\bar{h}\sim\overline{P^S}} e^{(m-|\mathbf{i}_{\bar{h}}|)2(\overline{R}_S(\bar{h})-\overline{R}_D(\bar{h}))^2} \\ &=\ \mathop{\mathbf{E}}_{\mathbf{i}\sim\overline{P}_{\mathcal{I}}}\ \mathop{\mathbf{E}}_{S_{\mathbf{i}}\sim D^{|\mathbf{i}|}}\ \mathop{\mathbf{E}}_{\bar{\mu}\sim\overline{P}_{S_{\mathbf{i}}}}\ \mathop{\mathbf{E}}_{S_{\mathbf{i}^c}\sim D^{m-|\mathbf{i}|}} e^{(m-|\mathbf{i}|)2\,(\overline{R}_S(h_{S_{\mathbf{i}}}^{\bar{\mu}})-\overline{R}_D(h_{S_{\mathbf{i}}}^{\bar{\mu}}))^2}\,, \end{aligned}$$

to prove Equation (6.53), it suffices to show that we have

$$\mathop{\mathbf{E}}_{S_{\mathbf{i}^c}\sim D^{m-|\mathbf{i}|}} e^{(m-|\mathbf{i}|)2\,(\overline{R}_S(h_{S_{\mathbf{i}}}^{\bar{\mu}})-\overline{R}_D(h_{S_{\mathbf{i}}}^{\bar{\mu}}))^2} \ \leq\ e^{4ld}\cdot 2\sqrt{m}\,, \tag{6.54}$$

for all $\mathbf{i}\in\mathcal{I}$, $S_{\mathbf{i}}\in(\mathcal{X}\times\mathcal{Y})^{|\mathbf{i}|}$, and $\bar{\mu}\in\mathcal{M}_{S_{\mathbf{i}}}$. Here is the sketch of the proof of

Equation (6.54). Justification for Line (6.55) to (6.58) follows below.

$$
\begin{aligned}
&\mathop{\mathbf{E}}_{S_{\mathbf{i}^c}\sim D^{m-|\mathbf{i}|}} e^{(m-|\mathbf{i}|)\cdot 2\,(\overline{R}_S(h^{\bar{\mu}}_{S_\mathbf{i}})-\overline{R}_D(h^{\bar{\mu}}_{S_\mathbf{i}}))^2} \\
&\quad= \mathop{\mathbf{E}}_{S_{\mathbf{i}^c}\sim D^{(m-|\mathbf{i}|)}} e^{(m-|\mathbf{i}|)\cdot 2\left(\overline{R}_S(h^{\bar{\mu}}_{S_\mathbf{i}})-\widetilde{R}_S(h^{\bar{\mu}}_{S_\mathbf{i}})+\widetilde{R}_S(h^{\bar{\mu}}_{S_\mathbf{i}})-\overline{R}_D(h^{\bar{\mu}}_{S_\mathbf{i}})\right)^2} \\
&\quad\le \mathop{\mathbf{E}}_{S_{\mathbf{i}^c}\sim D^{(m-|\mathbf{i}|)}} e^{(m-|\mathbf{i}|)\cdot 2\left(\left[\overline{R}_S(h^{\bar{\mu}}_{S_\mathbf{i}})-\widetilde{R}_S(h^{\bar{\mu}}_{S_\mathbf{i}})\right]^2+2\left|\overline{R}_S(h^{\bar{\mu}}_{S_\mathbf{i}})-\widetilde{R}_S(h^{\bar{\mu}}_{S_\mathbf{i}})\right|\cdot\left|\widetilde{R}_S(h^{\bar{\mu}}_{S_\mathbf{i}})-\overline{R}_D(h^{\bar{\mu}}_{S_\mathbf{i}})\right|+\left[\widetilde{R}_S(h^{\bar{\mu}}_{S_\mathbf{i}})-\overline{R}_D(h^{\bar{\mu}}_{S_\mathbf{i}})\right]^2\right)} \\
&\quad\le \mathop{\mathbf{E}}_{S_{\mathbf{i}^c}\sim D^{(m-|\mathbf{i}|)}} e^{(m-|\mathbf{i}|)\cdot 2\left(\left[\frac{|\mathbf{i}|}{m}\right]^2+2\frac{|\mathbf{i}|}{m}\cdot 1+\left[\widetilde{R}_S(h^{\bar{\mu}}_{S_\mathbf{i}})-\overline{R}_D(h^{\bar{\mu}}_{S_\mathbf{i}})\right]^2\right)} && (6.55) \\
&\quad\le \mathop{\mathbf{E}}_{S_{\mathbf{i}^c}\sim D^{(m-|\mathbf{i}|)}} e^{4ld\,+\,(m-|\mathbf{i}|)\cdot 2\left[\widetilde{R}_S(h^{\bar{\mu}}_{S_\mathbf{i}})-\overline{R}_D(h^{\bar{\mu}}_{S_\mathbf{i}})\right]^2} && (6.56) \\
&\quad\le \mathop{\mathbf{E}}_{S_{\mathbf{i}^c}\sim D^{(m-|\mathbf{i}|)}} e^{4ld\,+\,(m-|\mathbf{i}|)\cdot\mathrm{kl}\left(\widetilde{R}_S(h^{\bar{\mu}}_{S_\mathbf{i}})\|\overline{R}_D(h^{\bar{\mu}}_{S_\mathbf{i}})\right)} && (6.57) \\
&\quad= e^{4ld}\cdot \mathop{\mathbf{E}}_{S_{\mathbf{i}^c}\sim D^{(m-|\mathbf{i}|)}} e^{(m-|\mathbf{i}|)\cdot\mathrm{kl}\left(\widetilde{R}_S(h^{\bar{\mu}}_{S_\mathbf{i}})\|\overline{R}_D(h^{\bar{\mu}}_{S_\mathbf{i}})\right)} \\
&\quad\le e^{4ld}\cdot 2\sqrt{m-|\mathbf{i}|} && (6.58) \\
&\quad\le e^{4ld}\cdot 2\sqrt{m}
\end{aligned}
$$

Line (6.55) follows from Equation (6.10) and the fact that the exponential function is increasing. For Line (6.56) we have :

$$(m-|\mathbf{i}|)2\left(\frac{|\mathbf{i}|^2}{m^2}+\frac{2|\mathbf{i}|}{m}\right) = |\mathbf{i}|(m-|\mathbf{i}|)2\left(\frac{|\mathbf{i}|}{m^2}+\frac{2}{m}\right).$$

From $|\mathbf{i}|\le ld\le m$, it follows that $m-|\mathbf{i}|\le m$ and $\frac{|\mathbf{i}|}{m}\le 1$. Thus, we have:

$$(m-|\mathbf{i}|)2\left(\frac{|\mathbf{i}|}{m^2}+\frac{2}{m}\right)\le m2\left(\frac{|\mathbf{i}|}{m^2}+\frac{2}{m}\right)\le 2\frac{|\mathbf{i}|}{m}+2\le 4.$$

Therefore, we have:

$$(m-|\mathbf{i}|)2\left(\frac{|\mathbf{i}|^2}{m^2}+\frac{2|\mathbf{i}|}{m}\right) = |\mathbf{i}|(m-|\mathbf{i}|)2\left(\frac{|\mathbf{i}|}{m^2}+\frac{2}{m}\right)\le 4|\mathbf{i}|\le 4ld.$$

Line (6.57) follows directly from the property : $2(q-p)^2\le \mathrm{kl}(q\parallel p)$ (Pinsker's inequality [13]). Finally, for Line (6.58), first observe that $\widetilde{R}_S(h^{\bar{\mu}}_{S_\mathbf{i}})$ is an arithmetic mean of $(m-|\mathbf{i}|)$ iid random variables. Thus Line (6.58) is simply an application of

Lemma 6.2.1 with $M(X)$ replaced by $\widetilde{R}_S(h^{\bar{\mu}}_{S_{\mathbf{i}}})$, n replaced by $m - |\mathbf{i}|$, and ν replaced by $\overline{R}_D(h^{\bar{\mu}}_{S_{\mathbf{i}}})$. Thus, Equation (6.53) is proved.

By using Equations (6.12), (6.24), and (6.53) in Equation (6.52) we have:

$$(m-ld)2\left(\frac{1}{2}\left[1+\frac{1}{\zeta(1)}\;\zeta_S^Q\right]-\frac{1}{2}\left[1+\frac{1}{\zeta(1)}\;\zeta_D^Q\right]\right)^2 \le \ln\left[\frac{1}{\delta}\cdot e^{4ld}\cdot 2\sqrt{m}\right]$$

By rearranging the above equations we obtain:

$$\zeta_D^Q \le \zeta_S^Q + \frac{\zeta(1)}{\sqrt{\frac{1}{2}(m - l\,\mathrm{deg}(\zeta))}}\sqrt{4\,l\,\mathrm{deg}(\zeta)+\ln\frac{2\sqrt{m}}{\delta}}$$

□

In the particular case of non-sample compressed classifiers (when $l = 0$) Theorem 6.4.4 reduces to the following corollary:

Corollary 6.4.5. *For any D, any $\mathcal{H}$ of sets of classifiers any prior $\mathcal{P}$, any $\delta \in (0,1]$, and any margin loss function we have:*

$$\Pr_{S\sim D^m}\left(\forall Q \text{ aligned on } P\ :\quad \zeta_D^Q \le \zeta_S^Q + \frac{\zeta(1)}{\sqrt{\frac{1}{2}m}}\sqrt{\ln\frac{2\sqrt{m}}{\delta}}\right) \ge 1-\delta$$

Seeger bound without KL

Theorem 6.4.6. *For any D, for any family $(\mathcal{H}^S)_{S\in D^m}$ of sets of sc-classifiers of size at most l, for any prior $\mathcal{P}$, for any margin loss function ζ such that $l\cdot\mathrm{deg}(\zeta) < m$, and for for any $\delta \in (0,1]$, we have*

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\forall Q \text{ aligned on } P^S\ :\\ \quad\mathrm{kl}^+\left(\frac{m}{m-l\cdot\mathrm{deg}(\zeta)}\,(\frac{1}{2}\left[1+\frac{1}{\zeta(1)}\;\zeta_S^Q\right]+\frac{ld}{m})\;\|\;\frac{1}{2}\left[1+\frac{1}{\zeta(1)}\;\zeta_D^Q\right]\right)\le\frac{\ln\frac{2\sqrt{m}}{\delta}}{m-l\cdot\mathrm{deg}(\zeta)}\end{array}\right)\ge 1-\delta$$

where $\mathrm{kl}(q\|p) \overset{def}{=} q\ln\frac{q}{p} + (1-q)\ln\frac{1-q}{1-p}$, and where $\mathrm{kl}^+(q\|p) = \mathrm{kl}(q\|p)$ if $q \le p$ and 0 otherwise. Moreover, if $l = 0$, the function kl^+ can be replace by the function kl in the statement, giving rise to both a lower and an upper bound of ζ_D^Q.

Proof. The first part of the proof is very similar to the one of Theorem 6.4.4. We again define d, $\bar{h} = \overline{h_1..h_k}$ (with $k \in \{0,..,d\}$), $\overline{R}_D(\bar{h})$, $\overline{R}_S(\bar{h})$, $\overline{\mathcal{H}^S}$, $\overline{P^S}$, $\overline{Q}$, ζ_D^Q, and ζ_S^Q. However, we will instead consider this quite different random variable that among other thing is not based on $\overline{R}_S(\bar{h})$ but on a slightly different value $\widetilde{R}_S(\bar{h})$ given by Equation (6.8). Note that in the case of the McAllester bound (Theorem 6.4.4) , we manage to deal with the bias that comes from the examples of the compression set in the calculation of the random variable $X_{\overline{P^S}}$ (see Equations (6.55) to (6.58)). In the case of Seeger bound, this is no longer possible. In the presence of such bias, the random variable $X_{\overline{P^S}}$ using the kl as the divergence $\mathcal{D}$ can be huge. For this reason, contrarily to the preceding proof, in this proof, we consider the unbias $\widetilde{R}_S(\bar{h})$ instead of $\overline{R}_S(\bar{h})$ in the definition of the random variable $X_{\overline{P^S}}$[2]. Therefore, we consider the following random variable which is based on the value $\widetilde{R}_S(\bar{h})$.

$$X_{\overline{P^S}} \stackrel{\text{def}}{=} \mathop{\mathbf{E}}_{\bar{h}\sim\overline{p^S}} e^{(m-|\mathbf{i}_{\bar{h}}|)\mathrm{kl}(\widetilde{R}_S(\bar{h}),\overline{R}_D(\bar{h}))}\,, \tag{6.59}$$

By the same proof as in Lemma 6.4.3, except that everywhere in the proof, $\overline{R}_S$ is replaced by $\widetilde{R}_S$, and $D(\overline{R}_S(\bar{h})||\overline{R}_D(\bar{h}))^2$ is replaced by $\mathrm{kl}(\widetilde{R}_S(\bar{h}),\overline{R}_D(\bar{h}))$, one can show that : for any posterior Q aligned on P^S, we have

$$X_{\overline{P^S}} \;=\; \mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} e^{(m-|\mathbf{i}_{\bar{h}}|)\mathrm{kl}(\widetilde{R}_S(\bar{h}),\overline{R}_D(\bar{h}))}. \tag{6.60}$$

Now, again, as in the proof of Theorem 6.3.3, by Markov's inequality we have

$$\Pr_{S\sim D^m}\left(X_{\overline{P^S}} \le \frac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m} X_{\overline{P^S}}\right) \ge 1-\delta\,.$$

Thus, by applying the claim and by taking the logarithm on each side of the innermost inequality, we can obtain

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\forall Q \textit{ aligned on } P^S: \\ \ln\left[\mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} e^{(m-|\mathbf{i}_{\bar{h}}|)\mathrm{kl}(\widetilde{R}_S(\bar{h})\|\overline{R}_D(\bar{h}))}\right] \le \ln\left[\frac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{\bar{h}\sim\overline{P^S}} e^{(m-|\mathbf{i}_{\bar{h}}|)\mathrm{kl}(\widetilde{R}_S(\bar{h})\|\overline{R}_D(\bar{h}))}\right]\end{array}\right) \ge 1-\delta\,. \tag{6.61}$$

[2] This is the main reason why the mechanism of the proof of Theorems 6.4.4 and 6.4.7 seems so different, although in the non sample compression case where $\widetilde{R}_S(\bar{h}) = \overline{R}_S(\bar{h})$, the proof of these theorems are very similar (See Chapter 5, Section 5.1.2).

Jensen's inequality applied to the concave $\ln(x)$ gives

$$\ln\left[\mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} e^{(m-|\mathbf{i}_{\bar{h}}|)\mathrm{kl}(\widetilde{R}_S(\bar{h})\|\overline{R}_D(\bar{h}))}\right] \geq \mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} (m-|\mathbf{i}_{\bar{h}}|)\,\mathrm{kl}(\widetilde{R}_S(\bar{h})\|\overline{R}_D(\bar{h})\,. \tag{6.62}$$

Again from the Jensen's inequality, applied to the convex function $\mathrm{kl}(\cdot\|\cdot)$, together with the definition of ζ_D^Q and ζ_S^Q (see Equation (6.12)) and the fact that $m-|\mathbf{i}_{\bar{h}}| \geq m-l\cdot d$ we obtain:

$$\mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} (m-|\mathbf{i}_{\bar{h}}|)\mathrm{kl}\left(\widetilde{R}_S(\bar{h}) \parallel \overline{R}_D(\bar{h})\right) \;\geq\; (m-ld)\mathrm{kl}\left(\mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}}\widetilde{R}_S(\bar{h}) \parallel \mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}} \overline{R}_D(\bar{h})\right) \tag{6.63}$$

Let us now analyse the value of $\mathbf{E}_{S\sim D^m}X_{\overline{P^S}}$.

As in the preceding proof, let $\mathbf{i}^c$ be the vector of indices of $\mathcal{I}$ that are not in the vector $\mathbf{i}$, and note that

$$\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{\bar{h}\sim\overline{P^S}} e^{(m-|\mathbf{i}_{\bar{h}}|)\,\mathrm{kl}(\widetilde{R}_S(\bar{h}),\overline{R}_D(\bar{h}))} \;=\; \mathop{\mathbf{E}}_{\mathbf{i}\sim\overline{P}_{\mathcal{I}}}\;\mathop{\mathbf{E}}_{S_{\mathbf{i}}\sim D^{|\mathbf{i}|}}\;\mathop{\mathbf{E}}_{\bar{\mu}\sim\overline{P}_{S_{\mathbf{i}}}}\;\mathop{\mathbf{E}}_{S_{\mathbf{i}^c}\sim D^{m-|\mathbf{i}|}} e^{|\mathbf{i}^c|\,\mathrm{kl}(\widetilde{R}_S(h^{\bar{\mu}}_{S_{\mathbf{i}}}),\overline{R}_D(h^{\bar{\mu}}_{S_{\mathbf{i}}}))}$$

Since $\widetilde{R}_S(h^{\bar{\mu}}_{S_{\mathbf{i}}})$ is an arithmetic mean of iid random variables, one can apply Lemma 6.2.1 with $M(X)$ replaced by $\widetilde{R}_S(h^{\bar{\mu}}_{S_{\mathbf{i}}})$, n replaced by $m-|\mathbf{i}|$, and ν replaced by $\overline{R}_D(h^{\bar{\mu}}_{S_{\mathbf{i}}})$ to obtain:

$$\mathbf{E}_{S_{\mathbf{i}^c}\sim D^{m-|\mathbf{i}|}}\; e^{(m-|\mathbf{i}|)\,\mathrm{kl}(\widetilde{R}_S(h^{\bar{\mu}}_{S_{\mathbf{i}}}),\overline{R}_D(h^{\bar{\mu}}_{S_{\mathbf{i}}}))} \;\leq\; 2\sqrt{m-|\mathbf{i}|} \;\leq\; 2\sqrt{m}\,. \tag{6.64}$$

By rearranging Equation (6.61) based on Equation (6.64) and using Equations (6.62) and (6.63), we have:

$$\mathop{\mathbf{E}}_{S\sim D^m} (m-ld)\;\; \mathrm{kl}\left(\mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}}\widetilde{R}_S(\bar{h}) \parallel \mathop{\mathbf{E}}_{\bar{h}\sim\overline{Q}}\overline{R}_D(\bar{h})\right) \;\leq\; \ln\frac{2\sqrt{m}}{\delta} \tag{6.65}$$

Finally, observe that for any classifier $\bar{h}\in\overline{\mathcal{H}^S}$, we have

$$\begin{aligned}
\widetilde{R}_S(\bar{h}) \;&\leq\; \left(\overline{R}_S(\bar{h})+\frac{ld}{m}\right)\frac{m}{m-|\mathbf{i}|}\\
&\leq\; \left(\overline{R}_S(\bar{h})+\frac{ld}{m}\right)\frac{m}{m-ld}\\
&\leq\; \frac{m}{m-ld}\overline{R}_S(\bar{h})+\frac{ld}{m-ld}\\
&=\; \frac{m}{m-ld}\left(\overline{R}_S(\bar{h})+\frac{ld}{m}\right)\,,
\end{aligned} \tag{6.66}$$

and consider the following two cases.

case 1 : $l = 0$. In that case we have that $\mathop{\mathbf{E}}_{\bar h \sim \overline{Q}} \widetilde{R}_S(\bar h) = \mathop{\mathbf{E}}_{\bar h \sim \overline{Q}} \overline{R}_S(\bar h)$. Hence we have

$$\mathrm{kl}\left(\mathop{\mathbf{E}}_{\bar h \sim \overline{Q}} \widetilde{R}_S(\bar h) \,\|\, \mathop{\mathbf{E}}_{\bar h \sim \overline{Q}} \overline{R}_D(\bar h)\right) \;=\; \mathrm{kl}\left(\frac{m}{m-ld} \mathop{\mathbf{E}}_{\bar h \sim \overline{Q}} \overline{R}_S(\bar h) + \frac{ld}{m} \,\|\, \mathop{\mathbf{E}}_{\bar h \sim \overline{Q}} \overline{R}_D(\bar h)\right)$$

case 2 : $l > 0$. In that case, following Equation (6.66), we have:

$$\mathrm{kl}\left(\mathop{\mathbf{E}}_{\bar h \sim \overline{Q}} \widetilde{R}_S(\bar h) \,\|\, \mathop{\mathbf{E}}_{\bar h \sim \overline{Q}} \overline{R}_D(\bar h)\right) \;\geq\; \mathrm{kl}^{+}\left(\frac{m}{m-ld} \mathop{\mathbf{E}}_{\bar h \sim \overline{Q}} \overline{R}_S(\bar h) + \frac{ld}{m} \,\|\, \mathop{\mathbf{E}}_{\bar h \sim \overline{Q}} \overline{R}_D(\bar h)\right)$$

In each case, the result then follows from Equation (6.65):

$$(m-ld)\;\; \mathrm{kl}^{+}\left(\frac{m}{m-ld} \mathop{\mathbf{E}}_{\bar h \sim \overline{Q}} \overline{R}_S(\bar h) + \frac{ld}{m} \,\|\, \mathop{\mathbf{E}}_{\bar h \sim \overline{Q}} \overline{R}_D(\bar h)\right) \;\leq\; \ln \frac{2\sqrt{m}}{\delta}$$

Now, by replacing $\mathop{\mathbf{E}}_{\bar h \sim \overline{Q}} \overline{R}_U(\bar h)$ by $\frac{1}{2}\left[1 + \frac{1}{\zeta(1)}\; \zeta_U^Q\right]$ from Equation (6.12) for $U = D$ and $U = S$ we get:

$$\mathrm{kl}^{+}\left(\frac{m}{m - l\cdot \deg(\zeta)} \Big(\frac{1}{2}\left[1 + \frac{1}{\zeta(1)}\; \zeta_S^Q\right] + \frac{ld}{m}\Big) \,\|\, \frac{1}{2}\left[1 + \frac{1}{\zeta(1)}\; \zeta_D^Q\right]\right) \leq \frac{\ln \frac{2\sqrt{m}}{\delta}}{m - l\cdot \deg(\zeta)}$$

□

In the particular case of non-sample compressed classifiers (when $l = 0$) Theorem 6.4.7 reduces to the following corollary:

Corollary 6.4.7. *For any D, any $\mathcal{H}$ of sets of classifiers any prior $\mathcal{P}$, any $\delta \in (0,1]$, any positive real number C_1, and any margin loss function we have:*

$$\Pr_{S\sim D^m}\begin{pmatrix} \forall Q \text{ aligned on } P\;: \\ \quad \mathrm{kl}\left(\frac{1}{2}\left[1 + \frac{1}{\zeta(1)}\; \zeta_S^Q\right] \| \,\frac{1}{2}\left[1 + \frac{1}{\zeta(1)}\; \zeta_D^Q\right]\right) \;\leq\; \frac{\ln \frac{2\sqrt{m}}{\delta}}{m} \end{pmatrix} \geq 1 - \delta$$

where $\mathrm{kl}(q\|p) \overset{def}{=} q \ln \frac{q}{p} + (1-q) \ln \frac{1-q}{1-p}$.

Chapter 7

A general Sample compressed PAC-Baysian Approach to Kernel Methods.

In this chapter, we propose a PAC-Bayes sample-compression approach to kernel methods that can generalize to any bounded similarity function. We also show that the support vector machine (SVM) is a particular case of a more general class of data-dependent classifiers known as majority votes of sample-compressed classifiers. We then apply the risk bounds proposed in Chapter 6 to these majority votes and provide new algorithms that consist of minimizing these risk bounds. Empirically, we observe that the learning algorithms obtained in this way are very competitive when compared with the state-of-the-art algorithms.

7.1 Specialization to Majority Votes of Sc-classifiers of Compression Size of at Most One:

Here, we specialize ourselves to the case where each sc-classifier has a compression set size of at most one. In that case, each sample compression sequence $S_{\mathbf{i}}$ consists of at most a single training example and, consequently, each possible vector $\mathbf{i}$ has at most only one index (*i.e.*, $|\mathbf{i}| \leq 1$). When $|\mathbf{i}| = 1$ and its single index points to example

(x_i, y_i) of S, we have $\mathbf{i} = \langle i \rangle$ and $S_{\mathbf{i}} = S_{\langle i \rangle} = (x_i, y_i)$. When $|\mathbf{i}| = 0$, then $\mathbf{i} = \langle\rangle$ and $S_{\mathbf{i}} = S_{\langle\rangle} = \emptyset$. In this latter case, we will only consider the two constant sc-classifier $h_{S_{\langle\rangle}}^{(\varepsilon,+)}$ and $h_{S_{\langle\rangle}}^{(\varepsilon,-)}$ where $h_{S_{\langle\rangle}}^{(\varepsilon,+)}(\mathbf{x}) = +1$ and $h_{S_{\langle\rangle}}^{(\varepsilon,-)}(\mathbf{x}) = -1$ for all $\mathbf{x} \in \mathcal{X}$. Here ε denotes the empty message. As presented in Section 6.4.1, for each compression set $S_{\mathbf{i}}$ the message sets are of the form

$$\mathcal{M}_{S_{\mathbf{i}}} \;=\; \mathcal{M}^1_{S_{\mathbf{i}}} \times \{+,-\}\,,$$

In particular case where each sc-classifier has a compression set size of at most one a message is defined in the following way:

- $\mathcal{M}^1_{S_{\langle i \rangle}}$ is a real interval and we denote it by $\mathcal{M}^1$.
- $\mathcal{M}^1_{S_{\langle\rangle}} = \{\varepsilon\}$
- $\mathcal{M}_{S_{\mathbf{i}}} = \emptyset$ for all $\mathbf{i} \geq 2$.

Therefore, each sc-classifiers $h_{S_{\langle i \rangle}}^{(\sigma,s)}$ (that we will define below) of compression size 1 uses a message $(\sigma, s) \in \mathcal{M}^1 \times \{-,+\}$ where $\mathcal{M}^1$ is a real interval having a length denoted by $|\mathcal{M}^1|$.

We will see later (in Section 7.1.2), that this set of sc-classifiers can include SVM classifier.

7.1.1 The Choice of Prior and Posterior

Uniform priors:
Recall that in Chapter 5, we only considered the prior distribution P^S on $\mathcal{I} \times \mathcal{M}_{S_{\mathbf{i}}}$ in the sample-compression setting that can be written as: $P^S(h^{\mu}_{S_{\mathbf{i}}}) = P_{\mathcal{I}}(\mathbf{i})\, P_{S_{\mathbf{i}}}(\mu)$. (see Equation (5.7)). In this section, we use a uniform prior over all relevant parameters. More precisely, for all $i \in \{1, \ldots, m\}$, and $s \in \{-,+\}$, we have

$$P_{\mathcal{I}}(\langle\rangle) = P_{\mathcal{I}}(\langle i \rangle) = \frac{1}{m+1} \quad ; \quad P_{S_{\langle\rangle}}(\varepsilon, s) = \frac{1}{2} \quad ; \quad P_{S_{\langle i \rangle}}(\sigma, s) = \frac{1}{2|\mathcal{M}^1|} I(\sigma \in \mathcal{M}^1)\,. \tag{7.1}$$

Equation (5.7) implies that $P^S(h_{S_{\langle\rangle}}^{(\varepsilon,s)}) = P_{\mathcal{I}}(\langle\rangle)P_{S_{\langle\rangle}}(\varepsilon, s)$ and $P^S(h_{S_{\langle i\rangle}}^{(\sigma,s)}) = P_{\mathcal{I}}(\langle i\rangle)P_{S_{\langle i\rangle}}(\sigma, s)$. This way, we have:

$$\begin{array}{ll} P^S(h_{S_{\langle\rangle}}^{(\varepsilon,+)}) = \frac{1}{2m+2}, & P^S(h_{S_{\langle i\rangle}}^{(\sigma,+)}) = \frac{1}{2m+2}\frac{1}{|\mathcal{M}^1|}, \\ P^S(h_{S_{\langle\rangle}}^{(\varepsilon,-)}) = \frac{1}{2m+2}, & P^S(h_{S_{\langle i\rangle}}^{(\sigma,-)}) = \frac{1}{2m+2}\frac{1}{|\mathcal{M}^1|}, \end{array} \tag{7.2}$$

for any $i \in \{1, .., m\}$ and any $\sigma \in \mathcal{M}^1$.

We can then easily verify that we have

$$\sum_{s\in\{-,+\}} P^S(h_{S_{\langle\rangle}}^{(\varepsilon,s)}) + \sum_{i=1}^{m} \sum_{s\in\{-,+\}} \int_{\mathcal{M}^1} d\sigma P^S(h_{S_{\langle i\rangle}}^{(\sigma,s)}) \ = \ 1\,.$$

We saw earlier in Chapter 5, that we considered the posteriors that are written as: $Q(h_{S_{\mathbf{i}}}^{\mu}) = Q_{\mathcal{I}}(\mathbf{i})Q_{S_{\mathbf{i}}}(\mu)$ (See Equation (5.9)).

Uniform on messages posteriors:

Here, we restrict ourselves to the posteriors called *uniform on messages*. In other words, we restrict ourselves to a posterior distribution such that for any compression set $\mathbf{i}$, $Q_{S_{\mathbf{i}}}(\mu)$ is a function that does not depend on the σ part of the message (σ, s). More formally, the posterior Q will be defined as follows. A posterior Q is said to be *uniform on messages* if there exists $\mathbf{v} \overset{\text{def}}{=} (v_+, v_1, \ldots, v_{2m}, v_-)$, where $v \geq 0$ for all $v \in \mathbf{v}$, $\sum_{v\in\mathbf{v}} v = 1$, and such that:

$$\begin{array}{ll} Q(h_{S_{\langle\rangle}}^{(\varepsilon,+)}) = v_+, & Q(h_{S_{\langle i\rangle}}^{(\sigma,+)}) = v_i\frac{1}{|\mathcal{M}^1|}, \\ Q(h_{S_{\langle\rangle}}^{(\varepsilon,-)}) = v_-, & Q(h_{S_{\langle i\rangle}}^{(\sigma,-)}) = v_{m+i}\frac{1}{|\mathcal{M}^1|}, \end{array} \tag{7.3}$$

for any $i \in \{1, .., m\}$ and any $\sigma \in \mathcal{M}^1$.

A reason for this restriction is to simplify the computations of the posterior that minimizes our PAC-Bayes bounds. Indeed, as shown above the posterior is fully determined by the values $(v_+, v_1, \ldots, v_{2m}, v_-)$. Moreover, under this restriction, the set of these posteriors remains strong enough to include kernel methods such as the SVM as a special case, even if we also impose the posterior to be aligned to the prior (see Section 7.1.2).

Aligned and uniform on messages posteriors:

Encouraged by the theoretical results we obtained in the preceding section, we consider the case where the posterior is also aligned to the prior. In this case, an algorithm that minimizes a PAC-Bayes bound would not have to consider any Kullback-Leibler divergence term. Recall that, in Section 6.4.1 (Equation (6.42)) the aligned posteriors are defined as follows.

For the sc-classifiers of compression size zero, we have:

$$Q(h_{S_{\langle\rangle}}^{(\varepsilon,s)}) = \frac{1}{2}\left(P^S(h_{S_{\langle\rangle}}^{(\varepsilon,+)}) + P^S(h_{S_{\langle\rangle}}^{(\varepsilon,-)}) + s\cdot w(\langle\rangle,\varepsilon)\right) \tag{7.4}$$

For the sc-classifiers of compression size one, we have:

$$Q(h_{S_{\langle i\rangle}}^{(\sigma,s)}) = \frac{1}{2}\left(P^S(h_{S_{\langle i\rangle}}^{(\sigma,+)}) + P^S(h_{S_{\langle i\rangle}}^{(\sigma,-)}) + s\cdot w(\langle i\rangle,\sigma)\right) \tag{7.5}$$

It is easy to see that to obtain an aligned posterior that is uniform on the messages, one only needs to restrict the $w(\mathbf{i},\sigma)$ to the following form:

$$\begin{aligned} w(\langle i\rangle,\sigma) &= w_i\frac{1}{|\mathcal{M}^1|}I(\sigma\in\mathcal{M}^1)\ \textit{where}\ |w_i| \le \frac{1}{m+1} & (7.6)\\ w(\langle\rangle,\varepsilon) &\stackrel{\text{def}}{=} w_0\ \textit{where}\ |w_0| \le \frac{1}{m+1} & (7.7)\end{aligned}$$

Now, from Equation (7.2), we have:

$$P^S(h_{S_{\langle\rangle}}^{(\varepsilon,+)}) + P^S(h_{S_{\langle\rangle}}^{(\varepsilon,-)}) = \frac{1}{m+1},\quad P^S(h_{S_{\langle i\rangle}}^{(\sigma,+)}) + P^S(h_{S_{\langle i\rangle}}^{(\sigma,-)}) = \frac{1}{|\mathcal{M}^1|(m+1)}$$

Therefore, we get:

$$\begin{aligned} Q(h_{S_{\langle\rangle}}^{(\varepsilon,s)}) &= \frac{1}{2}\left(P^S(h_{S_{\langle\rangle}}^{(\varepsilon,+)}) + P^S(h_{S_{\langle\rangle}}^{(\varepsilon,-)}) + s\cdot w(\langle\rangle,\varepsilon)\right)\\ &= \frac{1}{2}\left(\frac{1}{m+1} + s\cdot w_0\right), & (7.8)\end{aligned}$$

where $s\in\{+,-\}$, $w_0 \stackrel{\text{def}}{=} w(\langle\rangle,\varepsilon)$ and must satisfy $|w_0| \le \frac{1}{m+1}$

and

$$\begin{aligned} Q(h_{S_{\langle i\rangle}}^{(\sigma,s)}) &= \frac{1}{2}\left(P^S(h_{S_{\langle i\rangle}}^{(\sigma,+)}) + P^S(h_{S_{\langle i\rangle}}^{(\sigma,-)}) + s\cdot w(\langle i\rangle,\sigma)\right)\\ &= \frac{1}{2}\left(\frac{1}{m+1} + s\cdot w_i\right)\frac{1}{|\mathcal{M}^1|}, & (7.9)\end{aligned}$$

where we have defined w_i by the equality $w(\langle i\rangle, \sigma) = w_i \frac{1}{|\mathcal{M}^1|}$, and where $(\sigma, s) \in \mathcal{M}^1 \times \{+, -\}$. Note that we must always satisfy $|w_i| \leq \frac{1}{m+1}$. Hence, similar to the uniform on messages case, aligned and uniform on messages posteriors are totally defined by a finite number of values, namely $(w_0, w_1, ..., w_m)$, provided that each w_i belongs to the interval $[\frac{-1}{m+1}, \frac{1}{m+1}]$. Note that in the uniform on messages case, we need $2m+2$ of such values, which is twice the number of values needed here. This smaller number of values provides a computational advantage to an algorithm based on minimizing a PAC-Bayes bound in the aligned setting. We will also see other advantages of that setting later in this chapter.

7.1.2 SVM as a Special Case:

The specialization to uniform on messages and aligned posterior might seem too restrictive. However, let us show that this setting remains powerful enough to include kernel methods such as the SVM as a *special case*. Recall that the output of an SVM classifier, f_{SVM}, is always of the form:

$$f_{\text{SVM}}(\mathbf{x}) = \text{sgn}\Big(\textstyle\sum_{i=1}^m y_i \alpha_i k(\mathbf{x}_i, \mathbf{x}) + b\Big)\,, \tag{7.10}$$

where $b \in \mathbb{R}$ and $\alpha_i \geq 0$ for all i. Hence, we have to show that, for any set of values $(b, \alpha_1, \cdots, \alpha_m)$ defining a predictor f_{SVM}, there exist values $(w_0, w_1, .., w_m)$ defining an aligned and uniform on messages posterior Q such that

$$f_{SVM}(\mathbf{x}) \;=\; B_Q(\mathbf{x}) \qquad \text{for all } \mathbf{x} \in \mathcal{X}\,. \tag{7.11}$$

First, let us show that for any kernel k such that $k(\mathbf{x}, \mathbf{x}') \leq 1$ for all $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$, one can define a set of sc-classifiers $\mathcal{H}^S = h_{S_{\langle\rangle}}^{(\varepsilon,+)}, h_{S_{\langle\rangle}}^{(\varepsilon,-)} \cup h_{S_{\langle i\rangle}}^{(\sigma,s)} : \sigma \in \mathcal{M}^1 \wedge s \in \{+, -\}$ such that for any aligned and uniform on messages posterior, Q, the output of $B_Q(\mathbf{x})$, on any $\mathbf{x} \in \mathcal{X}$ is given by

$$B_Q(\mathbf{x}) = \text{sgn}(\mathop{\mathbf{E}}_{h\sim Q} h(x)) = \text{sgn}\left(w_0 + \sum_{i=1}^m w_i k(\mathbf{x}_i, \mathbf{x})\right)\,, \tag{7.12}$$

To prove this, let us first note that in our setting:

$$\begin{aligned}
\mathop{\mathbf{E}}_{h\sim Q} h(\mathbf{x}) \;&=\; \sum_{s\in\{\text{-},+\}} Q(h_{S_{\langle\rangle}}^{(\varepsilon,s)}) h_{S_{\langle\rangle}}^{(\varepsilon,s)}(x) + \sum_{i=1}^m \sum_{s\in\{\text{-},+\}} \int_{\mathcal{M}^1} d\sigma Q(h_{S_{\langle i\rangle}}^{(\sigma,s)}) h_{S_{\langle i\rangle}}^{(\sigma,s)}(x) \\
&=\; w_0 + \sum_{i=1}^m w_i \frac{1}{|\mathcal{M}^1|} \int_{\mathcal{M}^1} h_{S_{\langle i\rangle}}^{(\sigma,+)}(x) d\sigma
\end{aligned} \tag{7.13}$$

The last equality is obtained by substituting $Q(h^{(\varepsilon,s)}_{S_{\langle\rangle}})$ with its value given by Equations (7.8) and (7.9), and by choosing $h^{(\varepsilon,+)}_{S_{\langle\rangle}}(\mathbf{x}) = 1$ and $h^{(\varepsilon,-)}_{S_{\langle\rangle}}(\mathbf{x}) = -1$ for the sc-classifiers that have compression set of size 0. This, therefore, implies that

$$\mathop{\mathbf{E}}_{h\sim Q} h(\mathbf{x}) = w_0 + \sum_{i=1}^{m} w_i k(\mathbf{x}_i, \mathbf{x}),$$

whenever one chooses both $h^{(\sigma,+)}_{S_{\langle i\rangle}}$ and $h^{(\sigma,-)}_{S_{\langle i\rangle}}$ in such a way that they satisfy the condition $\int_{\mathcal{M}^1} h^{(\sigma,+)}_{S_{\langle i\rangle}}(x)d\sigma = |\mathcal{M}^1|k(\mathbf{x}_i, \mathbf{x})$, for all i. One way to satisfy this condition is to choose:

$$h^{(\sigma,+)}_{S_{\langle i\rangle}}(\mathbf{x}) \; = \; \mathrm{sgn}\left(I(\frac{1}{2}|\mathcal{M}^1|k(\mathbf{x}_i, \mathbf{x}) > \sigma)\right) \;\; \forall \mathbf{x} \in \mathcal{X}\,, \tag{7.14}$$

$$h^{(\sigma,-)}_{S_{\langle i\rangle}}(\mathbf{x}) \; = \; \mathrm{sgn}\left(I(\frac{1}{2}|\mathcal{M}^1|k(\mathbf{x}_i, \mathbf{x}) \le \sigma)\right) \;\; \forall \mathbf{x} \in \mathcal{X} \tag{7.15}$$

with $\sigma \in \mathcal{M}^1 = [-1, +1]$ and with $k(\mathbf{x}', \mathbf{x}) \le 1 \;\forall(\mathbf{x}', \mathbf{x}) \in \mathcal{X}^2$ (note that, $I(a) = +1$ if predicate a is true and $I(a) = -1$ otherwise). This last condition implies that k must be bounded by 1. However, note that no other condition needs to be satisfied for k. Indeed, k can be any normalized similarity measure and does not even need to be symmetric in its two arguments.

Hence, if we compare the set of majority votes classifiers described by Equation (7.12) to the set of SVM classifiers where the output $f_{\text{SVM}}(\mathbf{x})$ for any $\mathbf{x} \in \mathcal{X}$ is given by Equation (7.10), we can conclude that the latter set of classifiers forms a strict subset of the former set. Indeed, even if for B_Q we must have $k(\mathbf{x}', \mathbf{x}) \le 1 \;\forall(\mathbf{x}', \mathbf{x}) \in \mathcal{X}^2$ and $|w_i| \le \frac{1}{m+1}$ for all i, while no such restriction exists for f_{SVM}, one can always choose $w_0 = \frac{b}{(m+1)Z}$ and $w_i = \frac{y_i \alpha_i}{(m+1)Z}$ where $Z \overset{\text{def}}{=} \sum_{i=1}^{m} \alpha_i + |b|$. Clearly, with these choices, we have that $f_{\text{SVM}}(\mathbf{x}) = B_Q(\mathbf{x}) \;\forall \mathbf{x} \in \mathcal{X}$. However, in our setting, k can be any similarity measure (possibly not symmetric in its two arguments), while k in f_{SVM} must be a positive semi-definite kernel [45].

Several generalizations from the above are possible. Indeed, for $Q(h^{(\sigma,s)}_{S_{\langle i\rangle}})$, we could consider distributions over σ that vary with i. This would effectively provide a mechanism for adapting the similarity measure to each training example. We could also use sc-classifiers having a compression size larger than one.

Finally, note that the risk bounds of Chapter 6 apply to this larger class of majority votes of sc-classifiers of compression size of at most one. In the following we presents how to apply these risk bounds to this class of majority votes of sc-classifiers of compression size of at most one.

7.1.3 Applying the Presented Risk Bounds in Chapter 6 to the Class of Majority Votes of Sc-classifiers of Compression Size of at Most One

As we saw earlier in Chapter 6, the presented risk bounds are based on the following elements: a training sequence S, a prior distribution P^S on $\mathcal{H}^S$ (a set of sc-classifiers of size at most one as defined in Section 7.1), a posterior distribution Q (either uniform on messages or aligned and uniform on messages as defined in Section 7.1.1) on $\mathcal{H}^S$, an empirical loss ζ_S^Q, and, possibly, the Kullback-Leibler Divergence $\mathrm{KL}(Q\|P^S)$ between distribution P^S and Q. Earlier, we defined the prior distribution P^S and the posterior distribution Q. In the following sections, we first calculate $\mathrm{KL}(Q\|P^S)$ for the case of posteriors that are uniform on messages (but not necessarily aligned). We, then, calculate the empirical loss ζ_S^Q for both cases where the considered posteriors are uniform on messages, and uniform on messages and aligned.

7.1.4 Calculation of Kullback-Leibler Divergence

Here, we present the calculation through which we obtain the Kullback-Leibler Divergence between distributions P^S and Q. In Chapter 6, we have seen that aligned posteriors lead us to risks bounds that have the unusual property of having no KL divergence. Therefore, in these cases, we do not need to calculate the $\mathrm{KL}(Q\|P^S)$. Thus, in the following we calculate $\mathrm{KL}(Q\|P^S)$ for the posterior Q that is uniform on messages but is not aligned (See Subsection 7.1.1).

$$\begin{aligned}
\mathrm{KL}(Q\|P^S) &= \mathop{\mathbf{E}}_{h\sim Q} \ln\tfrac{Q(h)}{P(h)} \\
&= \sum_{i=1}^{m} \sum_{s\in\{-,+\}} \int_{\mathcal{M}^1} d\sigma Q(h^{(\sigma,s)}_{S_{\langle i\rangle}}) \ln\left[\frac{Q(h^{(\sigma,s)}_{S_{\langle i\rangle}})}{P^S(h^{(\sigma,s)}_{S_{\langle i\rangle}})}\right] + \sum_{s\in\{-,+\}} Q(h^{(\varepsilon,s)}_{S_{\langle\rangle}}) \ln\left[\frac{Q(h^{(\varepsilon,s)}_{S_{\langle\rangle}})}{P^S(h^{(\varepsilon,s)}_{S_{\langle\rangle}})}\right] \\
&= \sum_{i=1}^{2m} \int_{\mathcal{M}^1} d\sigma \frac{v_i}{|\mathcal{M}^1|} \ln\left[\frac{\frac{v_i}{|\mathcal{M}^1|}}{\frac{1}{2|\mathcal{M}^1|(m+1)}}\right] + v_+ \ln\frac{v_+}{\frac{1}{2(m+1)}} + v_- \ln\frac{v_-}{\frac{1}{2(m+1)}} \\
&= \sum_{v\in\mathbf{v}} v \ln\left[\frac{v}{\frac{1}{2(m+1)}}\right] \\
&= \ln(2m+2) + \sum_{v\in\mathbf{v}} v\ln[v]\ . \qquad (7.16)
\end{aligned}$$

7.1.5 Choice of the Empirical Loss ζ_S^Q:

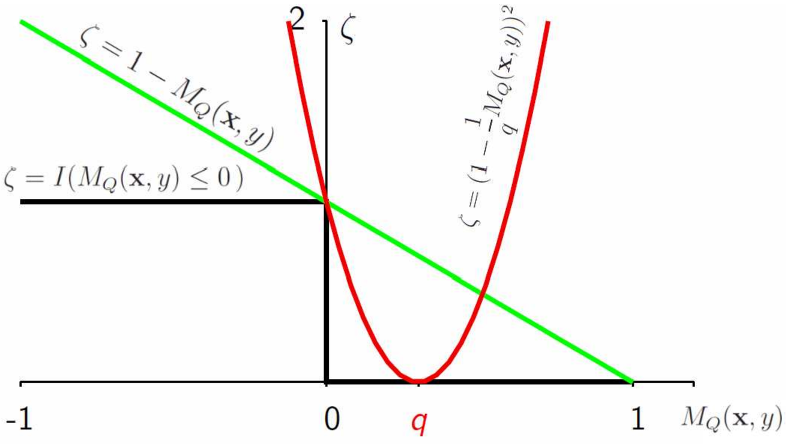

Figure 7.1: Three different loss functions.

As we mentioned earlier, in Chapter 6 Section 6.2, in this thesis, we restrict ourselves to losses that upper-bound the zero-one loss of B_Q (the loss is described by the black line in Figure 7.1). Moreover, and in order to obtain a tractable optimization problem, we propose to use a convex loss function of the margin of the Q-convex combination of sc-classifiers where the margin on example $(\mathbf{x}, y)$ of the Q-convex combination is given by

$$M_Q(\mathbf{x}, y) \stackrel{\text{def}}{=} \mathbf{E}_{h^\mu_{S_\mathbf{i}}\sim Q}\, y h^\mu_{S_\mathbf{i}}(\mathbf{x})\,.$$

Figure 7.1 shows three different loss functions: zero-one loss $(I(M_Q(\mathbf{x},y) \leq 0\,))$, linear loss $(1 - M_Q(\mathbf{x},y))$, and Quadratic loss $(1 - \frac{1}{q}M_Q(\mathbf{x},y))^2$ where q is the minimum of the parabolic of the loss function. As shown in this figure, the linear loss and quadratic loss upper bound the zero-one loss. Moreover, recall that,

$$R_D(G_Q) = \frac{1}{2} - \frac{1}{2}\,\mathbf{E}_{(\mathbf{x},y)\sim D} M_Q(\mathbf{x},y) \tag{7.17}$$

or equivalently,

$$2R_D(G_Q) = 1 - \mathbf{E}_{(\mathbf{x},y)\sim D} M_Q(\mathbf{x},y) \tag{7.18}$$

gives a relation between the Gibbs risk $R_D(G_Q)$ and the expected margin $E_{(\mathbf{x},y)\sim D} M_Q(\mathbf{x},y)$.

In [[19]], we developed a learning algorithm that minimizes a bound derived from the PAC-Bayes theory on the majority vote of complementary classifiers. The proposed bound applies the PAC-Bayes theory to general loss functions which leads us to minimization of a quadratic loss function. Inspired by [[19]], let us consider margin losses of the form $\zeta(\alpha) = \left(1 + \frac{1}{q}\,\alpha\right)^2$ where q is the minimum of the parabolic of the loss function. The parameter q will be, therefore, a hyper-parameter of the algorithms we will propose later in this chapter. This parameter will be determined on each dataset by cross-validation.

The reason for choosing margin losses of the form $\zeta(\alpha) = \left(1 + \frac{1}{q}\,\alpha\right)^2$ is that in our setting, the sc-classifiers are weak (their compression sets are of size at most one). This way, the majority vote of sc-classifiers may have (in practice, really have) a low risk even if the Gibbs risk, which is the average risk of all voters, is necessarily close to 1/2. Clearly, the average of weak voters is also weak. Now, it follows from Equation (7.18) that the expected margin $E_{(\mathbf{x},y)\sim S} M_Q(\mathbf{X},y)$ will be close to 0. Thus, if we consider the case where the linear loss is chosen (i.e., :$\zeta(M_Q(\mathbf{x},y)) = 1 - M_Q(\mathbf{x},y)$), the obtained bound on the risk of the majority vote will be close to one. Since we know that the risk of the majority vote never exceeds one, this bound gives us very little information. In this chapter, we will nevertheless consider this possible choice of loss function and empirically provide evidence that reinforce our point of view. On the other hand, and especially if the variance of $M_Q(\mathbf{X},y)$ is small, by taking a value of q close to $E_{(\mathbf{x},y)\sim S} M_Q(\mathbf{X},y)$ we will obtain a very small (quadratic) loss, and therefore, a very small bound on the risk of the majority vote, see Figure 7.1.

For margin losses of the form $\zeta(\alpha) = \left(1 + \frac{1}{q}\,\alpha\right)^2$, we have $\zeta(1) = (1 + q^{-1})^2$ and $\zeta^{'}(1) = (2q + 2)/q^2$ and, consequently, ζ_S^Q is convex in Q. We will see that this will result in a tractable optimization problem that can be efficiently solved by a learning algorithm (see Section 7.3). Moreover, using a small positive value for parameter q favors the majority vote of small margins unlike the Gibbs classifiers that favor large margins ((See Figure 7.1). It is known that when voters are weak, majority votes of small margins can result in a powerful classifier. In the next section, we present the calculation of the quadratic and linear empirical losses.

7.1.6 Calculation of Empirical Losses:

In this section, we present the calculations of the empirical quadratic loss of the form $\zeta(\alpha) = \left(1 + \frac{1}{q}\,\alpha\right)^2$, and linear empirical loss of the form of $\zeta(\alpha) = 1 + \alpha$.

Calculation of Empirical Quadratic Loss:

- **Empirical quadratic loss $\zeta_S^{Q,q}$: uniform on messages posteriors**

  Here, we present the calculation of the empirical quadratic loss for the case where posteriors are not necessarily aligned to the prior. Thus, Q is defined by Equation (7.3) and to express $\zeta_S^{Q,q}$ in terms of $\mathbf{v}$, the margin $M_Q(\mathbf{x}, y)$ on example $(\mathbf{x}, y)$ is given by

$$M_Q(\mathbf{x}, y) \overset{\text{def}}{=} E_{h^{\mu}_{S_\mathbf{i}} \sim Q} y h^{\mu}_{S_\mathbf{i}}(\mathbf{x}) = y\left[(v_+ - v_-) + \sum_{i=1}^{m}(v_i - v_{i+m})k(\mathbf{x}_i, \mathbf{x})\right].$$

  Consequently, we have

$$\zeta_S^{Q,q} = \frac{1}{mq^2}\sum_{j=1}^{m}\left(q - y_j\left[v_+ - v_- + \sum_{i=1}^{m}(v_i - v_{i+m})k(\mathbf{x}_i, \mathbf{x}_j)\right]\right)^2.$$

  Let $v_0 \overset{\text{def}}{=} v_+$ and $v_{2m+1} \overset{\text{def}}{=} v_-$. Let us define a matrix $\mathbf{G}$ of size $(2m+2) \times m$ as

$$G_{i,j} = \begin{cases} 1 & \text{if } i = 0, \\ k(\mathbf{x}_i, \mathbf{x}_j) & \text{if } 1 \le i, j \le m, \\ -k(\mathbf{x}_{i-m}, \mathbf{x}_j) & \text{if } m+1 \le i \le 2m \text{ (and } 1 \le j \le m\text{)}, \\ -1 & \text{if } i = 2m+1\,. \end{cases} \tag{7.19}$$

  With this notation we have:

$$\zeta_S^{Q,q} = \frac{1}{mq^2}\sum_{j=1}^{m}\left(q - y_j\sum_{i=0}^{2m+1} v_i G_{i,j}\right)^2 \tag{7.20}$$

- **Empirical quadratic loss $\zeta_S^{Q,q,al}$: aligned and uniform on messages posteriors**

  Now, we present the calculation of the empirical Quadratic loss for the case where posteriors are aligned and uniform on messages. Thus, Q is defined by Equations (7.8) and (7.9) and to express $\zeta_S^{Q,q,al}$ in terms of $\mathbf{w}$, the margin $M_Q(\mathbf{x}, y)$ on example $(\mathbf{x}, y)$ is given by:

  $$M_Q(\mathbf{x}, y) \stackrel{\text{def}}{=} E_{h^{\mu}_{S_\mathbf{i}} \sim Q} y h^{\mu}_{S_\mathbf{i}}(\mathbf{x}) = y\left[w_0 + \sum_{i=1}^{m} w_i k(\mathbf{x}_i, \mathbf{x})\right].$$

  Consequently, the empirical quadratic risk $\zeta_S^{Q,q,al}$ is given by

  $$\zeta_S^{Q,q,al} = \frac{1}{mq^2}\sum_{j=1}^{m}\left(q - y_j\left[w_0 + \sum_{i=1}^{m} w_i k(x_i, x_j)\right]\right)^2.$$

  Let us define a matrix $\mathbf{G}$ of size $(m+1)\times m$ as

  $$G_{i,j} = \begin{cases} 1 & \text{for } i = 0, \\ k(\mathbf{x}_i, \mathbf{x}_j) & \text{for } 1 \le i, j \le m\ . \end{cases} \tag{7.21}$$

  With this notion we have:

  $$\begin{aligned} \zeta_S^{Q,q,al} &= \frac{1}{m}\sum_{j=1}^{m}\left(1 - \frac{1}{q} y_j \sum_{k=0}^{m} w_k G_{k,j}\right)^2 \\ &= \frac{1}{q^2 m}\sum_{j=1}^{m}\left(q - y_j \sum_{k=0}^{m} w_k G_{k,j}\right)^2 \end{aligned} \tag{7.22}$$

Calculation of Empirical Linear Loss:

- **Empirical linear loss $\zeta_S^{Q,lin}$: uniform on messages posteriors**

  Here, we present the calculation of the empirical linear loss for the case where posteriors are not aligned to the prior. To express $\zeta_S^{Q,lin}$ in terms of $\mathbf{v}$, recall that the margin $M_Q(\mathbf{x}, y)$, on example $(\mathbf{x}, y)$, is given by

  $$M_Q(\mathbf{x}, y) = y\left[(v_+ - v_-) + \sum_{i=1}^{m}(v_i - v_{i+m}) k(\mathbf{x}_i, \mathbf{x})\right].$$

Therefore, we have:

$$\zeta_S^{Q,lin} = \frac{1}{m}\sum_{j=1}^{m}(1 - y_j \sum_{i=0}^{2m+1} v_i G_{i,j}),$$

where $G_{i,j}$ is given by Equation (7.19).

- **Empirical linear loss $\zeta_S^{Q,lin,al}$: aligned and uniform on messages posteriors**

  Here, we present the calculation of empirical linear loss for the case that posterior are aligned to the prior. To express $\zeta_S^{Q,lin,al}$ in terms of $\mathbf{w}$, recall that the margin $M_Q(\mathbf{x}, y)$, on example $(\mathbf{x}, y)$, is given by:

  $$M_Q(\mathbf{x}, y) = y\left(w_0 + \sum_{i=1}^{m} w_i k(\mathbf{x}_i, \mathbf{x})\right).$$

  Therefore, we have:

  $$\zeta_S^{Q,lin,al} = \frac{1}{m}\sum_{j=1}^{m}(1 - y_j \sum_{i=0}^{m} w_i G_{i,j}) \tag{7.23}$$

  where $G_{i,j}$ is given by Equation (7.21).

Note that, the matrix $\mathbf{G}$ is very similar to the Gram matrix used in kernel methods (except it does not have to be positive semi-definite) [50]. Note also that with this matricial notation, we can easily express both the margin and the majority vote classifier. Indeed, for any $(x, y) \in \mathcal{X} \times \mathcal{Y}$, we have $M_Q(x, y) = y\mathbf{w} \cdot \mathbf{G}(x)$ and $B_Q(x) = \operatorname{sgn}(\mathbf{w} \cdot \mathbf{G}(x))$.

7.2 The Influence of parameter q on the value of the proposed Bounds

In this section, we investigate the influence of parameter q on each of the proposed risk bound.

Figure 7.2 shows the value of the bounds of Theorems 6.4.7 and 6.4.4 according to q (for $m = 300$, $\zeta_S^{Q,q,al} = 0.2$, $\delta = \frac{1}{20}$). These two bounds are only valid for aligned posteriors and both have no KL terms.

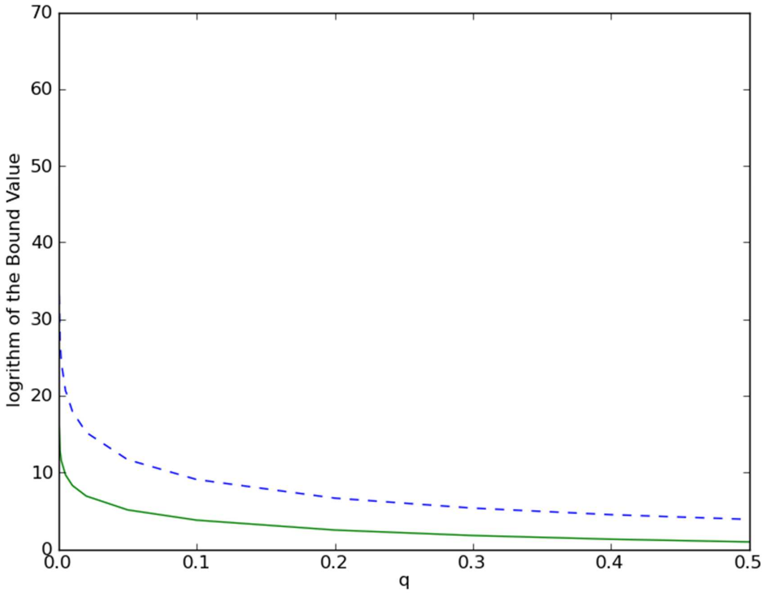

Figure 7.2: The value of the bounds for Theorem 6.4.7 (solid line) and Theorem 6.4.4 (dashed line) according to q (For $m = 300$, $\zeta_S^{Q,q,al} = 0.2$, and $\delta = \frac{1}{20}$).

Figure 7.3 shows the value of the bounds of Theorems 6.3.1, 6.3.3 and 6.3.5 (for $C_1 = 1$ $m = 300$, $\zeta_S^{Q,q} = 0.2$, $\delta = \frac{1}{20}$, and $\mathrm{KL}(Q \parallel P^S) = 20$) according to q.

As we can see in the case of without KL(Figure 7.2) the bound of Theorem 6.4.7 is tighter than the bound of Theorem 6.4.4. In the case with a KL term the bounds scale with similar values (as shown by the Figure 7.3), however, the Seeger bound (Theorem 6.3.5) is always tighter than the two others (the McAllester (Theorem 6.3.3) and the Catoni (Theorem 6.3.1) with C=1. It is important to notice that, $\zeta(1)$, $\zeta^{'}(1)$ are very large values if q, the minimum of the function of quadratic loss, is near 0. Consequently, the bounds become trivial (their values are greater than 1). However, the empirical results obtained by [[19]] show that it is appropriate to design a learning algorithm that minimizes such bounds. Many attempts have been made to tighten those bounds, but, unfortunately, without real success. Nevertheless, as it is also the case for other types of bounds, such as the VC-bounds [53], even if their values are very large, they are a good guidance to select classifiers. The rest of this chapter together with the next chapter will make this idea clearer.

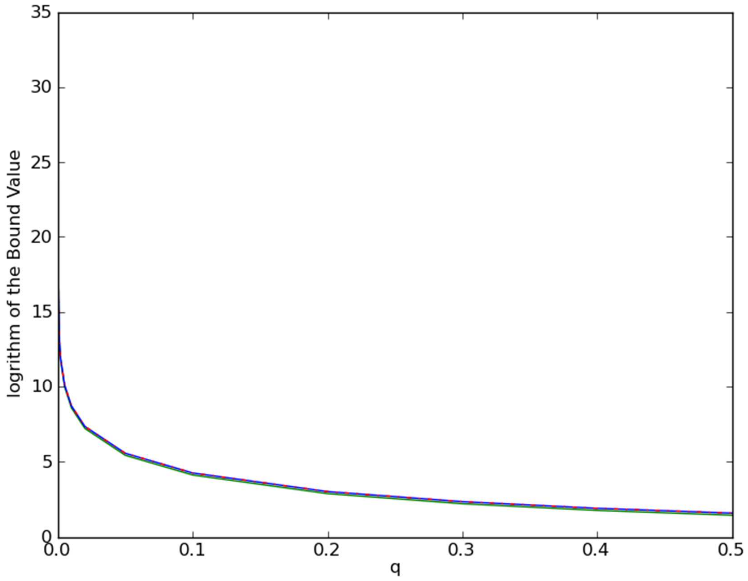

Figure 7.3: The value of the bounds of Theorems 6.3.1, 6.3.3 and 6.3.5 ((for $C_1 = 1$ $m = 300$, $\zeta_S^{Q,q} = 0.2$, $\delta = \frac{1}{20}$, and $\mathrm{KL}(Q \parallel P^S) = 20$)) according to q.

7.3 Bound Minimization Algorithms

Previously, we presented the majority vote of sample compressed classifiers and showed how we can apply the proposed risk bounds of Chapter 6 to bound the majority vote itself. This suggests the following learning algorithm: find the distribution Q minimizing a given bound, and then return the majority vote of sample compressed classifiers weighted by Q. We can categorize the learning algorithms obtained by minimizing these bounds into three groups, namely, PBSC-A, PBSC-L, and PBSC-N;

PBSC-A is the learning algorithm that finds the aligned and uniform on messages posterior Q minimizing the bounds of Chapter 6 with no KL term using a quadratic loss. Note that we have therefore two possibilities to define the PBSC-A algorithm: Theorem 6.4.4 (McAllester with no KL term) and Theorem 6.4.7 (Seeger with no KL term). We will show later that minimizing these two bounds give rise to the same algorithm. In both cases, the posterior Q minimizing these bounds will be equivalent to minimizing the empirical quadratic loss $\zeta_S^{Q,q,al}$.

PBSC-L is similar to the PBSC-A algorithm except that PBSC-L consists of minimizing the linear loss $\zeta_S^{Q,lin,al}$. Recall that this algorithm will is not expected to be accurate. We only define it for the sake of comparison.

PBSC-N is the learning algorithm that finds the uniform on messages posterior Q (without restricting to be aligned to the prior) minimizing the bound of Chapter 6 with KL term corresponding to Theorem 6.3.1. The bound is known as the Generalized version of the Catoni bound (with KL).[1] We will see later that finding the posterior Q (without restricting to be aligned to the prior) that minimizes this bound also minimizes the following function:

$$C \cdot \zeta_S^{Q,q} + \mathrm{KL}(Q \| P^S),$$

where C is some positive constant and $\zeta_S^{Q,q}$ is the empirical quadratic loss.

In the following we present these learning algorithms (PBSC-A, ,PBSC-L, PBSC-N) in more detail.

7.3.1 The PBSC-A Algorithm

Minimizing the Bound of Theorem 6.4.4 (McAllester With no KL Term)

Consider Theorem 6.4.4, given any training sequence S with m examples, any prior P^S, a confident parameter δ, and a fixed loss function ζ given by coefficient $\{a_k\}_{k=0}^{\deg(\zeta)}$, the objective is to find the posterior Q that minimizes the bound B which is given by the following function:

$$B = \zeta_S^{Q,q,al} + \frac{\zeta(1)}{\frac{1}{2}(\sqrt{m - l\deg(\zeta)})} \sqrt{4\,l\deg(\zeta) + \ln\frac{2\sqrt{m}}{\delta}}$$

Since δ and $\zeta(1)$ are constant, the posterior Q that minimizes the following function

[1] Note that, in Chapter 6, we presented two other bounds with KL term: Theorem 6.3.3 (McAllester with KL) and Theorem 6.3.5 (Seeger with KL)). Here, we do not present the algorithms that minimize these two bounds since the objective functions associated with these bounds may not be convex leading to a much more complicated optimization problem. Moreover, preliminary results (not given here) showed that these two other possible versions are not as accurate as the Catoni's version (Theorem 6.3.1).

also minimizes the bound B:

$$f_A(Q) \stackrel{\text{def}}{=} \zeta_S^{Q,q,al} \tag{7.24}$$

This theorem indicates that $l \cdot \deg(\zeta)$ should be small for the risk bound to be small. Consequently, as stated before, we consider here only margin losses that are quadratic or linear ($\deg(\zeta) \leq 2$) and sc-classifiers of compression sequence size of at most one ($l \leq 1$). Moreover, for algorithmic simplicity, here, we restrict ourselves to aligned and uniform on messages posteriors. Knowing that $\zeta_S^{Q,q,al}$ is convex in Q, it follows that objective function f to minimize is always convex in Q.

The Optimization Problem

By substituting $\zeta_S^{Q,q,al}$ with its value as given by Equation (7.22) in Equation (7.24), we obtain the following optimization problem:

$$\begin{aligned} \text{Minimize:} \quad f_A(\mathbf{w}) \quad &= \sum_{j=1}^{m} \left(q - y_j \sum_{i=0}^{m} w_i G_{i,j} \right)^2 \\ \text{subject to:} \quad &|w_i| \leq \tfrac{1}{m+1} \text{ for } i = 0, 1, \ldots, m \,, \end{aligned} \tag{7.25}$$

where $G_{i,j}$ is given by Equation (7.21).

We propose to solve this optimization problem by minimizing f_A coordinate-wise, similarly as it is done for AdaBoost ([46]), with the difference that we will have to ensure that Q remains an aligned distribution at each step of the algorithm. Starting from the uniform distribution P (i.e., $\mathbf{w} = \mathbf{0}$), the learning algorithm iteratively chooses (at random) $k \in \{0, \ldots, m\}$, and updates $w_k \leftarrow w_k + \theta$ (without updating the other weights) according to some optimally chosen value of θ. Let $\mathbf{w}_{k,\theta}$ be the new weight vector obtained with such an update ($\mathbf{w}_{k,\theta} = (w_0, \cdots, w_{k-1}, w_k + \theta, w_{k+1}, \cdots, w_m)$). After that update, the objective function becomes:

$$f_A(\mathbf{w}_{k,\theta}) \;=\; \sum_{j=1}^{m} \left[q - y_j \left(\sum_{i=0}^{m} w_i G_{i,j} + \theta G_{k,j} \right) \right]^2 .$$

The optimal value for θ is obtained when $\frac{df_A(\mathbf{w}_{k,\theta})}{d\theta} = 0$, provided that $w_k + \theta \in$

$[\frac{-1}{m+1}, \frac{1}{m+1}]$. The derivative of f_A with respect to the θ is given by

$$\begin{aligned}
\frac{\partial f_A(\mathbf{w}_{k,\theta})}{\partial \theta} &= \sum_{j=1}^{m} 2\left(q - y_j \sum_{i=0}^{m} w_i G_{i,j} - y_j \theta G_{k,j}\right)(-y_j G_{k,j}) \\
&= 2\sum_{j=1}^{m}\left[\theta G_{k,j}^2 + y_j G_{k,j}\left(y_j \sum_{i=0}^{m} w_i G_{i,j} - q\right)\right] \\
&= 2\left[\theta \sum_{j=1}^{m} G_{k,j}^2 + \sum_{j=1}^{m} G_{k,j}\left(\sum_{i=0}^{m} w_i G_{i,j} - q y_j\right)\right] \\
&= 2\left[\theta \sum_{j=1}^{m} G_{k,j}^2 + \sum_{j=1}^{m} G_{k,j} D_{\mathbf{w}}(j)\right], \qquad (7.26)
\end{aligned}$$

where $D_{\mathbf{w}}(j) \stackrel{\text{def}}{=} \sum_{i=0}^{m} w_i G_{i,j} - q y_j$.

Equation (7.26) implies that, for a given k, the optimal value for θ is given by:

$$\theta = -\frac{\sum_{j=1}^{m} G_{k,j}\, D_{\mathbf{w}}(j)}{\sum_{j=1}^{m} G_{k,j}^2}. \qquad (7.27)$$

Algorithm 3 presents the complete optimization procedure that we have used.

Algorithm 3 : PBSC-A optimization procedure

```
**Initialize:** $w_i = 0 \;\; \forall i \in \{0, \ldots, m\}$ and $D_{\mathbf{w}}(j) = -q\, y_j \;\; \forall j \in \{1, \ldots, m\}$ .
**repeat**
    Choose at random $k \in \{0, .., m\}$.
    Compute $\theta$ given by Equation (7.27) .
    If $[w_k + \theta > \frac{1}{m+1}]$ then $\theta \leftarrow \frac{1}{m+1} - w_k$.
    If $[w_k + \theta < \frac{-1}{m+1}]$ then $\theta \leftarrow \frac{-1}{m+1} - w_k$.
    $w_k \leftarrow w_k + \theta$.
    Update $D_{\mathbf{w}}(j) \leftarrow D_{\mathbf{w}}(j) + \theta\, G_{k,j} \;\; \forall j \in \{1, \ldots, m\}$.
**until** Convergence
```

Minimizing the Bound of Theorem 6.4.7 (Seeger With no KL Term)

In this section, we show that minimizing the bound given in Theorem 6.4.7 gives rise to the same optimization problem given by Equation (7.25). In other words, the posterior Q minimizing this bound also minimizes the empirical quadratic loss $\zeta_S^{Q,q,al}$.

Consider Theorem 6.4.7, given any training sequence S with m examples, a prior P^S, a confidence parameter δ, and a fixed loss function ζ given by coefficients $\{a_k\}_{k=0}^{\deg(\zeta)}$, the objective is to find the posterior Q that minimizes the bound given by the following function:

$$\mathrm{kl}^{+}\left(\frac{m}{m-l\cdot\deg\zeta}\,(\frac{1}{2}\left[1+\frac{1}{\zeta(1)}\ \zeta_S^{Q,q,al}\right]+\frac{ld}{m})\ \|\ \frac{1}{2}\left[1+\frac{1}{\zeta(1)}\ \zeta_D^{Q}\right]\right)\ \le\ \frac{\ln\frac{2\sqrt{m}}{\delta}}{m-l\cdot\deg\zeta}$$

Starting from the uniform distribution P^S (i.e., $\mathbf{w}=\mathbf{0}$), the learning algorithm iteratively chooses (at random) $k\in\{0,..,m\}$, and updates $w_k\leftarrow w_k+\theta$, (without updating the other weights) according to some optimally chosen value of θ. Let $Q_{k,\theta}$ and $\mathbf{w}_{k,\theta}$ be, respectively, the new posterior and the new weight vector obtained with such an update. Let f be a function given by the bound of Theorem 6.4.7. Here, we claim that, considering the function f given by the bound of Theorem 6.4.7, the optimal value for θ is obtained when $\frac{df(Q_{k,\theta})}{d\theta}=0$ provided that $w_k+\theta\in[\frac{-1}{m+1},\frac{1}{m+1}]$ where

$$\frac{df(Q_{k,\theta})}{d\theta}=\frac{d\zeta_S^{Q_{k,\theta},q,al}}{d\theta}=0$$

Proof of claim:

Let $M_{k,\theta}$ be the function that represents the bound of the $\zeta_S^{Q_{k,\theta},q,al}$ provided by the Theorem 6.4.7,

$$M_{k,\theta}\overset{\text{def}}{=}\max_{B\in[0,1]}\left\{B:\mathrm{kl}^{+}\left(\frac{m}{m-l\cdot\deg\zeta}\,(\frac{1}{2}\left[1+\frac{1}{\zeta(1)}\ \zeta_S^{Q_{k,\theta},q,al}\right]+\frac{ld}{m})\ \|\ \frac{1}{2}\left[1+\frac{1}{\zeta(1)}\ B\right]\right)\ =\ \frac{\ln\frac{2\sqrt{m}}{\delta}}{m-l\cdot\deg\zeta}\right\}$$

Note that, it follows from the definition of kl^{+} (Theorem 6.4.7) that one can replace kl^{+} by kl in the definition of $M_{k,\theta}$. Indeed, if $m=\max_{b\in[0,1]}\{b:kl(a||b)=c\}$, then $a\le m$, because $f(x)=\mathrm{kl}(a||x)$ is a convex function on $[0,1]$ having its minimum at $x=a$. Thus, the values of kl and kl^{+} coincide when $b=m$.

Here, we have to show that for any fixed k, we have:

$$\frac{d\zeta_S^{Q_{k,\theta},q,al}}{d\theta}=0\Leftrightarrow\frac{dM_{k,\theta}}{d\theta}=0$$

By definition of $M_{k,\theta}$ we have:

$$0=\mathrm{kl}\left(\frac{m}{m-l\cdot\deg\zeta}\,(\frac{1}{2}\left[1+\frac{1}{\zeta(1)}\ \zeta_S^{Q_{k,\theta},q,al}\right]+\frac{ld}{m})\ \|\ \frac{1}{2}\left[1+\frac{1}{\zeta(1)}\ M_{k,\theta}\right]\right)\ -\ \frac{\ln\frac{2\sqrt{m}}{\delta}}{m-l\cdot\deg\zeta}$$

For simplicity, we consider the following definitions:

$$A_S = \frac{m}{m - l \cdot \deg \zeta}(\frac{1}{2}\left[1 + \frac{1}{\zeta(1)}\ \zeta_S^{Q_{k,\theta},q,al}\right] + \frac{ld}{m}), \qquad A_D = \frac{1}{2}\left[1 + \frac{1}{\zeta(1)}\ M_{k,\theta}\right]$$

By the above definition, for all θ, we have:

$$0 = \mathrm{kl}\left(A_S \parallel A_D\right) - \frac{\ln \frac{2\sqrt{m}}{\delta}}{m - l \cdot \deg \zeta}.$$

By taking the derivative of each side of the above equation we have:

$$0 = \frac{d}{d\theta}\left[\mathrm{kl}\left(A_S \parallel A_D\right) - \frac{\ln \frac{2\sqrt{m}}{\delta}}{m - l \cdot \deg \zeta}\right] = \frac{A_D - A_S}{(1 - A_D)A_D}\frac{dA_D}{d\theta} + \log \frac{A_S(1 - A_D)}{A_D(1 - A_S)}\frac{dA_S}{d\theta}$$

for optimal value of θ, $\frac{dA_D}{d\theta} = 0$ it then follows that we have:

$$\log\left(\frac{A_S(1 - A_D)}{A_D(1 - A_S)}\right) \cdot \frac{dA_S}{d\theta} = 0 \qquad (7.28)$$

Since $\log(x) \neq 0 \quad \forall x \neq 1$ and $1 > A_D > A_S > 0$, Equation (7.28) is equivalent to:

$$\frac{dA_S}{d\theta} = 0$$

Therefore, we have:

$$\frac{dA_S}{d\theta} = 0 \Leftrightarrow \frac{dA_D}{d\theta} = 0$$

According to the definitions of A_S and A_D it immediately follows that:

$$\frac{d\zeta_S^{Q_{k,\theta},q,al}}{d\delta} = 0 \Leftrightarrow \frac{dM_{k,\theta}}{d\theta} = 0,$$

which proves the claim.

7.3.2 The PBSC-L Algorithm

To show the need for a non-linear margin loss function in PBSC-A algorithms, we propose the PBSC-L algorithm that consists of finding the aligned and uniform on messages posterior Q that minimizes the empirical linear loss $\zeta_S^{Q,lin,al}$ given by Equation (7.23).

This way, the optimization problem for PBSC-L can be written as:

$$\begin{aligned} \text{Minimize:} \quad & f_A(\mathbf{w}) &= \sum_{j=1}^{m} \left(1 - y_j \sum_{i=0}^{m} w_i G_{i,j}\right) \\ \text{subject to:} \quad & & |w_i| \le \tfrac{1}{m+1} \text{ for } i = 0, 1, \ldots, m \,, \end{aligned} \tag{7.29}$$

where $G_{i,j}$ is given by Equation (7.21).

We propose to solve this optimization problem by minimizing f_A in the following way: starting from the uniform distribution P (i.e., $\mathbf{w} = \mathbf{0}$), the learning algorithm iteratively chooses (at random) $k \in \{0, \ldots, m\}$, and updates w_k (without updating the other weights). Equation (7.29) shows that the weight given to w_k does not depend on the other weights. Therefore, the algorithm is very simple. It is sufficient to update the weights w_k for all $k \in \{0, \ldots, m\}$ for a single time in the following way: if $\sum_{j=1}^{m}(-y_j G_{k,j}) < 0$, it means that the derivative is negative and we need to update the w_k with the maximum weight ($w_k = \frac{1}{m+1}$). If $\sum_{j=1}^{m}(-y_j G_{k,j}) > 0$, it means that the derivative is positive and we need to update the w_k with the minimum weight ($w_k = \frac{-1}{m+1}$). Algorithm 4 presents the complete optimization procedure that we have used.

Algorithm 4 : PBSC-L optimization procedure

```
1: Initialize: $w_i = 0 \quad \forall i \in \{0, \ldots, m\}$.
2: For each $k \in \{0, .., m\}$ repeat steps 3 to 5:
3: Compute $\theta = \sum_{j=1}^{m}(-y_j G_{k,j})$ .
4: If $\theta < 0$ then $w_k \leftarrow \frac{1}{m+1}$.
5: If $\theta > 0$ then $w_k \leftarrow \frac{-1}{m+1}$.
```

7.3.3 The PBSC-N Algorithm

In this section, we present the PBSC-N algorithm that minimizes the bound of Theorem 6.3.1, (Catoni with KL).

Minimizing the Bound of Theorem 6.3.1 (Catoni with KL)

Consider Theorem 6.3.1, given any training sequence S with m examples, a prior P^S, a confidence parameter δ, a fixed loss function ζ given by coefficients $\{a_k\}_{k=0}^{\deg(\zeta)}$, and any positive value C', the objective is to find the uniform on messages posterior Q that minimizes the bound B which is given by the following function:

$$B = \zeta(1)[C'-1] + C' \cdot \left(\zeta_S^{Q,q} + \frac{2}{mC_1}[\zeta'(1) \cdot \mathrm{KL}(Q\|P^S) + \zeta(1) \cdot \ln\frac{1}{\delta}]\right)$$

Since δ and $\zeta(1)$ are constant the posterior Q that minimizes the following function also minimizes the bound B:

$$C \cdot \zeta_S^{Q,q} + \mathrm{KL}(Q\|P^S) \tag{7.30}$$

Note that C, in the above equation, is some positive constant obtained form C_1, $\zeta(1)$, $\zeta'(1)$, and m. The theorem 6.3.1 indicates that $l \cdot \deg(\zeta)$ should be small for the risk bound to be small. Consequently, we will, as usual, consider here only margin losses that are quadratic and sc-classifiers of compression sequence size of at most one.

Optimization Problem

By substituting $\zeta_S^{Q,q}$ with its value given by Equation (7.20) and $\mathrm{KL}(Q\|P^S)$ with its value given by Equation (7.16) in Equation (7.30) we obtain the following optimization problem:

$$\begin{aligned}
\text{Minimize:} \quad f_N(\mathbf{v}) \; &= \frac{C}{mq^2}\sum_{j=1}^{m}\left(q - y_j\sum_{i=0}^{2m+1} v_i G_{i,j}\right)^2 + \sum_{i=0}^{2m+1} v_i \ln v_i \\
\text{subject to:} \quad & v_i \geq 0 \quad \text{for } i = 0, 1, \ldots, 2m+1\,, \\
& \sum_{i=0}^{2m+1} v_i = 1
\end{aligned} \tag{7.31}$$

where $G_{i,j}$ is given by Equation (7.19).

We propose to solve this optimization problem by minimizing f_N with a coordinate-pair descent algorithm that works iteratively by exchanging weights between two components of $\mathbf{v}$. Starting from the uniform distribution P (i.e., $v_i = \frac{1}{2m+2}$ for $i = 0, 1, \ldots, 2m+1$), the learning algorithm iteratively chooses (at random) $k, l \in \{0, \ldots, 2m+1\}$ (with $k \neq l$), and updates $v_k \leftarrow v_k + \theta$ and $v_l \leftarrow v_l - \theta$ (without updating the other weights) according to some optimally chosen value of θ. Let $\mathbf{v}_{k,\theta}$ be the new weight vector obtained with such an update. After an update, the objective function becomes

$$\begin{aligned}
f_N(\mathbf{v}_{k,\theta}) \; &= \; \frac{C}{mq^2}\sum_{j=1}^{m}\left[q - y_j\left(\sum_{i=0}^{2m+1} v_i G_{i,j} + \theta G_{k,j} - \theta G_{l,j}\right)\right]^2 \\
&+ \sum_{i=0}^{2m+1} I(i \notin \{k,l\}) \cdot v_i \ln v_i + (v_k + \theta)\ln(v_k+\theta) + (v_l - \theta)\ln(v_l - \theta)
\end{aligned}$$

The optimal value for θ is obtained when $\frac{df_N(\mathbf{v}_{k,\theta})}{d\theta} = 0$, provided that $v_k + \theta \in [0, v_k + v_l]$ and $v_l - \theta \in [0, v_k + v_l]$. The derivative of f_N with respect to the θ is given by

$$\begin{aligned}
\frac{\partial f_N(\mathbf{v}_{k,\theta})}{\partial \theta} \; &= \; \frac{C}{mq^2}\sum_{j=1}^{m} 2\left(q - y_j\sum_{i=0}^{2m+1} v_i G_{i,j} - y_j\theta\left(G_{k,j} - G_{l,j}\right)\right)\left(-y_j\left(G_{k,j} - G_{l,j}\right)\right) + \ln\frac{v_k+\theta}{v_l-\theta} \\
&= \; \frac{2C}{mq^2}\sum_{j=1}^{m}\left[\theta\left(G_{k,j} - G_{l,j}\right)^2 + y_j\left(G_{k,j} - G_{l,j}\right)\left(y_j\sum_{i=0}^{2m+1} v_i G_{i,j} - q\right)\right] + \ln\frac{v_k+\theta}{v_l-\theta} \\
&= \; \frac{2C}{mq^2}\left[\theta\sum_{j=1}^{m}\left(G_{k,j} - G_{l,j}\right)^2 + \sum_{j=1}^{m}\left(G_{k,j} - G_{l,j}\right)\left(\sum_{i=0}^{2m+1} v_i G_{i,j} - qy_j\right)\right] + \ln\frac{v_k+\theta}{v_l-\theta} \\
&= \; \frac{2C}{mq^2}\left[\theta\sum_{j=1}^{m}\left(G_{k,j} - G_{l,j}\right)^2 + \sum_{j=1}^{m}\left(G_{k,j} - G_{l,j}\right) D_{\mathbf{v}}(j)\right] + \ln\frac{v_k+\theta}{v_l-\theta}\,,
\end{aligned} \tag{7.32}$$

where $D_{\mathbf{v}}(j) = \sum_{i=0}^{2m+1} v_i G_{i,j} - q y_j$.

We find the optimal value for θ with the help of a root finding method. Algorithm 5 presents the complete optimization procedure that we have used.

Algorithm 5 : PBSC-N (Catoni) optimization procedure

```
Initialize: $v_i = \frac{1}{2m+2}$ $\forall i \in \{0, \ldots, m\}$ and $D_{\mathbf{v}}(j) = -q\, y_j$ $\forall j \in \{1, \ldots, m\}$ .
repeat
    Choose at random $k, l \in \{0, \ldots, 2m+1\}$ (with $k \neq l$).
    Find $\theta$ given by the root of Equation (7.32).
    If $\theta > v_l$ then $\theta \leftarrow v_l$.
    If $\theta < -v_k$ then $\theta \leftarrow -v_k$.
    $v_k \leftarrow v_k + \theta$ and $v_l \leftarrow v_l - \theta$.
    Update $D_{\mathbf{v}}(j) \leftarrow D_{\mathbf{v}}(j) + \theta\,(G_{k,j} - G_{l,j})$ $\forall j \in \{1, \ldots, m\}$.
until Convergence
```

Recovering Ridge Regression from PBSC-N

We can find a quadratic upper bound on the KL$(Q \parallel P^S)$ term for Equation (7.16). According to the Equation (7.16) we have:

$$\begin{aligned}
\mathrm{KL}(Q \parallel P^S) &= \sum_{i=0}^{2m+1} v_i \ln\left[\frac{v_i}{\frac{1}{2(m+1)}}\right] \\
&= \frac{1}{2(m+1)} \sum_{i=0}^{2m+1} (2(m+1))v_i \ln\left[\frac{v_i}{\frac{1}{2(m+1)}}\right] \\
&\le \frac{1}{2(m+1)} \sum_{i=0}^{2m+1} [(2(m+1)v_i]^2 - [2(m+1)v_i] = 2(m+1)) \sum_{i=0}^{2m+1} [v_i]^2 - 1 \\
&\le 2(m+1) \sum_{i=0}^{2m+1} [v_i]^2
\end{aligned} \tag{7.33}$$

The first inequality is obtained using the following inequality:

$$x \ln(x) \le (x^2 - x)\ \forall x > 0$$

This way, the optimization problem for *PBSC-N* given by Equation (7.31) can be written as:

$$\begin{aligned}
\text{Minimize:} \quad f'_N(\mathbf{v}) \quad &= \frac{C}{m}\sum_{j=1}^{m}\left(1 - \frac{1}{q}y_j\sum_{i=0}^{2m+1} v_i G_{i,j}\right)^2 + \sum_{i=0}^{2m+1}(v_i)^2 \\
\text{subject to:} \quad & v_i \geq 0 \qquad \text{for } i = 0, 1, \ldots, 2m+1\,, \\
& \sum_{i=0}^{2m+1} v_i = 1\,.
\end{aligned}$$

Now, if we minimize f'_N for $\mathbf{v}^{'} = \frac{1}{q}\mathbf{v}$ we recover exactly the ridge regression [51] for the quadratic loss.

Chapter 8

Empirical Results

In this section, we present all the empirical results that we have obtained in our experiments.

We present the results of the Algorithm 3 (PBSC-A). Recall that minimizing the proposed bounds that do not depend on the KL term is equivalent to minimizing the empirical Quadratic loss $\zeta_S^{Q,q,al}$.

We also present the results of Algorithm 5 for the minimization of Theorem 6.3.1 (PBSC-N (Catoni)). We chose Theorem 6.3.1 since it has a hyper-parameter C that leads to a hyper-parameter for the algorithm itself and we wanted to compare our results with SVM which also has such a hyper-parameter.

We also compare both mentioned PBSC algorithms to Algorithm 4 PBSC-L that just consists of minimizing $\zeta_S^{Q,lin}$ for the linear margin loss function of the form $\zeta(\alpha) = \alpha + 1$ given by Equation (7.23). The comparison with the latter is only to point out the need for a non-linear margin loss function. Note that *PBSC-N* has two hyper-parameters (C and q) to tune whereas *PBSC-A* needs only one (q). The SVM also needs to tune only one hyper-parameter: the soft-margin C parameter. All these results are shown in Table 8.1. For all algorithms, we used the standard RBF kernel $k_{\text{RBF}}(x, x') = \exp(-\gamma\|x - x'\|)$ and the sigmoid kernel $k_{\text{SIG}}(x, x') = \tanh(s\, x \cdot x' + d)$. The RBF kernel adds the extra hyper-parameter γ and the sigmoid kernel adds the extra hyper-parameters s and d to each algorithm. All hyper-parameters C, q, γ, s, and d are determined by performing 10-fold cross validation on the training data. We performed experiments on 22 data sets (see Table A.1 in Appendix 9.2) that, except for MNIST, were taken from the UCI repository. Each data set was randomly partitioned into a training set S of size $|S|$ and a testing set T of size $|T|$.

Rbf kernel				
DataSet-Name	**SVM**	**PBSC-A**	**PBSC-N(Catoni)**	**PBSC-L**
Adult	0.158	**0.156**	0.160	0.193
BreastCancer	**0.038**	0.044	**0.038**	0.144
Credit-A	0.190	**0.140**	0.173	0.200
Glass	**0.150**	**0.150**	0.168	0.187
Heberman	**0.267**	0.280	**0.267**	**0.267**
Heart	**0.197**	0.204	0.218	0.238
Ionosphere	0.057	**0.040**	**0.040**	0.326
Letter:AB	**0.001**	**0.001**	**0.001**	0.038
Letter:DO	0.014	**0.011**	0.012	0.069
Letter:OQ	0.016	0.016	**0.014**	0.123
Liver	0.286	**0.280**	0.286	0.349
MNIST:0vs8	**0.003**	0.004	0.004	0.031
MNIST:1vs7	0.014	0.008	**0.007**	0.161
MNIST:1vs8	0.011	**0.010**	0.011	0.292
MNIST:2vs3	0.020	**0.019**	0.020	0.114
Mushroom	**0.000**	**0.000**	**0.000**	0.022
Ringnorm	0.015	**0.013**	**0.013**	0.103
sonar	0.154	**0.125**	0.192	0.490
Tic-tac-toe	**0.015**	0.019	0.052	0.365
Usvotes	0.075	**0.065**	**0.065**	0.140
Waveform	0.068	0.068	**0.066**	0.143
Wdbc	**0.042**	0.049	0.074	0.180

Table 8.1: The results of Algorithm 3 for the minimization of any proposed theorems that do not depend on KL term (PBSC-A1) and Algorithm 5 for the minimization of Theorem 6.3.1(PBSC-N (Catoni)), in comparison with SVM and PBSC-L.

As Table 8.1 shows, when using the RBF kernel, *PBSC-A* is very competitive with *PBSC-N(6.3.1)* and SVM, and outperforms PBSC-L. However, the differences between the three first algorithms are never statistically significant. To determine when a difference of empirical risk measured on set T is statistically significant, we have used the test set bound method (see Chapter 3) (based on the binomial tail inversion) with a

sigmoid kernel		
DataSet-Name	**SVM**	**PBSC-A**
Adult	0.163	**0.157**
BreastCancer	**0.038**	**0.038**
Credit-A	0.190	**0.170**
Glass	**0.355**	0.411
Heberman	**0.273**	**0.273**
Heart	**0.184**	0.197
Ionosphere	0.126	**0.091**
Letter:AB	0.009	**0.005**
Letter:DO	**0.022**	0.028
Letter:OQ	**0.018**	0.039
Liver	**0.400**	**0.400**
MNIST:0vs8	0.007	**0.0031**
MNIST:1vs7	0.012	**0.007**
MNIST:1vs8	**0.014**	0.015
MNIST:2vs3	**0.025**	0.031
Mushroom	**0.000**	0.010
Ringnorm	**0.020**	0.035
sonar	0.250	**0.183**
Tic-tac-toe	**0.023**	0.159
Usvotes	0.070	**0.065**
Waveform	**0.067**	**0.067**
Wdbc	**0.366**	**0.366**

Table 8.2: Results of *PBSC-A* in comparison with SVM using sigmoid kernel.

confidence level of 95%. Moreover, PBSC-L is statistically significantly worse than the others on 16 of the 22 datasets (for more details see Appendix C).

The Results using sigmoid kernel are shown in Table 8.2. Unlike the RBF kernel, the sigmoid kernel is not positive semi-definite for certain parameter values. In this case, the standard SVM algorithm might not converge to a solution (like the popular SVM-Light implementation). In our experiments, we used the LIBSVM implementation of [8] because it returns a solution even if the kernel is indefinite. In this context, it turns out that *PBSC-A* and LIBSVM are competitive.

Table 8.3: Mean and standard deviation (in parentheses) of the empirical risk across 20 partitions.

Dataset	**Linear SVM**		**k-NN**		**PBSC-A**	
Aural Sonar	**0.1425**	(0.694)	0.1825	(0.597)	0.1500	(0.827)
Voting	0.0534	(0.193)	0.0546	(0.174)	**0.0529**	(0.184)
Yeast-5-7	**0.2688**	(0.622)	0.3063	(0.580)	0.2975	(0.668)
Yeast-5-12	**0.1075**	(0.482)	0.1275	(0.439)	0.1088	(0.598)

To pursue the exploration with indefinite similarity measures (see Table 8.3), we have executed *PBSC-A* on four binary data sets referenced by [9, 10]. Since these data sets provide directly a similarity measure between each pair of examples, we used these similarities for the *PBSC-A* algorithm. To compare our results, we followed the same experimental framework as [9, 10] and computed the mean and standard deviation of the empirical risk across 20 test/training standardized partitions. Table 8.3 shows that *PBSC-A* is competitive with the Linear SVM using similarities as features and is better than the k-Nearest Neighbor using similarities as a measure of distance. Note that [10] suggests an algorithm that has generally better achievements on these data sets, however, these results are obtained by substituting a "surrogate kernel function" with the real similarity function that one wants to use.

Chapter 9

Conclusion and Future Work

9.1 Conclusion

The initial attempts in this work started by the results of research work in [[31, 19]] where we showed that the PAC-Bayesian theory is a good starting point for designing learning algorithms. Inspired by [[31]], who specialized the risk bound of [33] to SCM and proposed a learning strategy for SCM based on the minimization of the mentioned bound, and also by the success of kernel methods such as SVM, we proposed here a PAC-Bayes sample-compression approach to kernel methods that can accommodate any bounded similarity function.

In this thesis, we showed that the SVM classifier is actually a particular case of a (weighted) majority vote of sample-compressed classifiers where the compression sequence of each classifier consists of at most a single training example. Inspired by the work in [[19]] on general loss bounds for stochastic classifiers, we proposed different PAC-Bayes risk bounds for majority votes of sample-compressed classifiers which are valid for any bounded similarity measure of input examples. Consequently, we also applied the proposed bounds to the class of linear classifiers of similarity-based features that were studied by [9]. For the class of indefinite similarity measures, the risk bound proposed by [9] becomes trivial (and useless) in the limit where each training example is used for a prototype. In contrast, the risk bounds presented here did not suffer from such a limitation.

With the exception of the risk bounds of Theorem 6.3.3 and Theorem 6.3.5, for each proposed risk bound, we provided a learning algorithm that minimizes that bound. The first group of the proposed PAC-Bayes risk bounds depend on the KL divergence between the prior and the posterior over the set of sample-compressed classifiers and, consequently, we showed that the corresponding bound-minimizing learning algorithm is KL-regularized. The second group of proposed PAC-Bayes risk bounds have the unusual property of having no KL divergence when the posterior is aligned with the prior in some precise way that we defined in Chapter 7. Consequently, we showed that minimizing these risk upper bounds just amount to minimizing the proposed empirical loss under the constraint that the posterior is kept aligned with the prior.

When a positive semi-definite kernel is used, our experiments indicated that the proposed algorithms are very competitive with the SVM. Good empirical results are also obtained when the proposed algorithms are used with a non positive semi-definite kernel. Finally, we showed that the proposed algorithms are also competitive with the best similarity-based learning algorithms proposed by [9].

9.2 Future Work

Considering the results we obtained in this thesis, one can investigate the following future avenues:

Utilizing the risk bounds for model selection: The first important task that should be investigated is to see if the risk bounds that we proposed can successfully perform model selection (i.e., the selection of the hyper-parameters of our proposed algorithms in a similar way as in [[31]]). This could potentially eliminate the need to perform the time-consuming cross-validation method for selecting the model and provide better guarantees on the generalization error of classifiers output by learning algorithms. Unfortunately, our preliminarily investigation in that direction indicates that we are presently far from achieving such an objective.

Improve the bounds: It is worthwhile to theoretically and empirically compare the proposed bounds of Chapter 6 and investigate to see if we can improve these bounds to obtain tighter bounds which might lead to better learning algorithms.

Applying the proposed risk bounds in Chapter 6 to the majority vote of sc-classifiers having the compression set size of more than one: In this thesis, we proposed the PAC-Bayes sample compression bounds and then we applied the proposed bounds to construct algorithms that deal with the majority vote of sample compressed classifiers having a compression size of at most one. It would be worthwhile to extend our framework to the sample compressed classifiers where the compression sequence of each classifier consists of more than one training example.

Using the proposed algorithms in the unsupervised or semi-supervised learning framework: In many real world learning problems, obtaining sufficient amount of labeled data for a training algorithm is costly and time-consuming while an enormous amount of unlabeled data is available. Unsupervised and semi-supervised learning tackle these problems. Therefore, it would be worthwhile to investigate if we could apply the proposed algorithms of this thesis to these learning framework as well.

Transductive bounds: In this thesis, we have dealt with the *inductive learning* in which the learning algorithm is given a finite set of labeled examples (training set) from which a function (classifier) is constructed and this function (classifier) is then used to label a new unseen example. On the other hand, [54] pointed out that in many real life cases we are dealing with the problem in which a learning algorithm is given a set of labeled examples (training set) together with the set of unlabeled examples. The goal is to construct a function that labels the unlabeled examples. As denoted by [54], transduction is an easier task than induction therefore the above solution to the transductive problem is similar to transforming the problem to a more difficult one. However, although transduction seems to be an easier task there have not been many useful algorithms for it. Thus, it would be interesting to derive a risk bound similar to the proposed bound of Chapter 6 for the transductive framework. This may lead to algorithms for the transductive learning. This is an ongoing work.

Bibliography

[1] J.Y. Audibert. *Théorie statistique de l'apprentissage: une approche PAC-bayésienne*. PhD thesis, 2004.

[2] A. Ben-Hur and J. Weston. A user's guide to support vector machines. *Methods in Molecular Biology*, 609:223–239, 2010.

[3] B.E. Boser, I.M. Guyon, and V.N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

[4] L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[5] L. Breiman. Arcing classifier (with discussion and a rejoinder by the author). *The annals of statistics*, 26(3):801–849, 1998.

[6] O. Catoni. PAC-Bayes supervised classification: The thermodynamics of statistical learning, volume 56 of. *IMS Lecture Notes Monograph Series*, 2007.

[7] Olivier Catoni. *PAC-Bayesian supervised classification: the thermodynamics of statistical learning*. Monograph series of the Institute of Mathematical Statistics, http://arxiv.org/abs/0712.0248, December 2007.

[8] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[9] Yihua Chen, Eric K. Garcia, Maya R. Gupta, Ali Rahimi, and Luca Cazzanti. Similarity-based classification: Concepts and algorithms. *Journal of Machine Learning Research*, 10:747–776, 2009.

[10] Yihua Chen, Maya R. Gupta, and Benjamin Recht. Learning kernels from indefinite similarities. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, pages 145–152, New York, NY, USA, 2009. ACM.

[11] V. Chvátal. A greedy heuristic for the set covering problem. *Mathematics of Operations Research*, 4:233–235, 1979.

[12] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[13] T.M. Cover, J.A. Thomas, J. Wiley, et al. *Elements of information theory*, volume 6. Wiley Online Library, 1991.

[14] Sally Floyd and Manfred Warmuth. Sample compression, learnability, and the Vapnik-Chervonenkis dimension. *Machine Learning*, 21(3):269–304, 1995.

[15] Y. Freund and R.E. Schapire. Experiments with a new boosting algorithm. In *Machine Learning-International Workshop Then Conference-*, pages 148–156. Morgan Kaufmann Publishers, INC., 1996.

[16] Michael R. Garey and David S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness*. Freeman, New York, NY, 1979.

[17] P. Germain, A. Lacasse, F. Laviolette, and M. Marchand. Pac-bayesian learning of linear classifiers. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 353–360. ACM, 2009.

[18] Pascal Germain, Alexandre Lacasse, François Laviolette, and Mario Marchand. A PAC-Bayesian risk bound for general loss functions. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 449–456. MIT Press, Cambridge, MA, 2007.

[19] Pascal Germain, Alexandre Lacasse, François Laviolette, Mario Marchand, and Sara Shanian. From PAC-Bayes bounds to KL regularization. In J. Lafferty and C. Williams, editors, *Advances in Neural Information Processing Systems 22*, Cambridge, MA, 2009. MIT Press.

[20] Pascal Germain, Alexandre Lacoste, François Laviolette, Mario Marchand, and Sara Shanian. A PAC-Bayes Sample Compression Approach to Kernel Methods. In *Proceedings of the 28th International Conference on Machine Learning*, Bellevue, WA, USA, June 2011.

[21] Thore Graepel, Ralf Herbrich, and John Shawe-Taylor. PAC-Bayesian compression bounds on the prediction error of learning algorithms for classification. *Machine Learning*, 59:55–76, 2005.

[22] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422, 2002.

[23] L.K. Hansen and P. Salamon. Neural network ensembles. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(10):993–1001, 1990.

[24] D. Haussler. Quantifying inductive bias: AI learning algorithms and Valiant's learning framework. *Artificial Intelligence*, 36:177–221, 1988.

[25] T.K. Ho. The random subspace method for constructing decision forests. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(8):832–844, 1998.

[26] T. Hothorn and B. Lausen. Double-bagging: Combining classifiers by bootstrap aggregation. *Pattern Recognition*, 36(6):1303–1309, 2003.

[27] N. Japkowicz and M. Shah. *Evaluating Learning Algorithms: A Classification Perspective.* Cambridge Univ Pr, 2011.

[28] R.R. Johnson and P. Kuby. *Elementary statistics.* Brooks/Cole, 2011.

[29] Michael J. Kearns and Umesh V. Vazirani. *An Introduction to Computational Learning Theory.* MIT Press, Cambridge, Massachusetts, 1994.

[30] John Langford. Tutorial on practical prediction theory for classification. *Journal of Machine Learning Research*, 6:273–306, 2005.

[31] F. Laviolette, M. Marchand, M. Shah, and S. Shanian. Learning the set covering machine by bound minimization and margin-sparsity trade-off. *Machine learning*, 78(1):175–201, 2010.

[32] François Laviolette and Mario Marchand. PAC-Bayes risk bounds for sample-compressed Gibbs classifiers. *Proceedings of the 22nth International Conference on Machine Learning (ICML)*, pages 481–488, 2005.

[33] François Laviolette and Mario Marchand. PAC-Bayes risk bounds for stochastic averages and majority votes of sample-compressed classifiers. *Journal of Machine Learning Research*, 8:1461–1487, 2007.

[34] François Laviolette, Mario Marchand, and Mohak Shah. Margin-sparsity trade-off for the set covering machine. *Proceedings of the* 16^{th} *European Conference on Machine Learning (ECML 2005); Lecture Notes in Artificial Intelligence*, 3720:206–217, 2005.

[35] François Laviolette, Mario Marchand, and Mohak Shah. A PAC-Bayes approach to the set covering machine. *Proceedings of the 2005 conference on Neural Information Processing Systems (NIPS 2005)*, 2006.

[36] N. Littlestone and M. Warmuth. Relating data compression and learnability. Technical report, University of California Santa Cruz, Santa Cruz, CA, 1986.

[37] M. Marchand. Ift-65764 apprentissage automatique (machine learning). *Notes de cours IFT-65764, Université Laval.*

[38] Mario Marchand and John Shawe-Taylor. Learning with the set covering machine. *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001)*, pages 345–352, 2001.

[39] Mario Marchand and John Shawe-Taylor. The set covering machine. *Journal of Machine Learning Reasearch*, 3:723–746, 2002.

[40] Mario Marchand and Marina Sokolova. Learning with decision lists of data-dependent features. *Journal of Machine Learning Reasearch*, 6:427–451, 2005.

[41] Andreas Maurer. A note on the PAC-Bayesian theorem. *CoRR*, cs.LG/0411099, 2004.

[42] David McAllester. Some PAC-Bayesian theorems. *Machine Learning*, 37:355–363, 1999.

[43] David McAllester. PAC-Bayesian stochastic model selection. *Machine Learning*, 51:5–21, 2003.

[44] David A. McAllester. PAC-Bayesian model averaging. In *COLT*, pages 164–170, 1999.

[45] J. Platt. *Advances in Kernel Methods: Support Vector Learning*, chapter Fast training of SVMs using sequential minimal optimization, pages 185–208. MIT press, Cambridge, MA, 1999.

[46] Robert E. Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26:1651–1686, 1998.

[47] Matthias Seeger. PAC-Bayesian generalization bounds for Gaussian processes. *Journal of Machine Learning Research*, 3:233–269, 2002.

[48] M. Shah. Risk bounds for classifier evaluation: Possibilities and challenges. In *Proceedings of the 3rd Workshop on Evaluation Methods for Machine Learning. in conjunction with 25th International Conference on Machine Learning Helsinki, Finland*, 2008.

[49] M. Shah and S. Shanian. Hold-out risk bounds for classifier performance evaluation. In *Proceedings of the 4th Workshop on Evaluation Methods for Machine Learning. in conjunction with 26th International Conference on Machine Learning, Montreal, Canada*, 2008.

[50] J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis*. Cambridge Univ Pr, 2004.

[51] K.D. Smith. *Ridge regression: constrained optimization with applications in design of experiments*. PhD thesis, University of California, Riverside, 1981.

[52] L. G. Valiant. A theory of the learnable. *Communications of the Association of Computing Machinery*, 27(11):1134–1142, November 1984.

[53] V. Vapnik and A. Chervonenkis. Uniform convergence of frequencies of occurence of events to their probabilities. In *Dokl. Akad. Nauk SSSR*, volume 181, pages 915–918, 1968.

[54] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley, New York, NY, 1998.

[55] G.I. Webb. Multiboosting: A technique for combining boosting and wagging. *Machine learning*, 40(2):159–196, 2000.

[56] I.H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

Appendix A

Data Sets

In this thesis, we performed experiments on a variety of data sets. The following tables show some comprehensive details about used data sets.

Dataset			
Name	$\lvert T\rvert$	$\lvert S\rvert$	n
Adult	10000	1809	14
BreastCancer	340	343	9
Credit-A	300	353	15
Glass	107	107	9
Haberman	150	144	3
Heart	147	150	13
Ionosphere	175	176	34
Letter:AB	1055	500	16
Letter:DO	1058	500	16
Letter:OQ	1036	500	16
Liver	175	170	6
MNIST:0vs8	1916	500	784
MNIST:1vs7	1922	500	78
MNIST:1vs8	1936	500	784
MNIST:2vs3	1905	500	784
Mushroom	4062	4062	22
Ringnorm	3700	3700	20
sonar	104	104	60
sonar-mixed	104	104	60
Tic-tac-toe	479	479	9
Usvotes	200	235	16
Waveform	4000	4000	21
Wdbc	284	285	30

Table A.1: Data Set Description

Appendix B

Empirical Results of Chapter 3

Here we present the results of the experiments that have been done in Chapter 3 on more data sets.

Data-Set	**A**	R_T	B_l	B_u	CI_l	CI_u
Haberman	SVM	0.273	0.203	0.352	0.200	0.345
	Ada	0.233	0.185	0.330	0.163	0.302
	DT	0.273	0.203	0.352	0.200	0.345
	DL	0.273	0.203	0.352	0.200	0.345
	NB	0.246	0.180	0.323	0.175	0.316
	SCM	0.253	0.185	0.330	0.182	0.323
HeartS	SVM	0.204	0.142	0.278	0.137	0.270
	Ada	0.272	0.202	0.351	0.198	0.345
	DT	0.197	0.136	0.270	0.13	0.262
	DL	0.156	0.101	0.225	0.096	0.215
	NB	0.136	0.085	0.202	0.079	0.192
	SCM	0.190	0.130	0.263	0.125	0.254
Sonar	SVM	0.116	0.061	0.194	0.053	0.178
	Ada	0.135	0.076	0.217	0.067	0.202
	DT	0.365	0.099	0.251	0.270	0.459
	DL	0.281	0.197	0.378	0.192	0.369
	NB	0.262	0.180	0.358	0.175	0.348
	SCM	0.310	0.223	0.409	0.219	0.400
SonarM	SVM	0.182	0.113	0.270	0.106	0.257
	Ada	0.153	0.090	0.237	0.082	0.223
	DT	0.365	0.273	0.465	0.270	0.459
	DL	0.221	0.145	0.313	0.139	0.302
	NB	0.269	0.186	0.365	0.182	0.355
	SCM	0.403	0.308	0.504	0.306	0.499

Table B.1: Results of various classifiers on UCI Datasets

Data-Set	**A**	R_T	B_l	B_u	CI_l	CI_u
BreastCancer	SVM	0.038	0.020	0.063	0.0172	0.058
	Ada	0.049	0.029	0.078	0.0255	0.072
	DT	0.061	0.038	0.092	0.035	0.086
	DL	0.046	0.026	0.074	0.023	0.068
	NB	0.046	0.026	0.074	0.023	0.068
	SCM	0.037	0.020	0.063	0.016	0.057
Wdbc	SVM	0.070	0.043	0.106	0.039	0.100
	Ada	0.042	0.022	0.072	0.018	0.065
	DT	0.052	0.029	0.085	0.025	0.078
	DL	0.059	0.035	0.094	0.031	0.086
	NB	0.049	0.027	0.081	0.023	0.074
	SCM	0.056	0.032	0.089	0.0287	0.083
Tic-Tac-Toe	SVM	0.062	0.042	0.088	0.039	0.084
	Ada	0.016	0.007	0.326	0.004	0.027
	DT	0.135	0.106	0.169	0.103	0.166
	DL	0.048	0.030	0.071	0.0284	0.067
	NB	0.340	0.297	0.384	0.296	0.383
	SCM	0.106	0.080	0.137	0.077	0.134
Ionosphere	SVM	0.045	0.019	0.088	0.0136	0.076
	Ada	0.091	0.053	0.144	0.047	0.134
	DT	0.091	0.053	0.144	0.047	0.134
	DL	0.142	0.094	0.203	0.089	0.194
	NB	0.16	0.109	0.222	0.104	0.215
	SCM	0.24	0.178	0.310	0.175	0.304

Table B.2: Results of various classifiers on UCI Datasets

Data-Set	**A**	R_T	B_l	B_u	CI_l	CI_u
Letter-AB	SVM	0.001	0	0.005	-0.0009	0.002
	Ada	3.8e-3	0.001	0.009	1.148	0.007
	DT	0.017	0.010	0.026	0.009	0.024
	DL	0.016	0.009	0.025	0.008	0.023
	NB	0.080	0.064	0.098	0.063	0.096
	SCM	0.029	0.020	0.041	0.0186	0.039
Letter-OQ	SVM	0.010	0.005	0.018	0.003	0.016
	Ada	0.043	0.031	0.057	0.0303	0.055
	DT	0.077	0.061	0.095	0.060	0.093
	DL	0.055	0.041	0.070	0.0408	0.069
	NB	0.157	0.135	0.180	0.134	0.179
	SCM	0.109	0.090	0.129	0.089	0.128
Letter-DO	SVM	0.013	0.007	0.022	0.006	0.019
	Ada	0.024	0.016	0.035	0.0145	0.033
	DT	0.061	0.047	0.077	0.046	0.075
	DL	0.054	0.042	0.070	0.040	0.067
	NB	0.080	0.064	0.098	0.063	0.096
	SCM	0.061	0.047	0.077	0.046	0.075
Mushroom	SVM	0	0	0.0009	0.0	0.0
	Ada	0	0	0.0009	0.0	0.0
	DT	0	0	0.0009	0.0	0.0
	DL	0	0	0.0009	0.0	0.0
	NB	0.091	0.083	0.101	0.0534	0.068
	SCM	0.025	0.020	0.304	0.020	0.029

Table B.3: Results of various classifiers on UCI Datasets

Appendix C

Calculating the Test Set Bound for Table 8.1 of Chapter 7

Here, we present the lower and upper intervals generated from computing the lower (B_l) and upper (B_u) risk bounds of Theorems 3.1.2 and 3.1.1 of Section 3.1 with $\delta = 0.05$ for Table 8.1.

Data-Set	**A**	R_T	B_l	B_u
Adult	SVM	0.158	0.152	0.146
	PBSC-A	0.156	0.150	0.162
	PBSC-N	0.160	0.153	0.166
	PBSC-L	0.193	0.186	0.199
BreastCancer	SVM	0.038	0.022	0.060
	PBSC-A	0.044	0.027	0.067
	PBSC-N	0.038	0.022	0.060
	PBSC-L	0.144	0.027	0.067
Credit-A	SVM	0.190	0.153	0.231
	PBSC-A	0.140	0.108	0.177
	PBSC-N	0.173	0.138	0.213
	PBSC-L	0.200	0.162	0.241
Glass	SVM	0.150	0.103	0.228
	PBSC-A	0.150	0.103	0.228
	PBSC-N	0.168	0.111	0.239
	PBSC-L	0.187	0.135	0.270
Heberman	SVM	0.267	0.213	0.339
	PBSC-A	0.280	0.220	0.340
	PBSC-N	0.267	0.213	0.339
	PBSC-L	0.267.	0.213	0.339
Heart	SVM	0.197	0.144	0.259
	PBSC-A	0.204	0.187	0.310
	PBSC-N	0.218	0.169	0.288
	PBSC-L	0.238	0.175	0.295
Ionosphere	SVM	0.057	0.031	0.094
	PBSC-A	0.040	0.018	0.073
	PBSC-N	0.040	0.018	0.073
	PBSC-L	0.326	0.272	0.394
Letter-AB	SVM	0.001	0.0003	0.005
	PBSC-A	0.001	0.0003	0.005
	PBSC-N	0.001	0.0003	0.005
	PBSC-L	0.038	0.029	0.050

Table C.1: The lower and upper risk bounds of Theorems 3.1.2 and 3.1.1 of Section 3.1 with $\delta = 0.05$.

Data-Set	**A**	R_T	B_l	B_u
Letter-DO	SVM	0.014	0.008	0.021
	PBSC-A	0.011	0.006	0.018
	PBSC-N	0.012	0.007	0.014
	PBSC-L	0.069	0.034	0.056
Letter-OQ	SVM	0.016	0.010	0.024
	PBSC-A	0.016	0.010	0.024
	PBSC-N	0.014	0.008	0.022
	PBSC-L	0.0123	0.107	0.0141
Liver	SVM	0.286	0.235	0.353
	PBSC-A	0.280	0.224	0.341
	PBSC-N	0.0.286	0.235	0.353
	PBSC-L	0.349	0.294	0.418
MNIST:0vs8	SVM	0.003	0.001	0.006
	PBSC-A	0.004	0.002	0.007
	PBSC-N	0.004	0.002	0.007
	PBSC-L	0.031	0.025	0.038
MNIST:1vs7	SVM	0.014	0.009	0.019
	PBSC-A	0.008	0.005	0.012
	PBSC-N	0.007	0.004	0.011
	PBSC-L	0.161	0.147	0.175
MNIST:1vs8	SVM	0.011	0.007	0.016
	PBSC-A	0.010	0.006	0.014
	PBSC-N	0.011	0.007	0.016
	PBSC-L	0.292	0.275	0.309
MNIST:2vs3	SVM	0.020	0.014	0.026
	PBSC-A	0.019	0.0145	0.025
	PBSC-N	0.020	0.014	0.026
	PBSC-L	0.114	0.103	0.127

Table C.2: The lower and upper risk bounds of Theorems 3.1.2 and 3.1.1 of Section 3.1 with $\delta = 0.05$.

Data-Set	**A**	R_T	B_l	B_u
Mushroom	SVM	0.000	0.000	0.0007
	PBSC-A	0.000	0.000	0.0007
	PBSC-N	0.000	0.000	0.0007
	PBSC-L	0.022	0.018	0.026
Ringnorm	SVM	0.015	0.011	0.018
	PBSC-A	0.013	0.010	0.018
	PBSC-N	0.013	0.010	0.018
	PBSC-L	0.103	0.007	0.013
sonar	SVM	0.154	0.098	0.224
	PBSC-A	0.125	0.075	0.191
	PBSC-N	0.192	0.131	0.267
	PBSC-L	0.490	0.396	0.565
Tic-tac-toe	SVM	0.015	0.008	0.029
	PBSC-A	0.019	0.009	0.032
	PBSC-N	0.052	0.034	0.069
	PBSC-L	0.365	0.328	0.403
Usvotes	SVM	0.075	0.046	0.113
	PBSC-A	0.065	0.038	0.101
	PBSC-N	0.0.65	0.038	0.113
	PBSC-L	0.140	0.101	0.186
Waveform	SVM	0.068	0.061	0.074
	PBSC-A	0.068	0.061	0.074
	PBSC-N	0.066	0.059	0.072
	PBSC-L	0.143	0.133	0.074
Wdbc	SVM	0.042	0.024	0.067
	PBSC-A	0.049	0.030	0.075
	PBSC-N	0.074	0.053	0.108
	PBSC-L	0.180	0.146	0.225

Table C.3: The lower and upper risk bounds of Theorems 3.1.2 and 3.1.1 of Section 3.1 with $\delta = 0.05$.