

PAC-Bayes and Domain Adaptation

Pascal Germain, Amaury Habrard, François Laviolette, Emilie Morvant

▶ **To cite this version:**

Pascal Germain, Amaury Habrard, François Laviolette, Emilie Morvant. PAC-Bayes and Domain Adaptation. Neurocomputing, 2020, 379, pp.379-397. 10.1016/j.neucom.2019.10.105 . hal-01563152v3

HAL Id: hal-01563152
https://hal.science/hal-01563152v3

Submitted on 6 Nov 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.

PAC-Bayes and Domain Adaptation

Pascal Germain[c,b], Amaury Habrard[a], Francois Laviolette[b], Emilie Morvant[a]

[a] *Univ Lyon, UJM-Saint-Etienne, CNRS, Institut d Optique Graduate School, Laboratoire Hubert Curien UMR 5516, F-42023, Saint-Etienne, France*
[b] *Département d'Informatique et de Génie Logiciel, Université Laval, Québec, Canada*
[c] *Inria Lille - Nord Europe, Modal Project-Team, 59650 Villeneuve d'Ascq, France*

Abstract

We provide two main contributions in PAC-Bayesian theory for domain adaptation where the objective is to learn, from a source distribution, a well-performing majority vote on a different, but related, target distribution. Firstly, we propose an improvement of the previous approach we proposed in [1], which relies on a novel distribution pseudodistance based on a disagreement averaging, allowing us to derive a new tighter domain adaptation bound for the target risk. While this bound stands in the spirit of common domain adaptation works, we derive a second bound (introduced in [2]) that brings a new perspective on domain adaptation by deriving an upper bound on the target risk where the distributions' divergence—expressed as a ratio—controls the trade-off between a source error measure and the target voters' disagreement. We discuss and compare both results, from which we obtain PAC-Bayesian generalization bounds. Furthermore, from the PAC-Bayesian specialization to linear classifiers, we infer two learning algorithms, and we evaluate them on real data.

Keywords: Domain Adaptation, PAC-Bayesian Theory

1. Introduction

As human beings, we learn from what we saw before. Think about our education process: When a student attends to a new course, the knowledge he has acquired from previous courses helps him to understand the current one. However, traditional machine learning approaches assume that the learning and test data are drawn from the same probability distribution. This assumption may be too strong for a lot of real-world tasks, in particular those where we desire to reuse a model from one task to another one. For instance, a spam filtering system suitable for one user can be poorly adapted to another who receives significantly different emails. In other words, the learning data associated with one or several users could be unrepresentative of the test data coming from another one. This enhances the need to design methods for adapting a classifier from learning (source) data to test (target) data. One solution to tackle this issue is to consider

the *domain adaptation* framework,[1] which arises when the distribution generating the target data (the *target domain*) differs from the one generating the source data (the *source domain*). Note that, it is well known that domain adaptation is a hard and challenging task even under strong assumptions [10, 11, 12].

1.1. Approaches to Address Domain Adaptation

Many approaches exist in the literature to address domain adaptation, often with the same underlying idea: If we are able to apply a transformation in order to "move closer" the distributions, then we can learn a model with the available labels. This process can be performed by reweighting the importance of labeled data [13, 14, 15, 16]. This is one of the most popular methods when one wants to deal with the covariate-shift issue [*e.g.,* 13, 17], where source and target domains diverge only in their marginals, *i.e.*, when they share the same labeling function. Another technique is to exploit self-labeling procedures, where the objective is to transfer the source labels to the target unlabeled points [*e.g.*, 18, 19, 20]. A third solution is to learn a new common representation space from the unlabeled part of source and target data. Then, a standard supervised learning algorithm can be run on the source labeled [*e.g.*, 21, 22, 23, 24, 25], or can be learned simultaneously with the new representation; the latter method being increasingly used in state-of-the-art deep neural networks learning strategies [*e.g.*, 26, 27, 28, 29, 30]. A slightly different approach, known as hypothesis transfer learning, aims at directly transferring the learned source model to the target domain [31, 32].

The work presented in this paper stands into a fifth popular class of approaches, which has been especially explored to derive generalization bounds for domain adaptation. This kind of approach relies on the control of a measure of divergence/distance between the source distribution and target distribution [*e.g.* 33, 34, 35, 36, 37, 38, 39]. Such a distance usually depends on the set $\mathcal{H}$ of hypotheses considered by the learning algorithm. The intuition is that one must look for a set $\mathcal{H}$ that minimizes the distance between the distributions while preserving good performances on the source data; if the distributions are close under this measure, then generalization ability may be "easier" to quantify. In fact, defining such a measure to quantify how much the domains are related is a major issue in domain adaptation. For example, for binary classification with the 0-1-loss function, Ben-David et al. [34, 33] have considered the *$\mathcal{H}\Delta\mathcal{H}$-divergence* between the source and target marginal distributions. This quantity depends on the maximal disagreement between two classifiers, and allowed them to deduce a domain adaptation generalization bound based on the VC-dimension theory. The *discrepancy distance* proposed by Mansour et al. [40] generalizes this divergence to real-valued functions and more general losses, and is used to obtain a generalization bound based on the Rademacher complexity. In this context, Cortes and Mohri [41, 38] have specialized the minimization of the

[1] The reader can refer to the surveys proposed in [3, 4, 5, 6, 7, 8] (domain adaptation is often associated with *transfer learning* [9]).

discrepancy to regression with kernels. In these situations, domain adaptation can be viewed as a multiple trade-off between the complexity of the hypothesis class $\mathcal{H}$, the adaptation ability of $\mathcal{H}$ according to the divergence between the marginals, and the empirical source risk. Moreover, other measures have been exploited under different assumptions, such as the Rényi divergence suitable for importance weighting [42], or the measure proposed by Zhang et al. [36] which takes into account the source and target true labeling, or the Bayesian *divergence prior* [35] which favors classifiers closer to the best source model, or the Wassertein distance [39] that justifies the usefulness of optimal transport strategy in domain adaptation [23, 24]. However, a majority of methods prefer to perform a two-step approach: *(i)* First construct a suitable representation by minimizing the divergence, then *(ii)* learn a model on the source domain in the new representation space.

1.2. A PAC-Bayesian Standpoint

Given the multitude of concurrent approaches for domain adaptation, and the nonexistence of a predominant one, we believe that the problem still needs to be studied from different perspectives for a global comprehension to emerge. We aim to contribute to this study from a PAC-Bayesian standpoint. One particularity of the *PAC-Bayesian theory* (first set out by McAllester [43]) is that it focuses on algorithms that output a *posterior distribution* ρ over a classifier set $\mathcal{H}$ (*i.e.*, a ρ-average over $\mathcal{H}$) rather than just a single predictor $h \in \mathcal{H}$ (as in [33], and other works cited above). More specifically, we tackle the *unsupervised domain adaptation* setting for binary classification, where no target labels are provided to the learner. We propose two domain adaptation analyses, both introduced separately in previous conference papers [1, 2]. We refine these results, and provide in-depth comparison, full proofs and technical details. Our analyses highlight different angles that one can adopt when studying domain adaptation.

Our first approach follows the philosophy of the seminal work of Ben-David et al. [34, 33] and Mansour et al. [42]: The risk of the target model is upper-bounded jointly by the model's risk on the source distribution, a divergence between the marginal distributions, and a non-estimable term[2] related to the ability to adapt in the current space. To obtain such a result, we define a pseudometric which is ideal for the PAC-Bayesian setting by evaluating the domains' divergence according to the ρ-average disagreement of the classifiers over the domains. Additionally, we prove that this domains' divergence is always lower than the popular $\mathcal{H}\Delta\mathcal{H}$-divergence, and is easily estimable from samples. Note that, based on this disagreement measure, we derived in a previous work [1] a first PAC-Bayesian domain adaptation bound expressed as a ρ-averaging. We provide here a new version of this result, that does not change the underlying philosophy supported by the previous bound, but clearly improves the theoretical result: The domain adaptation bound is now tighter and easier to interpret.

[2] More precisely, this term can only be estimated in the presence of labeled data from both the source and the target domains.

Our second analysis (introduced in [2]) consists in a target risk bound that brings an original way to think about domain adaptation problems. Concretely, the risk of the target model is still upper-bounded by three terms, but they differ in the information they capture. The first term is estimable from unlabeled data and relies on the disagreement of the classifiers only on the target domain. The second term depends on the expected accuracy of the classifiers on the source domain. Interestingly, this latter is weighted by a divergence between the source and the target domains that enables controlling the relationship between domains. The third term estimates the "volume" of the target domain living apart from the source one,[3] which has to be small for ensuring adaptation.

Thanks to these results, we derive PAC-Bayesian generalization bounds for our two domain adaptation bounds. Then, in contrast to the majority of methods that perform a two-step procedure, we design two algorithms tailored to linear classifiers, called PBDA and DALC, which jointly minimize the multiple trade-offs implied by the bounds. On the one hand, PBDA is inspired by our first analysis for which the first two quantities being, as usual in the PAC-Bayesian approach, the complexity of the ρ-weighted majority vote measured by a Kullback-Leibler divergence and the empirical risk measured by the ρ-average errors on the source sample. The third quantity corresponds to our domains' divergence and assesses the capacity of the posterior distribution to distinguish some structural difference between the source and target samples. On the other hand, DALC is inspired by our second analysis from which we deduce that a good adaptation strategy consists in finding a ρ-weighted majority vote leading to a suitable trade-off—controlled by the domains' divergence—between the first two terms (and the usual Kullback-Leibler divergence): Minimizing the first one corresponds to look for classifiers that disagree on the target domain, and minimizing the second one to seek accurate classifiers on the source.

1.3. Paper Structure and Novelties

This paper aims at unifying contributions of our previous papers [1, 2]. We have also added the following contributions.

- Subsection 3.3 presents the explicit derivation of the algorithm PBGD3 [44] (see also the mathematical details of Appendix B). An original illustration of the algorithm optimized trade-off is also given (Subsection 3.3.4 and Figure 1).
- Theorem 4 introduces an improved version of the original PAC-Bayesian domain adaptation bound [1]. As discussed in Subsection 4.1.4, this new theorem provides tighter generalization guarantees and is easier to interpret. Moreover, the bound is not degenerated when the source and target distributions are the same or close, which was an undesirable behavior of the previous result.

[3] Here we do not focus on learning a new representation to help the adaptation: We directly aim at adapting in the current representation space.

- Section 5 presents comprehensive and reorganized proofs of the main PAC-Bayesian results. In this new version, both Theorem 4 (improved version of [1]) and Theorem 5 (from [2]) build on a common result, given by Corollary 1, instead of being proven independently.
- Section 6 gives the extended mathematical details leading to the two learning algorithms (PBDA [1] and DALC [2]), including the equation of their kernelized version (Subsection 6.3.3). Moreover, an original toy experiment illustrates the particularities of the two algorithms in regards of each other (Subsection 6.4 and Figure 2).

The rest of the paper is structured as follows. Section 2 deals with two seminal works on domain adaptation. The PAC-Bayesian framework is then recalled in Section 3, along with the details of PBGD3 algorithm [44]. Our main contribution, which consists in two domain adaptation bounds suitable for PAC-Bayesian learning, is presented in Section 4, the associated generalization bounds are derived in Section 5. Then, we design our new algorithms for PAC-Bayesian domain adaptation in Section 6, that we empirically evaluate in Section 7. We conclude in Section 8.

2. Domain Adaptation Related Works

In this section, we review the two seminal works in domain adaptation that are based on a divergence measure between the domains [34, 33, 40].

2.1. Notations and Setting

We consider domain adaptation for binary classification[4] tasks where $\mathbf{X} \subseteq \mathbb{R}^d$ is the input space of dimension d, and $Y = \{-1, +1\}$ is the output/label set. The *source domain* $\mathcal{S}$ and the *target domain* $\mathcal{T}$ are two different distributions (unknown and fixed) over $\mathbf{X} \times Y$, $\mathcal{S}_{\mathbf{X}}$ and $\mathcal{T}_{\mathbf{X}}$ being the respective marginal distributions over $\mathbf{X}$. We tackle the challenging task where we have no target labels, known as *unsupervised domain adaptation*. A learning algorithm is then provided with a *labeled source sample* $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m_s}$ consisting of m_s examples drawn *i.i.d.*[5] from $\mathcal{S}$, and an *unlabeled target sample* $T = \{\mathbf{x}_j\}_{j=1}^{m_t}$ consisting of m_t examples drawn *i.i.d.* from $\mathcal{T}_{\mathbf{X}}$. We denote the distribution $\mathcal{D}$ of a m-sample by $(\mathcal{D})^m$. We suppose that $\mathcal{H}$ is a set of hypothesis functions for $\mathbf{X}$ to Y. The *expected source error* and the *expected target error* of $h \in \mathcal{H}$ over $\mathcal{S}$, respectively $\mathcal{T}$, are the probability that h errs on the entire distribution $\mathcal{S}$, respectively $\mathcal{T}$,

$$\mathrm{R}_{\mathcal{S}}(h) \;=\; \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{S}} \mathcal{L}_{0\text{-}1}\big(h(\mathbf{x}), y\big)\,, \quad \text{and} \quad \mathrm{R}_{\mathcal{T}}(h) \;=\; \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{T}} \mathcal{L}_{0\text{-}1}\big(h(\mathbf{x}), y\big)\,,$$

[4] Our domain adaptation analysis is tailored for binary classification, and does not directly extend to multi-class and regression problems.

[5] *i.i.d.* stands for *independent and identically distributed*.

where $\mathcal{L}_{0\text{-}1}(a,b) = \mathrm{I}[a{\neq}b]$ is the 0-1-loss function which returns 1 if $a{\neq}b$ and 0 otherwise. The *empirical source error* $\widehat{\mathrm{R}}_S(h)$ of h on the learning source sample S is

$$\widehat{\mathrm{R}}_S(h) \;=\; \frac{1}{m_s} \sum_{(\mathbf{x},y)\in S} \mathcal{L}_{0\text{-}1}\big(h(\mathbf{x}), y\big)\,.$$

The main objective in domain adaptation is then to learn—without target labels—a classifier $h \in \mathcal{H}$ leading to the lowest expected target error $\mathrm{R}_{\mathcal{T}}(h)$.

Given two classifiers $(h', h) \in \mathcal{H}^2$, we also introduce the notion of *expected source disagreement* $\mathrm{R}_{\mathcal{S}_\mathbf{X}}(h,h')$ and the *expected target disagreement* $\mathrm{R}_{\mathcal{T}_\mathbf{X}}(h,h')$, which measure the probability that h and h' do not agree on the respective marginal distributions, and are defined by

$$\mathrm{R}_{\mathcal{S}_\mathbf{X}}(h,h') \;=\; \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{S}_\mathbf{X}} \mathcal{L}_{0\text{-}1}\big(h(\mathbf{x}), h'(\mathbf{x})\big) \quad\text{and}\quad \mathrm{R}_{\mathcal{T}_\mathbf{X}}(h,h') \;=\; \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{T}_\mathbf{X}} \mathcal{L}_{0\text{-}1}\big(h(\mathbf{x}), h'(\mathbf{x})\big)\,.$$

The *empirical source disagreement* $\widehat{\mathrm{R}}_S(h,h')$ on S and the *empirical target disagreements* $\widehat{\mathrm{R}}_T(h,h')$ on T are

$$\widehat{\mathrm{R}}_S(h,h') = \frac{1}{m_s}\sum_{\mathbf{x}\in S} \mathcal{L}_{0\text{-}1}\big(h(\mathbf{x}), h'(\mathbf{x})\big) \;\text{ and }\; \widehat{\mathrm{R}}_T(h,h') = \frac{1}{m_t}\sum_{\mathbf{x}\in T} \mathcal{L}_{0\text{-}1}\big(h(\mathbf{x}), h'(\mathbf{x})\big).$$

Note that, depending on the context, S denotes either the source labeled sample $\{(\mathbf{x}_i, y_i)\}_{i=1}^{m_s}$ or its unlabeled part $\{\mathbf{x}_i\}_{i=1}^{m_s}$. We can remark that the expected error $R_\mathcal{D}(h)$ on a distribution $\mathcal{D}$ can be viewed as a shortcut notation for the expected disagreement between a hypothesis h and a labeling function $f_\mathcal{D} : \mathbf{X} \to Y$ that assigns the true label to an example description with respect to $\mathcal{D}$. We have

$$\begin{aligned}\mathrm{R}_\mathcal{D}(h) \;&=\; \mathrm{R}_{\mathcal{D}_\mathbf{X}}(h, f_\mathcal{D}) \\ &=\; \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{D}_\mathbf{X}} \mathcal{L}_{0\text{-}1}\big(h(\mathbf{x}), f_\mathcal{D}(\mathbf{x})\big)\,.\end{aligned}$$

2.2. Necessity of a Domains' Divergence

The domain adaptation objective is to find a low-error target hypothesis, even if the target labels are not available. Even under strong assumptions, this task can be impossible to solve [10, 11, 12]. However, for deriving generalization ability in a domain adaptation situation (with the help of a domain adaptation bound), it is critical to make use of a divergence between the source and the target domains: The more similar the domains, the easier the adaptation appears. Some previous works have proposed different quantities to estimate how a domain is close to another one [33, 35, 40, 42, 34, 36]. Concretely, two domains $\mathcal{S}$ and $\mathcal{T}$ differ if their marginals $\mathcal{S}_\mathbf{X}$ and $\mathcal{T}_\mathbf{X}$ are different, or if the source labeling function differs from the target one, or if both happen. This suggests taking into account two divergences: One between $\mathcal{S}_\mathbf{X}$ and $\mathcal{T}_\mathbf{X}$, and one between the labeling. If we have some target labels, we can combine the two distances as done by Zhang et al. [36]. Otherwise, we preferably consider two separate measures, since it is

impossible to estimate the best target hypothesis in such a situation. Usually, we suppose that the source labeling function is somehow related to the target one, then we look for a representation where the marginals $\mathcal{S}_{\mathbf{X}}$ and $\mathcal{T}_{\mathbf{X}}$ appear closer without losing performances on the source domain.

2.3. Domain Adaptation Bounds for Binary Classification

We now review the first two seminal works which propose domain adaptation bounds based on a divergence between the two domains.

First, under the assumption that there exists a hypothesis in $\mathcal{H}$ that performs well on both the source and the target domain, Ben-David et al. [33, 34] have provided the following domain adaptation bound.

Theorem 1 (Ben-David et al. [34], Ben-David et al. [33]). *Let $\mathcal{H}$ be a (symmetric[6]) hypothesis class. We have*

$$\forall h \in \mathcal{H},\ \mathrm{R}_{\mathcal{T}}(h) \leq \mathrm{R}_{\mathcal{S}}(h) + \tfrac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}_{\mathbf{X}}, \mathcal{T}_{\mathbf{X}}) + \mu_{h^*}\,, \tag{1}$$

where

$$\tfrac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}_{\mathbf{X}}, \mathcal{T}_{\mathbf{X}}) = \sup_{(h,h') \in \mathcal{H}^2} \left| \mathrm{R}_{\mathcal{T}_{\mathbf{X}}}(h, h') - \mathrm{R}_{\mathcal{S}_{\mathbf{X}}}(h, h') \right|$$

is the $\mathcal{H}\Delta\mathcal{H}$-distance between marginals $\mathcal{S}_{\mathbf{X}}$ and $\mathcal{T}_{\mathbf{X}}$, and μ_{h^*}=$\mathrm{R}_{\mathcal{S}}(h^*) + \mathrm{R}_{\mathcal{T}}(h^*)$ is the error of the best hypothesis overall h^*=$\operatorname{argmin}_{h \in \mathcal{H}} \left(\mathrm{R}_{\mathcal{S}}(h) + \mathrm{R}_{\mathcal{T}}(h) \right)$.

This bound relies on three terms. The first term $\mathrm{R}_{\mathcal{S}}(h)$ is the classical source domain expected error. The second term $\frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}_{\mathbf{X}}, \mathcal{T}_{\mathbf{X}})$ depends on $\mathcal{H}$ and corresponds to the maximum deviation between the source and target disagreement between two hypotheses of $\mathcal{H}$. In other words, it quantifies how hypothesis from $\mathcal{H}$ can "detect" differences between these marginals: The lower this measure is for a given $\mathcal{H}$, the better are the generalization guarantees. The last term $\mu_{h^*} = \mathrm{R}_{\mathcal{S}}(h^*) + \mathrm{R}_{\mathcal{T}}(h^*)$ is related to the best hypothesis $h^* \in \mathcal{H}$ over the domains and acts as a quality measure of $\mathcal{H}$ in terms of labeling information. If h^* does not have a good performance on both the source and the target domain, then there is no way one can adapt from this source to this target. Hence, as pointed out by the authors, Equation (1) expresses a multiple trade-off between the accuracy of some particular hypothesis h, the complexity of $\mathcal{H}$ (quantified in [34] with the usual VC-bound theory), and the "incapacity" of hypotheses of $\mathcal{H}$ to detect difference between the source and the target domain.

Second, Mansour et al. [40] have extended the $\mathcal{H}\Delta\mathcal{H}$-distance to the discrepancy divergence for regression and any symmetric loss $\mathcal{L}$ fulfilling the triangle inequality. Given $\mathcal{L} : [-1, +1]^2 \to \mathbb{R}^+$ such a loss, the discrepancy $\mathrm{disc}_{\mathcal{L}}(\mathcal{S}_{\mathbf{X}}, \mathcal{T}_{\mathbf{X}})$ between $\mathcal{S}_{\mathbf{X}}$ and $\mathcal{T}_{\mathbf{X}}$ is

$$\mathrm{disc}_{\mathcal{L}}(\mathcal{S}_{\mathbf{X}}, \mathcal{T}_{\mathbf{X}}) = \sup_{(h,h') \in \mathcal{H}^2} \left| \mathop{\mathbf{E}}_{\mathbf{x} \sim \mathcal{T}_{\mathbf{X}}} \mathcal{L}(h(\mathbf{x}), h'(\mathbf{x})) - \mathop{\mathbf{E}}_{\mathbf{x} \sim \mathcal{S}_{\mathbf{X}}} \mathcal{L}(h(\mathbf{x}), h'(\mathbf{x})) \right| .$$

[6] In a symmetric hypothesis space $\mathcal{H}$, for every $h \in \mathcal{H}$, its inverse $-h$ is also in $\mathcal{H}$.

Note that with the 0-1-loss in binary classification, we have

$$\tfrac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}_{\mathbf{X}}, \mathcal{T}_{\mathbf{X}}) \;=\; \mathrm{disc}_{\mathcal{L}_{0\text{-}1}}(\mathcal{S}_{\mathbf{X}}, \mathcal{T}_{\mathbf{X}})\,.$$

Even if these two divergences may coincide, the following domain adaptation bound of Mansour et al. [40] differs from Theorem 1.

Theorem 2 (Mansour et al. [40]). *Let $\mathcal{H}$ be a (symmetric) hypothesis class. We have*

$$\forall h \in \mathcal{H},\; \mathrm{R}_{\mathcal{T}}(h) - \mathrm{R}_{\mathcal{T}}(h^*_T) \;\leq\; \mathrm{R}_{\mathcal{S}_{\mathbf{X}}}(h^*_S, h) + \mathrm{disc}_{\mathcal{L}_{0\text{-}1}}(\mathcal{S}_{\mathbf{X}}, \mathcal{T}_{\mathbf{X}}) + \nu_{(h^*_S, h^*_T)}\,, \quad (2)$$

with $\nu_{(h^*_S, h^*_T)} = \mathrm{R}_{\mathcal{S}_{\mathbf{X}}}(h^*_S, h^*_T)$ the disagreement between the ideal hypothesis on the target and source domains: $h^*_T = \mathrm{argmin}_{h\in\mathcal{H}} \mathrm{R}_{\mathcal{T}}(h)$, and $h^*_S = \mathrm{argmin}_{h\in\mathcal{H}} \mathrm{R}_{\mathcal{S}}(h)$.

Equation (2) can be tighter than Equation (1)[7] since it bounds the difference between the target error of a classifier and the one of the optimal h^*_T. Based on Theorem 2 and a Rademacher complexity analysis, Mansour et al. [40] provide a generalization bound on the target risk, that expresses a trade-off between the disagreement (between h and the best source hypothesis h^*_S), the complexity of $\mathcal{H}$, and—again—the "incapacity" of hypotheses to detect differences between the domains.

To conclude, the domain adaptation bounds of Theorems 1 and 2 suggest that if the divergence between the domains is low, a low-error classifier over the source domain might perform well on the target one. These divergences compute the *worst-case* of the disagreement between a pair of hypotheses. We propose in Section 4 two *average case* approaches by making use of the essence of the PAC-Bayesian theory, which is known to offer tight generalization bounds [43, 44, 45]. Our first approach (see Section 4.1) stands in the philosophy of these seminal works, and the second one (see Section 4.2) brings a different and novel point of view by taking advantages of the PAC-Bayesian framework we recall in the next section.

3. PAC-Bayesian Theory in Supervised Learning

Let us now review the classical supervised binary classification framework called the PAC-Bayesian theory, first introduced by McAllester [43]. This theory succeeds to provide tight generalization guarantees—without relying on any validation set—on weighted majority votes, *i.e.*, for ensemble methods [46, 47] where several classifiers (or voters) are assigned a specific weight. Throughout this section, we adopt an algorithm design perspective. Indeed, the PAC-Bayesian analysis of domain adaptation provided in the forthcoming sections is oriented by the motivation of creating new adaptive algorithms.

[7] Equation (1) can lead to an error term three times higher than Equation (2) in some cases (more details in [40]).

3.1. Notations and Setting

Traditionally, PAC-Bayesian theory considers weighted majority votes over a set $\mathcal{H}$ of binary hypothesis, often called voters. Let $\mathcal{D}$ be a fixed yet unknown distribution over $\mathbf{X} \times Y$, and S be a learning set where each example is drawn *i.i.d.* from $\mathcal{D}$. Then, given a *prior distribution* π over $\mathcal{H}$ (independent from the learning set S), the "PAC-Bayesian" learner aims at finding a *posterior distribution* ρ over $\mathcal{H}$ leading to a *ρ-weighted majority vote* B_ρ (also called the Bayes classifier) with good generalization guarantees and defined by

$$B_\rho(\mathbf{x}) \;=\; \operatorname{sign}\Big[\mathop{\mathbf{E}}_{h\sim\rho} h(\mathbf{x})\Big]\,.$$

However, minimizing the risk of B_ρ, defined as

$$\mathrm{R}_{\mathcal{D}}(B_\rho) \;=\; \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \mathcal{L}_{0\text{-}1}\big(B_\rho(\mathbf{x}), y\big)\,,$$

is known to be NP-hard. To tackle this issue, the PAC-Bayesian approach deals with the risk of the stochastic *Gibbs classifier* G_ρ associated with ρ and closely related to B_ρ. In order to predict the label of an example $\mathbf{x} \in \mathbf{X}$, the Gibbs classifier first draws a hypothesis h from $\mathcal{H}$ according to ρ, then returns $h(\mathbf{x})$ as label. Then, the error of the Gibbs classifier on a domain $\mathcal{D}$ corresponds to the expectation of the errors over ρ:

$$\mathrm{R}_{\mathcal{D}}(G_\rho) \;=\; \mathop{\mathbf{E}}_{h\sim\rho} \mathrm{R}_{\mathcal{D}}(h)\,. \tag{3}$$

In this setting, if B_ρ misclassifies $\mathbf{x}$, then at least half of the classifiers (under ρ) errs on $\mathbf{x}$. Hence, we have

$$\mathrm{R}_{\mathcal{D}}(B_\rho) \;\leq\; 2\,\mathrm{R}_{\mathcal{D}}(G_\rho)\,.$$

Another result on the relation between $\mathrm{R}_{\mathcal{D}}(B_\rho)$ and $\mathrm{R}_{\mathcal{D}}(G_\rho)$ is the C-bound of Lacasse et al. [48] expressed as

$$\mathrm{R}_{\mathcal{D}}(B_\rho) \;\leq\; 1 - \frac{\big(1 - 2\,\mathrm{R}_{\mathcal{D}}(G_\rho)\big)^2}{1 - 2\,\mathrm{d}_{\mathcal{D}_{\mathbf{X}}}(\rho)}\,, \tag{4}$$

where $\mathrm{d}_{\mathcal{D}_{\mathbf{X}}}(\rho)$ corresponds to the *expected disagreement* of the classifiers over ρ:

$$\mathrm{d}_{\mathcal{D}_{\mathbf{X}}}(\rho) \;=\; \mathop{\mathbf{E}}_{(h,h')\sim\rho^2} \mathrm{R}_{\mathcal{D}_{\mathbf{X}}}(h, h')\,. \tag{5}$$

Equation (4) suggests that for a fixed numerator, *i.e.*, a fixed risk of the Gibbs classifier, the best ρ-weighted majority vote is the one associated with the lowest denominator, *i.e.*, with the greatest disagreement between its voters (for further analysis, see [49]).

We now introduce the notion of *expected joint error* of a pair of classifiers $(h, h') \in \mathcal{H}^2$ drawn according to the distribution ρ, defined as

$$\mathrm{e}_{\mathcal{D}}(\rho) \;=\; \mathop{\mathbf{E}}_{(h,h')\sim\rho^2} \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \mathcal{L}_{0\text{-}1}\big(h(\mathbf{x}), y\big) \times \mathcal{L}_{0\text{-}1}\big(h'(\mathbf{x}), y\big)\,. \tag{6}$$

From the definitions of the expected disagreement and the joint error, Lacasse et al. [48] (see also [49]) observed that in a binary classification context, given a domain $\mathcal{D}$ on $\mathbf{X} \times Y$ and a distribution ρ on $\mathcal{H}$, we can decompose the Gibbs risk as

$$\mathrm{R}_{\mathcal{D}}(G_\rho) \;=\; \frac{1}{2}\,\mathrm{d}_{\mathcal{D}_{\mathbf{X}}}(\rho) + \mathrm{e}_{\mathcal{D}}(\rho)\,. \tag{7}$$

Indeed, since $Y = \{-1, +1\}$, we have

$$\begin{aligned}
2\,\mathrm{R}_{\mathcal{D}}(G_\rho) &= \mathop{\mathbf{E}}_{(h,h')\sim\rho^2}\ \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \Big[\mathcal{L}_{0\text{-}1}\big(h(\mathbf{x}), y\big) + \mathcal{L}_{0\text{-}1}\big(h'(\mathbf{x}), y\big)\Big] \\
&= \mathop{\mathbf{E}}_{(h,h')\sim\rho^2}\ \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \Big[\mathcal{L}_{0\text{-}1}\big(h(\mathbf{x}), h'(\mathbf{x})\big) + 2{\times}\mathcal{L}_{0\text{-}1}\big(h(\mathbf{x}), y\big)\,\mathcal{L}_{0\text{-}1}\big(h'(\mathbf{x}), y\big)\Big] \\
&= \mathrm{d}_{\mathcal{D}_{\mathbf{X}}}(\rho) + 2 \times \mathrm{e}_{\mathcal{D}}(\rho)\,.
\end{aligned}$$

Lastly, PAC-Bayesian theory allows one to bound the expected error $\mathrm{R}_{\mathcal{D}}(G_\rho)$ in terms of two major quantities: The empirical error

$$\widehat{\mathrm{R}}_S(G_\rho) \;=\; \mathop{\mathbf{E}}_{h\sim\rho}\ \widehat{\mathrm{R}}_S(h)$$

estimated on a sample $S \sim (\mathcal{D})^m$, and the Kullback-Leibler divergence

$$\mathrm{KL}(\rho\|\pi) \;=\; \mathop{\mathbf{E}}_{h\sim\rho}\ \ln\frac{\rho(h)}{\pi(h)}\,.$$

In the next section, we introduce a PAC-Bayesian theorem proposed by Catoni [50].[8]

3.2. A Usual PAC-Bayesian Theorem

Usual PAC-Bayesian theorems suggest that, in order to minimize the expected risk, a learning algorithm should perform a trade-off between the empirical risk minimization $\widehat{\mathrm{R}}_S(G_\rho)$ and KL-divergence minimization $\mathrm{KL}(\rho\,\|\,\pi)$ (roughly speaking the complexity term). The nature of this trade-off can be explicitly controlled in Theorem 3 below. This PAC-Bayesian result, first proposed by Catoni [50], is defined with a hyperparameter (here named ω). It appears to be a natural tool to design PAC-Bayesian algorithms. We present this result in the simplified form suggested by Germain et al. [54].

Theorem 3 (Catoni [50]). *For any domain $\mathcal{D}$ over $\mathbf{X} \times Y$, for any set of hypotheses $\mathcal{H}$, any prior distribution π over $\mathcal{H}$, any $\delta \in (0, 1]$, and any real number $\omega > 0$, with a probability at least $1 - \delta$ over the random choice of $S \sim (\mathcal{D})^m$, for every posterior distribution ρ on $\mathcal{H}$, we have*

$$\mathrm{R}_{\mathcal{D}}(G_\rho) \;\leq\; \frac{\omega}{1 - e^{-\omega}}\left[\widehat{\mathrm{R}}_S(G_\rho) + \frac{\mathrm{KL}(\rho\|\pi) + \ln\frac{1}{\delta}}{m \times \omega}\right]. \tag{8}$$

[8] Two other common forms of the PAC-Bayesian theorem are the one of McAllester [43] and the one of Seeger [51], Langford [52]. We refer the reader to our research report [53] for a larger variety of PAC-Bayesian theorems in a domain adaptation context.

Similarly to McAllester and Keshet [55], we could choose to restrict $\omega \in (0, 2)$ to obtain a slightly looser but simpler bound. Using $e^{-\omega} \leq 1 - \omega - \frac{1}{2}\omega^2$ to upper-bound on the right-hand side of Equation (8), we obtain

$$\mathrm{R}_{\mathcal{D}}(G_\rho) \;\leq\; \frac{1}{1-\frac{1}{2}\omega}\left[\widehat{\mathrm{R}}_S(G_\rho) + \frac{\mathrm{KL}(\rho\|\pi) + \ln\frac{1}{\delta}}{m \times \omega}\right]. \tag{9}$$

The bound of Theorem 3—in both forms of Equations (8) and (9)—has two appealing characteristics. First, choosing $\omega = 1/\sqrt{m}$, the bound becomes consistent: It converges to $1 \times \left[\,\widehat{\mathrm{R}}_S(G_\rho) + 0\,\right]$ as m grows. Second, as described in Section 3.3, its minimization is closely related to the minimization problem associated with the Support Vector Machine (SVM) algorithm when ρ is an isotropic Gaussian over the space of linear classifiers [44]. Hence, the value ω allows us to control the trade-off between the empirical risk $\widehat{\mathrm{R}}_S(G_\rho)$ and the "complexity term" $\frac{1}{m}\,\mathrm{KL}(\rho\|\pi)$.

3.3. Supervised PAC-Bayesian Learning of Linear Classifiers

Let us consider $\mathcal{H}$ as a set of linear classifiers in a d-dimensional space. Each $h_{\mathbf{w}'} \in \mathcal{H}$ is defined by a weight vector $\mathbf{w}' \in \mathbb{R}^d$:

$$h_{\mathbf{w}'}(\mathbf{x}) \;=\; \mathrm{sign}\left(\mathbf{w}' \cdot \mathbf{x}\right),$$

where $\cdot$ denotes the dot product.

By restricting the prior and the posterior distributions over $\mathcal{H}$ to be Gaussian distributions, Langford and Shawe-Taylor [56] have specialized the PAC-Bayesian theory in order to bound the expected risk of any linear classifier $h_{\mathbf{w}} \in \mathcal{H}$. More precisely, given a prior $\pi_{\mathbf{0}}$ and a posterior $\rho_{\mathbf{w}}$ defined as spherical Gaussians with identity covariance matrix respectively centered on vectors $\mathbf{0}$ and $\mathbf{w}$, for any $h_{\mathbf{w}'} \in \mathcal{H}$, we have

$$\pi_{\mathbf{0}}(h_{\mathbf{w}'}) \;=\; \left(\frac{1}{\sqrt{2\boldsymbol{\pi}}}\right)^d \exp\left(-\frac{1}{2}\|\mathbf{w}'\|^2\right),$$

$$\text{and} \quad \rho_{\mathbf{w}}(h_{\mathbf{w}'}) \;=\; \left(\frac{1}{\sqrt{2\boldsymbol{\pi}}}\right)^d \exp\left(-\frac{1}{2}\|\mathbf{w}' - \mathbf{w}\|^2\right).$$

An interesting property of these distributions—also seen as multivariate normal distributions, $\pi_{\mathbf{0}} = \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\rho_{\mathbf{w}} = \mathcal{N}(\mathbf{w}, \mathbf{I})$—is that the prediction of the $\rho_{\mathbf{w}}$-weighted majority vote $B_{\rho_{\mathbf{w}}}$ coincides with the one of the linear classifier $h_{\mathbf{w}}$. Indeed, we have

$$\begin{aligned}\forall\, \mathbf{x} \in \mathbf{X},\ \forall\, \mathbf{w} \in \mathcal{H}, \quad h_{\mathbf{w}}(\mathbf{x}) \;&=\; B_{\rho_{\mathbf{w}}}(\mathbf{x}) \\ &=\; \mathrm{sign}\left[\mathop{\mathbf{E}}_{h_{\mathbf{w}'}\sim\rho_{\mathbf{w}}} h_{\mathbf{w}'}(\mathbf{x})\right].\end{aligned}$$

Moreover, the expected risk of the Gibbs classifier $G_{\rho_{\mathbf{w}}}$ on a domain $\mathcal{D}$ is then given by[9]

$$\mathrm{R}_{\mathcal{D}}(G_{\rho_{\mathbf{w}}}) \;=\; \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim P_S} \Phi_{\mathrm{R}}\left(y\,\frac{\mathbf{w}\cdot\mathbf{x}}{\|\mathbf{x}\|}\right), \tag{10}$$

where

$$\Phi_{\mathrm{R}}(x) \;=\; \tfrac{1}{2}\left[1-\mathrm{Erf}\left(\tfrac{x}{\sqrt{2}}\right)\right], \tag{11}$$

with Erf is the Gauss error function defined as

$$\mathrm{Erf}\,(x) \;=\; \frac{2}{\sqrt{\boldsymbol{\pi}}}\int_0^x \exp\left(-t^2\right)\mathrm{d}t\,. \tag{12}$$

Here, $\Phi_{\mathrm{R}}(x)$ can be seen as a *smooth* surrogate of the 0-1-loss function $\mathrm{I}\,[x\leq 0]$ relying on $y\,\frac{\mathbf{w}\cdot\mathbf{x}}{\|\mathbf{x}\|}$. This function Φ_{R} is sometimes called the *probit–loss* [*e.g.*, 55]. It is worth noting that $\|\mathbf{w}\|$ plays an important role on the value of $\mathrm{R}_{\mathcal{D}}(G_{\rho_{\mathbf{w}}})$, but not on $\mathrm{R}_{\mathcal{D}}(h_{\mathbf{w}})$. Indeed, $\mathrm{R}_{\mathcal{D}}(G_{\rho_{\mathbf{w}}})$ tends to $\mathrm{R}_{\mathcal{D}}(h_{\mathbf{w}})$ as $\|\mathbf{w}\|$ grows, which can provide *very tight bounds* (see the empirical analyses of [57, 44]). Finally, the KL-divergence between $\rho_{\mathbf{w}}$ and $\pi_{\mathbf{0}}$ becomes simply

$$\begin{aligned}\mathrm{KL}\big(\rho_{\mathbf{w}}\|\pi_{\mathbf{0}}\big) \;&=\; \mathrm{KL}\big(\mathcal{N}(\mathbf{w},\mathbf{I})\,\|\,\mathcal{N}(\mathbf{0},\mathbf{I})\big)\\ &=\; \tfrac{1}{2}\|\mathbf{w}\|^2\,,\end{aligned}$$

and turns out to be a measure of *complexity* of the learned classifier.

3.3.1. Objective Function and Gradient

Based on the specialization of the PAC-Bayesian theory to linear classifiers, Germain et al. [44] suggested minimizing a PAC-Bayesian bound on $\mathrm{R}_{\mathcal{D}}(G_{\rho_{\mathbf{w}}})$. For sake of completeness, we provide here more mathematical details than in the original conference paper [44]. In forthcoming Section 6, we will extend this supervised learning algorithm to the domain adaptation setting.

Given a sample $S=\{(\mathbf{x}_i,y_i)\}_{i=1}^m$ and a hyperparameter $\Omega>0$, the learning algorithm performs gradient descent in order to find an optimal weight vector $\mathbf{w}$ that minimizes

$$\begin{aligned}F(\mathbf{w}) \;&=\; \Omega\, m\,\widehat{\mathrm{R}}_S(G_{\rho_{\mathbf{w}}}) + \mathrm{KL}(\rho_{\mathbf{w}}\|\pi_{\mathbf{0}})\\ &=\; \Omega\sum_{i=1}^m \Phi_{\mathrm{R}}\left(y\frac{\mathbf{w}\cdot\mathbf{x}_i}{\|\mathbf{x}_i\|}\right) + \frac{1}{2}\|\mathbf{w}\|^2\,.\end{aligned} \tag{13}$$

It turns out that the optimal vector $\mathbf{w}$ corresponds to the distribution $\rho_{\mathbf{w}}$ minimizing the value of the bound on $\mathrm{R}_{\mathcal{D}}(G_{\rho_{\mathbf{w}}})$ given by Theorem 3, with the parameter ω of the theorem being the hyperparameter Ω of the learning

[9] The calculations leading to Equation (10) can be found in Langford [52]. For sake of completeness, we provide a slightly different derivation in Appendix B.

algorithm. It is important to point out that PAC-Bayesian theorems bound simultaneously $\mathrm{R}_{\mathcal{D}}(G_{\rho_{\mathbf{w}}})$ *for every* $\rho_{\mathbf{w}}$ *on* $\mathcal{H}$. Therefore, one can "freely" explore the domain of objective function F to choose a posterior distribution $\rho_{\mathbf{w}}$ that gives, thanks to Theorem 3, a bound valid with probability $1-\delta$.

The minimization of Equation (13) by gradient descent corresponds to the learning algorithm called PBGD3 of Germain et al. [44]. The gradient of $F(\mathbf{w})$ is given the vector $\nabla F(\mathbf{w})$:

$$\nabla F(\mathbf{w}) \;=\; \Omega \sum_{i=1}^{m} \Phi'_{\mathrm{R}}\left(y_i \frac{\mathbf{w}\cdot\mathbf{x}_i}{\|\mathbf{x}_i\|}\right) \frac{y_i\,\mathbf{x}_i}{\|\mathbf{x}_i\|} + \mathbf{w}\,,$$

where $\Phi'_{\mathrm{R}}(x) = -\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}x^2\right)$ is the derivative of Φ_{R} at point x.

Similarly to SVM, the learning algorithm PBGD3 realizes a trade-off between the empirical risk—expressed by the loss Φ_{R}—and the complexity of the learned linear classifier—expressed by the regularizer $\|\mathbf{w}\|^2$. This similarity increases when we use a kernel function, as described next.

3.3.2. Using a Kernel Function

The kernel trick allows substituting inner products by a kernel function $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ in Equation (13). If k is a Mercer kernel, it implicitly represents a function $\phi : X \to \mathbb{R}^{d'}$ that maps an example of $\mathbf{X}$ into an arbitrary d'-dimensional space, such that

$$\forall(\mathbf{x},\mathbf{x}') \in \mathbf{X}^2, \quad k(\mathbf{x},\mathbf{x}') \;=\; \phi(\mathbf{x})\cdot\phi(\mathbf{x}')\,.$$

Then, a dual weight vector $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_m) \in \mathbb{R}^m$ encodes the linear classifier $\mathbf{w} \in \mathbb{R}^{d'}$ as a linear combination of examples of S:

$$\mathbf{w} \;=\; \sum_{i=1}^{m} \alpha_i\,\phi(\mathbf{x}_i)\,, \qquad \text{and thus} \qquad h_{\mathbf{w}}(\mathbf{x}) \;=\; \operatorname{sign}\left[\sum_{i=1}^{m} \alpha_i k(\mathbf{x}_i,\mathbf{x})\right].$$

By the representer theorem [58], the vector $\mathbf{w}$ minimizing Equation (13) can be recovered by finding the vector $\boldsymbol{\alpha}$ that minimizes

$$F(\boldsymbol{\alpha}) \;=\; C\sum_{i=1}^{m} \Phi_{\mathrm{R}}\left(y_i \frac{\sum_{j=1}^{m}\alpha_j K_{i,j}}{\sqrt{K_{i,i}}}\right) + \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i\alpha_j K_{i,j}\,, \tag{14}$$

where K is the kernel matrix of size $m \times m$.[10] That is, $K_{i,j} = k(\mathbf{x}_i,\mathbf{x}_j)$. The gradient of $F(\boldsymbol{\alpha})$ is simply given the vector $\nabla F(\boldsymbol{\alpha}) = (\alpha'_1, \alpha'_2, \ldots, \alpha'_m)$, with

$$\alpha'_{\#} \;=\; \Omega \sum_{i=1}^{m} \Phi_{\mathrm{R}}\left(y_i \frac{\sum_{j=1}^{m}\alpha_j K_{i,j}}{\sqrt{K_{i,i}}}\right) \frac{y_i\,K_{i,\#}}{\sqrt{K_{i,i}}} + \sum_{j=1}^{m} \alpha_i K_{i,\#}\,,$$

[10] It is non-trivial to show that the kernel trick holds when $\pi_{\mathbf{0}}$ and $\rho_{\mathbf{w}}$ are Gaussian over infinite-dimensional feature space. As mentioned by McAllester and Keshet [55], it is, however, the case provided we consider Gaussian processes as measure of distributions $\pi_{\mathbf{0}}$ and $\rho_{\mathbf{w}}$ over (infinite) $\mathcal{H}$. The same analysis holds for the kernelized versions of the two forthcoming domain adaptation algorithms (Section 6.3.3).

for $\# \in \{1, 2, \ldots, m\}$.

3.3.3. Improving the Algorithm Using a Convex Objective

An annoying drawback of PBGD3 is that the objective function is non-convex and the gradient descent implementation needs many random restarts. In fact, we made extensive empirical experiments after the ones described by Germain et al. [44] and saw that PBGD3 achieves an equivalent accuracy (and at a fraction of the running time) by replacing the loss function Φ_{R} of Equations (13) and (14) by its convex relaxation, which is

$$\begin{aligned} \widetilde{\Phi}_{\mathrm{R}}(x) &= \max\left\{\Phi_{\mathrm{R}}(x),\, \frac{1}{2} - \frac{x}{\sqrt{2\pi}}\right\} \\ &= \begin{cases} \dfrac{1}{2} - \dfrac{x}{\sqrt{2\pi}} & \text{if } x \leq 0, \\ \Phi_{\mathrm{R}}(x) & \text{otherwise.} \end{cases} \end{aligned} \tag{15}$$

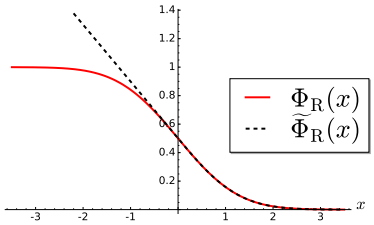
The derivative of $\widetilde{\Phi}_{\mathrm{R}}$ at point x is then $\widetilde{\Phi}'_{\mathrm{R}}(x) = \Phi'_{\mathrm{R}}(\max\{0, x\})$; In other words, $\widetilde{\Phi}'_{\mathrm{R}}(x) = -1/\sqrt{2\pi}$ if $x < 0$, and $\Phi'_{\mathrm{R}}(x)$ otherwise. Figure 1a illustrates the functions Φ_{R} and $\widetilde{\Phi}_{\mathrm{R}}$. Note that the latter can be interpreted as a *smooth* version the SVM's hinge loss, $\max\{0, 1 - x\}$. The toy experiment of Figure 1d (described in the next subsection) provides another empirical evidence that the minima of Φ_{R} and $\widetilde{\Phi}_{\mathrm{R}}$ tend to coincide.

3.3.4. Illustration on a Toy Dataset

To illustrate the trade-off coming into play in PBGD3 algorithm (and its convexified version), we conduct a small experiment on a two-dimensional toy dataset. That is, we generate 100 positive examples according to a Gaussian of mean $(-1, -1)$ and 100 negative examples generated by a Gaussian of mean $(-1, +1)$ (both of these Gaussian have a unit variance), as shown by Figure 1c. We then compute the risks associated with linear classifiers $h_{\mathbf{w}}$, with $\mathbf{w} = \|\mathbf{w}\|(\cos\theta, \sin\theta) \in \mathbb{R}^2$. Figure 1d shows the risks of three different classifiers for $\|\mathbf{w}\| \in \{1, 2, 5\}$, while rotating the decision boundary $\theta \in [-\pi, +\pi]$ around the origin. The 0-1-loss associated with the majority vote classifier $\widehat{\mathrm{R}}_S(B_{\rho_{\mathbf{w}}})$ does not rely on the norm $\|\mathbf{w}\|$. However, we clearly see that probit-loss of the Gibbs classifier $\widehat{\mathrm{R}}_S(G_{\rho_{\mathbf{w}}})$ converges to $\widehat{\mathrm{R}}_S(B_{\rho_{\mathbf{w}}})$ as $\|\mathbf{w}\|$ increases (the dashed lines correspond to the convex surrogate of the probit-loss given by Equation (15)). Thus, thanks to the specialization of to the linear classifier, the *smoothness* of the surrogate loss is regularized by the norm $\|\mathbf{w}\|^2$.

4. Two New Domain Adaptation Bounds

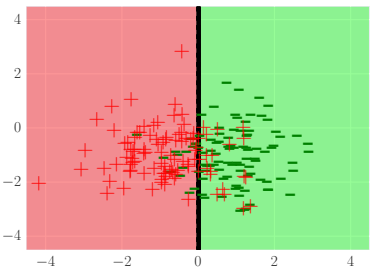
The originality of our contribution is to theoretically design two domain adaptation frameworks suitable for the PAC-Bayesian approach. In Section 4.1, we first follow the spirit of the seminal works recalled in Section 2 by proving a similar trade-off for the Gibbs classifier. Then in Section 4.2, we propose



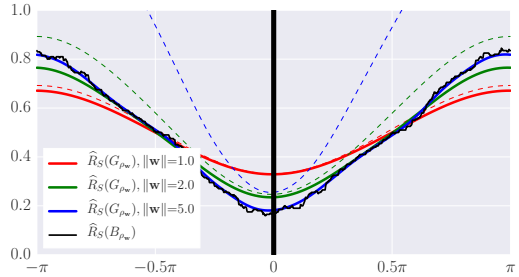
(a) Loss functions for linear classifiers.

FUNCTION	DERIVATIVE
$\Phi_{\mathrm{R}}(x) = \frac{1}{2}\left[1 - \mathrm{Erf}\left(\frac{x}{\sqrt{2}}\right)\right]$	$\Phi'_{\mathrm{R}}(x) = -\frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$
$\widetilde{\Phi}_{\mathrm{R}}(x) = \max\left\{\Phi_{\mathrm{R}}(x), \frac{1}{2} - \frac{x}{\sqrt{2\pi}}\right\}$	$\widetilde{\Phi}'_{\mathrm{R}}(x) = \Phi'_{\mathrm{R}}\left(\max\{0, x\}\right)$

(b) Loss functions definitions and their derivatives.



(c) Toy dataset, and the decision boundary for $\theta = 0$ (matching the vertical line of Figure (d)).



(d) Risk values according to θ, for $\|\mathbf{w}\| \in \{1, 2, 5\}$. Each dashed line shows convex counterpart of the continuous line of the same color.

Figure 1: Understanding PBGD3 supervised learning algorithm in terms of loss functions. Upper Figures (a-b) show the loss functions, and lower Figures (c-d) illustrate the behavior on a toy dataset.

a novel trade-off based on the specificities of the Gibbs classifier that come from Equation (7). Note that both results relies on the notion of expected disagreement of binary classifiers (Equation 5). Consequently, our analysis does not directly extend to multi-class prediction and regression frameworks.

4.1. In the Spirit of the Seminal Works

In the following, while the domain adaptation bounds presented in Section 2 focus on a single classifier, we first define a ρ-average divergence measure to compare the marginals. This leads us to derive our first domain adaptation bound.

4.1.1. A Domains' Divergence for PAC-Bayesian Analysis

As discussed in Section 2.2, the derivation of generalization ability in domain adaptation critically needs a divergence measure between the source and target marginals. For the PAC-Bayesian setting, we propose a *domain disagreement pseudometric*[11] to measure the structural difference between domain marginals in terms of posterior distribution ρ over $\mathcal{H}$. Since we are interested in learning

[11] A pseudometric d is a metric for which the property $d(x, y) = 0 \iff x = y$ is relaxed to $d(x, y) = 0 \impliedby x = y$.

a ρ-weighted majority vote B_ρ leading to good generalization guarantees, we propose to follow the idea spurred by the C-bound of Equation (4): Given a source domain $\mathcal{S}$, a target domain $\mathcal{T}$, and a posterior distribution ρ, if $\mathrm{R}_{\mathcal{S}}(G_\rho)$ and $\mathrm{R}_{\mathcal{T}}(G_\rho)$ are similar, then $\mathrm{R}_{\mathcal{S}}(B_\rho)$ and $\mathrm{R}_{\mathcal{T}}(B_\rho)$ are similar when $\mathrm{d}_{\mathcal{S}_\mathbf{X}}(\rho)$ and $\mathrm{d}_{\mathcal{T}_\mathbf{X}}(\rho)$ are also similar. Thus, the domains $\mathcal{S}$ and $\mathcal{T}$ are close according to ρ if the expected disagreement over the two domains tends to be close. We then define our pseudometric as follows.

Definition 1. *Let $\mathcal{H}$ be a hypothesis class. For any marginal distributions $\mathcal{S}_\mathbf{X}$ and $\mathcal{T}_\mathbf{X}$ over $\mathbf{X}$, any distribution ρ on $\mathcal{H}$, the domain disagreement* $\mathrm{dis}_\rho(\mathcal{S}_\mathbf{X}, \mathcal{T}_\mathbf{X})$ *between $\mathcal{S}_\mathbf{X}$ and $\mathcal{T}_\mathbf{X}$ is defined by*

$$\begin{aligned}\mathrm{dis}_\rho(\mathcal{S}_\mathbf{X}, \mathcal{T}_\mathbf{X}) &= \left| \mathrm{d}_{\mathcal{T}_\mathbf{X}}(\rho) - \mathrm{d}_{\mathcal{S}_\mathbf{X}}(\rho) \right| \\ &= \left| \mathop{\mathbf{E}}_{(h,h')\sim\rho^2} \left[\mathrm{R}_{\mathcal{T}_\mathbf{X}}(h,h') - \mathrm{R}_{\mathcal{S}_\mathbf{X}}(h,h') \right] \right| .\end{aligned}$$

Note that dis_ρ is symmetric and fulfills the triangle inequality.

4.1.2. Comparison of the $\mathcal{H}\Delta\mathcal{H}$-divergence and our domain disagreement

While the $\mathcal{H}\Delta\mathcal{H}$-divergence of Theorem 1 is difficult to jointly optimize with the empirical source error, our empirical disagreement measure is easier to manipulate: We simply need to compute the ρ-average of the classifiers disagreement instead of finding the pair of classifiers that maximizes the disagreement. Indeed, $\mathrm{dis}_\rho(\mathcal{S}_\mathbf{X}, \mathcal{T}_\mathbf{X})$ depends on the majority vote, which suggests that we can directly minimize it via its empirical counterpart. This can be done without instance reweighting, space representation changing or family of classifiers modification. On the contrary, $\frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}_\mathbf{X}, \mathcal{T}_\mathbf{X})$ is a supremum over all $h \in \mathcal{H}$ and hence, does not depend on the classifier on which the risk is considered. Moreover, $\mathrm{dis}_\rho(\mathcal{S}_\mathbf{X}, \mathcal{T}_\mathbf{X})$ (the ρ-average) is lower than the $\frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}_\mathbf{X}, \mathcal{T}_\mathbf{X})$ (the worst-case). Indeed, for every $\mathcal{H}$ and ρ over $\mathcal{H}$, we have

$$\begin{aligned}\tfrac{1}{2}\, d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}_\mathbf{X}, \mathcal{T}_\mathbf{X}) &= \sup_{(h,h')\in\mathcal{H}^2} \left| \mathrm{R}_{\mathcal{T}_\mathbf{X}}(h,h') - \mathrm{R}_{\mathcal{S}_\mathbf{X}}(h,h') \right| \\ &\geq \mathop{\mathbf{E}}_{(h,h')\sim\rho^2} \left| \mathrm{R}_{\mathcal{T}_\mathbf{X}}(h,h') - \mathrm{R}_{\mathcal{S}_\mathbf{X}}(h,h') \right| \\ &\geq \mathrm{dis}_\rho(\mathcal{S}_\mathbf{X}, \mathcal{T}_\mathbf{X}) .\end{aligned}$$

4.1.3. A Domain Adaptation Bound for the Stochastic Gibbs Classifier

We now derive our first main result in the following theorem: A domain adaptation bound relevant in a PAC-Bayesian setting, and that relies on the domain disagreement of Definition 1.

Theorem 4. *Let $\mathcal{H}$ be a hypothesis class. We have*

$$\forall \rho \text{ on } \mathcal{H},\ \mathrm{R}_{\mathcal{T}}(G_\rho) \leq \mathrm{R}_{\mathcal{S}}(G_\rho) + \frac{1}{2}\mathrm{dis}_\rho(\mathcal{S}_\mathbf{X}, \mathcal{T}_\mathbf{X}) + \lambda_\rho ,$$

where λ_ρ is the deviation between the expected joint errors (Equation 6) of G_ρ on the target and source domains:

$$\lambda_\rho \quad = \quad \Big| \mathrm{e}_\mathcal{T}(\rho) - \mathrm{e}_\mathcal{S}(\rho) \Big| \,. \tag{16}$$

Proof. First, from Equation (7), we recall that, given a domain $\mathcal{D}$ on $\mathbf{X} \times Y$ and a distribution ρ over $\mathcal{H}$, we have

$$\mathrm{R}_\mathcal{D}(G_\rho) \;=\; \frac{1}{2}\mathrm{d}_{\mathcal{D}_\mathbf{X}}(\rho) + \mathrm{e}_\mathcal{D}(\rho) \,.$$

Therefore,

$$\begin{aligned}
\mathrm{R}_\mathcal{T}(G_\rho) - \mathrm{R}_\mathcal{S}(G_\rho) &= \frac{1}{2}\Big(\mathrm{d}_{\mathcal{T}_\mathbf{X}}(\rho) - \mathrm{d}_{\mathcal{S}_\mathbf{X}}(\rho)\Big) + \Big(\mathrm{e}_\mathcal{T}(\rho) - \mathrm{e}_\mathcal{S}(\rho)\Big) \\
&\leq \frac{1}{2}\Big|\mathrm{d}_{\mathcal{T}_\mathbf{X}}(\rho) - \mathrm{d}_{\mathcal{S}_\mathbf{X}}(\rho)\Big| + \Big|\mathrm{e}_\mathcal{T}(\rho) - \mathrm{e}_\mathcal{S}(\rho)\Big| \\
&= \frac{1}{2}\,\mathrm{dis}_\rho(\mathcal{S}_\mathbf{X}, \mathcal{T}_\mathbf{X}) + \lambda_\rho \,.
\end{aligned}$$

■

4.1.4. Meaningful Quantities

Similar to the bounds of Theorems 1 and 2, our bound can be seen as a trade-off between different quantities. Concretely, the terms $\mathrm{R}_\mathcal{S}(G_\rho)$ and $\mathrm{dis}_\rho(\mathcal{S}_\mathbf{X}, \mathcal{T}_\mathbf{X})$ are akin to the first two terms of the domain adaptation bound of Theorem 1: $\mathrm{R}_\mathcal{S}(G_\rho)$ is the ρ-average risk over $\mathcal{H}$ on the source domain, and $\mathrm{dis}_\rho(\mathcal{T}_\mathbf{X}, \mathcal{S}_\mathbf{X})$ measures the ρ-average disagreement between the marginals but is specific to the current model depending on ρ. The other term λ_ρ measures the deviation between the expected joint target and source errors of G_ρ. According to this theory, a good domain adaptation is possible if this deviation is low. However, since we suppose that we do not have any label in the target sample, we cannot control or estimate it. In practice, we suppose that λ_ρ is low and we neglect it. In other words, we assume that the labeling information between the two domains is related and that considering only the marginal agreement and the source labels is sufficient to find a good majority vote. Another important point is that the above theorem improves the one we proposed in Germain et al. [1] with regard to two aspects.[12] On the one hand, this bound is not degenerated when the source and target distributions are the same or close. On the other hand, our result contains only half of $\mathrm{dis}_\rho(\mathcal{S}_\mathbf{X}, \mathcal{T}_\mathbf{X})$, contrary to our first bound proposed in Germain et al. [1]. Finally, due to the dependence of $\mathrm{dis}_\rho(\mathcal{T}_\mathbf{X}, \mathcal{S}_\mathbf{X})$ and λ_ρ on the learned posterior, our bound is, in general incomparable with the ones of Theorems 1 and 2. However, it brings the same underlying idea: Supposing that the two domains are sufficiently related, one must look for a model that minimizes a trade-off between its source risk and a distance between the domains' marginal.

[12] More details are given in our research report [53].

4.2. A Novel Perspective on Domain Adaptation

In this section, we introduce an original approach to upper-bound the non-estimable risk of a ρ-weighted majority vote on a target domain $\mathcal{T}$ thanks to a term depending on its marginal distribution $\mathcal{T}_{\mathbf{X}}$, another one on a related source domain $\mathcal{S}$, and a term capturing the "volume" of the source distribution uninformative for the target task. We base our bound on Equation (7) (recalled below) that decomposes the Gibbs classifier into the trade-off between the half of the expected disagreement $\mathrm{d}_{\mathcal{D}_{\mathbf{X}}}(\rho)$ of Equation (5) and the expected joint error $\mathrm{e}_{\mathcal{D}}(\rho)$ of Equation (6):

$$\mathrm{R}_{\mathcal{D}}(G_\rho) \;=\; \tfrac{1}{2}\,\mathrm{d}_{\mathcal{D}_{\mathbf{X}}}(\rho) + \mathrm{e}_{\mathcal{D}}(\rho)\,. \tag{7}$$

A key observation is that the *voters' disagreement does not rely on labels*; we can compute $\mathrm{d}_{\mathcal{D}_{\mathbf{X}}}(\rho)$ using the marginal distribution $\mathcal{D}_{\mathbf{X}}$. Thus, in the present domain adaptation context, we have access to $\mathrm{d}_{\mathcal{T}_{\mathbf{X}}}(\rho)$ even if the target labels are unknown. However, the expected joint error can only be computed on the labeled source domain, that is what we kept in mind to define our new domain divergence.

4.2.1. Another Domain Divergence for the PAC-Bayesian Approach

We design a domains' divergence that allows us to link the target joint error $\mathrm{e}_{\mathcal{T}}(\rho)$ with the source one $\mathrm{e}_{\mathcal{S}}(\rho)$ by reweighting the latter. This new divergence is called the β_q-divergence and is parametrized by a real value $q > 0$:

$$\beta_q(\mathcal{T}\|\mathcal{S}) \;=\; \left[\,\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{S}}\left(\frac{\mathcal{T}(\mathbf{x},y)}{\mathcal{S}(\mathbf{x},y)}\right)^{q}\,\right]^{\frac{1}{q}}. \tag{17}$$

It is worth noting that considering some q values allow us to recover well-known divergences. For instance, choosing $q=2$ relates our result to the χ^2-distance, between the domains as $\beta_2(\mathcal{T}\|\mathcal{S}) = \sqrt{\chi^2(\mathcal{T}\|\mathcal{S})+1}$. Moreover, we can link $\beta_q(\mathcal{T}\|\mathcal{S})$ to the Rényi divergence,[13] which has led to generalization bounds in the specific context of importance weighting [15]. We denote the limit case $q \to \infty$ by

$$\beta_\infty(\mathcal{T}\|\mathcal{S}) \;=\; \sup_{(\mathbf{x},y)\in\textsc{supp}(\mathcal{S})}\left(\frac{\mathcal{T}(\mathbf{x},y)}{\mathcal{S}(\mathbf{x},y)}\right), \tag{18}$$

with $\textsc{supp}(\mathcal{S})$ the support of the domain $\mathcal{S}$. The β_q-divergence handles the input space areas where the source domain support and the target domain support $\textsc{supp}(\mathcal{T})$ intersect. It seems reasonable to assume that, when adaptation is achievable, such areas are fairly large. However, it is likely that $\textsc{supp}(\mathcal{T})$ is *not entirely* included in $\textsc{supp}(\mathcal{S})$. We denote $\mathcal{T}\backslash\mathcal{S}$ the distribution of $(\mathbf{x},y)\sim\mathcal{T}$ conditional to $(\mathbf{x},y) \in \textsc{supp}(\mathcal{T})\backslash\textsc{supp}(\mathcal{S})$. Since it is hardly conceivable to

[13] For $q \geq 0$, we can easily show $\beta_q(\mathcal{T}\|\mathcal{S}) = 2^{\frac{q-1}{q}D_q(\mathcal{T}\|\mathcal{S})}$, where $D_q(\mathcal{T}\|\mathcal{S})$ is the Rényi divergence between $\mathcal{T}$ and $\mathcal{S}$.

estimate the joint error $\mathrm{e}_{\mathcal{T}\setminus\mathcal{S}}(\rho)$ without making extra assumptions, we need to define the worst possible risk for this *unknown* area,

$$\eta_{\mathcal{T}\setminus\mathcal{S}} = \Pr_{(\mathbf{x},y)\sim\mathcal{T}}\Big((\mathbf{x},y)\notin \textsc{supp}(\mathcal{S})\Big)\ \sup_{h\in\mathcal{H}} \mathrm{R}_{\mathcal{T}\setminus\mathcal{S}}(h)\,. \tag{19}$$

Even if we cannot evaluate $\sup_{h\in\mathcal{H}} \mathrm{R}_{\mathcal{T}\setminus\mathcal{S}}(h)$, the value of $\eta_{\mathcal{T}\setminus\mathcal{S}}$ is necessarily lower than $\Pr_{(\mathbf{x},y)\sim\mathcal{T}}\big((\mathbf{x},y)\notin \textsc{supp}(\mathcal{S})\big)$.

4.2.2. A Novel Domain Adaptation Bound

Let us state the result underlying the novel domain adaptation perspective of this paper.

Theorem 5. *Let $\mathcal{H}$ be a hypothesis space, let $\mathcal{S}$ and $\mathcal{T}$ respectively be the source and the target domains on $\mathbf{X}\times Y$. Let $q>0$ be a constant. We have,*

$$\forall \rho \text{ on } \mathcal{H},\quad \mathrm{R}_{\mathcal{T}}(G_\rho) \le \frac{1}{2}\,\mathrm{d}_{\mathcal{T}_\mathbf{X}}(\rho) + \beta_q(\mathcal{T}\|\mathcal{S})\times\Big[\mathrm{e}_{\mathcal{S}}(\rho)\Big]^{1-\frac{1}{q}} + \eta_{\mathcal{T}\setminus\mathcal{S}}\,,$$

where $\mathrm{d}_{\mathcal{T}_\mathbf{X}}(\rho)$, $\mathrm{e}_{\mathcal{S}}(\rho)$, $\beta_q(\mathcal{T}\|\mathcal{S})$ and $\eta_{\mathcal{T}\setminus\mathcal{S}}$ are respectively defined by Equations (5), (6), (17) *and* (19).

Proof. From Equation (7), we know that $\mathrm{R}_{\mathcal{T}}(G_\rho) = \frac{1}{2}\,\mathrm{d}_{\mathcal{T}_\mathbf{X}}(\rho) + \mathrm{e}_{\mathcal{T}}(\rho)$. Let us split $\mathrm{e}_{\mathcal{T}}(\rho)$ in two parts:

$$\begin{aligned}
\mathrm{e}_{\mathcal{T}}(\rho) &= \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{T}}\ \mathop{\mathbf{E}}_{(h,h')\sim\rho^2} \mathcal{L}_{0\text{-}1}\left(h(\mathbf{x}),y\right)\mathcal{L}_{0\text{-}1}\left(h'(\mathbf{x}),y\right)\\
&= \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{S}}\ \frac{\mathcal{T}(\mathbf{x},y)}{\mathcal{S}(\mathbf{x},y)}\ \mathop{\mathbf{E}}_{(h,h')\sim\rho^2} \mathcal{L}_{0\text{-}1}\left(h(\mathbf{x}),y\right)\mathcal{L}_{0\text{-}1}\left(h'(\mathbf{x}),y\right) &\text{(20)}\\
&\quad + \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{T}} \mathrm{I}\left[(\mathbf{x},y)\notin \textsc{supp}(\mathcal{S})\right]\ \mathop{\mathbf{E}}_{(h,h')\sim\rho^2} \mathcal{L}_{0\text{-}1}\left(h(\mathbf{x}),y\right)\mathcal{L}_{0\text{-}1}\left(h'(\mathbf{x}),y\right). &\text{(21)}
\end{aligned}$$

(i) On the one hand, we upper-bound the first part (Line 20) using the Hölder's inequality (see Lemma 1 in Appendix A), with p such that $\frac{1}{p}=1-\frac{1}{q}$:

$$\begin{aligned}
&\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{S}}\ \frac{\mathcal{T}(\mathbf{x},y)}{\mathcal{S}(\mathbf{x},y)}\ \mathop{\mathbf{E}}_{(h,h')\sim\rho^2} \mathcal{L}_{0\text{-}1}\left(h(\mathbf{x}),y\right)\mathcal{L}_{0\text{-}1}\left(h'(\mathbf{x}),y\right)\\
&\le \left[\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{S}}\left(\frac{\mathcal{T}(\mathbf{x},y)}{\mathcal{S}(\mathbf{x},y)}\right)^q\right]^{\frac{1}{q}}\left[\mathop{\mathbf{E}}_{(h,h')\sim\rho^2}\ \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{S}}\ \left[\mathcal{L}_{0\text{-}1}\left(h(\mathbf{x}),y\right)\mathcal{L}_{0\text{-}1}\left(h'(\mathbf{x}),y\right)\right]^p\right]^{\frac{1}{p}}\\
&= \beta_q(\mathcal{T}\|\mathcal{S})\ \Big[\mathrm{e}_{\mathcal{S}}(\rho)\Big]^{\frac{1}{p}},
\end{aligned}$$

where we have removed the exponent from expression $[\mathcal{L}_{0\text{-}1}(h(\mathbf{x}),y)\mathcal{L}_{0\text{-}1}(h'(\mathbf{x}),y)]^p$ without affecting its value, which is either 1 or 0.

(ii) On the other hand, we upper-bound the second part (Line 21) by the term $\eta_{\mathcal{T}\setminus\mathcal{S}}$:

$$
\begin{aligned}
&\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{T}}\left(\mathrm{I}\left[(\mathbf{x},y)\notin \textsc{supp}(\mathcal{S})\right]\mathop{\mathbf{E}}_{(h,h')\sim\rho^2}\mathcal{L}_{0\text{-}1}\left(h(\mathbf{x}),y\right)\mathcal{L}_{0\text{-}1}\left(h'(\mathbf{x}),y\right)\right)\\
&= \left(\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{T}}\mathrm{I}[(\mathbf{x},y)\notin\textsc{supp}(\mathcal{S})]\right)\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{T}\setminus\mathcal{S}}\mathop{\mathbf{E}}_{(h,h')\sim\rho^2}\mathcal{L}_{0\text{-}1}\left(h(\mathbf{x}),y\right)\,\mathcal{L}_{0\text{-}1}\left(h'(\mathbf{x}),y\right)\\
&= \left(\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{T}}\mathrm{I}[(\mathbf{x},y)\notin\textsc{supp}(\mathcal{S})]\right)\mathrm{e}_{\mathcal{T}\setminus\mathcal{S}}(\rho)\\
&= \left(\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{T}}\mathrm{I}[(\mathbf{x},y)\notin\textsc{supp}(\mathcal{S})]\right)\left(\mathrm{R}_{\mathcal{T}\setminus\mathcal{S}}(G_\rho)-\tfrac{1}{2}\mathrm{d}_{\mathcal{T}\setminus\mathcal{S}}(\rho)\right)\\
&\leq \left(\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{T}}\mathrm{I}[(\mathbf{x},y)\notin\textsc{supp}(\mathcal{S})]\right)\sup_{h\in\mathcal{H}}\mathrm{R}_{\mathcal{T}\setminus\mathcal{S}}(h) \;=\; \eta_{\mathcal{T}\setminus\mathcal{S}}\,.
\end{aligned}
$$

■

Note that the bound of Theorem 5 is reached whenever the domains are equal ($\mathcal{S}=\mathcal{T}$). Thus, when adaptation is not necessary, our analysis is still sound and non-degenerated:

$$
\begin{aligned}
\mathrm{R}_{\mathcal{S}}(G_\rho) \;=\; \mathrm{R}_{\mathcal{T}}(G_\rho) \;&\leq\; \tfrac{1}{2}\,\mathrm{d}_{\mathcal{T}_{\mathbf{X}}}(\rho)+1\times\left[\mathrm{e}_{\mathcal{S}}(\rho)\right]^1+0\\
&=\; \tfrac{1}{2}\,\mathrm{d}_{\mathcal{S}_{\mathbf{X}}}(\rho)+\mathrm{e}_{\mathcal{S}}(\rho)=\;\mathrm{R}_{\mathcal{S}}(G_\rho)\,.
\end{aligned}
$$

4.2.3. Meaningful Quantities and Connection with Some Domain Adaptation Assumptions

Similarly to the previous results recalled in Section 2, our domain adaptation theorem bounds the target risk by a sum of three terms. However, our approach breaks the problem into *atypical* quantities:

(i) The expected disagreement $\mathrm{d}_{\mathcal{T}_{\mathbf{X}}}(\rho)$ captures *second degree* information about the target domain (without any label).

(ii) The β_q-divergence $\beta_q(\mathcal{T}\|\mathcal{S})$ is not an additional term: It weighs the influence of the expected joint error $\mathrm{e}_{\mathcal{S}}(\rho)$ of the source domain; the parameter q allows us to consider different relationships between $\beta_q(\mathcal{T}\|\mathcal{S})$ and $\mathrm{e}_{\mathcal{S}}(\rho)$.

(iii) The term $\eta_{\mathcal{T}\setminus\mathcal{S}}$ quantifies the worst feasible target error on the regions where the source domain is uninformative for the target one. In the current work, we assume that this area is small.

We now establish some connections with existing common domain adaptation assumptions in the literature. Recall that in order to characterize which domain adaptation task may be *learnable*, Ben-David et al. [59] presented three *assumptions that can help domain adaptation*. Our Theorem 5 does not rely on these assumptions, and remains valid in the absence of these assumptions, but they can be interpreted in our framework as discussed below.

On the covariate shift. A domain adaptation task fulfills the *covariate shift* assumption [60] if the source and target domains only differ in their marginals according to the input space, *i.e.,* $\mathcal{T}_{Y|\mathbf{x}}(y) = \mathcal{S}_{Y|\mathbf{x}}(y)$. In this scenario, one may estimate the values of $\beta_q(\mathcal{T}_{\mathbf{X}} \| \mathcal{S}_{\mathbf{X}})$, and even $\eta_{\mathcal{T}\setminus\mathcal{S}}$, by using unsupervised density estimation methods. Interestingly, with the additional assumption that the domains share the same support, we have $\eta_{\mathcal{T}\setminus\mathcal{S}} = 0$. Then from Line (20) we obtain

$$\mathrm{R}_{\mathcal{T}}(G_\rho) \;=\; \tfrac{1}{2}\mathrm{d}_{\mathcal{T}_{\mathbf{X}}}(\rho) + \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{S}_{\mathbf{X}}} \frac{\mathcal{T}_{\mathbf{X}}(\mathbf{x})}{\mathcal{S}_{\mathbf{X}}(\mathbf{x})} \mathop{\mathbf{E}}_{h\sim\rho} \mathop{\mathbf{E}}_{h'\sim\rho} \mathcal{L}_{0\text{-}1}\left(h(\mathbf{x}), y\right) \mathcal{L}_{0\text{-}1}\left(h'(\mathbf{x}), y\right),$$

which suggests a way to correct the *shift* between the domains by reweighting the labeled source distribution, while considering the information from the target disagreement.

On the weight ratio. The *weight ratio* [59] of source and target domains, with respect to a collection of input space subsets $\mathcal{B} \subseteq 2^{\mathbf{X}}$, is given by

$$C_{\mathcal{B}}(\mathcal{S}, \mathcal{T}) \;=\; \inf_{b\in\mathcal{B},\, \mathcal{T}_{\mathbf{X}}(b)\neq 0} \frac{\mathcal{S}_{\mathbf{X}}(b)}{\mathcal{T}_{\mathbf{X}}(b)}\,.$$

When $C_{\mathcal{B}}(\mathcal{S}, \mathcal{T})$ is bounded away from 0, adaptation should be achievable under covariate shift. In this context, and when $\textsc{supp}(\mathcal{S}) = \textsc{supp}(\mathcal{T})$, the limit case of $\beta_\infty(\mathcal{T}\|\mathcal{S})$ is equal to the inverse of the *pointwise weight ratio* obtained by letting $\mathcal{B} = \{\{\mathbf{x}\} : \mathbf{x} \in \mathbf{X}\}$ in $C_{\mathcal{B}}(\mathcal{S}, \mathcal{T})$. Indeed, both β_q and $C_{\mathcal{B}}$ compare the density of source and target domains, but provide distinct strategies to relax the pointwise weight ratio; the former by lowering the value of q and the latter by considering larger subspaces $\mathcal{B}$.

On the cluster assumption. A target domain fulfills the *cluster assumption* when examples of the same label belong to a common "area" of the input space, and the differently labeled "areas" are well separated by *low-density regions* (formalized by the *probabilistic Lipschitzness* [61]). Once specialized to linear classifiers, $\mathrm{d}_{\mathcal{T}_{\mathbf{X}}}(\rho)$ behaves nicely in this context (see Section 6).

On representation learning. The main assumption underlying our domain adaptation algorithm exhibited in Section 6 is that the support of the target domain is mostly included in the support of the source domain, *i.e.*, the value of the term $\eta_{\mathcal{T}\setminus\mathcal{S}}$ is small. In situations when $\mathcal{T}\setminus\mathcal{S}$ is sufficiently large to prevent proper adaptation, one could try to reduce its volume while taking care to preserve a good compromise between $\mathrm{d}_{\mathcal{T}_{\mathbf{X}}}(\rho)$ and $\mathrm{e}_{\mathcal{S}}(\rho)$, using a *representation learning* approach, *i.e.*, by projecting source and target examples into a new common input space, as done for example by Chen et al. [22], Ganin et al. [26] (see [6] for a survey of representation learning approaches for domain adaptation vision tasks).

4.3. Comparison of the Two Domain Adaptation Bounds

Since they rely on different approximations, the gap between the bounds of Theorems 4 and 5 varies according to the context. As presented in Sections 4.1.4 and 4.2.3, the main difference between our two bounds lies in the estimable terms, from which we will derive algorithms in Section 6. In Theorem 5, the non-estimable terms are the domains' divergence $\beta_q(\mathcal{T}\|\mathcal{S})$ and the term $\eta_{\mathcal{T}\setminus\mathcal{S}}$. Contrary to the non-controllable term λ_ρ of Theorem 4, these terms do not depend on the *learned* posterior distribution ρ: For every ρ on $\mathcal{H}$, $\beta_q(\mathcal{T}\|\mathcal{S})$ and $\eta_{\mathcal{T}\setminus\mathcal{S}}$ are constant values measuring the relation between the domains for the considered task. Moreover, the fact that the β_q-divergence is not an additive term but a multiplicative one (as opposed to $\mathrm{dis}_\rho(\mathcal{S}_\mathbf{X}, \mathcal{T}_\mathbf{X}) + \lambda_\rho$ in Theorem 4) is a contribution of our new perspective. Consequently, $\beta_q(\mathcal{T}\|\mathcal{S})$ can be viewed as a hyperparameter allowing us to tune the trade-off between the target voters' disagreement and the source joint error. Experiments of Section 7 confirm that this hyperparameter can be successfully selected.

Note that, when $\mathrm{e}_\mathcal{T}(\rho) \geq \mathrm{e}_\mathcal{S}(\rho)$, we can upper-bound the term λ_ρ of Theorem 4 by using the same trick as in Theorem 5 proof. This leads to

$$\begin{aligned} \mathrm{e}_\mathcal{T}(\rho) \,\geq\, \mathrm{e}_\mathcal{S}(\rho) \quad \Longrightarrow \quad \lambda_\rho \;&=\; \mathrm{e}_\mathcal{T}(\rho) - \mathrm{e}_\mathcal{S}(\rho) \\ &\leq\; \beta_q(\mathcal{T}\|\mathcal{S}) \times \big[\mathrm{e}_\mathcal{S}(\rho)\big]^{1-\frac{1}{q}} + \eta_{\mathcal{T}\setminus\mathcal{S}} - \mathrm{e}_\mathcal{S}(\rho)\,. \end{aligned}$$

Thus, in this particular case, we can rewrite Theorem 4 statement as for all ρ on $\mathcal{H}$, we have

$$\mathrm{R}_\mathcal{T}(G_\rho) \leq \mathrm{R}_\mathcal{S}(G_\rho) + \frac{1}{2}\,\mathrm{dis}_\rho(\mathcal{S}_\mathbf{X}, \mathcal{T}_\mathbf{X}) + \beta_q(\mathcal{T}\|\mathcal{S}) \times \big[\mathrm{e}_\mathcal{S}(\rho)\big]^{1-\frac{1}{q}} - \mathrm{e}_\mathcal{S}(\rho) + \eta_{\mathcal{T}\setminus\mathcal{S}}\,.$$

It turns out that, if $\mathrm{d}_{\mathcal{T}_\mathbf{X}}(\rho) \geq \mathrm{d}_{\mathcal{S}_\mathbf{X}}(\rho)$ in addition to $\mathrm{e}_\mathcal{T}(\rho) \geq \mathrm{e}_\mathcal{S}(\rho)$, the above statement reduces to the one of Theorem 5. In words, this occurs in the very particular case where the target disagreement and the target expected joint error are both greater than their source counterparts, which may be interpreted as a rather favorable situation. However, Theorem 5 is tighter in all other cases. This highlights that introducing absolute values in Theorem 4 proof leads to a crude approximation. Remember that we have first followed this path to stay aligned with classical domain adaptation analysis, but our second approach leads to a more suitable analysis in a PAC-Bayesian context. Our experiments of Subsection 6.4 illustrate this empirically, once the domain adaptation bounds are converted into PAC-Bayesian generalization guarantees for linear classifiers.

5. PAC-Bayesian Generalization Guarantees

To compute our domain adaptation bounds, one needs to know the distributions $\mathcal{S}$ and $\mathcal{T}_\mathbf{X}$, which is never the case in real life tasks. PAC-Bayesian theory provides tools to convert the bounds of Theorems 4 and 5 into generalization bounds on the target risk computable from a pair of source-target samples

$(S,T)\sim(\mathcal{S})^{m_s}\times(\mathcal{T}_{\mathbf{X}})^{m_t}$. To achieve this goal, we first provide generalization guarantees for the terms involved in our domain adaptation bounds: $\mathrm{d}_{\mathcal{T}_{\mathbf{X}}}(\rho)$, $\mathrm{e}_{\mathcal{S}}(\rho)$, and $\mathrm{dis}_{\rho}(\mathcal{S}_{\mathbf{X}},\mathcal{T}_{\mathbf{X}})$. These results are presented as corollaries of Theorem 6 below, that generalizes the PAC-Bayesian of Catoni [50] (see Theorem 3 in Section 3.2) to arbitrary loss functions. Indeed, Theorem 6, with $\ell(h,\mathbf{x},y)=\mathcal{L}_{0\text{-}1}\left(h(\mathbf{x}),y\right)$ and Equation (3), gives the usual bound on the Gibbs risk.

Note that the proofs of Theorem 6 (deferred in Appendix C) and Corollary 1 (below) reuse techniques from related results presented in Germain et al. [49]. Indeed, PAC-Bayesian bounds on $\mathrm{d}_{\mathcal{T}_{\mathbf{X}}}(\rho)$ and $\mathrm{e}_{\mathcal{S}}(\rho)$ appeared in the latter, but under different forms.

Theorem 6. *For any domain $\mathcal{D}$ over $\mathbf{X}\times Y$, for any set of hypotheses $\mathcal{H}$, any prior π over $\mathcal{H}$, any loss $\ell:\mathcal{H}\times\mathbf{X}\times Y\to[0,1]$, any real number $\alpha>0$, with a probability at least $1-\delta$ over the random choice of $\{(\mathbf{x}_i,y_i)\}_{i=1}^{m}\sim(\mathcal{D})^m$, we have, for all ρ on $\mathcal{H}$,*

$$\underset{(\mathbf{x},y)\sim\mathcal{D}}{\mathbf{E}}\ \underset{h\sim\rho}{\mathbf{E}}\ \ell(h,\mathbf{x},y)\ \leq\ \frac{\alpha}{1-e^{-\alpha}}\left[\frac{1}{m}\sum_{i=1}^{m}\underset{h\sim\rho}{\mathbf{E}}\ \ell(h,\mathbf{x}_i,y_i)+\frac{\mathrm{KL}(\rho\|\pi)+\ln\frac{1}{\delta}}{m\times\alpha}\right].$$

We now exploit Theorem 6 to obtain generalization guarantees on the expected disagreement, the expected joint error, and the domain disagreement. In Corollary 1 below, we are especially interested in the possibility of controlling the trade-off—between the empirical estimate computed on the samples and the complexity term $\mathrm{KL}(\rho\|\pi)$—with the help of parameters a, b and c.

Corollary 1. *For any domains $\mathcal{S}$ and $\mathcal{T}$ over $\mathbf{X}\times Y$, any set of voters $\mathcal{H}$, any prior π over $\mathcal{H}$, any $\delta\in(0,1]$, any real numbers $a>0$, $b>0$ and $c>0$, we have*

— with a probability at least $1-\delta$ over $T\sim(\mathcal{T}_{\mathbf{X}})^{m_t}$,

$$\forall\rho\ \text{on}\ \mathcal{H},\ \mathrm{d}_{\mathcal{T}_{\mathbf{X}}}(\rho)\ \leq\ \frac{c}{1-e^{-c}}\left[\widehat{\mathrm{d}}_T(\rho)+\frac{2\,\mathrm{KL}(\rho\|\pi)+\ln\frac{1}{\delta}}{m_t\times c}\right],$$

— with a probability at least $1-\delta$ over $S\sim(\mathcal{S})^{m_s}$,

$$\forall\rho\ \text{on}\ \mathcal{H},\ \mathrm{e}_{\mathcal{S}}(\rho)\ \leq\ \frac{b}{1-e^{-b}}\left[\widehat{\mathrm{e}}_S(\rho)+\frac{2\,\mathrm{KL}(\rho\|\pi)+\ln\frac{1}{\delta}}{m_s\times b}\right],$$

— with a probability at least $1-\delta$ over $S\times T\sim(\mathcal{S}_{\mathbf{X}}\times\mathcal{T}_{\mathbf{X}})^m$,

$$\forall\rho\ \text{on}\ \mathcal{H},\ \mathrm{dis}_{\rho}(\mathcal{S}_{\mathbf{X}},\mathcal{T}_{\mathbf{X}})\ \leq\ \frac{2\,a}{1-e^{-2a}}\left[\widehat{\mathrm{dis}}_{\rho}(S,T)+\frac{2\,\mathrm{KL}(\rho\|\pi)+\ln\frac{2}{\delta}}{m\times a}+1\right]-1\,,$$

where $\widehat{\mathrm{d}}_T(\rho)$, $\widehat{\mathrm{e}}_S(\rho)$, and $\widehat{\mathrm{dis}}_{\rho}(S,T)$ are the empirical estimations of the target voters' disagreement, the source joint error, and the domain disagreement.

Proof. Given π and ρ over $\mathcal{H}$, we consider a new prior π^2 and a new posterior ρ^2, both over $\mathcal{H}^2$, such that: $\forall\, h_{ij}=(h_i,h_j)\in\mathcal{H}^2,\ \pi^2(h_{ij})=\pi(h_i)\pi(h_j)$, and

$\rho^2(h_{ij}) = \rho(h_i)\rho(h_j)$. Thus, $\mathrm{KL}(\rho^2\|\pi^2) = 2\,\mathrm{KL}(\rho\|\pi)$ (see Lemma 7 in Appendix A). Let us define four new loss functions for a "paired voter" $h_{ij} \in \mathcal{H}^2$:

$$\begin{aligned}
\ell_d(h_{ij}, \mathbf{x}, y) &= \mathcal{L}_{0\text{-}1}\left(h_i(\mathbf{x}), h_j(\mathbf{x})\right), \\
\ell_e(h_{ij}, \mathbf{x}, y) &= \mathcal{L}_{0\text{-}1}\left(h_i(\mathbf{x}), y\right) \times \mathcal{L}_{0\text{-}1}\left(h_j(\mathbf{x}), y\right), \\
\ell_{d^{(1)}}(h_{ij}, (\mathbf{x}^s, \mathbf{x}^t), \cdot) &= \frac{1 + \mathcal{L}_{0\text{-}1}\left(h_i(\mathbf{x}^s), h_j(\mathbf{x}^s)\right) - \mathcal{L}_{0\text{-}1}\left(h_i(\mathbf{x}^t), h_j(\mathbf{x}^t)\right)}{2}, \\
\ell_{d^{(2)}}(h_{ij}, (\mathbf{x}^s, \mathbf{x}^t), \cdot) &= \frac{1 + \mathcal{L}_{0\text{-}1}\left(h_i(\mathbf{x}^t), h_j(\mathbf{x}^t)\right) - \mathcal{L}_{0\text{-}1}\left(h_i(\mathbf{x}^s), h_j(\mathbf{x}^s)\right)}{2}.
\end{aligned}$$

Thus, from Theorem 6:

- The bound on $\mathrm{d}_{\mathcal{T}_\mathbf{X}}(\rho)$ is obtained with $\ell \coloneqq \ell_d$, and Equation (5);
- The bound on $\mathrm{e}_{\mathcal{S}}(\rho)$ is similarly obtained with $\ell \coloneqq \ell_e$, and Equation (6);
- The bound on $\mathrm{dis}_\rho(\mathcal{S}_\mathbf{X}, \mathcal{T}_\mathbf{X})$ is obtained with $\ell \coloneqq \ell_{d^{(1)}}$, by upper-bounding

$$d^{(1)} = \mathrm{d}_{\mathcal{S}_\mathbf{X}}(\rho) - \mathrm{d}_{\mathcal{T}_\mathbf{X}}(\rho) = 2 \mathop{\mathbf{E}}_{h_{ij}\sim\rho^2} \mathop{\mathbf{E}}_{(\mathbf{x}^s,\mathbf{x}^t)\sim\mathcal{S}_\mathbf{X}\times\mathcal{T}_\mathbf{X}} \ell_{d^{(1)}}(h_{ij}, (\mathbf{x}^s, \mathbf{x}^t), \cdot) - 1,$$

from its empirical counterpart

$$\widehat{d}^{(1)} = \widehat{\mathrm{d}}_S(\rho) - \widehat{\mathrm{d}}_T(\rho) = \frac{2}{m} \mathop{\mathbf{E}}_{h_{ij}\sim\rho^2} \sum_{k=1}^{m} \ell_{d^{(1)}}(h_{ij}, (\mathbf{x}^s_k, \mathbf{x}^t_k), \cdot) - 1.$$

We then have, with probability $1 - \frac{\delta}{2}$ over the choice of $S \times T \sim (\mathcal{S}_\mathbf{X} \times \mathcal{T}_\mathbf{X})^m$,

$$\frac{|d^{(1)}| + 1}{2} \;\leq\; \frac{a}{1 - e^{-2a}} \left[|\widehat{d}^{(1)}| + 1 + \frac{2\,\mathrm{KL}(\rho\|\pi) + \ln\frac{2}{\delta}}{m \times a} \right].$$

In turn, with $\ell \coloneqq \ell_{d^{(2)}}$ we bound $d^{(2)} = \mathrm{d}_{\mathcal{T}_\mathbf{X}}(\rho) - \mathrm{d}_{\mathcal{S}_\mathbf{X}}(\rho)$ by its empirical counterpart $\widehat{d}^{(2)} = \widehat{\mathrm{d}}_T(\rho) - \widehat{\mathrm{d}}_S(\rho)$, with probability $1 - \frac{\delta}{2}$ over the choice of $S \times T \sim (\mathcal{S}_\mathbf{X} \times \mathcal{T}_\mathbf{X})^m$,

$$\frac{|d^{(2)}| + 1}{2} \;\leq\; \frac{a}{1 - e^{-2a}} \left[|\widehat{d}^{(2)}| + 1 + \frac{2\,\mathrm{KL}(\rho\|\pi) + \ln\frac{2}{\delta}}{m \times a} \right].$$

Finally, by the union bound, with probability $1 - \delta$, we have

$$\begin{aligned}
\mathrm{dis}_\rho(\mathcal{S}_\mathbf{X}, \mathcal{T}_\mathbf{X}) &= \max\left\{ d^{(1)}, d^{(2)} \right\} \\
&\leq \frac{2a}{1 - e^{-2a}} \left[\widehat{\mathrm{dis}}_\rho(S, T) + \frac{2\,\mathrm{KL}(\rho\|\pi) + \ln\frac{2}{\delta}}{m \times a} + 1 \right] - 1,
\end{aligned}$$

and we are done. ∎

The following bound is based on the above Catoni's approach for our domain adaptation bound of Theorem 4 and corresponds to the one from which we derive—in Section 6—PBDA our first algorithm for PAC-Bayesian domain adaptation.

Theorem 7. *For any domains $\mathcal{S}$ and $\mathcal{T}$ (resp. with marginals $\mathcal{S}_{\mathbf{X}}$ and $\mathcal{T}_{\mathbf{X}}$) over $\mathbf{X} \times Y$, any set of hypotheses $\mathcal{H}$, any prior distribution π over $\mathcal{H}$, any $\delta \in (0, 1]$, any real numbers $\omega > 0$ and $a > 0$, with a probability at least $1 - \delta$ over the choice of $S \times T \sim (\mathcal{S} \times \mathcal{T}_{\mathbf{X}})^m$, for every posterior distribution ρ on $\mathcal{H}$, we have*

$$\mathrm{R}_{\mathcal{T}}(G_\rho) \leq \omega' \widehat{\mathrm{R}}_S(G_\rho) + a' \tfrac{1}{2} \widehat{\mathrm{dis}}_\rho(S, T) + \left(\tfrac{\omega'}{\omega} + \tfrac{a'}{a} \right) \frac{\mathrm{KL}(\rho \| \pi + \ln \frac{3}{\delta}}{m} + \lambda_\rho + \tfrac{1}{2}(a' - 1) \,,$$

where $\widehat{\mathrm{R}}_S(G_\rho)$ and $\widehat{\mathrm{dis}}_\rho(S, T)$ are the empirical estimates of the target risk and the domain disagreement; λ_ρ is defined by Equation (16); $\omega' = \frac{\omega}{1 - e^{-\omega}}$ and $a' = \frac{2a}{1 - e^{-2a}}$.

Proof. In Theorem 4, we replace $\mathrm{R}_{\mathcal{S}}(G_\rho)$ and $\mathrm{dis}_\rho(\mathcal{S}_{\mathbf{X}}, \mathcal{T}_{\mathbf{X}})$ by their upper bound, obtained from Theorem 3 and Corollary 1, with δ chosen respectively as $\frac{\delta}{3}$ and $\frac{2\delta}{3}$. In the latter case, we use

$$2\,\mathrm{KL}(\rho \| \pi) + \ln \tfrac{2}{2\delta/3} \;=\; 2\,\mathrm{KL}(\rho \| \pi) + \ln \tfrac{3}{\delta} \;<\; 2\left(\mathrm{KL}(\rho \| \pi) + \ln \tfrac{3}{\delta}\right).$$

■

We now derive a PAC-Bayesian generalization bound for our second domain adaptation bound of Theorem 5 from which we derive—in Section 6—our second algorithm for PAC-Bayesian domain adaptation DALC. For algorithmic simplicity, we deal with Theorem 5 when $q \to \infty$. Thanks to Corollary 1, we obtain the following generalization bound defined with respect to the empirical estimates of the target disagreement and the source joint error.

Theorem 8. *For any domains $\mathcal{S}$ and $\mathcal{T}$ over $\mathbf{X} \times Y$, any set of voters $\mathcal{H}$, any prior π over $\mathcal{H}$, any $\delta \in (0, 1]$, any real numbers $b > 0$ and $c > 0$, with a probability at least $1 - \delta$ over the choices of $S \sim (\mathcal{S})^{m_s}$ and $T \sim (\mathcal{T}_{\mathbf{X}})^{m_t}$, for every posterior distribution ρ on $\mathcal{H}$, we have*

$$\mathrm{R}_{\mathcal{T}}(G_\rho) \;\leq\; c' \tfrac{1}{2} \widehat{\mathrm{d}}_T(\rho) + b' \widehat{\mathrm{e}}_S(\rho) + \eta_{\mathcal{T} \setminus \mathcal{S}} + \left(\frac{c'}{m_t\, c} + \frac{b'}{m_s\, b} \right) \left(2\,\mathrm{KL}(\rho \| \pi) + \ln \tfrac{2}{\delta} \right),$$

where $\widehat{\mathrm{d}}_T(\rho)$ and $\widehat{\mathrm{e}}_S(\rho)$ are the empirical estimations of the target voters' disagreement and the source joint error, and $b' = \frac{b}{1 - e^{-b}}\, \beta_\infty(\mathcal{T} \| \mathcal{S})$, and $c' = \frac{c}{1 - e^{-c}}$.

Proof. We bound separately $\mathrm{d}_{\mathcal{T}_{\mathbf{X}}}(\rho)$ and $\mathrm{e}_{\mathcal{S}}(\rho)$ using Corollary 1 (with probability $1 - \frac{\delta}{2}$ each), and then combine the two upper bounds according to Theorem 5. ■

From an optimization perspective, the problem suggested by the bound of Theorem 8 is much more convenient to minimize than the PAC-Bayesian bound derived in Theorem 7. The former is *smoother* than the latter: The absolute value related to the domain disagreement $\mathrm{dis}_\rho(\mathcal{S}_{\mathbf{X}}, \mathcal{T}_{\mathbf{X}})$ disappears in benefit of the domain divergence $\beta_\infty(\mathcal{T} \| \mathcal{S})$, which is constant and can be considered as an hyperparameter of the algorithm. Additionally, Theorem 7 requires equal source and target sample sizes while Theorem 8 allows $m_s \neq m_t$. Moreover,

for algorithmic purposes, we ignore the ρ-dependent non-constant term λ_ρ of Theorem 7. In our second analysis, such compromise is not mandatory in order to apply the theoretical result to real problems, since the non-estimable term $\eta_{\mathcal{T}\setminus\mathcal{S}}$ is constant and does not depend on the learned ρ. Hence, we can neglect $\eta_{\mathcal{T}\setminus\mathcal{S}}$ without any impact on the optimization problem described in the next section. Besides, it is realistic to consider $\eta_{\mathcal{T}\setminus\mathcal{S}}$ as a small quantity in situations where the source and target supports are similar.

6. PAC-Bayesian Domain Adaptation Learning of Linear Classifiers

In this section, we design two learning algorithms for domain adaptation[14] inspired by the PAC-Bayesian learning algorithm of Germain et al. [44]. That is, we adopt the specialization of the PAC-Bayesian theory to linear classifiers described in Section 3.3. The taken approach is the one privileged in numerous PAC-Bayesian works [*e.g.*, 56, 57, 55, 45, 44, 1], as it makes the risk of the linear classifier $h_{\mathbf{w}}$ and the risk of a (properly parametrized) majority vote coincide, while in the same time promoting large margin classifiers.

6.1. Domain and Expected Disagreement, Joint Error of Linear Classifiers

Let us consider a prior $\pi_{\mathbf{0}}$ and a posterior $\rho_{\mathbf{w}}$ that are spherical Gaussian distributions over a space of linear classifiers, exactly as defined in Section 3.3. We seek to express the *domain disagreement* $\mathrm{dis}_{\rho_{\mathbf{w}}}(\mathcal{S}_{\mathbf{X}}, \mathcal{T}_{\mathbf{X}})$, *expected disagreement* $\mathrm{d}_{\mathcal{D}_{\mathbf{X}}}(\rho_{\mathbf{w}})$ and the *expected joint error* $\mathrm{e}_{\mathcal{D}}(\rho_{\mathbf{w}})$.
First, for any marginal $\mathcal{D}_{\mathbf{X}}$, the expected disagreement for linear classifiers is

$$
\begin{aligned}
\mathrm{d}_{\mathcal{D}_{\mathbf{X}}}(\rho_{\mathbf{w}}) &= \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{X}}} \mathop{\mathbf{E}}_{(h,h')\sim\rho_{\mathbf{w}}^2} \mathcal{L}_{0\text{-}1}\big(h(\mathbf{x}), h'(\mathbf{x})\big) \\
&= \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{X}}} \mathop{\mathbf{E}}_{(h,h')\sim\rho_{\mathbf{w}}^2} \mathrm{I}[h(\mathbf{x}) \neq h'(\mathbf{x})] \\
&= \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{X}}} \mathop{\mathbf{E}}_{(h,h')\sim\rho_{\mathbf{w}}^2} \Big(\mathrm{I}[h(\mathbf{x})=1]\,\mathrm{I}[h'(\mathbf{x})=-1] + \mathrm{I}[h(\mathbf{x})=-1]\,\mathrm{I}[h'(\mathbf{x})=1] \Big) \\
&= 2 \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{X}}} \mathop{\mathbf{E}}_{(h,h')\sim\rho_{\mathbf{w}}^2} \mathrm{I}[h(\mathbf{x}) = 1]\,\mathrm{I}[h'(\mathbf{x}) = -1] \\
&= 2 \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{X}}} \mathop{\mathbf{E}}_{h\sim\rho_{\mathbf{w}}} \mathrm{I}[h(\mathbf{x}) = 1] \mathop{\mathbf{E}}_{h'\sim\rho_{\mathbf{w}}} \mathrm{I}[h'(\mathbf{x}) = -1] \\
&= 2 \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{X}}} \Phi_{\mathrm{R}}\left(\frac{\mathbf{w}\cdot\mathbf{x}}{\|\mathbf{x}\|}\right) \Phi_{\mathrm{R}}\left(-\frac{\mathbf{w}\cdot\mathbf{x}}{\|\mathbf{x}\|}\right) \\
&= \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{X}}} \Phi_{\mathrm{d}}\left(\frac{\mathbf{w}\cdot\mathbf{x}}{\|\mathbf{x}\|}\right), \qquad (22)
\end{aligned}
$$

where

$$
\Phi_{\mathrm{d}}(x) \;=\; 2\,\Phi_{\mathrm{R}}(x)\,\Phi_{\mathrm{R}}(-x)\,. \qquad (23)
$$

[14] The code of our algorithms are available on-line. More details are given in Section 7.

Thus, the domain disagreement for linear classifiers is

$$\begin{aligned}\mathrm{dis}_{\rho_{\mathbf{w}}}(\mathcal{S}_{\mathbf{X}},\mathcal{T}_{\mathbf{X}}) &= \left|\,\mathrm{d}_{\mathcal{S}_{\mathbf{X}}}(\rho_{\mathbf{w}}) - \mathrm{d}_{\mathcal{T}_{\mathbf{X}}}(\rho_{\mathbf{w}})\,\right| \\ &= \left|\,\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{S}_{\mathbf{X}}} \Phi_{\mathrm{d}}\left(\frac{\mathbf{w}\cdot\mathbf{x}}{\|\mathbf{x}\|}\right) - \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{T}_{\mathbf{X}}} \Phi_{\mathrm{d}}\left(\frac{\mathbf{w}\cdot\mathbf{x}}{\|\mathbf{x}\|}\right)\right|. \qquad (24)\end{aligned}$$

Following a similar approach, the expected joint error is, for all $\mathbf{w}\in\mathbb{R}$,

$$\begin{aligned}\mathrm{e}_{\mathcal{D}}(\rho_{\mathbf{w}}) &= \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \mathop{\mathbf{E}}_{h\sim\rho_{\mathbf{w}}} \mathop{\mathbf{E}}_{h'\sim\rho_{\mathbf{w}}} \mathcal{L}_{0\text{-}1}\left(h(\mathbf{x}),y\right)\times\mathcal{L}_{0\text{-}1}\left(h'(\mathbf{x}),y\right) \\ &= \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \mathop{\mathbf{E}}_{h\sim\rho_{\mathbf{w}}} \mathcal{L}_{0\text{-}1}\left(h(\mathbf{x}),y\right) \mathop{\mathbf{E}}_{h'\sim\rho_{\mathbf{w}}} \mathcal{L}_{0\text{-}1}\left(h'(\mathbf{x}),y\right) \\ &= \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \Phi_{\mathrm{e}}\left(y\,\frac{\mathbf{w}\cdot\mathbf{x}}{\|\mathbf{x}\|}\right), \qquad (25)\end{aligned}$$

with

$$\Phi_{\mathrm{e}}(x) \;=\; \left[\Phi_{\mathrm{R}}(x)\right]^2. \qquad (26)$$

Functions Φ_{e} and Φ_{d} defined above can be interpreted as loss functions for linear classifiers (illustrated by Figure 2a).

6.2. Domain Adaptation Bounds

Theorems 4 and 5 (when $q\to\infty$) specialized to linear classifiers give the two following corollaries. We recall that $\mathrm{R}_{\mathcal{T}}(h_{\mathbf{w}}) = \mathrm{R}_{\mathcal{T}}(B_{\rho_{\mathbf{w}}}) \le 2\,\mathrm{R}_{\mathcal{T}}(G_{\rho_{\mathbf{w}}})$.

Corollary 2. *Let $\mathcal{S}$ and $\mathcal{T}$ respectively be the source and the target domains on $\mathbf{X}\times Y$. For all $\mathbf{w}\in\mathbb{R}$, we have*

$$\mathrm{R}_{\mathcal{T}}(h_{\mathbf{w}}) \;\le\; 2\,\mathrm{R}_{\mathcal{S}}(G_{\rho_{\mathbf{w}}}) + \mathrm{dis}_{\rho_{\mathbf{w}}}(\mathcal{S}_{\mathbf{X}},\mathcal{T}_{\mathbf{X}}) + 2\lambda_{\rho_{\mathbf{w}}}\,,$$

where $\mathrm{dis}_{\rho_{\mathbf{w}}}(\mathcal{S}_{\mathbf{X}},\mathcal{T}_{\mathbf{X}})$ and $\lambda_{\rho_{\mathbf{w}}}$ are respectively defined by Equations (24) *and* (16)*.*

Corollary 3. *Let $\mathcal{S}$ and $\mathcal{T}$ respectively be the source and the target domains on $\mathbf{X}\times Y$. For all $\mathbf{w}\in\mathbb{R}$, we have*

$$\mathrm{R}_{\mathcal{T}}(h_{\mathbf{w}}) \;\le\; \mathrm{d}_{\mathcal{T}_{\mathbf{X}}}(\rho_{\mathbf{w}}) + 2\,\beta_{\infty}(\mathcal{T}\|\mathcal{S})\times\mathrm{e}_{\mathcal{S}}(\rho_{\mathbf{w}}) + 2\,\eta_{\mathcal{T}\setminus\mathcal{S}}\,,$$

where $\mathrm{d}_{\mathcal{T}_{\mathbf{X}}}(\rho_{\mathbf{w}})$, $\mathrm{e}_{\mathcal{S}}(\rho_{\mathbf{w}})$, $\beta_{\infty}(\mathcal{T}\|\mathcal{S})$ and $\eta_{\mathcal{T}\setminus\mathcal{S}}$ are respectively defined by Equations (22)*,* (25)*,* (18) *and* (19)*.*

For fixed values of $\beta_{\infty}(\mathcal{T}\|\mathcal{S})$ and $\eta_{\mathcal{T}\setminus\mathcal{S}}$, the target risk $\mathrm{R}_{\mathcal{T}}(h_{\mathbf{w}})$ is upper-bounded by a β_{∞}-weighted sum of two losses. The expected Φ_{e}-loss (*i.e.*, the joint error) is computed on the (labeled) source domain; it aims to label the source examples correctly, but is more permissive on the required margin than the Φ-loss (*i.e.*, the Gibbs risk). The expected Φ_{d}-loss (*i.e.*, the disagreement) is computed on the target (unlabeled) domain; it promotes large *unsigned* target margins. Thus, if a target domain fulfills the *cluster assumption* (described in Section 4.2.3), $\mathrm{d}_{\mathcal{T}_{\mathbf{X}}}(\rho_{\mathbf{w}})$ will be low when the decision boundary crosses a

low-density region between the homogeneous labeled clusters. Hence, Corollary 3 reflects that some source errors may be allowed if, doing so, the separation of the target domain is improved. Figure 2a leads to an insightful geometric interpretation of the two domain adaptation trade-off promoted by Corollaries 2 and 3.

6.3. Generalization Bounds and Learning Algorithms

6.3.1. First Domain Adaptation Learning Algorithm (PBDA).

Theorem 7 specialized to linear classifiers gives the following.

Corollary 4. *For any domains $\mathcal{S}$ and $\mathcal{T}$ over $\mathbf{X} \times Y$, any $\delta \in (0,1]$, any $\omega > 0$ and $a > 0$, with a probability at least $1-\delta$ over the choices of $S \sim (\mathcal{S})^m$ and $T \sim (\mathcal{T}_{\mathbf{X}})^m$, we have, for all $\mathbf{w} \in \mathbb{R}$,*

$$\mathrm{R}_{\mathcal{T}}(h_{\mathbf{w}}) \leq 2\omega' \widehat{\mathrm{R}}_S(G_{\rho_{\mathbf{w}}}) + a' \widehat{\mathrm{dis}}_{\rho_{\mathbf{w}}}(S,T) + 2\lambda_{\rho_{\mathbf{w}}} + 2\left(\frac{\omega'}{\omega} + \frac{a'}{a}\right) \frac{\|\mathbf{w}\|^2 + \ln\frac{3}{\delta}}{m} + (a'-1)\,,$$

where $\widehat{\mathrm{R}}_S(G_{\rho_{\mathbf{w}}})$ and $\widehat{\mathrm{dis}}_{\rho_{\mathbf{w}}}(S,T)$, are the empirical estimates of the target risk and the domain disagreement; $\lambda_{\rho_{\mathbf{w}}}$ is obtained using Equation (16); $\omega' = \frac{\omega}{1-e^{-\omega}}$, and $a' = \frac{2a}{1-e^{-2a}}$.

Given a source sample $S = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^m$ and a target sample $T = \{(\mathbf{x}_i^t)\}_{i=1}^m$, we focus on the minimization of the bound given by Corollary 4. We work under the assumption that the term $\lambda_{\rho_{\mathbf{w}}}$ of the bound is negligible. Thus, the posterior distribution $\rho_{\mathbf{w}}$ that minimizes the bound on $\mathrm{R}_{\mathcal{T}}(h_{\mathbf{w}})$ is the same that minimizes

$$\begin{aligned} &\Omega\, m\, \widehat{\mathrm{R}}_S(G_{\rho_{\mathbf{w}}}) + A\, m\, \widehat{\mathrm{dis}}_{\rho_{\mathbf{w}}}(S,T) + \mathrm{KL}(\rho_{\mathbf{w}} \| \pi_{\mathbf{0}}) \qquad (27)\\ &= \Omega \sum_{i=1}^m \Phi_{\mathrm{R}}\left(y_i^s \frac{\mathbf{w}\cdot\mathbf{x}_i^s}{\|\mathbf{x}_i^s\|}\right) + A \left| \sum_{i=1}^m \left[\Phi_{\mathrm{d}}\left(\frac{\mathbf{w}\cdot\mathbf{x}_i^s}{\|\mathbf{x}_i^s\|}\right) - \Phi_{\mathrm{d}}\left(\frac{\mathbf{w}\cdot\mathbf{x}_i^t}{\|\mathbf{x}_i^t\|}\right)\right]\right| + \frac{1}{2}\|\mathbf{w}\|^2\,. \end{aligned}$$

The values $\Omega > 0$ and $A > 0$ are hyperparameters of the algorithm. Note that the constants ω and a of Theorem 7 can be recovered from any Ω and A.

Equation (27) is difficult to minimize by gradient descent, as it contains an absolute value and it is highly non-convex. To make the optimization problem more tractable, we replace the loss function Φ_{R} by its convex relaxation $\widetilde{\Phi}_{\mathrm{R}}$ (as in Section 3.3.3). Even if this optimization task is still not convex (Φ_{d} is quasiconcave), our empirical study shows no need to perform many restarts while performing gradient descent to find a suitable solution.[15] We name this domain adaptation algorithm PBDA.

To sum up, given a source sample $S = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^m$, a target sample $T = \{(\mathbf{x}_i^t)\}_{i=1}^m$, and hyperparameters Ω and A, the algorithm PBDA performs

[15] We observe empirically that a good strategy is to first find the vector $\mathbf{w}$ minimizing the convex problem of PBGD3 described in Section 3.3.3, and then use this $\mathbf{w}$ as a starting point for the gradient descent of PBDA.

gradient descent to minimize the following objective function:

$$G(\mathbf{w}) = \Omega \sum_{i=1}^{m} \widetilde{\Phi}_{\mathrm{R}}\left(y_i^s \frac{\mathbf{w}\cdot\mathbf{x}_i^s}{\|\mathbf{x}_i^s\|}\right) + A\left|\sum_{i=1}^{m}\left[\Phi_{\mathrm{d}}\left(\frac{\mathbf{w}\cdot\mathbf{x}_i^s}{\|\mathbf{x}_i^s\|}\right) - \Phi_{\mathrm{d}}\left(\frac{\mathbf{w}\cdot\mathbf{x}_i^t}{\|\mathbf{x}_i^t\|}\right)\right]\right| + \tfrac{1}{2}\|\mathbf{w}\|^2\,, \quad (28)$$

where $\widetilde{\Phi}_{\mathrm{R}}(x) = \max\left\{\Phi_{\mathrm{R}}(x),\ \frac{1}{2} - \frac{x}{\sqrt{2\pi}}\right\}$ and $\Phi_{\mathrm{d}}(x) = 2\,\Phi_{\mathrm{R}}(x)\,\Phi_{\mathrm{R}}(-x)$ have been defined by Equations (15) and (23). Figure 2a illustrates these three functions.

The gradient $\nabla G(\mathbf{w})$ of the Equation (28) is then given by

$$\begin{aligned}\nabla G(\mathbf{w}) =& \Omega \sum_{i=1}^{m} \widetilde{\Phi}'_{\mathrm{R}}\left(\frac{y_i^s \mathbf{w}\cdot\mathbf{x}_i^s}{\|\mathbf{x}_i^s\|}\right)\frac{y_i^s\mathbf{x}_i^s}{\|\mathbf{x}_i^s\|} \\ &+ s\times A\left(\sum_{i=1}^{m}\left[\Phi'_{\mathrm{d}}\left(\frac{\mathbf{w}\cdot\mathbf{x}_i^t}{\|\mathbf{x}_i^t\|}\right)\frac{\mathbf{x}_i^t}{\|\mathbf{x}_i^t\|} - \Phi'_{\mathrm{d}}\left(\frac{\mathbf{w}\cdot\mathbf{x}_i^s}{\|\mathbf{x}_i^s\|}\right)\frac{\mathbf{x}_i^s}{\|\mathbf{x}_i^s\|}\right]\right) + \mathbf{w}\,,\end{aligned}$$

where $\widetilde{\Phi}'_{\mathrm{R}}(x)$ and $\Phi'_{\mathrm{d}}(x)$ are respectively the derivatives of functions $\widetilde{\Phi}_{\mathrm{R}}$ and Φ_{d} evaluated at point x, and

$$s = \mathrm{sign}\left(\sum_{i=1}^{m}\left[\Phi_{\mathrm{d}}\left(\frac{\mathbf{w}\cdot\mathbf{x}_i^s}{\|\mathbf{x}_i^s\|}\right) - \Phi_{\mathrm{d}}\left(\frac{\mathbf{w}\cdot\mathbf{x}_i^t}{\|\mathbf{x}_i^t\|}\right)\right]\right).$$

We extend these equations to kernels in Section 6.3.3 below.

6.3.2. Second Domain Adaptation Learning Algorithm (DALC).

Now, Theorem 8 specialized to linear classifiers gives the following.

Corollary 5. *For any domains $\mathcal{S}$ and $\mathcal{T}$ over $\mathbf{X}\times Y$, any $\delta\in(0,1]$, any $b>0$ and $c>0$, with a probability at least $1-\delta$ over the choices of $S\sim(\mathcal{S})^{m_s}$ and $T\sim(\mathcal{T}_{\mathbf{X}})^{m_t}$, we have, for all $\mathbf{w}\in\mathbb{R}$,*

$$\mathrm{R}_{\mathcal{T}}(h_{\mathbf{w}}) \le c'\,\widehat{\mathrm{d}}_T(\rho_{\mathbf{w}}) + 2\,b'\,\widehat{\mathrm{e}}_S(\rho_{\mathbf{w}}) + 2\,\eta_{\mathcal{T}\setminus\mathcal{S}} + 2\left(\frac{c'}{m_t\times c} + \frac{b'}{m_s\times b}\right)\left(\|\mathbf{w}\|^2 + \ln\tfrac{2}{\delta}\right),$$

where $\widehat{\mathrm{d}}_T(\rho_{\mathbf{w}})$ and $\widehat{\mathrm{e}}_S(\rho_{\mathbf{w}})$ are the empirical estimations of the target voters' disagreement and the source joint error, $b' = \frac{b}{1-e^{-b}}\,\beta_\infty(\mathcal{T}\|\mathcal{S})$, and $c' = \frac{c}{1-e^{-c}}$.

For a source $S=\{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{m_s}$ and a target $T=\{(\mathbf{x}_i^t)\}_{i=1}^{m_t}$ samples of potentially *different size*, and some hyperparameters $B>0$, $C>0$, minimizing the next objective function *w.r.t* $\mathbf{w}\in\mathbb{R}$ is equivalent to minimize the above bound.

$$\begin{aligned}&C\,\widehat{\mathrm{d}}_T(\rho_{\mathbf{w}}) + B\,\widehat{\mathrm{e}}_S(\rho_{\mathbf{w}}) + \|\mathbf{w}\|^2 \\ &= C\sum_{i=1}^{m_t}\Phi_{\mathrm{d}}\left(\frac{\mathbf{w}\cdot\mathbf{x}_i^t}{\|\mathbf{x}_i^t\|}\right) + B\sum_{i=1}^{m_s}\Phi_{\mathrm{e}}\left(y_i^s\,\frac{\mathbf{w}\cdot\mathbf{x}_i^s}{\|\mathbf{x}_i^s\|}\right) + \|\mathbf{w}\|^2\,. \qquad (29)\end{aligned}$$

We call the optimization of Equation (29) by gradient descent the DALC algorithm, for Domain Adaptation of Linear Classifiers. The gradient of Equation (29) is

$$C\sum_{i=1}^{m_t}\Phi'_{\mathrm{d}}\left(\frac{\mathbf{w}\cdot\mathbf{x}_i^t}{\|\mathbf{x}_i^t\|}\right)\frac{\mathbf{x}_i^t}{\|\mathbf{x}_i^t\|} + B\sum_{i=1}^{m_s}\Phi'_{\mathrm{e}}\left(y_i^s\,\frac{\mathbf{w}\cdot\mathbf{x}_i^s}{\|\mathbf{x}_i^s\|}\right)\frac{y_i^s\mathbf{x}_i^s}{\|\mathbf{x}_i^s\|} + \frac{1}{2}\,\mathbf{w}\,.$$

Contrary to the algorithm PBDA described above, our empirical study shows that there is no need to convexify any component of Equation (29): We obtain as good prediction accuracy when we initialize gradient descent to a uniform vector ($w_i = \frac{1}{d}$ for $i \in \{1, \ldots, d\}$) and when we perform multiple restarts from random initializations. Even if the objective function is not convex, the gradient descent is easy to perform. Indeed, Φ_{d} is smooth and its derivative is continuous, in contrast with the absolute value of $\widehat{\mathrm{dis}}_{\rho_{\mathbf{w}}}(S,T)$ in Equation (27) (see also the forthcoming toy experiment of Figure 2d). Thus, the actual optimization problem of DALC is closer to the theoretical analysis than the PBDA one.

6.3.3. Using a Kernel Function

Like the algorithm PBGD3 in Subsection 3.3.2, the kernel trick applies to PBDA and DALC. Given a kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, one can express a linear classifier in an *RKHS* by a dual weight vector $\boldsymbol{\alpha} \in \mathbb{R}^{m_s+m_t}$,

$$h_{\mathbf{w}}(\mathbf{x}) \;=\; \mathrm{sign}\left[\sum_{i=1}^{m} \alpha_i k(\mathbf{x}_i^s, \mathbf{x}) + \sum_{i=1}^{m} \alpha_{i+m} k(\mathbf{x}_i^t, \mathbf{x})\right].$$

Let $S = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{m_s}$, $T = \{\mathbf{x}_i^t\}_{i=1}^{m_t}$ and $\mathrm{M} = m_s + m_t$. We denote K the kernel matrix of size $\mathrm{M} \times \mathrm{M}$ such as $K_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$, where

$$\mathbf{x}_{\#} \;=\; \begin{cases} \mathbf{x}_i^s & \text{if } \# \leq m_s \quad \text{(source examples)} \\ \mathbf{x}_{\#-m_s}^t & \text{otherwise.} \quad \text{(target examples)} \end{cases}$$

On the one hand, in that case, with $m_s = m_t = m$, the objective function of PBDA (Equation 28) is rewritten in terms of the vector $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots \alpha_{2m})$ as

$$G(\boldsymbol{\alpha}) = \Omega \sum_{i=1}^{m} \widetilde{\Phi}_{\mathrm{R}}\left(y_i \tfrac{\sum_{j=1}^{2m} \alpha_j K_{i,j}}{\sqrt{K_{i,i}}}\right) + A\left|\sum_{i=1}^{m}\left[\Phi_{\mathrm{d}}\left(\tfrac{\sum_{j=1}^{2m} \alpha_j K_{i,j}}{\sqrt{K_{i,i}}}\right) - \Phi_{\mathrm{d}}\left(\tfrac{\sum_{j=1}^{2m} \alpha_j K_{i+m,j}}{\sqrt{K_{i+m,i+m}}}\right)\right]\right| + \tfrac{1}{2}\sum_{i=1}^{2m}\sum_{j=1}^{2m} \alpha_i \alpha_j K_{i,j}\,.$$

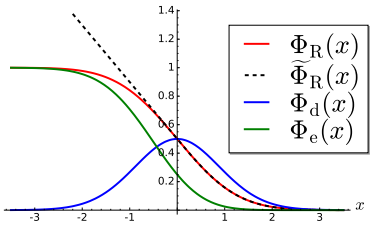
On the other hand, the objective function of DALC (Equation 29) can be rewritten in terms of the vector $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots \alpha_{\mathrm{M}})$ as

$$C \sum_{i=m_s+1}^{\mathrm{M}} \Phi_{\mathrm{d}}\left(\tfrac{\sum_{j=1}^{\mathrm{M}} \alpha_j K_{i,j}}{\sqrt{K_{i,i}}}\right) + B \sum_{i=1}^{m_s} \Phi_{\mathrm{e}}\left(y_i \tfrac{\sum_{j=1}^{\mathrm{M}} \alpha_j K_{i,j}}{\sqrt{K_{i,i}}}\right) + \sum_{i=1}^{\mathrm{M}}\sum_{j=1}^{\mathrm{M}} \alpha_i \alpha_j K_{i,j}\,.$$

To perform the gradient descent in terms of dual weights $\boldsymbol{\alpha}$, we start the gradient descent from the point $\alpha_i = \frac{y_i}{\mathrm{M}}$ for $i \in \{1, \ldots, m_s\}$, and $\alpha_i = \frac{1}{\mathrm{M}}$ for $i \in \{m_s + 1, \ldots, \mathrm{M}\}$.

6.4. Illustration on a Toy Dataset

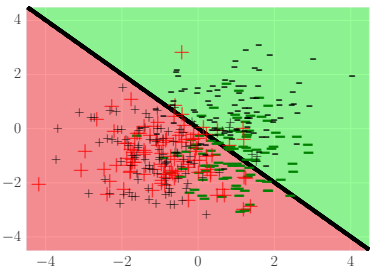
To illustrate and compare the trade-offs of both algorithms PBDA and DALC, we extend the toy experiment of Subsection 3.3.4 (see Figure 1). To obtain the



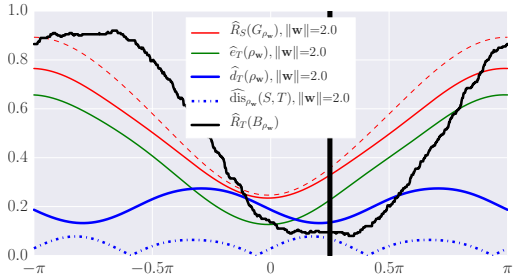
(a) Loss functions given by the specialization to linear classifiers.

FUNCTION	DERIVATIVE
$\Phi_{\mathrm{R}}(x) = \frac{1}{2}\left[1 - \mathrm{Erf}\left(\frac{x}{\sqrt{2}}\right)\right]$	$\Phi'_{\mathrm{R}}(x) = -\frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$
$\widetilde{\Phi}_{\mathrm{R}}(x) = \max\left\{\Phi_{\mathrm{R}}(x), \frac{1}{2} - \frac{x}{\sqrt{2\pi}}\right\}$	$\Phi'_{\mathrm{R}}\left(\max\{0, x\}\right)$
$\Phi_{\mathrm{d}}(x) = 2\,\Phi_{\mathrm{R}}(x)\,\Phi_{\mathrm{R}}(-x)$	$-\sqrt{\frac{2}{\pi}}\mathrm{Erf}\left(\frac{x}{\sqrt{2}}\right) e^{-\frac{1}{2}x^2}$
$\Phi_{\mathrm{e}}(x) = \left[\Phi_{\mathrm{R}}(x)\right]^2$	$2\,\Phi_{\mathrm{R}}(x)\,\Phi'_{\mathrm{R}}(x)$

(b) Loss functions definitions and their derivatives.



(c) Toy dataset, and the decision boundary for $\theta = \frac{\pi}{4}$ (matching the vertical line of Fig.(d)). Source points are red and green, target points are black.



(d) PBDA and DALC loss values, according to θ. PBDA performs a trade-off between $\widehat{\mathrm{dis}}_{\rho_{\mathbf{w}}}(S,T)$ and $\widehat{\mathrm{R}}_S(G_{\rho_{\mathbf{w}}})$ (or its convex surrogate, see the red dashed line); DALC's trade-off is between $\widehat{\mathrm{d}}_T(\rho_{\mathbf{w}})$ and $\widehat{\mathrm{e}}_S(\rho_{\mathbf{w}})$.

Figure 2: Understanding PBDA and DALC domain adaptation learning algorithms in terms of loss functions. Upper Figures (a-b) show the loss functions, and lower Figures (c-d) illustrate the behavior on a toy dataset.

two-dimensional dataset illustrated by Figure 2c, we use, as the source sample, the 200 examples of the supervised experiment—generated by Gaussians of mean $(-1,-1)$ for the positives and $(-1,1)$ for the negatives (see Figure 1c). Then, we generate 100 positive target examples according to a Gaussian of mean $(-1,-1)$ and 100 negative target examples according to a Gaussian of mean $(1,1)$. All Gaussian distributions have unit variance. Note that positive source and target examples are generated by the same distribution.

We study linear classifiers $h_{\mathbf{w}}$, with $\mathbf{w} = 2\,(\cos\theta, \sin\theta) \in \mathbb{R}^2$. That is, we fix the norm value $\|\mathbf{w}\| = 2$. Figure 2d shows the quantities varying in our two domain adaptation approaches while rotating the decision boundary $\theta \in [-\pi, \pi]$ around the origin. On the one hand, PBDA algorithm minimizes a trade-off between the domain disagreement $\widehat{\mathrm{dis}}_{\rho_{\mathbf{w}}}(S,T)$ and the source Gibbs risk $\widehat{\mathrm{R}}_S(G_{\rho_{\mathbf{w}}})$ convex surrogate given by Equation (15). On the other hand, DALC minimizes a trade-off between the target disagreement $\widehat{\mathrm{d}}_T(\rho_{\mathbf{w}})$ and source joint error $\widehat{\mathrm{e}}_S(\rho_{\mathbf{w}})$. From both Figures 2a and 2d, we see that the Gibbs risk, its convex surrogate, and the joint error behave similarly; they are following the linear classifier accuracy on the source sample. However, the domains' divergence

and the target joint error values notably differ for the experiment of Figure 2d : When the target accuracy is optimal (*i.e.*, $\theta \approx \frac{\pi}{4}$) the target disagreement is close to its lowest value, while it is the opposite for the domain divergence. Thus, provided that the hyperparameters handling the trade-off between $\widehat{\mathrm{d}}_T(\rho_{\mathbf{w}})$ and $\widehat{\mathrm{e}}_S(\rho_{\mathbf{w}})$ are well chosen, the DALC minimization procedure is able to find a solution *close to* the one minimizing the target risk. On the contrary, for all hyperparameters, PBDA will prefer the solution that minimizes the source risk ($\theta \approx 0$), as it minimizes $\widehat{\mathrm{dis}}_{\rho_{\mathbf{w}}}(S,T)$ and $\widehat{\mathrm{R}}_S(G_{\rho_{\mathbf{w}}})$ simultaneously.

7. Experiments

Our domain adaptation algorithms PBDA[16] and DALC[17] have been evaluated on a toy problem and a sentiment dataset. In both cases, we minimize the objective function using a *Broyden-Fletcher-Goldfarb-Shanno method (BFGS)* implemented in the *scipy* python library.

7.1. Toy Problem: Two Inter-Twinning Moons

Figure 4 illustrates the behavior of the decision boundary of our algorithms PBDA and DALC on an intertwining moons toy problem,[18] where each moon corresponds to a label. The target domain, for which we have no label, is a rotation of the source one. The figure shows clearly that PBDA and DALC succeed to adapt to the target domain, even for a rotation angle of 50°. We see that our algorithms do not rely on the restrictive *covariate shift* assumption, as some source examples are misclassified. This behavior illustrates the PBDA and DALC trade-off in action, that concede some errors on the source sample to lower the disagreement on the target sample.

7.2. Sentiment Analysis Dataset

We consider the popular *Amazon reviews* dataset [63] composed of reviews of four types of *Amazon.com*© products (books, DVDs, electronics, kitchen appliances). Originally, the reviews are encoded in a bag-of-words representation (unigrams and bigrams) of approximately $100,000$ features, and the labels are user rating between one and five stars. For sake of simplicity, we adopt the pre-processing proposed by Chen et al. [64]. Hence, we tackle a binary classification task: a review is labeled $+1$ for a rank higher than 3 stars, and -1 for a rank lower or equal to 3 stars. Also, the feature space dimensionality is reduced as follows: Chen et al. [64] only kept the features that appear at least ten times in a particular domain adaptation task (about $40,000$ features remain), and pre-processed the data with a standard tf-idf re-weighting. Considering each

[16] PBDA's code is available here: `https://github.com/pgermain/pbda`

[17] DALC's code is available here: `https://github.com/GRAAL-Research/domain_adaptation_of_linear_classifiers`

[18] We generate each pair of moons with the `make_moons` function provided in `scikit-learn` [62].

type of product as a domain, we perform twelve domain adaptation tasks. For instance, "books→DVDs" corresponds to the task for which books is the source domain and DVDs the target one. The learning algorithms are trained with $2,000$ labeled source examples and $2,000$ unlabeled target examples, and we evaluate them on the same separate target test sets proposed by Chen et al. [64] (between $3,000$ and $6,000$ examples).

7.2.1. Experiment Details

PBDA and DALC with a linear kernel have been compared with:

- SVM learned only from the source domain without adaptation. We made use of the SVM-light library [65].
- PBGD3, presented in Section 3.3, and learned only from the source domain without adaptation.
- DASVM of Bruzzone and Marconcini [18], an iterative domain adaptation algorithm which aims to maximize iteratively a notion of margin on self-labeled target examples. We implemented DASVM with the LibSVM library [66].
- CODA of Chen et al. [64], a co-training domain adaptation algorithm, which looks iteratively for target features related to the training set. We used the implementation provided by the authors. Note that Chen et al. [64] have shown best results on the dataset considered in Section 7.2.

Each parameter is selected with a grid search via a classical 5-folds cross-validation (^{CV}) on the source sample for PBGD3 and SVM, and via a 5-folds reverse/circular validation (^{RCV}) on the source and the (unlabeled) target samples for CODA, DASVM, PBDA, and DALC. We describe this latter method in the following subsection. For PBDA, respectively DALC, we search on a 20×20 parameter grid for a Ω, respectively C, between 0.01 and 10^6 and a parameter A, respectively B, between 1.0 and 10^8, both on a logarithm scale.

Note that we did not compare the performance of our algorithms to state-osuf-the art (deep) representation learning techniques [22, 26, 27, 28, 29, 30], but to methods that learn a predictor directly on the original input space. Both strategies are not to be opposed, as one can learn a common representation space for the source and the target domains, and afterwards learn a predictor that optimizes an adaptation criteria within that new space.

7.2.2. A Note about the Reverse Validation

A crucial question in domain adaptation is the validation of the hyperparameters. One solution is to follow the principle proposed by Zhong et al. [67] which relies on the use of a reverse validation approach. This approach is based on a so-called reverse classifier evaluated on the source domain. We propose to follow it for tuning the parameters of DALC, PBDA, DASVM and CODA. Note that Bruzzone and Marconcini [18] have proposed a similar method, called circular validation, in the context of DASVM.

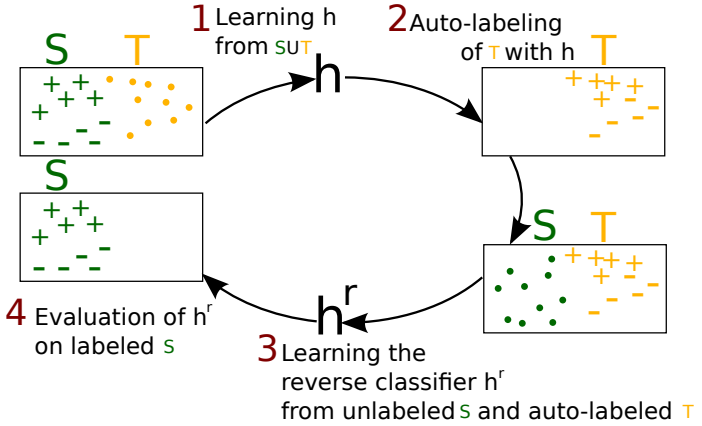


Figure 3: The principle of the reverse/circular validation in our setting.

Concretely, in the current setting, given k-folds on the source labeled sample, $S = S_1 \cup \ldots \cup S_k$, k-folds on the unlabeled target sample, $T = T_1 \cup \ldots \cup T_k$) and a learning algorithm (parametrized by a fixed tuple of hyperparameters), the reverse cross-validation risk on the i^{th} fold is computed as follows. Firstly, the source set $S \backslash S_i$ is used as a labeled sample and the target set $T \backslash T_i$ is used as an unlabeled sample for learning a classifier h'. Secondly, using the same algorithm, a reverse classifier h'^r is learned using the *self-labeled* sample $\{(\mathbf{x}, h'(\mathbf{x}))\}_{\mathbf{x} \in T \backslash T_i}$ as the source set and the unlabeled part of $S \backslash S_i$ as target sample. Finally, the reverse classifier h'^r is evaluated on S_i. We summarize this principle on Figure 3. The process is repeated k times to obtain the reverse cross-validation risk averaged across all folds.

7.2.3. Empirical Results

Table 1 contains the test accuracies on the sentiment analysis dataset. We make the following observations. Above all, the domain adaptation approaches provide the best average results, implying that tackling this problem with a domain adaptation method is reasonable. Then, our method DALC based on the novel domain adaptation analysis is the best algorithm overall on this task. Except for the two adaptive tasks between "electronics" and "DVDs", DALC is either the best one (five times), or the second one (five times). Moreover, according to a Wilcoxon signed rank test with a 5% significance level, we obtain a probability of 89.5% that DALC is better than PBDA. This test tends to confirm that the analysis with the new perspective improves the analysis based on a domains' divergence point of view. Moreover, PBDA is on average better than CODA, but less accurate than DASVM. However, PBDA is competitive: the results are not significantly different from CODA and DASVM. It is important to notice that DALC and PBDA are significantly faster than CODA and DASVM: These two algorithms are based on costly iterative procedures increasing the running time by at least a factor of five in comparison of DALC and PBDA. In fact, the clear

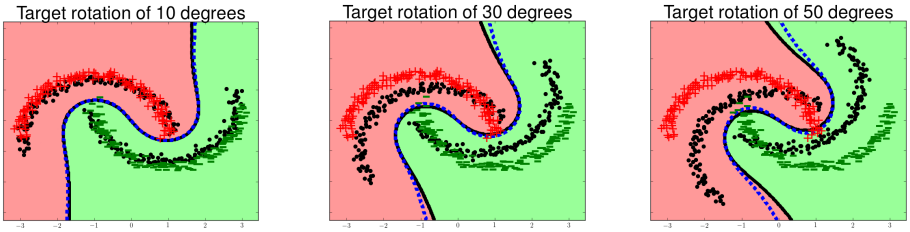


Figure 4: Decision boundaries of PBDA (in blue dashed) and DALC (in black) on the *intertwining moons* toy problem, for fixed parameters $\alpha = A = 1$ and $B = C = 1$, and an RBF kernel $k(\mathbf{x}, \mathbf{x}') = \exp(-\|\mathbf{x} - \mathbf{x}'\|^2)$. The target points are black. The positive, respectively negative, source points are red, respectively green.

Table 1: Error rates for the sentiment analysis dataset. B, D, E, K respectively denotes Books, DVDs, Electronics, Kitchen. In **bold** are highlighted the best results, in *italic* the second ones.

	PBGD3 ^{CV}	SVM ^{CV}	DASVM ^{RCV}	CODA ^{RCV}	PBDA ^{RCV}	DALC ^{RCV}
B→D	**0.174**	0.179	0.193	0.181	0.183	*0.178*
B→E	0.275	0.290	*0.226*	0.232	0.263	**0.212**
B→K	0.236	0.251	**0.179**	0.215	0.229	*0.194*
D→B	*0.192*	0.203	0.202	0.217	0.197	**0.186**
D→E	0.256	0.269	**0.186**	*0.214*	0.241	0.245
D→K	0.211	0.232	0.183	*0.181*	0.186	**0.175**
E→B	0.268	0.287	0.305	0.275	**0.232**	*0.240*
E→D	0.245	0.267	**0.214**	0.239	*0.221*	0.256
E→K	*0.127*	0.129	0.149	0.134	0.141	**0.123**
K→B	0.255	0.267	0.259	*0.247*	*0.247*	**0.236**
K→D	0.244	0.253	**0.198**	0.238	0.233	*0.225*
K→E	0.235	0.149	0.157	0.153	**0.129**	*0.131*
Average	0.226	0.231	*0.204*	0.210	0.208	**0.200**

advantage of the PAC-Bayesian approach is that we jointly optimize the terms of our bounds in one step.

8. Conclusion and Future Work

In this paper, we present two domain adaptation analyses for the PAC-Bayesian framework that focuses on models that take the form of a majority vote over a set of classifiers: The first one is based on a common principle in domain adaptation, and the second one brings a novel perspective on domain adaptation.

To begin, we follow the underlying philosophy of the seminal works of Ben-David et al. [33], Ben-David et al. [34] and Mansour et al. [40]; in other words, we derive an upper bound on the target risk (of the Gibbs classifier) thanks to a domains' divergence measure suitable for the PAC-Bayesian setting. We define this divergence as the average deviation between the disagreement over a set of

classifiers on the source and target domains. This leads to a bound that takes the form of a trade-off between the source risk, the domains' divergence and a term that captures the ability to adapt for the current task. Then, we propose another domain adaptation bound while taking advantage of the inherent behavior of the target risk in the PAC-Bayesian setting. We obtain a different upper bound that is expressed as a trade-off between the disagreement only on the target domain, the joint errors of the classifiers only on the source domain, and a term reflecting the worst-case error in regions where the source domain is non-informative. To the best of our knowledge, a crucial novelty of this contribution is that the trade-off is controlled by a domains' divergence: Contrary to our first bound, the divergence is not an additive term (as in many domain adaptation bounds) but is a factor weighing the importance of the source information.

Our analyses, combined with PAC-Bayesian generalization bounds, lead to two new domain adaptation algorithms for linear classifiers: PBDA associated with the previous works philosophy, and DALC associated with the novel perspective. The empirical experiments show that the two algorithms are competitive with other approaches, and that DALC outperforms significantly PBDA. We believe that our PAC-Bayesian analyses open the door to develop new domain adaptation methods by making use of the possibilities offered by the PAC-Bayesian theory, and give rise to new interesting directions of research, among which the following ones.

Firstly, the PAC-Bayesian approach allows one to deal with an *a priori* belief about the classifiers accuracy; in this paper we opted for a non-informative prior that is a Gaussian centered at the origin of the linear classifier space. The question of finding a relevant prior in a domain adaptation situation is an exciting direction which could also be exploited when some few target labels are available. Moreover, this notion of prior distribution could model information learned from previous tasks as pointed out by Pentina and Lampert [68], or from other views/representations of the data [69]. This suggests that we can extend our analyses to multisource domain adaptation [70, 71, 34, 72] and lifelong learning where the objective is to perform well on future tasks, for which no data has been observed so far [73].

Another promising issue is to address the problem of the hyperparameter selection. Indeed, the adaptation capability of our algorithms PBDA and DALC could be even put further with a specific PAC-Bayesian validation procedure. An idea would be to propose a (reverse) validation technique that takes into account some particular prior distributions. Another solution could be to explicitly control the neglected terms in the domain adaptation bound. This is also linked with model selection for domain adaptation tasks.

Besides, deriving a result similar to Equation (4) (the C-bound) for domain adaptation could be of high interest. Indeed, such an approach considers the first two moments of the margin of the weighted majority vote. This could help to take into account both margin information over unlabeled data and the distribution disagreement (these two elements seem of crucial importance in domain adaptation).

Concerning DALC, we would like to investigate the case where the domains'

divergence can be estimated, *i.e.*, when the covariate shift assumption holds or when some target labels are available. In this scenario, the domains' divergence might not be considered as a hyperparameter to tune.

The results obtained in this paper are dedicated to binary classification, one another interesting issue to tackle could be the case of multiclass or multilabel classification.

Last but not least, the non-estimable term of DALC—suggesting that the domains should live in the same regions—can be dealt with representation learning approach. This could be an incentive to combine DALC with existing representation learning techniques.

Acknowledgements

This work was supported in part by the French projects LIVES ANR-15-CE23-0026-03, and in part by NSERC discovery grant 262067, and by the European Research Council under the European Unions Seventh Framework Programme (FP7/2007-2013)/ERC grant agreement no 308036. Computations were performed on Compute Canada and Calcul Québec infrastructures (founded by CFI, NSERC and FRQ). We thank Christoph Lampert and Anastasia Pentina for helpful discussions. A part of the work of this paper was carried out while E. Morvant was affiliated with IST Austria, and while P. Germain was affiliated with Département d'informatique et de génie logiciel, Université Laval, Québec, Canada.

Appendix A. Some Tools

Lemma 1 (Hölder's Inequality). *Let S be a measure space and let $(p, q) \in [1, \infty]^2$ with $\frac{1}{p} + \frac{1}{q} = 1$. Then, for all measurable real-valued functions f and g on S,*

$$\|fg\|_1 \leq \|f\|_p \|g\|_q.$$

Lemma 2 (Markov's inequality). *Let Z be a random variable and $t \geq 0$, then*

$$\Pr\left(|Z| \geq t\right) \;\leq\; \frac{\mathbf{E}(|Z|)}{t}\,.$$

Lemma 3 (Jensen's inequality). *Let Z be an integrable real-valued random variable and g any function. If g is convex, then*

$$g(\mathbf{E}[Z]) \;\leq\; \mathbf{E}[g(Z)]\,.$$

Lemma 4 (from Lemma 3 of [74]). *Let $X{=}(X_1, \ldots, X_m)$ be a vector of i.i.d. random variables, $0 \leq X_i \leq 1$, with $\mathbf{E}\, X_i = \mu$. Denote $X'{=}(X'_1, \ldots, X'_m)$, where X'_i is the unique Bernoulli ($\{0,1\}$-valued) random variable with $\mathbf{E}\, X'_i = \mu$. If $f : [0,1]^m \to \mathbb{R}$ is convex, then*

$$\mathbf{E}[f(X)] \;\leq\; \mathbf{E}[f(X')]\,.$$

Lemma 5 (from Inequalities 1 and 2 of [74]). *Let $X = (X_1, \ldots, X_m)$ be a vector of i.i.d. random variables, $0 \leq X_i \leq 1$. Then*

$$\sqrt{m} \;\leq\; \mathbf{E} \exp\left[m \operatorname{kl}\left(\frac{1}{m} \sum_{i=1}^{m} X_i \,\middle\|\, \mathbf{E}[X_i] \right) \right] \;\leq\; 2\sqrt{m}\,.$$

Lemma 6. (Change of measure inequality[19]) *For any set $\mathcal{H}$, for any distributions π and ρ on $\mathcal{H}$, and for any measurable function $\phi : \mathcal{H} \to \mathbb{R}$, we have*

$$\mathop{\mathbf{E}}_{f\sim\rho} \phi(f) \;\leq\; \mathrm{KL}(\rho\|\pi) + \ln\left(\mathop{\mathbf{E}}_{f\sim\pi} e^{\phi(f)} \right).$$

Lemma 7 (from Theorem 25 of [49]). *Given any set $\mathcal{H}$, and any distributions π and ρ on $\mathcal{H}$, let $\hat{\rho}$ and $\hat{\pi}$ two distributions over $\mathcal{H}^2$ such that $\hat{\rho}(h, h') = \rho(h)\rho(h')$ and $\hat{\pi}(h, h') = \pi(h)\pi(h')$. Then*

$$\mathrm{KL}(\hat{\rho}\|\hat{\pi}) \;=\; 2\,\mathrm{KL}(\rho\|\pi)\,.$$

Appendix B. Proof of Equation (10)

Given $(\mathbf{x}, y) \in \mathbb{R}^d \times \{-1, 1\}$ and $\mathbf{w} \in \mathbb{R}^d$, we consider—without loss of generality—a vector basis where $y\frac{\mathbf{x}}{\|\mathbf{x}\|}$ is the first coordinate. Thus, the first component of any vector $\mathbf{w}' \in \mathbb{R}^d$ is given by $w'_1 = y\frac{\mathbf{w}'\cdot\mathbf{x}}{\|\mathbf{x}\|}$, which leads to

$$\begin{aligned}
\mathop{\mathbf{E}}_{h_{\mathbf{w}'}\sim\rho_{\mathbf{w}}} \mathcal{L}_{0\text{-}1}\big(h_{\mathbf{w}'}(\mathbf{x}), y\big) &= \mathop{\mathbf{E}}_{h_{\mathbf{w}'}\sim\rho_{\mathbf{w}}} \mathrm{I}\big[h_{\mathbf{w}'}(\mathbf{x}) \neq y\big] \\
&= \mathop{\mathbf{E}}_{h_{\mathbf{w}'}\sim\rho_{\mathbf{w}}} \mathrm{I}\big[y\,\mathbf{w}'\cdot\mathbf{x} \leq 0\big] \\
&= \int_{\mathbb{R}^d} \frac{1}{\sqrt{(2\boldsymbol{\pi})^d}} \exp\big(-\tfrac{1}{2}\|\mathbf{w}' - \mathbf{w}\|^2\big)\ \mathrm{I}\big[y\,\mathbf{w}'\cdot\mathbf{x} \leq 0\big]\, d\,\mathbf{w}' \\
&= \int_{\mathbb{R}} \frac{1}{\sqrt{2\boldsymbol{\pi}}} \exp\big(-\tfrac{1}{2}(w'_1 - w_1)^2\big)\ \mathrm{I}\big[w'_1 \leq 0\big]\, dw'_1 \\
&= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\boldsymbol{\pi}}} \exp\big(-\tfrac{1}{2}\,t^2\big)\ \mathrm{I}\big[t \leq -w_1\big]\, dt \\
&= \int_{-\infty}^{-w_1} \frac{1}{\sqrt{2\boldsymbol{\pi}}} \exp\big(-\tfrac{1}{2}\,t^2\big)\ dt \\
&= 1 - \Pr_{t\sim\mathcal{N}(0,1)}\Big(t \;\leq\; y\,\tfrac{\mathbf{w}\cdot\mathbf{x}}{\|\mathbf{x}\|}\Big) \\
&= \Phi_{\mathrm{R}}\left(y\,\tfrac{\mathbf{w}\cdot\mathbf{x}}{\|\mathbf{x}\|}\right),
\end{aligned}$$

[19] See [75, Lemma 4], [76, Equation 20], or [49, Lemma 17].

where we used $y\,\mathbf{w}'\cdot\mathbf{x} = w_1'\|\mathbf{x}\|$, $t \coloneqq w_1' - w_1$, $w_1 = y\frac{\mathbf{w}\cdot\mathbf{x}}{\|\mathbf{x}\|}$, and the definition of Φ_{R} given by Equation (11). We then have

$$\mathrm{R}_{\mathcal{D}}(G_{\rho_{\mathbf{w}}}) \;=\; \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim P_S}\; \mathop{\mathbf{E}}_{h_{\mathbf{w}'}\sim\rho_{\mathbf{w}}} \mathcal{L}_{0\text{-}1}\big(h_{\mathbf{w}'}(\mathbf{x}), y\big) \;=\; \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim P_S} \Phi_{\mathrm{R}}\left(y\,\tfrac{\mathbf{w}\cdot\mathbf{x}}{\|\mathbf{x}\|}\right).$$

Appendix C. Proof of Theorem 6

Proof. We use the shorthand notation

$$\mathcal{L}_{\mathcal{D}}(h) = \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \ell(h,\mathbf{x},y) \;\text{ and }\; \mathcal{L}_S(h) = \tfrac{1}{m}\sum_{(\mathbf{x},y)\in S} \ell(h,\mathbf{x},y)\,.$$

Consider any convex function $\Delta : [0,1]\times[0,1] \to \mathbb{R}$. Applying consecutively Jensen's Inequality (Lemma 3) and the *change of measure inequality* (Lemma 6), we obtain

$$\begin{aligned}\forall \rho \text{ on } \mathcal{H}, \quad m\times\Delta\left(\mathop{\mathbf{E}}_{h\sim\rho}\mathcal{L}_S(h), \mathop{\mathbf{E}}_{h\sim\rho}\mathcal{L}_{\mathcal{D}}(h)\right) \;&\leq\; \mathop{\mathbf{E}}_{h\sim\rho} m\times\Delta\left(\mathcal{L}_S(h), \mathcal{L}_{\mathcal{D}}(h)\right)\\ &\leq\; \mathrm{KL}(\rho\|\pi) + \ln\Big[X_\pi(S)\Big],\end{aligned}$$

with

$$X_\pi(S) = \mathop{\mathbf{E}}_{h\sim\pi} e^{m\times\Delta(\mathcal{L}_S(h),\,\mathcal{L}_{\mathcal{D}}(h))}.$$

Then, Markov's Inequality (Lemma 2) gives

$$\Pr_{S\sim\mathcal{D}^m}\left(X_\pi(S) \leq \tfrac{1}{\delta}\mathop{\mathbf{E}}_{S'\sim\mathcal{D}^m} X_\pi(S')\right) \;\geq\; 1-\delta\,,$$

and

$$\begin{aligned}\mathop{\mathbf{E}}_{S'\sim\mathcal{D}^m} X_\pi(S') \;&=\; \mathop{\mathbf{E}}_{S'\sim\mathcal{D}^m}\mathop{\mathbf{E}}_{h\sim\pi} e^{m\times\Delta(\mathcal{L}_{S'}(h),\,\mathcal{L}_{\mathcal{D}}(h))}\\ &=\; \mathop{\mathbf{E}}_{h\sim\pi}\mathop{\mathbf{E}}_{S'\sim\mathcal{D}^m} e^{m\times\Delta(\mathcal{L}_{S'}(h),\,\mathcal{L}_{\mathcal{D}}(h))}\\ &\leq\; \mathop{\mathbf{E}}_{h\sim\pi}\sum_{k=0}^{m}\binom{k}{m}(\mathcal{L}_{\mathcal{D}}(h))^k(1-\mathcal{L}_{\mathcal{D}}(h))^{m-k}e^{m\times\Delta\left(\frac{k}{m},\,\mathcal{L}_{\mathcal{D}}(h)\right)}, \qquad \text{(C.1)}\end{aligned}$$

where the last inequality is given by Lemma 5 (we have equality when the output of ℓ is in $\{0,1\}$). As shown in Germain et al. [44, Corollary 2.2], by fixing

$$\Delta(q,p) = -\alpha\times q - \ln[1-p\,(1-e^{-\alpha})]\,,$$

Line (C.1) becomes equal to 1, and then $\mathbf{E}_{S'\sim\mathcal{D}^m} X_\pi(S') \leq 1$. Hence, with probability $1-\delta$ over the choice of $S\in\mathcal{D}^m$, we have

$$\forall\rho \text{ on } \mathcal{H},\; -\alpha\mathop{\mathbf{E}}_{h\sim\rho}\mathcal{L}_S(h) - \ln[1-\mathop{\mathbf{E}}_{h\sim\rho}\mathcal{L}_{\mathcal{D}}(h)\,(1-e^{-\alpha})] \leq \frac{\mathrm{KL}(\rho\|\pi) + \ln\frac{1}{\delta}}{m}\,.$$

By reorganizing the terms, we have, with probability $1-\delta$ over the choice of $S \in \mathcal{D}^m$,

$$\forall \rho \text{ on } \mathcal{H},\ \mathop{\mathbf{E}}_{h\sim\rho} \mathcal{L}_{\mathcal{D}}(h) \leq \frac{1}{1-e^{-\alpha}} \left[1 - \exp\left(-\alpha \mathop{\mathbf{E}}_{h\sim\rho} \mathcal{L}_S(h) - \frac{\mathrm{KL}(\rho\|\pi) + \ln\frac{1}{\delta}}{m}\right)\right].$$

The final result is obtained by using the inequality $1-\exp(-z) \leq z$. ■

References

1. Germain P, Habrard A, Laviolette F, Morvant E. A PAC-Bayesian approach for domain adaptation with specialization to linear classifiers. In: *International Conference on Machine Learning*. 2013:738–46.

2. Germain P, Habrard A, Laviolette F, Morvant E. A new PAC-Bayesian perspective on domain adaptation. In: *International Conference on Machine Learning*; vol. 48. 2016:859–68.

3. Jiang J. A literature survey on domain adaptation of statistical classifiers. Tech. Rep.; CS Department at Univ. of Illinois at Urbana-Champaign; 2008.

4. Quionero-Candela J, Sugiyama M, Schwaighofer A, Lawrence N. Dataset Shift in Machine Learning. MIT Press; 2009. ISBN 0262170051, 9780262170055.

5. Margolis A. A literature review of domain adaptation with unlabeled data. Tech. Rep.; University of Washington; 2011.

6. Wang M, Deng W. Deep visual domain adaptation: A survey. *Neurocomputing* 2018;.

7. Kouw WM, Loog M. A review of single-source unsupervised domain adaptation. *CoRR* 2019;`arXiv:1901.05335`.

8. Ievgen Redko Emilie Morvant AHMSYB. Domain Adaptation Theory: Available Theoretical Results. ISTE Press - Elsevier; 2019. ISBN 9781785482366.

9. Pan S, Yang Q. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on* 2010;22(10):1345–59.

10. Ben-David S, Lu T, Luu T, Pal D. Impossibility theorems for domain adaptation 2010;9:129–36.

11. Ben-David S, Urner R. On the hardness of domain adaptation and the utility of unlabeled target samples. In: *Proceedings of Algorithmic Learning Theory*. 2012:139–53.

12. Ben-David S, Urner R. Domain adaptation-can quantity compensate for quality? *Ann Math Artif Intell* 2014;70(3):185–202.

13. Huang J, Smola A, Gretton A, Borgwardt K, Schölkopf B. Correcting sample selection bias by unlabeled data. In: *Advances in Neural Information Processing Systems*. 2006:601–8.

14. Sugiyama M, Nakajima S, Kashima H, von Bünau P, Kawanabe M. Direct importance estimation with model selection and its application to covariate shift adaptation. In: *Advances in Neural Information Processing Systems*. 2007:.

15. Cortes C, Mansour Y, Mohri M. Learning bounds for importance weighting. In: *Advances in Neural Information Processing Systems*. 2010:442–50.

16. Cortes C, Mohri M, Medina AM. Adaptation algorithm and theory based on generalized discrepancy. In: *ACM SIGKDD*. 2015:169–78.

17. Sugiyama M, Nakajima S, Kashima H, Buenau PV, Kawanabe M. Direct importance estimation with model selection and its application to covariate shift adaptation. In: *Advances in Neural Information Processing Systems*. 2008:1433–40.

18. Bruzzone L, Marconcini M. Domain adaptation problems: A DASVM classification technique and a circular validation strategy. *Transaction Pattern Analysis and Machine Intelligence* 2010;32(5):770–87.

19. Habrard A, Peyrache JP, Sebban M. Iterative self-labeling domain adaptation for linear structured image classification. *International Journal on Artificial Intelligence Tools* 2013;22(05).

20. Morvant E. Domain adaptation of weighted majority votes via perturbed variation-based self-labeling. *Pattern Recognition Letters* 2015;51:37–43.

21. Glorot X, Bordes A, Bengio Y. Domain adaptation for large-scale sentiment classification: A deep learning approach. In: *Proceedings of the International Conference on Machine Learning*. 2011:513–20.

22. Chen M, Xu ZE, Weinberger KQ, Sha F. Marginalized denoising autoencoders for domain adaptation. In: *International Conference on Machine Learning*. 2012:.

23. Courty N, Flamary R, Tuia D, Rakotomamonjy A. Optimal transport for domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2016;.

24. Courty N, Flamary R, Tuia D, Rakotomamonjy A. Optimal transport for domain adaptation. *IEEE Trans Pattern Anal Mach Intell* 2017;39(9):1853–65.

25. Li J, Lu K, Huang Z, Zhu L, Shen HT. Transfer independently together: A generalized framework for domain adaptation. *IEEE Trans Cybernetics* 2019;49(6):2144–55.

26. Ganin Y, Ustinova E, H A, Germain P, Larochelle H, Laviolette F, Marchand M, Lempitsky V. Domain-adversarial training of neural networks. *Journal of Machine Learning Research* 2016;17(59):1–35.

27. Ding Z, Fu Y. Deep domain generalization with structured low-rank constraint. *IEEE Trans Image Processing* 2018;27(1):304–13.

28. Shu R, Bui HH, Narui H, Ermon S. A DIRT-T approach to unsupervised domain adaptation. In: *International Conference on Learning Representations*. 2018:.

29. Li J, Lu K, Huang Z, Zhu L, Shen HT. Heterogeneous domain adaptation through progressive alignment. *IEEE Trans Neural Netw Learning Syst* 2019;30(5):1381–91.

30. Sebag AS, Heinrich L, Schoenauer M, Sebag M, Wu LF, Altschuler SJ. Multi-domain adversarial learning. In: *International Conference on Learning Representations*. 2019:.

31. Kuzborskij I, Orabona F. Stability and hypothesis transfer learning. In: *International Conference on Machine Learning*. 2013:942–50.

32. Kuzborskij I. Theory and algorithms for hypothesis transfer learning. Ph.D. thesis; Ecole Polytechnique Fédérale de Lausanne; 2018.

33. Ben-David S, Blitzer J, Crammer K, Pereira F. Analysis of representations for domain adaptation. In: *Advances in Neural Information Processing Systems*. 2006:137–44.

34. Ben-David S, Blitzer J, Crammer K, Kulesza A, Pereira F, Vaughan J. A theory of learning from different domains. *Machine Learning* 2010;79(1-2):151–75.

35. Li X, Bilmes J. A Bayesian divergence prior for classifier adaptation. In: *International Conference on Artificial Intelligence and Statistics*. 2007:275–82.

36. Zhang C, Zhang L, Ye J. Generalization bounds for domain adaptation. In: *Advances in Neural Information Processing Systems*. 2012:.

37. Morvant E, Habrard A, Ayache S. Parsimonious Unsupervised and Semi-Supervised Domain Adaptation with Good Similarity Functions. *Knowledge and Information Systems* 2012;33(2):309–49.

38. Cortes C, Mohri M. Domain adaptation and sample bias correction theory and algorithm for regression. *Theoretical Computer Science* 2014;519:103–26.

39. Redko I, Habrard A, Sebban M. Theoretical analysis of domain adaptation with optimal transport. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer; 2017:737–53.

40. Mansour Y, Mohri M, Rostamizadeh A. Domain adaptation: Learning bounds and algorithms. In: *Conference on Learning Theory*. 2009:19–30.

41. Cortes C, Mohri M. Domain adaptation in regression. In: *Algorithmic Learning Theory*. Springer; 2011:308–23.

42. Mansour Y, Mohri M, Rostamizadeh A. Multiple source adaptation and the Rényi divergence. In: *Conference on Uncertainty in Artificial Intelligence*. AUAI Press; 2009:367–74.

43. McAllester DA. Some PAC-Bayesian theorems. *Machine Learning* 1999;37:355–63.

44. Germain P, Lacasse A, Laviolette F, Marchand M. PAC-Bayesian learning of linear classifiers. In: *International Conference on Machine Learning*. 2009:.

45. Parrado-Hernández E, Ambroladze A, Shawe-Taylor J, Sun S. PAC-Bayes bounds with data dependent priors. *Journal of Machine Learning Research* 2012;13:3507–31.

46. Dietterich TG. Ensemble methods in machine learning. In: *International workshop on multiple classifier systems*. Springer; 2000:1–15.

47. Re M, Valentini G. Ensemble methods: a review. *Advances in machine learning and data mining for astronomy* 2012;:563–82.

48. Lacasse A, Laviolette F, Marchand M, Germain P, Usunier N. PAC-Bayes bounds for the risk of the majority vote and the variance of the Gibbs classifier. In: *Advances in Neural Information Processing Systems*. 2006:769–76.

49. Germain P, Lacasse A, Laviolette F, Marchand M, Roy JF. Risk bounds for the majority vote: From a PAC-Bayesian analysis to a learning algorithm. *Journal of Machine Learning Research* 2015;16:787–860.

50. Catoni O. PAC-Bayesian supervised classification: the thermodynamics of statistical learning; vol. 56. Inst. of Mathematical Statistic; 2007.

51. Seeger M. PAC-Bayesian generalization bounds for gaussian processes. *Journal of Machine Learning Research* 2002;3:233–69.

52. Langford J. Tutorial on practical prediction theory for classification. *Journal of Machine Learning Research* 2005;6:273–306.

53. Germain P, Habrard A, Laviolette F, Morvant E. PAC-Bayesian theorems for domain adaptation with specialization to linear classifiers. *Research Report arXiv preprint arXiv:150306944* 2015;.

54. Germain P, Lacasse A, Laviolette F, Marchand M, Shanian S. From PAC-Bayes bounds to KL regularization. In: *Advances in Neural Information Processing Systems*. 2009:603–10.

55. McAllester DA, Keshet J. Generalization bounds and consistency for latent structural probit and ramp loss. In: *Advances in Neural Information Processing System*. 2011:2205–12.

56. Langford J, Shawe-Taylor J. PAC-Bayes & margins. In: *Advances in Neural Information Processing Systems*. 2002:439–46.

57. Ambroladze A, Parrado-Hernández E, Shawe-Taylor J. Tighter PAC-Bayes bounds. In: *Advances in Neural Information Processing Systems*. 2006:9–16.

58. Schölkopf B, Herbrich R, Smola AJ. A generalized representer theorem. In: *Annual Conference on Computational Learning Theory, and European Conference on Computational Learning Theory*. 2001:416–26.

59. Ben-David S, Shalev-Shwartz S, Urner R. Domain adaptation—can quantity compensate for quality? In: *International Symposium on Artificial Intelligence and Mathematics (ISAIM)*. 2012:.

60. Shimodaira H. Improving predictive inference under covariate shift by weighting the log-likelihood function. *J Statist Plann Inference* 2000;90(2):227–44.

61. Urner R, Shalev-Shwartz S, Ben-David S. Access to unlabeled data can speed up prediction time. In: *International Conference on Machine Learning*. 2011:641–8.

62. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E. Scikit-learn: Machine learning in Python. *JMLR* 2011;12:2825–30.

63. Blitzer J, McDonald R, Pereira F. Domain adaptation with structural correspondence learning. In: *Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics; 2006:120–8.

64. Chen M, Weinberger KQ, Blitzer J. Co-training for domain adaptation. In: *Advances in Neural Information Processing Systems*. 2011:2456–64.

65. Joachims T. Transductive inference for text classification using support vector machines. In: *International Conference on Machine Learning*. 1999:200–9.

66. Chang CC, Lin CJ. LibSVM: a library for support vector machines; 2001. `www.csie.ntu.edu.tw/~cjlin/libsvm`.

67. Zhong E, Fan W, Yang Q, Verscheure O, Ren J. Cross validation framework to choose amongst models and datasets for transfer learning. In: *Machine Learning and Knowledge Discovery in Databases*; vol. 6323 of *LNCS*. Springer; 2010:547–62.

68. Pentina A, Lampert C. A PAC-Bayesian bound for lifelong learning. In: *JMLR W&CP, Proceedings of International Conference on Machine Learning*; vol. 32. 2014:991–9.

69. Goyal A, Morvant E, Germain P, Amini MR. Pac-bayesian analysis for a two-step hierarchical multiview learning approach. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer; 2017:205–21.

70. Crammer K, Kearns M, Wortman J. Learning from multiple sources. *Advances in Neural Information Processing Systems* 2007;19:321.

71. Mansour Y, Mohri M, Rostamizadeh A. Domain adaptation with multiple sources. In: *Advances in Neural Information Processing Systems*. 2009:1041–8.

72. Hoffman J, Mohri M, Zhang N. Algorithms and theory for multiple-source adaptation. In: *Conference on Neural Information Processing Systems*. 2018:.

73. Thrun S, Mitchell TM. Lifelong robot learning. *Robotics and Autonomous Systems* 1995;15(1-2):25–46.

74. Maurer A. A note on the PAC Bayesian theorem. *CoRR* 2004;cs.LG/0411099.

75. Seldin Y, Tishby N. PAC-Bayesian analysis of co-clustering and beyond. *Journal of Machine Learning Research* 2010;11:3595–646.

76. McAllester D. A PAC-Bayesian tutorial with a dropout bound. *CoRR* 2013;abs/1307.2118.