# Learning a Multiview Weighted Majority Vote Classifier : Using PAC-Bayesian Theory and Boosting

Anil Goyal

▶ To cite this version:

Anil Goyal. Learning a Multiview Weighted Majority Vote Classifier : Using PAC-Bayesian Theory and Boosting. Data Structures and Algorithms [cs.DS]. Université de Lyon, 2018. English. NNT : 2018LYSES037 . tel-01881069v2

HAL Id: tel-01881069
https://hal.science/tel-01881069v2

Submitted on 4 Oct 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N°d'ordre NNT : 2018LYSES037

**THESE de DOCTORAT DE L'UNIVERSITE DE LYON**
opérée au sein de
**Université Jean Monnet**

**Ecole Doctorale** N° 488
**Sciences, Ingénierie, Santé**

**Spécialité / discipline de doctorat** :
**Machine Learning / Informatique**

Soutenue publiquement le 23/10/2018, par :
**Anil Goyal**

# Learning a Multiview Weighted Majority Vote Classifier: Using PAC-Bayesian Theory and Boosting

Devant le jury composé de :

**Habrard, Amaury** Professeur, Université Jean Monnet Président

**Janodet, Jean-Christophe** Professeur, Université d'Évry Rapporteur
**Capponi, Cécile** Maître de Conférences, Université d'Aix-Marseille Rapportrice

**Amini, Massih-Reza** Professeur, Université de Grenoble-Alpes, Directeur de thèse
**Morvant, Emilie** Maître de Conférences, Université JeanMonnet, Co-directrice de thèse

N° d'ordre NNT : 2018LYSES037

THÈSE de DOCTORAT DE L'UNIVERSITÉ DE LYON

opérée au sein de

**Laboratoire Hubert Curien et Laboratoire d'Informatique de Grenoble**

**Ecole Doctorale** ED SIS 488

**(École Doctorale Sciences, Ingénierie, Santé)**

**Spécialité de doctorat:** Machine Learning

Discipline: **Informatique**

Soutenue publiquement le 23/10/2018, par

**Anil Goyal**

# Learning a Multiview Weighted Majority Vote Classifier:

## *Using PAC-Bayesian Theory and Boosting*

Devant le jury composé de:

| | | |
|---|---|---|
| Jean-Christophe Janodet | Professeur, Université d'Évry | Rapporteur |
| Cécile Capponi | Maître de Conférences, Aix-Marseille Université | Rapportrice |
| Amaury Habrard | Professeur, Université Jean Monnet | Examinateur, Président |
| Massih-Reza Amini | Professeur, Université de Grenoble-Alpes | Directeur |
| Emilie Morvant | Maître de Conférences, Université Jean Monnet | Co-Directrice |

# ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to all the people who have contributed in some way to the work presented in this thesis. First of all, I would like to thank Prof. Jean-Christophe Janodet and Dr. Cécile Capponi for accepting to review my thesis. Their insightful remarks allowed me to improve the quality of this manuscript. I am also grateful to Prof. Amaury Habrard for agreeing to be an examinateur.

I express my deepest gratitude and indebtedness to my thesis supervisors Prof. Massih-Reza Amini and Dr. Emilie Morvant for accepting me into their research teams and seeing potential in me. I am grateful for their continuous support and advice during past three years. I would like to thank them for encouraging my research and for allowing me to grow as a researcher. During my thesis, I was fortunate to have an opportunity to work with Dr. Pascal Germain. I would like to thank him for being a great research collaborator and for suggesting me interesting directions for my work.

I thank my fellow lab mates from Laboratoire Hubert Curien and Laboratoire d'Informatique de Grenoble who are more than just colleagues. I learned a lot of things from them and shared many beautiful moments during lunch and coffee breaks.

Last but not least, I would like to thank from the depths of my heart my family back in India: my father Dr. Rakesh Kumar Goyal, my mother Mrs. Nirmala Goyal, my brother Dr. Divyadeep Goyal for their unconditional support and love throughout my years of study. Thank you for everything you did and continue doing for me.

Thanking You,
Anil Goyal

# ABSTRACT

With tremendous generation of data, we have data collected from different information sources having heterogeneous properties, thus it is important to consider these representations or views of the data. This problem of machine learning is referred as multiview learning. It has many applications for e.g. in medical imaging, we can represent human brain with different set of features for example MRI, t-fMRI, EEG, etc. In this thesis, we focus on supervised multiview learning, where we see multiview learning as combination of different view-specific classifiers or views. Therefore, according to our point of view, it is interesting to tackle multiview learning issue through PAC-Bayesian framework. It is a tool derived from statistical learning theory studying models expressed as majority votes. One of the advantages of PAC-Bayesian theory is that it allows to directly capture the trade-off between accuracy and diversity between voters, which is important for multiview learning. The first contribution of this thesis is extending the classical PAC-Bayesian theory (with a single view) to multiview learning (with more than two views). To do this, we considered a two-level hierarchy of distributions over the view-specific voters and the views. Based on this strategy, we derived PAC-Bayesian generalization bounds (both probabilistic and expected risk bounds) for multiview learning. From practical point of view, we designed two multiview learning algorithms based on our two-level PAC-Bayesian strategy. The first algorithm is a one-step boosting based multiview learning algorithm called as `PB-MVBoost`. It iteratively learns the weights over the views by optimizing the multiview $\mathcal{C}$-Bound which controls the trade-off between the accuracy and the diversity between the views. The second algorithm is based on late fusion approach (referred as $\texttt{Fusion}^{\texttt{all}}_{\texttt{Cq}}$) where we combine the predictions of view-specific classifiers using the PAC-Bayesian algorithm `CqBoost` proposed by Roy et al. Finally, we show that minimization of classification error for multiview weighted majority vote is equivalent to the minimization of Bregman divergences. This allowed us to derive a parallel update optimization algorithm (referred as $\texttt{M}\omega\texttt{MvC}^2$) to learn our multiview weighted majority vote.

# TABLE OF CONTENTS

# LIST OF TABLES

**TABLE** **Page**

# LIST OF FIGURES

**FIGURE** **Page**

# List of Notations

| Notation | Description |
|---|---|
| $\mathcal{X} \in \mathbb{R}^d$ | Input space of $d$ dimension |
| $\mathcal{Y} = \{-1, +1\}$ | Binary output space |
| $(x, y) \in \mathcal{X} \times \mathcal{Y}$ | An example $x$ and its label $y$ |
| $S = \{(x_i, y_i)\}_{i=1}^m$ | Labeled training sample of size $m$ |
| $\mathcal{D}$ | Joint distribution over $\mathcal{X} \times \mathcal{Y}$ |
| $\mathcal{D}_{\mathcal{X}}$ | Marginal distribution on $\mathcal{X}$ |
| $(\mathcal{D})^m$ | Distribution of a $m$-sample |
| $\mathcal{H}$ | Hypothesis space consisting of set of classifiers $h : \mathcal{X} \to \mathcal{Y}$ |
| $P$ | Prior distribution over $\mathcal{H}$ |
| $Q$ | Posterior distribution over $\mathcal{H}$ |
| $Q_S$ | Posterior distribution over $\mathcal{H}$ after observing learning sample $S$ |
| $\mathrm{KL}(Q\|\|P)$ | Kullback-Leibler divergence between $Q$ and $P$ distributions |
| $G_Q$ | Gibbs Classifier |
| $B_Q$ | Majority vote Classifier |
| $R_{\mathcal{D}}(.)$ | True risk of a classifier |
| $R_S(.)$ | Empirical risk of a classifier |
| $VC(\mathcal{H})$ | VC-dimension of hypothesis space $\mathcal{H}$ |
| $\mathfrak{R}_S(\mathcal{H})$ | Empirical Rademacher complexity of hypothesis space $\mathcal{H}$ |
| $\mathfrak{R}_{\mathcal{D}}(\mathcal{H})$ | Rademacher complexity of hypothesis space $\mathcal{H}$ |
| $\mathbb{1}_{[p]}$ | Indicator function equal to 1 if predicate $p$ is true and 0 otherwise |
| $\langle \cdot, \cdot \rangle$ | Dot product |

List of Notations for single view learning

| Notation | Description |
|---|---|
| $V$ | Number of views |
| $\mathcal{V}$ | Set of $V$ views |
| $\mathcal{X}_v \in \mathbb{R}^{d_v}$ | Input space for view $v$ |
| $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_V$ | Joint input space |
| $\mathcal{Y} = \{-1, 1\}$ | Binary output space |
| $\mathbf{x} = (x^1, x^2, \dots, x^V)$ | Multiview example described by $V$ views |
| $y \in \mathcal{Y}$ | Label of an example |
| $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$ | Labeled training sample of size $m$ |
| $\mathcal{D}$ | Joint distribution over $\mathcal{X} \times \mathcal{Y}$ |
| $\mathcal{D}_{\mathcal{X}}$ | Marginal distribution on $\mathcal{X}$ |
| $(\mathcal{D})^m$ | Distribution of a $m$-sample |
| $\mathcal{H}_v$ | View-specific hypothesis space consisting of set of classifiers $h_v : \mathcal{X}_v \to \mathcal{Y}$ |
| $n_v$ | Number of view-specific classifiers for view $v \in \mathcal{V}$ |
| $P_v$ | Prior distribution over view-specific hypothesis space $\mathcal{H}_v$ |
| $Q_v$ | Posterior distribution over $\mathcal{H}_v$ |
| $Q_{v,S}$ | Posterior distribution over $\mathcal{H}_v$ after observing the learning sample $S$ |
| $\pi$ | Hyper-prior distribution over the set of views $\mathcal{V}$ |
| $\rho$ | Hyper-posterior distribution over the set of views $\mathcal{V}$ |
| $\rho_S$ | Hyper-posterior distribution over the set of views $\mathcal{V}$ after observing the learning sample $S$ |
| $G_\rho^{\mathrm{MV}}$ | Multiview gibbs Classifier |
| $B_\rho^{\mathrm{MV}}$ | Multiview majority vote Classifier |
| $\mathbf{M}_v$ | $m \times n_v$ matrix such that $(\mathbf{M}_v)_{ij} = y_i h_v^j(x_i^v)$ |

List of Notations for multiview learning

# Chapter 1

# Introduction

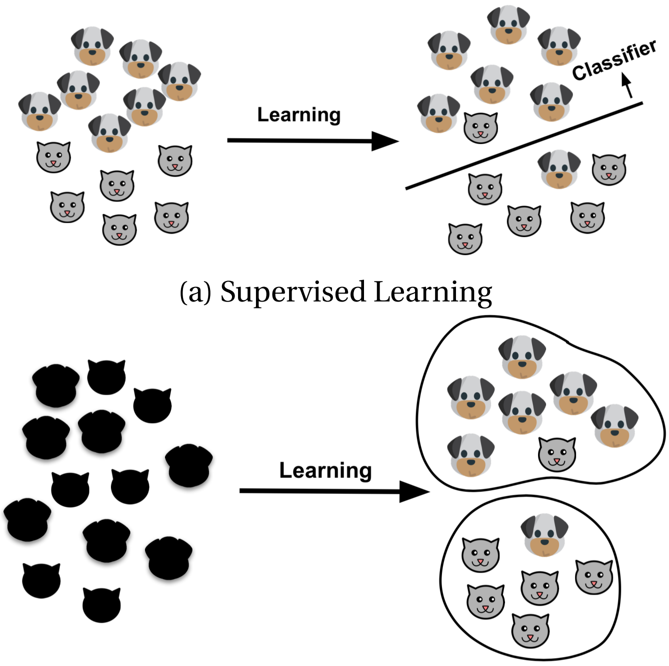
Machine learning (ML) is a subfield of Artificial Intelligence that focuses on the study of algorithms that “learn” from input data (past experiences) to make decisions (knowledge or expertise) on new data. Typically, the input to a learning algorithm is training data (corresponds to experience) and the output is a model (corresponds to expertise) that make decisions on the new data. In the past few decades, machine learning algorithms have been applied to many real-word applications for example spam filtering, credit card fraud detection, digit recognition, medical diagnostics, recommendation systems, search engines, etc. Machine learning algorithms can be broadly divided into two categories i.e. supervised [59, 87] and unsupervised learning [37][1].

- **Supervised Learning**: The learning algorithm is provided with input examples and their desired outputs (called as labels). The objective is to learn a classification function (referred as classifier) which classifies new examples into different labels (as illustrated in Figure 1.1(a)).

- **Unsupervised Learning**: The learning algorithm is provided with input examples but without any labels. The objective is to cluster the data into different categories or classes based on similarities between the data points (as illustrated in Figure 1.1 (b)).

In this thesis, we focus on the problem of supervised learning where we learn a classifier using the training data drawn from an unknown distribution $\mathcal{D}$ which performs well on

[1] There exists other learning paradigms, such as semi-supervised learning [16], transfer learning [64], reinforcement learning [82], etc.



(a) Supervised Learning

(b) Unsupervised Learning

Figure 1.1: An example of (a) Supervised Learning (classification of cats and dogs) and (b) Unsupervised Learning (clustering of cats and dogs)

new unseen data drawn from the same distribution. In other words, we want the learned classifier to generalize well from the training data to any data drawn from the same distribution. Therefore, we need to study the generalization guarantees in form of generalization bounds [86, 88] for learning algorithms. Probabilistic generalization bounds with a high probability on learning sample of size $m$ drawn from distribution $\mathcal{D}$, provides an estimation of true risk (or generalization error) of a classifier in terms of empirical error on training data, complexity of classifier and size of learning sample. In contrast, non-probabilistic generalization bounds are the expectation bounds on all possible learning samples of size $m$ drawn from distribution $\mathcal{D}$. In our work, we derive non-probabilistic generalization bounds for the PAC-Bayesian theory [35, 57] which provide theoretical guarantees for models that take the form of majority vote over set of classifiers. Assuming a priori weights (or distribution) over set of classifiers, PAC-Bayesian theory after seeing learning sample, aims at finding a posterior distribution over these set of classifiers leading to well performing weighted majority vote. Therefore, it is interesting to derive the non-probabilistic generalization bounds from PAC-Bayesian standpoint where the posterior distribution is data dependent.

In many real-life applications, we have data collected from different information sources having heterogeneous properties, so it is important to consider these multiple representations or views of the data. This issue is referred as multiview learning. It has been applied to many real applications for example in multilingual regions of the world, including many regions of Europe or in Canada, documents are available in more than one language [2]. As another example, in multimedia content understanding, multimedia segments can be described by their audio and video signals [4]. Therefore, multiview learning has become a promising topic with wide applicability.

In literature, there exists different ways to tackle multiview learning, spurred by seminal work of Blum and Mitchell [9] on co-training. To adapt multiview learning to single view setting, traditional machine learning algorithms such as support vector machines concatenate all the views of the data (also referred as early fusion [79]). However, this method does not take into account the view-specific properties of the mutliview data therefore tends to overfit in the case when we have small number of training examples [91]. Another approach is based on late fusion [6] where we combine classifiers learned on each view (view-specific classifiers) in order to exploit different representations for improving the performance of the final learned model [79]. In such situation, it is important to consider the relationships between the multiple views appropriately or in other words consensus or diversity between the views [2, 41, 50, 60]. In this thesis, we focus on supervised multiview learning, where we see multiview learning as combination of different view-specific classifiers or views. Therefore, according to our point of view, it is interesting to tackle multiview learning issue through PAC-Bayesian framework. It is an interesting theoretical tool to understand this setting as it allows to directly capture the trade-off between accuracy and diversity between voters [35, 60]. In consequence, we have extended the single-view PAC-Bayesian analysis to multiview learning. Moreover, compared to the PAC-Bayesian work of Sun et al. [81], we are interested here to the more general and natural case of multiview learning with more than two views. From practical point of view, we designed three multiview learning algorithms exploiting late fusion and boosting paradigms.

**Context of this work** This thesis was carried out in machine learning teams from two establishments: the Data Intelligence group of Laboratoire Hubert Curien UMR CNRS 5516, part of University of Saint-Étienne and University of Lyon, and the Data Analysis, Modeling and Machine Learning (AMA) group of Laboratoire Informatique de Grenoble, part of Grenoble Alps University. This project is partially funded by the “Région Rhône-Alpes” and ANR project LIVES (Learning with Interacting ViEws).

**Organization of the thesis** This dissertation is organized as follows. Part I reviews the background work relevant to this thesis:

- Chapter 2 introduces the notions related to statistical learning theory that are necessary for the rest of this document. We begin with two main principles of risk minimization for learning theory: *i)* empirical risk minimization (ERM) and *ii)* structural risk minimization (SRM), followed by examples of some classic supervised machine learning algorithms.
- Chapter 3 is dedicated to multiview learning in general. We present basic concepts and background for multiview learning. We introduce two fundamental principles of multiview learning: *i)* consensus and *ii)* diversity. For each principle, we present some of multiview learning algorithms.

In Part II, we present the contributions of our work:

- Chapter 4 presents a detailed overview of the PAC-Bayesian theory for single view learning. In this chapter, we derive the non-probabilistic generalization bounds expressed as expected risk bounds for the PAC-Bayesian theory.
- Chapter 5 presents the PAC-Bayesian analysis for multiview learning with more than two views. We considered a hierarchy of distributions, i.e. weights, over the views and the view-specific classifiers: *i)* for each view a posterior and prior distributions over the view-specific classifiers, and *ii)* a hyper-posterior and hyper-prior distribution over the set of views. Based on this setting, we derive PAC-Bayesian generalization bounds (both probabilistic and expected risk bounds) for multiview learning with more than two views. Moreover, we derive the generalization bound for the multiview $\mathcal{C}$-Bound which we use to derive boosting based algorithm `PB-MVBoost`.
- Chapter 6 presents two multiview learning algorithms based on boosting and late fusion approaches. First algorithm is a boosting-based learning algorithm, called as `PB-MVBoost`. It iteratively learns the weights over the view-specific classifiers and the weights over the views by optimizing the multiview $\mathcal{C}$-Bound which controls a trade-off between the accuracy and the diversity between the views. Second algorithm is a two-step learning algorithm $\texttt{Fusion}_{\texttt{Cq}}^{\texttt{all}}$ which combines the predictions of view-specific classifiers using a PAC-Bayesian algorithm `CqBoost` [73]. In order, to see the potential of proposed algorithm,

- Chapter 7 shows that the minimization of the classification error of multiview weighted majority vote is equivalent to the minimization of Bregman divergences. This allows us to derive a parallel-update multiview learning algorithm $\texttt{M}\omega\texttt{MvC}^2$. We experimently study our algorithm on three publicly available datasets.

Finally, in Chapter 8 we conclude our work and discuss possible directions for future work.

# Part I

# Background

# Chapter 2

# Single View Supervised Learning

In machine learning, the goal of supervised learning is to generally learn a classification function or a classifier using a set of labeled examples in order to make predictions on new unseen data. In this chapter, we introduce formally the supervised learning setting (for single view) and describe the main ideas of statistical learning theory, with a focus on binary classification. We present two main principles of risk minimization for learning theory: *i)* empirical risk minimization (ERM) and *ii)* structural risk minimization (SRM). Lastly, we recall some of supervised machine learning algorithms.

## 2.1 Introduction

The objective of supervised learning [59, 87] is to learn a classification function or a model using a set of labeled examples in order to make predictions on new unlabeled data [1]. A learning algorithm is provided with a training sample of $m$ examples denoted by $S = \{(x_i, y_i)\}_{i=1}^{m}$, that is assumed to be independently and identically distributed (i.i.d.) according to a unknown joint distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} \in \mathbb{R}^d$ is a $d-$dimensional input space and $\mathcal{Y}$ is the output space. The notation $(\mathcal{D})^m$ stands for the distribution of such a $m$-sample, and $\mathcal{D}_{\mathcal{X}}$ for the marginal distribution on $\mathcal{X}$. In this thesis, we consider binary classification tasks where $\mathcal{Y} = \{-1, +1\}$.

[1] Note that there exists other learning paradigms, such as unsupervised learning [37], semi-supervised learning [16], transfer learning [64], reinforcement learning [82], etc.

We consider a hypothesis space $\mathcal{H}$ consisting of a set of classifiers such that $\forall h \in \mathcal{H}, h : \mathcal{X} \to \mathcal{Y}$. In supervised learning, our objective is to learn a function (or in other words a classifier) $h_S : \mathcal{X} \to \mathcal{Y}$ belonging to the hypothesis space $\mathcal{H}$ using a training sample $S$, such that $h_S$ "best" predicts the label $y$ from $x$ for any input example $(x, y)$ drawn from the unknown distribution $\mathcal{D}$. In order to learn $h_S$, we need a criterion to evaluate the quality of any hypothesis $h \in \mathcal{H}$. Therefore, we define the notion of true risk:

**Definition 2.1.** (True Risk). The true risk is the expectation of the classification errors of a classifier $h \in \mathcal{H}$ over the data distribution $\mathcal{D}$:

$$R_{\mathcal{D}}(h) = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \mathbb{1}_{[h(x)\neq y]}, \tag{2.1}$$

where $\mathbb{1}_{[p]} = 1$ if predicate $p$ is true and 0 otherwise.

The goal of supervised learning is to find a classifier which has the smallest true risk. However, we can not compute the true risk of a classifier $h \in \mathcal{H}$ as the distribution $\mathcal{D}$ over data is unknown. Therefore, we rely on its empirical counterpart, i.e., we compute the error of the classifier on the training sample $S$. This is referred as the empirical risk.

**Definition 2.2.** (Empirical Risk). For a given training sample $S = \{(x_i, y_i)\}_{i=1}^m$ consisting of $m$ examples drawn from the unknown data distribution $\mathcal{D}$, we define the empirical risk of a classifier $h \in \mathcal{H}$ as:

$$R_S(h) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}_{[h(x_i)\neq y_i]}. \tag{2.2}$$

## 2.2 Strategies for Minimization of Risk

In order to solve the binary classification task, one natural solution is to pick a classifier $h \in \mathcal{H}$ which minimizes the empirical risk over the learning sample $S$. However, in real world scenarios, we have a limited number of training examples and we can always find a complex hypothesis which perfectly fits the training samples, i.e., $R_S(h) = 0$. It can happen that the learned hypothesis $h$ commits a lot of errors on new unseen data drawn from the distribution $\mathcal{D}$. This problem of having the empirical risk tending to zero and large deviation between the true risk and the empirical risk is called *overfitting*. Therefore, while finding a good hypothesis from hypothesis space we need to control the trade-off between the minimization of empirical risk and complexity of hypothesis space. This trade-off is called bias-variance trade-off. The solution to avoid overfitting is to restrict the hypothesis space to simple ones. In this section, we present two principles of risk minimization for learning theory: *i)* empirical risk minimization (ERM) and *ii)* structural risk minimization (SRM) [85]. We present these principles in the next sections.

### 2.2.1 Empirical Risk Minimization (ERM)

The idea behind the ERM principle is to restrict the hypothesis space $\mathcal{H}$ (consisting of simple classifiers) and then pick the classifier $h_S^* \in \mathcal{H}$ which has the smallest empirical error [87], such that

$$h_S^* = \operatorname*{argmin}_{h \in \mathcal{H}} R_S(h). \tag{2.3}$$

The fundamental question with ERM principle is: *does the minimization of the empirical risk leads to a good solution in terms of the true risk?* The answer to this question lies in statistical notion called consistency. According to this concept, we need to pick the hypothesis $h \in \mathcal{H}$ which has low deviation between the true risk and the empirical risk when the size of training examples tends to infinity. Vapnik [85] proved that ERM principle is consistent if and only if:

$$\forall \epsilon > 0, \lim_{m \to \infty} \Pr_{S \sim (\mathcal{D})^m} \left[ \sup_{h \in \mathcal{H}} \left[ R_{\mathcal{D}}(h) - R_S(h) \right] \leq \epsilon \right] = 0, \tag{2.4}$$

The direct implication of the above result is the generalization bound which is a tool to evaluate the deviation between the true risk and the empirical risk for all hypothesis $h \in \mathcal{H}$ learned on a learning sample $S$. These bounds are referred as PAC (Probably Approximately Correct) bounds [83]. The general form of PAC bounds is given as:

$$\forall h \in \mathcal{H}, \Pr_{S \sim (\mathcal{D})^m} \left[ \left| R_{\mathcal{D}}(h) - R_S(h) \right| \leq \epsilon \right] \geq 1 - \delta, \tag{2.5}$$

where $\epsilon \geq 0$ and $\delta \in (0,1]$. It means that with a high probability on the random choice of the learning sample the deviation between the true risk $R_{\mathcal{D}}(h)$ and the empirical risk $R_S(h)$, for a given $h \in \mathcal{H}$, is less than certain value $\epsilon$ (that we want as small as possible).

In this section, we present two generalization bounds that hold for any hypothesis $h \in \mathcal{H}$ and takes the form given by Equation (2.5). With a high probability on the random choice of the learning sample, they bound the difference between the true risk of a classifier and its empirical risk in terms of number of examples $m$ and the complexity of hypothesis space $\mathcal{H}$. Note that, these bounds holds for any classifier $h \in \mathcal{H}$.

#### 2.2.1.1 Bound Based on the VC-dimension

One of the possible way to measure the complexity of the hypothesis space $\mathcal{H}$ is the VC-dimension, proposed by Vapnik and Chervonenkis [88]. We consider $m$ data points in learning sample $S$. These $m$ points can be labeled in $2^m$ ways for binary classification task. Therefore, we can define $2^m$ different classification problems with $m$ data points. If for any of these classification problems, we can find $h \in \mathcal{H}$ that separates the two classes with a zero empirical error, then we say $\mathcal{H}$ shatters $m$ points. The maximum number of data points that can be shattered by any hypothesis space $\mathcal{H}$ is the VC-dimension of $\mathcal{H}$. For example, the VC-dimension for linear classifiers is $d+1$, where $d$ is the dimension of the hypothesis space $\mathcal{H}$. It helps us to measure the complexity of the hypothesis space or in other words the learning capacity of the hypothesis space. Using this notion of calculating the learning capacity of a hypothesis space, *Vapnik & Chervonenkis* [88] derived following generalization bound:

**Theorem 2.1.** *(Generalization bound based on the VC-dimension). Let $\mathcal{D}$ be an unknown distribution on $\mathcal{X} \times \mathcal{Y}$, let $\mathcal{H}$ be a continuous hypothesis space with VC-dimension $VC(\mathcal{H})$. For any $h \in \mathcal{H}$, with probability of at least $1-\delta$ on the random choice of the learning sample $S \sim (\mathcal{D})^m$, we have:*

$$R_{\mathcal{D}}(h) \leq R_S(h) + \sqrt{\frac{VC(\mathcal{H})\big(\ln \frac{2m}{VC(\mathcal{H})} + 1\big) + \ln(4/\delta)}{m}}.$$

This bound suggests that with a high probability on the random choice of the learning sample, the empirical risk of a classifier tends to its true risk if we have a large number of training examples and a hypothesis space $\mathcal{H}$ with a low VC-dimension. In practice, for some cases computation of the VC-dimension is not feasible and there are cases for which the VC-dimension equals to infinity. For example the VC-dimension of the K-nearest neighbour classifier is infinite for $K=1$. Moreover, the VC-dimension focuses on the worst labeling of examples for the hypothesis space $\mathcal{H}$. In the next section, we present the generalization bound based on the Rademacher complexity which is calculated on average on all possible labels instead of the worst labeling scenario and that are data dependent.

#### 2.2.1.2 Bound Based on the Rademacher Complexity

Generalization bounds based on the Rademacher complexity [49] measures how well any hypothesis $h \in \mathcal{H}$ correlates with random noise variables $\sigma_i$ instead of true labels $y_i$. These random variables $\sigma_i$ are called as Rademacher random variables $\sigma_i$ and are defined by

$$\sigma_i = \begin{cases} +1 & \text{with prob. } 1/2, \\ -1 & \text{with prob. } 1/2. \end{cases}$$

We compute the *empirical Rademacher complexity* of hypothesis space $\mathcal{H}$ with respect to a learning sample $S$ by

$$\mathfrak{R}_S(\mathcal{H}) = \mathop{\mathbb{E}}_{\sigma}\left[ \sup_{h\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^{m} \sigma_i h(x_i) \right]. \tag{2.6}$$

The above expression measures the correlation of $\mathcal{H}$ with random noise over the learning sample $S$ and has the advantage of measuring the complexity of a hypothesis space that is data dependent. However, we are interested in measuring the correlation of $\mathcal{H}$ with respect to data distribution $\mathcal{D}_{\mathcal{X}}$ over $\mathcal{X}$. Therefore, we compute the expectation of $R_S(\mathcal{H})$ over all learning samples of size $m$ drawn i.i.d. from distribution $\mathcal{D}_{\mathcal{X}}$:

$$\mathfrak{R}_{\mathcal{D}}(\mathcal{H}) = \mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\left[\mathfrak{R}_S(\mathcal{H})\right].$$

This is the *Rademacher complexity* of a given hypothesis space $\mathcal{H}$. Following theorem presents the classical generalization bound based on Rademacher complexity [6, 49]:

**Theorem 2.2.** *(Generalization bound based on the Rademacher Complexity). Let $\mathcal{D}$ be an unknown distribution on $\mathcal{X}\times\mathcal{Y}$, let $\mathcal{H}$ be a continuous hypothesis space, for any $h\in\mathcal{H}$, with probability of at least $1-\delta$ on the random choice of the learning sample $S\sim(\mathcal{D})^m$, we have:*

$$R_{\mathcal{D}}(h) \le R_S(h) + \mathfrak{R}_{\mathcal{D}}(\mathcal{H}) + \sqrt{\frac{\log\frac{1}{\delta}}{2m}},$$
$$\textit{and } R_{\mathcal{D}}(h) \le R_S(h) + \mathfrak{R}_S(\mathcal{H}) + 3\sqrt{\frac{\log\frac{2}{\delta}}{2m}}.$$

The second bound in the above theorem depends on the *empirical Rademacher complexity* which can be easily calculated from the learning sample $S$. These bounds suggests that with a high probability on the random choice of the learning sample, the empirical risk of a classifier tends to its true risk if we have a large number of training examples and hypothesis space $\mathcal{H}$ of low Rademacher complexity.

### 2.2.2 Structural Risk Minimization (SRM)

Since restricting the hypothesis space $\mathcal{H}$ requires prior knowledge about the learning task and the data. One solution is to consider an infinite sequence of hypothesis classes $\mathcal{H}_1,\mathcal{H}_2,\ldots$ with increasing complexities such that $\forall i\in\{1,2,\ldots\}, \mathcal{H}_i\subset\mathcal{H}_{i+1}$. For each hypothesis space $\mathcal{H}_i$, the learning algorithm selects a hypothesis that minimizes the empirical risk. Finally, from these (sub-)optimal hypotheses, one picks the hypothesis $h_S^*$ which has smallest empirical risk.

$$h_S^* = \operatorname*{argmin}_{h \in \mathcal{H}_i, i \in \{1,2,\dots\}} \left\{ R_S(h) + pen(\mathcal{H}_i) \right\}, \tag{2.7}$$

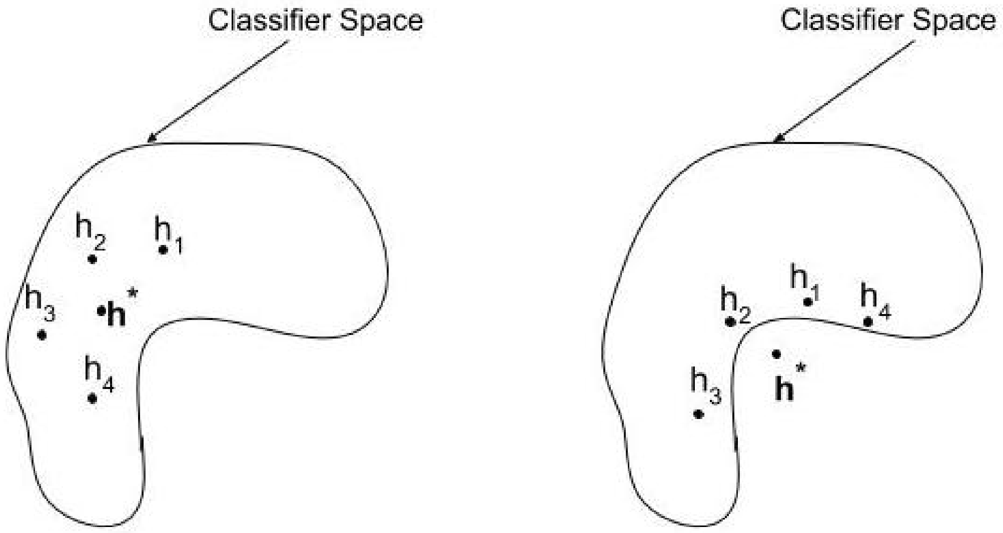
where $pen(\mathcal{H}_i)$ is the penalty of the hypothesis space $\mathcal{H}_i$ depending upon its complexity.

In the next section, we present classifier combination approaches where instead of picking a $h \in \mathcal{H}$, we construct a weighted majority vote over all the classifiers from $\mathcal{H}$. Classifier combination approaches has shown to perform well in practice as it helps to reduce both bias and variance.

## 2.3 Classifier Combination Approaches

Ensemble methods or classifier combination approaches [26, 50, 71] aims at combining the outputs of individual classifiers (weighted or unweighted combination) from the hypothesis space $\mathcal{H}$. In practice, it has been shown that the final learned combination performs better than individual classifiers [26, 50]. A necessary condition for better performance of ensemble methods is that individual classifiers should be weak and diverse [43]. A classifier is weak when its error rate is better than random guessing on any new example $x$ drawn from the unknown distribution $\mathcal{D}$ and two classifiers are diverse if they make errors on different examples. If these conditions are satisfied then the reason for better performance of ensemble methods is that they try to decrease both variance and bias. Variance is the amount by which the prediction, over one training sample, differs from the expected value over all the training sample. Bias is the amount by which the expected classifier prediction differs from the true prediction of the training sample.

**Reducing Variance.** We can learn different classifiers from a classifier space $\mathcal{H}$ having the same empirical risk on the training data. However, the learned classifiers can have different generalization guarantees. Therefore, instead of picking a single classifier we can combine the classifiers. The combination may not perform better than a single classifier but it will eliminate the risk of picking a bad single classifier. This scenario is illustrated by Figure 2.1 (a). In the figure, $\{h_1, \dots, h_4\}$ are single classifiers which has similar empirical risk but different generalization guarantees and $h^*$ is an optimal classifier in the terms of the true risk. By combining the four accurate classifiers, we can find a good approximation to $h^*$.



(a) Reducing Variance (b) Reducing Bias

Figure 2.1: Reasons for better performance of ensemble methods than a single classifier. By combining the accurate classifiers $\{h_1,\ldots,h_4\}$, we can find a good approximation to the optimial classifier $h^*$.

**Reducing Bias:** There is a possibility that the considered classifier space does not contain the optimal classifier $h^*$. Then the classifier combination may help us to find a good approximation to the optimal classifier. Figure 2.1 (b) shows the case when the optimal classifier is outside the considered classifier space.

### 2.3.1 Notations and Setting

In order to learn a weighted combination over all the classifiers in $\mathcal{H}$, we define the majority vote classifier as follows:

$$B_Q(x) = \operatorname{sign}\left[\mathop{\mathbb{E}}_{h\sim Q} h(x)\right], \tag{2.8}$$

where $Q$ is a distribution over $h \in \mathcal{H}$. The learner objective is to find the posterior distribution $Q$ that leads to well-performing majority vote $B_Q$. We define the true risk and the empirical risk of the majority vote as follows:

**Definition 2.3.** (True Risk of the majority vote). The true risk of the weighted majority vote classifier over the data distribution $\mathcal{D}$:

$$R_{\mathcal{D}}(B_Q) = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \mathbb{1}_{[B_Q(x)\neq y]}. \tag{2.9}$$

**Definition 2.4.** (Empirical Risk of the majority vote). For a given training sample $S = \{(x_i, y_i)\}_{i=1}^m$ consisting of $m$ examples drawn from the unknown data distribution $\mathcal{D}$, we define the empirical risk of the majority vote classifier as:

$$R_S(B_Q) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}_{[B_Q(x_i) \neq y_i]}. \tag{2.10}$$

In the next section, we present the classical PAC-Bayesian generalization bound which upper bounds the deviation between the true risk and the empirical risk of the majority vote classifier in terms of Kullback-Leibler divergence between the prior and the posterior distributions over the set of classifiers and the number of traning examples $m$.

### 2.3.2 The PAC-Bayesian Generalization Bound

The PAC-Bayesian theory, introduced by McAllester [57], is a tool to derive theoretical guarantees for models that take the form of a majority vote over the hypothesis space $\mathcal{H}$ (defined as in Equation (2.8)). The PAC-Bayesian theory assumes a prior distribution $P$ over the set of classifiers from hypothesis space $\mathcal{H}$, aims at learning – from the learning sample $S$ – a posterior distribution $Q$ that leads to a well-performing weighted majority vote $B_Q$ i.e with a low true risk. The following theorem is the PAC-Bayesian generalization bound proposed by *McAllester* [58]:

**Theorem 2.3.** *(The PAC-Bayesian Theorem [58]) For any distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, for any hypothesis space $\mathcal{H}$, for any prior distribution $P$ over $\mathcal{H}$, for any $\delta \in (0,1]$, with a probability of at least $1-\delta$ over the learning sample $S \sim (\mathcal{D})^m$, we have for all posterior distributions $Q$ over $\mathcal{H}$:*

$$R_{\mathcal{D}}(B_Q) \leq 2 \cdot \mathop{\mathbb{E}}_{h \sim Q} R_{\mathcal{D}}(h) \leq 2 \cdot \left[ \mathop{\mathbb{E}}_{h \sim Q} R_S(h) + \sqrt{\frac{KL(Q||P) + \ln \frac{2\sqrt{m}}{\delta}}{2m}} \right],$$

*where $KL(Q||P) = \mathop{\mathbb{E}}_{h \sim Q} \ln \frac{Q(h)}{P(h)}$ is the Kullback-Leibler divergence between the learned posterior distribution $Q$ and $P$.*

From the above theorem, the true risk of the majority vote can be seen as trade-off between expectation of individual classifiers risk, KL divergence term that captures the deviation between prior $P$ and posterior $Q$ distributions over the set of classifiers from hypothesis space $\mathcal{H}$ and the number of training examples $m$. In Chapter 4, we present the general PAC-Bayesian theory in more details. Since by definition the learned posterior distribution $Q$

is data dependent, we propose in Chapter 4 a new formulation of the PAC-Bayesian theorem that aims at bounding the expectation over all the posterior distributions we can learn for a given algorithm (and a learning sample size). Our new PAC-Bayesian theorem is then not anymore expressed as a probabilistic bound over a random choice of learning sample $S$, but as an expectation risk bound, bringing another point of view on the PAC-Bayesian analysis. Note that all the generalization bounds we have presented so far are probabilistic bounds.

## 2.4 Some Supervised Learning Algorithms

In this section, we present three algorithms for the supervised machine learning.

### 2.4.1 Support Vector Machines

Support Vector Machines [10, 20] is one of the most commonly used supervised learning algorithm for binary classification tasks. Support vector machines (SVM) outputs a classifier which takes the form of an optimal hyperplane classifying a new given example into one of the label. For example, in a two dimensional space this hyperplane is a line dividing the plane in two parts corresponding to each label. Here, "optimal hyperplane" means the seperating hyperplane that maximizes the "margin" of the training sample. In this thesis, we use SVM models to derive a new algorithm for multiview learning (in Chapter 6).

In a $d$-dimensional input space $\mathcal{X} \in \mathbb{R}^d$, the equation of hyperplane is defined as follows:

$$\langle w, x \rangle + b = 0 \Longleftrightarrow \sum_{i=1}^{d} w_i x_i + b = 0,$$

where $\langle \cdot, \cdot \rangle$ is dot product between two vectors, $w \in \mathbb{R}^d$ is the *normal vector* for the hyperplane and $b \in \mathbb{R}$ is the *intercept* of the hyperplane. The hyperplane divides the plane in two classes $\mathcal{Y} = \{-1, +1\}$ and the classifier for the hyperplane is defined as follows:

$$\begin{aligned} h(x) &= \operatorname{sign}\left[\langle w, x \rangle + b\right] \\ &= \begin{cases} +1 & \text{if } \langle w, x \rangle + b > 0, \\ -1 & \text{if } \langle w, x \rangle + b < 0. \end{cases} \end{aligned}$$

It is clear that there exist an infinite number of hyperplanes (with different $w$ and $b$) that separate two classes. However, SVM chooses the hyperplane which has the maximum

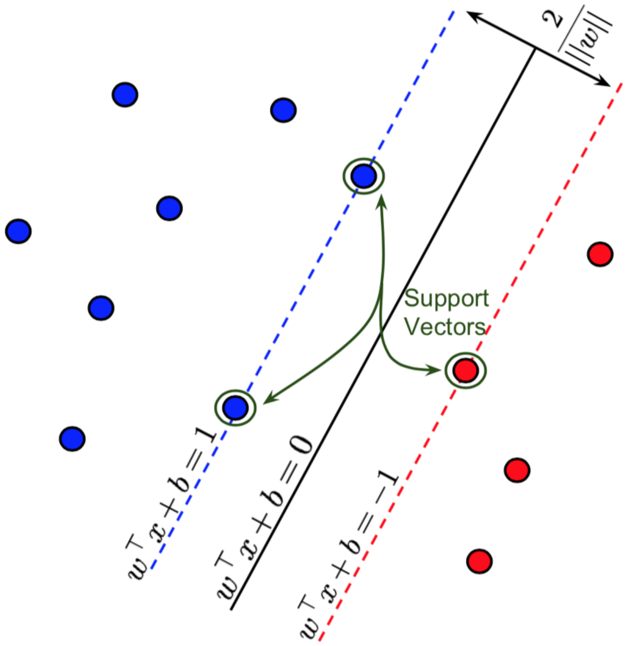


Figure 2.2: Maximum-margin hyperplane for a binary SVM. Support vectors are marked with a green outline.

distance from the nearest training examples from the hyperplane. Since the number of training examples is finite $\exists \epsilon \geq 0$ such that:

$$\begin{aligned} h(x) &= \text{sign}\left[\langle w, x\rangle + b\right] \\ &= \begin{cases} +1 & \text{if } \langle w, x\rangle + b \geq \epsilon, \\ -1 & \text{if } \langle w, x\rangle + b \leq -\epsilon. \end{cases} \end{aligned}$$

As we can scale $w$ and $b$, we can rewrite above equation as follows:

$$\begin{aligned} h(x) &= \text{sign}\left[\langle w, x\rangle + b\right] \\ &= \begin{cases} +1 & \text{if } \langle w, x\rangle + b \geq 1, \\ -1 & \text{if } \langle w, x\rangle + b \leq -1. \end{cases} \end{aligned}$$

From this equation, we can easily deduce that there is no training example between two parallel hyperplanes $\langle w, x\rangle + b = 1$ and $\langle w, x\rangle + b = -1$. Therefore, the distance between two hyperplanes is $\frac{2}{||w||}$ which is referred as margin of the separating hyperplane. For a given learning sample $S = \{(x_i, y_i)\}_{i=1}^m$, support vector machines chooses the hyperplane which maximizes the margin by solving the optimization problem:

$$\begin{aligned} &\min_{w} \quad \frac{1}{2} w^T w, \\ &\text{s.t.} \quad y_i\big(\langle w, x\rangle + b\big) \geq 1, \ \ i = 1, \ldots, m. \end{aligned} \tag{2.11}$$

SVM uses the Lagrange duality techniques [11] to transform the above optimization problem into a dual optimization problem:

$$\begin{aligned} &\max_{\mu} \quad \sum_{i=1}^{m} \mu_i - \frac{1}{2} \sum_{i,j=1}^{m} \mu_i \mu_j y_i y_j x_i^T x_j, \\ &s.t. \quad \mu_i \geq 0, \ \sum_{i=1}^{m} y_i \mu_i = 0, \ \ i = 1, \ldots, m, \end{aligned} \tag{2.12}$$

where $\mu_i$ are the Lagrange multipliers corresponding to each example in the training sample $S$. Training examples which have non-zero Lagrange multipliers are called as support vectors and they lie on two parallel hyperplanes $\langle w, x\rangle + b = 1$ and $\langle w, x\rangle + b = -1$ (see Figure 2.2 for illustration).

We have assumed that the training data is linearly separable. This is a strong assumption for many real world scenarios. One solution to this issue is to transform the original input space $\mathcal{X} \in \mathbb{R}^d$ into some higher dimensional space $H$ (a Hilbert Space) and learn a linear classifier in $H$.

Assume we have a mapping function $\phi$ from the original space $\mathbb{R}^d$ to a high dimensional space $H$, then we can rewrite the dual optimization problem (Equation (2.12)) as:

$$\begin{aligned} &\max_{\mu} \quad \sum_{i=1}^{m} \mu_i - \frac{1}{2} \sum_{i,j=1}^{m} \mu_i \mu_j y_i y_j \langle \phi(x_i), \phi(x_j)\rangle, \\ &s.t. \quad \mu_i \geq 0, \ \sum_{i=1}^{m} y_i \mu_i = 0, \ \ i = 1, \ldots, m, \end{aligned}$$

It is trivial to note that the knowledge of $\langle \phi(x_i), \phi(x_j)\rangle$ is sufficient to solve the above optimization problem. Therefore, we denote $\langle \phi(x_i), \phi(x_j)\rangle$ by a kernel function $K(x_i, x_j) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ and by computing $K(x_i, x_j)$ directly, we can avoid the explicit mapping of our data from $\mathbb{R}^d$ to $H$. This is called as kernel trick. Finally the classifier becomes:

$$h(x) = \text{sign}\left[\sum_{i=1}^{m} \mu_i K(x, x_i) + b\right]. \tag{2.13}$$

Note that, $K$ is a kernel function if it is a symmetric positive semi-definite (PSD), i.e.,

$$\sum_{i=1}^{m}\sum_{j=1}^{m} c_i c_j K(x_i, x_j) \geq 0,$$

for all finite sequences of $x_1, \ldots, x_m \in \mathcal{X}$ and $c_1, \ldots, c_m \in \mathbb{R}$. The kernel function which satisfies above property are referred as Mercer Kernel. In the literature, there exists different types of Mercer kernels, some of them are:

- **Linear Kernel**:

  $$\forall (x_i, x_j) \in \mathcal{X}^2, K(x_i, x_j) = x_i^T x_j + c,$$

  where constant $c$ is a hyperparameter.

- **Polynomial Kernel**:

  $$\forall (x_i, x_j) \in \mathcal{X}^2, K(x_i, x_j) = \Big(a x_i^T x_j + c\Big)^p,$$

  where slope $a$, constant $c$ and degree $p$ are hyperparameters.

- **Gaussian Kernel**:

  $$\forall (x_i, x_j) \in \mathcal{X}^2, K(x_i, x_j) = \exp\Big(-\frac{||x_i - x_j||^2}{2\sigma^2}\Big),$$

  where $\sigma$ (standard deviation which measures the amount of variation of a set of input examples) is a hyperparameter.

- **Radial Basis Function (RBF) kernel**:

  $$\forall (x_i, x_j) \in \mathcal{X}^2, K(x_i, x_j) = \exp\Big(-\gamma ||x_i - x_j||^2\Big),$$

  where $\gamma$ is a hyperparameter. Note that, this kernel can project the data in infinite dimension.

### 2.4.2 Adaboost

Adaboost [32] is a classifier combination approach based on boosting [75]. Typically, Adaboost repeatedly ($T$ times) learn a “weak”classifier using a learning algorithm with different probability distributions over the learning sample $S$. Finally, it combines all the weak classifiers in order to have one single strong classifier ($B_Q$) which performs better than the individual weak classifiers (see Algorithm 1). In this thesis, we use adaboost as one of our

**Algorithm 1** Adaboost

```
Input: Learning Sample $S = (x_i, y_i), \dots, (x_m, y_m)$, where $x_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$.
       Hypothesis space $\mathcal{H}$.
       Number of iterations $T$.
for $x_i \in S$ do
    $\mathcal{D}_1(x_i) \leftarrow \frac{1}{m}$
for $t = 1, \dots, T$ do
    Learn a weak classifier $h^{(t)}$ using distribution $\mathcal{D}_{(t)}$
    Compute error: $\epsilon^{(t)} \leftarrow \underset{(x_i,y_i)\sim\mathcal{D}_{(t)}}{\mathbb{E}} \left[\mathbb{1}_{[h^{(t)}(x_i)\neq y_i]}\right]$
    Compute classifier weight: $Q^{(t)} \leftarrow \frac{1}{2}\left[\ln\left(\frac{1-\epsilon^{(t)}}{\epsilon^{(t)}}\right)\right]$
    for $\mathbf{x}_i \in S$ do
        $\mathcal{D}_{(t+1)}(x_i) \leftarrow \frac{\mathcal{D}_{(t)}(x_i)\exp\left(-y_i Q^{(t)} h^{(t)}(x_i)\right)}{\sum_{j=1}^{m} \mathcal{D}_{(t)}(x_j)\exp\left(-y_j Q^{(t)} h^{(t)}(x_j)\right)}$
Return: $B_Q(x) = \left(\sum_{t=1}^{T} Q^{(t)} h^{(t)}(x)\right)$
```

baselines and we also exploit the boosting paradigm to derive new algorithms for multiview learning (in Chapters 6 and 7).

At each iteration $t$ of the algorithm, we select a weak classifier (Step 4) and compute the weight over the classifier (Step 6) as follows:

$$Q^{(t)} = \frac{1}{2}\left[\ln\left(\frac{1-\epsilon^{(t)}}{\epsilon^{(t)}}\right)\right],$$

where $\epsilon^{(t)} = \underset{(x_i,y_i)\sim\mathcal{D}_t}{\mathbb{E}} \left[\mathbb{1}_{[h^{(t)}(x_i)\neq y_i]}\right]$ is the classification error of the selected classifier. Intuitively, we give more weight to the classifiers (in the final combination) which have low classification error on the learning sample. Finally, we update the weight for any example $x_i$ (Step 8) as follows:

$$\mathcal{D}_{(t+1)}(x_i) \leftarrow \frac{\mathcal{D}_{(t)}(x_i)\exp\left(-y_i Q^{(t)} h^{(t)}(x_i)\right)}{\sum_{j=1}^{m} \mathcal{D}_{(t)}(x_j)\exp\left(-y_j Q^{(t)} h^{(t)}(x_j)\right)}.$$

Intuitively, we are increasing the weight of the examples which are misclassified by the current classifier. This is done to learn a classifier at the next iteration $t+1$ which focuses on these misclassified examples.

### 2.4.3 CqBoost

`CqBoost` [73] is a column generation ensemble learning algorithm based on the $\mathcal{C}$-Bound [51] (recalled in Theorem 2.4 below). In this thesis, we use this algorithm to derive a multiview learning algorithm in Chapter 6.

The $\mathcal{C}$-Bound gives a tight upper bound on the risk of the majority vote defined by Equation (2.9). It depends on the first and second statistical moments of the margin of the majority vote $B_Q$, defined as:

**Definition 2.5.** (Margin) Let $M_Q$ be a random variable that outputs the margin of the majority vote on the example $(x, y)$ drawn from distribution $\mathcal{D}$, given by

$$M_Q(x,y) = \mathop{\mathbb{E}}_{h\sim Q} y h(x),$$

The first and second statistical moments of the margin are respectively given by

$$\begin{aligned} \mu_1(M_Q^{\mathcal{D}}) &= \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} M_Q(x,y), \\ \mu_2(M_Q^{\mathcal{D}}) &= \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \big(M_Q(x,y)\big)^2. \end{aligned}$$

According to this definition, the risk of the weighted majority vote can be rewritten as:

$$R_{\mathcal{D}}(B_Q) = \mathop{\Pr}_{(x,y)\sim\mathcal{D}} \big(M_Q(x,y) \leq 0\big).$$

From this observation, one can derive the $\mathcal{C}$-Bound stated as follows:

**Theorem 2.4.** *(The $\mathcal{C}$-Bound [34, 72, 73]) For any distribution $Q$ over $\mathcal{H}$ and for any distribution $\mathcal{D}$, if $\mu_1\left(M_Q^{\mathcal{D}}\right) \geq 0$, we have:*

$$R_{\mathcal{D}}(B_Q) \leq 1 - \frac{\left(\mu_1\left(M_Q^{\mathcal{D}}\right)\right)^2}{\mu_2\left(M_Q^{\mathcal{D}}\right)}.$$

The minimization of the empirical counterpart of the $\mathcal{C}$-Bound is a natural way to learn the distribution $Q$ in order to lead a well performing majority vote $B_Q(x)$. CqBoost minimizes the second moment of margin by fixing its first moment $\mu_1(M_Q^S) = \mu$ (hyperparameter to tune).

Let $\mathbf{H}$ be the classification matrix of size $m \times n$ where $m$ is the number of examples and $n$ is the number of classifiers in $\mathcal{H}$. Each element $H_{ij} = h_j(x_i)$ contains the output of the classifier $h_j \in \mathcal{H}$ for any input example $x_i$. Let $\mathbf{y}$ be the vector of labels of the training examples and $\mathbf{q}$ be the vector of weights over set of classifiers. Now, we can rewrite the first and second

**Algorithm 2** CqBoost

```
Initialize: Let q be the vector of n zeros, α be the vector of m values with $\frac{1}{m}$ and let $\hat{\mathbf{H}}$ be a empty matrix.
loop
    Select the column $j$ violating the most constraint of dual problem (Equation 2.15).
    If $\sum_{i=1}^{m} \alpha_i y_i H_{ij} \leq \nu + \epsilon$:
        Break
    Add the $j$-th column of $\mathbf{H}$ to matrix $\hat{\mathbf{H}}$.
    Update $\mathbf{q}, \alpha$ and $\nu$ by solving the primal or dual optimization problem of Equations 2.14 or 2.15 using matrix $\hat{\mathbf{H}}$.
Return: q
```

moments of margin as follows:

$$\begin{aligned}\mu_1(M_Q^{\mathcal{D}}) &= \frac{1}{m}\sum_{i=1}^{m} M_Q(x_i, y_k)\\ &= \frac{1}{m}\mathbf{y}^\top \mathbf{H}\mathbf{q},\\ \text{and } \mu_2(M_Q^{\mathcal{D}}) &= \frac{1}{m}\sum_{i=1}^{m} \big(M_Q(x_i, y_k)\big)^2\\ &= \frac{1}{m}\mathbf{q}^\top \mathbf{H}^\top \mathbf{H}\mathbf{q}.\end{aligned}$$

Finally, CqBoost solves the following constrained optimization problem:

$$\begin{aligned}&\underset{\mathbf{q},\gamma}{\text{argmin}} \quad \frac{1}{m}\gamma^\top\gamma,\\ &s.t. \quad \gamma = \text{diag}(\mathbf{y})\mathbf{H}\mathbf{q}, \ \ \frac{1}{m}\mathbf{1}^\top\gamma \leq \mu, \ \ \mathbf{q} \geq \mathbf{0}, \ \ \mathbf{1}^\top\mathbf{q} = 1,\end{aligned} \tag{2.14}$$

where **0** and **1** are the vector of zeros and ones of size $n$. CqBoost uses the Lagrange duality techniques [11] to transform the above optimization problem to a dual optimization problem:

$$\begin{aligned}&\underset{\alpha,\beta,\nu}{\text{argmin}} \quad \frac{m}{4}\alpha^\top\alpha + \frac{\beta}{2}\mathbf{1}^\top\alpha + \frac{\beta^2}{4} + \beta\mu + \nu,\\ &s.t. \quad \mathbf{H}^\top \text{diag}(\mathbf{y})\alpha \leq \nu\mathbf{1}, \ \ \beta \geq 0,\end{aligned} \tag{2.15}$$

where, $\alpha, \beta$ and $\nu$ are Lagrange multipliers. A column generation based algorithm CqBoost [73] (see Algorithm 2) is designed based on above optimization problem. At each iteration $t$, the algorithm selects a new column from matrix **H** which is added to the problem. Finally, it solves the primal or dual problem of original problem. It stops when no more column violates the dual constraint. Note that the $\mathcal{C}$-Bound has led to another algorithm MinCq [72].

## 2.5 Conclusion

In this chapter, we introduce the basic concepts of supervised machine learning for classification where we consider a labeled training sample. Here, our objective is to learn a classifier in order to make predictions on new data coming from same distribution than the one that have generated the learning sample. In addition, we introduce three supervised learning algorithms: SVM, Adaboost and Cqboost. In many real-life applications, we can have data produced by more than one source and are so-called as multiview data. In this thesis, we are particularly interested in deriving supervised learning algorithms for multiview learning when we have multiple representations of the input data. In the next chapter, we introduce the concepts and state-of-art methods for multiview learning.

CHAPTER 3

# MULTIVIEW LEARNING

In this chapter, we present some basic concepts and notions for multiview learning. We start by a brief introduction on multiview learning in Section 3.1. Then, we present the two fundamental principles of multiview learning: *i)* consensus (Section 3.3), and *ii)* diversity principles (Section 3.4). We discuss some of algorithms in details for both principles.

## 3.1 Introduction

With the tremendous generation of data, we have data collected from different information sources having heterogeneous properties, thus it is important to consider these representations or views of the data. This problem of machine learning is referred as multiview learning, spurred by the seminal work of Blum and Mitchell on co-training [9]. Multiview learning has many applications (see Figure 3.1), some of them are

- **Image Classification:** We can represent each image by different sets of features such as Histograms of Oriented Gradient (HOG) and Region-Of-Interest (ROI). These different features can be seen as different views of data [84].
- **Multilingual Document Classification:** We have documents written in different languages like French, German, English etc. We can see different languages as different views of data [2].
- **Webpage Classification:** We can represent each webpage with different descriptions for example textual content, images, inbound and outbound hyperlinks etc [9].

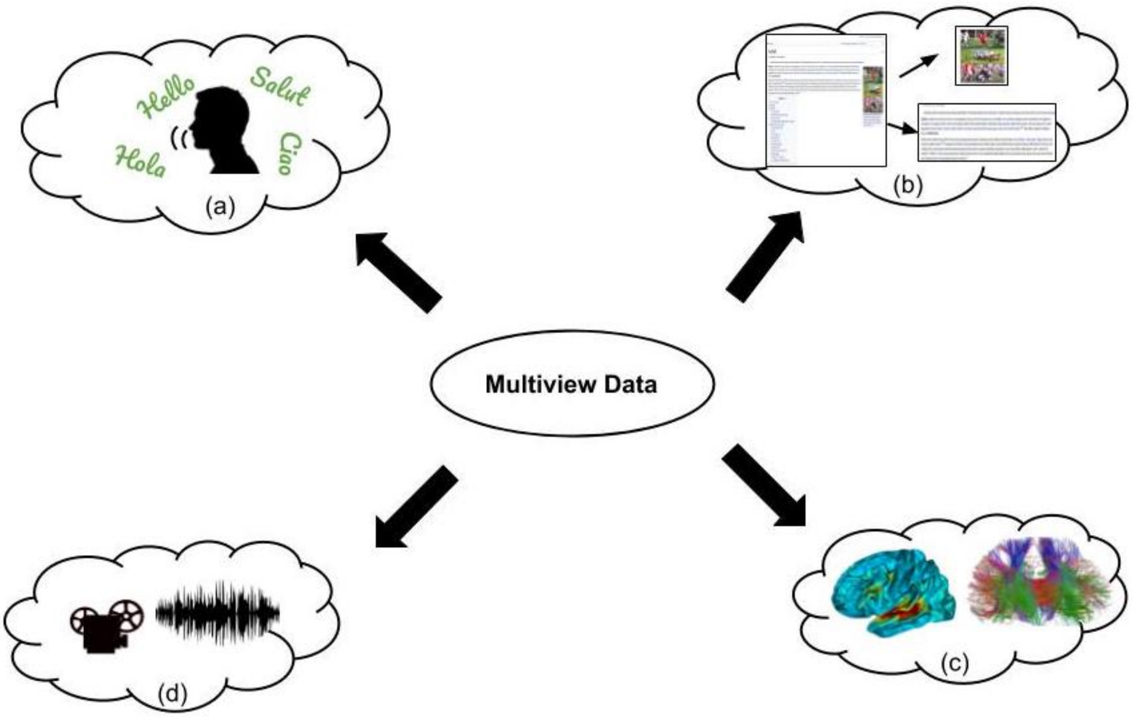


Figure 3.1: Examples of multiview learning data: (a) Multilingual document classification, (b) Webpage classification based on both textual and image data, (c) Medical Imaging where each brain image is represented by its MRI and t-fMRI images and (d) Multimedia data which is combination of both video and audio signals.

- **Multimedia Data Processing:** We can describe each multimedia segment by their audio and video signals [4].
- **Medical Imaging:** We can represent human brain with different set of features for example MRI, t-fMRI, fMRI, EEG etc [63].

One natural solution is to adapt multiview learning problem to monoview setting by concatenating all the views of the data and learn the final model using traditional machine learning algorithms such as support vector machines (see Section 2.4.1) This method is referred as early fusion [79]. However, early fusion based approaches do not take into account the view-specific properties of the multiview data, therefore they tend to overfit when we have a small number of training examples [91]. Another approach is to see multiview learning as a combination of different view-specific classifiers corresponding to each view. The goal is to learn a multiview model over the predictions of view-specific classifiers. This method of learning a multiview model in two stages is called as late fusion [79] (sometimes referred as stacking [89]), as illustrated in Figure 3.2. In contrast to monoview learning, multiview

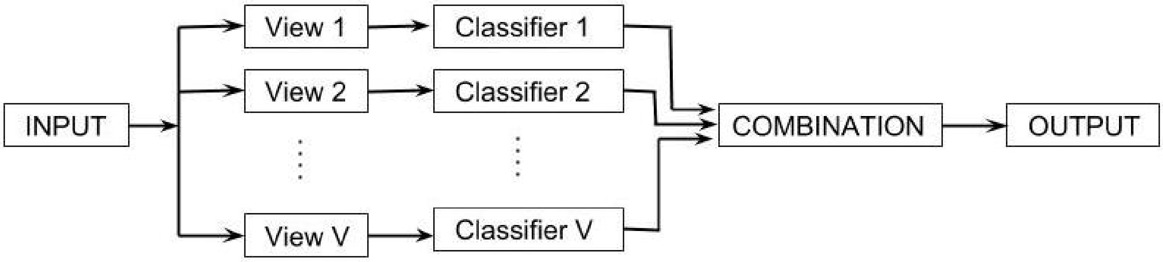


Figure 3.2: Classifier Combination approach to multiview learning

learning aims at exploiting different representations or views of the input data in order to improve the learning performance. Most of multiview learning algorithms exploits two fundamental principles which ensures their success: *i) consensus*, and *ii) diversity* principles. In the next section, we first present basic notations and setting for multiview learning. In sections 3.3 and 3.4, we present consensus and diversity principles in more details followed by conclusion in Section 3.5.

## 3.2 Notations and Setting

We consider multiview binary classification tasks where the examples are described with $V \geq 2$ different representation spaces, i.e., views; let $\mathcal{V}$ be the set of these $V$ views. Formally, we focus on tasks for which the input space is $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_V$, where $\forall v \in \mathcal{V},\ X_v \subseteq \mathbb{R}^{d_v}$ is a $d_v$-dimensional input space, and the binary output space is $\mathcal{Y} = \{-1, +1\}$. We assume that $\mathcal{D}$ is an unknown distribution over $\mathcal{X} \times \mathcal{Y}$. Each multiview example $\mathbf{x} = (x^1, x^2, \ldots, x^V) \in \mathcal{X}$ is given with its label $y \in \mathcal{Y}$, and is independently and identically drawn (*i.i.d.*) from $\mathcal{D}$. In the case of supervised learning, an algorithm is provided with a training sample $S$ of $m$ examples *i.i.d.* from $\mathcal{D}$: $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m} \sim (\mathcal{D})^m$, where $(\mathcal{D})^m$ stands for the distribution of a $m$-sample. Note that, in the case of semi-supervised learning one has access to an additional unlabeled training data $S_u = \{\mathbf{x}_j\}_{j=1}^{m_u} \sim (\mathcal{D}_{\mathcal{X}})^{m_u}$ along with labeled data $S_l = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m_l} \sim (\mathcal{D})^{m_l}$. Moreover, for each view, we consider a view-specific set $\mathcal{H}_v$ of classifiers $h_v : \mathcal{X}_v \rightarrow \mathcal{Y}$. The goal of multiview learning is to exploit multiple representations of the input data and improve the learning performance.

## 3.3 Consensus Principle

The consensus principle [22, 50] seeks to maximize the agreement between multiple representations of the data. Consider a two-view multiview data (i.e. $V = 2$) where each input example $\mathbf{x} = (x^1, x^2) \in \mathcal{X}$ with its label $y \in \mathcal{Y}$. The agreement between two classifiers on two views can be formulated as follows

$$\texttt{Agreement} = \Pr_{\mathbf{x} \sim \mathcal{D}_{\mathcal{X}}} \left[ h_1(x^1) = h_2(x^2) \right]. \tag{3.1}$$

In recent years, many multiview learning algorithms have been proposed based on the above consensus principle. In this section, we present some approaches which exploits this principle.

### 3.3.1 Canonical Correlation Analysis

Canonical Correlation Analysis (CCA) [44] based approaches maximize the inter-relationships between two (or more) sets of variables. CCA has been applied to multiview learning[1, 17, 25, 95] which aims at learning a latent subspace shared by multiple views by assuming that the input views are generated from this subspace. The dimensionality of this latent subspace is lower than the original views, so CCA is an effective way to eliminate the curse of dimensionality problem.

Consider a two-view multiview data (i.e. $V = 2$) where the input examples are drawn from $\mathcal{X}_1 \in \mathbb{R}^{d_1}$ and $\mathcal{X}_1 \in \mathbb{R}^{d_2}$ input spaces. Let $\mathbf{X}_1$ and $\mathbf{X}_2$ be $m \times d_1$ and $m \times d_2$ data matrices corresponding to each view $v_1$ and $v_2$ respectively. The goal of CCA is to find two projection vectors $w_1 \in \mathbb{R}^{d_1}$ and $w_2 \in \mathbb{R}^{d_2}$ such that the projected data on $w_1$ and $w_2$ have a maximum correlation, $\texttt{Corr}$ defined as following:

$$\begin{aligned} \texttt{Corr} &= \frac{\texttt{cov}\left(w_1^\top \mathbf{X}_1, w_2^\top \mathbf{X}_2\right)}{\sqrt{\texttt{var}\left(w_1^\top \mathbf{X}_1\right) \texttt{var}\left(w_2^\top \mathbf{X}_2\right)}} \\ &= \frac{w_1^\top C_{12} w_2}{\sqrt{\left(w_1^\top C_{11} w_1\right)\left(w_2^\top C_{22} w_2\right)}}, \end{aligned} \tag{3.2}$$

where covariance matrices $C_{11}, C_{22}, C_{12}$ are defined as follows:

$$C_{11} = \frac{1}{m}\sum_{i=1}^{m}\left(x_i^1 - \bar{x_1}\right)\left(x_i^1 - \bar{x_1}\right)^\top,$$
$$C_{22} = \frac{1}{m}\sum_{i=1}^{m}\left(x_i^2 - \bar{x_2}\right)\left(x_i^2 - \bar{x_2}\right)^\top,$$
$$C_{12} = \frac{1}{m}\sum_{i=1}^{m}\left(x_i^1 - \bar{x_1}\right)\left(x_i^2 - \bar{x_2}\right)^\top,$$

where $\bar{x_1}$ and $\bar{x_2}$ are the means for two views:

$$\bar{x_1} = \frac{1}{m}\sum_{i=1}^{m} x_i^1,$$
$$\text{and } \bar{x_2} = \frac{1}{m}\sum_{i=1}^{m} x_i^2.$$

Since the correlation (Equation (3.2)) does not change with a rescaling of $w_1$ and $w_2$, CCA can be formulated as following:

$$\begin{aligned} \max_{w_1, w_2} \quad & w_1^\top C_{12} w_2, \\ \texttt{s.t.} \quad & w_1^\top C_{11} w_1 = 1, \quad w_2^\top C_{11} w_2 = 1. \end{aligned} \tag{3.3}$$

The above maximization problem is solved using Lagrange multiplier technique [11] . The Lagrangian function for above problem is defined as:

$$L = w_1^\top C_{12} w_2 - \frac{\lambda_1}{2}\left(w_1^\top C_{11} w_1 - 1\right) - \frac{\lambda_2}{2}\left(w_2^\top C_{22} w_2 - 1\right), \tag{3.4}$$

where $\lambda_1$ and $\lambda_2$ are the Lagrange multipliers. Differentiating $L$ with respect to $w_1$ and $w_2$, we obtain

$$C_{12} w_2 - \lambda_1 C_{11} w_1 = 0, \tag{3.5}$$
$$\text{and} \quad C_{21} w_1 - \lambda_2 C_{22} w_2 = 0. \tag{3.6}$$

By multiplying $w_1^\top$ and $w_2^\top$, we respectively have

$$w_1^\top C_{12} w_2 - \lambda_1 w_1^\top C_{11} w_1 = 0,$$
$$\text{and} \quad w_2^\top C_{21} w_1 - \lambda_2 w_2^\top C_{22} w_2 = 0.$$

Since $w_1^\top C_{11} w_1 = 1$ and $w_2^\top C_{11} w_2 = 1$, we can deduce that $\lambda_1 = \lambda_2 = \lambda$. Substituting $\lambda$ in Equations (3.5) and (3.6) and solving for $w_1$ and $w_2$, we obtain:

$$w_1 = \frac{C_{11}^{-1} C_{12} w_2}{\lambda}, \tag{3.7}$$

$$\text{and} \quad C_{21} C_{11}^{-1} C_{12} w_2 = \lambda^2 C_{22} w_2. \tag{3.8}$$

Form above, Equation (3.8) is equivalent to solving standard eigenvalue problem given as

$$C_{22}^{-1} C_{21} C_{11}^{-1} C_{12} w_2 = \lambda^2 w_2.$$

Finally, the correlation between different views is provided by the eigenvector corresponding to the largest eigenvalues.

This method leverages only two views data, extending it to multiple views is achieved with generalized version of CCA by Kettenring [47]. Moreover, CCA based approaches does not scale well as for large training datasets the inversion of matrix $C_{11}$ in Equation (3.7) is tedious. CCA is a linear feature extraction algorithm and Akaho [1] derived a kernel version of original CCA algorithm (KCCA) to handle non-linear data. The non-linear projections learned by KCCA are limited by the choice of a fixed kernel, Andrew et al. [3] proposed deep CCA approach using neural networks in order to learn more flexible representations. In contrast to CCA, which ignores label information of input examples, Diethe et al. [25] generalized Fisher's Discriminant analysis to find the projections for multiview data which takes into account the labels of examples.

### 3.3.2 Co-training and Co-regularization

Semi-supervised learning is the problem of learning when we have both labeled and unlabeled training data. In many real-world scenarios, it is both expensive and time consuming to annotate the data. Therefore, it is interesting to consider both labeled and unlabeled data in order to learn an effective classification model.

The Co-training algorithm proposed by *Blum and Mitchell* [9] is a classical algorithm in multiview semi-supervised learning. This algorithm combines both labeled and unlabeled data under the two-view setting. It iteratively learns two classifiers corresponding to each view using the labeled data. Then at each iteration, the learner on one view is used to label the unlabeled data which is added to the training pool of the other learner. In this way, on the unlabeled data, classifiers learned on two views exchange informations on the two views

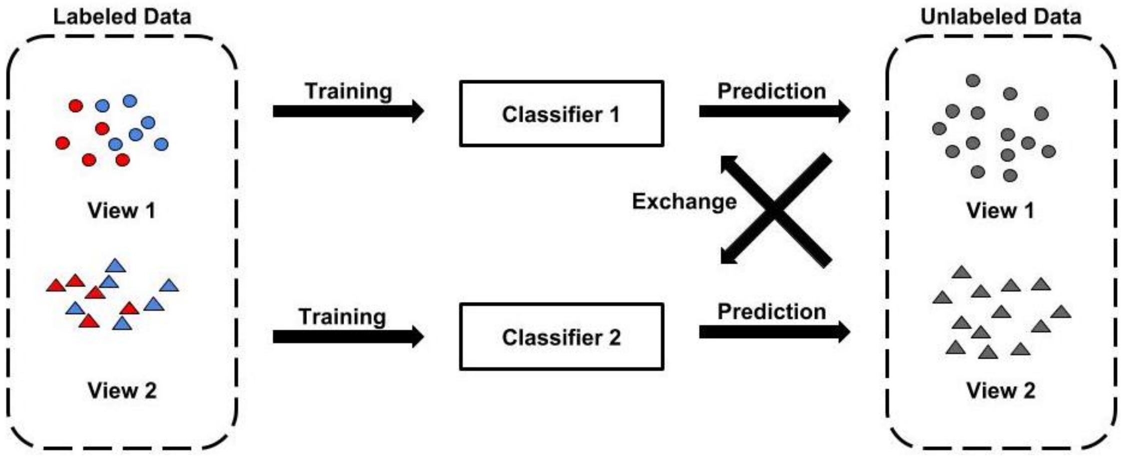


Figure 3.3: Co-training style algorithm: It iteratively learns two classifiers corresponding to each view of data. Then, the classifier on one view labels the unlabeled data for another view.

(see Figure 3.3). Finally, the learned view-specific classifiers are used separately or jointly to make predictions on new examples.

The Co-EM algorithm, proposed by Nigam and Ghani [62], is a variant of co-training algorithm which combines the co-training with the probabilistic expectation maximization approach [24]. Basically, it gives unlabeled examples probabilistic labels using the naive Bayes algorithm [28, 33]. As SVM (described in Section 2.4.1) is known to be a better fit for many classification problems, Brefeld and Scheffer [12] developed the SVM version of the co-EM algorithm.

Co-regularization based approaches can be seen as regularized versions of co-training algorithm. The co-regularization based algorithms [69] return two classifiers corresponding to each view by simultaneously maximizing the agreement between the two views on unlabeled data and the empirical error on the labeled data. Sindhwani et al. [78] proposed a co-regularized least squared approach for multiview learning with two views where the following objective function is optimized:

$$\begin{aligned}(h_1^*, h_2^*) = \underset{h_1 \in \mathcal{H}_1, h_2 \in \mathcal{H}_2}{\operatorname{argmin}} \sum_{i \in S_l} \left[y_i - h_1(x_i^1)\right]^2 + \mu \sum_{i \in S_l} \left[y_i - h_2(x_i^1)\right]^2 \\ + \gamma_1 ||h_1||^2_{\mathcal{H}_1} + \gamma_2 ||h_2||^2_{\mathcal{H}_2} + \frac{\gamma_C}{m_u} \sum_{i=1}^{m_u} \left[h_1\left(x_i^1\right) - h_2\left(x_i^2\right)\right]^2,\end{aligned} \tag{3.9}$$

where parameter $\mu, \gamma_1, \gamma_2$ and $\gamma_C$ are regularization parameters. In Equation (3.9), the first two terms evaluate the classification error made by the labeled data on the two views, the third and fourth terms measure the complexity of the hypothesis space using $L_2$ norm and the final term enforces the agreement among the classifiers on unlabeled data. Finally, the prediction for any new input example **x** is given as follows:

$$h^*(\mathbf{x}) = \text{sign}\left[\frac{1}{2}\left(h_1^*(x^1) + h_2^*(x^2)\right)\right].$$

Following a similar strategy as co-training, Amini et al. [2] proposed a *self-learning multiview algorithm* for more general and natural of multiview learning with more than two views. Firstly, it learns view-specific classifiers for each view separately using the labeled data. Given the view-specific classifiers, it iteratively assigns the labels (which is generally referred as pseudo-labels) to the unlabeled data for which all classifier predictions agree. Finally, it trains the new view-specific classifiers using the labeled data and pseudo-labeled unlabeled examples.

### 3.3.3 SVM-like Algorithms

Farquhar et al. [30] combined the kernel Canonical Correlation Analysis (KCCA) with the Support Vector Machines (recalled in Section 2.4.1) to derive a single optimization problem called as SVM-2K for two views. This is done by introducing the agreement constraint between the projections of two SVMs corresponding to each view. Formally, the constraint is defined as:

$$\left|w_1^\top \phi_1(x_i^1) + b_1 - w_2^\top \phi_2(x_i^2) - b_2\right| \le \eta_i + \epsilon$$

where $w_1$ and $w_2$ are the weight vectors for two SVMs, $b_1$ and $b_2$ are biases for two SVMs and $\eta_i$ is the slack variable. Note that $\phi_v(x_i^v)$ is the feature projection (ideally, in high dimension) for any example $\mathbf{x}_i$ for view $v \in \{1,2\}$ with corresponding kernel function $K_v$. Basically, the above constraint measures the amount by which the examples fail to meet $\epsilon$ similarity. Finally, combining the above constraint with the usual SVM constraints for two views leads to the minimization of the following optimization problem:

$$\begin{aligned}
\min_{w_1,w_2,b_1,b2} \quad & \frac{1}{2}||w_1||^2 + \frac{1}{2}||w_2||^2 + C_1\sum_{i=1}^{m}\xi_i^1 + C_2\sum_{i=1}^{m}\xi_i^2 + D\sum_{i=1}^{m}\eta_i \\
s.t. \quad & \left|w_1^\top \phi_1(x_i^1) + b_1 - w_2^\top \phi_2(x_i^2) - b_2\right| \le \eta_i + \epsilon, \quad \forall 1 \le i \le m \\
& y_i\left(w_1^\top \phi_1(x_i^1) + b_1\right) \ge 1 - \xi_i^1, \quad \forall 1 \le i \le m \\
& y_i\left(w_2^\top \phi_2(x_i^2) + b_2\right) \ge 1 - \xi_i^2, \quad \forall 1 \le i \le m \\
& \xi_i^1 \ge 0,\ \xi_i^2 \ge 0,\ \eta_i \ge 0, \quad \forall 1 \le i \le m
\end{aligned}$$

where, $\xi^1$ and $\xi^2$ are the slack variables and $C_1, C_2$ and $D$ are the hyperparameters. The final prediction function for any input example $\mathbf{x}$ is defined as follows:

$$h(\mathbf{x}) = \frac{1}{2}\,\mathrm{sign}\left[\left(w_1^\top \phi_1(x^1) + b_1\right) + \left(w_1^\top \phi_2(x^2) + b_2\right)\right].$$

SVM-2K exploits only two-view data. However, in many real world scenarios we can have data represented by more than two views. In this thesis, we are interested in more general case of supervised multiview learning when we have data represented by more than two views.

In order to handle multiview learning with more than two views, one natural solution is to see multiview learning as combination of different view-specific classifiers corresponding to each view. The goal is to learn a final combination over the set of view-specific classifiers which performs better than individual view-specific classifiers (as shown in figure 3.2). Amini et al. [2] proposed a multiview majority voting scheme (`MV-MV`) for more than two views, where view-specific classifiers are learned using SVM by minimizing the following empirical risk:

$$h_v^* = \underset{h_v \in \mathcal{H}_v}{\mathrm{argmin}} \frac{1}{m} \sum_{(\mathbf{x}_i, y_i) \sim (\mathcal{D})^m} \mathbb{1}_{[h_v(x_i^v) \neq y_i]}. \tag{3.10}$$

Finally, the prediction for a multiview example $\mathbf{x}$ is then based on the majority vote over these view-specific classifiers:

$$\mathtt{MV\text{-}MV}(\mathbf{x}) = \mathrm{sign}\left[\frac{1}{V}\sum_{v=1}^{V} h_v^*(x^v)\right]. \tag{3.11}$$

Amini et al. [2] proposed a Rademacher analysis of the risk of above multiview majority vote classifier. The generalization bound is given by following theorem

**Theorem 3.1.** *(Generalization bound for `MV-MV`). Let $\mathcal{D}$ be an unknown distribution $\mathcal{X} \times \mathcal{Y}$, for each view $v \in \mathcal{V}$ we consider a continuous hypothesis space $\mathcal{H}_v$, with probability of at least $1-\delta$ on the random choice of the learning sample $S \sim (\mathcal{D})^m$, we have:*

$$R_{\mathcal{D}}(\mathtt{MV-MV}) \leq \frac{1}{V}\sum_{v=1}^{V} R_S(h_v^*) + \frac{2}{V}\sum_{v=1}^{V} \mathfrak{R}_S(\mathcal{H}_v) + 6\sqrt{\frac{\log\frac{2}{\delta}}{2m}}.$$

*where $\mathfrak{R}_S(\mathcal{H}_v)$ is the empirical Rademacher complexity for hypothesis space $\mathcal{H}_v$ defined by Equation* (2.6).

### 3.3.4 PAC-Bayesian Analysis of Multiview Learning

Sun et al. [81] proposed a PAC-Bayesian analysis (introduced in Section 2.3.2) for co-regularization style multiview learning approaches with two views. In order to derive PAC-Bayesian generalization bounds, they considered the linear classifiers of the following form

$$h(\mathbf{x}) = \operatorname{sign}\left(u^\top \phi(X)\right), \tag{3.12}$$

where $u = \left[u_1^\top, u_2^\top\right]^\top$ is the concatenated weight vector for two views and $\phi(x)$ is the kernel-induced feature projection for concatenated views where $X = \left[x^{1^\top}, x^{2^\top}\right]^\top$. Similar to the usual PAC-Bayesian theory, they assume a prior distribution for a classifier defined as

$$P(u) \propto \mathcal{N}(0, \mathbf{I}) \times V(u_1, u_2), \tag{3.13}$$

where $\mathcal{N}(0, \mathbf{I})$ is a Gaussian distribution with zero mean and identity covariance matrix $\mathbf{I}$, and

$$V(u_1, u_2) = \exp\left\{-\frac{1}{2\sigma^2} \underset{(x^1,x^2)}{\mathbb{E}} \left(x^{1^\top} u_1 - x^{2^\top} u_2\right)^2\right\},$$

$V(u_1, u_2)$ emphasizes those classifiers which has high view agreements. By defining $\bar{X} = \left[x^{1^\top}, -x^{2^\top}\right]^\top$ and solving the prior distribution given by Equation 3.13, we have

$$\begin{aligned} P(u) &\propto \mathcal{N}(0, \mathbf{I}) \times V(u_1, u_2) \\ &\propto \exp\left\{-\frac{1}{2} u^\top \left(\mathbf{I} + \frac{\mathbb{E}(\bar{X}\bar{X}^\top)}{\sigma^2}\right) u\right\}. \end{aligned}$$

Finally, $P(u) = \mathcal{N}(0, \sum)$, where $\sum = \left(\mathbf{I} + \frac{\mathbb{E}(\bar{X}\bar{X}^\top)}{\sigma^2}\right)^{-1}$. The posterior is chosen to be of the following form

$$Q(u) = \mathcal{N}\left(\mu w, \mathbf{I}\right), \tag{3.14}$$

where $||w|| = 1$ and therefore the distance between the center of posterior and origin is $\mu$. Finally, the PAC-Bayesian generalization bound for multiview learning with two views in specific case of linear classifiers is given by the following theorem

**Theorem 3.2.** *(PAC-Bayesian generalization bound for multiview learning [81]) For linear classifier (Equation* (3.12)*) with prior and posterior given by Equations* (3.13) *and* (3.14)*, for any unknown distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, with a high probability $1 - \delta$ over the random choice of learning sample $S \sim (\mathcal{D})^m$, we have*

$$\forall w, \mu: \; KL\Big( R_S\big(h(w,\mu)\big) \,\big|\big|\, R_{\mathcal{D}}\big(h(w,\mu)\big) \Big) \leq$$
$$\frac{-d \, \ln\left[f - \left(\sqrt[2]{(R/\sigma)^2 + 1} - 1\right)\sqrt{\frac{1}{2m} \, \ln\frac{3}{\delta}}\right] + \frac{H}{\sigma^2} + \frac{(1+\mu^2)R^2}{2\sigma^2}\sqrt{\frac{1}{2m} \, \ln\frac{3}{\delta}} + \mu^2 + 2\ln\left(\frac{m+1}{\delta/3}\right)}{2m},$$

*where*

$$h(w,\mu) = F\left(\frac{\mu\, y\, w^\top \phi(x)}{||\phi(x)||}\right),$$
$$f = \frac{1}{m}\sum_{i=1}^{m}\left|\mathbf{I} + \frac{X_i X_i^\top}{\sigma^2}\right|^{\frac{d}{2}},$$
$$H = \frac{1}{m}\sum_{i=1}^{m}\left[X_i^\top X_i + \mu^2\left(w^\top X_i\right)^2\right],$$

$||w|| = 1$*, $d$ is the sum of dimensions of both views, $R = \sup_X ||X||$ and $F(z)$ is the Gaussian cumulative distribution*

$$F(z) = \int_z^\infty \frac{1}{\sqrt{2\pi}} e^{\frac{-z^2}{2}}\, dz.$$

The above generalization bound has allowed them to derive a SVM-like learning algorithm but limited to two views. The objective function of the multiview SVMs (`MvSVMs`) is given by:

$$\begin{aligned}
\min_{w_1,w_2,\xi_1,\xi 2} \quad & \frac{1}{2}\left(||w_1||^2 + ||w_2||^2\right) + C_1\sum_{i=1}^{m}\left(\xi_i^1 + \xi_i^2\right) + C_2\sum_{i=1}^{m}\left(w_1^\top x_i^1 - w_2^\top x_i^2\right)^2 \\
s.t. \quad & y_i\left(w_1^\top x_i^1\right) \geq 1 - \xi_i^1, \quad \forall 1 \leq i \leq m \\
& y_i\left(w_2^\top x_i^2\right) \geq 1 - \xi_i^2, \quad \forall 1 \leq i \leq m \\
& \xi_i^1 \geq 0,\ \xi_i^2 \geq 0, \quad \forall 1 \leq i \leq m
\end{aligned}$$

where $w_1$ and $w_2$ are the weight vectors for two SVMs corresponding to each view, $\xi^1$ and $\xi^2$ are the slack variables and $C_1$ and $C_2$ are the hyperparameters. The final prediction function for any input example $\mathbf{x}$ is given as follows:

$$h^*(\mathbf{x}) = \frac{1}{2}\,\mathrm{sign}\left[\left(w_1^\top x^1\right) + \left(w_1^\top x^2\right)\right].$$

Note that they also extended above PAC-Bayesian bound for a data dependent prior distribution and the semi-supervised learning setting. In this thesis, our objective is to derive PAC-Bayesian generalization bounds for more general and natural case of multiview learning with more than two views and not limited to linear classifiers.

## 3.4 Diversity Principle

The diversity principle demonstrates that in a multiview learning problem, each representation or view of the data may contain some information which other views do not have. Intuitively, while combining different views, we want views to be as accurate as possible, in case they make errors, these errors should be on different examples. In contrary to consensus principle, the diversity based approaches tries to increase the disagreement between the views while controlling the accuracy of the view-specific classifiers. Many multiview learning algorithms based on ensemble learning have been proposed which takes into account the diversity between views [41, 46, 48, 66, 67, 90, 94] in different manners.

Janodet et al. [46] proposed a boosting based multiview learning algorithm for 2 views, called 2-Boost. At each iteration, the algorithm learns the weights over the view-specific voters by maintaining a single distribution over the learning examples. Conversely, Koço et al. [48] proposed Mumbo that maintains separate distributions for each view. For each view, the algorithm reduces the weights associated with the examples that are hard to classify, and increases the weights of those examples on the other views. This trick allows a communication between the views such that other views can compensate the information lacked by a particular view.

Xu and Sun [94] proposed EMV-AdaBoost, an embedded multiview Adaboost algorithm, restricted to two views. At each iteration, an example contributes to the error if it is misclassified by any of the view-specific voters and the diversity between the views is captured by weighting the error by the agreement between the views. Xiao and Guo [90] derived a weighted majority voting Adaboost algorithm `MVWAB` (for more than two views) which learns weights over view-specific voters at each iteration of the algorithm. Peng et al. [66, 67] proposed variants of Boost.SH (boosting with SHared weight distribution) which controls the diversity for more than two views. They maintain a single global distribution over the learning examples for all the views. In order to control the diversity between the views, at each iteration they update the distribution over the views by casting the algorithm in two ways: *i)* a multiarmed bandit framework (`rBoost.SH`) and *ii)* an expert strategy framework (`eBoost.SH`) consisting of set of strategies (distribution over views) for weighing views.

Moreover, Morvant et al [61] proposed a late fusion[79] approach to handle multimedia data in a PAC-Bayesian fashion, but without any theoretical justifications and in a ranking setting. Concretely, they learn a multiview model over the predictions of view-specific classifiers using a PAC-Bayesian algorithm MinCq [72]. MinCq algorithm is based on $\mathcal{C}$-Bound (given by Theorem 2.4) which is able to control the trade-off between the accuracy and the diversity between the view-specific classifiers.

In the next sections, we formally present `MVWAB` and `rBoost.SH` algorithms.

### 3.4.1 Multiview Weighted Adaboost (MVWAB)

**Algorithm 3** MVWAB

```
Input: Training set $S = \{(\mathbf{x}_i, y_i), \dots, (\mathbf{x}_m, y_m)\}$, where $\mathbf{x}_i = (x^1, x^2, \dots, x^V)$ and $y_i \in \{-1, 1\}$.
       For each view $v \in \mathcal{V}$, a view-specific hypothesis set $\mathcal{H}_v$.
       Number of iterations $T$.
  for $\mathbf{x}_i \in S$ do
      $\mathcal{D}_1(\mathbf{x}_i) \leftarrow \frac{1}{n}$
  for $t = 1, \dots, T$ do
      $\forall v \in \mathcal{V},\ h_v^{(t)} \leftarrow \operatorname{argmin}_{h_v \in \mathcal{H}_v} \mathop{\mathbb{E}}_{(\mathbf{x}_i, y_i) \sim \mathcal{D}_{(t)}} \left[ \mathbb{1}_{[h_v(x_i^v) \neq y_i]} \right]$
      Optimize the weighted least square loss to learn the weight parameters $\{\beta_v\}_{v=1}^{V}$
          $\min_{\beta} \ \sum_{i=1}^{m} \mathcal{D}_{(t)}(i) \Big( \sum_{v=1}^{V} \beta_v h_v^{(t)}(x_i^v) - y_i \Big)^2,$
          $s.t. \quad 0 \leq \beta_v \leq 1 \text{ and } \sum_{v=1}^{V} \beta_v = 1.$
      Compute the base classifier $h^{(t)}(\mathbf{x}) = \operatorname{sign}\Big( \sum_{v=1}^{V} \beta_v h_v^{(t)}(x^v) \Big)$
      Compute error: $\epsilon^{(t)} \leftarrow \mathop{\mathbb{E}}_{(\mathbf{x}_i, y_i) \sim \mathcal{D}_{(t)}} \left[ \mathbb{1}_{[h^{(t)}(x_i^v) \neq y_i]} \right]$
      Compute weight over the base classifier:
          $Q^{(t)} \leftarrow \frac{1}{2} \Big[ \ln \Big( \frac{1 - \epsilon^{(t)}}{\epsilon^{(t)}} \Big) \Big]$
      for $\mathbf{x}_i \in S$ do
          $\mathcal{D}_{(t+1)}(\mathbf{x}_i) \leftarrow \frac{\mathcal{D}_{(t)}(\mathbf{x}_i) \exp\big(-y_i Q^{(t)} h^{(t)}(\mathbf{x}_i)\big)}{\sum_{j=1}^{m} \mathcal{D}_{(t)}(\mathbf{x}_j) \exp\big(-y_j Q^{(t)} h^{(t)}(\mathbf{x}_j)\big)}$
  Return: $B_Q^{\mathrm{MV}}(\mathbf{x}) = \sum_{t=1}^{T} Q^{(t)} h^{(t)}(x)$
```

Xiao et al. [90] combined multiview learning and the Adaboost (presented in Section 2.4.2) techniques to derive a boosting based algorithm (see Algorithm 3) for multiview learning. At each iteration $t$, the Multiview Weighted Adaboost (MVWAB) algorithm separately trains the view-specific classifiers $h_v^{(t)}$ for each view $v \in \mathcal{V}$ using the probability distribution ($\mathcal{D}_{(t)}$) over the learning sample $S$. Then, the base classifier $h^{(t)}$ is a linear combination of the view-specific classifiers which are weighted according to the set of weight parameters $\{\beta_v\}_{v=1}^{V}$:

$$h^{(t)}(\mathbf{x}) = \operatorname{sign}\Big( \sum_{v=1}^{V} \beta_v h_v^{(t)}(x^v) \Big), \tag{3.15}$$

where $0 \le \beta_v \le 1$ and $\sum_{v=1}^{V} \beta_v = 1$. The weight parameters $\{\beta_v\}_{v=1}^{V}$ are obtained by minimizing the following weighted least square loss:

$$\begin{aligned} \min_{\beta} \quad & \sum_{i=1}^{m} \mathcal{D}_{(t)}(i) \Big( \sum_{v=1}^{V} \beta_v h_v^{(t)}(x_i^v) - y_i \Big)^2, \\ s.t. \quad & 0 \le \beta_v \le 1 \text{ and } \sum_{v=1}^{V} \beta_v = 1. \end{aligned} \tag{3.16}$$

Following the similar strategy as in Adaboost, weight over the base classifier (Equation 3.15) is computed as follows:

$$Q^{(t)} = \frac{1}{2} \Big[ \ln \Big( \frac{1 - \epsilon^{(t)}}{\epsilon^{(t)}} \Big) \Big],$$

where $\epsilon^{(t)} = \mathop{\mathbb{E}}_{(\mathbf{x}_i, y_i) \sim \mathcal{D}_{(t)}} \big[ \mathbb{1}_{h^{(t)}(\mathbf{x}_i) \neq y_i} \big]$ is the classification error of the base classifier. Finally, the distribution $\mathcal{D}_{(t)}$ over the learning sample $S$ is updated as follows:

$$\mathcal{D}_{(t+1)}(\mathbf{x}_i) \leftarrow \frac{\mathcal{D}_{(t)}(\mathbf{x}_i) \exp\big(-y_i Q^{(t)} h^{(t)}(x_i^v)\big)}{\sum_{j=1}^{m} \mathcal{D}_{(t)}(\mathbf{x}_j) \exp\big(-y_j Q^{(t)} h^{(t)}(x_j^v)\big)}.$$

Finally, after $T$ iterations, the algorithm returns following classifier:

$$B_Q^{\text{MV}}(x) = \sum_{t=1}^{T} Q^{(t)} h^{(t)}(x). \tag{3.17}$$

### 3.4.2 Randomized Boosting with SHared weight distribution (rBoost.SH)

Peng et al. [66, 67] proposed a multiview boosting algorithm called rBoost.SH that learns the view-specific classifiers independently for each view but maintains a shared distribution over the learning sample to propagate the information among the different views. Moreover, in order to capture the diversity between the views, they learn a distribution over the views using the multiarmed bandit framework [5]. In multiarmed bandit framework, an algorithm tries one out of $V$ actions (in this case, number of views) at any time $t$. For each action, there is an associated reward and the objective of the algorithm is to maximize the total reward after taking the actions over a period of time.

rBoost.SH maintains a probability distribution $\pi$ over the views. It updates the distribution $\pi$ according to the estimated cumulative reward at step $10(ii)$ of the Algorithm 4. At step 5, a view is chosen according to the distribution $\rho$ which is weighted combination of uniform

**Algorithm 4** `rBoost.SH`

```
Input: Training set $S = \{(\mathbf{x}_i, y_i), \dots, (\mathbf{x}_m, y_m)\}$, where $\mathbf{x}_i = (x^1, x^2, \dots, x^V)$ and $y_i \in \{-1, 1\}$.
       For each view $v \in \mathcal{V}$, a view-specific hypothesis set $\mathcal{H}_v$.
       Number of iterations $T$, $\sigma > 0$ and $\gamma \in (0, 1]$.
for $\mathbf{x}_i \in S$ do
    $\mathcal{D}_1(\mathbf{x}_i) \leftarrow \frac{1}{n}$
$\forall v \in \mathcal{V},\ \pi_v^1 \leftarrow \exp\left(\frac{\sigma\gamma}{3}\sqrt{\frac{T}{M}}\right)$
for $t = 1, \dots, T$ do
    $\forall v \in \mathcal{V}, \rho_v^{(t)} \leftarrow (1-\gamma)\frac{\pi_v^{(t)}}{\sum_{i=1}^{V}\pi_i^{(t)}} + \frac{\gamma}{V}$
    Let $j$ be the selected view according to distribution $\rho^{(t)}$
    For $j$ -th view, $h_j^{(t)} \leftarrow \text{argmin}_{h_j \in \mathcal{H}_j} \underset{(\mathbf{x}_i, y_i)\sim\mathcal{D}_t}{\mathbb{E}} \left[\mathbb{1}_{[h_j(x_i^j) \neq y_i]}\right]$
    Compute edge $\theta$ of view-specific weak classifier $h_j^{(t)}$: $\theta_{(t)}(j) = \underset{(\mathbf{x}_i, y_i)\sim\mathcal{D}_{(t)}}{\mathbb{E}} \left[y_i h_j^{(t)}(x_i^j)\right]$
    Compute the reward function for $j$ -th view: $r_{(t)}(j) = \sqrt{1 - \theta_{(t)}^2(j)}$
    $\forall v \in \mathcal{V}$, set:
        i) $\hat{r}_{(t)}(v) = \begin{cases} \frac{r_{(t)}(v)}{\rho_v^{(t)}} & \text{if } v = j \\ 0 & \text{otherwise} \end{cases}$
        ii) $\pi_v^{(t+1)} = \pi_v^{(t)} \exp\left(\frac{\gamma}{3V}\left(\hat{r}_{(t)}(v) + \frac{\sigma}{\rho_v^{(t)}\sqrt{TV}}\right)\right)$
    Let $h_*^{(t)} = h_j^{(t)}$ and compute weight over the classifier:
        $Q^{(t)} \leftarrow \frac{1}{2}\left[\ln\left(\frac{1+\theta_{(t)}(j)}{1-\theta_{(t)}(j)}\right)\right]$
    for $\mathbf{x}_i \in S$ do
        $\mathcal{D}_{(t+1)}(\mathbf{x}_i) \leftarrow \frac{\mathcal{D}_{(t)}(\mathbf{x}_i)\exp\left(-y_i Q^{(t)} h_*^{(t)}(x_i^*)\right)}{\sum_{j=1}^{m}\mathcal{D}_{(t)}(\mathbf{x}_j)\exp\left(-y_j Q^{(t)} h_*^{(t)}(x_j^*)\right)}$
Return: $B_Q^{\text{MV}}(\mathbf{x}) = \sum_{t=1}^{T} Q^{(t)} h_*^{(t)}(x^*)$
```

distribution over the views and $\pi$, which encourages exploration of different views. For the chosen view $j$, a weak view-specific classifier is learned and weight over the classifier is computed as following:

$$Q^{(t)} \leftarrow \frac{1}{2}\left[\ln\left(\frac{1+\theta_{(t)}(j)}{1-\theta_{(t)}(j)}\right)\right],$$

where $\theta_{(t)}(j) = \underset{(\mathbf{x}_i,y_i)\sim\mathcal{D}_t}{\mathbb{E}}\left[y_i h_j^{(t)}(x_i^j)\right]$ is the edge of view-specific weak classifier. Then, the distribution $\mathcal{D}_{(t)}$ over the learning sample $S$ is updated as follows:

$$\mathcal{D}_{(t+1)}(\mathbf{x}_i) \leftarrow \frac{\mathcal{D}_{(t)}(\mathbf{x}_i)\exp\left(-y_i Q^{(t)} h_*^{(t)}(x_i^*)\right)}{\sum_{j=1}^{m}\mathcal{D}_{(t)}(\mathbf{x}_j)\exp\left(-y_j Q^{(t)} h_*^{(t)}(x_j^*)\right)},$$

where $h_*^{(t)}$ is the learned view-specific classifier at iteration $t$. Finally, after $T$ iterations, `rBoost.SH` returns a weighted majority of $T$ view-specific classifiers:

$$B_Q^{\mathrm{MV}}(\mathbf{x}) = \sum_{t=1}^{T} Q^{(t)} h_*^{(t)}(x^*).$$

## 3.5 Conclusion

In this chapter, we introduced basic concepts and background for multiview learning where we have multiple representations or views of the input data. The objective of multiview learning is to learn a multiview classifier which takes into account different views of the data in order to improve the learning performance. We present two fundamental principles of multiview learning i.e. *i)* consensus, and *ii)* diversity principles. In addition, for each principle, we introduced some of multiview learning algorithms.

In this thesis, we derive a PAC-Bayesian generalization bound for multiview learning (with more than two views) on the risk of the multiview majority vote which exhibits a term of diversity in the predictions of the view-specific classifiers. This is done by considering a hierarchy of distributions over the view-specific classifiers and views. Based on this hierarchy of weights, we derive three multiview learning algorithms. We will discuss these contributions in the next part of this thesis.

# Part II

# Contributions

# CHAPTER 4

# THE PAC-BAYESIAN THEOREM AS AN EXPECTED RISK BOUND

The PAC-Bayesian theory provides generalization guarantees for classifiers expressed as a weighted combination of voters. In this chapter, we derive a new kind of PAC-Bayesian generalization bounds which are expressed as expected risk bounds instead of probabilistic bound (presented in Section 2.3.2 of Chapter 2). Note that in this chapter, we are in single view setting and we present extension to multiview learning of Chapter 5. This work has been done in collaboration with Dr. Pascal Germain from INRIA, Lille, France. It has been accepted at CAp, 2017 [40] and published in the proceedings of ECML-PKDD, 2017 [41].

## 4.1 Introduction

The PAC-Bayesian approach introduced by McAllester [57] provides Probably Approximately Correct (PAC) generalization guarantees for models expressed as a weighted majority vote over the hypothesis space $\mathcal{H}$.[1] In this framework one assumes a prior distribution $P$ over $\mathcal{H}$ which models the *a priori* weights associated with each classifier[2] in $\mathcal{H}$. After observing the learning sample $S$, the learner aims at finding a posterior distribution $Q$ (that modalizes the weight associated to each voter in the majority vote ) over $\mathcal{H}$ that leads to a well-performing majority vote (see Figure 4.1). It is well-known that the error of deterministic majority vote is upper bounded by twice the error of stochastic Gibbs classifier. PAC-Bayesian theorems

[1] Note that the majority vote setting is not too restrictive since many machine learning approaches can be considered as majority vote learning, notably ensemble methods [26, 71] (as pointed out in Section 2.3 in Chapter 2).

[2] For example, the voters expected to be most accurate for the task can have the largest weights under $P$.

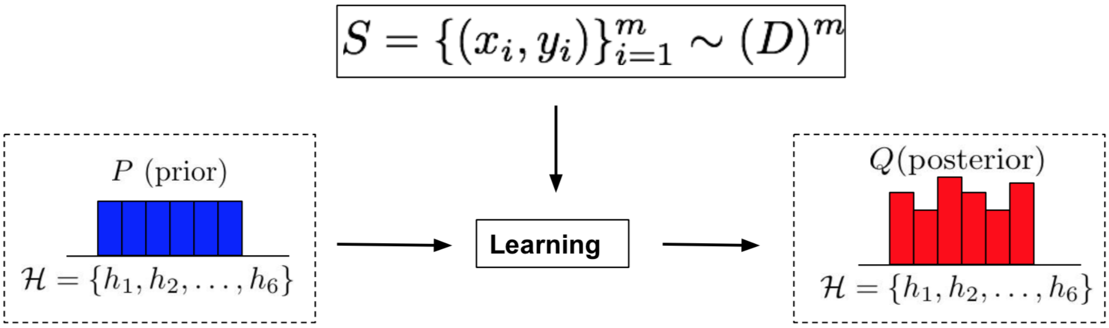

$S = \{(x_i, y_i)\}_{i=1}^m \sim (D)^m$
$P$ (prior)
$\mathcal{H} = \{h_1, h_2, \dots, h_6\}$
Learning
$Q$(posterior)
$\mathcal{H} = \{h_1, h_2, \dots, h_6\}$

Figure 4.1: The PAC-Bayesian theory assumes a prior distribution $P$ (in blue) over the hypothesis space $\mathcal{H}$ and aims at learning — from the learning sample $S$ — a posterior distribution $Q$ (in red).

(*e.g.,* [14, 34, 35, 54, 57, 58, 77]) typically provide generalization bounds on the true risk of the Gibbs classifier—*a fortiori* of the majority vote—uniformly for all learned distribution $Q$, but with a high probability over the random choice of the learning sample $S$.

Since by definition the learned posterior distribution is data dependent, we propose in this chapter a new formulation of the PAC-Bayesian theorem that aims at bounding the expectation directly by the risk of the Gibbs classifier over all the possible learning samples of a given size $m$, *i.e.,* we upper bound the expectation over all posterior distributions we can learn from all the possible learning samples of a given size $m$. Our new PAC-Bayesian theorem is then not anymore expressed as a probabilistic bound, but as an expectation risk bound, bringing another point of view on the PAC-Bayesian analysis.

## 4.2 The Usual PAC-Bayesian Theory

In this section, we first recall the usual PAC-Bayesian theorem in the form proposed by Germain et al. [34, 35]. This general result can be seen as a theoretical tool to recover most of the known PAC-Bayesian *probabilistic bounds* (among them, the one recalled in Theorem 2.3 in Chapter 2). Then, in Section 4.3, we provide a novel formulation of the PAC-Bayesian theorem expressed as an *expectation bound.*

### 4.2.1 Notations and Setting

For a binary classification task, in a single view setting, on data drawn from a fixed yet unknown distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, the PAC-Bayesian approach assumes a prior distribution

$P$ over the hypothesis space $\mathcal{H}$ that models on *a priori* belief[3] on the classifiers from $\mathcal{H}$ before the observation of the learning sample $S$. Given the learning sample $S = \{(x_i, y_i)\}_{i=1}^m \sim (\mathcal{D})^m$, the learner objective is then to find a posterior distribution $Q$ over $\mathcal{H}$ leading to an accurate weighted majority vote $B_Q(x)$ defined as

$$B_Q(x) = \mathrm{sign}\left[\mathop{\mathbb{E}}_{h\sim Q} h(x)\right] = \mathrm{sign}\left[\frac{1}{|\mathcal{H}|}\sum_{i=1}^{|\mathcal{H}|} Q(h)h(x)\right].$$

In other words, one wants to learn $Q$ over $\mathcal{H}$ such that it minimizes the true risk $R_{\mathcal{D}}(B_Q)$ of $B_Q(x)$:

$$R_{\mathcal{D}}(B_Q) = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \mathbb{1}_{[B_Q(x)\neq y]}\,,$$

where $\mathbb{1}_{[p]} = 1$ if predicate $p$ holds, and 0 otherwise. However, a PAC-Bayesian generalization bound does not directly focus on the risk of the deterministic weighted majority vote $B_Q$. Instead, it upper-bounds the risk of the stochastic Gibbs classifier $G_Q$, which predicts the label of an example $x$ by drawing $h$ from $\mathcal{H}$ according to the posterior distribution $Q$ and predicts $h(x)$. Therefore, the true risk $R_D(G_Q)$ of the Gibbs classifier on a data distribution $\mathcal{D}$, and its empirical risk $R_S(G_Q)$ estimated on a sample $S \sim (\mathcal{D})^m$ are respectively given by

$$\begin{aligned} R_{\mathcal{D}}(G_Q) &= \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\ \mathop{\mathbb{E}}_{h\sim Q} \mathbb{1}_{[h(x)\neq y]}\,, \\ \text{and}\quad R_S(G_Q) &= \frac{1}{m}\sum_{i=1}^{m} \mathop{\mathbb{E}}_{h\sim Q} \mathbb{1}_{[h(x_i)\neq y_i]}\,. \end{aligned}$$

The above Gibbs classifier is closely related to the weighted majority vote $B_Q$. Indeed, if $B_Q$ misclassifies $x \in \mathcal{X}$, then at least half of the classifiers (under measure $Q$) make an error on $x$. Therefore, we have

$$R_{\mathcal{D}}(B_Q) \le 2R_{\mathcal{D}}(G_Q). \tag{4.1}$$

Thus, an upper bound on $R_{\mathcal{D}}(G_Q)$ gives rise to an upper bound on $R_{\mathcal{D}}(B_Q)$. Other tighter relations exist [35, 51, 54], such as the so-called $\mathcal{C}$-Bound [51] that involves the *expected disagreement* $d_{\mathcal{D}}(Q)$ between all the pair of classifiers, and that can be expressed as follows (when $R_{\mathcal{D}} < \frac{1}{2}$):

$$R_{\mathcal{D}}(B_Q) \ \le\ 1 - \frac{\big(1-2R_{\mathcal{D}}(G_Q)\big)^2}{1-2d_{\mathcal{D}}(Q)}, \tag{4.2}$$

where $d_{\mathcal{D}}(Q)$ is the expected disagreement between the pair of classifiers, defined as:

$$d_{\mathcal{D}}(Q) = \mathop{\mathbb{E}}_{x\sim\mathcal{D}_{\mathcal{X}}}\ \mathop{\mathbb{E}}_{(h,h')\sim Q^2} \mathbb{1}_{[h(x)\neq h'(x)]}.$$

[3] When one has no priori information, one usually use $P$ as uniform distribution as shown in Figure 4.1.

Note that, we presented the another form of the $\mathcal{C}$-bound in Section 2.4.3 of Chapter 2 and we provide the proof of $\mathcal{C}$-bound in Appendix B.1. Moreover, Germain et al. [35] have shown that the Gibbs classifier's risk can be rewritten in terms of $d_{\mathcal{D}}(Q)$ and *expected joint error* $e_{\mathcal{D}}(Q)$ between all the pair of classifiers as

$$\begin{aligned} R_{\mathcal{D}}(G_Q) &= \frac{1}{2} d_{\mathcal{D}}(Q) + e_{\mathcal{D}}(Q), \qquad (4.3) \\ \text{where} \quad e_{\mathcal{D}}(Q) &= \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \mathop{\mathbb{E}}_{(h,h')\sim Q^2} \mathbb{1}_{[h(x)\neq y]} \mathbb{1}_{[h'(x)\neq y]}. \end{aligned}$$

It is worth noting that from a multiview learning standpoint where the notion of diversity among classifiers is known to be important [2, 4, 50, 56, 81], Equations (4.2) and (4.3) directly capture the trade-off between diversity and accuracy. Indeed, $d_{\mathcal{D}}(Q)$ involves the diversity between classifiers, while $e_{\mathcal{D}}(Q)$ takes into account the errors. Note that the principle of controlling the trade-off between diversity and accuracy through the $\mathcal{C}$-bound of Equation (4.2) (also Section 2.4.3 of Chapter 2) has been exploited by Roy et al. [72, 73] and Morvant et al. [60, 61] to derive well-performing PAC-Bayesian algorithms that aims at minimizing it.

Last but not least, PAC-Bayesian generalization bounds take into account the given prior distribution $P$ on $\mathcal{H}$ through the Kullback-Leibler divergence between the learned posterior distribution $Q$ and $P$:

$$\mathrm{KL}(Q\|P) = \mathop{\mathbb{E}}_{h\sim Q} \ln \frac{Q(h)}{P(h)}.$$

### 4.2.2 The usual PAC-Bayesian Theorem

In this section, we present the general PAC-Bayesian theorem in its probabilistic form. A key step in PAC-Bayesian proofs is the use of a change of measure inequality [58], based on the Donsker-Varadhan inequality [27]. The change of measure inequality is recalled in the following Lemma:

**Lemma 4.1.** *For any hypothesis space $\mathcal{H}$, for any prior $P$ and any posterior $Q$ on $\mathcal{H}$, and for any measurable function $\phi : \mathcal{H} \to \mathbb{R}$, we have*

$$\mathop{\mathbb{E}}_{h\sim Q} \phi(h) \leq \mathrm{KL}(Q\|P) + \ln\left(\mathop{\mathbb{E}}_{h\sim P} e^{\phi(h)}\right).$$

***Proof.*** Deferred to Appendix B.2 ■

Based on Lemma 4.1, the following theorem can be seen as a general PAC-Bayesian theorem which takes the form of a probabilistic bound (we recalled its one form in Theorem 2.3

in Chapter 2). Concretely, with a high probability over the random choice of the learning sample, it upper-bounds the "deviation" between the true and empirical risks of the Gibbs classifier uniformly for all distribution $Q$, according to a convex function $D : [0,1]\times[0,1]\to\mathbb{R}$.

**Theorem 4.1** (Germain et al. [34, 35])**.** *For any distribution $\mathcal{D}$ on $\mathcal{X}\times\mathcal{Y}$, for any set of voters $\mathcal{H}$, for any prior distribution $P$ on $\mathcal{H}$, for any $\delta\in(0,1]$, for any convex function $D : [0,1]\times[0,1]\to\mathbb{R}$, with a probability at least $1-\delta$ over the random choice of $S\sim(\mathcal{D})^m$, we have for all posterior distribution $Q$ on $\mathcal{H}$:*

$$D\big(R_S(G_Q), R_{\mathcal{D}}(G_Q)\big) \le \frac{1}{m}\left[\mathrm{KL}(Q\|P)+\ln\left(\frac{1}{\delta}\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{h\sim P} e^{m\,D(R_S(h),R_{\mathcal{D}}(h))}\right)\right],$$

*where $R_{\mathcal{D}}(h)$ and $R_S(h)$ are respectively the true and the empirical risks of individual voters, and $\mathrm{KL}(Q\|P) = \mathop{\mathbb{E}}_{h\sim Q}\ln\frac{Q(h)}{P(h)}$ is the Kullback-Leibler divergence between the learned posterior distribution $Q$ and $P$.*

***Proof.*** Deferred to Appendix B.3. ■

As stated by Germain et al. [34, 35], we can retrieve the classical versions of the PAC-Bayesian theorem [14, 58, 77] by selecting a well-suited deviation function $D$, and by upper-bounding $\mathbb{E}_S\mathbb{E}_h e^{m\,D(R_S(h),R_{\mathcal{D}}(h))}$. Note that, we recalled one of the classical version in in Theorem 2.3 in Chapter 2. In the next section, we provide a novel formulation of the PAC-Bayesian theorem expressed as an *expectation bound.*

## 4.3 A New PAC-Bayesian Theorem as an Expected Risk Bound

In this section, we introduce a new variation of the general PAC-Bayesian theorem of Germain et al. [34, 35]. While most of the PAC-Bayesian bounds are probabilistic bounds, we state here an *expected risk bound.* More specifically, Theorem 4.2 below is a tool to upper-bound $\mathbb{E}_{S\sim\mathcal{D}^m} R_{\mathcal{D}}(G_{Q_S})$—where $Q_S$ is the posterior distribution outputted after observing the learning sample $S$—while PAC-Bayes usually bounds $R_{\mathcal{D}}(G_Q)$ uniformly for all distribution $Q$, but with high probability over the random choice of learning sample $S\sim(\mathcal{D})^m$. Since by definition posterior distributions are data dependent, this different point of view on PAC-Bayesian analysis has the advantage to involve an expectation over all the possible learning samples (of a given size $m$) in bounds itself.

**Theorem 4.2.** *For any distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$, for any set of voters $\mathcal{H}$, for any prior distribution $P$ on $\mathcal{H}$, for any convex function $D : [0,1] \times [0,1] \to \mathbb{R}$, we have*

$$D\left(\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m} R_S(G_{Q_S}), \mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m} R_{\mathcal{D}}(G_{Q_S})\right) \le \frac{1}{m}\left[\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m} \mathrm{KL}(Q_S\|P) + \ln\left(\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{h\sim P} e^{m\,D(R_S(h),R_{\mathcal{D}}(h))}\right)\right],$$

*where $R_{\mathcal{D}}(h)$ and $R_S(h)$ are respectively the true and the empirical risks of individual classifiers.*

Similarly to Germain et al. [34, 35], by selecting a well-suited deviation function $D$ and by upper-bounding $\mathbb{E}_S \mathbb{E}_h e^{m\,D(R_S(h),R_{\mathcal{D}}(h))}$, we can prove the *expected bound* counterparts of the classical PAC-Bayesian theorems of [14, 58, 77]. The proof presented below borrows the straightforward proof technique of Bégin et al. [7]. Interestingly, this approach highlights that the expectation bounds are obtained simply by replacing the *Markov inequality* by the *Jensen inequality* (respectively Theorems A.1 and A.2, in Appendix).

***Proof of Theorem 4.2*** The last three inequalities below are obtained by applying Jensen's inequality on the convex function $D$ (Theorem A.2), the change of measure inequality (Lemma 4.1), and Jensen's inequality on the concave function ln.

$$\begin{aligned} mD\left(\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m} R_S(G_{Q_S}), \mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m} R_{\mathcal{D}}(G_{Q_S})\right) &= mD\left(\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{h\sim Q_S} R_S(h), \mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{h\sim Q_S} R_{\mathcal{D}}(h)\right)\\ &\le \mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{h\sim Q_S} mD(R_S(h), R_{\mathcal{D}}(h))\\ &\le \mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\left[\mathrm{KL}(Q_S\|P) + \ln\left(\mathop{\mathbb{E}}_{h\sim P} e^{m\,D(R_S(h),R_{\mathcal{D}}(h))}\right)\right]\\ &\le \mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m} \mathrm{KL}(Q_S\|P) + \ln\left(\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{h\sim P} e^{m\,D(R_S(h),R_{\mathcal{D}}(h))}\right). \end{aligned}$$

■

Since the $\mathcal{C}$-bound of Equation (4.2) involves the expected disagreement $d_{\mathcal{D}}(Q)$, we also derive below the expected bound that upper-bounds the deviation between $\mathbb{E}_{S\sim(\mathcal{D})^m}\, d_S(Q_S)$ and $\mathbb{E}_{S\sim(\mathcal{D})^m}\, d_{\mathcal{D}}(Q_S)$ under a convex function $D$. Theorem 4.3 can be seen as the *expectation* version of probabilistic bounds over $d_S(Q_S)$ proposed in [35, 51].

**Theorem 4.3.** *For any distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$, for any set of voters $\mathcal{H}$, for any prior distribution $P$ on $\mathcal{H}$, for any convex function $D : [0,1] \times [0,1] \to \mathbb{R}$, we have*

$$D\left(\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m} d_S(Q_S), \mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m} d_{\mathcal{D}}(Q_S)\right) \le \frac{2}{m}\left[\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m} \mathrm{KL}(Q_S\|P) + \ln\sqrt{\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{(h,h')\sim P^2} e^{mD(d_S(h,h'),d_{\mathcal{D}}(h,h'))}}\right],$$

*where $d_{\mathcal{D}}(h,h') = \mathbb{E}_{x\sim\mathcal{D}_{\mathcal{X}}}\, \mathbb{1}_{[h(x)\neq h'(x)]}$ is the disagreement between classifiers $h$ and $h'$ on the distribution $\mathcal{D}$, and $d_S(h,h')$ is its empirical counterpart.*

***Proof.*** The last three inequalities below are obtained by applying Jensen's inequality on the convex function $D$ (Theorem A.2), the change of measure inequality (Lemma 4.1), and Jensen's inequality on the concave function ln.

$$\begin{aligned}
mD\left(\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m} d_S(Q_S), \mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m} d_{\mathcal{D}}(Q_S)\right) &= mD\left(\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\ \mathop{\mathbb{E}}_{(h,h')\sim Q_S^2} d_S(h,h'), \mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\ \mathop{\mathbb{E}}_{(h,h')\sim Q_S^2} d_{\mathcal{D}}(h,h')\right)\\
&\le \mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{(h,h')\sim Q_S^2} mD\big(d_S(h,h'), d_{\mathcal{D}}(h,h')\big)\\
&\le \mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\left[\mathrm{KL}(Q_S^2\|P^2) + \ln\left(\mathop{\mathbb{E}}_{(h,h')\sim P^2} e^{mD(d_S(h,h'),d_{\mathcal{D}}(h,h'))}\right)\right]\\
&\le \mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m} \mathrm{KL}(Q_S^2\|P^2) + \ln \mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}\mathop{\mathbb{E}}_{(h,h')\sim P^2} e^{mD(d_S(h,h'),d_{\mathcal{D}}(h,h'))}.
\end{aligned}$$

Then, we use the fact that $\mathrm{KL}(Q_S^2\|P^2) = 2\,\mathrm{KL}(Q_S\|P)$ [35], which is detailed below:

$$\begin{aligned}
\mathrm{KL}(Q_S^2\|P^2) &= \mathop{\mathbb{E}}_{(h,h')\sim Q_S^2} \ln\frac{Q_S(h)Q_S(h')}{P(h)P(h')}\\
&= \mathop{\mathbb{E}}_{(h,h')\sim Q_S^2}\left[\ln\frac{Q_S(h)}{P(h)} + \ln\frac{Q_S(h')}{P(h')}\right]\\
&= 2\,\mathrm{KL}(Q_S\|P)
\end{aligned}$$

■

In the next section for sake of completeness, we derive the *expected bound* variations associated to the classical PAC-Bayesian theorems of [14, 58, 77], by selecting a well-suited deviation function $D$ and by upper-bounding $\mathbb{E}_S\mathbb{E}_h e^{mD(R_S(h),R_{\mathcal{D}}(h))}$.

## 4.4 Specialization of our Theorem to the Classical Approaches

In this section, we provide the specialization of our PAC-Bayesian theorem (Theorem 4.2) to the most popular PAC-Bayesian approaches [14, 58, 77].

### 4.4.1 Square Root Bound

We derive in Corollary 4.1 the specialization of Theorem 4.2 to the McAllester [58]'s point of view.

**Corollary 4.1.** *For any distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$, for any set of voters $\mathcal{H}$, for any prior distribution $P$ on $\mathcal{H}$, we have:*

$$\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m} R_{\mathcal{D}}(G_{Q_S}) \le \mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m} R_S(G_{Q_S}) + \sqrt{\frac{1}{2m}\left[\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m} \mathrm{KL}(Q_S\|P) + \ln 2\sqrt{m}\right]}.$$

***Proof.*** Deferred to Appendix B.4 ■

The generalization bound presented in Corollary 4.1 suggests that in order to minimize expectation of true risk of Gibbs classifier over all possible posterior distributions, one needs to control the trade-off between its empirical counterpart $\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m} R_S(G_{Q_S})$ and KL-divergence term $\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m} \mathrm{KL}(Q_S\|P)$. This bound is easy to interpret as it links the true risk and the empirical risk of the Gibbs classifier by a linear relation.

### 4.4.2 Parametrized Bound

To derive the generalization bound with the Catoni's [14] point of view, given a convex function $\mathcal{F}$ and a real number $C > 0$ we define the measure of deviation between the empirical disagreement/joint error and the true risk as $D(a,b) = \mathcal{F} - Ca$ [34, 35]. Then, we obtain following generalization bound.

**Corollary 4.2.** *For any distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$, for any set of voters $\mathcal{H}$, for any prior distribution $P$ on $\mathcal{H}$, for all $C > 0$, we have:*

$$\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m} R_{\mathcal{D}}(G_{Q_S}) \le \frac{1}{1-e^{-C}}\left(1 - \exp\left[-C \mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m} R_S(G_{Q_S}) - \frac{1}{m}\left[\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m} \mathrm{KL}(Q_S\|P)\right]\right]\right).$$

***Proof.*** Deferred to Appendix B.5 ■

The generalization bound given by Corollary 4.2 allows us to explicitly control the trade-off between empirical risk $\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m} R_S(G_{Q_S})$ and KL-divergence term $\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m} \mathrm{KL}(Q_S\|P)$ using the hyperparameter $C$. It appears to be a natural tool to design PAC-Bayesian algorithms. Moreover, following the similar approach as Germain et al. [36], we can derive a simplified form of above generalization bound using $1 - e^{-x} \le x$:

$$\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m} R_{\mathcal{D}}(G_{Q_S}) \le \frac{C}{1-e^{-C}}\left(\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m} R_S(G_{Q_S}) + \frac{1}{m\times C}\left[\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m} \mathrm{KL}(Q_S\|P)\right]\right).$$

### 4.4.3 Small kl Bound

If we make use, for function $D(a,b)$ between the empirical risk and the true risk, of the Kullback-Leibler divergence between two Bernoulli distributions with probability of success a and b, we can obtain a bound similar to[53, 77]. Concretely, we apply Theorem 3 with:

$$D(a,b) \leq \mathrm{kl}(a,b) = a \ln \frac{a}{b} + (1-a) \ln \frac{1-a}{1-b}.$$

**Corollary 4.3.** *For any distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$, for any set of voters $\mathcal{H}$, for any prior distribution $P$ on $\mathcal{H}$, we have:*

$$\mathrm{kl}\left( \mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m} R_{\mathcal{D}}(G_{Q_S}) \,||\, \mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m} R_S(G_{Q_S}) \right) \leq \frac{1}{m} \left[ \mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m} \mathrm{KL}(Q_S \| P) + \ln\, 2\sqrt{m} \right].$$

***Proof.*** Deferred to Appendix B.6 ■

The generalization bound given by Corollary 4.3 controls the trade-off between the KL divergence term and the empirical risk using kl(·||·). It is difficult to interpret due to kl-divergence term between true risk and empirical risk of Gibbs classifier. In order to upper bound the true risk, one needs to solve the following problem:

$$\begin{aligned} &\max \quad b \\ &s.t. \quad \mathrm{kl}\left( b \,||\, \mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m} R_S(G_{Q_S}) \right) = \frac{1}{m} \left[ \mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m} \mathrm{KL}(Q_S \| P) + \ln\, 2\sqrt{m} \right] \text{ and } 0 \leq b \leq 1. \end{aligned}$$

## 4.5 Conclusion

In this chapter, we propose a new PAC-Bayesian theorem that is not a probabilistic bound as usual. Indeed, it is expressed as an expectation over the posterior distributions that we can learn for a given learning sample size, while the usual PAC-Bayesian theorem stands uniformly for all the possible posterior distributions but with high probability over the random choice of the learning sample. Since by definition posterior distributions are data dependent, this different point of view on PAC-Bayesian analysis has the advantage to involve an expectation over all the possible learning samples (of a given size $m$) in bounds itself. Moreover, we specialize our PAC-Bayesian theorem to three most popular PAC-Bayesian approaches. In the next chapter, we provide an extension of this bound to mutlview learning where we have multiple representations or views of the input data.

CHAPTER 5

# PAC-Bayesian Analysis of Multiview Learning

In this chapter, we study a two-level multiview learning learning with more than two views under the PAC-Bayesian framework. This approach, sometimes referred as late fusion, consists in learning sequentially multiple view-specific classifiers at the first level, and then combining these view-specific classifiers at the second level. Our main theoretical result is a generalization bound on the risk of the majority vote which exhibits a term of diversity in the predictions of the view-specific classifiers. From this result it comes out that controlling the trade-off between diversity and accuracy is a key element for multiview learning, which complements other results (Theorems 3.1 and 3.2 in Chapter 3) in multiview learning. This work has been done in collaboration with Dr. Pascal Germain from INRIA, Lille, France. It has been accepted at CAp, 2016 [42] and CAp, 2017 [40]; published in the proceedings of ECML-PKDD, 2017 [41]; and submitted to Neurocomputing Journal.

## 5.1 Introduction

We make use of the PAC-Bayesian framework [57] of Chapter 4 to study the issue of learning a binary classification model while taking into account different information sources. This issue is referred as multiview learning [4, 80] and is described in Chapter 3. Here, our goal is to propose a theoretically grounded criteria to "correctly" combine different views while taking into account the diversity between the views (see Section 3.4 for more details). With this in mind we propose to study multiview learning through the PAC-Bayesian framework that allows to derive generalization bounds for models that are expressed as a combination over a set of classifiers or views (in our case). In this chapter, we extend the PAC-Bayesian theory (both the probabilistic and the expected risk generalization bounds) to multiview with more than two views. Concretely, given a set of view-specific voters, we define a hierarchy of posterior and prior distributions over the views, such that *(i)* for each view $v$, we consider a prior $P_v$ distribution and learn a posterior $Q_v$ distribution over each view-specific voters' set, and *(ii)* we consider a prior $\pi$ and learn a posterior $\rho$ distribution over the set of views (see Figure 5.1), respectively called hyper-prior and hyper-posterior[1]. In this way, our proposed approach encompasses the one of Amini et al.[2] (recalled in Theorem 3.1) that considered uniform distribution to combine the view-specific classifiers' predictions. Moreover, compared to the PAC-Bayesian work of Sun et al. [81] (recalled in Theorem 3.2), we are interested here to the more general and natural case of multiview learning with more than two views.

Our theoretical study also includes a notion of disagreement between all the voters, allowing to take into account a notion of diversity between them which is known as a key element in multiview learning [2, 15, 50, 56].

In Section 5.2, we present the notations and setting for our two-level hierarchical multiview learning followed by the instantiation of the PAC-Bayesian generalization bounds to the two-level multiview approach (in Section 5.3). We present the generalization bound for the multiview $\mathcal{C}$-bound in Section 5.5. Before concluding in Section 5.7, we discuss the relation between our analysis and previous works in Section 5.6.

## 5.2 Notations and Setting

We consider binary classification problems where the multiview observations $\mathbf{x} = (x^1, \dots, x^V)$ belong to a multiview input set $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_V$, where $V \geq 2$ is the number of views of not-necessarily the same dimension. We denote $\mathcal{V}$ the set of the $V$ views. In binary classification, we assume that examples are pairs $(\mathbf{x}, y)$, with $y \in \mathcal{Y} = \{-1, +1\}$, drawn according to an unknown distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$. To model the two-level multiview approach, we follow the next setting. For each view $v \in \mathcal{V}$, we consider a view-specific set $\mathcal{H}_v$ of voters $h_v : \mathcal{X}_v \to \mathcal{Y}$, and a prior distribution $P_v$ on $\mathcal{H}_v$. Given a *hyper-prior* distribution $\pi$ over the views $\mathcal{V}$, and a multiview learning sample $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^m \sim (\mathcal{D})^m$, our PAC-Bayesian learner objective

[1] Our notion of hyper-prior and hyper-posterior distributions is different than the one proposed for lifelong learning [68], where they basically consider hyper-prior and hyper-posterior over the set of possible priors: The prior distribution $P$ over the voters' set is viewed as a random variable, which is not the case in this thesis.

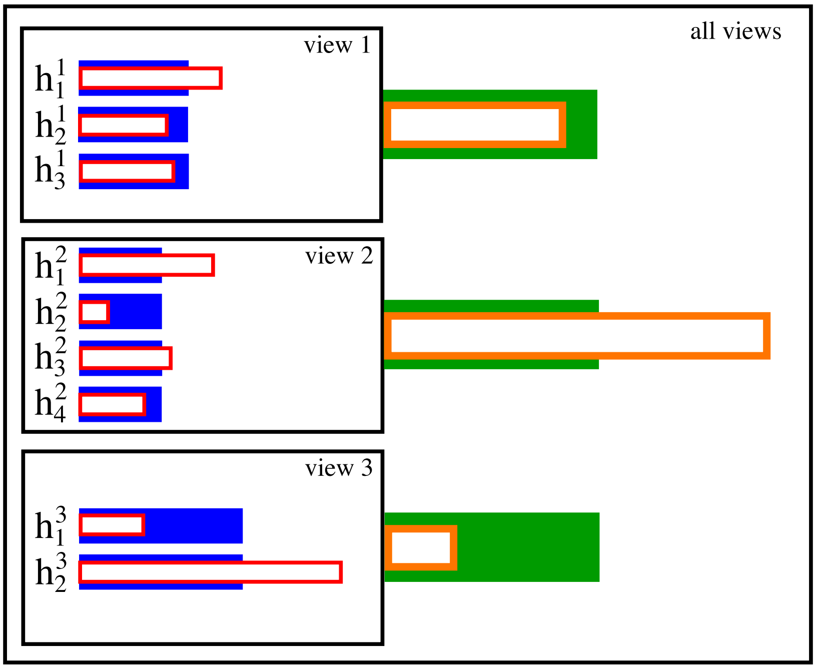


Figure 5.1: Example of the multiview distributions hierarchy with 3 views. For all views $v \in \{1,2,3\}$, we have a set of $n_v$ voters $\mathcal{H}_v = \{h_v^1,\dots,h_v^{n_v}\}$ on which we consider a prior $P_v$ view-specific distribution (in blue). A hyper-prior $\pi$ distribution (in green) over the set of 3 views is also considered. The objective is to learn a set of posterior $\{Q_v\}_{v=1}^3$ (in red) view-specific distributions and a hyper-posterior $\rho$ distribution (in orange) leading to a good model. The length of a rectangle represents the weight (or probability) assigned to a voter or a view.

is twofold: *(i)* finding a posterior distribution $Q_v$ over $\mathcal{H}_v$ for all views $v \in \mathcal{V}$; *(ii)* finding a *hyper-posterior* distribution $\rho$ on the set of views $\mathcal{V}$. This hierarchy of distributions is illustrated by Figure 5.1. The learned distributions express a multiview weighted majority vote $B_\rho^{\mathrm{MV}}$ defined as

$$B_\rho^{\mathrm{MV}}(\mathbf{x}) = \operatorname{sign}\left[\mathop{\mathbb{E}}_{v\sim\rho}\ \mathop{\mathbb{E}}_{h_v\sim Q_v} h(x^v)\right]. \tag{5.1}$$

Thus, the learner aims at constructing the posterior and hyper-posterior distributions that minimize the true risk $R_{\mathcal{D}}(B_\rho^{\mathrm{MV}})$ of the multiview weighted majority vote:

$$R_{\mathcal{D}}(B_\rho^{\mathrm{MV}}) = \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \mathbb{1}_{[B_\rho^{\mathrm{MV}}(\mathbf{x})\neq y]}.$$

As pointed out in Section 4.2, the PAC-Bayesian approach deals with the risk of the stochastic Gibbs classifier $G_\rho^{\mathrm{MV}}$ defined as follows in our multiview setting, and that can be rewritten in terms of *expected disagreement* $d_{\mathcal{D}}^{\mathrm{MV}}(\rho)$ and *expected joint error* $e_{\mathcal{D}}^{\mathrm{MV}}(\rho)$:

$$\begin{aligned}
R_{\mathcal{D}}(G_\rho^{\mathrm{MV}}) &= \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \mathop{\mathbb{E}}_{v\sim\rho} \mathop{\mathbb{E}}_{h_v\sim Q_v} \mathbb{1}_{[h_v(x^v)\neq y]} && (5.2)\\
&= \tfrac{1}{2}\, d_{\mathcal{D}}^{\mathrm{MV}}(\rho) + e_{\mathcal{D}}^{\mathrm{MV}}(\rho)\,, && (5.3)\\
\text{where}\quad d_{\mathcal{D}}^{\mathrm{MV}}(\rho) &= \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{X}}} \mathop{\mathbb{E}}_{v\sim\rho} \mathop{\mathbb{E}}_{v'\sim\rho} \mathop{\mathbb{E}}_{h_v\sim Q_v} \mathop{\mathbb{E}}_{h'_v\sim Q_{v'}} \mathbb{1}_{[h_v(x^v)\neq h'_v(x^{v'})]},\\
\text{and}\quad e_{\mathcal{D}}^{\mathrm{MV}}(\rho) &= \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \mathop{\mathbb{E}}_{v\sim\rho} \mathop{\mathbb{E}}_{v'\sim\rho} \mathop{\mathbb{E}}_{h_v\sim Q_v} \mathop{\mathbb{E}}_{h'_v\sim Q_{v'}} \mathbb{1}_{[h_v(x^v)\neq y]}\mathbb{1}_{[h'_v(x^{v'})\neq y]}.
\end{aligned}$$

Obviously, the empirical counterpart of the Gibbs classifier's risk $R_{\mathcal{D}}(G_\rho^{\mathrm{MV}})$ is

$$\begin{aligned}
R_S(G_\rho^{\mathrm{MV}}) &= \frac{1}{m}\sum_{i=1}^{m} \mathop{\mathbb{E}}_{v\sim\rho} \mathop{\mathbb{E}}_{h_v\sim Q_v} \mathbb{1}_{[h_v(x_i^v)\neq y_i]}\\
&= \frac{1}{2} d_S^{\mathrm{MV}}(\rho) + e_S^{\mathrm{MV}}(\rho)\,,
\end{aligned}$$

where $d_S^{\mathrm{MV}}(\rho)$ and $e_S^{\mathrm{MV}}(\rho)$ are respectively the empirical estimations of $d_{\mathcal{D}}^{\mathrm{MV}}(\rho)$ and $e_{\mathcal{D}}^{\mathrm{MV}}(\rho)$ on the learning sample $S$. As in the single-view PAC-Bayesian setting, the multiview weighted majority vote $B_\rho^{\mathrm{MV}}$ is closely related to this stochastic multiview Gibbs classifier $G_\rho^{\mathrm{MV}}$, and a generalization bound for $G_\rho^{\mathrm{MV}}$ gives rise to a generalization bound for $B_\rho^{\mathrm{MV}}$. Indeed, it is easy to show that $R_{\mathcal{D}}(B_\rho^{\mathrm{MV}}) \leq 2R_{\mathcal{D}}(G_\rho^{\mathrm{MV}})$, meaning that an upper bound over $R_{\mathcal{D}}(G_\rho^{\mathrm{MV}})$ gives an upper bound for the majority vote. Moreover the C-Bound of Equation (4.2) can be extended to our multiview setting by Lemma 5.1 below. Equation (5.4) is a straightforward generalization of the single-view $\mathcal{C}$-bound of Equation (4.2). Afterward, Equation (5.5) is a looser version obtained by rewriting $R_{\mathcal{D}}(G_\rho^{\mathrm{MV}})$ as the $\rho$-average of the risk associated to each view, and lower-bounding $d_{\mathcal{D}}^{\mathrm{MV}}(\rho)$ by the $\rho$-average of the disagreement associated to each view.

**Lemma 5.1.** *Let $V \geq 2$ be the number of views. For all posterior $\{Q_v\}_{v=1}^V$ distributions over $\{\mathcal{H}_v\}_{v=1}^V$ and hyper-posterior $\rho$ distributions on views $\mathcal{V}$, if $R_{\mathcal{D}}(G_\rho^{\mathrm{MV}}) < \frac{1}{2}$, then we have*

$$\begin{aligned}
R_{\mathcal{D}}(B_\rho^{\mathrm{MV}}) &\leq 1 - \frac{\left(1-2R_{\mathcal{D}}(G_\rho^{\mathrm{MV}})\right)^2}{1-2d_{\mathcal{D}}^{\mathrm{MV}}(\rho)} && (5.4)\\
&\leq 1 - \frac{\Big(1-2\,\mathbb{E}_{v\sim\rho}\, R_{\mathcal{D}}(G_{Q_v})\Big)^2}{1-2\,\mathbb{E}_{v\sim\rho}\, d_{\mathcal{D}}(Q_v)}\,, && (5.5)
\end{aligned}$$

*where $R_{\mathcal{D}}(G_{Q_v})$ and $d_{\mathcal{D}}(Q_v)$ are respectively the true view-specific Gibbs risk and the expected disagreement defined as*

$$\begin{aligned}
R_{\mathcal{D}}(G_{Q_v}) &= \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \mathop{\mathbb{E}}_{h_v\sim Q_v} \mathbb{1}_{[h_v(x^v)\neq y]}\,,\\
\textit{and}\quad d_{\mathcal{D}}(Q_v) &= \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{X}}} \mathop{\mathbb{E}}_{h_v\sim Q_v} \mathop{\mathbb{E}}_{h'_v\sim Q_v} \mathbb{1}_{[h_v(x^v)\neq h'_v(x^v)]}.
\end{aligned}$$

***Proof.*** Proof of Equation (5.4) is given in Appendix C.1. To prove Equation (5.5), we first notice that in the binary setting where $y \in \{-1,1\}$ and $h_v : \mathcal{X} \to \{-1,1\}$, we have

$$\mathbb{1}_{[h_v(x^v)\neq y]} = \frac{1}{2}(1 - y\, h_v(x^v))$$

and

$$\begin{aligned} R_{\mathcal{D}}(G_\rho^{\text{MV}}) &= \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \mathop{\mathbb{E}}_{v\sim\rho} \mathop{\mathbb{E}}_{h_v\sim Q_v} \mathbb{1}_{[h_v(x^v)\neq y]} \\ &= \frac{1}{2}\Big(1 - \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \mathop{\mathbb{E}}_{v\sim\rho} \mathop{\mathbb{E}}_{h_v\sim Q_v} y\, h_v(x^v)\Big) \\ &= \mathop{\mathbb{E}}_{v\sim\rho} R_{\mathcal{D}}(G_{Q^v})\,. \end{aligned}$$

Moreover, we have

$$\begin{aligned} d_{\mathcal{D}}^{\text{MV}}(\rho) &= \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{X}}} \mathop{\mathbb{E}}_{v\sim\rho} \mathop{\mathbb{E}}_{v'\sim\rho} \mathop{\mathbb{E}}_{h_v\sim Q_v} \mathop{\mathbb{E}}_{h'_v\sim Q_{v'}} \mathbb{1}_{[h_v(x^v)\neq h'_v(x^{v'})]} \\ &= \frac{1}{2}\Big(1 - \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{X}}} \mathop{\mathbb{E}}_{v\sim\rho} \mathop{\mathbb{E}}_{v'\sim\rho} \mathop{\mathbb{E}}_{h_v\sim Q_v} \mathop{\mathbb{E}}_{h_v\sim Q_{v'}} h_v(x^v) \times h'_v(x^{v'})\Big) \\ &= \frac{1}{2}\Big(1 - \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{X}}} \Big[\mathop{\mathbb{E}}_{v\sim\rho} \mathop{\mathbb{E}}_{h_v\sim Q_v} h_v(x^v)\Big]^2\Big). \end{aligned}$$

From Jensen's inequality (Theorem A.2, in Appendix) it comes

$$\begin{aligned} d_{\mathcal{D}}^{\text{MV}}(\rho) &\geq \frac{1}{2}\Big(1 - \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{X}}} \mathop{\mathbb{E}}_{v\sim\rho} \Big[\mathop{\mathbb{E}}_{h_v\sim Q_v} h_v(x^v)\Big]^2\Big) \\ &= \mathop{\mathbb{E}}_{v\sim\rho}\Big[\frac{1}{2}\Big(1 - \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{X}}} \Big[\mathop{\mathbb{E}}_{h_v\sim Q_v} h_v(x^v)\Big]^2\Big)\Big] \\ &= \mathop{\mathbb{E}}_{v\sim\rho} d_{\mathcal{D}}(Q_v)\,. \end{aligned}$$

By replacing $R_{\mathcal{D}}(G_\rho^{\text{MV}})$ and $d_{\mathcal{D}}^{\text{MV}}(\rho)$ in Equation (5.4), we obtain

$$1 - \frac{\big(1 - 2R_{\mathcal{D}}(G_\rho^{\text{MV}})\big)^2}{1 - 2d_{\mathcal{D}}^{\text{MV}}(\rho)} \leq 1 - \frac{\Big(1 - 2\mathbb{E}_{v\sim\rho} R_{\mathcal{D}}(G_{Q^v})\Big)^2}{1 - 2\mathbb{E}_{v\sim\rho} d_{\mathcal{D}}(Q_v)}\,.$$

■

Similarly than for the mono-view setting, Equations (5.3) and (5.4) suggest that a good trade-off between the risk of the Gibbs classifier $G_\rho^{\text{MV}}$ and the disagreement $d_{\mathcal{D}}^{\text{MV}}(\rho)$ between pairs of voters will lead to a well-performing majority vote. Equation (5.5) exhibits the role of diversity among the views thanks to the disagreement's expectation over the views $\mathbb{E}_{v\sim\rho}\, d_{\mathcal{D}}(Q_v)$.

## 5.3 General Multiview PAC-Bayesian Theorem

Now we state our general PAC-Bayesian theorem suitable for the above multiview learning setting with a two-level hierarchy of distributions over views (or voters). As pointed out in Chapter 4, ap key step in PAC-Bayesian proofs is the use of a *change of measure inequality* [58], based on the Donsker-Varadhan inequality [27]. Lemma 5.2 below extends this tool to our multiview setting.

**Lemma 5.2.** *For any set of prior distributions $\{P_v\}_{v=1}^V$ and any set of posterior distributions $\{Q_v\}_{v=1}^V$ over $\{\mathcal{H}_v\}_{v=1}^V$, for any hyper-prior distribution $\pi$ on views $\mathcal{V}$ and hyper-posterior distribution $\rho$ on $\mathcal{V}$, and for any measurable function $\phi : \mathcal{H}_v \to \mathbb{R}$, we have*

$$\mathop{\mathbb{E}}_{v\sim\rho}\mathop{\mathbb{E}}_{h_v\sim Q_v}\phi(h_v) \le \mathop{\mathbb{E}}_{v\sim\rho}\mathrm{KL}(Q_v\|P_v) + \mathrm{KL}(\rho\|\pi) + \ln\left(\mathop{\mathbb{E}}_{v\sim\pi}\mathop{\mathbb{E}}_{h_v\sim P_v} e^{\phi(h_v)}\right).$$

***Proof.*** We have

$$\begin{aligned}
\mathop{\mathbb{E}}_{v\sim\rho}\mathop{\mathbb{E}}_{h_v\sim Q_v}\phi(h_v) &= \mathop{\mathbb{E}}_{v\sim\rho}\mathop{\mathbb{E}}_{h_v\sim Q_v}\ln e^{\phi(h_v)}\\
&= \mathop{\mathbb{E}}_{v\sim\rho}\mathop{\mathbb{E}}_{h_v\sim Q_v}\ln\left(\frac{Q_v(h_v)}{P_v(h_v)}\frac{P_v(h_v)}{Q_v(h_v)}e^{\phi(h_v)}\right)\\
&= \mathop{\mathbb{E}}_{v\sim\rho}\left[\mathop{\mathbb{E}}_{h_v\sim Q_v}\ln\left(\frac{Q_v(h_v)}{P_v(h_v)}\right) + \mathop{\mathbb{E}}_{h_v\sim Q_v}\ln\left(\frac{P_v(h_v)}{Q_v(h_v)}e^{\phi(h_v)}\right)\right].
\end{aligned}$$

According to the definition of Kullback-Leibler divergence, we have

$$\mathop{\mathbb{E}}_{v\sim\rho}\mathop{\mathbb{E}}_{h_v\sim Q_v}\phi(h_v) = \mathop{\mathbb{E}}_{v\sim\rho}\left[\mathrm{KL}(Q_v\|P_v) + \mathop{\mathbb{E}}_{h_v\sim Q_v}\ln\left(\frac{P_v(h_v)}{Q_v(h_v)}e^{\phi(h_v)}\right)\right].$$

By applying Jensen's inequality (Theorem A.2, in Appendix) on the concave function ln, we have

$$\begin{aligned}
\mathop{\mathbb{E}}_{v\sim\rho}\mathop{\mathbb{E}}_{h_v\sim Q_v}\phi(h_v) &\le \mathop{\mathbb{E}}_{v\sim\rho}\left[\mathrm{KL}(Q_v\|P_v) + \ln\left(\mathop{\mathbb{E}}_{h_v\sim P_v} e^{\phi(h_v)}\right)\right]\\
&= \mathop{\mathbb{E}}_{v\sim\rho}\mathrm{KL}(Q_v\|P_v) + \mathop{\mathbb{E}}_{v\sim\rho}\ln\left(\frac{\rho(v)}{\pi(v)}\frac{\pi(v)}{\rho(v)}\mathop{\mathbb{E}}_{h_v\sim P_v} e^{\phi(h_v)}\right)\\
&= \mathop{\mathbb{E}}_{v\sim\rho}\mathrm{KL}(Q_v\|P_v) + \mathrm{KL}(\rho\|\pi) + \mathop{\mathbb{E}}_{v\sim\rho}\ln\left(\frac{\pi(v)}{\rho(v)}\mathop{\mathbb{E}}_{h_v\sim P_v} e^{\phi(h_v)}\right).
\end{aligned}$$

Finally, we apply again the Jensen inequality (Theorem A.2) on ln to obtain the lemma. ■

Based on Lemma 5.2, the following theorems can be seen as a generalization of Theorem 4.1 and Theorem 4.2 to multiview respectively. Note that we still rely on a general convex function $D\colon[0,1]\times[0,1]\to\mathbb{R}$, that measures the "deviation" between the empirical disagreement/joint error and the true risk of the Gibbs classifier. Note that, in both of following theorems, we split empirical Gibbs risk into empirical expected disagreement and expected joint error using Equation (4.3). This is done in order to highlight the trade-off between disagreement (or diversity between views) and joint error which is important for multiview learning (as discussed in Chapter 3).

**Theorem 5.1** (Probabilistic bound for Multiview Learning)**.** *Let $V\geq 2$ be the number of views. For any distribution $\mathcal{D}$ on $\mathcal{X}\times\mathcal{Y}$, for any set of prior distributions $\{P_v\}_{v=1}^V$ over $\{\mathcal{H}_v\}_{v=1}^V$, for any hyper-prior distributions $\pi$ over $\mathcal{V}$, for any convex function $D\colon[0,1]\times[0,1]\to\mathbb{R}$, for any $\delta\in(0,1]$, with a probability at least $1-\delta$ over the random choice of $S\sim(\mathcal{D})^m$, for all posterior $\{Q_v\}_{v=1}^V$ over $\{\mathcal{H}_v\}_{v=1}^V$ and hyper-posterior $\rho$ over $\mathcal{V}$ distributions, we have:*

$$D\Big(\tfrac{1}{2}d_S^{\mathrm{MV}}(\rho_S)+e_S^{\mathrm{MV}}(\rho_S),R_{\mathcal{D}}(G_\rho^{\mathrm{MV}})\Big)\leq$$
$$\frac{1}{m}\Big[\mathop{\mathbb{E}}_{v\sim\rho}\mathrm{KL}(Q_v\|P_v)+\mathrm{KL}(\rho\|\pi)+\ln\Big(\frac{1}{\delta}\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{v\sim\pi}\mathop{\mathbb{E}}_{h\sim P_v}e^{mD(R_S(h),R_{\mathcal{D}}(h))}\Big)\Big].$$

***Proof.*** Deferred to Appendix C.2.; ■

**Theorem 5.2** (Expected Risk Bound for Multiview Learning)**.** *Let $V\geq 2$ be the number of views. For any distribution $\mathcal{D}$ on $\mathcal{X}\times\mathcal{Y}$, for any set of prior distributions $\{P_v\}_{v=1}^V$, for any hyper-prior distribution $\pi$ over $\mathcal{V}$, for any convex function $D\colon[0,1]\times[0,1]\to\mathbb{R}$, we have*

$$D\Big(\tfrac{1}{2}\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}d_S^{\mathrm{MV}}(\rho_S)+\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}e_S^{\mathrm{MV}}(\rho_S),\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}R_{\mathcal{D}}(G_{\rho_S}^{\mathrm{MV}})\Big)\leq\frac{1}{m}\Big[\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{v\sim\rho_S}\mathrm{KL}(Q_{v,S}\|P_v)$$
$$+\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathrm{KL}(\rho_S\|\pi)+\ln\Big(\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{v\sim\pi}\mathop{\mathbb{E}}_{h_v\sim P_v}e^{mD(R_S(h_v),R_{\mathcal{D}}(h_v))}\Big)\Big].$$

***Proof.*** We follow the same steps as in Theorem 4.2 proof.

$$\begin{aligned}
&mD\Big(\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}R_S(G_{\rho_S}^{\mathrm{MV}}),\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}R_{\mathcal{D}}(G_{\rho_S}^{\mathrm{MV}})\Big)\\
&=mD\Big(\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{v\sim\rho_S}\mathop{\mathbb{E}}_{h_v\sim Q_{v,S}}R_S(h_v),\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{v\sim\rho_S}\mathop{\mathbb{E}}_{h_v\sim Q_{v,S}}R_{\mathcal{D}}(h_v)\Big)\\
&\leq\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{v\sim\rho_S}\mathop{\mathbb{E}}_{h_v\sim Q_{v,S}}mD\left(R_S(h_v),R_{\mathcal{D}}(h_v)\right)\\
&\leq\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\Big[\mathop{\mathbb{E}}_{v\sim\rho_S}\mathrm{KL}(Q_{v,S}\|P_v)+\mathrm{KL}(\rho_S\|\pi)+\ln\Big(\mathop{\mathbb{E}}_{v\sim\pi}\mathop{\mathbb{E}}_{h_v\sim P_v}e^{mD(R_S(h_v),R_{\mathcal{D}}(h_v))}\Big)\Big],
\end{aligned}$$

where the last inequality is obtained using Lemma 5.2. After distributing the expectation of $S\sim(\mathcal{D})^m$, the final statement follows from Jensen's inequality (Theorem A.2)

$$\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\ln\Big(\mathop{\mathbb{E}}_{v\sim\pi}\mathop{\mathbb{E}}_{h_v\sim P_v}e^{mD(R_S(h_v),R_{\mathcal{D}}(h_v))}\Big)\leq\ln\Big(\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{v\sim\pi}\mathop{\mathbb{E}}_{h_v\sim P_v}e^{mD(R_S(h_v),R_{\mathcal{D}}(h_v))}\Big),$$

and from Equation (4.3): $R_S(G_{\rho_S}^{\mathrm{MV}})=\frac{1}{2}d_S^{\mathrm{MV}}(\rho_S)+e_S^{\mathrm{MV}}(\rho_S)$. ■

It is interesting to compare this generalization bound to Theorem 4.1 (and Theorem 4.2). The main difference relies on the introduction of view-specific prior and posterior distributions, which mainly leads to an additional term $\mathbb{E}_{v\sim\rho}\mathrm{KL}(Q_v\|P_v)$ $\big($ and $\mathbb{E}_{S\sim(\mathcal{D})^m}\mathbb{E}_{v\sim\rho_S}\mathrm{KL}(Q_{v,S}\|P_v)\big)$, expressed as the expectation of the view-specific Kullback-Leibler divergence term over the views $\mathcal{V}$ according to the hyper-posterior distribution $\rho$. This additional term captures the deviation between the view-specific posterior and prior distributions over all the views. We also introduce the empirical disagreement allowing us to directly highlight the presence of the diversity between voters and between views. As Theorem 4.2 (and Theorem 4.2), Theorem 5.1 (and Theorem 5.2) provides a tool to derive PAC-Bayesian generalization bounds for two-level multiview supervised learning setting. Indeed, by making use of the same trick as Germain et al. [34, 35], generalization bounds can be derived from Theorem 5.1 (and Theorem 5.2) by choosing a suitable convex function $D$ and upper-bounding $\mathbb{E}_S\mathbb{E}_v\mathbb{E}_{h_v} e^{mD(R_S(h_v),R_{\mathcal{D}}(h_v))}$. We provide the specialization to the three most popular PAC-Bayesian approaches [14, 53, 58, 77] in the next section.

Since the multiview $\mathcal{C}$-bound of Equation (5.4) involves the expected disagreement $d_{\mathcal{D}}^{\mathrm{MV}}(\rho)$, we also derive below the probabilistic and expected bounds that upper-bounds the deviation between $d_S^{\mathrm{MV}}(\rho)$ and $d_{\mathcal{D}}^{\mathrm{MV}}(\rho)$ under a convex function $D$. Theorem 5.3 can be seen as the of probabilistic and expected bounds over $d_S^{\mathrm{MV}}(\rho)$ proposed by [35, 51].

**Theorem 5.3.** *Let $V \geq 2$ be the number of views. For any distribution $\mathcal{D}$ on $\mathcal{X}\times\mathcal{Y}$, for any set of prior distributions $\{P_v\}_{v=1}^V$, for any hyper-prior distribution $\pi$ over $\mathcal{V}$, for any convex function $D : [0,1]\times[0,1]\to\mathbb{R}$, we have*

$$\Pr_{S\sim(\mathcal{D})^m}\Big(D\Big(d_S^{\mathrm{MV}}(\rho), d_{\mathcal{D}}^{\mathrm{MV}}(\rho)\Big)$$
$$\leq \frac{2}{m}\Big[\mathbb{E}_{v\sim\rho}\mathrm{KL}(Q_v\|P_v) + \ \mathrm{KL}(\rho\|\pi) + \ln\sqrt{\mathbb{E}_{S\sim(\mathcal{D})^m}\ \mathbb{E}_{(h_v,h'_v)\sim P^2} e^{mD\left(d_S(h_v,h'_v),d_{\mathcal{D}}(h_v,h'_v)\right)}}\Big]\Big) \geq 1-\delta\,,$$

*and*

$$D\Big(\mathbb{E}_{S\sim(\mathcal{D})^m} d_S^{\mathrm{MV}}(\rho_S), \mathbb{E}_{S\sim(\mathcal{D})^m} d_{\mathcal{D}}^{\mathrm{MV}}(\rho_S)\Big)$$
$$\leq \frac{2}{m}\Big[\mathbb{E}_{S\sim(\mathcal{D})^m}\mathbb{E}_{v\sim\rho_S}\mathrm{KL}(Q_{v,S}\|P_v) + \mathbb{E}_{S\sim(\mathcal{D})^m}\mathrm{KL}(\rho_S\|\pi) + \ln\sqrt{\mathbb{E}_{S\sim(\mathcal{D})^m}\ \mathbb{E}_{(h_v,h'_v)\sim P^2} e^{mD\left(d_S(h_v,h'_v),d_{\mathcal{D}}(h_v,h'_v)\right)}}\Big].$$

***Proof.*** The result is obtained straightforwardly by following the proof steps of Theorem 5.1 and Theorem 5.2 respectively, using the disagreement instead of the Gibbs risk. Then, similarly at what we have done to obtain Theorem 4.3, we substitute $\mathrm{KL}(Q_{v,S}^2\|P_v^2)$ by $2\,\mathrm{KL}(Q_{v,S}\|P_v)$, and $\mathrm{KL}(\rho_S^2\|\pi^2)$ by $2\,\mathrm{KL}(\rho_S\|\pi)$. ■

## 5.4 Specialization of our Theorem to the Classical Approaches

In this section, we provide specialization of our multiview theorem to the most popular PAC-Bayesian approaches [14, 53, 57, 77]. To do so, we follow the same principles as Germain et al. [34, 35] as recalled in Chapter 4.

### 5.4.1 Square Root Bound

We derive here the specialization of our multiview PAC-Bayesian theorem to the McAllester[58]'s point of view.

**Corollary 5.1.** *Let $V \geq 2$ be the number of views. For any distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$, for any set of prior distributions $\{P_v\}_{v=1}^V$ over $\{\mathcal{H}\}_{v=1}^V$, for any hyper-prior distribution $\pi$ over $\mathcal{V}$, we have*

$$\Pr_{S \sim (\mathcal{D})^m}\Bigg( R_{\mathcal{D}}(G_\rho^{\mathrm{MV}}) \leq \frac{1}{2} d_S^{\mathrm{MV}}(\rho) + e_S^{\mathrm{MV}}(\rho) + \sqrt{\frac{\mathop{\mathbb{E}}\limits_{v \sim \rho_S} \mathrm{KL}(Q_{v,S} \| P_v) + \mathop{\mathbb{E}}\limits_{S \sim (\mathcal{D})^m} \mathrm{KL}(\rho_S \| \pi) + \ln \frac{2\sqrt{m}}{\delta}}{2m}} \Bigg) \geq 1 - \delta$$

*and*

$$\mathop{\mathbb{E}}\limits_{S \sim (\mathcal{D})^m} R_{\mathcal{D}}(G_{\rho_S}^{\mathrm{MV}}) \leq \frac{1}{2} \mathop{\mathbb{E}}\limits_{S \sim (\mathcal{D})^m} d_S^{\mathrm{MV}}(\rho_S) + \mathop{\mathbb{E}}\limits_{S \sim (\mathcal{D})^m} e_S^{\mathrm{MV}}(\rho_S) + \sqrt{\frac{\mathop{\mathbb{E}}\limits_{S \sim (\mathcal{D})^m} \mathop{\mathbb{E}}\limits_{v \sim \rho_S} \mathrm{KL}(Q_{v,S} \| P_v) + \mathop{\mathbb{E}}\limits_{S \sim (\mathcal{D})^m} \mathrm{KL}(\rho_S \| \pi) + \ln 2\sqrt{m}}{2m}}.$$

***Proof.*** Deferred to Appendix C.3. ■

### 5.4.2 Parametrized Bound

To derive a generalization bound with the Catoni's [14] point of view—given a convex function $\mathcal{F}$ and a real number $C > 0$— define the measure of deviation between the empirical disagreement/joint error and the true risk as $D(a, b) = \mathcal{F}(b) - C\,a$ [34, 35]. We obtain the following generalization bound.

**Corollary 5.2.** *Let $V \geq 2$ be the number of views. For any distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$, for any set of prior distributions $\{P_v\}_{v=1}^V$ over $\{\mathcal{H}\}_{v=1}^V$, for any hyper-prior distributions $\pi$ over $\mathcal{V}$, for all*

$C>0$*, we have:*

$$\Pr_{S\sim(\mathcal{D})^m}\Bigg(R_{\mathcal{D}}(G_\rho^{\mathrm{MV}}) \;\le\; \frac{1}{1-e^{-C}}\bigg(1-\exp\bigg[-\bigg[C\Big(\tfrac{1}{2}d_S^{\mathrm{MV}}(\rho)+e_S^{\mathrm{MV}}(\rho)\Big)+ \frac{1}{m}\Big[\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{v\sim\rho_S}\mathrm{KL}(Q_{v,S}\|P_v)+\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathrm{KL}(\rho_S\|\pi)+\ln\tfrac{1}{\delta}\Big]\bigg]\bigg]\bigg)\Bigg)\ge 1-\delta$$

*and*

$$\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}R_{\mathcal{D}}(G_{\rho_S}^{\mathrm{MV}}) \;\le\; \frac{1}{1-e^{-C}}\bigg(1-\exp\bigg[-\bigg[C\Big(\tfrac{1}{2}\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}d_S^{\mathrm{MV}}(\rho_S)+\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}e_S^{\mathrm{MV}}(\rho_S)\Big)+ \frac{1}{m}\Big[\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{v\sim\rho_S}\mathrm{KL}(Q_{v,S}\|P_v)+\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathrm{KL}(\rho_S\|\pi)\Big]\bigg]\bigg]\bigg)$$

***Proof.*** Deferred to Appendix C.4. ■

### 5.4.3 Small kl Bound

If we make use, for function $D(a,b)$ between the empirical risk and the true risk, of the Kullback-Leibler divergence between two Bernoulli distributions with probability of success $a$ and $b$, we can obtain a bound similar to [53, 77]. Concretely, we apply Theorem 5.1 and Theorem 5.2 with:

$$D(a,b) \;=\; \mathrm{kl}(a,b) \;=\; a\ln\frac{a}{b}+(1-a)\ln\frac{1-a}{1-b}.$$

**Corollary 5.3.** *Let* $V\ge 2$ *be the number of views. For any distribution* $\mathcal{D}$ *on* $\mathcal{X}\times\mathcal{Y}$*, for any set of prior distributions* $\{P_v\}_{v=1}^V$ *over* $\{\mathcal{H}\}_{v=1}^V$*, for any hyper-prior distributions* $\pi$ *over views* $\mathcal{V}$*, we have:*

$$\Pr_{S\sim(\mathcal{D})^m}\Bigg(\mathrm{kl}\Big(\tfrac{1}{2}d_S^{\mathrm{MV}}(\rho)+e_S^{\mathrm{MV}}(\rho),R_{\mathcal{D}}(G_{\rho_S}^{\mathrm{MV}})\Big) \le \frac{1}{m}\Big[\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{v\sim\rho_S}\mathrm{KL}(Q_{v,S}\|P_v)+\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathrm{KL}(\rho_S\|\pi)+\ln\frac{2\sqrt{m}}{\delta}\Big]\Bigg)\ge 1-\delta$$

*and*

$$\mathrm{kl}\Big(\tfrac{1}{2}\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}d_S^{\mathrm{MV}}(\rho_S)+\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}e_S^{\mathrm{MV}}(\rho_S),\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}R_{\mathcal{D}}(G_{\rho_S}^{\mathrm{MV}})\Big) \le \frac{1}{m}\Big[\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{v\sim\rho_S}\mathrm{KL}(Q_{v,S}\|P_v)+\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathrm{KL}(\rho_S\|\pi)+\ln\, 2\sqrt{m}\Big].$$

***Proof.*** Deferred to Appendix C.5. ■

## 5.5 Generalization Bound for the Multiview $\mathcal{C}$-Bound

From a practical standpoint, as pointed out before, controlling the multiview C-Bound of Equation (5.5) can be very useful for tackling multiview learning. The next theorem is a generalization bound that justify the empirical minimization of the multiview C-bound (we use in our algorithm `PB-MVBoost` derived in Section 6.2 of Chapter 6).

**Theorem 5.4.** *Let $V \geq 2$ be the number of views. For any distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$, for any set of prior distributions $\{P_v\}_{v=1}^{V}$ over $\{\mathcal{H}\}_{v=1}^{V}$, for any hyper-prior distributions $\pi$ over views $\mathcal{V}$, and for any convex function $D : [0,1] \times [0,1] \to \mathbb{R}$, with a probability at least $1-\delta$ over the random choice of $S \sim (D)^m$ for all posterior $\{Q_v\}_{v=1}^{v}$ and hyper-posterior $\rho$ distributions, we have:*

$$R_{\mathcal{D}}(B_{\rho}^{\mathrm{MV}}) \leq 1 - \frac{\left(1 - 2 \mathop{\mathbb{E}}_{v\sim\rho} \mathtt{sup}\left(\mathbf{r}_{Q_v,\mathcal{S}}^{\delta/2}\right)\right)^2}{1 - 2 \mathop{\mathbb{E}}_{v\sim\rho} \mathtt{inf}\, \mathbf{d}_{Q_v,\mathcal{S}}^{\delta/2}},$$

*where*

$$\mathbf{r}_{Q_v,\mathcal{S}}^{\delta/2} = \left\{ r : \mathrm{kl}(R_S(G_{Q_v}) \| r) \leq \frac{1}{m}\left[\mathrm{KL}(Q_v \| P_v) + \ln\frac{4\sqrt{m}}{\delta}\right] \textit{ and } r \leq \frac{1}{2}\right\} \tag{5.6}$$

$$\textit{and} \quad \mathbf{d}_{Q_v,\mathcal{S}}^{\delta/2} = \left\{ d : \mathrm{kl}(d_{Q_v}^{S} \| d) \leq \frac{1}{m}\left[2.\mathrm{KL}(Q_v \| P_v) + \ln\frac{4\sqrt{m}}{\delta}\right]\right\} \tag{5.7}$$

***Proof.*** Let assume that the Gibbs risk $R_{\mathcal{D}}(G_{Q_v}) \leq \frac{1}{2}$. Then with a high probability over the random choice of learning sample, probabilistic bound of Corollary 5.3 in Chapter 5 says that the true Gibbs risk $R_{\mathcal{D}}(G_{Q_v})$ is included in the continuous set $\mathbf{r}_{Q_v,\mathcal{S}}^{\delta}$ defined as

$$\mathbf{r}_{Q_v,\mathcal{S}}^{\delta} = \left\{ r : \mathrm{kl}(R_S(G_{Q_v}) \| r) \leq \frac{1}{m}\left[\mathrm{KL}(Q_v \| P_v) + \ln\frac{2\sqrt{m}}{\delta}\right] \text{ and } r \leq \frac{1}{2}\right\}$$

Thus, an upper bound on $R_{\mathcal{D}}(G_{Q_v})$ is obtained from maximum value of $\mathbf{r}_{Q_v,\mathcal{S}}^{\delta}$ . From probabilistic small kl bound of expected disagreement (see Theorem 5.3), we can easily derive the continuous set $\mathbf{d}_{Q_v,\mathcal{S}}^{\delta}$ defined as

$$\mathbf{d}_{Q_v,\mathcal{S}}^{\delta} = \left\{ d : \mathrm{kl}(d_{Q_v}^{S} \| d) \leq \frac{1}{m}\left[2\,\mathrm{KL}(Q_v \| P_v) + \ln\frac{2\sqrt{m}}{\delta}\right]\right\}$$

Finally, the bound is obtained (from Equation (5.5) of Lemma 5.1) by replacing the view-specific Gibbs risk $R_{\mathcal{D}}(G_{Q_v})$ by its upper bound $\mathtt{sup}\, \mathbf{r}_{Q_v,\mathcal{S}}^{\delta/2}$ and expected disagreement $d_{\mathcal{D}}(Q_v)$ by its lower bound $\mathtt{inf}\, \mathbf{d}_{Q_v,\mathcal{S}}^{\delta/2}$. ■

## 5.6 Discussion on Related Works

In this section, we discuss two related theoretical studies of multiview learning (that are recalled in Chapter 3) related to the notion of majority vote.

Massih et al. [2] proposed a Rademacher analysis (Theorem 3.1 in Chapter 3) of the risk of the multiview majority vote over the view-specific classifiers (for more than two views) where the distribution over the views is restricted to the uniform distribution. In their work, each view-specific classifier is learned by minimizing the empirical risk: $h_v^* = \underset{h_v \in \mathcal{H}_v}{\operatorname{argmin}} \dfrac{1}{m} \sum_{(\mathbf{x},y)\sim(\mathcal{D})^m} \mathbb{1}_{[h_v(x^v)\neq y]}$. Finally, the prediction for any multiview example $\mathbf{x}$ is based on the majority vote over these view-specific classifiers. The risk of the multiview majority vote (MV-MV) is hence given by

$$R_{\mathcal{D}}(\mathtt{MV\text{-}MV}(\mathbf{x})) = \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \frac{1}{V} \sum_{v=1}^{V} \mathbb{1}_{[h_v^*(x^v)\neq y]}.$$

In comparison to our work, we considered a non-uniform distribution over the views by following a hierarchy of distributions over the view-specific classifiers. Moreover, in our bounds (Theorem 5.1 and Theorem 5.1), we have explicitly highlighted the term to control the diversity between the views, which is important for multiview learning.

Sun et al. [81] proposed a PAC-Bayesian analysis for co-regularization style multiview learning approaches (Theorem 3.2 in Chapter 3) but limited to two views and in a more restrictive setting. Indeed, they considered linear classifiers over the concatenation of two views and defined a prior distribution over the classifiers that promotes similar classification among the two views (see Section 3.3.4 for more details). In contrast to our work, we are interested in more general case when we have more than two views. Moreover, our generalization bounds are not specific to any type of classifiers and the notion of diversity among the views is handled in a different way and is inherent of the definition of the Gibbs Risk.

Lastly, both of the above approaches exploit the consensus principle (Section 3.3 in Chapter 3) of multiview learning where the objective is to maximize the agreement between the multiple views of the data. Whereas, our bounds exploits the diversity principle (Section 3.4 in Chapter 3) where our objective is to control the trade-off between the diversity between the views and the accuracy of the view-specific voters. According to the diversity principle, we exploit different informations from different views of the data. This is done by following a two-level hierarchical strategy over the view-specific classifiers and the views.

## 5.7 Conclusion

In this chapter, we propose a first PAC-Bayesian analysis of weighted majority vote classifiers for multiview learning when observations are described by more than two views. Here, our goal is to correctly combine the multiple views of the data while taking into account the diversity between the views. Therefore, we study multview learning using the PAC-Bayesian theory which allows us to derive generalization bounds for models expressed as a combination over the view-specific voters and the views. Our analysis is based on a hierarchy of distributions, *i.e.* weights, over the views and voters: *(i)* for each view $v$ a posterior and prior distributions over the view-specific voter's set, and *(ii)* a hyper-posterior and hyper-prior distribution over the set of views. We derive general PAC-Bayesian theorems (probabilistic and expected risk bounds ) tailored for this setting, that can be specialized to any convex function to compare the empirical and true risks of the stochastic Gibbs classifier associated with the weighted majority vote. We also presented a similar theorem for the expected disagreement, a notion that turns out to be crucial in multiview learning. Moreover, we derive the generalization bound for the multiview $\mathcal{C}$-bound which we use to design a boosting based algorithm `PB-MVBoost` in the next chapter. We present multiview learning algorithms based on this two-level hierarchical strategy in the next chapters.

CHAPTER 6

# MULTIVIEW BOOSTING ALGORITHM BASED ON THE MULTIVIEW $\mathcal{C}$-BOUND

In this chapter, we design a boosting based multiview learning algorithm based on two-level hierarchical strategy presented in Chapter 5, referred as `PB-MVBoost`. It iteratively learns i) weights over view-specific voters capturing view-specific information, and ii) weights over views by optimizing a PAC-Bayesian multiview $\mathcal{C}$-Bound (Equation (5.5) of Lemma 5.1) that takes into account the accuracy of the view- specific voters and the diversity between the views. Moreover, we derive another two-step multiview algorithm based on late fusion [79] strategy. It learns view-specific voters at the base level of hierarchy and then learn a multiview model over the predictions of the view-specific voters using PAC-Bayesian algorithm CqBoost (Algorithm 2 in Chapter 2). Different experiments on three publicly available datasets show the efficiency of the proposed approaches with respect to state-of-art models. This work has been done in collaboration with Dr. Pascal Germain from INRIA, Lille, France. It has been published in the proceedings of ECML-PKDD, 2017 [41] and submitted to Neurocomputing Journal.

## 6.1 Introduction

We follow the two-level hierarchical learning strategy proposed in Chapter 5, in order to design a multview learning algorithm based on the idea of boosting [31, 32, 74, 75]. We recall the idea of the two-level hierarchical learning strategy in Figure 6.1. Concretely, *i)* for each

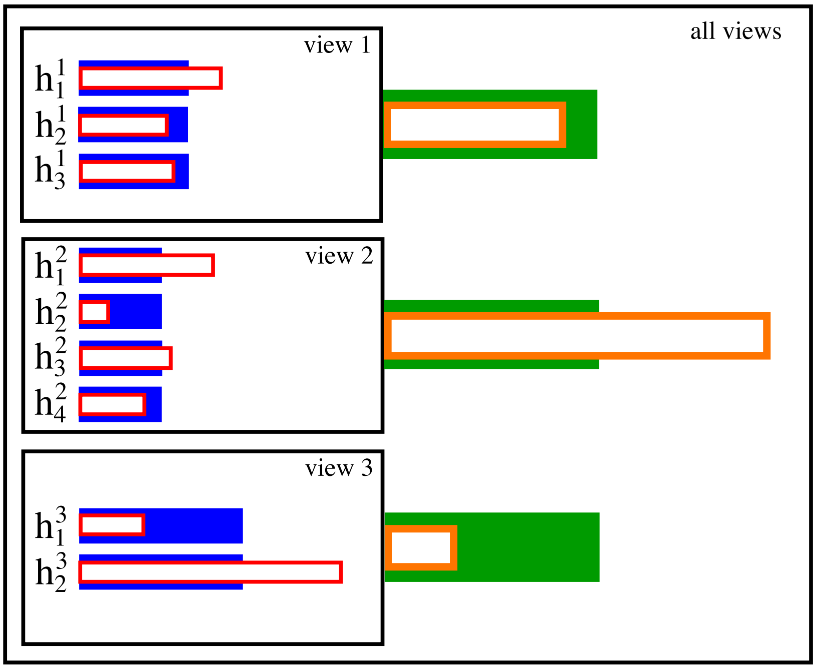


Figure 6.1: Example of the multiview distributions hierarchy with 3 views. For all views $v \in \{1,2,3\}$, we have a set of $n_v$ voters $\mathcal{H}_v = \{h_v^1, \ldots, h_v^{n_v}\}$ on which we consider a prior $P_v$ view-specific distribution (in blue). A hyper-prior $\pi$ distribution (in green) over the set of 3 views is also considered. The objective is to learn a set of posterior $\{Q_v\}_{v=1}^3$ (in red) view-specific distributions and a hyper-posterior $\rho$ distribution (in orange) leading to a good model. The length of a rectangle represents the weight (or probability) assigned to a voter or a view.

view $v$, we consider a prior $P_v$ and a posterior $Q_v$ distributions over view-specific voters to capture view-specific informations and *ii)* a hyper-prior $\pi_v$ and a hyper-posterior $\rho_v$ distributions over the set of views to capture the accuracy of view-specific classifiers and diversity between the views. Following this distributions' hierarchy, we define a multiview majority vote classifier where view-specific classifiers are weighted according to posterior and hyper-posterior distributions. By doing so, we extended the usual monoview $\mathcal{C}$-Bound to multview $\mathcal{C}$-Bound (Lemma 5.1) which bounds the error of the multiview majority vote in terms of multiview gibbs classifier and the expected disagreement (see Equation (5.2)).

From the practical point of view, we design two algorithms based on the idea of boosting [31, 32, 74, 75] and late fusion [79] (also referred as stacking [89]). Our boosting-based multiview learning algorithm, called `PB-MVBoost`, deals with the two-level hierarchical learning strategy. `PB-MVBoost` is an ensemble method and outputs a multiview classifier that is a combination of view-specific voters. It is well known that controlling the diversity between the view-specific classifiers or the views is a key element in multiview learning [2, 15, 41, 50, 56, 61] (as discussed in Chapters 3 and 5). Therefore, to learn the weights over the views, we minimize an upper-bound on the error of the majority vote using the multiview $\mathcal{C}$-bound [35, 41, 73] (proposed in Lemma 5.1), allowing us to control a trade-off between accuracy and diversity. Concretely, at each iteration of our multiview algorithm, we learn *i)* weights over view-specific voters based on their ability to deal with examples on the corresponding view (capturing view-specific informations); and *ii)* weights over views by minimizing the multiview $\mathcal{C}$-bound. Second algorithm is a two-step learning algorithm $\texttt{Fusion}^{\texttt{all}}_{\texttt{Cq}}$ [41] based on the PAC-Bayesian theory. It learns the view-specific voters at the base level of hierarchy. Finally, at second level, we combine the predictions of view-specific voters using a PAC-Bayesian algorithm CqBoost [73] (Algorithm 2 in Chapter 2) which captures both accuracy and diversity between view-specific voters.

In order to show the potential of our algorithms, we empirically evaluate our approach on $\texttt{MNIST}_1$, $\texttt{MNIST}_2$ [55] and Reuters RCV1/RCV2 collections [2]. We observe that our algorithm PB-MVBoost, empirically minimizes the multiview $\mathcal{C}$-Bound over iterations, and lead to good performances even when the classes are unbalanced. We compare PB-MVBoost with our two-step learning algorithm $\texttt{Fusion}^{\texttt{all}}_{\texttt{Cq}}$ and it came out that PB-MVBoost is more stable algorithm across different datasets and computationally faster than $\texttt{Fusion}^{\texttt{all}}_{\texttt{Cq}}$ .

In the next section, we derive our multiview learning algorithm PB-MVBoost. In Section 6.3, we discuss the relation between our algorithm with previous works. Before concluding in Section 6.5, we experiment our algorithms in Section 6.4.

## 6.2 PB-MVBoost

Following our two-level hierarchical strategy (see Figure 6.1), we aim at combining the view-specific voters (or views) leading to a well performing multiview majority vote given by Equation (5.1). Boosting [75] (presented in Sections 2.3 and 2.4.2 of Chapter 2) is a well known approach which aims at combining a set of weak voters in order to build a more efficient classifier than each of the view-specific classifiers alone. Typically, boosting algorithms repeatedly learn a “weak" voter using a learning algorithm with different probability distribution over the learning sample $S$. Finally, it combines all the weak voters in order to have one single strong classifier expressed as a majority vote which performs better than the individual weak voters. We exploit boosting paradigm to derive a multiview learning

**Algorithm 5** `PB-MVBoost`

```
Input: Training set $S = \{(\mathbf{x}_i, y_i), \dots, (\mathbf{x}_m, y_m)\}$, where $\mathbf{x}_i = (x^1, x^2, \dots, x^V)$ and $y_i \in \{-1, 1\}$.
       For each view $v \in \mathcal{V}$, a view-specific hypothesis set $\mathcal{H}_v$.
       Number of iterations $T$.
for $\mathbf{x}_i \in S$ do
    $\mathcal{D}_1(\mathbf{x}_i) \leftarrow \frac{1}{m}$
$\forall v \in \mathcal{V}$, $\rho_v^1 \leftarrow \frac{1}{V}$ and $H_v \leftarrow \phi$
for $t = 1, \dots, T$ do
    For each view $v \in \mathcal{V}$, learn a view-specific weak classifier $h_v^{(t)}$ using distribution $\mathcal{D}_{(t)}$
    Compute error: $\forall v \in \mathcal{V}$, $\epsilon_v^{(t)} \leftarrow \mathop{\mathbb{E}}_{(\mathbf{x}_i, y_i) \sim \mathcal{D}_{(t)}} \left[ \mathbb{1}_{[h_v^t(x_i^v) \neq y_i]} \right]$
    Compute classifier's weight (taking into account view specific information):
        $$\forall v \in \mathcal{V}, Q_v^{(t)} \leftarrow \frac{1}{2} \left[ \ln \left( \frac{1 - \epsilon_v^{(t)}}{\epsilon_v^{(t)}} \right) \right]$$
    $\forall v \in \mathcal{V}$, $H_v \leftarrow H_v \cup \{h_v^{(t)}\}$
    Optimize the multiview $\mathcal{C}$-Bound to learn weights over the views
        $$\max_\rho \quad \frac{\left[ 1 - 2 \sum_{v=1}^{V} \rho_v^{(t)} r_v^{(t)} \right]^2}{1 - 2 \sum_{v=1}^{V} \rho_v^{(t)} d_v^{(t)}}$$
        $$s.t. \quad \sum_{v=1}^{V} \rho_v^{(t)} = 1, \quad \rho_v^{(t)} \geq 0 \quad \forall v \in \{1, \dots, V\}$$
        where, $\forall v \in \mathcal{V}$, $r_v^{(t)} \leftarrow \mathop{\mathbb{E}}_{(\mathbf{x}_i, y_i) \sim \mathcal{D}_{(t)}} \mathop{\mathbb{E}}_{h_v \sim H_v} \left[ \mathbb{1}_{[h_v(x_i^v) \neq y_i]} \right]$
        $\forall v \in \mathcal{V}$, $d_v^{(t)} \leftarrow \mathop{\mathbb{E}}_{(\mathbf{x}_i, y_i) \sim \mathcal{D}_{(t)}} \mathop{\mathbb{E}}_{h_v, h'_v \sim H_v} \left[ \mathbb{1}_{[h_v(x_i^v) \neq h'_v(x_i^v)]} \right]$
    for $\mathbf{x}_i \in S$ do
        $\mathcal{D}_{(t+1)}(\mathbf{x}_i) \leftarrow \frac{\mathcal{D}_{(t)}(\mathbf{x}_i) \exp(-y_i \sum_{v=1}^{V} \rho_v^{(t)} (Q_v^{(t)} h_v^{(t)}(x_i^v)))}{\sum_{j=1}^{m} \mathcal{D}_{(t)}(\mathbf{x}_j) \exp(-y_j \sum_{v=1}^{V} \rho_v^{(t)} (Q_v^{(t)} h_v^{(t)}(x_j^v)))}$
Return: For each view $v \in \mathcal{V}$, weights over view-specific voters and weights over views
    i.e. $\rho^T$. Such that, for any input example $\mathbf{x}$ multiview weighted majority vote is defined
    as:
        $$B_\rho^{\mathrm{MV}}(\mathbf{x}) = \mathrm{sign} \left( \sum_{v=1}^{V} \rho_v^T \sum_{t=1}^{T} Q_v^{(t)} h_v^{(t)}(x^v) \right).$$
```

algorithm `PB-MVBoost` (see Algorithm 5) for our setting.

Note that we keep the same notations and setting as in Section 5.2 of Chapter 5. For a given training set $S = \{(\mathbf{x}_i, y_i), \ldots, (\mathbf{x}_m, y_m)\} \in (\mathcal{X} \times \{-1, +1\})^m$ of size $m$; our proposed algorithm (Algorithm 5) maintains a distribution over the examples which is initialized as the uniform distribution. Then at each iteration $t$, $V$ view-specific weak classifiers are learned according to the current distribution $\mathcal{D}_t$ (Step 5), and their corresponding errors $\epsilon_v^t$ are estimated (Step 6).

Similarly to the Adaboost algorithm [32] (recalled in Section 2.4.2 of Chapter 2), the weights of each view-specific classifier $(Q_v^{(t)})_{1\le v\le V}$ are then computed with respect to these errors as :

$$\forall v \in \mathcal{V}, Q_v^{(t)} \leftarrow \frac{1}{2}\left[\ln\left(\frac{1-\epsilon_v^{(t)}}{\epsilon_v^{(t)}}\right)\right]$$

To learn the weights $(\rho_v)_{1\le v\le V}$ over the views, we optimize the multiview $\mathcal{C}$-Bound, given by Equation (5.5) of Chapter 5 (Step 8 of algorithm). The multiview $\mathcal{C}$-Bound controls the trade-off between the expectation of the Gibbs risk $\mathop{\mathbb{E}}_{v\sim\rho} r_v^{(t)}$ and the expected disagreement $\mathop{\mathbb{E}}_{v\sim\rho} d_v^{(t)}$ over all view-specific classifiers defined as follows

$$\mathop{\mathbb{E}}_{v\sim\rho} r_v^{(t)} = \mathop{\mathbb{E}}_{(\mathbf{x}_i,y_i)\sim\mathcal{D}_{(t)}} \mathop{\mathbb{E}}_{v\sim\rho} \mathop{\mathbb{E}}_{h_v\sim H_v} \mathrm{I}[h_v(x_i^v) \neq y_i], \tag{6.1}$$

$$\text{and } \mathop{\mathbb{E}}_{v\sim\rho} d_v^{(t)} = \mathop{\mathbb{E}}_{(\mathbf{x}_i,y_i)\sim\mathcal{D}_{(t)}} \mathop{\mathbb{E}}_{v\sim\rho} \mathop{\mathbb{E}}_{h_v,h'_v\sim H_v} \mathrm{I}[h_v(x_i^v) \neq h'_v(x_i^v)]. \tag{6.2}$$

Intuitively, the minimization of the multiview $\mathcal{C}$-Bound tries to diversify the view-specific voters and views (Equation (6.2)) while controlling the classification error of the view-specific classifiers (Equation (6.1)). This allows us to control the accuracy and the diversity between the views which is an important ingredient in multiview learning [41, 61, 66, 67, 94] as discussed in Chapters 3 and 5. In Section 6.4, we empirically show that our algorithm minimizes the multiview $\mathcal{C}$-Bound over the iterations of the algorithm (this is theoretically justified by the generalization bound of Theorem 5.4). Finally, we update the distribution over training examples $\mathbf{x}_i$ (Step 9), by following the Adaboost algorithm and in a way that the weights of misclassified (resp. well classified) examples by the final weigthed majority classifier increase (resp. decrease).

$$\mathcal{D}_{(t+1)}(\mathbf{x}_i) \leftarrow \frac{\mathcal{D}_{(t)}(\mathbf{x}_i)\exp(-y_i\sum_{v=1}^{V}\rho_v^{(t)}(Q_v^{(t)}h_v^{(t)}(x_i^v)))}{\sum_{j=1}^{m}\mathcal{D}_{(t)}(\mathbf{x}_j)\exp(-y_j\sum_{v=1}^{V}\rho_v^{(t)}(Q_v^{(t)}h_v^{(t)}(x_j^v)))}$$

Intuitively, this forces the view-specific classifiers to be consistent with each other, which is important for multiview learning [46, 48, 90]. Finally, after $T$ iterations of algorithm, we learn the weights over the view-specific voters and weights over the views leading to a well-performing weighted multiview majority vote

$$B_{\rho}^{\mathrm{MV}}(\mathbf{x}) = \mathrm{sign}\left(\sum_{v=1}^{V} \rho_v^T \sum_{t=1}^{T} Q_v^{(t)} h_v^{(t)}(x^v)\right).$$

## 6.3 Discussion on Related Works

In this section, we compare existing ensemble-based multiview learning algorithms [2, 46, 48, 66, 67, 81, 90, 94] (discussed in Sections 3.3 and 3.4 of Chapter 3) with our approach.

Janodet et al. [46] and Xu and Sun [94] designed boosting based multiview learning algorithms 2-Boost and EMV-AdaBoost respectively for two-view setting. In this work, we are interested in deriving multiview learning algorithms for more general and natural case of more than two views. Koço et al. [48] proposed Mumbo that maintains separate distributions for each view in order to communicate between the views or in other words, to control the diversity between the views. On the other hand, Peng et al. [66, 67], for controlling the diversity between the views, learn the weights over the views by casting the algorithm in two ways: *i)* a multiarmed bandit framework (`rBoost.SH`) (Algorithm 4 in Chapter 3) and *ii)* an expert strategy framework (`eBoost.SH`) consisting of set of strategies (distribution over views) for weighing views. Whereas, we follow a two-level learning strategy where we learn (hyper-)posterior distributions/weights over the view-specific voters and the views. In order to take into account the accuracy and the diversity between the views, we optimize the multiview $\mathcal{C}$-Bound (an upper-bound over the risk of the multiview majority vote learned, see e.g. [35, 41, 73])

Furthermore, our approach encompasses the one of Amini et al. [2] and Xiao and Guo [90]. Amini et al. [2] proposed a Rademacher analysis (Theorem 3.1 in Chapter 3) for the majority vote over the set of view-specific classifiers (for more than two views). Xiao and Guo [90] derived a weighted majority voting Adaboost algorithm (Algorithm 3 in Chapter 3) which learns weights over view-specific voters at each iteration of the algorithm. Both of these approaches maintain a uniform distribution over the views whereas our algorithm learns the weights over the views such that they capture diversity between the views. Moreover, Sun et al.[81] proposed a PAC-Bayesian analysis for multiview learning over the concatenation of views but limited to two views and to a particular kind of voters: linear classifiers (as discussed in Section 3.3.4 of Chapter 3). This has allowed them to derive a SVM-like learning algorithm but dedicated to multiview with exactly two views. In our work, we are interested in learning from more than two views and no restriction on the classifier type. Contrary to them, we followed a two-level distributions' hierarchy where we learn weights over view-specific classifiers and weights over views.

## 6.4 Experimental Results

In this section, we present experiments to show the potential of our algorithms on the following datasets.

### 6.4.1 Datasets

**MNIST**

MNIST is a publicly available dataset consisting of 70,000 images of handwritten digits distributed over ten classes [55]. The size of the different classes in the number of images is given in Table 6.1. For our experiments, we generated 2 four-view datasets where each view is a vector of $\mathbb{R}^{14\times 14}$. Similarly than done by Chen et al. [18], the first dataset ($\texttt{MNIST}_1$) is generated by considering 4 quarters of image as 4 views. For the second dataset ($\texttt{MNIST}_2$) we consider 4 overlapping views around centre of image: this dataset brings redundancy between the views. These two datasets allow us to check if our algorithm is able to capture redundancy between the views. We reserve 10,000 of images as test samples and remaining as training samples.

| Class | zero | one | two | three | four |
|---|---|---|---|---|---|
| # Images | 6903 | 7877 | 6990 | 7141 | 6824 |

| Class | five | six | seven | eight | nine |
|---|---|---|---|---|---|
| # Images | 6313 | 6876 | 7293 | 6825 | 6958 |

Table 6.1: Number of images per class in MNIST.

**Multilingual, Multiview Text categorization**

This dataset is a multilingual text classification data extracted from Reuters RCV1/RCV2 corpus[1]. It consists of more than 110,000 documents written in five different languages (English, French, German, Italian and Spanish) distributed over 6 classes. We see different languages as different views of the data. The statistics of this dataset are presented in Table 6.2. We reserve 30% of documents as test samples and remaining as training data.

| Language | # Docs |
|---|---|
| English | 18,758 |
| French | 26,648 |
| German | 29,953 |
| Italian | 24,039 |
| Spanish | 12,342 |
| Total | 111,740 |

| Class | # Docs |
|---|---|
| C15 | 18,816 |
| CCAT | 21,426 |
| E21 | 13,701 |
| ECAT | 19,198 |
| GCAT | 19,178 |
| M11 | 19,421 |

Table 6.2: Number of documents per language (left) and per class (right) in Reuters RCV1/RCV2 corpus.

### 6.4.2 Experimental Protocol

While the datasets are multiclass, we transformed them as binary tasks by considering *one-vs-all* classification problems: for each class we learn a binary classifier by considering all the learning samples from that class as positive examples and the others as negative examples. We consider different size of learning samples $S$ (150, 200, 250, 300, 500, 800, 1000) that are chosen randomly from the training data. Since the classes are unbalanced, we report the accuracy along with the standard F1-measure [70], which is the harmonic average of precision and recall defined as

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall}\,.$$

Experiments are repeated 20 times by at each time splitting the training and the test sets at random over the initial datasets and all the scores are averaged over all the *one-vs-all* classification problems.

We design two multiview learning algorithms based on our two-step hierarchical strategy. The first algorithm is the boosting based multiview learning algorithm PB-MVBoost

[1] https://archive.ics.uci.edu/ml/datasets/Reuters+RCV1+RCV2+Multilingual,+Multiview+Text+Categorization+Test+collection

| Strategy | MNIST$_1$ | | MNIST$_2$ | | Reuters | |
|---|---|---|---|---|---|---|
| | Accuracy | $F_1$ | Accuracy | $F_1$ | Accuracy | $F_1$ |
| $\texttt{Mono}_v$ | .9034 ± .001$^\downarrow$ | .5353 ± .006$^\downarrow$ | .9164 ± .001$^\downarrow$ | .5987 ± .007$^\downarrow$ | .8420 ± .002$^\downarrow$ | .5051 ± .007$^\downarrow$ |
| `Concat` | .9224 ± .002$^\downarrow$ | .6168 ± .011$^\downarrow$ | .9214 ± .002$^\downarrow$ | .6142 ± .013$^\downarrow$ | .8431 ± .004$^\downarrow$ | .5088 ± .012$^\downarrow$ |
| $\texttt{Fusion}_{\texttt{dt}}$ | .9320 ± .001$^\downarrow$ | .5451 ± .019$^\downarrow$ | .9366 ± .001$^\downarrow$ | .5937 ± .020$^\downarrow$ | .8587 ± .003$^\downarrow$ | .4128 ± .017$^\downarrow$ |
| `MV-MV` | .9402 ± .001$^\downarrow$ | .6321 ± .009$^\downarrow$ | .9450 ± .001$^\downarrow$ | .6849 ± .008$^\downarrow$ | .8780 ± .002$^\downarrow$ | .5443 ± .012$^\downarrow$ |
| `rBoost.SH` | .9256 ± .001$^\downarrow$ | .5315 ± .009$^\downarrow$ | .9545 ± .0007 | .7258 ± .005$^\downarrow$ | .8853 ± .002 | .5718 ± .011$^\downarrow$ |
| `MV-AdaBoost` | *.9514* ± .001 | .6510 ± .012$^\downarrow$ | *.9641* ± .0009 | .7776 ± .007$^\downarrow$ | .8942 ± .006 | .5581 ± .013$^\downarrow$ |
| `MV-Boost` | .9494 ± .003$^\downarrow$ | *.7733* ± .009$^\downarrow$ | .9555 ± .002 | *.7910* ± .006$^\downarrow$ | .8627 ± .007$^\downarrow$ | .5789 ± .012$^\downarrow$ |
| $\texttt{Fusion}_{\texttt{Cq}}^{\texttt{all}}$ | .9418 ± .002$^\downarrow$ | .6120 ± .040$^\downarrow$ | .9548 ± .003$^\downarrow$ | .7217 ± .041$^\downarrow$ | **.9001** ± .003 | **.6279** ± .019 |
| `PB-MVBoost` | **.9661** ± .0009 | **.8066** ± .005 | **.9674** ± .0009 | **.8166** ± .006 | *.8953* ± .002 | *.5960* ± .015$^\downarrow$ |

Table 6.3: Test classification accuracy and $F_1$-measure of different approaches averaged over all the classes and over 20 random sets of $m = 500$ labeled examples per training set. Along each column, the best result is in bold, and second one in italic. $^\downarrow$ indicates that a result is statistically significantly worse than the best result, according to a Wilcoxon rank sum test with $p < 0.02$.

described in Section 6.2. Second one is a two-step multiview learning algorithm based on classifier late fusion approach [79]. We call this algorithm $\texttt{Fusion}_{\texttt{Cq}}^{\texttt{all}}$ [41]. Concretely, at first level, we learn different view-specific linear SVM models (recalled in Section 2.4.1 of Chapter 2) with different hyperparameter $C$ values (12 values between $10^{-8}$ and $10^{3}$). Finally, at the second level, we learn a weighted combination over the predictions of view-specific voters using PAC-Bayesian algorithm `CqBoost`[73] (recalled in Section 2.4.3 of Chapter 2) with a RBF kernel. Note that, `CqBoost` tends to minimize the PAC-Bayesian $\mathcal{C}$-Bound [35] controlling the trade-off between accuracy and disagreement between voters. The hyperparameter $\gamma$ of the RBF kernel (presented in Section 2.4.1 of Chapter 2 ) is chosen over a set of 9 values between $10^{-6}$ and $10^{2}$; and hyperparameter $\mu$ of $\texttt{Fusion}_{\texttt{Cq}}^{\texttt{all}}$ is chosen over a set of 8 values between $10^{-8}$ and $10^{-1}$. To study the potential of our algorithms ($\texttt{Fusion}_{\texttt{Cq}}^{\texttt{all}}$ and `PB-MVBoost`), we considered following 7 baseline approaches:

- $\texttt{Mono}_v$: We learn a view-specific model for each view using a decision tree classifier and report the results of the best performing view.
- `Concat`: We learn one model using a decision tree classifier by concatenating features of all the views.
- $\texttt{Fusion}_{\texttt{dt}}$ : This is a late fusion approach where we first learn the view-specific classifiers using 60% of learning sample Then, we learn a final multiview weighted model over the predictions of the view-specific classifiers. For this approach, we used decision tree classifier at both levels of learning.

- MV-MV: We compute a multiview uniform majority vote (similar to approach followed by Amini et al. [2]) over all the view-specific classifiers' outputs in order to make final prediction. We learn view-specific classifiers using decision tree classifier (Equation (3.11) in Chapter 3).

- rBoost.SH (Algorithm 4 in Chapter 3): This is the multiview learning algorithm proposed by Peng et al. [66, 67] where a single global distribution is maintained over the learning sample for all the views and the distribution over views are updated using multiarmed bandit framework. At each iteration, rBoost.SH selects a view according to the current distribution and learns the corresponding view-specific voter. For tuning the parameters, we followed the same experimental setting as Peng et al. [66].

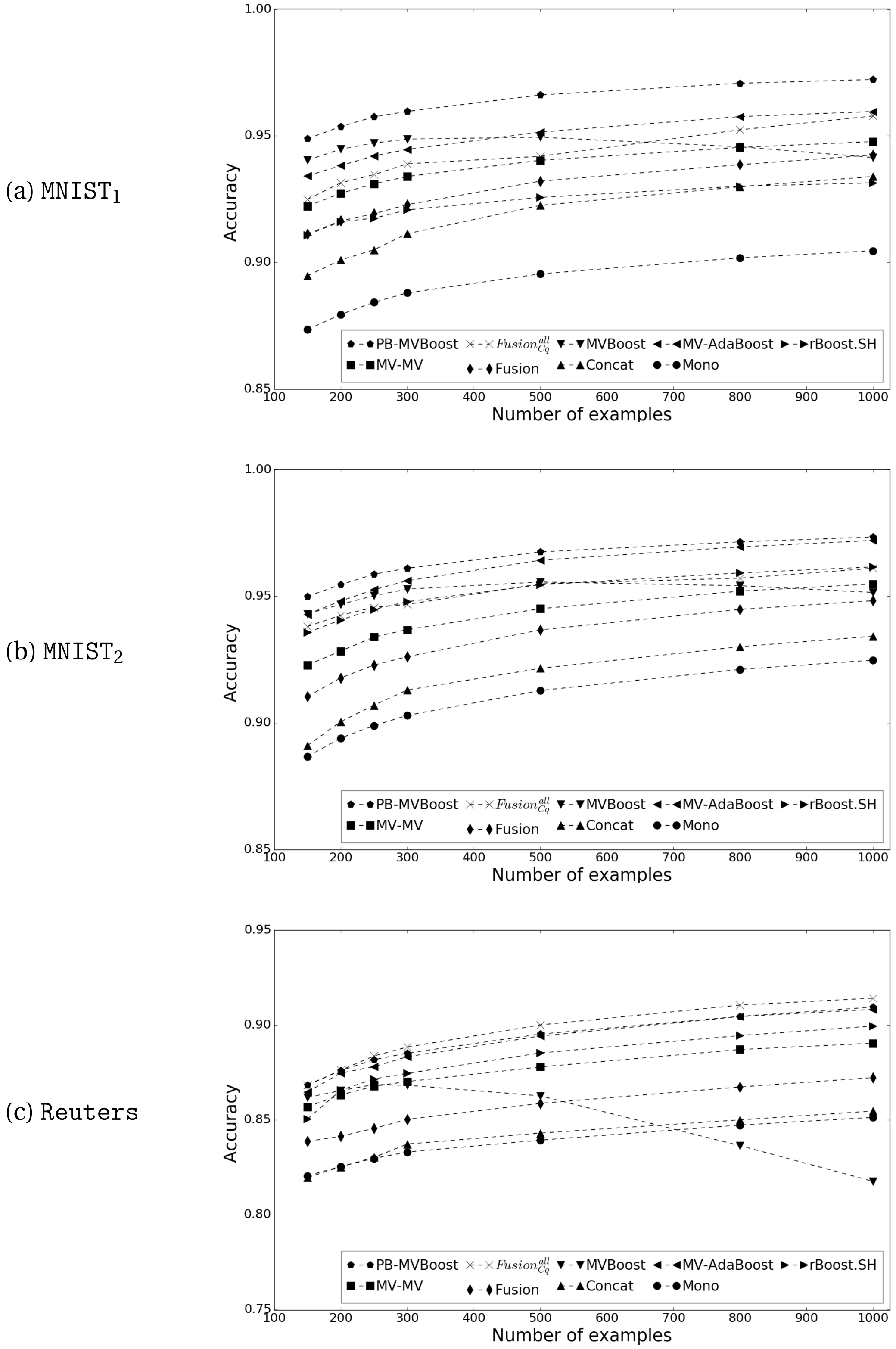
- MV-AdaBoost: This is a majority vote classifier over the view-specific voters trained using Adaboost algorithm. Here, our objective is to see the effect of maintaining separate distributions for all the views.

- MV-Boost (Algorithm 6): This is a variant of our algorithm PB-MVBoost but without learning weights over views by optimizing multiview $\mathcal{C}$-Bound. Here, our objective is to see the effect of learning the weights over the views for multiview learning tasks.

For all boosting based approaches (rBoost.SH, MV-AdaBoost, MV-Boost and PB-MVBoost), we learn the view-specific voters using a decision tree classifier with depth 2 and 4 as a weak classifier for MNIST, and Reuters RCV1/RCV2 datasets respectively. For all these approaches, we set the number of iterations to $T = 100$. For optimizing the multiview $\mathcal{C}$-Bound, we used Sequential Least SQuares Programming (SLSQP) implementation provided by scikit-learn [65]. Note that we made use of the scikit-learn [65] implementation for learning the decision tree models.

### 6.4.3 Results

Firstly, we report the comparison of our algorithms $\texttt{Fusion}_{\texttt{Cq}}^{\texttt{all}}$ and PB-MVBoost (for $m = 500$) with all the considered baseline methods in Table 6.3. Secondly, Figure 6.2 and Figure 6.3, illustrates the evolution of the accuracy and the $F_1$-measure according to the size of the learning sample. From the table, the proposed two-step learning algorithm $\texttt{Fusion}_{\texttt{Cq}}^{\texttt{all}}$ is significantly better than the baseline approaches for Reuters dataset. Whereas, our boosting based algorithm PB-MVBoost is significantly better than all the baseline approaches for all the datasets. This shows that considering a two-level hierarchical strategy in a PAC-Bayesian manner is an effective way to handle multiview learning.

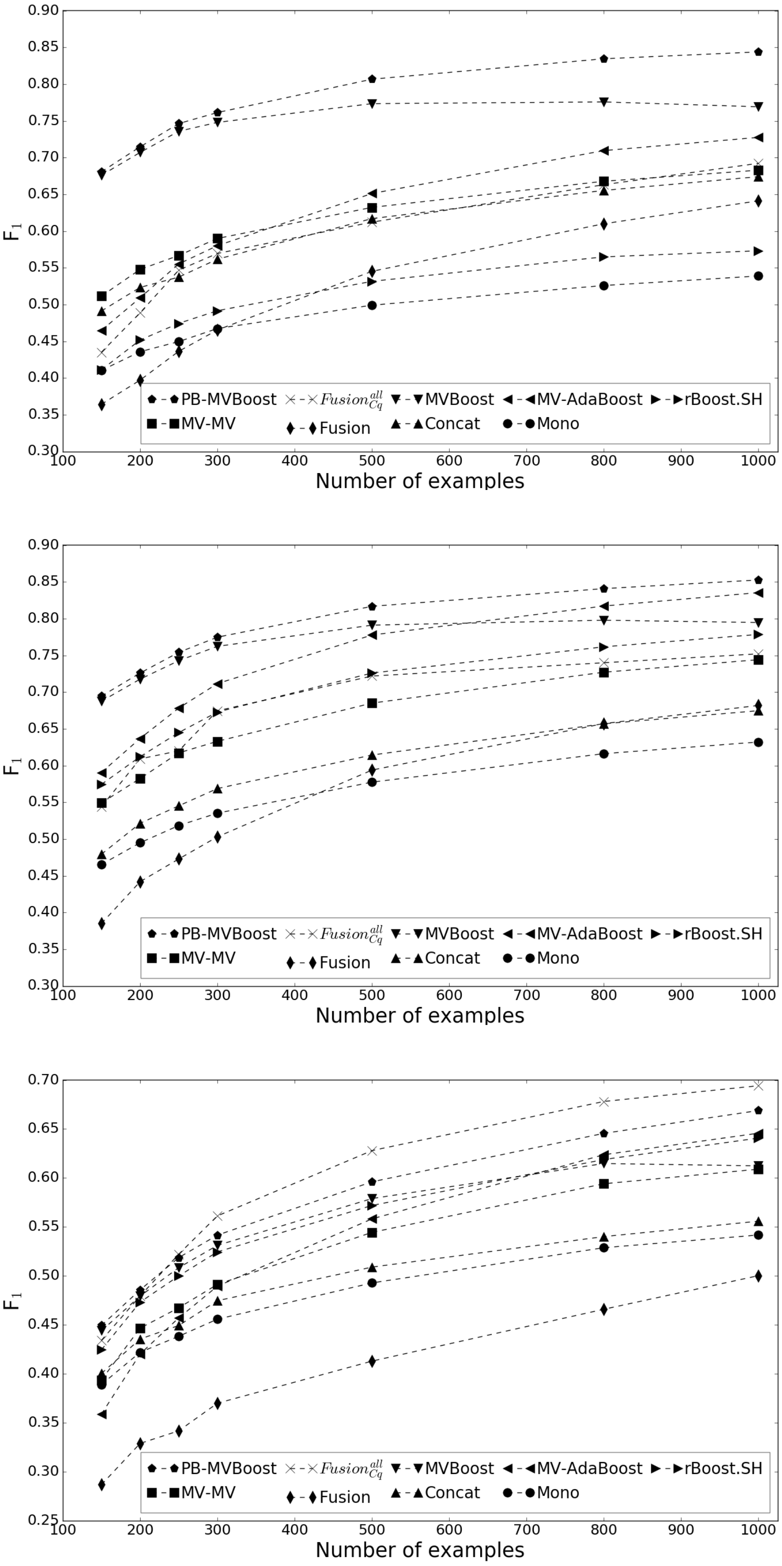


(a) MNIST$_1$

(b) MNIST$_2$

(c) Reuters

Figure 6.2: Evolution of accuracy with respect to the number of labeled examples in the initial labeled training sets on MNIST$_1$, MNIST$_2$ and Reuters datasets.



(a) $\mathtt{MNIST}_1$

(b) $\mathtt{MNIST}_2$

(c) Reuters

Figure 6.3: Evolution of $F_1$-measure with respect to the number of labeled examples in the initial labeled training sets on $\mathtt{MNIST}_1$, $\mathtt{MNIST}_2$ and Reuters datasets.

**Algorithm 6** `MV-Boost`

```
Input: Training set $S = \{(\mathbf{x}_i, y_i), \dots, (\mathbf{x}_m, y_m)\}$, where $\mathbf{x}_i = (x^1, x^2, \dots, x^V)$ and $y_i \in \{-1, 1\}$.
       For each view $v \in \mathcal{V}$, a view-specific hypothesis set $\mathcal{H}_v$.
       Number of iterations $T$.
for $\mathbf{x}_i \in S$ do
    $\mathcal{D}_1(\mathbf{x}_i) \leftarrow \frac{1}{m}$
for $t = 1, \dots, T$ do
    For each view $v \in \mathcal{V}$, learn a view-specific weak classifier $h_v^{(t)}$ using distribution $\mathcal{D}_{(t)}$
    Compute error: $\forall v \in \mathcal{V},\ \epsilon_v^{(t)} \leftarrow \underset{(\mathbf{x}_i, y_i) \sim \mathcal{D}_{(t)}}{\mathbb{E}} \left[ \mathbb{1}_{[h_v^{(t)}(x_i^v) \neq y_i]} \right]$
    Compute classifier's weight (taking into account view specific information):
        $$\forall v \in \mathcal{V}, Q_v^{(t)} \leftarrow \frac{1}{2} \left[ \ln \left( \frac{1 - \epsilon_v^{(t)}}{\epsilon_v^{(t)}} \right) \right]$$
    for $\mathbf{x}_i \in S$ do
        $\mathcal{D}_{(t+1)}(\mathbf{x}_i) \leftarrow \frac{\mathcal{D}_{(t)}(\mathbf{x}_i) \exp(-y_i \sum_{v=1}^{V} (1/V)(Q_v^{(t)} h_v^{(t)}(x_i^v)))}{\sum_{j=1}^{m} \mathcal{D}_{(t)}(\mathbf{x}_j) \exp(-y_j \sum_{v=1}^{V} (1/V)^t (Q_v^{(t)} h_v^{(t)}(x_j^v)))}$
Return: For each view $v \in \mathcal{V}$, weights over view-specific voters.
```

In Figure 6.4, we compare proposed algorithms $\texttt{Fusion}_{\texttt{Cq}}^{\texttt{all}}$ and `PB-MVBoost` in terms of accuracy, $F_1$-score and time complexity for $m = 500$ examples. For `MNIST` datasets, `PB-MVBoost` is significantly better than $\texttt{Fusion}_{\texttt{Cq}}^{\texttt{all}}$ . For `Reuters` dataset, $\texttt{Fusion}_{\texttt{Cq}}^{\texttt{all}}$ performs better than `PB-MVBoost` but the computation time for $\texttt{Fusion}_{\texttt{Cq}}^{\texttt{all}}$ is much higher than the one of `PB-MVBoost`. Moreover, in Figure 6.3, we can see that the performance (in terms of $F_1$-measure) for $\texttt{Fusion}_{\texttt{Cq}}^{\texttt{all}}$ is worse than `PB-MVBoost` when we have less number of training examples ($m = 150$ and 200). This shows that the proposed boosting based one-step algorithm `PB-MVBoost` is more stable and more effective for multiview learning.

From Table 6.3, Figure 6.2 and Figure 6.3, we can observe that `MV-AdaBoost` (where we have different distributions for each view over the learning sample) provides better results compared to other baselines in terms of accuracy but not in terms of F1-measure. On the other hand, `MV-Boost` (where we have single global distribution over learning sample but without learning weights over views) is the best among baselines in terms of F1-measure. Moreover, the performances of `MV-Boost` first increases with an increase of the quantity of the training examples, then decreases. Whereas our algorithm `PB-MVBoost` provides the best results in terms of both accuracy and F1-measure, and leads to a monotonically increase of the performances with respect to the addition of labeled examples. This confirms

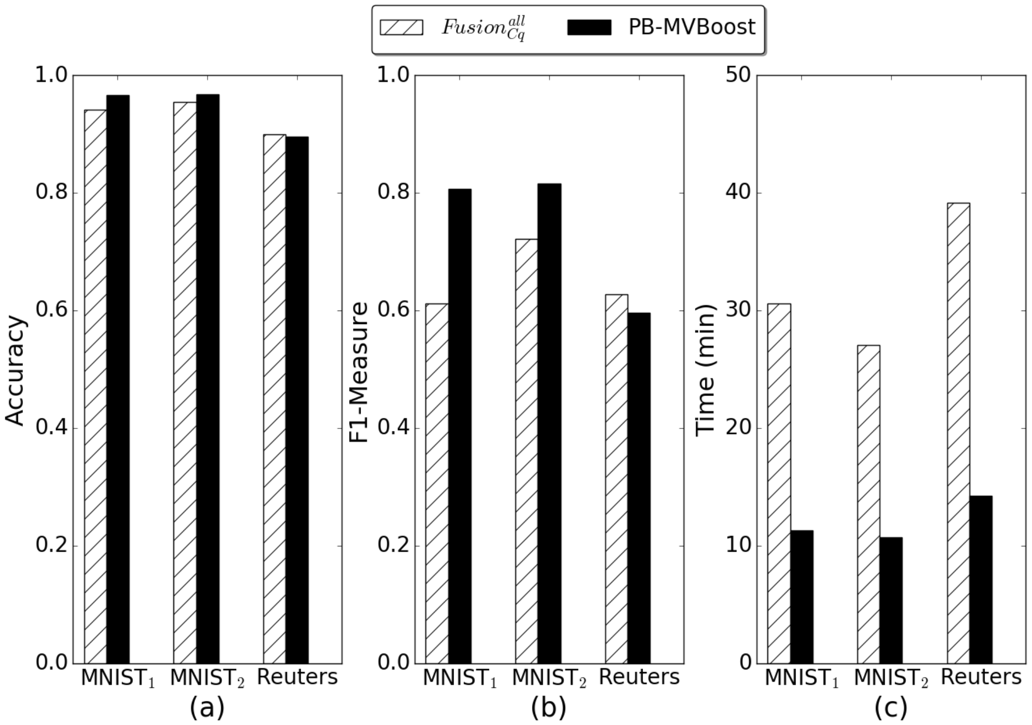
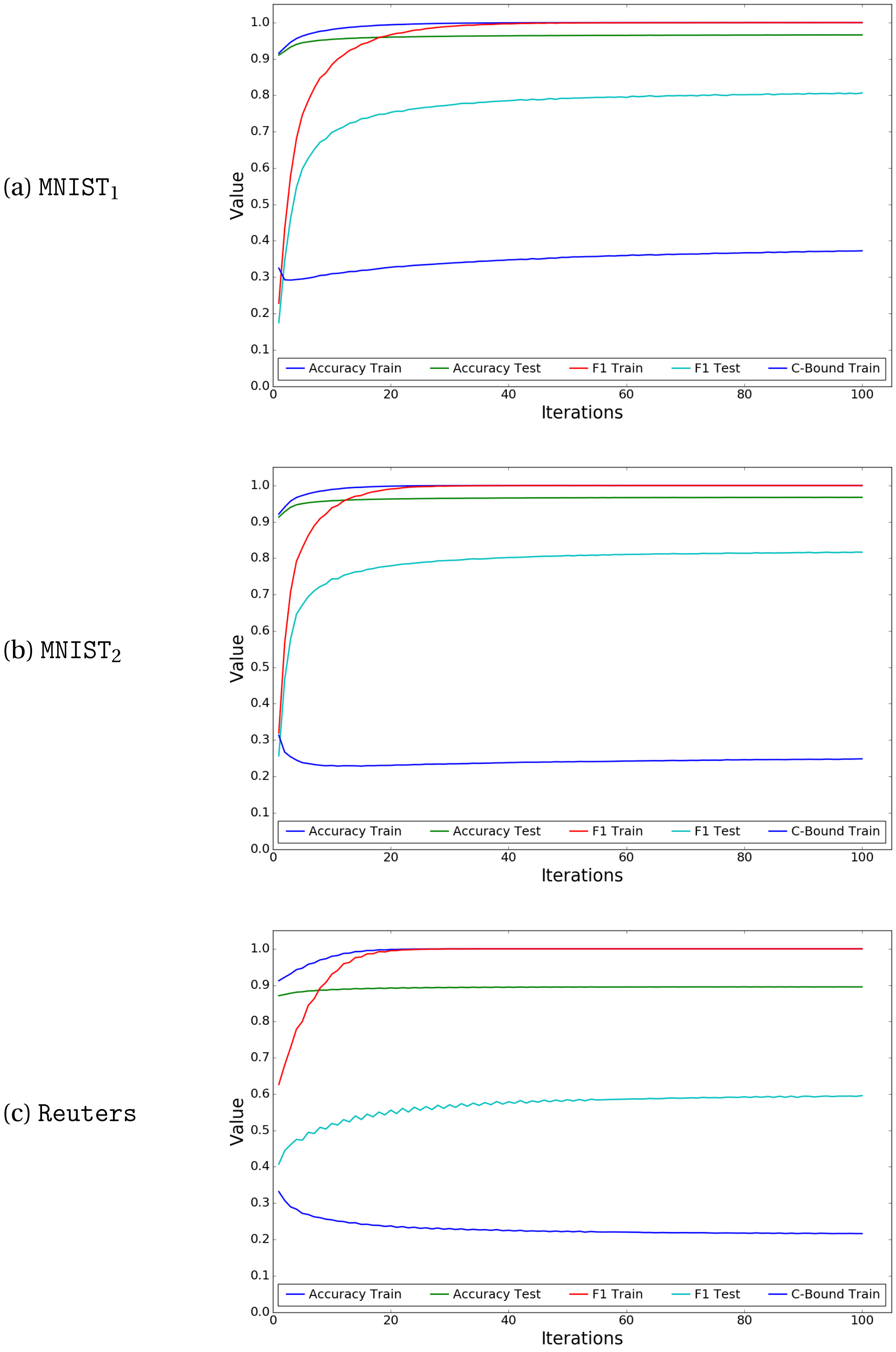


Figure 6.4: Comparison between $\texttt{Fusion}_{\texttt{Cq}}^{\texttt{all}}$ and `PB-MVBoost` in terms Accuracy (a), F1-Measure (b) and Time Complexity (c) for $m = 500$

that by maintaining a single global distribution over the views and learning the weights over the views using a PAC-Bayesian framework, we are able to take advantage of different representations (or views) of the data.

Finally, we plot the behaviour of our algorithm `PB-MVBoost` over $T = 100$ iterations on Figure 6.5 for all the datasets. We plot accuracy and F1-measure of the learned model on training and test data along with the empirical multiview $\mathcal{C}$-Bound on the training data at each iteration of our algorithm. Over the iterations, the F1-measure on the test data keeps on increasing for all the datasets even if F1-measure and accuracy on the training data reach the maximal value. This confirms that our algorithm handles unbalanced data well. Moreover, the empirical multiview $\mathcal{C}$-Bound (which controls the trade-off between accuracy and diversity between views) keeps on decreasing over the iterations. This validates that by combining the PAC-Bayesian framework with the boosting one, we can empirically ensure the view specific informations and diversity between the views for multiview learning.



(a) $\mathtt{MNIST}_1$

(b) $\mathtt{MNIST}_2$

(c) $\mathtt{Reuters}$

Figure 6.5: Plots for classification accuracy and F1-measure on training and test data; and empirical multiview $\mathcal{C}$-Bound on training data over the iterations for all datasets with $m = 500$.

### 6.4.4 A note on the Complexity of PB-MVBoost

The complexity of learning decision tree classifiers is $O(d\,mlog(m))$, where $d$ is depth of decision tree. We learn the weights over the views by optimizing Equation (5.5) (Step 8 of our algorithm) using SLSQP method which has time complexity of $O(V^3)$. Therefore, the overall complexity is $O\big(T\,\big(V^3 + V\,d_v\,m.log(m)\big)\big)$. Note that it is easy to parallelize our algorithm: by using $V$ different machines, we can learn the view-specific classifiers and weights over them (Steps 4 to 7).

## 6.5 Conclusion

We propose a boosting-based learning algorithm, called as PB-MVBoost. At each iteration of the algorithm, we learn the weights over the view-specific voters and the weights over the views by optimizing an upper-bound over the risk of the majority vote (the multiview $\mathcal{C}$-Bound) that has the advantage to allow to control a trade-off between accuracy and the diversity between the views. The empirical evaluation shows that PB-MVBoost leads to good performances and confirms that our two-level PAC-Bayesian strategy is indeed a nice way to tackle multiview learning. Moreover, we compare the effect of maintaining separate distributions over learning sample for each view; single global distribution over views; and single global distribution along with learning weights over views on results of multiview learning. We show that by maintaining a single global distribution over learning sample for all the views and learning the weights over the views is effective way to deal with multiview learning. In this way, we are able to capture the view-specific informations and control the diversity between the views. Moreover, we proposed a two-step learning algorithm $\texttt{Fusion}_{\texttt{Cq}}^{\texttt{all}}$ which is based on PAC-Bayesian theory. Finally, we experimentally show that PB-MVBoost is more stable and computationally faster than $\texttt{Fusion}_{\texttt{Cq}}^{\texttt{all}}$ . In the next chapter, we show that the empirical risk minimization of the multiview majority vote is equivalent to the minimization of Bregman divergences. This allowed us to derive a parallel-update optimization algorithm for multiview learning.

# Chapter 7

# Multiview Learning as Bregman Distance Optimization

In this chapter, we derive a multiview learning algorithm where we jointly learns view-specific weighted majority vote classifiers (*i.e.* for each view) over a set of base voters, and a second weighted majority vote classifier over the set of these view-specific weighted majority vote classifiers. We show that the empirical risk minimization of the final majority vote given a multiview training set can be cast as the minimization of Bregman divergences. This allows us to derive a parallel-update optimization algorithm for learning our multiview model. We empirically study our algorithm with a particular focus on the impact of the training set size on the multiview learning results. The experiments show that our approach is able to overcome the lack of labeled information. It has been accepted at CAp, 2018 [39] and published in the proceedings of IDA, 2018 [38].

## 7.1 Introduction

In this chapter, we propose a multiview Boosting-based algorithm, called $\mathtt{M}\omega\mathtt{MvC}^2$, for the general case where observations are described by more than two views. Our algorithm combines previously learned view-specific classifiers as in [2] but with the difference that it jointly learns two sets of weights for, first, combining view-specific *weak classifiers*; and then combining the obtained view-specific weighted majority vote classifiers to get a final weighted majority vote classifier. We show that the minimization of the classification error over a multiview training set can be cast as the minimization of Bregman divergences allowing the development of an efficient parallel update scheme to learn the weights. Using a large publicly available corpus of multilingual documents extracted from the `Reuters` RCV1 and RCV2 corpora as well as $\texttt{MNIST}_1$ and $\texttt{MNIST}_2$ collections, we show that our approach consistently improves over other methods, in the particular when there are only few training examples available for learning. This is a particularly interesting setting when resources are limited, and corresponds, for example, to the common situation of multilingual data.

In the next section, we present the double weighted majority vote classifier for multiview learning. Section 7.3 shows that the learning problem is equivalent to a Bregman-divergence minimization and describes the Boosting-based algorithm we developed to learn the classifier. In Section 7.4, we present experimental results obtained with our approach. Finally, in Section 7.5 we discuss the outcomes of this study and give some pointers to further research.

## 7.2 Notations and Setting

We consider binary classification problems where the multiview observations $\mathbf{x} = (x^1, \ldots, x^V)$ belong to a multiview input set $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_V$, where $V \geq 2$ is the number of views of not-necessarily the same dimension. We denote $\mathcal{V}$ the set of the $V$ views. In binary classification, we assume that examples are pairs $(\mathbf{x}, y)$, with $y \in \mathcal{Y} = \{-1, +1\}$, drawn according to an unknown distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$. We further assume that we have a finite set of *weak classifiers* $\mathcal{H}_v = \{h_v^j : \mathcal{X}_v \to \{-1, +1\} \mid j \in \{1, \ldots, n_v\}\}$, where $n_v$ is number of view-specific weak classifiers. We aim at learning a two-level encompassed weighted majority vote classifier where at the first level a weighted majority vote is build for each view $v \in \mathcal{V}$ over the associated set of weak classifiers $\mathcal{H}_v$, and the final classifier, referred to as the Multiview double $\omega$eighted Majority vote Classifier ($\texttt{M}\omega\texttt{MvC}^2$), is a weighted majority vote over the previous view-specific majority vote classifiers (see Figure 7.1 for an illustration). Given a training set $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$ of size $m$ drawn *i.i.d.* with respect to a fixed, yet unknown, distribution $\mathcal{D}$ over $(\mathcal{X}_1 \times \cdots \times \mathcal{X}_V) \times \mathcal{Y}$, the learning objective is to train the weak view-specific classifiers $(\mathcal{H}_v)_{1 \leq v \leq V}$ and to choose two sets of weights; $\mathbf{Q} = (Q_v)_{1 \leq v \leq V}$, where $\forall v \in \mathcal{V}$, $Q_v = (Q_v^j)_{1 \leq j \leq n_v}$, and $\rho = (\rho_v)_{1 \leq v \leq V}$, such that the multiview weighted majority vote classifier $B_\rho^{\mathrm{MV}}$

$$B_\rho^{\mathrm{MV}}(\mathbf{x}) = \sum_{v=1}^{V} \rho_v \sum_{j=1}^{n_v} Q_v^j h_v^j(x^v) \tag{7.1}$$

has the smallest possible generalization error on $\mathcal{D}$. We follow the Empirical Risk Minimization principle [87], and aim at minimizing the 0/1-loss over $S$:

$$R_S(B_\rho^{\mathrm{MV}}) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}_{[y_i B_\rho^{\mathrm{MV}}(\mathbf{x}_i) \leq 0]},$$

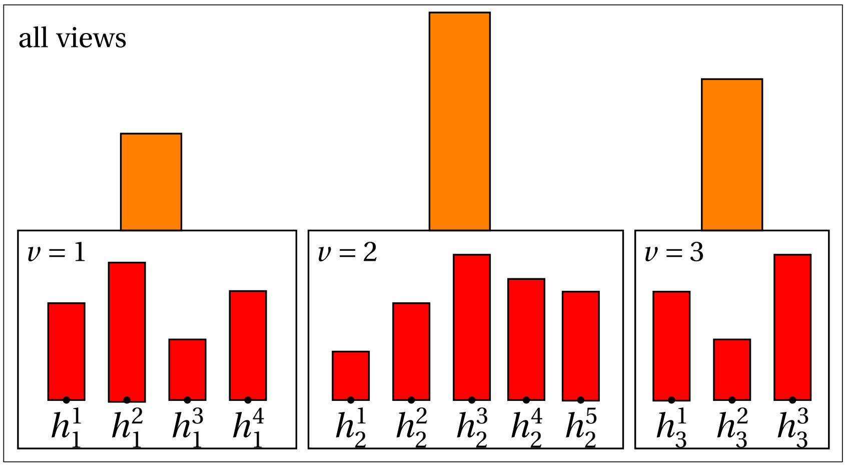

all views
$v = 1$
$v = 2$
$v = 3$
$h_1^1$ $h_1^2$ $h_1^3$ $h_1^4$
$h_2^1$ $h_2^2$ $h_2^3$ $h_2^4$ $h_2^5$
$h_3^1$ $h_3^2$ $h_3^3$

Figure 7.1: Illustration of $\texttt{M}\omega\texttt{MvC}^2$ with $V$=3. For all views $v \in \{1,2,3\}$, we have a set of view-specific weak classifiers $(\mathcal{H}_v)_{1\le v\le V}$ that are learned over a multiview training set. The objective is then to learn the weights $\mathbf{Q}$ (red histograms) associated to $(\mathcal{H}_v)_{1\le v\le V}$; and the weights $\rho$ (orange histograms) associated to weighted majority vote classifiers such that the multiview weighted majority vote classifier $B_\rho$ (Equation 7.1) will have the smallest possible generalization error.

where $\mathbb{1}_p$ is equal to 1 if the predicate $p$ is true, and 0 otherwise. As this loss function is non-continuous and non-differentiable, it is typically replaced by an appropriate convex and differentiable proxy. Here, we replace $\mathbb{1}_{z\le 0}$ by the logistic upper bound $a\log(1+e^{-z})$, with $a$=$(\log 2)^{-1}$. The misclassification cost becomes

$$R_S(B_\rho^{\mathrm{MV}}) = \frac{a}{m}\sum_{i=1}^{m} \ln\Big(1+\exp\big(-y_i B_\rho^{\mathrm{MV}}(\mathbf{x}_i)\big)\Big), \tag{7.2}$$

and the objective would be then to find the optimal combination weights $\mathbf{Q}^\star$ and $\rho^\star$ that minimize this surrogate logistic loss.

Note that the two-level hierarchical strategy considered in Chapters 5 and 6 is different from this one. In the PAC-Bayesian theory, for a given set of view-specific classifiers, we assume (hyper-)prior distributions over the view-specific classifiers and the views. Then, after seeing the learning sample, our objective is to learn the (hyper-)posterior distributions over the view-specific classifiers and the views leading to a well-performing multiview majority vote. However, in this chapter, there is no notion of (hyper-)prior distributions over the set of view-specific classifiers. Here, the objective is to learn the optimal combination weights $\mathbf{Q}^\star$ and $\rho^\star$ by minimizing surrogate logistic loss defined by Equation (7.2).

## 7.3 An Iterative Parallel update Algorithm to Learn $\mathtt{M}\omega\mathtt{MvC}^2$

In this section, we first show how the minimization of the surrogate loss of Equation (7.2) is equivalent to the minimization of a given Bregman divergence (Definition 7.1). Then, this equivalence allows us to employ a parallel-update optimization algorithm to learn the weights $\mathbf{Q}{=}(Q_v)_{1\le v\le V}$ and $\rho$ leading to this minimization.

### 7.3.1 Bregman-divergence optimization

We first recall the definition of a Bregman divergence [13, 52].

**Definition 7.1** (Bregman divergence)**.** Let $\Omega\subseteq\mathbb{R}^m$ and $F:\Omega\to\mathbb{R}$ be a continuously differentiable and strictly convex real-valued function. The Bregman divergence $D_F$ associated to $F$ is defined for all $(\mathbf{p},\mathbf{q})\in\Omega\times\Omega$ as

$$D_F(\mathbf{p}||\mathbf{q}) = F(\mathbf{p}) - F(\mathbf{q}) - \left\langle \nabla F(\mathbf{q}), (\mathbf{p}-\mathbf{q})\right\rangle, \tag{7.3}$$

where $\nabla F(\mathbf{q})$ is the gradient of $F$ estimated at $\mathbf{q}$, and the operator $\langle\cdot,\cdot\rangle$ is the dot product function.

The optimization problem arising from this definition that we are interested in, is to find a vector $\mathbf{p}^\star\in\Omega$—that is the closest to a given vector $\mathbf{q}_0\in\Omega$—under the set $\mathcal{P}$ of $V$ linear constraints

$$\mathcal{P} = \{\mathbf{p}\in\Omega|\forall v\in\mathcal{V},\ \rho_v\mathbf{p}^\top\mathbf{M}_v = \rho_v\tilde{\mathbf{p}}^\top\mathbf{M}_v\},$$

where $\tilde{\mathbf{p}}\in\Omega$ is a specified vector, and $\mathbf{M}_v$ is a $m{\times}n_v$ matrix with $n_v{=}|\mathcal{H}_v|$ the number of weak classifiers for view $v{\in}\mathcal{V}$. Defining the Legendre transform as

$$L_F\left(\mathbf{q},\sum_{v=1}^{V}\rho_v\mathbf{M}_vQ_v\right) = \operatorname*{arg\,min}_{\mathbf{p}\in\Omega}\left\{D_F(\mathbf{p}||\mathbf{q}) + \sum_{v=1}^{V}\left\langle\rho_v\mathbf{M}_vQ_v,\mathbf{p}\right\rangle\right\}.$$

the dual optimization problem can be stated as finding a vector $\mathbf{q}^\star$ in $\bar{\mathcal{Q}}$, the closure of the set

$$\mathcal{Q} = \left\{\mathbf{q} = L_F\left(\mathbf{q}_0,\sum_{v=1}^{V}\rho_v\mathbf{M}_vQ_v\right)\middle|\rho\in\mathbb{R}^V;\forall v, Q_v\in\mathbb{R}^{n_v}\right\},$$

for which $D_F(\tilde{\mathbf{p}}||\mathbf{q}^\star)$ is the lowest. It can be shown that both of these optimization problems have the same unique solution [23, 52], with the advantage of having parallel-update optimization algorithms to find the solution of the dual form in the mono-view case [19, 21, 23], making the use of the latter more appealing.

According to our multiview setting and to optimize Equation (7.2) through a Bregman divergence, we consider the function $F$ defined for all $\mathbf{p}{\in}\Omega{=}[0,1]^m$ as

$$F(\mathbf{p}) = \sum_{i=1}^{m} p_i \ln(p_i) + (1-p_i)\ln(1-p_i),$$

which from Definition 7.1 and the definition of the Legendre transform, yields that for all $(\mathbf{p},\mathbf{q}) \in \Omega \times \Omega$ and $\mathbf{r} \in \Omega$

$$D_F(\mathbf{p}||\mathbf{q}) = \sum_{i=1}^{m} p_i \ln\left(\frac{p_i}{q_i}\right) + (1-p_i)\ln\left(\frac{1-p_i}{1-q_i}\right), \tag{7.4}$$

$$\text{and } \forall i \in [m],\ L_F(\mathbf{q},\mathbf{r})_i = \frac{q_i e^{-r_i}}{1-q_i+q_i e^{-r_i}}, \tag{7.5}$$

with $a_i$ the $i^{th}$ characteristic of $\mathbf{a}{=}(a_i)_{1\le i\le m}$ ($\mathbf{a}$ being $\mathbf{p}$, $\mathbf{q}$, $\mathbf{r}$ or $L_F(\mathbf{q},\mathbf{r})$).

Now, let $\mathbf{q}_0 = \frac{1}{2}\mathbf{1}_m$ be the vector with all its components set to $\frac{1}{2}$. For all $i \in \{1,\dots,m\}$, we define $L_F(\mathbf{q}_0,\mathbf{v})_i = \sigma(v_i)$ with $\sigma(z) = (1+e^z)^{-1}$, $\forall z \in \mathbb{R}$. We set the matrix $\mathbf{M}_v$ of size $m \times n_v$, $(\mathbf{M}_v)_{ij} = y_i h_v^j(x_i^v)$. Then using Equations (7.4) and (7.5), it comes

$$D_F\left(\mathbf{0}\middle|\middle|L_F\left(\mathbf{q}_0, \sum_{v=1}^{V}\rho_v\mathbf{M}_vQ_v\right)\right) = \sum_{i=1}^{m}\ln\left(1+\exp\left(-y_i\sum_{v=1}^{V}\rho_v\sum_{j=1}^{n_v}Q_v^jh_v^j(x_i^v)\right)\right). \tag{7.6}$$

As a consequence, minimizing Equation (7.2) is equivalent to minimizing $D_F(\mathbf{0}||\mathbf{q})$ over $\mathbf{q} \in \bar{\mathcal{Q}}_0$, where for $\Omega = [0,1]^m$

$$\mathcal{Q}_0 = \left\{\mathbf{q} \in \Omega \middle| q_i = \sigma\left(y_i\sum_{v=1}^{V}\rho_v\sum_{j=1}^{n_v}Q_v^jh_v^j(x_i^v)\right); \rho, \mathbf{Q}\right\}. \tag{7.7}$$

For a set of weak-classifiers $(\mathcal{H}_v)_{1\le v\le V}$ learned over a training set $S$; this equivalence allows us to adapt the parallel-update optimization algorithm described in [19] to find the optimal weights $\mathbf{Q}$ and $\rho$ defining MωMvC$^2$ of Equation (7.1), as described in Algorithm 7.

### 7.3.2 A Multiview Parallel Update Algorithm

Once all view-specific *weak classifiers* $(\mathcal{H}_v)_{1\le v\le V}$ have been trained, we start from an initial point $\mathbf{q}^{(1)} \in \mathcal{Q}_0$ (Equation (7.7)) corresponding to uniform values of weights $\rho^{(1)} = \frac{1}{V}\mathbf{1}_V$ and $\forall v \in [V]$, $Q_v^{(1)} = \frac{1}{n_v}\mathbf{1}_{n_v}$. Then, we iteratively update the weights such that at each iteration $t$, using the current parameters $\rho^{(t)}, \mathbf{Q}^{(t)}$ and $\mathbf{q}^{(t)} \in \mathcal{Q}_0$, we seek new parameters $\rho^{(t+1)}$ and $\boldsymbol{\delta}_v^{(t)}$ such that for

$$\mathbf{q}^{(t+1)} = L_F(\mathbf{q}_0, \sum_{v=1}^{V}\rho_v^{(t+1)}\mathbf{M}_v(Q_v^{(t)} + \boldsymbol{\delta}_v^{(t)})), \tag{7.9}$$

**Algorithm 7** Learning MωMvC$^2$

```
Input: Training set $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$, where $\forall i, \mathbf{x}_i = (x_i^1, \ldots, x_i^V)$ and $y_i \in \{-1, 1\}$; and a maximal number of iterations $T$.
Initialization: $\rho^{(1)} \leftarrow \frac{1}{V}\mathbf{1}_V$ and $\forall v, \mathbf{Q}_v^{(1)} \leftarrow \frac{1}{n_V}\mathbf{1}_{n_v}$
Train the weak classifiers $(\mathcal{H}_v)_{1 \le v \le V}$ over $S$
For $v \in \mathcal{V}$ set the $m \times n_v$ matrix $\mathbf{M}_v$ such that $(\mathbf{M}_v)_{ij} = y_i h_v^j(x_i^v)$
for $t = 1, \ldots, T$ do
    for $i = 1, \ldots, m$ do
        $q_i^{(t)} = \sigma\left(y_i \sum_{v=1}^{V} \rho_v^{(t)} \sum_{j=1}^{n_v} Q_v^{j\,(t)}\ h_v^j(x_i^v)\right)$
    for $v = 1, \ldots, V$ do
        for $j = 1, \ldots, n_v$ do
            $W_{v,j}^{(t)+} = \sum_{i:\mathrm{sign}((\mathbf{M}_v)_{ij})=+1} q_i^{(t)} |(\mathbf{M}_v)_{ij}|$
            $W_{v,j}^{(t)-} = \sum_{i:\mathrm{sign}((\mathbf{M}_v)_{ij})=-1} q_i^{(t)} |(\mathbf{M}_v)_{ij}|$
            $\delta_{v,j}^{(t)} = \frac{1}{2}\ln\left(\frac{W_{v,j}^{(t)+}}{W_{v,j}^{(t)-}}\right)$
        $Q_v^{(t+1)} = Q_v^{(t)} + \boldsymbol{\delta}_v^{(t)}$
    Set $\rho^{(t+1)}$, as the solution of :
        $\min_\rho \quad -\sum_{v=1}^{V} \rho_v \sum_{j=1}^{n_v} \left(\sqrt{W_{v,j}^{(t)+}} - \sqrt{W_{v,j}^{(t)-}}\right)^2$        (7.8)
        $\text{s.t.} \quad \sum_{v=1}^{V} \rho_v = 1, \quad \rho_v \ge 0 \quad \forall v \in \mathcal{V}$
Return: Weights $\rho^{(T)}$ and $\mathbf{Q}^{(T)}$.
```

we get $D_F(0||\mathbf{q}^{(t+1)}) \le D_F(0||\mathbf{q}^{(t)})$.

Following the same strategy as in [19, Theorem 3], it is straightforward to show that in this case, the following inequality holds:

$$D_F(\mathbf{0}||\mathbf{q}^{(t+1)}) - D_F(\mathbf{0}||\mathbf{q}^{(t)}) \le A^{(t)}, \tag{7.10}$$
$$\text{where} \quad A^{(t)} = -\sum_{v=1}^{V} \rho_v^{(t+1)} \sum_{j=1}^{n_v} \left(W_{v,j}^{(t)+}(e^{-\delta_{v,j}^{(t)}} - 1) - W_{v,j}^{(t)-}(e^{\delta_{v,j}^{(t)}} - 1)\right)^2,$$

with $\forall j \in \{1, \ldots, n_v\}$; $W_{v,j}^{(t)\pm} = \sum_{i:\mathrm{sign}((\mathbf{M}_v)_{ij})=\pm1} q_i^{(t)} |(\mathbf{M}_v)_{ij}|$. Note that we provide the proof of Equation 7.10 in Appendix D.

By fixing the set of parameters $\rho^{(t+1)}$; the parameters $\boldsymbol{\delta}_v^{(t)}$ that minimize $A^{(t)}$ are defined as

$\forall v \in \mathcal{V}, \forall j \in 1, \ldots, n_v; \delta_{v,j}^{(t)} = \frac{1}{2} \ln \left( \frac{W_{v,j}^{(t)+}}{W_{v,j}^{(t)-}} \right)$. Plugging back these values into the above equation gives

$$A^{(t)} = -\sum_{v=1}^{V} \rho_v^{(t+1)} \sum_{j=1}^{n_v} \left( \sqrt{W_{v,j}^{(t)+}} - \sqrt{W_{v,j}^{(t)-}} \right)^2. \tag{7.11}$$

Now by fixing the set of parameters $(W_{v,j}^{(t)\pm})_{v,j}$, the weights $\rho^{(t+1)}$ are found by minimizing Equation (7.11) under the linear constraints $\forall v \in \mathcal{V}, \rho_v \geq 0$ and $\sum_{v=1}^{V} \rho_v = 1$. This alternating optimization of $A^{(t)}$ bears similarity with the block-coordinate descent technique [8], where at each iteration, variables are split into two subsets—the set of the active variables, and the set of the inactive ones—and the objective function is minimized along active dimensions while inactive variables are fixed at current values.

**Convergence of Algorithm.** The sequences of weights $(\mathbf{Q}^{(t)})_{t \in \mathbb{N}}$ and $(\rho^{(t)})_{t \in \mathbb{N}}$ found by Algorithm 7 converge to the minimizers of the multiview classification loss (Equation (7.2)), as with the resulting sequence $(\mathbf{q}^{(t)})_{t \in \mathbb{N}}$ (Equation 7.9), the sequence $(D_F(\mathbf{0} || \mathbf{q}^{(t)}))_{t \in \mathbb{N}}$ is decreasing and since it is lower-bounded (Equation (7.6)), it converges to the minimum of Equation (7.2).

## 7.4 Experimental Results

We present below the results of the experiments we have performed to evaluate the efficiency of Algorithm 7 to learn the set of weights $\mathbf{Q}$ and $\rho$ involved in the definition of the multiview weighted majority vote classifier $B_\rho^{\mathrm{MV}}$ (Equation (7.1)). Note that we keep the same datasets as of Chapter 6.

### 7.4.1 Experimental Protocol

In our experiments, we set up binary classification tasks by using all multiview observations from one class as positive examples and all the others as negative examples. We reduced the imbalance between positive and negative examples by subsampling the latter in the training sets, and used decision trees as view specific weak classifiers[1]. We compare our approach to the following seven algorithms.

- `Mono`$_v$ is the best performing decision tree model operating on a single view.

[1] Note that, the experimental protocol in Chapter 6 is different from this one. In Chapter 6, we kept the original distribution of classes whereas in this chapter we reduce the imbalance by subsampling.

- `Concat` is an early fusion approach, where a mono-view decision tree operates over the concatenation of all views of multiview observations.
- `Fusion` is a late fusion approach, sometimes referred to as stacking, where view-specific classifiers are trained independently over different views using 60% of the training examples. A final multiview model is then trained over the predictions of the view-specific classifiers using the rest of the training examples.
- `MVMLsp`[2] [45] is a multiview metric learning approach, where multiview kernels are learned to capture the view-specific information and relation between the views. We kept the experimental setup of [45] with Nyström parameter 0.24.[3]
- `MV-MV` [2] is a multiview algorithm where view-specific classifiers are trained over the views using all the training examples. The final model is the uniformly weighted majority vote (Equation 3.11 in Chapter 3).
- `MVWAB` [90] (Algorithm 3 in Chapter 3) is a Multiview Weighted Voting AdaBoost algorithm, where multiview learning and ababoost techniques are combined to learn a weighted majority vote over view-specific classifiers but without any notion of learning weights over views.
- `rBoost.SH` [66, 67] (Algorithm 4 in Chapter 3) is a multiview boosting approach where a single distribution over different views of training examples is maintained and, the distribution over the views are updated using the multiarmed bandit framework. For the tuning of parameters, we followed the experimental setup of [66].

`Fusion`, `MV-MV`, `MVWAB`, and `rBoost.SH` make decision based on some majority vote strategies, as the proposed M$\omega$MvC$^2$ classifier. The difference relies on how the view-specific classifiers are combined. For `MVWAB` and `rBoost.SH`, we used decision tree model to learn view-specific weak classifiers at each iteration of algorithm and fixed the maximum number of iterations to $T = 100$. To learn M$\omega$MvC$^2$, we generated the matrix $\mathbf{M}_v$ by considering a set of weak decision tree classifiers with different depths (from 1 to $\max_d - 2$, where $\max_d$ is maximum possible depth of a decision tree). We tuned the maximum number of iterations by cross-validation which came out to be $T = 2$ in most of the cases and that we fixed throughout all of the experiments. To solve the optimization problem for finding the weights $\rho$ (Equation 7.8), we used the Sequential Least SQuares Programming (SLSQP) implementation of scikit-learn [65], that we also used to learn the decision trees. Results are computed

[2] We used the Python code available from `https://lives.lif.univ-mrs.fr/?page_id=12`

[3] Note that, based on Nyström parameter, this algorithm uses the part of learning sample while training.

Table 7.1: Test classification accuracy and $F_1$-score of different approaches averaged over all the classes and over 20 random sets of $m = 100$ labeled examples per training set. Along each column, the best result is in bold, and second one in italic. $^{\downarrow}$ indicates that a result is statistically significantly worse than the best result, according to a Wilcoxon rank sum test with $p < 0.02$.

| Strategy | MNIST$_1$ | | MNIST$_2$ | | Reuters | |
|---|---|---|---|---|---|---|
| | Accuracy | $F_1$ | Accuracy | $F_1$ | Accuracy | $F_1$ |
| Mono$_v$ | .8369 ± .002$^{\downarrow}$ | .5206 ± .003$^{\downarrow}$ | .8540 ± .003$^{\downarrow}$ | .5523 ± .004$^{\downarrow}$ | .7651 ± .005$^{\downarrow}$ | .5276 ± .005$^{\downarrow}$ |
| Concat | .8708 ± .005$^{\downarrow}$ | .5851 ± .011$^{\downarrow}$ | .8719 ± .004$^{\downarrow}$ | .5866 ± .010$^{\downarrow}$ | .7661 ± .009$^{\downarrow}$ | .5298 ± .008$^{\downarrow}$ |
| Fusion | .8708 ± .005$^{\downarrow}$ | .5851 ± .010$^{\downarrow}$ | .9029 ± .009$^{\downarrow}$ | .6559 ± .018$^{\downarrow}$ | .7926 ± .013$^{\downarrow}$ | .5533 ± .015$^{\downarrow}$ |
| MVMLsp | .7783 ± .041$^{\downarrow}$ | .4185 ± .051$^{\downarrow}$ | .7766 ± .062$^{\downarrow}$ | .4813 ± .067$^{\downarrow}$ | .6241 ± .032$^{\downarrow}$ | .3488 ± .045$^{\downarrow}$ |
| Aggreg$_L$ | .8956 ± .003$^{\downarrow}$ | .6404 ± .005$^{\downarrow}$ | *.9045* ± .004$^{\downarrow}$ | .6627 ± .009$^{\downarrow}$ | .8179 ± .007$^{\downarrow}$ | .6083 ± .007$^{\downarrow}$ |
| MVWAB | *.9175* ± .003$^{\downarrow}$ | *.7011* ± .009$^{\downarrow}$ | .9038 ± .003$^{\downarrow}$ | *.6838* ± .008$^{\downarrow}$ | .8147 ± .007$^{\downarrow}$ | .6045 ± .009$^{\downarrow}$ |
| rBoost.SH | .7950 ± .006$^{\downarrow}$ | .4652 ± .006$^{\downarrow}$ | .8762 ± .004$^{\downarrow}$ | .6089 ± .007$^{\downarrow}$ | *.8200* ± .007 | *.6164* ± .007$^{\downarrow}$ |
| M$\omega$MvC$^2$ | **.9260** ± .004 | **.7122** ± .010 | **.9169** ± .005 | **.6977** ± .012 | **.8269** ± .013 | **.6280** ± .010 |

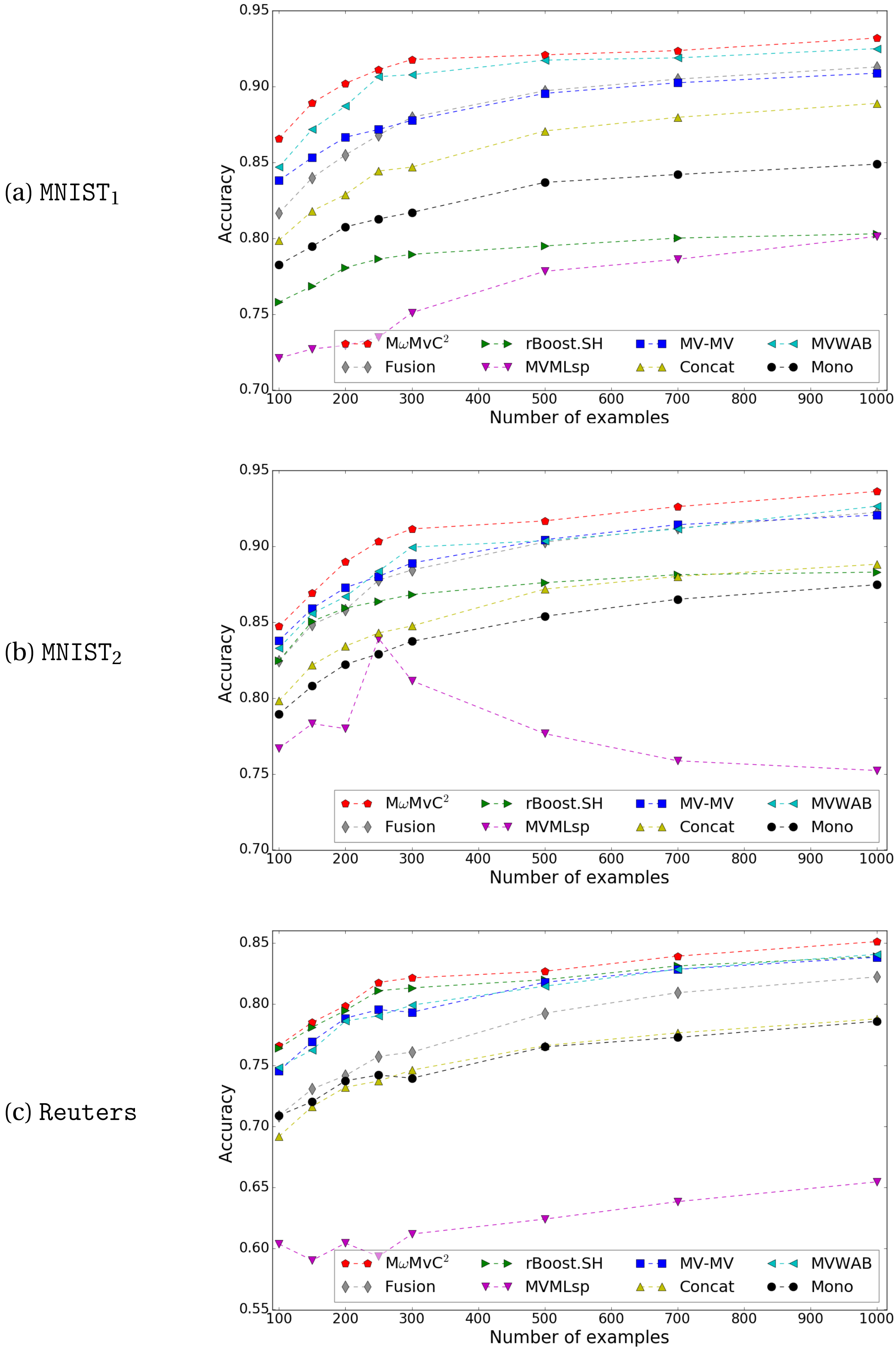
over the test set using the accuracy and the standard $F_1$-score [70], which is the harmonic average of precision and recall. Experiments are repeated 20 times by each time splitting the training and the test sets at random over the initial datasets.

### 7.4.2 Results

Table 7.1 reports the results obtained for $m$=500 training examples by different methods averaged over all classes and the 20 test results obtained over 20 random experiments[4]. From these results it becomes clear that late fusion and other multiview approaches (except MVMLsp) provide consistent improvements over training independent mono-view classifiers and with early fusion, when the size of the training set is small. Furthermore, M$\omega$MvC$^2$ outperforms the other approaches and compared to the second best strategy the gain in accuracy (*resp.* $F_1$-score) varies between 0.8% and 1.3% (*resp.* 1.5% and 2%) across the collections. These results provide evidence that majority voting for multiview learning is an effective way to overcome the lack of labeled information and that all the views do not have the same strength (or do not bring information in the same way) as the learning of weights, as it is done in M$\omega$MvC$^2$, is much more effective than the uniform combination of view-specific classifiers as it is done in MV-MV.

We also analyze the behavior of the algorithms for growing initial amounts of labeled data. Figure 7.2 and Figure 7.3 illustrates this by showing the evolution of the accuracy and the

[4] We also did experiments for Mono$_v$, Concat, Fusion, MV-MV using Adaboost. The performance of Adaboost for these baselines is similar to that of decision trees.

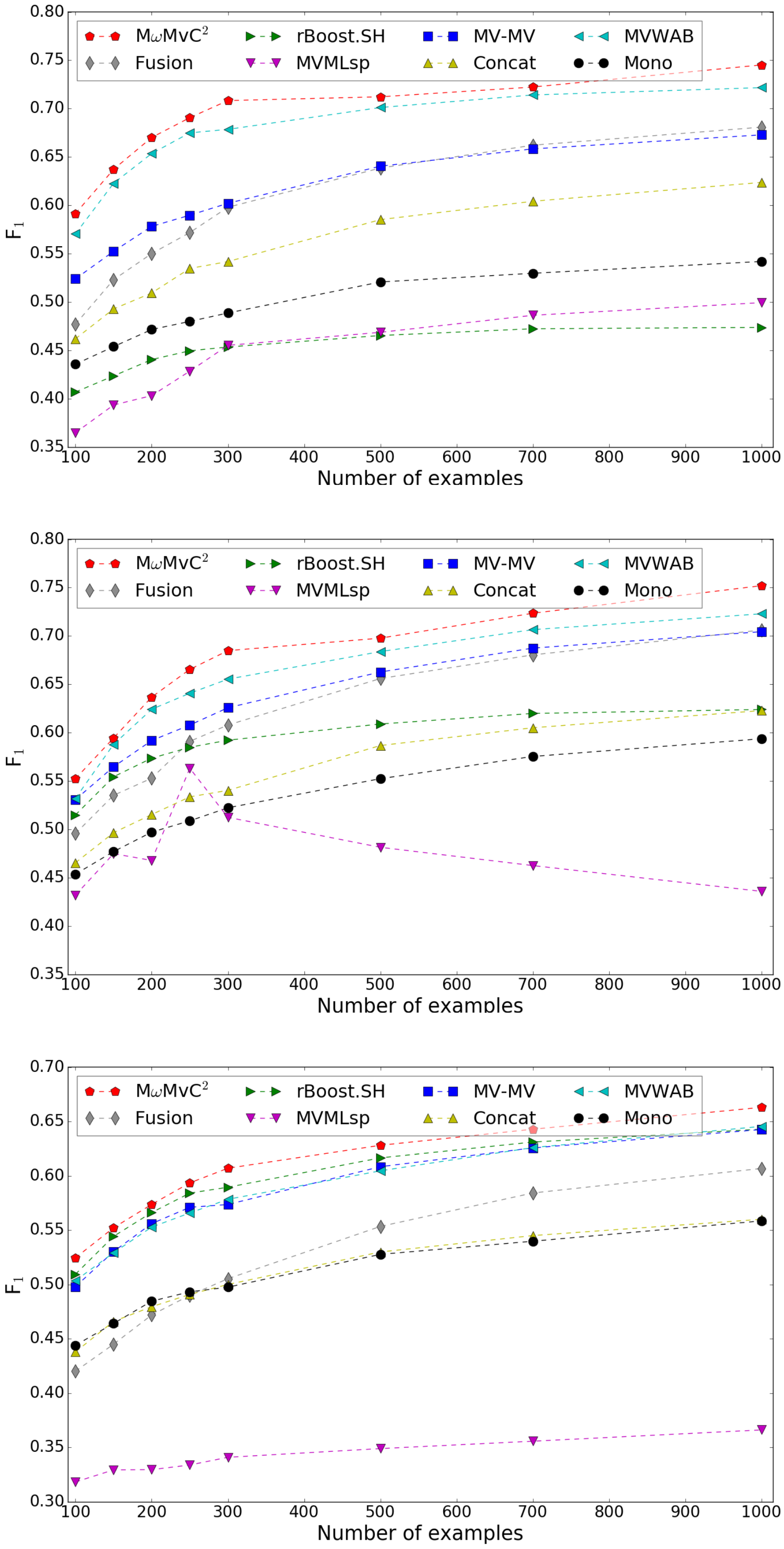


(a) $\mathtt{MNIST}_1$

(b) $\mathtt{MNIST}_2$

(c) `Reuters`

Figure 7.2: Evolution of accuracy *w.r.t* the number of labeled examples in the initial labeled training sets on $\mathtt{MNIST}_1$, $\mathtt{MNIST}_2$ and `Reuters` datasets.



(a) MNIST$_1$

(b) MNIST$_2$

(c) Reuters

Figure 7.3: Evolution of $F_1$-measure *w.r.t* the number of labeled examples in the initial labeled training sets on MNIST$_1$, MNIST$_2$ and Reuters datasets.

$F_1$-score with respect to the number of labeled examples in the initial labeled training sets on $\texttt{MNIST}_1$, $\texttt{MNIST}_2$ and Reuters datasets. As expected, all performance curves increase monotonically *w.r.t* the additional labeled data. When there are sufficient labeled examples, the performance increase of all algorithms actually begins to slow, suggesting that the labeled data carries sufficient information and that the different views do not bring additional information.

An important point here is that rBoost.SH—which takes into account both view-consistency and diversity between views—provides the worst results on $\texttt{MNIST}_1$ where there is no overlapping between the views, while the weighted majority vote as it is performed in $\texttt{M}\omega\texttt{MvC}^2$ still provides an efficient model. Furthermore, MVMLsp—which learns multiview kernels to capture views-specific informations and relation between views—performs worst on all the datasets. We believe that the superior performance of our method stands in our two-level framework. Indeed, thanks to this trick, we are able to consider the view-specific information by learning weights over view-specific classifiers, and to capture the importance of each view in the final ensemble by learning weights over the views.

### 7.4.3 A note on the Complexity of the Algorithm

For each view $v$, the complexity of learning decision tree classifiers is $O(d_v\, mlog(m))$. We learn the weights over the views by optimizing Equation (7.11) (Step 10 of our algorithm) using SLSQP method which has time complexity of $O(V^3)$. Therefore, the overall complexity is $O(V\, d_v\, m.log(m) + TV^3)$. Note that it is easy to parallelize our algorithm: by using $V$ different machines, we can learn the view-specific classifiers and weights over them (Steps 4 to 9).

### 7.4.4 Comparison with $\texttt{Fusion}^{\texttt{all}}_{\texttt{Cq}}$ and PB-MVBoost

In Figure 7.4, we compare $\texttt{M}\omega\texttt{MvC}^2$with $\texttt{Fusion}^{\texttt{all}}_{\texttt{Cq}}$ and PB-MVBoost (proposed in Chapter 6) for $m = 500$ training examples. From Figure 7.4, we can deduce that $\texttt{M}\omega\texttt{MvC}^2$performs worse than $\texttt{Fusion}^{\texttt{all}}_{\texttt{Cq}}$ and PB-MVBoost algorithms. However, computationally $\texttt{M}\omega\texttt{MvC}^2$is faster than $\texttt{Fusion}^{\texttt{all}}_{\texttt{Cq}}$ and PB-MVBoost. As discussed in Chapter 6, time complexity of PB-MVBoost is $O\big(T\,\big(V^3 + V\, d_v\, m.log(m)\big)\big)$, whereas the time complexity for $\texttt{M}\omega\texttt{MvC}^2$is $O(V\, d_v\, m.log(m) + TV^3)$. The reason for better performance of $\texttt{Fusion}^{\texttt{all}}_{\texttt{Cq}}$ and PB-MVBoost is that they control the trade-off between the accuracy and the diversity between the views. The major drawback of $\texttt{M}\omega\texttt{MvC}^2$is that it is unable to handle the unbalanced data whereas PB-MVBoost leads to good performances even when the classes are unbalanced.

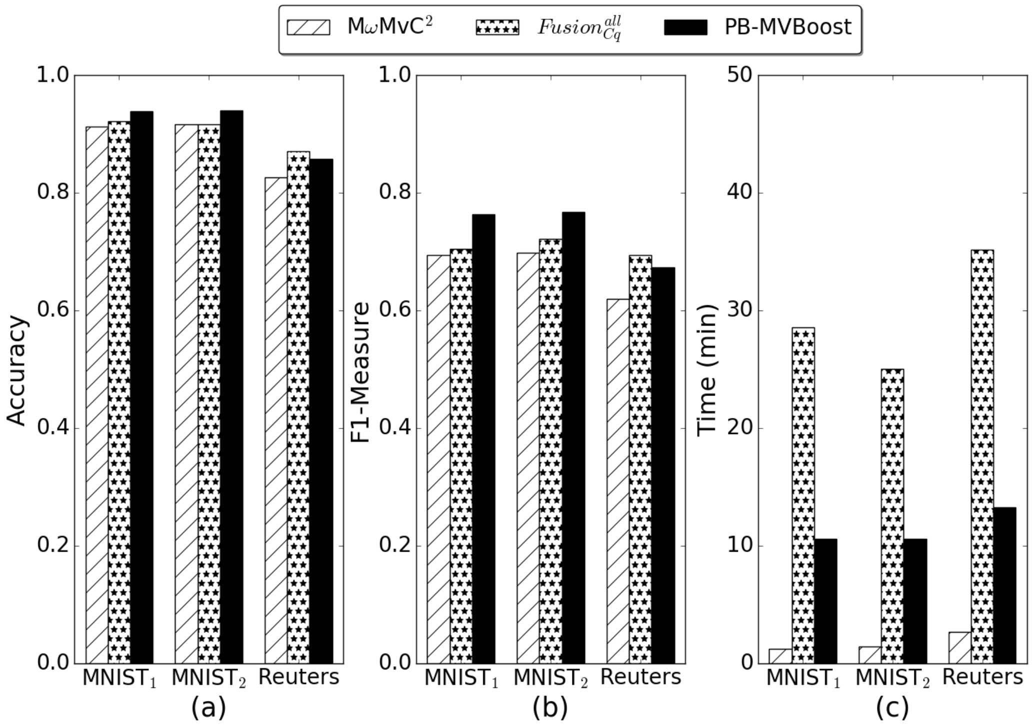


Figure 7.4: Comparison between MωMvC$^2$, $\texttt{Fusion}_{\texttt{Cq}}^{\texttt{all}}$ and PB-MVBoost in terms of (a) Accuracy , (b) F1-Measure and (c) Time Complexity for $m = 500$.

## 7.5 Conclusion

In this chapter, we show that the minimization of the multiview classification error is equivalent to the minimization of Bregman divergences. This embedding allowed us to derive a parallel-update optimization boosting-like algorithm (referred as MωMvC$^2$) in order to learn the weights of over the view-specific classifiers and the views. Our results show clearly that our method allows to reach high performance in terms of accuracy and $F_1$-score on three datasets in the situation where few initial labeled training documents are available. It also comes out that compared to the uniform combination of view-specific classifiers, the learning of weights allows to better capture the strengths of different views. Moreover, we show that this new algorithm is computationally much faster than our previous algorithms ($\texttt{Fusion}_{\texttt{Cq}}^{\texttt{all}}$ and PB-MVBoost) based on PAC-Bayesian theory.

CHAPTER 8

# Conclusion and Perspectives

In this thesis, we have studied the problem of learning the majority vote classifiers for supervised multiview learning where we have multiple representations or views of the input data. We see multiview learning as combination of different view-specific classifiers or views. Therefore, we rely on the PAC-Bayesian theory and the boosting paradigm to derive theoretical guarantees and to design multiview learning algorithms for more general and natural case of more than two views.

The PAC-Bayesian theory provides theoretical guarantees for models that take the form of majority vote over the set of classifiers. The usual PAC-Bayesian generalization bounds are probabilistic bounds which upper bounds (with a high probability on learning sample of size $m$ drawn from distribution $\mathcal{D}$) the true risk of a gibbs classifier in terms of its empirical risk on the training data, the Kullback-Leibler divergence between the posterior and the prior distributions and the size of learning sample. Since posterior distributions are data dependent, our first contribution was to derive a non-probabilistic expected risk bound for the PAC-Bayesian theory in a single view learning setting. This different point of view on PAC-Bayesian analysis has the advantage to involve an expectation over all the posterior distributions that we can learn from a given learning sample size.

Our second contribution was to extend the PAC-Bayesian theory to multiview learning with more than two views. We considered a two-level hierarchy of distributions over the view-specific voters and the views, such that *i)* for each view, we consider a prior and learn a posterior distribution over the view-specific voters, and *ii)* we consider a hyper-prior distribution and learn a hyper-posterior distribution over the views. Based on this strategy, we derived PAC-Bayesian generalization bounds (both probabilistic and expected risk bounds) for multiview learning. Our generalization bounds include a notion of disagreement between all the voters and the views which allowed us to take into account the diversity between them which is known to be a key element in multiview learning. Note that, compared to PAC-Bayesian analysis of Sun et al. [81] we are interested in more natural case of multiview learning with more than two views. Moreover, Amini et al. [2] derived a generalization bounds based on Rademacher complexity for the risk of multiview majority vote over the view-specific classifiers where the distribution over the views is restricted to the uniform distribution (not in our case). Additionally, we derived the generalization bound for the multiview $\mathcal{C}$-bound which we use to design a boosting based algorithm for multiview learning.

From practical point of view, we designed two multiview learning algorithms based on our two-level PAC-Bayesian strategy. The first algorithm is a one-step boosting based multiview learning algorithm called as PB-MVBoost. It iteratively learns the weights over the view-specific classifiers (in order to capture the view-specific informations) and the weights over the views by optimizing the multiview $\mathcal{C}$-Bound which controls the trade-off between the accuracy and the diversity between the views. The second algorithm is based on late fusion approach (referred as $\texttt{Fusion}_{\texttt{Cq}}^{\texttt{all}}$) where we combine the predictions of view-specific classifiers using the PAC-Bayesian algorithm CqBoost [73] which controls the trade-off between the accuracy and the diversity between the view-specific classifiers. We empirically evaluated both of above algorithms on three publicly available datasets $\texttt{MNIST}_1$, $\texttt{MNIST}_2$ and Reuters. We empirically show that the proposed algorithms performs better than considered baseline approaches. We show that PB-MVBoost minimizes the multiview $\mathcal{C}$-Bound over the iterations and able to handle the unbalanced classes. Moreover, we compare PB-MVBoost with $\texttt{Fusion}_{\texttt{Cq}}^{\texttt{all}}$ and show that PB-MVBoost is more stable algorithm across different datasets and computationally faster.

Finally, we show that minimization of classification error for multiview weighted majority vote is equivalent to the minimization of Bregman divergences. This allowed us to derive a parallel-update optimization algorithm (referred as $\texttt{M}\omega\texttt{MvC}^2$) to learn our multiview weighted majority vote. We experimentally evaluated our algorithms on three publicly available datasets and showed that proposed algorithm performs better than the considered baseline approaches. Moreover, M$\omega$MvC$^2$is computationally faster than PB-MVBoost. However, it unable to handle the unbalanced classes.

As future work, we would like to specialize our PAC-Bayesian generalization bounds to linear classifiers for which PAC-Bayesian approaches are known to lead to tight bounds and efficient learning algorithms [34]. This clearly opens the door to derive theoretically founded algorithms for multiview learning. Another perspective is to extend our bounds for diversity-dependent prior similar to the approach used by Sun et al. [81] for more than two views to additionally consider a priori knowledge on the diversity. In addition, we would like to explore our proposed expectation risk bound for PAC-Bayesian theory from algorithmic point of view.

For PB-MVBoost, we fix the number of iterations to $T = 100$. Therefore, we would like to find the suitable stopping criteria for our boosting algorithm. As shown in our experiments (Figure 6.5 in Chapter 6), the F1-measure on test data keeps on increasing even if the classification accuracy and F1-measure on training data reaches to maximal value. Schapire et al. [76] explained this behaviour for boosting methods using margins explanation. They showed that boosting is effective if margins of the training examples keeps on increasing over the iterations. In our case, one of the possible direction to find the suitable stopping criteria is to exploit the margin behaviour for PB-MVBoost. The major drawback of M$\omega$MvC$^2$algorithm is that it is unable to handle the unbalanced data. One possible solution to handle the unbalanced data is to learn the view-specific classifiers (input to our algorithm) such that they take into account original class distributions in the training data.

Another possible direction is to explore *semi-supervised* multiview learning where we have unlabeled data $S_u = \{x_j\}_{j=1}^{m_u}$ along with labeled data $S_l = \{(x_i, y_i)\}_{i=1}^{m_l}$ during training. For our algorithms, one of the possible way is to learn a view-specific classifier using pseudo-labels (for unlabeled data) generated from the classifiers trained from other views, e.g. [29, 93]. For PB-MVBoost, another possible direction is to make use of unlabeled data while computing view-specific disagreement for optimizing multiview $\mathcal{C}$-Bound.

Moreover, the question of extending our work to the case where all the views are not necessarily available or not complete (*missing views* or *incomplete views*, e.g. [2, 92]), is very exciting. For PB-MVBoost, one possible solution is to learn the view-specific voters using available view-specific training examples and adapt the distribution over the learning sample accordingly. For $\mathtt{M}\omega\mathtt{MvC}^2$, one solution could be to adapt the definition of the matrix $\mathbf{M}_v$ to allow to deal with incomplete data; this may be done by considering the notion of diversity to complete $\mathbf{M}_v$.

# List of Publications

## International Journals

Anil Goyal, Emilie Morvant, Pascal Germain, Massih-Reza Amini. *Multiview Boosting by Controlling the Diversity and the Accuracy of View-specific Voters*, Neurocomputing, 2018 (Submitted).

## International Conferences

Anil Goyal, Emilie Morvant, Massih-Reza Amini. *Multiview Learning of Weighted Majority Vote by Bregman Divergence Minimization*, Intelligent Data Analysis (IDA), 2018.

Anil Goyal, Emilie Morvant, Pascal Germain, Massih-Reza Amini. *PAC-Bayesian Analysis for a two-step Hierarchical Mutliview Learning Approach*, European Conference on Machine Learning & Principles and Pratice of Knowledge Discovery in Databases (ECML-PKDD), 2017.

## French Conferences

Anil Goyal, Emilie Morvant, Massih-Reza Amini. *Apprentissage d'un vote de majorité hiérarchique pour l'apprentissage multivue*, Conférence sur l'Apprentissage Automatique (CAp), 2018.

Anil Goyal, Emilie Morvant, Pascal Germain. *Une borne PAC-Bayésienne en espérance et son extension à l'apprentissage multivues*, Conférence sur l'Apprentissage Automatique(CAp), 2017.

Anil Goyal, Emilie Morvant, Pascal Germain, Massih-Reza Amini. *Théorèmes PAC-Bayésiens pour l'apprentissage multivues*, Conférence sur l'Apprentissage Automatique (CAp), 2016.

# Appendix A

# Mathematical Tools

**Theorem A.1** (Markov's ineq.)**.** *For any random variable $X$ s.t.* $\mathbb{E}(|X|) = \mu$, *for any* $a > 0$, *we have* $\mathbb{P}(|X| \geq a) \leq \dfrac{\mu}{a}$.

**Theorem A.2** (Jensen's ineq.)**.** *For any random variable $X$, for any concave function $g$, we have* $g(\mathbb{E}[X]) \geq \mathbb{E}[g(X)]$.

**Theorem A.3** (Cantelli-Chebyshev ineq.)**.** *For any random variable $X$ s.t.* $\mathbb{E}(X) = \mu$ *and* $\mathbf{Var}(X) = \sigma^2$, *and for any* $a > 0$, *we have* $\mathbb{P}(X - \mu \geq a) \leq \frac{\sigma^2}{\sigma^2 + a^2}$.

# Appendix B

# Appendix of Chapter 4

## B.1 Proof of $\mathcal{C}$-bound

In this section, we present the proof of Lemma **??** [35]. Firstly, we need to define the margin of the weighted majority vote $B_Q$ and its first and second statistical moments.

**Definition B.1.** Let $M_Q$ is a random variable that outputs the margin of the weighted majority vote on the example $(x, y)$ drawn from distribution $\mathcal{D}$, given by:

$$M_Q(x, y) = \mathop{\mathbb{E}}_{h \sim Q} y\, h(x).$$

The first and second statistical moments of the margin are respectively given by

$$\mu_1(M_Q^{\mathcal{D}}) = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} M_Q(x, y). \tag{B.1}$$

and,

$$\begin{aligned} \mu_2(M_Q^{\mathcal{D}}) &= \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \big[M_Q(x, y)\big]^2 \\ &= \mathop{\mathbb{E}}_{x\sim\mathcal{D}_X} y^2 \Big[\mathop{\mathbb{E}}_{h\sim Q} h(x^v)\Big]^2 = \mathop{\mathbb{E}}_{x\sim\mathcal{D}_X} \Big[\mathop{\mathbb{E}}_{h\sim Q} h(x)\Big]^2. \end{aligned} \tag{B.2}$$

According to this definition, the risk of the weighted majority vote can be rewritten as follows:

$$R_{\mathcal{D}}(B_Q) = \Pr_{(x,y)\sim\mathcal{D}} \big(M_Q(x, y) \le 0\big).$$

Moreover, the risk of the Gibbs classifier can be expressed thanks to the first statistical moment of the margin. Note that in the binary setting where $y \in \{-1,1\}$ and $h:\mathcal{X}\to\{-1,1\}$, we have $\mathbb{1}_{[h(x)\neq y]} = \frac{1}{2}(1-y\,h(x))$, and therefore

$$\begin{aligned}
R_{\mathcal{D}}(G_Q) &= \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\ \mathop{\mathbb{E}}_{h\sim Q} \mathbb{1}_{[h(x)\neq y]} \\
&= \frac{1}{2}\Big(1-\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\mathop{\mathbb{E}}_{h\sim Q} y\,h(x)\Big) \\
&= \frac{1}{2}(1-\mu_1(M_Q^{\mathcal{D}}))\,.
\end{aligned} \tag{B.3}$$

Similarly, the expected disagreement can be expressed thanks to the second statistical moment of the margin by

$$\begin{aligned}
d_{\mathcal{D}}(Q) &= \mathop{\mathbb{E}}_{x\sim\mathcal{D}_{\mathcal{X}}}\ \mathop{\mathbb{E}}_{h\sim Q}\mathop{\mathbb{E}}_{h'\sim Q} \mathbb{1}_{[h(x)\neq h'(x)]} \\
&= \frac{1}{2}\Big(1-\mathop{\mathbb{E}}_{x\sim\mathcal{D}_{\mathcal{X}}}\ \mathop{\mathbb{E}}_{h\sim Q}\mathop{\mathbb{E}}_{h'\sim Q} h(x)\times h'(x)\Big) \\
&= \frac{1}{2}\Big(1-\mathop{\mathbb{E}}_{x\sim\mathcal{D}_{\mathcal{X}}}\Big[\mathop{\mathbb{E}}_{h\sim Q} h(x)\Big]\times\Big[\mathop{\mathbb{E}}_{h'\sim Q} h'(x)\Big]\Big) \\
&= \frac{1}{2}\Big(1-\mathop{\mathbb{E}}_{x\sim\mathcal{D}_{\mathcal{X}}}\Big[\mathop{\mathbb{E}}_{h\sim Q} h(x)\Big]^2\Big) \\
&= \frac{1}{2}(1-\mu_2(M_Q^{\mathcal{D}}))\,.
\end{aligned} \tag{B.4}$$

From above, we can easily deduce that $0 \leq d_{\mathcal{D}}(Q) \leq 1/2$ as $0 \leq \mu_2(M_Q^{\mathcal{D}}) \leq 1$. Therefore, the variance of the margin can be written as:

$$\begin{aligned}
\mathrm{Var}(M_Q^{\mathcal{D}}) &= \mathop{\mathbf{Var}}_{(x,y)\sim\mathcal{D}}(M_Q(x,y)) \\
&= \mu_2(M_Q^{\mathcal{D}}) - (\mu_1(M_Q^{\mathcal{D}}))^2.
\end{aligned} \tag{B.5}$$

### The proof of the $\mathcal{C}$-bound

***Proof.*** By making use of one-sided Chebyshev inequality (Theorem A.3 of A), with $X = -M_Q(x,y)$, $\mu = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}(M_Q(x,y))$ and $a = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} M_Q(x,y)$, we have

$$
\begin{aligned}
R_{\mathcal{D}}(B_Q) &= \Pr_{(x,y)\sim\mathcal{D}}\Big(M_Q(x,y) \leq 0\Big) \\
&= \Pr_{(x,y)\sim\mathcal{D}}\Big(-M_Q(x,y) + \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} M_Q(x,y) \geq \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} M_Q(x,y)\Big) \\
&\leq \frac{\mathop{\mathbf{Var}}_{(\mathbf{x},y)\sim\mathcal{D}}(M_Q(x,y))}{\mathop{\mathbf{Var}}_{(x,y)\sim\mathcal{D}}(M_Q(x,y)) + \Big(\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} M_Q(x,y)\Big)^2} \\
&= \frac{\mathrm{Var}(M_Q^{\mathcal{D}})}{\mu_2(M_Q^{\mathcal{D}}) - \Big(\mu_1(M_Q^{\mathcal{D}})\Big)^2 + \Big(\mu_1(M_Q^{\mathcal{D}})\Big)^2} \\
&= \frac{\mathrm{Var}(M_Q^{\mathcal{D}})}{\mu_2(M_Q^{\mathcal{D}})} \\
&= \frac{\mu_2(M_Q^{\mathcal{D}}) - \Big(\mu_1(M_Q^{\mathcal{D}})\Big)^2}{\mu_2(M_Q^{\mathcal{D}})} \\
&= 1 - \frac{\Big(\mu_1(M_Q^{\mathcal{D}})\Big)^2}{\mu_2(M_Q^{\mathcal{D}})} \\
&= 1 - \frac{\Big(1 - 2\,R_{\mathcal{D}}(G_Q)\Big)^2}{1 - 2\,d_{\mathcal{D}}(Q)}
\end{aligned}
$$

■

## B.2 Proof of Change of measure inequality

We have

$$
\begin{aligned}
\mathop{\mathbb{E}}_{h\sim Q}\phi(h) &= \mathop{\mathbb{E}}_{h\sim Q}\ln e^{\phi(h)} \\
&= \mathop{\mathbb{E}}_{h\sim Q}\ln\Big(\frac{Q(h)}{P(h)}\frac{P(h)}{Q(h)}e^{\phi(h)}\Big) \\
&= \mathop{\mathbb{E}}_{h\sim Q}\ln\Big(\frac{Q(h)}{P(h)}\Big) + \mathop{\mathbb{E}}_{h\sim Q}\ln\Big(\frac{P(h)}{Q(h)}e^{\phi(h)}\Big).
\end{aligned}
$$

According to the Kullback-Leibler definition, we have

$$\mathop{\mathbb{E}}_{h\sim Q} \phi(h) \;=\; \mathrm{KL}(Q\|P) + \mathop{\mathbb{E}}_{h\sim Q} \ln\Big(\frac{P(h)}{Q(h)} e^{\phi(h)}\Big).$$

By applying Jensen's inequality (Theorem A.2, in Appendix) on the concave function ln, we have

$$\mathop{\mathbb{E}}_{h\sim Q} \phi(h) \;\leq \mathrm{KL}(Q\|P) + \ln\Big(\mathop{\mathbb{E}}_{h\sim P} e^{\phi(h)}\Big)$$

## B.3 Proof of Theorem 4.1

First note that $\mathop{\mathbb{E}}_{h\sim P} e^{mD(R_S(h),R_{\mathcal{D}}(h))}$ is a non-negative random variable. Using Markov's inequality (Theorem A.1 in Appendix A), with $\delta \in (0,1]$, and a probability at least $1-\delta$ over the random choice of the learning sample $S \sim (\mathcal{D})^m$, we have

$$\mathop{\mathbb{E}}_{h\sim P} e^{mD(R_S(h),R_{\mathcal{D}}(h))} \leq \frac{1}{\delta} \mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m} \mathop{\mathbb{E}}_{h\sim P} e^{mD(R_S(h),R_{\mathcal{D}}(h))}$$

By taking the logarithm on both sides, with a probability at least $1-\delta$ over $S \sim (\mathcal{D})^m$, we have

$$\ln\left[\mathop{\mathbb{E}}_{h\sim P} e^{mD(R_S(h),R_{\mathcal{D}}(h))}\right] \leq \ln\left[\frac{1}{\delta} \mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m} \mathop{\mathbb{E}}_{h\sim P} e^{mD(R_S(h),R_{\mathcal{D}}(h))}\right]$$

We now apply Lemma 4.1 on the left-hand side of the above inequality with $\phi(h) = mD\left(R_S(h), R_{\mathcal{D}}(h)\right)$. Therefore, for any $Q$ on $\mathcal{H}$ with a probability at least $1-\delta$ over $S \sim (\mathcal{D})^m$, we have

$$\begin{aligned}\ln\left[\mathop{\mathbb{E}}_{h\sim P} e^{mD(R_S(h),R_{\mathcal{D}}(h))}\right] &\geq m \mathop{\mathbb{E}}_{h\sim Q} D(R_S(h), R_{\mathcal{D}}(h)) - KL(Q||P)\\ &\geq mD(\mathop{\mathbb{E}}_{h\sim Q} R_S(h), \mathop{\mathbb{E}}_{h\sim Q} R_{\mathcal{D}}(h)) - KL(Q||P)\end{aligned}$$

where the last inequality is obtained by applying Jensen's inequality (Theorem A.2 in Appendix A) on the convex function $D$. By rearranging the terms we have

$$D\big(R_S(G_Q), R_{\mathcal{D}}(G_Q)\big) \leq \frac{1}{m}\left[\mathrm{KL}(Q\|P) + \ln\Big(\frac{1}{\delta} \mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m} \mathop{\mathbb{E}}_{h\sim P} e^{mD(R_S(h),R_{\mathcal{D}}(h))}\Big)\right]$$

## B.4 Proof of Square Root Bound

We apply Theorem 4.2 with $D(a,b)=2(a-b)^2$.
Then, we upper-bound $\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m} \mathop{\mathbb{E}}_{h\sim P} e^{mD(R_S(h),R_{\mathcal{D}}(h))}$. By considering $R_S(h)$ as a random variable following a binomial distribution of $m$ trials with a prob. of success $R(h)$, we have

$$\begin{aligned}
\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{h\sim P} e^{m\,D(R_S(h),R_{\mathcal{D}}(h))} &\leq \mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{h\sim P} e^{m\,\mathrm{kl}(R_S(h),R_{\mathcal{D}}(h))}\\
&= \mathop{\mathbb{E}}_{h\sim P}\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\left[\frac{R_S(h)}{R_{\mathcal{D}}(h)}\right]^{mR_S(h)}\left[\frac{1-R_S(h)}{1-R_{\mathcal{D}}(h)}\right]^{m(1-R_S(h))}\\
&= \mathop{\mathbb{E}}_{h\sim P}\sum_{k=0}^{m}\Pr_{S\sim(\mathcal{D})^m}\!\left[R_S(h)=\tfrac{k}{m}\right]\left[\frac{k/m}{R_{\mathcal{D}}(h)}\right]^{k}\left[\frac{1-k/m}{1-R_{\mathcal{D}}(h)}\right]^{m-k}\\
&= \sum_{k=0}^{m}\binom{m}{k}\left[\frac{k}{m}\right]^{k}\left[1-\frac{k}{m}\right]^{m-k} \;\leq\; 2\sqrt{m}.
\end{aligned}$$

## B.5 Proof of Parametrized Bound

The result comes from Theorem 4.2 by taking $D(a,b)=\mathcal{F}-Ca$ , for a convex function $\mathcal{F}$ and $C>0$, and upper-bounding $\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{h\sim P} e^{m\,D(R_S(h),R_{\mathcal{D}}(h))}$. We consider $R_S(h)$ as a random variable following a binomial distribution of $m$ trials with a prob. of success $R(h)$. We have

$$\begin{aligned}
\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{h\sim P} e^{m\,D(R_S(h),R_{\mathcal{D}}(h))} &= \mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{h\sim P} e^{m\mathcal{F}(R_{\mathcal{D}}(h)-C\,m\,R_S(h))}\\
&= \mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{h\sim P} e^{m\mathcal{F}(R_{\mathcal{D}}(h))}\sum_{k=0}^{m}\Pr_{S\sim(\mathcal{D})^m}\left(R_S(h)=\frac{k}{m}\right)e^{-Ck}\\
&= \mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{h\sim P} e^{m\mathcal{F}(R_{\mathcal{D}}(h))}\sum_{k=0}^{m}\tbinom{m}{k}R_{\mathcal{D}}(h)^k(1-R_{\mathcal{D}}(h))^{m-k}e^{-Ck}\\
&= \mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{h\sim P} e^{m\mathcal{F}(R_{\mathcal{D}}(h))}\big(R_{\mathcal{D}}(h)\,e^{-C}+(1-R_{\mathcal{D}}(h))\big)^m.
\end{aligned}$$

The corollary is obtained with

$$\mathcal{F}(p)=\ln\frac{1}{(1-p[1-e^{-C}])}.$$

## B.6 Proof of Small kl Bound

We apply Theorem 4.2 with $D(a,b)\leq \mathrm{kl}(a,b)$.
Then, we upper-bound $\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{h\sim P} e^{m\,D(R_S(h),R_{\mathcal{D}}(h))}$. By considering $R_S(h)$ as a random variable following a binomial distribution of $m$ trials with a prob. of success $R(h)$, we have

$$\begin{aligned}
\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{h\sim P} e^{m\,D(R_S(h),R_{\mathcal{D}}(h))} &\leq \mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{h\sim P} e^{m\,\mathrm{kl}(R_S(h),R_{\mathcal{D}}(h))}\\
&= \mathop{\mathbb{E}}_{h\sim P}\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\left[\frac{R_S(h)}{R_{\mathcal{D}}(h)}\right]^{mR_S(h)}\left[\frac{1-R_S(h)}{1-R_{\mathcal{D}}(h)}\right]^{m(1-R_S(h))}\\
&= \mathop{\mathbb{E}}_{h\sim P}\sum_{k=0}^{m}\Pr_{S\sim(\mathcal{D})^m}\Big[R_S(h)=\tfrac{k}{m}\Big]\left[\frac{k/m}{R_{\mathcal{D}}(h)}\right]^{k}\left[\frac{1-k/m}{1-R_{\mathcal{D}}(h)}\right]^{m-k}\\
&= \sum_{k=0}^{m}\binom{m}{k}\left[\frac{k}{m}\right]^{k}\left[1-\frac{k}{m}\right]^{m-k} \;\leq\; 2\sqrt{m}.
\end{aligned}$$

APPENDIX C

# APPENDIX OF CHAPTER 5

## C.1 Proof of $\mathcal{C}$-Bound for Multiview Learning

In this section, we present the proof of Lemma 5.1, inspired by the proof provided by Germain et al. [35]. Firstly, we need to define the margin of the multiview weighted majority vote $B_\rho$ and its first and second statistical moments.

**Definition C.1.** Let $M_\rho$ is a random variable that outputs the margin of the multiview weighted majority vote on the example $(\mathbf{x}, y)$ drawn from distribution $\mathcal{D}$, given by:

$$M_\rho(\mathbf{x}, y) = \mathop{\mathbb{E}}_{v\sim\rho} \mathop{\mathbb{E}}_{h\sim Q_v} y\, h(x^v).$$

The first and second statistical moments of the margin are respectively given by

$$\mu_1(M_\rho^{\mathcal{D}}) = \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{D}} M_\rho(\mathbf{x}, y). \tag{C.1}$$

and,

$$\begin{aligned}\mu_2(M_\rho^{\mathcal{D}}) &= \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \big[M_\rho(\mathbf{x}, y)\big]^2 \\ &= \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}_X} y^2 \Big[\mathop{\mathbb{E}}_{v\sim\rho} \mathop{\mathbb{E}}_{h\sim Q_v} h(x^v)\Big]^2 = \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}_X} \Big[\mathop{\mathbb{E}}_{v\sim\rho} \mathop{\mathbb{E}}_{h\sim Q_v} h(x^v)\Big]^2. \end{aligned} \tag{C.2}$$

According to this definition, the risk of the multiview weighted majority vote can be rewritten as follows:

$$R_{\mathcal{D}}(B_\rho^{\mathrm{MV}}) = \mathop{\mathbb{P}}_{(\mathbf{x},y)\sim\mathcal{D}} \big(M_\rho(\mathbf{x}, y) \le 0\big).$$

Moreover, the risk of the multiview Gibbs classifier can be expressed thanks to the first statistical moment of the margin. Note that in the binary setting where $y \in \{-1,1\}$ and $h : \mathcal{X} \to \{-1,1\}$, we have $\mathbb{1}_{[h(x^v)\neq y]} = \frac{1}{2}(1 - y\,h(x^v))$, and therefore

$$\begin{aligned}
R_{\mathcal{D}}(G_\rho) &= \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \mathop{\mathbb{E}}_{v\sim\rho} \mathop{\mathbb{E}}_{h\sim Q_v} \mathbb{1}_{[h(x^v)\neq y]} \\
&= \frac{1}{2}\Big(1 - \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \mathop{\mathbb{E}}_{v\sim\rho} \mathop{\mathbb{E}}_{h\sim Q_v} y\,h(x^v)\Big) \\
&= \frac{1}{2}(1 - \mu_1(M_\rho^{\mathcal{D}}))\,.
\end{aligned} \tag{C.3}$$

Similarly, the expected disagreement can be expressed thanks to the second statistical moment of the margin by

$$\begin{aligned}
d_{\mathcal{D}}^{\mathrm{MV}}(\rho) &= \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{X}}} \mathop{\mathbb{E}}_{v\sim\rho} \mathop{\mathbb{E}}_{v'\sim\rho} \mathop{\mathbb{E}}_{h\sim Q_v} \mathop{\mathbb{E}}_{h'\sim Q_{v'}} \mathbb{1}_{[h(x^v)\neq h'(x^{v'})]} \\
&= \frac{1}{2}\Big(1 - \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{X}}} \mathop{\mathbb{E}}_{v\sim\rho} \mathop{\mathbb{E}}_{v'\sim\rho} \mathop{\mathbb{E}}_{h\sim Q_v} \mathop{\mathbb{E}}_{h\sim Q_{v'}} h(x^v)\times h'(x^{v'})\Big) \\
&= \frac{1}{2}\Big(1 - \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{X}}} \Big[\mathop{\mathbb{E}}_{v\sim\rho} \mathop{\mathbb{E}}_{h\sim Q_v} h(x^v)\Big] \times \Big[\mathop{\mathbb{E}}_{v'\sim\rho} \mathop{\mathbb{E}}_{h'\sim Q_{v'}} h'(x^{v'})\Big]\Big) \\
&= \frac{1}{2}\Big(1 - \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{X}}} \Big[\mathop{\mathbb{E}}_{v\sim\rho} \mathop{\mathbb{E}}_{h\sim Q_v} h(x^v)\Big]^2\Big) \\
&= \frac{1}{2}(1 - \mu_2(M_\rho^{\mathcal{D}}))\,.
\end{aligned} \tag{C.4}$$

From above, we can easily deduce that $0 \le d_{\mathcal{D}}^{\mathrm{MV}}(\rho) \le 1/2$ as $0 \le \mu_2(M_\rho^{\mathcal{D}}) \le 1$. Therefore, the variance of the margin can be written as:

$$\begin{aligned}
\mathrm{Var}(M_\rho^{\mathcal{D}}) &= \mathop{\mathbf{Var}}_{(\mathbf{x},y)\sim\mathcal{D}}(M_\rho(\mathbf{x},y)) \\
&= \mu_2(M_\rho^{\mathcal{D}}) - (\mu_1(M_\rho^{\mathcal{D}}))^2.
\end{aligned} \tag{C.5}$$

### The proof of the $\mathcal{C}$-bound

***Proof.*** By making use of one-sided Chebyshev inequality (Theorem A.3 of Appendix A), with $X = -M_\rho(\mathbf{x},y)$, $\mu = \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}}(M_\rho(\mathbf{x},y))$ and $a = \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}} M_\rho(\mathbf{x},y)$, we have

$$\begin{aligned}
R_{\mathcal{D}}(B_\rho) &= \mathop{\mathbb{P}}_{(\mathbf{x},y)\sim\mathcal{D}}\Big(M_\rho(\mathbf{x},y)\le 0\Big)\\
&= \mathop{\mathbb{P}}_{(\mathbf{x},y)\sim\mathcal{D}}\Big(-M_\rho(\mathbf{x},y) + \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}} M_\rho(\mathbf{x},y) \ge \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}} M_\rho(\mathbf{x},y)\Big)\\
&\le \frac{\mathop{\mathbf{Var}}_{(\mathbf{x},y)\sim\mathcal{D}}(M_\rho(\mathbf{x},y))}{\mathop{\mathbf{Var}}_{(\mathbf{x},y)\sim\mathcal{D}}(M_\rho(\mathbf{x},y)) + \Big(\mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}} M_\rho(\mathbf{x},y)\Big)^2}\\
&= \frac{\mathrm{Var}(M_\rho^{\mathcal{D}})}{\mu_2(M_\rho^{\mathcal{D}}) - \Big(\mu_1(M_\rho^{\mathcal{D}})\Big)^2 + \Big(\mu_1(M_\rho^{\mathcal{D}})\Big)^2}\\
&= \frac{\mathrm{Var}(M_\rho^{\mathcal{D}})}{\mu_2(M_\rho^{\mathcal{D}})}\\
&= \frac{\mu_2(M_\rho^{\mathcal{D}}) - \Big(\mu_1(M_\rho^{\mathcal{D}})\Big)^2}{\mu_2(M_\rho^{\mathcal{D}})}\\
&= 1 - \frac{\Big(\mu_1(M_\rho^{\mathcal{D}})\Big)^2}{\mu_2(M_\rho^{\mathcal{D}})}\\
&= 1 - \frac{\Big(1 - 2\,R_{\mathcal{D}}(G_\rho)\Big)^2}{1 - 2\,d_{\mathcal{D}}^{\mathrm{MV}}(\rho)}
\end{aligned}$$

■

## C.2 Proof of Probabilistic Bound for Multiview Learning

First, note that $\mathop{\mathbb{E}}_{v\sim\pi}\mathop{\mathbb{E}}_{h\sim P_v} e^{mD(R_S(h),R_{\mathcal{D}}(h))}$ is a non-negative random variable. Using Markov's inequality (Theorem A.1), with $\delta \in (0,1]$, and a probability at least $1-\delta$ over the random choice of the multiview learning sample $S\sim(\mathcal{D})^m$, we have

$$\mathop{\mathbb{E}}_{v\sim\pi}\mathop{\mathbb{E}}_{h\sim P_v} e^{mD(R_S(h),R_{\mathcal{D}}(h))} \le \frac{1}{\delta}\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{v\sim\pi}\mathop{\mathbb{E}}_{h\sim P_v} e^{mD(R_S(h),R_{\mathcal{D}}(h))}.$$

By taking the logarithm on both sides, with a probability at least $1-\delta$ over $S\sim(\mathcal{D})^m$, we have

$$\ln\Big[\mathop{\mathbb{E}}_{v\sim\pi}\mathop{\mathbb{E}}_{h\sim P_v} e^{mD(R_S(h),R_{\mathcal{D}}(h))}\Big] \le \ln\Big[\frac{1}{\delta}\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{v\sim\pi}\mathop{\mathbb{E}}_{h\sim P_v} e^{mD(R_S(h),R_{\mathcal{D}}(h))}\Big]. \tag{C.6}$$

We now apply Lemma 5.2 on the left-hand side of the Inequality (C.6) with $\phi(h) = m\,D(R_S(h), R_{\mathcal{D}}(h))$. Therefore, for any $Q_v$ on $\mathcal{H}_v$ for all views $v \in \mathcal{V}$, and for any $\rho$ on views $\mathcal{V}$, with a probability at least $1-\delta$ over $S \sim (\mathcal{D})^m$, we have

$$\begin{aligned}
&\ln\Big[\mathop{\mathbb{E}}_{v\sim\pi}\mathop{\mathbb{E}}_{h\sim P_v} e^{m\,D(R_S(h),R_{\mathcal{D}}(h))}\Big]\\
&\geq\; m\mathop{\mathbb{E}}_{v\sim\rho}\mathop{\mathbb{E}}_{h\sim Q_v} D(R_S(h), R_{\mathcal{D}}(h)) - \mathop{\mathbb{E}}_{v\sim\rho}\mathrm{KL}(Q_v\|P_v) - \mathrm{KL}(\rho\|\pi)\\
&\geq\; m\,D\Big(\mathop{\mathbb{E}}_{v\sim\rho}\mathop{\mathbb{E}}_{h\sim Q_v} R_S(h), \mathop{\mathbb{E}}_{v\sim\rho}\mathop{\mathbb{E}}_{h\sim Q_v} R_{\mathcal{D}}(h)\Big) - \mathop{\mathbb{E}}_{v\sim\rho}\mathrm{KL}(Q_v\|P_v) - \mathrm{KL}(\rho\|\pi),
\end{aligned}$$

where the last inequality is obtained by applying Jensen's inequality on the convex function $D$. By rearranging the terms, we have

$$\begin{aligned}
D\Big(\mathop{\mathbb{E}}_{v\sim\rho}\mathop{\mathbb{E}}_{h\sim Q_v} R_S(h), \mathop{\mathbb{E}}_{v\sim\rho}\mathop{\mathbb{E}}_{h\sim Q_v} R_{\mathcal{D}}(h)\Big) \leq \frac{1}{m}\Big[&\mathop{\mathbb{E}}_{v\sim\rho}\mathrm{KL}(Q_v\|P_v) + \mathrm{KL}(\rho\|\pi)\\
&+\ln\Big(\frac{1}{\delta}\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{v\sim\pi}\mathop{\mathbb{E}}_{h\sim P_v} e^{n\,D(R_S(h),R_{\mathcal{D}}(h))}\Big)\Big].
\end{aligned}$$

Finally, the theorem statement is obtained by rewriting

$$\begin{aligned}
\mathop{\mathbb{E}}_{v\sim\rho}\mathop{\mathbb{E}}_{h\sim Q_v} R_S(h) &= R_S(G_\rho^{\mathrm{MV}}),\\
\mathop{\mathbb{E}}_{v\sim\rho}\mathop{\mathbb{E}}_{h\sim Q_v} R_{\mathcal{D}}(h) &= R_{\mathcal{D}}(G_\rho^{\mathrm{MV}}).
\end{aligned}$$

and from Equation (4.3): $R_S(G_\rho^{\mathrm{MV}}) = \frac{1}{2}d_S^{\mathrm{MV}}(\rho) + e_S^{\mathrm{MV}}(\rho)$.

## C.3 Proof of Square Root Bound

To prove the above result, we apply Theorem 5.1 and Theorem 5.2 with $D(a,b) = 2(a-b)^2$. Then, we upper-bound $\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{v\sim\pi}\mathop{\mathbb{E}}_{h_v\sim P_v} e^{m\,D(R_S(h_v),R_{\mathcal{D}}(h_v))}$. According to Pinsker's inequality, we have

$$D(a,b) \;\leq\; \mathrm{kl}(a,b) \;=\; a\ln\frac{a}{b} + (1-a)\ln\frac{1-a}{1-b}.$$

By considering $R_S(h)$ as a random variable which follows a binomial distribution of $m$ trials with a probability of success $R(h)$, we obtain

$$
\begin{aligned}
\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{v\sim\pi}\mathop{\mathbb{E}}_{h_v\sim P_v} e^{mD(R_S(h_v),R_\mathcal{D}(h_v))} &\leq \mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{v\sim\pi}\mathop{\mathbb{E}}_{h_v\sim P_v} e^{m\,\mathrm{kl}(R_S(h_v),R_\mathcal{D}(h_v))}\\
&= \mathop{\mathbb{E}}_{v\sim\pi}\mathop{\mathbb{E}}_{h_v\sim P_v}\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\left[\frac{R_S(h_v)}{R_\mathcal{D}(h_v)}\right]^{mR_S(h_v)}\left[\frac{1-R_S(h_v)}{1-R_\mathcal{D}(h_v)}\right]^{m(1-R_S(h_v))}\\
&= \mathop{\mathbb{E}}_{v\sim\pi}\mathop{\mathbb{E}}_{h_v\sim P_v}\sum_{k=0}^{m}\Pr_{S\sim(\mathcal{D})^m}\left[R_S(h_v)=\tfrac{k}{m}\right]\left[\frac{k/m}{R_\mathcal{D}(h_v)}\right]^k\left[\frac{1-k/m}{1-R_\mathcal{D}(h_v)}\right]^{m-k}\\
&= \sum_{k=0}^{m}\binom{m}{k}\left[\frac{k}{m}\right]^k\left[1-\frac{k}{m}\right]^{m-k}\\
&\leq 2\sqrt{m}.
\end{aligned}
$$

## C.4 Proof of Parametrized Bound

The result comes from Theorem 5.1 and Theorem 5.2 by taking $D(a,b)=\mathcal{F}(b)-Ca$, for a convex $\mathcal{F}$ and $C>0$, and by upper-bounding $\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{v\sim\pi}\mathop{\mathbb{E}}_{h_v\sim P_v} e^{mD(R_S(h_v),R_\mathcal{D}(h_v))}$. We consider $R_S(h_v)$ as a random variable following a binomial distribution of $m$ trials with a probability of success $R(h_v)$. We have:

$$
\begin{aligned}
\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{v\sim\pi}\mathop{\mathbb{E}}_{h_v\sim P_v} e^{mD(R_S(h_v),R_\mathcal{D}(h_v))} &= \mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{v\sim\pi}\mathop{\mathbb{E}}_{h\sim P_v} e^{m\mathcal{F}(R_\mathcal{D}(h_v)-C\,mR_S(h_v))}\\
&= \mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{v\sim\pi}\mathop{\mathbb{E}}_{h_v\sim P_v} e^{m\mathcal{F}(R_\mathcal{D}(h_v))}\sum_{k=0}^{m}\Pr_{S\sim(\mathcal{D})^m}\left(R_S(h_v)=\frac{k}{m}\right)e^{-Ck}\\
&= \mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{v\sim\pi}\mathop{\mathbb{E}}_{h_v\sim P_v} e^{m\mathcal{F}(R_\mathcal{D}(h_v))}\sum_{k=0}^{m}\tbinom{m}{k}R_\mathcal{D}(h_v)^k(1-R_\mathcal{D}(h_v))^{m-k}e^{-Ck}\\
&= \mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{v\sim\pi}\mathop{\mathbb{E}}_{h_v\sim P_v} e^{m\mathcal{F}(R_\mathcal{D}(h_v))}\big(R_\mathcal{D}(h_v)\,e^{-C}+(1-R_\mathcal{D}(h_v))\big)^m.
\end{aligned}
$$

The corollary is obtained with

$$
\mathcal{F}(p)=\ln\frac{1}{(1-p[1-e^{-C}])}.
$$

## C.5 Proof of Small kl Bound

The result follows from Theorem 5.1 and Theorem 5.2 by taking $D(a,b)=\mathrm{kl}(a,b)$, and upper-bounding $\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{v\sim\pi}\mathop{\mathbb{E}}_{h_v\sim P_v} e^{m\,\mathrm{kl}(R_S(h_v),R_\mathcal{D}(h_v))}$. By considering $R_S(h_v)$ as a random variable which follows a binomial distribution of $m$ trials with a probability of success $R(h_v)$, we can prove:

$$\begin{aligned}
\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\mathop{\mathbb{E}}_{v\sim\pi}\mathop{\mathbb{E}}_{h_v\sim P_v} e^{m\,\mathrm{kl}(R_S(h_v),R_{\mathcal{D}}(h_v))} &= \mathop{\mathbb{E}}_{v\sim\pi}\mathop{\mathbb{E}}_{h_v\sim P_v}\mathop{\mathbb{E}}_{S\sim(\mathcal{D})^m}\left[\frac{R_S(h_v)}{R_{\mathcal{D}}(h_v)}\right]^{mR_S(h_v)}\left[\frac{1-R_S(h_v)}{1-R_{\mathcal{D}}(h)}\right]^{m(1-R_S(h_v))}\\
&= \mathop{\mathbb{E}}_{v\sim\pi}\mathop{\mathbb{E}}_{h_v\sim P_v}\sum_{k=0}^{m}\Pr_{S\sim(\mathcal{D})^m}\left(R_S(h_v)=\tfrac{k}{m}\right)\left[\frac{k/m}{R_{\mathcal{D}}(h_v)}\right]^k\left[\frac{1-k/m}{1-R_{\mathcal{D}}(h_v)}\right]^{m-k}\\
&= \sum_{k=0}^{m}\binom{m}{k}\left[\frac{k}{m}\right]^k\left[1-\frac{k}{m}\right]^{m-k}\\
&= \xi(m).
\end{aligned}$$

where $\xi(m)=\sum_{k=0}^{m}\binom{m}{k}\left(\frac{k}{m}\right)^k\left(1-\frac{k}{m}\right)^{m-k}\leq 2\sqrt{m}$.

APPENDIX

# D

# APPENDIX OF CHAPTER 7

## D.1 Proof of Equation 7.10

Firstly, we can show that distribution $\mathbf{q}^{(t+1)}$ is a simple function of previous distribution $\mathbf{q}^{(t)}$

$$\begin{aligned}
\mathbf{q}^{(t+1)} &= L_F\left(\mathbf{q}_0, \sum_{v=1}^{V} \rho_v^{(t+1)} \mathbf{M}_v (Q_v^{(t)} + \boldsymbol{\delta}_v^{(t)})\right) \\
&= L_F\left(\mathbf{q}_0 + \sum_{v=1}^{V} \rho_v^{(t+1)} \mathbf{M}_v Q_v^{(t)}, \sum_{v=1}^{V} \rho_v^{(t+1)} \mathbf{M}_v\, \boldsymbol{\delta}_v^{(t)}\right) \\
&= L_F\left(\mathbf{q}^{(t)}, \sum_{v=1}^{V} \rho_v^{(t+1)} \mathbf{M}_v\, \boldsymbol{\delta}_v^{(t)}\right). \qquad \text{(D.1)}
\end{aligned}$$

From the definition of $D_F(\mathbf{p}||\mathbf{q})$ and $L_F(\mathbf{q}, \mathbf{r})$ given by equations (7.4) and (7.5) respectively, we can show that,

$$D_F(0||L_F(\mathbf{q},\mathbf{r})) - D_F(0||\mathbf{q}) \leq \sum_{i=1}^{m} q_i(e^{-r_i} - 1). \qquad \text{(D.2)}$$

Let $s_v^{ij} = \text{sign}\left((\mathbf{M}_v)_{ij}\right)$ and from Equations (D.1) and (D.2), we have following

$$\begin{aligned}
D_F(\mathbf{0}||\mathbf{q}^{(t+1)}) - D_F(\mathbf{0}||\mathbf{q}^{(t)}) &= D_F\left(\mathbf{0} \,\middle\|\, L_F\left(\mathbf{q}^{(t)}, \sum_{v=1}^{V} \rho_v^{(t+1)} \mathbf{M}_v\, \boldsymbol{\delta}_v^{(t)}\right)\right) - D_F(\mathbf{0}||\mathbf{q}^{(t)}) \\
&\leq \sum_{i=1}^{m} \mathbf{q}_i^{(t)} \left[\exp\left(\sum_{v=1}^{V} \rho_v^{(t+1)} \sum_{j=1}^{n_v} \boldsymbol{\delta}_{v,j}^{(t)} s_v^{ij} \left|(\mathbf{M}_v)_{ij}\right|\right) - 1\right], \\
&\leq \sum_{i=1}^{m} \mathbf{q}_i^{(t)} \left[\sum_{v=1}^{V} \rho_v^{(t+1)} \sum_{j=1}^{n_v} \left|(\mathbf{M}_v)_{ij}\right| \left(e^{-\boldsymbol{\delta}_{v,j}^{(t)} s_v^{ij}} - 1\right)\right], \qquad \text{(D.3)}
\end{aligned}$$

by assuming $\forall j \in \{1,\dots,n_v\};\ W^{(t)\pm}_{v,j} = \sum_{i:\text{sign}((\mathbf{M}_v)_{ij})=\pm 1} q^{(t)}_i |(\mathbf{M}_v)_{ij}|$, we have

$$D_F(\mathbf{0}||\mathbf{q}^{(t+1)}) - D_F(\mathbf{0}||\mathbf{q}^{(t)}) \leq -\sum_{v=1}^{V} \rho_v^{(t+1)} \sum_{j=1}^{n_v} \left( W^{(t)+}_{v,j}\ e^{-\delta^{(t)}_{v,j}} - W^{(t)-}_{v,j}\ e^{\delta^{(t)}_{v,j}} - W^{(t)+}_{v,j} + W^{(t)-}_{v,j} \right)$$

put $\forall v \in \mathcal{V}, \forall j \in 1,\dots,n_v;\ \delta^{(t)}_{v,j} = \frac{1}{2}\ln\left(\frac{W^{(t)+}_{v,j}}{W^{(t)-}_{v,j}}\right)$, we have

$$\begin{aligned} & D_F(\mathbf{0}||\mathbf{q}^{(t+1)}) - D_F(\mathbf{0}||\mathbf{q}^{(t)}) \leq A^{(t)}, \\ \text{where} \quad & A^{(t)} = -\sum_{v=1}^{V} \rho_v^{(t+1)} \sum_{j=1}^{n_v} \left( \sqrt{W^{(t)+}_{v,j}} - \sqrt{W^{(t)-}_{v,j}} \right)^2. \end{aligned}$$

Equation (D.3) uses the fact that, for any $z_v^j$'s, for $\rho_v \geq 0$ with $\sum_v \rho_v \leq 1$ and for $p_v^j \geq 0$ with $\sum_j p_v^j \leq 1$, we have

$$\begin{aligned} \exp\left(\sum_v \rho_v \sum_j p_v^j z_v^j\right) - 1 &= \exp\left(\sum_v \rho_v \sum_j p_v^j z_v^j + 0.\left(1 - \sum_v \rho_v \sum_j p_v^j z_v^j\right)\right) - 1 \\ &\leq \sum_v \rho_v \sum_j p_v^j z_v^j + \left(1 - \sum_v \rho_v \sum_j p_v^j z_v^j\right) - 1 \qquad \text{(D.4)} \\ &= \sum_v \rho_v \sum_j p_v^j \left(e^{z_v^j} - 1\right) \end{aligned}$$

Equation (D.4) is obtained using the Jensen's inequality applied to convex function exponential.

# BIBLIOGRAPHY

[1] S. AKAHO, *A kernel method for canonical correlation analysis*, in In Proceedings of the International Meeting of the Psychometric Society (IMPS2001, Springer-Verlag, 2001.

[2] M.-R. AMINI, N. USUNIER, AND C. GOUTTE, *Learning from Multiple Partially Observed Views - an Application to Multilingual Text Categorization*, in NIPS, 2009, pp. 28–36.

[3] G. ANDREW, R. ARORA, J. BILMES, AND K. LIVESCU, *Deep canonical correlation analysis*, in Proceedings of the 30th International Conference on Machine Learning, S. Dasgupta and D. McAllester, eds., vol. 28 of Proceedings of Machine Learning Research, Atlanta, Georgia, USA, 17–19 Jun 2013, PMLR, pp. 1247–1255.

[4] P. K. ATREY, M. A. HOSSAIN, A. EL-SADDIK, AND M. S. KANKANHALLI, *Multimodal fusion for multimedia analysis: a survey*, Multimedia Syst., 16 (2010), pp. 345–379.

[5] P. AUER, N. CESA-BIANCHI, Y. FREUND, AND R. E. SCHAPIRE, *The nonstochastic multi-armed bandit problem*, SIAM J. Comput., 32 (2003), pp. 48–77.

[6] P. L. BARTLETT AND S. MENDELSON, *Rademacher and gaussian complexities: Risk bounds and structural results*, JMLR, (2002), pp. 463–482.

[7] L. BÉGIN, P. GERMAIN, F. LAVIOLETTE, AND J.-F. ROY, *PAC-Bayesian bounds based on the Rényi divergence*, in AISTATS, 2016, pp. 435–444.

[8] D. P. BERTSEKAS, *Nonlinear Programming*, Athena Scientific, 1999.

[9] A. BLUM AND T. M. MITCHELL, *Combining Labeled and Unlabeled Data with Co-Training*, in COLT, 1998, pp. 92–100.

[10] 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, COLT '92, New York, NY, USA, 1992, ACM, pp. 144–152.

[11] S. BOYD AND L. VANDENBERGHE, *Convex Optimization*, Cambridge University Press, New York, NY, USA, 2004.

[12] U. BREFELD AND T. SCHEFFER, *Co-em support vector learning*, in Proceedings of the twenty-first international conference on Machine learning, ACM, 2004, p. 16.

[13] L. BREGMAN, *The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming*, USSR Computational Mathematics and Mathematical Physics, 7 (1967), pp. 200–217.

[14] O. CATONI, *PAC-Bayesian supervised classification: the thermodynamics of statistical learning*, vol. 56, Inst. of Mathematical Statistic, 2007.

[15] O. CHAPELLE, B. SCHLKOPF, AND A. ZIEN, *Semi-Supervised Learning*, The MIT Press, 1st ed., 2010.

[16] O. CHAPELLE, B. SCHÖLKOPF, AND A. ZIEN, eds., *Semi-Supervised Learning*, MIT Press, Cambridge, MA, 2006.

[17] K. CHAUDHURI, S. M. KAKADE, K. LIVESCU, AND K. SRIDHARAN, *Multi-view clustering via canonical correlation analysis*, in Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09, New York, NY, USA, 2009, ACM, pp. 129–136.

[18] M. CHEN AND L. DENOYER, *Multi-view generative adversarial networks*, in Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18-22, 2017, Proceedings, Part II, 2017, pp. 175–188.

[19] M. COLLINS, R. E. SCHAPIRE, AND Y. SINGER, *Logistic regression, adaboost and bregman distances*, Mach. Learn., 48 (2002), pp. 253–285.

[20] C. CORTES AND V. VAPNIK, *Support-vector networks*, Machine learning, 20 (1995), pp. 273–297.

[21] J. N. DARROCH AND D. RATCLIFF, *Generalized iterative scaling for log-linear models*, in The Annals of Mathematical Statistics, vol. 43, 1972, pp. 1470–1480.

[22] S. DASGUPTA, M. L. LITTMAN, AND D. A. MCALLESTER, *Pac generalization bounds for co-training*, in Advances in Neural Information Processing Systems 14, T. G. Dietterich, S. Becker, and Z. Ghahramani, eds., MIT Press, 2002, pp. 375–382.

[23] S. Della Pietra, V. Della Pietra, and J. Lafferty, *Inducing features of random fields*, IEEE TPAMI, 19 (1997), pp. 380–393.

[24] A. P. Dempster, N. M. Laird, and D. B. Rubin, *Maximum likelihood from incomplete data via the em algorithm*, Journal of the royal statistical society. Series B (methodological), (1977), pp. 1–38.

[25] T. Diethe, D. R. Hardoon, and J. Shawe-taylor, *Multiview fisher discriminant analysis*, in In NIPS Workshop on Learning from Multiple Sources, 2008.

[26] T. G. Dietterich, *Ensemble methods in machine learning*, in Multiple Classifier Systems, 2000, pp. 1–15.

[27] M. D. Donsker and S. S. Varadhan, *Asymptotic evaluation of certain markov process expectations for large time, i*, Communications on Pure and Applied Mathematics, 28 (1975), pp. 1–47.

[28] R. O. Duda and P. E. Hart, *Pattern classification and scene analysis*, A Wiley-Interscience Publication, New York: Wiley, 1973, (1973).

[29] A. Fakeri-Tabrizi, M. Amini, C. Goutte, and N. Usunier, *Multiview self-learning*, Neurocomputing, 155 (2015), pp. 117–127.

[30] J. Farquhar, D. Hardoon, H. Meng, J. S. Shawe-taylor, and S. Szedmák, *Two view learning: Svm-2k, theory and practice*, in NIPS, 2006, pp. 355–362.

[31] Y. Freund, *Boosting a weak learning algorithm by majority*, Inf. Comput., 121 (1995), pp. 256–285.

[32] Y. Freund and R. E. Schapire, *A decision-theoretic generalization of on-line learning and an application to boosting*, J. Comput. Syst. Sci., 55 (1997), pp. 119–139.

[33] N. Friedman, D. Geiger, and M. Goldszmidt, *Bayesian network classifiers*, Machine learning, 29 (1997), pp. 131–163.

[34] P. Germain, A. Lacasse, F. Laviolette, and M. Marchand, *PAC-Bayesian learning of linear classifiers*, in ICML, 2009, pp. 353–360.

[35] P. Germain, A. Lacasse, F. Laviolette, M. Marchand, and J. Roy, *Risk bounds for the majority vote: from a PAC-Bayesian analysis to a learning algorithm*, JMLR, 16 (2015), pp. 787–860.

[36] P. GERMAIN, A. LACASSE, M. MARCHAND, S. SHANIAN, AND F. LAVIOLETTE, *From pac-bayes bounds to kl regularization*, in Advances in Neural Information Processing Systems 22, Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, eds., Curran Associates, Inc., 2009, pp. 603–610.

[37] Z. GHAHRAMANI, *Unsupervised learning*, in Advanced Lectures on Machine Learning, ML Summer Schools 2003, Canberra, Australia, February 2-14, 2003, Tübingen, Germany, August 4-16, 2003, Revised Lectures, 2003, pp. 72–112.

[38] A. GOYAL, E. MORVANT, AND M. AMINI, *Multiview Learning of Weighted Majority Vote by Bregman Divergence Minimization*, in Intelligent Data Analysis, s-Hertogenbosch, the Netherlands, 2018.

[39] A. GOYAL, E. MORVANT, AND M.-R. AMINI, *Apprentissage d'un vote de majorité hiérarchique pour l'apprentissage multivue*, in Conférence Francophone sur l'Apprentissage Automatique (CAp), 2018.

[40] A. GOYAL, E. MORVANT, AND P. GERMAIN, *Une borne pac-bayésienne en espérance et son extension à l'apprentissage multivues*, in Conférence Francophone sur l'Apprentissage Automatique (CAp), 2017.

[41] A. GOYAL, E. MORVANT, P. GERMAIN, AND M. AMINI, *PAC-Bayesian Analysis for a Two-Step Hierarchical Multiview Learning Approach*, in Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18-22, 2017, Proceedings, Part II, 2017, pp. 205–221.

[42] A. GOYAL, E. MORVANT, P. GERMAIN, AND M.-R. AMINI, *Théorèmes pac-bayésiens pour l'apprentissage multi-vues*, in Conférence Francophone sur l'Apprentissage Automatique (CAp), 2016.

[43] L. K. HANSEN AND P. SALAMON, *Neural network ensembles*, IEEE Trans. Pattern Anal. Mach. Intell., 12 (1990), pp. 993–1001.

[44] H. HOTELLING, *Relations Between Two Sets of Variates*, Springer New York, New York, NY, 1992, pp. 162–190.

[45] R. HUUSARI, H. KADRI, AND C. CAPPONI, *Multi-view Metric Learning in Vector-valued Kernel Spaces*, in AISTATS, 2018.

[46] J.-C. JANODET, M. SEBBAN, AND H.-M. SUCHIER, *Boosting Classifiers built from Different Subsets of Features*, Fundamenta Informaticae, 94 (2009), pp. 1–21.

[47] J. R. KETTENRING, *Canonical analysis of several sets of variables,* 1969.

[48] S. KOÇO AND C. CAPPONI, *A boosting approach to multiview classification with cooperation,* in Machine Learning and Knowledge Discovery in Databases, D. Gunopulos, T. Hofmann, D. Malerba, and M. Vazirgiannis, eds., Berlin, Heidelberg, 2011, Springer Berlin Heidelberg, pp. 209–228.

[49] V. KOLTCHINSKII AND D. PANCHENKO, *Rademacher processes and bounding the risk of function learning,* in High Dimensional Probability II, E. Giné, D. M. Mason, and J. A. Wellner, eds., Boston, MA, 2000, Birkhäuser Boston, pp. 443–457.

[50] L. I. KUNCHEVA, *Combining Pattern Classifiers: Methods and Algorithms,* Wiley-Interscience, 2004.

[51] A. LACASSE, F. LAVIOLETTE, M. MARCHAND, P. GERMAIN, AND N. USUNIER, *PAC-Bayes bounds for the risk of the majority vote and the variance of the Gibbs classifier,* in NIPS, 2006, pp. 769–776.

[52] J. LAFFERTY, *Additive models, boosting, and inference for generalized divergences,* in COLT, 1999, pp. 125–133.

[53] J. LANGFORD, *Tutorial on practical prediction theory for classification,* JMLR, 6 (2005), pp. 273–306.

[54] J. LANGFORD AND J. SHAWE-TAYLOR, *PAC-Bayes & margins,* in NIPS, MIT Press, 2002, pp. 423–430.

[55] Y. LECUN, L. BOTTOU, Y. BENGIO, AND P. HAFFNER, *Gradient-based learning applied to document recognition,* in Proceedings of the IEEE, 1998, pp. 2278–2324.

[56] O. MAILLARD AND N. VAYATIS, *Complexity versus agreement for many views,* in ALT, 2009, pp. 232–246.

[57] D. A. MCALLESTER, *Some PAC-Bayesian theorems,* Machine Learning, 37 (1999), pp. 355–363.

[58] D. A. MCALLESTER, *PAC-Bayesian stochastic model selection,* in Machine Learning, 2003, pp. 5–21.

[59] M. MOHRI, A. ROSTAMIZADEH, AND A. TALWALKAR, *Foundations of Machine Learning,* Adaptive computation and machine learning, MIT Press, 2012.

[60] E. MORVANT, *Domain adaptation of weighted majority votes via perturbed variation-based self-labeling*, Pattern Recognition Letters, 51 (2015), pp. 37–43.

[61] E. MORVANT, A. HABRARD, AND S. AYACHE, *Majority vote of diverse classifiers for late fusion*, in Structural, Syntactic, and Statistical Pattern Recognition - Joint IAPR International Workshop, S+SSPR 2014, Joensuu, Finland, August 20-22, 2014. Proceedings, 2014, pp. 153–162.

[62] K. NIGAM AND R. GHANI, *Analyzing the effectiveness and applicability of co-training*, in Proceedings of the ninth international conference on Information and knowledge management, ACM, 2000, pp. 86–93.

[63] P. L. NUNEZ AND R. B. SILBERSTEIN, *On the relationship of synaptic activity to macroscopic measurements: Does co-registration of eeg with fmri make sense?*, Brain Topography, 13 (2000), pp. 79–96.

[64] S. J. PAN AND Q. YANG, *A survey on transfer learning*, IEEE Trans. Knowl. Data Eng., 22 (2010), pp. 1345–1359.

[65] F. PEDREGOSA, G. VAROQUAUX, A. GRAMFORT, V. MICHEL, B. THIRION, O. GRISEL, M. BLONDEL, P. PRETTENHOFER, R. WEISS, V. DUBOURG, J. VANDERPLAS, A. PASSOS, D. COURNAPEAU, M. BRUCHER, M. PERROT, AND E. DUCHESNAY, *Scikit-learn: Machine learning in Python*, Journal of Machine Learning Research, 12 (2011), pp. 2825–2830.

[66] J. PENG, A. J. AVED, G. SEETHARAMAN, AND K. PALANIAPPAN, *Multiview boosting with information propagation for classification*, IEEE Transactions on Neural Networks and Learning Systems, PP (2017), pp. 1–13.

[67] J. PENG, C. BARBU, G. SEETHARAMAN, W. FAN, X. WU, AND K. PALANIAPPAN, *Shareboost: Boosting for multi-view learning with performance guarantees*, in Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011, Proceedings, Part II, 2011, pp. 597–612.

[68] A. PENTINA AND C. H. LAMPERT, *A PAC-Bayesian bound for lifelong learning*, in ICML, 2014, pp. 991–999.

[69] T. POGGIO AND F. GIROSI, *Regularization algorithms for learning that are equivalent to multilayer networks*, Science, 247 (1990), pp. 978–982.

[70] D. M. POWERS, *Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation*, J. of Machine Learning Technologies, 1 (2011), pp. 37—-63.

[71] M. RE AND G. VALENTINI, *Ensemble methods: a review*, Advances in machine learning and data mining for astronomy, (2012), pp. 563–582.

[72] J. ROY, F. LAVIOLETTE, AND M. MARCHAND, *From pac-bayes bounds to quadratic programs for majority votes*, in Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011, 2011, pp. 649–656.

[73] J.-F. ROY, M. MARCHAND, AND F. LAVIOLETTE, *A column generation bound minimization approach with PAC-Bayesian generalization guarantees*, in Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, 2016, pp. 1241–1249.

[74] R. E. SCHAPIRE, *A brief introduction to boosting*, in Proceedings of the 16th International Joint Conference on Artificial Intelligence - Volume 2, IJCAI'99, San Francisco, CA, USA, 1999, Morgan Kaufmann Publishers Inc., pp. 1401–1406.

[75] R. E. SCHAPIRE, *The Boosting Approach to Machine Learning: An Overview*, Springer New York, New York, NY, 2003.

[76] R. E. SCHAPIRE, Y. FREUND, P. BARTLETT, W. S. LEE, ET AL., *Boosting the margin: A new explanation for the effectiveness of voting methods*, The annals of statistics, 26 (1998), pp. 1651–1686.

[77] M. W. SEEGER, *PAC-Bayesian generalisation error bounds for gaussian process classification*, JMLR, 3 (2002), pp. 233–269.

[78] V. SINDHWANI, P. NIYOGI, AND M. BELKIN, *A co-regularized approach to semi-supervised learning with multiple views*, in Proceedings of the ICML Workshop on Learning with Multiple Views, 2005.

[79] C. SNOEK, M. WORRING, AND A. W. M. SMEULDERS, *Early versus late fusion in semantic video analysis*, in ACM Multimedia, 2005, pp. 399–402.

[80] S. SUN, *A survey of multi-view machine learning*, Neural Comput Appl, 23 (2013), pp. 2031–2038.

[81] S. SUN, J. SHAWE-TAYLOR, AND L. MAO, *PAC-Bayes analysis of multi-view learning*, Information Fusion, 35 (2017), pp. 117–131.

[82] R. S. SUTTON AND A. G. BARTO, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA, USA, 1998.

[83] L. G. VALIANT, *A theory of the learnable*, Commun. ACM, 27 (1984), pp. 1134–1142.

[84] M. VAN BREUKELEN, R. P. DUIN, D. M. TAX, AND J. DEN HARTOG, *Handwritten digit recognition by combined classifiers*, Kybernetika, 34 (1998), pp. 381–386.

[85] V. VAPNIK, *Principles of risk minimization for learning theory*, in Advances in Neural Information Processing Systems 4, J. E. Moody, S. J. Hanson, and R. P. Lippmann, eds., Morgan-Kaufmann, 1992, pp. 831–838.

[86] V. N. VAPNIK, *Statistical Learning Theory*, Wiley-Interscience, 1998.

[87] V. N. VAPNIK, *The Nature of Statistical Learning Theory*, Springer, 1999.

[88] V. N. VAPNIK AND A. Y. CHERVONENKIS, *On the uniform convergence of relative frequencies of events to their probabilities*, Theory of Probability and its Applications, 16 (1971), pp. 264–280.

[89] D. H. WOLPERT, *Stacked generalization*, Neural Networks, 5 (1992), pp. 241–259.

[90] M. XIAO AND Y. GUO, *Multi-view adaboost for multilingual subjectivity analysis*, in COLING 2012, 24th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, 8-15 December 2012, Mumbai, India, 2012, pp. 2851–2866.

[91] C. XU, D. TAO, AND C. XU, *A survey on multi-view learning*, CoRR, abs/1304.5634 (2013).

[92] C. XU, D. TAO, AND C. XU, *Multi-view learning with incomplete views*, IEEE Transactions on Image Processing, 24 (2015), pp. 5812–5825.

[93] X. XU, W. LI, D. XU, AND I. W. TSANG, *Co-labeling for multi-view weakly labeled learning*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 38 (2016), pp. 1113–1125.

[94] Z. XU AND S. SUN, *An algorithm on multi-view adaboost*, in Neural Information Processing. Theory and Algorithms, K. W. Wong, B. S. U. Mendis, and A. Bouzerdoum, eds., Berlin, Heidelberg, 2010, Springer Berlin Heidelberg, pp. 355–362.

[95] J. ZHANG AND D. ZHANG, *A novel ensemble construction method for multi-view data using random cross-view correlation between within-class examples*, Pattern. Recogn., 44 (2011), pp. 1162–1171.