Multiview Boosting by Controlling the Diversity and the Accuracy of View-specific Voters
Anil Goyal, Emilie Morvant, Pascal Germain, Massih-Reza Amini

To cite this version:

HAL Id: hal-01857463
https://hal.archives-ouvertes.fr/hal-01857463v4
Submitted on 26 May 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.
Multiview Boosting by Controlling the Diversity and the Accuracy of View-specific Voters

Anil Goyal\textsuperscript{1,2} Emilie Morvant\textsuperscript{1} Pascal Germain\textsuperscript{3} Massih-Reza Amini\textsuperscript{2}

\textsuperscript{1} Univ Lyon, UJM-Saint-Etienne, CNRS, Institut d’Optique Graduate School, Laboratoire Hubert Curien UMR 5516, F-42023, Saint-Etienne, France
\textsuperscript{2} Univ. Grenoble Alps, Laboratoire d’Informatique de Grenoble, AMA, Centre Equation 4, BP 53, F-38041 Grenoble Cedex 9, France
\textsuperscript{3} Inria Lille - Nord Europe, Modal Project-Team, 59650 Villeneuve d’Ascq, France

May 26, 2019

Abstract

In this paper we propose a boosting based multiview learning algorithm, referred as PB-MVBoost, which iteratively learns i) weights over view-specific voters capturing view-specific information, and ii) weights over views by optimizing a PAC-Bayes multiview C-Bound that takes into account the accuracy of view-specific classifiers and the diversity between the views. We derive a generalization bound for this strategy following the PAC-Bayes theory which is a suitable tool to deal with models expressed as weighted combination over a set of voters. Different experiments on three publicly available datasets show the efficiency of the proposed approach with respect to state-of-art models.

1 Introduction

With the tremendous generation of data, there are more and more situations where observations are described by more than one view. This is for example the case with multilingual documents that convey the same information in different languages or images that are naturally described according to different set of features (for example SIFT, HOG, CNN, etc). In this paper, we study the related machine learning problem that consists in finding an efficient classification model from different information sources that describe the observations. This topic, called multiview (or multimodal) learning[Atrey et al. [2010], Sun [2013], Baltrušaitis et al. [2019], has been expanding over the past decade, spurred by the seminal work of Blum and Mitchell [1998] on co-training (with only two views). The aim is to learn a classifier which performs better than classifiers trained over each view separately (called here view-specific classifier). Usually, this is done by directly concatenating the representations (early fusion) or by combining the predictions of view-specific classifiers (late fusion) Snoek et al. [2005]. In this work, we stand in the latter situation. Concretely, we study a two-level multiview learning strategy based on the PAC-Bayesian theory (introduced by McAllester [1999] for monoview learning). This theory provides Probably Approximately Correct (PAC) generalization guarantees for models expressed as a weighted combination over a set of functions/voters (i.e., for a weighted majority vote). In this framework, given a prior distribution over a set of functions, called voters, \( \mathcal{H} \) and a learning sample, one aims at learning a posterior distribution over \( \mathcal{H} \) leading to a well-performing majority vote; each voter from \( \mathcal{H} \) is weighted by its probability to appear according to the posterior distribution. Note that, PAC-Bayesian studies have not only been conducted to characterize the error of such weighted majority votes Catoni [2007], Seeger [2002], Langford and Shawe-Taylor [2002], Germain et al. [2009, 2015], but have also been used to derive theoretically grounded learning algorithms (such as for supervised learning Germain et al. [2009], Parrado-Hernández et al. [2012], Alquier et al. [2015], Roy et al. [2016], Morvant et al. [2014] or transfer learning Germain et al. [2016]). To tackle multiview learning in a PAC-Bayesian fashion, we propose to define a two-level hierarchy of prior and posterior distributions over the views: i) for each view \( v \), we consider a prior \( P_v \) and a posterior \( Q_v \) distributions over view-specific voters to capture view-specific information 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.
Figure 1: Illustration of the multiview distributions hierarchy with 3 views. For all views $v \in \{1, 2, 3\}$, we have a set of voters $\mathcal{H}_v = \{h_{v1}^1, \ldots, h_{vn}^v\}$ on which we consider prior $P_v$ view-specific distribution (in blue), and we consider a hyper-prior $\pi$ distribution (in green) over the set of 3 views. The objective is to learn a posterior $Q_v$ (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.

between the views (see Figure 1). Following this distributions’ hierarchy, we define a multiview majority vote classifier where the view-specific classifiers are weighted according to posterior and hyper-posterior distributions. By doing so, we extend the classical PAC-Bayesian theory to multiview learning with more than two views and derive a PAC-Bayesian generalization bound for our multiview majority vote classifier.

From a practical point of view, we design an algorithm based on the idea of boosting Freund [1995], Freund and Schapire [1997], Schapire [1999, 2003], an ensemble method well known to be able to learn well-performing majority vote. Our boosting-based multiview learning algorithm, called PB-MVBoost, deals with the two-level hierarchical learning strategy. PB-MVBoost is then an ensemble method that outputs a multiview classifier expressed as a weighted combination of view-specific voters. It is important to notice that controlling the diversity between the view-specific classifiers or the views is a key element in multiview learning Amini et al. [2009], Goyal et al. [2017], Chapelle et al. [2010], Kuncheva [2004], Maillard and Vayatis [2009], Morvant et al. [2014]. Therefore, to learn the weights over the views, we minimize an upper-bound on the error of the majority vote, called the multiview C-bound Germain et al. [2015], Roy et al. [2016], Goyal et al. [2017], 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 information), and ii) weights over views by minimizing the multiview C-bound. To show the potential of our algorithm, we empirically evaluate our approach on MNIST, MNIST2 and Reuters RCV1/RCV2 collections Lecun et al. [1998], Amini et al. [2009]. We observe that our algorithm PB-MVBoost, empirically minimizes the multiview C-Bound over iterations, and leads to good performances even when the classes are unbalanced. We compare PB-MVBoost with a previously developed multiview algorithm, denoted by FusionCq Goyal et al. [2017], which first learns the view-specific voters at the base level of the hierarchy, and then, combines the predictions of view-specific voters using a PAC-Bayesian algorithm CqBoost Roy et al. [2016]. From the experimental results, it came out that PB-MVBoost is more stable across different datasets and computationally faster than FusionCq.

In the next section, we discuss some related works. In Section 3 we present the PAC-Bayesian multiview learning framework Goyal et al. [2017]. In Section 4 we derive our multiview learning algorithm PB-MVBoost. Before concluding in Section 6 we experiment our algorithm in Section 5.

2 Related Work

Learning a weighted majority vote is closely related to ensemble methods Dietterich [2000], Re and Valentini [2012]. In the ensemble methods literature, it is well known that we desire to combine voters that make errors on different data points Kuncheva [2004]. Intuitively, this means that the voters disagree on some data points. This notion of disagreement (or agreement) is sometimes called diversity between
classifiers [Diez-Pastor et al., 2015, Brown and Kuncheva, 2010, Kuncheva, 2004]. Even if there is no consensus on the definition of “diversity”, controlling it while keeping good accuracy is at the heart of a majority of ensemble methods: indeed if all the voters agree on all the points then there is no interest to combine them, only one will be sufficient. Similarly, when we combine multiple views (or representations), it is known that controlling diversity between the views plays a vital role for learning the final majority vote [Amini et al., 2009, Goyal et al., 2017, Chapelle et al., 2010, Maillard and Vayatis, 2009]. Most of the existing ensemble-based multiview learning algorithms try to exploit either view consistency (agreement between views) [Janodet et al., 2009, Koço and Capponi, 2011, Xiao and Guo, 2012] or diversity between views [Xu and Sun, 2010, Goyal et al., 2017, Peng et al., 2011, 2017] in different manners. Janodet et al. [2009] proposed a boosting-based multiview learning algorithm for two 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 and Capponi [2011] proposed Munbo that maintains separate distributions for each view. For each view, the algorithm reduces the weights associated with the examples hard to classify, and increases the weights of those examples in the other views. This trick allows a communication between the views with the objective to maintain view consistency. Compared to our approach, we follow a two-level learning strategy where we learn (hyper-)posterior distributions/weights over view-specific voters and views. In order to take into account accuracy and diversity between the views, we optimize the multiview C-Bound (an upper-bound over the risk of multiview majority vote learned, see e.g. Germain et al., 2015, Roy et al., 2016, Goyal et al., 2017).

Xu and Sun [2010] 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. Peng et al. [2011, 2017] proposed variants of Boost.SH (boosting with SHared weight distribution) which controls the diversity for more than two views. Similarly than our approach, they maintain a single global distribution over the learning examples for all the views. 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. At the end, their multiview majority vote is a combination of T weighted base voters, where T is the number of iterations for boosting. Whereas, our multiview majority vote is a weighted combination of the view-specific voters over all the weighted views.

Furthermore, our approach encompasses the one of [Amini et al., 2009] and Xiao and Guo [2012]. Amini et al. [2009] proposed a Rademacher analysis for expectation of individual risks of each view-specific classifier (for more than two views). Xiao and Guo [2012] derived a weighted majority voting Adaboost algorithm 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, it is important to note that Sun et al. [2017] 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. 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 without any restrictions 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.

3 The Multiview PAC-Bayesian Framework

3.1 Notations and Setting

In this work, we tackle multiview binary classification tasks where the observations 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 \( \forall v \in \mathcal{V} \), \( \mathcal{X}_v \subseteq \mathbb{R}^{d_v} \) is a \( d_v \)-dimensional input space, and the binary output space is \( \mathcal{Y} = \{-1, +1\} \). We assume that \( \mathcal{D} \) is a fixed but unknown distribution over \( \mathcal{X} \times \mathcal{Y} \). We stand in the PAC-Bayesian supervised learning setting where an observation \( x = (x^1, x^2, \ldots, x^V) \in \mathcal{X} \) is given with its label \( y \in \mathcal{Y} \), and is independently and identically drawn (i.i.d.) from \( \mathcal{D} \). A learning algorithm is then provided with a training sample \( S \) of \( n \) examples i.i.d. from \( \mathcal{D} \): \( S = \{(x_i, y_i)\}_{i=1}^n \sim (\mathcal{D})^n \), where \((\mathcal{D})^n\) stands for the distribution of a \( n \)-sample. For each view \( v \in \mathcal{V} \), we consider a view-specific set \( \mathcal{H}_v \) of voters \( h : \mathcal{X}_v \to \mathcal{Y} \), 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 \( S \), our PAC-Bayesian learner
objective is twofold: \(i\) finding a posterior distribution \(Q_v\) over \(\mathcal{H}_v\) for all views \(v \in \mathcal{V}\), and \(ii\) finding a hyper-posterior distribution \(\rho\) on the set of the views \(\mathcal{V}\). This defines a hierarchy of distributions illustrated on Figure 1. The learned distributions express a multiview weighted majority vote\(^1\) defined as

\[ B_\rho(x) = \text{sign} \left( \mathbb{E}_{v \sim \rho} \mathbb{E}_{h \sim Q_v} h(x^v) \right). \tag{1} \]

Thus, the learner aims at constructing the posterior and hyper-posterior distributions that minimize the true risk \(R_D(B_\rho)\) of the multiview weighted majority vote

\[ R_D(B_\rho) = \mathbb{E}_{(x,y) \sim D} \mathbb{I}_{[B_\rho(x) \neq y]}, \]

where \(\mathbb{I}_{[\pi]} = 1\) if the predicate \(\pi\) is true and 0 otherwise. The above risk of the deterministic weighted majority vote is closely related to the Gibbs risk \(R_D(G_\rho)\) defined as the expectation of the individual risks of each voter that appears in the majority vote. More formally, in our multiview setting, we have

\[ R_D(G_\rho) = \mathbb{E}_{(x,y) \sim D} \mathbb{E}_{v \sim \rho} \mathbb{E}_{h \sim Q_v} \mathbb{I}_{[h(x^v) \neq y]}, \]

and its empirical counterpart is

\[ R_S(G_\rho) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{v \sim \rho} \mathbb{E}_{h \sim Q_v} \mathbb{I}_{[h(x^v_i) \neq y_i]}. \]

In fact, if \(B_\rho\) misclassifies \(x \in \mathcal{X}\), then at least half of the view-specific voters from all the views (according to hyper-posterior and posterior distributions) makes an error on \(x\). Then, it is well known that \(R_D(B_\rho)\) is upper-bounded by twice \(R_D(G_\rho)\):

\[ R_D(B_\rho) \leq 2R_D(G_\rho). \]

In consequence, a generalization bound for \(R_D(G_\rho)\) gives rise to a generalization bound for \(R_D(B_\rho)\).

There exist tighter relations, such as the C-Bound\(^2\) which captures a trade-off between the Gibbs risk \(R_D(G_\rho)\) and the disagreement between pairs of voters. This latter can be seen as a measure of diversity among the voters involved in the majority vote, that is a key element to control from a multiview point of view.\(^3\) The C-Bound can be extended to our multiview setting as below.

**Lemma 1 (Multiview C-Bound)** 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 \(\rho\) distribution over views \(V\), if \(R_D(G_\rho) < \frac{1}{2}\), then we have

\[ R_D(B_\rho) \leq 1 - \frac{(1 - 2R_D(G_\rho))^2}{1 - 2d_D(\rho)}, \tag{2} \]

\[ \leq 1 - \frac{(1 - 2\mathbb{E}_{v \sim \rho} R_D(G_{Q_v}))^2}{1 - 2\mathbb{E}_{v \sim \rho} d_D(Q_v)}, \tag{3} \]

where \(d_D(\rho)\) is the expected disagreement between pairs of voters defined as

\[ d_D(\rho) = \mathbb{E}_{x \sim D} \mathbb{E}_{v \sim \rho} \mathbb{E}_{h \sim Q_v} \mathbb{E}_{h' \sim Q_v} \mathbb{I}_{[h(x^v) \neq h'(x^v)]}, \]

and \(R_D(G_{Q_v})\) and \(d_D(Q_v)\) are respectively the true view-specific Gibbs risk and the expected disagreement defined as

\[ R_D(G_{Q_v}) = \mathbb{E}_{(x,y) \sim D} \mathbb{I}_{[h(x^v) \neq y]}, \]

\[ d_D(Q_v) = \mathbb{E}_{x \sim D} \mathbb{E}_{h \sim Q_v} \mathbb{E}_{h' \sim Q_v} \mathbb{I}_{[h(x^v) \neq h'(x^v)]}. \]

\(^1\)In the PAC-Bayesian literature, the weighted majority vote is sometimes called the Bayes classifier.
3.2 The General Multiview PAC-Bayesian Theorem

In this section, we give a general multiview PAC-Bayesian theorem Goyal et al. [2017] that takes the form of a generalization bound for the Gibbs risk in the context of a two-level hierarchy of distributions. A key step in PAC-Bayesian proofs is the use of a change of measure inequality McAllester [2003], based on the Donsker-Varadhan inequality Donsker and Varadhan [1975]. Lemma 2 below extends this tool to our multiview setting.

**Lemma 2** For any set of priors \( \{ P_v \}_{v=1}^V \) over \( \{ H_v \}_{v=1}^V \) and any set of posteriors \( \{ Q_v \}_{v=1}^V \) over \( \{ H_v \}_{v=1}^V \), for any hyper-prior distribution \( \pi \) on views \( V \) and hyper-posterior distribution \( \rho \) on \( V \), and for any measurable function \( \phi : H_v \to \mathbb{R} \), we have

\[
E_{v \sim \rho} E_{h \sim Q_v} \phi(h) \leq E_{v \sim \rho} KL(Q_v || P_v) + KL(\rho || \pi) + \ln \left( E_{v \sim \pi} E_{h \sim P_v} e^{\phi(h)} \right).
\]

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

$$
D(R_S(G_\rho), R_D(G_\rho)) \leq \frac{1}{m} \left[ \sum_{v=1}^V \text{KL}(Q_v || P_v) + \text{KL}(\rho || \pi) + \ln \left( \frac{1}{\delta} \mathbb{E}_{v \sim \pi} \mathbb{E}_{h \sim P_v} e^{nD(R_S(h), R_D(h))} \right) \right] .
$$

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

$$
\mathbb{E}_{v \sim \pi} \mathbb{E}_{h \sim P_v} e^{nD(R_S(h), R_D(h))} \leq \frac{1}{\delta} \mathbb{E}_{S \sim (D)^n} \mathbb{E}_{v \sim \pi} \mathbb{E}_{h \sim P_v} e^{nD(R_S(h), R_D(h))} .
$$

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

$$
\ln \left( \mathbb{E}_{v \sim \pi} \mathbb{E}_{h \sim P_v} e^{nD(R_S(h), R_D(h))} \right) \leq \ln \left( \frac{1}{\delta} \mathbb{E}_{S \sim (D)^n} \mathbb{E}_{v \sim \pi} \mathbb{E}_{h \sim P_v} e^{nD(R_S(h), R_D(h))} \right) .
$$

(4)

We now apply Lemma 2 on the left-hand side of Inequality 4 with $\phi(h) = nD(R_S(h), R_D(h))$. Therefore, for any $Q_v$ on $H_v$ for all views $v \in \mathcal{V}$, and for any $\rho$ on views $\mathcal{V}$, with a probability at least $1 - \delta$ over $S \sim (D)^n$, we have

$$
\ln \left( \mathbb{E}_{v \sim \pi} \mathbb{E}_{h \sim P_v} e^{nD(R_S(h), R_D(h))} \right)
\geq
n \mathbb{E}_{v \sim \pi} \mathbb{E}_{h \sim Q_v} D(R_S(h), R_D(h)) - \mathbb{E}_{v \sim \pi} \text{KL}(Q_v || P_v) - \text{KL}(\rho || \pi)
\geq
n \mathbb{E}_{v \sim \pi} \mathbb{E}_{h \sim Q_v} R_S(h) - \mathbb{E}_{v \sim \pi} \mathbb{E}_{h \sim Q_v} R_D(h) - \mathbb{E}_{v \sim \pi} \text{KL}(Q_v || P_v) - \text{KL}(\rho || \pi),
$$

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

$$
D \left( \mathbb{E}_{v \sim \pi} \mathbb{E}_{h \sim Q_v} R_S(h), \mathbb{E}_{v \sim \pi} \mathbb{E}_{h \sim Q_v} R_D(h) \right) \leq \frac{1}{m} \left[ \mathbb{E}_{v \sim \pi} \text{KL}(Q_v || P_v) + \text{KL}(\rho || \pi) + \ln \left( \frac{1}{\delta} \mathbb{E}_{S \sim (D)^n} \mathbb{E}_{v \sim \pi} \mathbb{E}_{h \sim P_v} e^{nD(R_S(h), R_D(h))} \right) \right] .
$$

Finally, the theorem statement is obtained by rewriting

$$
\mathbb{E}_{v \sim \pi} \mathbb{E}_{h \sim Q_v} R_S(h) = R_S(G_\rho),
\mathbb{E}_{v \sim \pi} \mathbb{E}_{h \sim Q_v} R_D(h) = R_D(G_\rho).
$$

(5)

(6)

Compared to the classical single-view PAC-Bayesian Bound of [Germain et al., 2009, 2015], the main difference relies on the introduction of the view-specific prior and posterior distributions, which mainly leads to an additional term $\mathbb{E}_{v \sim \pi} \text{KL}(Q_v || P_v)$ expressed as the expectation of the view-specific Kullback-Leibler divergence term over the views $\mathcal{V}$ according to the hyper-posterior distribution $\rho$.

Theorem 1 provides tools to derive PAC-Bayesian generalization bounds for a multiview supervised learning setting. Indeed, by making use of the same trick as [Germain et al., 2009, 2015], by choosing a suitable convex function $D$ and upper-bounding $\mathbb{E}_{S \sim (D)^n} \mathbb{E}_{v \sim \pi} \mathbb{E}_{h \sim P_v} e^{nD(R_S(h), R_D(h))}$, we obtain an instantiation of Theorem 1. In the next section, we give an example of this kind of deviation through the approach of [Catoni, 2007], that is one of the three classical PAC-Bayesian Theorems [McAllester, 1999, Catoni, 2007, Seeger, 2002, Langford, 2005].
3.3 An Example of Instantiation of the Multiview PAC-Bayesian Theorem

To obtain the following theorem which is a generalization bound with the Catoni [2007]’s point of view, we put \( D \) as \( D(a, b) = \mathcal{F}(b) - C a \) where \( \mathcal{F} \) is a convex function \( \mathcal{F} \) and \( C > 0 \) is a real number.\cite{Germain2009, Germain2015}.

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

\[
R_D(G_\rho) \leq \frac{1}{1 - e^{-\delta}} \left( 1 - \exp \left[ - \left( C R_S(G_\rho) + \frac{1}{n} \left[ \mathbb{E} \text{KL}(Q_v \| P_v) + \text{KL}(\rho \| \pi) + \ln \frac{4}{\delta} \right] \right) \right].
\]

**Proof.** Deferred to D.\footnote{This bound has the advantage of expressing a trade-off between the empirical Gibbs risk and the Kullback-Leibler divergences.}

3.4 A Generalization Bound for the C-Bound

From a practical standpoint, as pointed out before, controlling the multiview C-Bound of Equation \( \text{(3)} \) 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 \textsc{PB-MVBoost} derived in Section 4).

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

\[
R_D(B_\rho) \leq 1 - \frac{\left( 1 - 2 \mathbb{E} \sup_{v \sim \rho} (r_{Q_v,S}^{1/2})^2 \right)^2}{1 - 2 \mathbb{E} \inf_{v \sim \rho} d_{Q_v,S}^{1/2}},
\]

where

\[ r_{Q_v,S}^{1/2} = \left\{ r : \text{kl}(R_S(Q_v) \| r) \leq \frac{1}{n} \left[ \text{KL}(Q_v \| P_v) + \ln \frac{4 \sqrt{m}}{\delta} \right] \text{ and } r \leq \frac{1}{2} \}, \] (7)

and

\[ d_{Q_v,S}^{1/2} = \left\{ d : \text{kl}(d^{S_v} \| d) \leq \frac{1}{n} \left[ 2 \text{KL}(Q_v \| P_v) + \ln \frac{4 \sqrt{m}}{\delta} \right] \}. \] (8)

**Proof.** Similarly to Equations (23) and (24) of\cite{Germain2015}, we define the sets \( r_{Q_v,S}^{1/2} \) (Equation (7)) and \( d_{Q_v,S}^{1/2} \) (Equation (8)) for our setting. Finally, the bound is obtained (from Equation (3) of Lemma 1) by replacing the view-specific Gibbs risk \( R_D(G_{Q_v}) \) by its upper bound \( \sup r_{Q_v,S}^{1/2} \) and expected disagreement \( d_{Q_v} \) by its lower bound \( \inf d_{Q_v,S}^{1/2} \).\footnote{In boosting, the performance of a weak classifier is only slightly better than random guessing.}

4 The \textsc{PB-MVBoost} algorithm

In this section we exploit our two-level hierarchical strategy (see Figure 1) in order to learn a well-performing weighted combination of view-specific voters (or views) as in Equation (11). Therefore, we propose to follow a well-known approach to learn weighted combination of voters, that is boosting. Indeed, boosting aims at combining a set of weak voters to construct a good majority vote. 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.
Algorithm 1 PB-MVBoost

Input: Training set $S = (x_i, y_i), \ldots, (x_n, y_n)$, where $x_i = (x_1, x_2, \ldots, x_V)$ and $y_i \in \{-1, 1\}$. For each view $v \in V$, a view-specific hypothesis set $H_v$. Number of iterations $T$.

1: for $x_i \in S$ do
2: \quad $D_1(x_i) \leftarrow \frac{1}{n}$
3: \quad $\forall v \in V \rho_{1v} \leftarrow \frac{1}{V}$
4: for $t = 1, \ldots, T$ do
5: \quad $\forall v \in V, \ h^t_v \leftarrow \arg\min_{h \in H_v} \mathbb{E}_{(x_i, y_i) \sim D_t} \left[ I[h(x^t_v) \neq y_i] \right]$
6: \quad Compute error: $\forall v \in V, \epsilon^t_v \leftarrow \mathbb{E}_{(x_i, y_i) \sim D_t} \left[ I[h^t_v(x^t_v) \neq y_i] \right]$
7: \quad Compute voter weights (taking into account view specific information):
\quad $\forall v \in V, Q^t_v \leftarrow \frac{1}{2} \left[ \ln \left( \frac{1 - \epsilon^t_v}{\epsilon^t_v} \right) \right]$
8: \quad **Optimize** the multiview C-Bound to learn weights over the views
\quad $\rho^t \leftarrow \arg\max_\rho \left[ \frac{1 - 2 \sum_{v=1}^{V} \rho_v r^t_v}{1 - 2 \sum_{v=1}^{V} \rho_v d^t_v} \right]$
\quad such that $\sum_{v=1}^{V} \rho_v = 1, \ \rho_v \geq 0 \ \forall v \in \{1, \ldots, V\}$
\quad where $\forall v \in V, r^t_v \leftarrow \mathbb{E}_{(x_i, y_i) \sim D_t} h \sim H_v \left[ I[h(x^t_v) \neq y_i] \right]$
\quad $\forall v \in V, d^t_v \leftarrow \mathbb{E}_{(x_i, y_i) \sim D_t} h, h' \sim H_v \left[ I[h(x^t_v) \neq h'(x^t_v)] \right]$\n9: \quad for $x_i \in S$ do
10: \quad \quad $D_{t+1}(x_i) \leftarrow \frac{D_t(x_i) \exp \left( -y_i \sum_{v=1}^{V} \rho^t_v Q^t_v h^t_v(x^t_v) \right)}{\sum_{x'_i \in S} D_t(x'_i) \exp \left( -y_i \sum_{v=1}^{V} \rho^t_v Q^t_v h^t_v(x^t_v) \right)}$
11: \quad Return: For each view $v \in V$, weights over view-specific voters and weights over views, i.e., $\rho_T$
which performs better than the individual weak voters. Recall that in multiview learning it is crucial to take into account the interactions between voters and views [Amini et al. 2009, Goyal et al. 2017, Chapelle et al. 2010, Maillard and Vayatis 2009]. We adapt this principle to our setting for combining a set of view-specific weak voters while taking into account the accuracy and diversity between them. We develop a multiview learning algorithm PB-MVBoost (see Algorithm 1), which allows to iteratively learn the set of view-specific classifiers that the algorithm will combine.

Concretely, for a given training set \( S = \{(x_1, y_1), \ldots, (x_n, y_n)\} \in (\mathcal{X} \times \{-1, +1\})^n \) of size \( n \), our algorithm PB-MVBoost maintains a distribution over the examples which is initialized as uniform. Then at each iteration, \( V \) view-specific weak classifiers are learned according to the current distribution \( D_t \) (Step 5), and their corresponding errors \( e'_v \) are estimated (Step 6).

Similarly to the Adaboost algorithm [Freund and Schapire 1997], the weights of each view-specific classifier \( (Q'_v)_1 \leq v \leq V \) are then computed with respect to these errors as

\[
\forall v \in V, Q'_v \leftarrow \frac{1}{2} \left[ \ln \left( \frac{1 - e'_v}{e'_v} \right) \right].
\]

To learn the weights \( (\rho_v)_1 \leq v \leq V \) over the views, we optimize the multiview C-Bound, given by Equation 3 of Lemma 1 (Step 8 of algorithm), which in our case writes as a constraint maximization problem:

\[
\max_{\rho} \left[ \frac{1 - 2 \sum_{v=1}^{V} \rho_v r'_v^2}{1 - 2 \sum_{v=1}^{V} \rho_v d'_v} \right],
\]

\[
\text{s.t. } \sum_{v=1}^{V} \rho_v = 1, \quad \rho_v \geq 0 \quad \forall v \in \{1, \ldots, V\}.
\]

where \( r_v \) is the view-specific Gibbs risk, and \( d_v \) the expected disagreement over all view-specific voters defined as follows.

\[
Q'_v = \mathbb{E}_{(x, y) \sim D_t, h \sim H_v} 1[h(x^*_v) \neq y], \tag{9}
\]

\[
D'_v = \mathbb{E}_{(x, y) \sim D_t, h \sim H_v} 1[h(x^*_v) \neq h(x)] \tag{10}.
\]

Intuitively, the multiview C-Bound tries to diversify the view-specific voters and views (Equation 10) while controlling the classification error of the view-specific classifiers (Equation 9). This allows us to control the accuracy and the diversity between the views which is an important ingredient in multiview learning [Xu and Sun 2010, Goyal et al. 2017, Peng et al. 2011, 2017, Morvant et al. 2014].

In Section 5, we empirically show that our algorithm minimizes the multiview C-Bound over the iterations of the algorithm (this is theoretically justified by the generalization bound of Theorem 2). Finally, we update the distribution over training examples \( x_i \) (Step 9), by following the Adaboost algorithm and in a way that the weights of misclassified (resp. well classified) examples by the final weighted majority classifier increase (resp. decrease).

\[
D_{t+1}(x_i) \leftarrow \frac{D_t(x_i) \exp \left( -y_i \sum_{v=1}^{V} \rho'_v Q'_v h'_v(x^*_v) \right)}{\sum_{j=1}^{n} D_t(x_j) \exp \left( -y_j \sum_{v=1}^{V} \rho'_v Q'_v h'_v(x^*_v) \right)}.
\]

Intuitively, this forces the view-specific classifiers to be consistent with each other, which is important for multiview learning [Janodet et al. 2009, Koço and Capponi 2011, Xiao and Guo 2012]. Finally, after \( T \) iterations of the algorithm, we learn the weights over the view-specific voters and weights over the views leading to a well-performing weighted multiview majority vote defined as

\[
B_p(x) = \text{sign} \left( \sum_{v=1}^{V} \rho_v^T \sum_{t=1}^{T} Q'_v h'_v(x^v) \right).
\]

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

The complexity of learning a decision tree classifier is \( O(d n \log(n)) \), where \( d \) is the depth of the decision tree. We learn the weights over the views by optimizing Equation 3 (Step 8 of our algorithm) using SLSQP method which has time complexity of \( O(V^3) \). Therefore, the overall complexity is \( O(T (V^3 + V d_n n \log(n))) \). 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).
We reserve 30% of documents as test samples and remaining as training data.

This dataset is a multilingual text classification data extracted from Reuters RCV1/RCV2 corpus. It consists of more than 110,000 documents written in five different languages (English, French, German, Italian and Spanish) distributed over six classes. We see different languages as different views of the data.

In this section, we present experiments to show the potential of our algorithm PB-MVBoost on the following datasets.

5.1 Datasets

**MNIST**

MNIST is a publicly available dataset consisting of 70,000 images of handwritten digits distributed over ten classes [Lecun et al. 1998]. For our experiments, we generated 2 four-view datasets where each view is a vector of $\mathbf{R}^{14}$. Similarly than done by Chen and Denoyer [2017], the first dataset (MNIST$_1$) is generated by considering 4 quarters of image as 4 views. For the second dataset (MNIST$_2$), we consider 4 overlapping views around the centre of images: 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.

**Multilingual, Multiview Text categorization**

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

5.2 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 sizes of learning sample $S$ (150, 200, 250, 300, 500, 800, 1000) that are chosen randomly from the training data. Moreover, all the results are averaged over 20 random runs of the experiments. Since the classes are unbalanced, we report the accuracy along with F1-measure for the methods and all the scores are averaged over all the one-vs-all classification problems.

We consider two multiview learning algorithms based on our two-step hierarchical strategy, and compare the PB-MVBoost algorithm described in Section 3, with a previously developed multiview learning algorithm [Goyal et al. 2017], based on classifier late fusion approach [Snoek et al. 2005], and referred to as Fusion$_{MV}$. Concretely, at the first level, this algorithm trains different view-specific linear SVM models with different hyperparameter $C$ values (12 values between $10^{-8}$ and $10^3$). And, at the second level, it learns a weighted combination over the predictions of view-specific voters using PAC-Bayesian

---

### Table 1: Test classification accuracy and $F_1$-score of different approaches averaged over all the classes and over 20 random sets of $n = 500$ labeled examples per training set. Along each column, the best result is in bold, and second one in italic. ↓ indicates that a result is statistically significantly worse than the best result, according to a Wilcoxon rank sum test with $p < 0.02$.

<table>
<thead>
<tr>
<th>Strategy</th>
<th>MNIST$_1$</th>
<th>MNIST$_2$</th>
<th>Reuters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy</td>
<td>$F_1$</td>
<td>Accuracy</td>
</tr>
<tr>
<td>Mono</td>
<td>0.9034 ± 0.001$^+$</td>
<td>0.5355 ± 0.006$^+$</td>
<td>0.9164 ± 0.001$^+$</td>
</tr>
<tr>
<td>Concat</td>
<td>0.9224 ± 0.002$^+$</td>
<td>0.6168 ± 0.011$^+$</td>
<td>0.9214 ± 0.002$^+$</td>
</tr>
<tr>
<td>Fusion$_{MV}$</td>
<td>0.9320 ± 0.001$^+$</td>
<td>0.5451 ± 0.019$^+$</td>
<td>0.9366 ± 0.001$^+$</td>
</tr>
<tr>
<td>MV-MVBoost</td>
<td>0.9402 ± 0.001$^+$</td>
<td>0.6321 ± 0.009$^+$</td>
<td>0.9450 ± 0.001$^+$</td>
</tr>
<tr>
<td>rBoost.SH</td>
<td>0.9256 ± 0.001$^+$</td>
<td>0.5315 ± 0.009$^+$</td>
<td>0.9545 ± 0.007</td>
</tr>
<tr>
<td>MV-AdaBoost</td>
<td>0.9514 ± 0.001</td>
<td>0.6510 ± 0.012$^+$</td>
<td>0.9611 ± 0.0009</td>
</tr>
<tr>
<td>MVBoost</td>
<td>0.9494 ± 0.003$^+$</td>
<td>0.7733 ± 0.006$^+$</td>
<td>0.9555 ± 0.002</td>
</tr>
<tr>
<td>Fusion$_{S^3}$</td>
<td>0.9418 ± 0.002$^+$</td>
<td>0.6120 ± 0.040$^+$</td>
<td>0.9548 ± 0.003$^+$</td>
</tr>
<tr>
<td>PB-MVBoost</td>
<td>0.9661 ± 0.0009</td>
<td><strong>0.8066 ± 0.005</strong></td>
<td><strong>0.9674 ± 0.0009</strong></td>
</tr>
</tbody>
</table>

---

3MMNIST$_1$ and MMNIST$_2$ datasets are available at [https://github.com/goyalanil/Multiview_Dataset_MNIST](https://github.com/goyalanil/Multiview_Dataset_MNIST)

4Reuters RCV1/RCV2 corpus is available at [https://archive.ics.uci.edu/ml/datasets/Reuters+RCV1+RCV2](https://archive.ics.uci.edu/ml/datasets/Reuters+RCV1+RCV2)

5Multilingual, Multiview Text Categorization Test Collection

6Code for PB-MVBoost is available at [https://github.com/goyalanil/PB-MVBoost](https://github.com/goyalanil/PB-MVBoost)
algorithm \texttt{CqBoost} \cite{Roy et al. 2016} with a RBF kernel. Note that, algorithm \texttt{CqBoost} tends to minimize the PAC-Bayesian C-Bound \cite{Germain et al. 2015} controlling the trade-off between accuracy and disagreement among voters. The hyperparameter \( \gamma \) of the RBF kernel is chosen over a set of 9 values between \( 10^{-6} \) and \( 10^2 \); and hyperparameter \( \mu \) is chosen over a set of 8 values between \( 10^{-8} \) and \( 10^{-1} \). To study the potential of our algorithms (\texttt{Fusion}\textsubscript{all} and \texttt{PB-MVBoost}), we considered following 7 baseline approaches:

- \textbf{Mono}: We learn a view-specific model for each view using a decision tree classifier and report the results of the best performing view.

- \textbf{Concat}: We learn one model using a decision tree classifier by concatenating features of all the views.

- \textbf{Fusion}\textsubscript{all}: This is a late fusion approach where we first learn the view-specific classifiers using 60\% of learning samples. Then, we learn a final multiview weighted model over the predictions of the view-specific classifiers. For this approach, we used decision tree classifiers at both levels of learning.

- \textbf{MV-MV}: We compute a multiview uniform majority vote (similar to approach followed by \cite{Amini et al. 2009}) over all the view-specific classifiers’ outputs in order to make final prediction. We learn view-specific classifiers using decision tree classifiers.

- \textbf{rBoost.SH}: This is the multiview learning algorithm proposed by \cite{Peng et al. 2011, 2017} 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, \texttt{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 \cite{Peng et al. 2017}.

- \textbf{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.

- \textbf{MVBoost}: This is a variant of our algorithm \texttt{PB-MVBoost} but without learning weights over views by optimizing multiview C-Bound. Here, our objective is to see the effect of learning weights over views on multiview learning.

For all boosting based approaches (\texttt{rBoost.SH}, \texttt{MV-AdaBoost}, \texttt{MVBoost} and \texttt{PB-MVBoost}), we learn the view-specific voters using a decision tree classifier with depth 2 and 4 as a weak classifier for \texttt{MNIST}, and \texttt{Reuters} RCV1/rcv2 datasets respectively. For all these approaches, we kept \( T = 100 \) as the number of iterations. For optimization of multiview C-Bound, we used Sequential Least Squares Programming (SLSQP) implementation provided by scikit-learn \cite{Pedregosa et al. 2011} and the decision trees implementation from scikit-learn \cite{Pedregosa et al. 2011}.

### 5.3 Results

Firstly, we report the comparison of our algorithms \texttt{Fusion}\textsubscript{all} and \texttt{PB-MVBoost} (for \( m = 500 \)) with all the considered baseline methods in Table 1. Secondly, Figure 2 illustrates the evolution of the performances according to the size of the learning sample. From the table, proposed two-step learning algorithm \texttt{Fusion}\textsubscript{all} is significantly better than the baseline approaches for \texttt{Reuters} dataset. Whereas, our boosting based algorithm \texttt{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.

In Figure 3, we compare proposed algorithms \texttt{Fusion}\textsubscript{all} and \texttt{PB-MVBoost} in terms of accuracy, \( F_1 \)-score and time complexity for \( m = 500 \) examples. For \texttt{MNIST} datasets, \texttt{PB-MVBoost} is significantly better than \texttt{Fusion}\textsubscript{all}. For \texttt{Reuters} dataset, \texttt{Fusion}\textsubscript{all} performs better than \texttt{PB-MVBoost} but computation time for \texttt{Fusion}\textsubscript{all} is much higher than that of \texttt{PB-MVBoost}. Moreover, in Figure 2 we can see that the performance (in terms of \( F_1 \)-score) for \texttt{Fusion}\textsubscript{all} is worse than \texttt{PB-MVBoost} when we have less training examples (\( n = 150 \) and 200). This shows the proposed boosting based one-step algorithm \texttt{PB-MVBoost} is more stable and more effective for multiview learning.

From Table 1 and Figure 2, we can observe that \textbf{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 \( F_1 \)-measure. On the other hand, \textbf{MVBoost} (where we have single global
distribution over the learning sample but without learning weights over views) is better compared to other baselines in terms of F1-measure. Moreover, the performances of MVBoost 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 monotonic increase of the performances with respect to the addition of labeled examples. This confirms 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 behaviour of our algorithm PB-MVBoost over $T = 100$ iterations on Figure 4 for all the datasets. We plot accuracy and F1-measure of learned models on training and test data along with empirical multiview C-Bound on 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 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 information and diversity between the views for multiview learning.
6 Conclusion

In this paper, we provide a PAC-Bayesian analysis for a two-level hierarchical multiview learning approach with more than two views, when the model takes the form of a weighted majority vote over a set of functions/voters. We consider a hierarchy of weights modelled by distributions where for each view we aim at learning \(i\) posterior \(Q_v\) distributions over the view-specific voters capturing the view-specific information and \(ii\) hyper-posterior \(\rho_v\) distributions over the set of the views. Based on this strategy, we derived a general multiview PAC-Bayesian theorem that can be specialized to any convex function to compare the empirical and true risks of the stochastic multiview Gibbs classifier. We propose a boosting-based learning algorithm, called as \textbf{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 C-Bound) that has the advantage of controlling a trade-off between accuracy and the diversity between the views. The empirical evaluation shows that \textbf{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 the 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 the learning sample for all the views and learning the weights over the views is an effective way to deal with multiview learning. In this way, we are able to capture the view-specific information and control the diversity between the views. Finally, we compare \textbf{PB-MVBoost} with a two-step learning algorithm \textbf{Fusion} which is based on PAC-Bayesian theory. We show that \textbf{PB-MVBoost} is more stable and computationally faster than \textbf{Fusion}.

For future work, we would like to specialize our PAC-Bayesian generalization bounds to linear classifiers \cite{Germain2009} which will clearly open the door to derive theoretically founded multiview learning algorithms. We would also like to extend our algorithm to semi-supervised multiview learning where one has access to an additional unlabeled data during training. One possible way is to learn a view-specific voter using pseudo-labels (for unlabeled data) generated from the voters trained from other views (as done for example in \cite{Xu2016}). Another possible direction is to make use of unlabeled data while computing view-specific disagreement for optimizing multiview C-Bound. This clearly opens the door to derive theoretically founded algorithms for semi-supervised multiview learning using PAC-Bayesian theory. We would like to extend our algorithm to transfer learning setting where training and test data are drawn from different distributions. An interesting direction would be to bind the data distribution to the different views of the data, as in some recent zero-shot learning approaches \cite{Socher2013}. Moreover, we would like to extend our work to the case of missing views or incomplete views e.g. \cite{Amini2009} and \cite{Xu2015}. 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.
Figure 4: Plots for classification error and F1-measure on training and test data; and empirical multiview C-Bound on training data over the iterations for all datasets with $n = 500$. 
Acknowledgements

This work was partially funded by the French ANR project LIVES ANR-15-CE23-0026-03 and the “Région Rhône-Alpes”.

Appendix

A Mathematical Tools

Theorem 3 (Markov’s ineq.) For any random variable $X$ s.t. $E(|X|) = \mu$, for any $a > 0$, we have $P(|X| \geq a) \leq \frac{\mu}{a}$.

Theorem 4 (Jensen’s ineq.) For any random variable $X$, for any concave function $g$, we have $g(E[X]) \geq E[g(X)]$.

Theorem 5 (Cantelli-Chebyshev ineq.) For any random variable $X$ s.t. $E(X) = \mu$ and $\text{Var}(X) = \sigma^2$, and for any $a > 0$, we have $P(X - \mu \geq a) \leq \frac{\sigma^2}{\sigma^2 + a^2}$.

B Proof of $C$-Bound for Multiview Learning (Lemma 1)

In this section, we present the proof of Lemma 1 inspired by the proof provided by Germain et al. [2015]. Firstly, we need to define the margin of the multiview weighted majority vote $B_\rho$ and its first and second statistical moments.

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

$$M_\rho(x,y) = E_{v \sim \rho} E_{h \sim Q_v} y h(x^v).$$

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

$$\mu_1(M^D_\rho) = E_{(x,y) \sim D} M_\rho(x,y).$$

and,

$$\mu_2(M^D_\rho) = E_{(x,y) \sim D} [M_\rho(x,y)]^2$$

$$= E_{x \sim D} y^2 \left[ E_{v \sim \rho} E_{h \sim Q_v} h(x^v) \right]^2 = E_{x \sim D} \left[ E_{v \sim \rho} E_{h \sim Q_v} h(x^v) \right]^2.$$  \hspace{1cm} (12)

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

$$R_D(B_\rho) = \frac{P}{(x,y) \sim D} \left( M_\rho(x,y) \leq 0 \right).$$

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 $h : \mathcal{X} \rightarrow \{-1,1\}$, we have $\mathbb{1}_{[h(x^v) \neq y]} = \frac{1}{2}(1 - y h(x^v))$, and therefore

$$R_D(G_\rho) = \frac{E}{(x,y) \sim D} E_{v \sim \rho} E_{h \sim Q_v} \mathbb{1}_{[h(x^v) \neq y]}$$

$$= \frac{1}{2} \left( 1 - \frac{E}{(x,y) \sim D} E_{v \sim \rho} E_{h \sim Q_v} y h(x^v) \right)$$

$$= \frac{1}{2} (1 - \mu_1(M^D_\rho)).$$  \hspace{1cm} (13)

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

$$d_D(\rho) = \frac{E}{x \sim D} E_{v \sim \rho} E_{v^\prime \sim \rho} E_{h \sim Q_v} E_{h^\prime \sim Q_{v'}} \mathbb{1}_{[h(x^v) \neq h'(x'^v)]}$$
\[ = \frac{1}{2} \left( 1 - \mathbb{E}_{x \sim D, v \sim \rho} \mathbb{E}_{v \sim \rho, h \sim Q_v} \mathbb{E}_{v' \sim \rho, h' \sim Q_{v'}} h(x^v) \times h'(x'^{v'}) \right) \]
\[ = \frac{1}{2} \left( \mathbb{E}_{x \sim D, v \sim \rho} \mathbb{E}_{v \sim \rho, h \sim Q_v} h(x^v) \times \mathbb{E}_{v \sim \rho, h \sim Q_v} h'(x'^{v'}) \right) \]
\[ = \frac{1}{2} \left( 1 - \mathbb{E}_{x \sim D, v \sim \rho} \mathbb{E}_{v \sim \rho, h \sim Q_v} h(x^v)^2 \right) \]
\[ = \frac{1}{2} (1 - \mu_2(M_D^\rho)) . \]

From above, we can easily deduce that \( 0 \leq d_D(\rho) \leq 1/2 \) as \( 0 \leq \mu_2(M_D^\rho) \leq 1 \). Therefore, the variance of the margin can be written as:
\[
\text{Var}(M_D^\rho) = \text{Var}_{(x,y) \sim D} (M_\rho(x,y)) = \mu_2(M_D^\rho) - (\mu_1(M_D^\rho))^2.
\]

**The proof of the C-bound**

**Proof.** By making use of one-sided Chebyshev inequality (Theorem 5 of [1]), with \( X = -M_\rho(x,y), \mu = \mathbb{E}_{(x,y) \sim D} (M_\rho(x,y)) \) and \( a = \mathbb{E}_{(x,y) \sim D} M_\rho(x,y) \), we have
\[
R_D(B_\rho) = \mathbb{P}_{(x,y) \sim D} (M_\rho(x,y) \leq 0)
\[ = \mathbb{P}_{(x,y) \sim D} \left( -M_\rho(x,y) + \mathbb{E}_{(x,y) \sim D} M_\rho(x,y) \geq \mathbb{E}_{(x,y) \sim D} M_\rho(x,y) \right)
\[ \leq \frac{\text{Var}_{(x,y) \sim D} (M_\rho(x,y))}{\text{Var}(M_D^\rho)} + \left( \mathbb{E}_{(x,y) \sim D} M_\rho(x,y) \right)^2
\[ = \frac{\mu_2(M_D^\rho) - (\mu_1(M_D^\rho))^2 + (\mu_1(M_D^\rho))^2}{\mu_2(M_D^\rho)}
\[ = \frac{\mu_2(M_D^\rho) - (\mu_1(M_D^\rho))^2}{\mu_2(M_D^\rho)}
\[ = 1 - \frac{(\mu_1(M_D^\rho))^2}{\mu_2(M_D^\rho)}
\[ = 1 - \frac{1 - 2 R_D(G_\rho)}{1 - 2 d_D(\rho)}
\]

**C Proof of Lemma 2**

We have
\[
\mathbb{E}_{v \sim \rho, h \sim Q_v} \phi(h) = \mathbb{E}_{v \sim \rho, h \sim Q_v} \ln e^{\phi(h)}
\[ = \mathbb{E}_{v \sim \rho, h \sim Q_v} \ln \left( Q_v(h) P_v(h) Q_v(h) \right)^{\phi(h)}
\]
According to the Kullback-Leibler definition, we have

$$E = E_{v \sim \rho} \left[ E_{h \sim Q_v} \ln \left( \frac{Q_v(h)}{P_v(h)} \right) + E_{h \sim Q_v} \ln \left( \frac{P_v(h)}{Q_v(h)} e^{\phi(h)} \right) \right].$$

By applying Jensen’s inequality (Theorem 4 in Appendix) on the concave function ln, we have

$$E_{v \sim \rho} E_{h \sim Q_v} \phi(h) = E_{v \sim \rho} \left[ KL(Q_v || P_v) + E_{h \sim Q_v} \ln \left( \frac{P_v(h)}{Q_v(h)} e^{\phi(h)} \right) \right].$$

By applying Jensen’s inequality (Theorem 4) on ln to obtain the lemma.

**References**


