A Bayesian Divergence Prior for Classifier Adaptation

Xiao Li[*] and **Jeff Bilmes**[*]
Dept. of Electrical Engineering.
University of Washington, Seattle WA 98195-2500

Abstract

Adaptation of statistical classifiers is critical when a target (or testing) distribution is different from the distribution that governs training data. In such cases, a classifier optimized for the training distribution needs to be adapted for optimal use in the target distribution. This paper presents a Bayesian "divergence prior" for generic classifier adaptation. Instantiations of this prior lead to simple yet principled adaptation strategies for a variety of classifiers, which yield superior performance in practice. In addition, this paper derives several adaptation error bounds by applying the divergence prior in the PAC-Bayesian setting.

1 Introduction

Many statistical learning techniques assume that training and test samples are generated from the same underlying distribution. Often, however, an "unadapted classifier" is trained on samples drawn from a training distribution that is close to but not the same as the target (or testing) distribution. Moreover, in many applications, while there may be essentially an unlimited amount of labeled "training data," only a small amount of labeled "adaptation data" drawn from the target distribution is available. The problem of adaptation, then, is to utilize the unadapted classifier and the limited adaptation data to obtain a new classifier optimized for the target distribution. For example, in speech and handwriting recognition, an unadapted classifier may be trained on a database consisting of samples from an enormous number of users. The target distribution would correspond only to a specific user, from whom it would be unrealistic to obtain a large amount of labeled data. A system, however, should be able to quickly adapt to that user using as small an amount of adaptation data as possible. Note that in our setting, the training data is no longer available at adaptation time — the only information preserved from training is the unadapted classifier; this happens often in real-world scenarios, where an end user can hardly afford to store and manipulate a large amount of training data directly.

The adaptation problem studied in this paper can be considered as a special setting of *multi-task learning* [1, 2, 3, 4] in that learning the unadapted and the adapted model can be viewed as two related tasks. In our paradigm, however, we are only concerned with the performance of the target task rather than the "average" performance over all tasks. In fact, there has been a large amount of practical work on adaptation developed under similar assumptions. Adaptation of generative models, such as Gaussian mixture models (GMM), has been vastly investigated in the area of speech recognition [5, 6]. Regarding discriminative classifiers, different adaptation strategies have been proposed for support vector machines (SVMs) [7, 8], multi-layer perceptrons (MLPs) [1, 2, 9], and conditional maximum entropy (MaxEnt) models [10]. While these algorithms have demonstrated empirically the effectiveness of adaptation in various tasks, it is interesting to ask whether there is a principled approach that unifies these different treatments. Moreover, a more fundamental question would be whether we can relate the adaptation sample complexity to the divergence between training and target distributions.

This work makes an initial attempt to answer these questions. We utilize the concept of "accuracy-regularization", where we use a Bayesian "divergence prior" (on the function space) as the regularizer. In this regard, our method is strongly related to hierarchical Bayesian inference, *e.g.* [11]. The key difference is that our proposed prior is essentially a posterior determined by a *training distribution* rather than by a train-

[*] This material is based on work supported by the National Science Foundation under grant IIS-0326382.

ing set. This formulation unifies adaptation strategies for a variety of classifiers, and relates the adaptation error bounds to the divergence between training and target distributions in the PAC-Bayesian setting.

2 Inductive learning vs. adaptation

Throughout this paper, all densities are taken w.r.t. the Lebesgue measure in their respective spaces. We assume that $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$ is a pair of random variables where $\mathcal{X}$ is a feature space and $\mathcal{Y} = \{\pm 1\}$ is a set of class labels (binary in our case). Taking the Bayesian perspective, we further assume that a decision function (or a classifier) $f \in \mathcal{F} : \mathcal{X} \to \mathcal{Y}$ is a random variable and that it has a "standard prior" distribution $\pi(f)$ (which has to be chosen before seeing any training or test data, often based on domain knowledge). A fundamental problem in inductive learning is to find such an f that minimizes the true risk $R_{p(\mathbf{x},y)}(f) \triangleq \mathrm{E}_{p(\mathbf{x},y)}[Q(f(\mathbf{x}), y)]$ under certain loss function $Q(\cdot)$ (*e.g.* 0-1 loss, log loss, or hinge loss).

A key assumption in inductive learning is that training and test samples are generated from the same underlying distribution. This paper is interested in the case where the target sample distribution, denoted by $p^{ad}(\mathbf{x}, y)$, varies from that of training, denoted by $p^{tr}(\mathbf{x}, y)$. We formulate the adaptation problem as follows: given a training distribution $p^{tr}(\mathbf{x}, y)$ and a function space $\mathcal{F}$ with a finite VC dimension [12], we assume the availability of an "unadapted classifier" (learned from a sufficiently large amount of training data), which is an approximately correct estimate of

$$f^{tr} \in \operatorname*{argmin}_{f \in \mathcal{F}} R_{p^{tr}(\mathbf{x},y)}(f), \tag{1}$$

In this paper, we let f^{tr} denote the unadapted model for simplicity. Secondly, we assume that m adaptation samples are drawn, in an i.i.d. fashion, from a target distribution,

$$\mathcal{D}_m^{ad} = \{(\mathbf{x}_i, y_i) | (\mathbf{x}_i, y_i) \sim p^{ad}(\mathbf{x}, y)\}_{i=1}^m, \tag{2}$$

which we call "adaptation data". The goal of adaptation is to produce an adapted classifier $\hat{f}$ that is as close as possible to our *desired* decision function $f^{ad} \in \operatorname{argmin}_{f \in \mathcal{F}} R_{p^{ad}(\mathbf{x},y)}(f)$, by combining the two sources of information f^{tr} and $\mathcal{D}_m^{ad}$.

There are two extreme strategies for learning $\hat{f}$. First, we can train a classifier that minimizes the empirical risk $R_{\text{emp}}(f) \triangleq \frac{1}{m}\sum_{i=1}^m Q(f(\mathbf{x}_i), y_i)$, $(\mathbf{x}_i, y_i) \in \mathcal{D}_m^{ad}$, but this might cause overfitting when m is small. At the other extreme, we can simply let $\hat{f} = f^{tr}$, but this might yield a high empirical risk on $\mathcal{D}_m^{ad}$, especially when $p^{ad}(\mathbf{x}, y)$ significantly differs from $p^{tr}(\mathbf{x}, y)$. This work seeks a strategy between these two extremes in which one would hope to achieve better performance.

3 A Bayesian divergence prior

As mentioned in the introduction, we propose to use an "accuracy-regularization" objective for adaptation, where we minimize the empirical risk on the adaptation data while maximizing a Bayesian "divergence prior" $p_{\text{div}}(f)$ (which will be defined shortly). This divergence prior should be distinguished from the standard prior $\pi(f)$ in that the latter is chosen *before* training the unadapted model, whereas the former is chosen *after* the unadapted model is obtained. Specifically, our adaptation objective is as follows,

$$\min_f R_{\text{emp}}(f) - \lambda \ln p_{\text{div}}(f) \tag{3}$$

where λ is a regularization coefficient, and the divergence prior $p_{\text{div}}(f)$ is defined as

$$\ln p_{\text{div}}(f) = \mathrm{E}_{p^{tr}(\mathbf{x},y)}[\ln p(f|\mathbf{x}, y)] + \gamma \tag{4}$$

In this definition, $p^{tr}(\mathbf{x}, y)$ again is the training distribution, $p(f|\mathbf{x}, y)$ is the posterior probability of a classifier given a sample (which will be discussed in detail in the following subsections), and γ is a normalization constant such that $p_{\text{div}}(f)$ sums to unity. This prior essentially can be viewed as an approximate posterior probability of a classifier given a training distribution. The reason we choose such a prior is that, as will be seen shortly, $p_{\text{div}}(f)$ incorporates information from both the standard prior $\pi(f)$ and the unadapted model f^{tr}, and that it assigns higher probabilities to classifiers "closer to" f^{tr}. More importantly, this prior analytically relates $p_{\text{div}}(f^{ad})$ (the prior probability of the *desired* classifier), and hence the generalization error bound at f^{ad}, to the divergence between training and target distributions.

Our adaptation objective in Equation (3), therefore, becomes a tradeoff between fitting the adaptation data and staying "close" to the unadapted classifier. Next, we discuss its instantiations for generative and discriminative classifiers respectively.

3.1 Generative classifiers

We first study classifiers using generative models, which have long been used in speech, text, vision and bioinformatics applications. In such a case, the function space $\mathcal{F}$ consists of generative models f that describe the sample distribution $p(\mathbf{x}, y|f)$ (here we slightly abuse notation by letting f denote a generative model instead of a decision function). The classification decision is made via $\operatorname{argmax}_{y \in \mathcal{Y}} \ln p(\mathbf{x}, y|f)$. If we use $Q(\cdot) = -\ln p(\mathbf{x}, y|f)$, the unadapted model f^{tr} in

Equation (1) is the *true* model generating the training distribution, *i.e.*, $p(\mathbf{x}, y|f^{tr}) = p^{tr}(\mathbf{x}, y)$. Similarly, we have $p(\mathbf{x}, y|f^{ad}) = p^{ad}(\mathbf{x}, y)$. Note that by doing this, we implicitly assume that our function space $\mathcal{F}$ contains the true generative models in both cases.

Furthermore, applying Bayes rule, the posterior probability in Equation (4) can be expressed as

$$p(f|\mathbf{x}, y) = \frac{p(\mathbf{x}, y|f)\pi(f)}{p(\mathbf{x}, y)} = \frac{p(\mathbf{x}, y|f)\pi(f)}{\int p(\mathbf{x}, y|f)\pi(f)\, df} \quad (5)$$

where $\pi(f)$ is again the standard prior chosen before seeing the training data. Plugging Equation (5) into (4) leads to the following theorem,

Theorem 3.1 *For generative classifiers, the divergence prior defined in Equation* (4) *satisfies*

$$\ln p_{div}(f) = -D(p(\mathbf{x}, y|f^{tr})||p(\mathbf{x}, y|f)) + \ln \pi(f) + \beta \quad (6)$$

where $\beta > 0$ is a normalization constant.

Proof

$$\begin{aligned} & \ln p_{\text{div}}(f) = \mathrm{E}_{p(\mathbf{x},y|f^{tr})}[\ln p(f|\mathbf{x}, y)] + \gamma \\ = \; & \mathrm{E}_{p(\mathbf{x},y|f^{tr})} \ln[\frac{p(\mathbf{x}, y|f)\pi(f)}{p(\mathbf{x}, y|f^{tr})} \cdot \frac{p(\mathbf{x}, y|f^{tr})}{p(\mathbf{x}, y)}] + \gamma \\ = \; & -D(p(\mathbf{x}, y|f^{tr})||p(\mathbf{x}, y|f)) + \ln \pi(f) \\ & +D(p(\mathbf{x}, y|f^{tr})||p(\mathbf{x}, y)) + \gamma \end{aligned}$$

Letting $\beta \triangleq D(p(\mathbf{x}, y|f^{tr})||p(\mathbf{x}, y)) + \gamma$, we have

$$\begin{aligned} 1 \; & = \int p_{\text{div}}(f)\, df \\ = \; & \int \exp\{-D(p(\mathbf{x}, y|f^{tr})||p(\mathbf{x}, y|f)) + \ln \pi(f) + \beta\}\, df \\ < \; & \int \exp\{\ln \pi(f) + \beta\}\, df = \exp \beta \end{aligned}$$

The inequality follows since $D(p(\mathbf{x}, y|f^{tr})||p(\mathbf{x}, y|f)) \geq 0$ with equality achieved only at $f = f^{tr}$. Therefore we have $\beta > 0$. ■

This result explains why we use the term "divergence prior"; the prior is essentially determined by the KL-divergence between the sample distribution generated by the unadapted model and that generated from the model of interest, and it favors those models "similar to" the unadapted model. In particular, we inspect the prior probability of our desired model, *i.e.*, $\ln p_{\text{div}}(f^{ad}) = -D(p^{tr}||p^{ad}) + \ln \pi(f^{ad}) + \beta$, from which we can draw some intuitive insights about why using the divergence would help. As implied in the above equation, if $D(p^{tr}||p^{ad}) < \beta$, we have $p_{\text{div}}(f^{ad}) > \pi(f^{ad})$, and thus we are more likely to learn the desired model using the divergence prior than using only the standard prior. Since $\beta > 0$, there must exist distributions p^{ad} for which the above statement is true.

Consequently, our adaptation objective for generative classifiers becomes

$$\min_f R_{\text{emp}}(f) + \lambda D(p(\mathbf{x}, y|f^{tr})||p(\mathbf{x}, y|f)) - \lambda \pi(f) \quad (7)$$

When $\pi(f)$ is uniform[1], this objective asks to minimize the empirical risk as well as the KL-divergence between the joint distributions.

The divergence prior, and hence the corresponding adaptation objective, can be easily derived if a joint distribution $p(\mathbf{x}, y|f)$ has a close-form KL-divergence. An important example is a class-conditional d-dimensional Gaussian distribution, *i.e.*, $p(\mathbf{x}|y, f^{tr}) = \mathcal{N}(\mathbf{x}; \mu_y^{tr}, \Sigma_y^{tr})$ and $p(\mathbf{x}|y, f) = \mathcal{N}(\mathbf{x}; \mu_y, \Sigma_y)$. We also define the class prior probabilities $p(y|f^{tr}) = \omega_y^{tr}$ and $p(y|f) = \omega_y$. Thus f is represented by $(\omega_y, \mu_y, \Sigma_y)$. In this case,

$$\begin{aligned} & D(p(\mathbf{x}, y|f^{tr})||p(\mathbf{x}, y|f)) = \\ & \sum_y \frac{1}{2}\omega_y^{tr}\Bigg(\mathrm{tr}(\Sigma_y^{tr}\Sigma_y^{-1}) + (\mu_y - \mu_y^{tr})^T \Sigma_y^{-1}(\mu_y - \mu_y^{tr}) \\ & + \ln\left(\frac{|\Sigma_y|}{|\Sigma_y^{tr}|}\right) - \frac{d}{2}\Bigg) + \sum_y \omega_y^{tr} \ln \frac{\omega_y^{tr}}{\omega_y} \end{aligned} \quad (8)$$

If $\pi(f)$ is uniform, we see that the prior of the class-conditional parameter (μ_y, Σ_y) becomes a normal-Wishart distribution. This prior has long been used in MAP estimation of Gaussian models due to its tractable mathematical properties as a conjugate prior. Here we have derived it from the perspective of KL-divergence. In fact, we can show that the KL-divergence, and hence the divergence prior, can be conveniently calculated if the class-conditional distribution $p(x|y, f)$ belongs to the exponential family.

In practice, mixture models are more useful for their ability to approximate arbitrary distributions. Mathematically, $p(\mathbf{x}|y, f) = \sum_k c_{y,k} p(\mathbf{x}|y, k, f)$, where $c_{y,k}$, $k = 1..K$, are component responsibilities for class y. There is no close-form solution to the KL-divergence of mixture models. However, we can derive an upper bound on the KL-divergence, and hence a lower bound on the divergence prior, using log sum inequality.

$$\begin{aligned} & D(p(\mathbf{x}, y|f^{tr})||p(\mathbf{x}, y|f)) \leq \\ & \sum_y \omega_y^{tr} \sum_k c_{y,k}^{tr} D(p(\mathbf{x}|y, k, f^{tr})||p(\mathbf{x}|y, m(k), f)) \\ & + \sum_y \omega_y^{tr} \sum_k c_{y,k}^{tr} \ln \frac{c_{y,k}^{tr}}{c_{y,m(k)}} + \sum_y \omega_y^{tr} \ln \frac{\omega_y^{tr}}{\omega_y} \end{aligned} \quad (9)$$

where $(m(1), \ldots, m(K))$ is any permutation of $(1, \ldots, K)$. Since the above inequality holds for an arbitrary alignment of the mixture components, we can

[1] Although improper on unbounded support, a uniform prior does not cause problems in a Bayesian analysis as long as the posterior corresponding to this prior is integrable.

always choose the alignment, based on the similarity between the mixture components, that yields the minimum KL-divergence in order to tighten the bound.

3.2 Discriminative Classifiers

Generative approaches are suboptimal from a classification point of view, as they ask to solve a more difficult density estimation problem. Discriminative approaches, which directly model the conditional relationship of class label given input features, often give better classification performance. One class of discriminative classifiers, including MLPs, SVMs, CRFs and conditional MaxEnt models, can be viewed as hyperplane classifiers in a transformed feature space: $f(\mathbf{x}) = \text{sgn}\left(\mathbf{w}^T\phi(\mathbf{x})+b\right)$, where f is represented by $(\mathbf{w}, b)$ and $\phi(\cdot)$ is a nonlinear transformation. In MLPs, for example, $\phi(\mathbf{x})$ is represented by hidden neurons, and in SVMs $\phi(\mathbf{x})$ is implicitly determined by a reproducing kernel. Here we use $\mathbf{x}$ to represent features for consistency, but $\mathbf{x}$ can be readily replaced by $\phi(\mathbf{x})$ for nonlinear cases. Moreover, a logistic function

$$p(y|\mathbf{x}, f) = \frac{1}{1+e^{-y(\mathbf{w}^T\mathbf{x}+b)}} \tag{10}$$

is often used to model conditional distributions in such classifiers (while a softmax function is often used for the multi-class case). Note that although kernel machines such as SVMs in general do not explicitly model $p(y|\mathbf{x}, f)$, there have been methods to fit SVM outputs to a probability function using a sigmoid function [13]. Here we assume that $p(y|\mathbf{x}, f)$ exists in all cases in the form of Equation (10).

The function space $\mathcal{F}$, therefore, consists of conditional models f, and the classification decision is made via $\text{argmax}_{y\in\mathcal{Y}} \ln p(y|x, f)$. Analogous to our discussion on generative classifiers, if we use $Q(\cdot) = -\ln p(y|\mathbf{x}, f)$, the unadapted model obtained in Equation (1) is the *true* model that describes the conditional distribution in training, *i.e.*, $p(y|\mathbf{x}, f^{tr}) = p^{tr}(y|\mathbf{x})$; and similarly $p(y|\mathbf{x}, f^{ad}) = p^{ad}(y|\mathbf{x})$. Furthermore, the posterior probability can be expressed as

$$p(f|\mathbf{x}, y) = \frac{p(y|\mathbf{x}, f)p(f, \mathbf{x})}{p(\mathbf{x}, y)} = \frac{p(y|\mathbf{x}, f)\pi(f)}{\int p(y|\mathbf{x}, f)\pi(f)\, df} \tag{11}$$

where f and $\mathbf{x}$ are assumed to be independent variables. This factorization leads to a result analogous to Theorem 3.1: assuming that $p^{tr}(\mathbf{x}, y)$ is known, the divergence prior for discriminative classifiers becomes

$$\ln p_{\text{div}}(f) = -D(p(y|\mathbf{x}, f^{tr})||p(y|\mathbf{x}, f)) + \ln\pi(f) + \beta \tag{12}$$

where $\beta > 0$.

The training distribution $p^{tr}(\mathbf{x}, y)$, however, is sometimes unknown to discriminative models (the only information preserved from training is f^{tr} which reflects only the conditional distribution in this case), thereby making $D(p(y|\mathbf{x}, f^{tr})||p(y|\mathbf{x}, f))$ uncomputable. The major goal of this subsection is to derive an upper bound on $D(p(y|\mathbf{x}, f^{tr})||p(y|\mathbf{x}, f))$, and hence a lower bound on the divergence prior, that does not require the knowledge of $p^{tr}(\mathbf{x}, y)$. Then we use this bound instead of $\ln p_{\text{div}}(f)$ in the adaptation objective.

Plugging Equation (10) into Equation (12), we arrive at the following theorem.

Theorem 3.2 *For hyperplane classifiers* $\mathbf{w}^T\mathbf{x}+b$*, the divergence prior in Equation* (12) *satisfies*

$$\ln p_{div}(f) \geq -\alpha\|\mathbf{w}-\mathbf{w}^{tr}\| - |b-b^{tr}| + \ln\pi(f) + \beta \tag{13}$$

where $\alpha = \text{E}_{p^{tr}(x)}[\|\mathbf{x}\|]$.

Proof Using the fact that $|\ln\frac{1+a}{1+b}| < |\ln a - \ln b|$,

$$\begin{aligned} & D(p(y|\mathbf{x}, f^{tr})||p(y|\mathbf{x}, f)) \\ = \; & -\int p^{tr}(\mathbf{x}, y)\ln\frac{1+e^{-y(\mathbf{w}^T\mathbf{x}+b)}}{1+e^{-y(\mathbf{w}^{tr^T}\mathbf{x}+b^{tr})}}\, d\mathbf{x}\, dy \\ \leq \; & \int p^{tr}(\mathbf{x}, y)|y(\mathbf{w}-\mathbf{w}^{tr})^T\mathbf{x} + y(b-b^{tr})|\, d\mathbf{x}\, dy \\ \leq \; & \|\mathbf{w}-\mathbf{w}^{tr}\| \int p^{tr}(\mathbf{x})\|\mathbf{x}\|\, d\mathbf{x} + |b-b^{tr}| \\ = \; & \alpha\|\mathbf{w}-\mathbf{w}^{tr}\| + |b-b^{tr}| \quad \blacksquare \end{aligned} \tag{14}$$

Hence, the accuracy-regularization objective becomes

$$\min_f R_{\text{emp}}(f) + \frac{\lambda_1}{2}\|\mathbf{w}-\mathbf{w}^{tr}\| + \frac{\lambda_2}{2}|b-b^{tr}| - \ln\pi(f) \tag{15}$$

where λ_1 and λ_2 are regularization coefficients.[2] Next, we apply this objective to MLP and SVM adaptation. We focus on these two classifiers because we have not noticed similar adaptation work in the literature (while a similar approach to conditional MaxEnt model adaptation can be found in [10]).

MLP adaptation

Equation (15) can be applied to the adaptation of the hidden-to-out layer of a binary MLP, where we the log loss in optimization and we let λ_1=λ_2=λ. We can extend this to a multi-class, two-layer MLP where we regularize the input-to-hidden weight matrix W_{i2h} (including the offsets) and the hidden-to-output matrix W_{h2o} with separate tradeoff coefficients ν and λ, and we regularize using the squared ℓ_2-norm. Note that we apply such a regularizer to the *input-to-hidden* (i.e.,

[2] The choice of using one or two such coefficients is one of experimental design. We choose two here to derive results later in the paper as will be seen.

first) layer only because we have found it to be practically advantageous (it works well, and it is mathematically easy) — the regularizer on W_{i2h} is not derived from our divergence.

$$\begin{aligned}\min_{W_{h2o},W_{i2h}} \; & R_{\text{emp}}(W_{h2o}, W_{i2h}) \\ +& \left(\frac{\lambda}{2}\|W_{h2o} - W_{h2o}^{tr}\|^2 + \frac{\nu}{2}\|W_{i2h} - W_{i2h}^{tr}\|^2\right)\end{aligned} \qquad (16)$$

where $\|A\|^2 = tr(AA^T)$. In fact, Equation (16) is akin to training an MLP with weight decay if zeros are used as the unadapted weights.

SVM adaptation

Secondly, we apply Equation (15) to SVM adaptation, which utilizes the hinge loss $Q(f(\mathbf{x}_t), y_t) = |1 - y_t(\mathbf{w}^T\phi(\mathbf{x}_t) + b)|_+$ in optimization, and we let $\lambda_2 = 0$. Applying constrained optimization and using the "kernel trick", we obtain the optimal decision function:

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{m} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + \sum_j \alpha_j^{tr} y_j^{tr} k(\mathbf{x}_j^{tr}, \mathbf{x})\right), \qquad (17)$$

where $(\mathbf{x}_j^{tr}, y_j^{tr})$ are support vectors from the unadapted model with coefficients α_j^{tr}, which are *fixed* during adaptation. Optimal α_i are solved in the dual space using the adaptation data only, where the number of new support vectors is controlled by λ_1 in (15). Alternatively, since the support vectors from the unadapted model are available at adaptation time, we can update both α_i and α_j^{tr} in (17) by performing optimization on both the old support vectors and the adaptation data with the constraint $\sum_j \alpha_j^{tr} y_j^{tr} = 0$. These two algorithms will be referred to as "regularized I" and "regularized II" in our experiments in Section 5. Before we evaluate these algorithms, we derive generalization error bounds for adaptation in the PAC-Bayesian framework.

4 PAC-Bayes Error Bound Analysis

A fundamental problem in machine learning is to study the generalization performance of a classifier in terms of an error bound or, equivalently, a sample complexity bound. A PAC-Bayesian approach [14] incorporates domain knowledge in the form of a Bayesian prior and provides a guarantee on generalization error regardless of the truth of the prior. In this work, we are particularly interested in how well an adapted classifier generalizes to unseen data drawn from the target distribution. We derive the error bounds by using our proposed prior in the PAC-Bayesian setting. Specifically, for a countable function space, we apply Occam's Razor bound (Lemma 1 in [14]) which bounds the true error of a single classifier; while for a continuous function space, we apply McAllester's PAC-Bayes bound (Theorem 1 in [14]) which bounds the true stochastic error of a Gibbs classifier.

It is important to note that, although we may apply different loss functions $Q(\cdot)$, usually surrogates (and mostly upper bounds) of the 0-1 loss [15], in actually *training* a classifier, we use the 0-1 loss in *evaluating* error bounds in all cases below. In other words, we have $R(f) = \mathrm{E}_{p^{ad}(\mathbf{x},y)}[I(f(\mathbf{x}) \neq y)]$, and $R_{\text{emp}}(f) = \frac{1}{m}\sum_{i=1}^{m} I(f(\mathbf{x}_i) \neq y_i)$, $(\mathbf{x}_i, y_i) \in \mathcal{D}_m^{ad}$ in the following text.

4.1 An Occam's Razor adaptation bound

The Occam's Razor bound (Lemma 1 in [14]) states that for a countable function space, for any prior distribution $\pi(f)$ and for any f for which $\pi(f) > 0$, the following bound holds with probability of at least $1-\delta$,

$$R(f) \leq R_{\text{emp}}(f) + \sqrt{\frac{-\ln \pi(f) - \ln \delta}{2m}}, \qquad (18)$$

For adaptation, we replace the standard prior $\pi(f)$ in Equation (18) by our proposed divergence prior $p_{\text{div}}(f)$ for a countable function space of generative models. Based on Theorem 3.1 and the Occam's Razor bound, the following bound holds true with probability of at least $1 - \delta$,

$$\begin{aligned}R(f) &\leq R_{\text{emp}}(f) \\ &+\sqrt{\frac{D(p(\mathbf{x},y|f^{tr})||p(\mathbf{x},y|f)) - \ln \pi(f) - \beta - \ln \delta}{2m}}\end{aligned} \qquad (19)$$

This result has important implications: for the set of classifiers $\mathcal{G} = \{f \in \mathcal{F} : D((p(\mathbf{x},y|f^{tr})||p(\mathbf{x},y|f)) < \beta\}$, their error bounds in Equation (19) which use the divergence prior are tighter than those in Equation (18) which use the standard prior. Since $\beta > 0$, $\mathcal{G}$ is always nonempty. For classifiers in the complementary set $\bar{\mathcal{G}}$, however, we reach the opposite argument. An important question to ask is: in which set does our estimated classifier belongs? We are particularly interested in f^{ad}, *i.e.*, the optimal classifier w.r.t. the target distribution. If $D(p^{tr}||p^{ad}) < \beta$, we have $f^{ad} \in \mathcal{G}$ and we achieve better generalization performance at f^{ad} by using the divergence prior. Recall that β normalizes $p_{\div}(f)$ to unity. This constant can be analytically calculated for certain models (*e.g.* Gaussian models), while approximations are needed for general cases. Additionally, we can derive a similar bound for discriminative classifiers, where the divergence in Equation (19) is between conditional distributions instead of joint distributions.

4.2 Adaptation bounds for Gibbs classifiers

McAllester's PAC-Bayesian bound for Gibbs classifiers is applicable to both countable and uncountable function spaces. A Gibbs classifier is a stochastic classifier drawn from a posterior distribution $q(f)$. Consequently the true and empirical risks also become stochastic in the form of $\mathrm{E}_{f\sim q}[R(f)]$ and $\mathrm{E}_{f\sim q}[R_{\mathrm{emp}}(f)]$. McAllester's PAC-Bayesian bound [14] states that for any prior distribution $\pi(f)$ and any posterior distribution $q(f)$, the following holds with probability $1-\delta$:

$$\begin{aligned} &\mathrm{E}_{f\sim q}[R(f)] \le \mathrm{E}_{f\sim q}[R_{\mathrm{emp}}(f)] \\ &+\sqrt{\frac{D(q(f)||\pi(f)) - \ln\delta + \ln m + 2}{2m-1}} \end{aligned} \tag{20}$$

The choice of a prior distribution $\pi(f)$ is again critical in order to achieve a small error bound. Intuitively we should choose a distribution $\pi(f)$ such that $\mathrm{E}_{f\sim\pi}[R_{\mathrm{emp}}(f)]$ is small. As a ramification of this theorem, *PAC-Bayesian margin bounds* have been developed which provide theoretical foundations for SVMs [16]. The key idea involves choosing a prior $\pi(f)$ and a posterior $q(f)$ such that, in addition to our intuition above, it is easy to compute $D(q(f)||\pi(f))$ and $\mathrm{E}_{f\sim q}[R_{\mathrm{emp}}(f)]$. Usually $q(f)$ is chosen to be in the same family as $\pi(f)$.

In this section, we obtain the error bounds for adaptation in a similar fashion as [16], but with simpler derivations. Since the derivation requires specification of a classifier, we first investigate generative Gaussian models where only Gaussian means are adapted. We further assume equal class prior probabilities $\omega_+{=}\omega_-{=}1/2$, equal covariance matrices $\Sigma_+{=}\Sigma_-{=}\Sigma^{tr}$, and opposite means $\mu_+{=}{-}\mu_-{=}\mu$, thereby leading to a linear decision boundary. In such a case, f is represented by μ. We make such assumptions only to simplify the calculation of the stochastic error in this work, while similar bounds can be derived for more general cases.

McAllester's PAC-Bayesian bound allows to choose any prior distribution and posterior distribution. Here we use $p_{\mathrm{div}}(f)$ as the prior distribution, where we assume a uniform $\pi(f)$ and renormalize $p_{\mathrm{div}}(f)$ accordingly. The resulting prior is a Gaussian centered at the unadapted means $[\mu^{tr}, -\mu^{tr}]^T$. Furthermore, we define the posterior distribution $q(f)$ to be a Gaussian centered at some means $[\mu', -\mu']^T$. Mathematically, $p_{\mathrm{div}}(f) = \mathcal{N}(\mu;\mu^{tr},\Sigma^{tr})$, and $q(f) = \mathcal{N}(\mu;\mu',\Sigma^{tr})$. It is easy to compute the KL-divergence

$$D(q(f)||p_{\mathrm{div}}(f)) = \frac{1}{2}(\mu'-\mu^{tr})^T{\Sigma^{tr}}^{-1}(\mu'-\mu^{tr})$$

which gives the second term in Equation (20). On the other hand, to calculate $\mathrm{E}_{f\sim q}[R_{\mathrm{emp}}(f)])$, we first inspect the decision function regarding sample $(\mathbf{x}_i, y_i)$, *i.e.*, $\mathrm{sgn}\,(y_i(\mu_+ - \mu_-)^T{\Sigma^{tr}}^{-1}\mathbf{x}_i) = \mathrm{sgn}\,(y_i\mathbf{x}_i^T{\Sigma^{tr}}^{-1}\mu)$. Since $q(\mu) = \mathcal{N}(\mu;\mu',\Sigma^{tr})$, $y\mathbf{x}^T{\Sigma^{tr}}^{-1}\cdot\mu$ is a univariate Gaussian with mean $y\mathbf{x}^T{\Sigma^{tr}}^{-1}\mu'$ and variance $(y\mathbf{x}^T{\Sigma^{tr}}^{-1})\Sigma^{tr}(y\mathbf{x}^T{\Sigma^{tr}}^{-1})^T = \mathbf{x}^T{\Sigma^{tr}}^{-1}\mathbf{x}$. The stochastic empirical risk hence becomes

$$\begin{aligned} &\mathrm{E}_{f\sim q}[R_{\mathrm{emp}}(f)] \\ =\ & \frac{1}{m}\sum_{i=1}^{m}\mathrm{E}_{\mu\sim\mathcal{N}(\mu;\mu',\Sigma^{tr})}[\mathrm{I}(y_i\mathbf{x}_i^T{\Sigma^{tr}}^{-1}\mu<0)] \\ =\ & \frac{1}{m}\sum_{i=1}^{m}\mathrm{E}_{t\sim\mathcal{N}(t;y_i\mathbf{x}_i^T{\Sigma^{tr}}^{-1}\mu',\mathbf{x}_i^T{\Sigma^{tr}}^{-1}\mathbf{x}_i)}[\mathrm{I}(t<0)] \\ =\ & \frac{1}{m}\sum_{i=1}^{m}F\Big(\frac{y_i\mathbf{x}_i^T{\Sigma^{tr}}^{-1}\mu'}{\sqrt{\mathbf{x}_i^T{\Sigma^{tr}}^{-1}\mathbf{x}_i}}\Big) \end{aligned} \tag{21}$$

where $F(t) = \int_t^\infty \frac{1}{\sqrt{2\pi}}e^{-\frac{s^2}{2}}\,ds$, and $(\mathbf{x}_i, y_i)\in\mathcal{D}_m^{ad}$.

In conclusion, to adapt Gaussian means in the above setting, for any choice of μ', the following bound holds true with probability at least $1-\delta$.

$$\begin{aligned} &\mathrm{E}_{f\sim q}[R(f)] \le \frac{1}{m}\sum_{i=1}^{m}F\Big(\frac{y_i\mathbf{x}_i^T{\Sigma^{tr}}^{-1}\mu'}{(\mathbf{x}_i^T{\Sigma^{tr}}^{-1}\mathbf{x}_i)^{1/2}}\Big) \\ &+\sqrt{\frac{\frac{1}{2}(\mu'-\mu^{tr})^T{\Sigma^{tr}}^{-1}(\mu'-\mu^{tr}) - \ln\delta + \ln m + 2}{2m-1}} \end{aligned} \tag{22}$$

Lastly, we derive an adaptation error bound for hyperplane classifiers $\mathbf{w}^T\mathbf{x}+b$, which is an important representative for discriminative classifiers (see Section 3.2). In this case, $f = (\mathbf{w}, b)$ where we assume that $\mathbf{w}$ and b are independent variables. We use a Gaussian prior $p(f)$ centered at $(\mathbf{w}^{tr}, b^{tr})$. Note that the choice of this prior relates to previous work on margin bounds; [16] used a Gaussian prior centered at zero, and [17] estimated Gaussian priors from previous training subsets for incremental learning. The key difference is that we choose a Gaussian centered at the unadapted parameters. Furthermore, we choose a posterior $q(f)$ in the same family. Mathematically, $p(\mathbf{w}, b) = \mathcal{N}(\mathbf{w};\mathbf{w}^{tr}, I)\cdot\mathcal{N}(b;b^{tr},1)$, and $q(\mathbf{w}, b) = \mathcal{N}(\mathbf{w};\mathbf{w}', I)\cdot\mathcal{N}(b;b',1)$.

Following the derivation in our previous example, we arrive at the following result: for any choice of $(\mathbf{w}', b')$, the following bound holds true with probability at least $1-\delta$.

$$\begin{aligned} &\mathrm{E}_{q(f)}[R_{p(\mathbf{x},y)}(f)] \le \frac{1}{m}\sum_{i=1}^{m}F\Big(\frac{y_i(\mathbf{x}_i^T\mathbf{w}'+b')}{\sqrt{\|\mathbf{x}_i\|^2+1}}\Big) \\ &+\sqrt{\frac{\frac{\|\mathbf{w}'-\mathbf{w}^{tr}\|^2+|b'-b^{tr}|^2}{2} - \ln\delta + \ln m + 2}{2m-1}} \end{aligned} \tag{23}$$

where $F(t) = \int_t^\infty \frac{1}{\sqrt{2\pi}}e^{-\frac{s^2}{2}}\,ds$, and $(\mathbf{x}_i, y_i)\in\mathcal{D}_m^{ad}$.

# adapt. samples	0.8K	1.6K	2.4K
Unadapted	38.21	38.21	38.21
Retrained	24.70	**18.94**	**14.00**
Boosted	29.66	26.54	28.85
Regularized I	**23.28**	**19.01**	15.00
Regularized II	28.55	25.38	20.36

Table 1: Adaptation of a SVM vowel classifier; The highlighted entries include the best error rate and those not significantly different at the $p < 0.001$ level using a *difference of proportions* significant test (the same below).

# adapt. samples	0.8K	1.6K	2.4K
Unadapted	32.03	32.03	32.03
Retrained zero	14.21	11.20	9.09
Retrained W^{tr}	12.15	9.64	7.88
Retrained last	15.45	13.32	11.40
Regularized	**11.56**	**8.16**	**7.30**

Table 2: Adaptation of an MLP vowel classifier; The highlighted entries include the best error rate and those not significantly different at th e $p < 0.001$ level.

5 Experiments

This section evaluates the adaptation algorithms derived from the divergence prior, *i.e.*, Equation (7) and Equation (15). We present classification experiments using adapted SVMs and MLPs, as well as a simulation of empirical error bounds on Gaussian models.

5.1 Vowel classification

Our first task involves a dataset of 8 vowel classes articulated in different manners (by varying pitch, volume and duration) [18]. We used 182 cepstral features (from 7 frames of 16kHz waveforms). Our target was to perform speaker adaptation and evaluate frame-level vowel classification error rates. The training/testing set had 420K/200K samples. But in training an unadapted SVM, we used only 80K samples randomly selected from the training set for computational tractability. For each speaker in the testing set, we performed 6-fold adaptation-evaluation, where each adaptation/evaluation set had 2.4K/12K samples. We repeated the same experiments for 10 test speakers, and computed the average error rates.

We first adapted an SVM classifier with fixed Gaussian kernels, and compared the following SVM adaptation algorithms: (1) “unadapted”; (2) “retrained” using only adaptation data; (3) “boosted” which combines the old support vectors with misclassified adaptation data [7]; (4) “regularized I” which follows Equation (17); and (5) “regularized II” which updates α_j^{tr} as well. In this task, where it was easy to obtain adaptation data (100 samples correspond only to a 1-second utterance), our adaptation data sizes were relatively large, and the “retrained” classifier in general works well, as shown in Table 1. “Regularized I”, however, had a statistically significant gain over “retrained” when the adaptation data size was restricted.

# adaptation samples	90	180
Unadapted	12.5	12.5
Retrained	30.1	18.9
Boosted	**12.1**	**10.7**
Regularized I	14.8	13.4
Regularized II	**11.0**	**10.4**

Table 3: adaptation of a SVM object classifier; The highlighted entries include the best error rate and those not significantly different at the $p < 0.001$ level.

Secondly, we implemented a two-layer MLP with 50 hidden neurons, and compared MLP adaptation algorithms including: (1) ”unadapted”; (2) ”retrained zero” which learns a new MLP from randomly initialized weights and regularizes with weight decay; (3) ”retrained W^{tr}” which starts from the unadapted weights; (4) ”retrained last” which fixes the first layer and retrains the second layer (akin to [1]); and (5) ”regularized” which starts from the unadapted weights and regularizes as in Equation (16). As shown in Table 2, our proposed adaptation algorithm gave superior performance in all cases.

5.2 Object recognition

Our second task was on an object recognition dataset comprised of 5 generic classes (animals, human figures, airplanes, trucks and cars), each with 10 objects [19]. The images of each object were captured from 18 angles and under 6 lighting conditions (a subset of [19]). The training and testing set each had 2700 samples. We conducted similar n-fold adaptation-evaluation experiments, where each adaptation set had either 90 or 180 samples for each lighting condition, and the remaining 450 or 360 samples, under the same lighting condition, were used for evaluation.

We compared the SVM adaptation algorithms listed in the vowel classification experiments. As shown in Table 3, on this data set where the adaptation sample size was extremely small, both ”boosted” and “regularized II” worked remarkably well as they incorporate more information from the training data. We also tried using an MLP classifier, and the regularized adaptation had only a trivial improvement over the unadapted classifier.

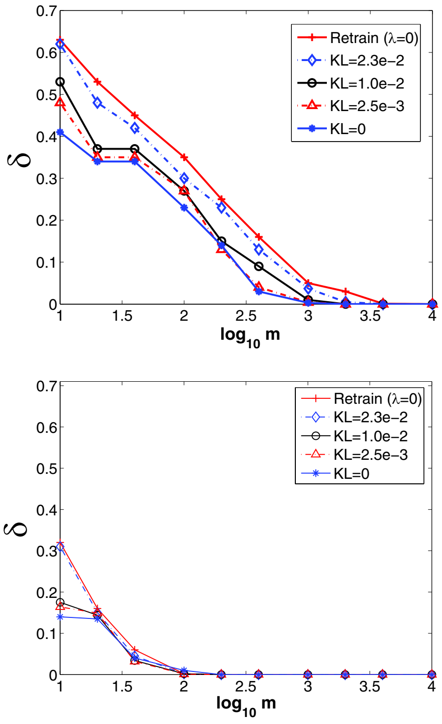

Figure 1: Empirical error bound study: δ vs. $\log m$ for $\epsilon = 0.02$ (upper figure) and $\epsilon = 0.1$ (lower figure)

5.3 Simulations of empirical error bounds

We simulated *empirical* adaptation error bounds for a Gaussian model classifier. Given an unadapted model, and an adaptation set with m samples randomly generated from a target distribution, we learned an adapted classifier using our regularized adaptation objective in Equation (7), where the log joint likelihood loss and a uniform $\pi(f)$ were used, and λ's for different m's were discovered using a development set with 5K samples. We computed the empirical error R_{emp} on the adaptation set, and estimated the true error R on a testing set with 10K samples, both corresponding to the 0-1 loss. We then estimated $\delta = \text{E}[\text{I}(R > R_{\text{emp}} + \epsilon)]$ using 1K separate runs (10K samples each). Figure 1 plots δ vs. $\log m$ for $\epsilon = 0.02$ and $\epsilon = 0.1$ with different $D(p^{tr}||p^{ad})$ and m on simulated 2D-Gaussians. The $\lambda = 0$ line corresponds to retraining from scratch (no adaptation), and also to large KL-divergences, as then optimal λ discovery produces $\lambda = 0$. Although we do not yet have a theoretical result to bound $R(f)$ by $R_{\text{emp}}(f)$ in the Gaussian model case, as the function space is continuous (Section 4.1), we have empirically shown that fewer samples were needed for smaller KL values to achieve the same confidence δ.

The authors would like to thank Patrick Haffner for useful comments.

References

[1] J. Baxter, "Learning internal representations," in *COLT*, 1995.

[2] R. Caruana, "Multitask learning," *Machine Learning Journal*, vol. 28, 1997.

[3] S. Thrun and L.Y. Pratt, *Learning To Learn*, Kluwer Academic Publishers, Boston, MA, 1998.

[4] S. Ben-David and R. Schuller, "Exploiting task relatedness for multiple task learning," in *COLT*, 2003.

[5] J. L.Gauvain and C. H. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains," *IEEE Trans. on Speech and Audio Processing*, vol. 2, 1994.

[6] C. Leggetter and P. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer, Speech and Language*, vol. 9, 1995.

[7] N. Mati'c, I. Guyon, J. Denker, and V. Vapnik, "Writer adaptation for on-line handwritten character recognition," in *Proc. Intl. Conf. on Document Analysis and Recognition*, 1993.

[8] P. Wu and T. G. Dietterich, "Improving svm accuracy by training on auxiliary data sources," in *ICML*, 2004.

[9] X. Li and J. Bilmes, "Regularized adaptation of discriminative classifiers," in *ICASSP*, 2006.

[10] C. Chelba and A. Acero, "Adaptation of maximum entropy capitalizer: Little data can help a lot," in *Empirical Methods in Natural Language Processing*, July 2004.

[11] J. Baxter, "A bayesian/information theoretic model of learning to learn via multiple task sampling," *Machine Learning*, 1997.

[12] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, 1995.

[13] J. Platt, "Probabilistic outputs for support vector machines and comparison to regularized likelihood methods," in *Advances in Large Margin Classifiers*, A.J. Smola et. al., Ed., 2000, pp. 61–74.

[14] D. A. McAllester, "PAC-Bayesian stochastic model selection," *Machine learning journal*, 2001.

[15] P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe, "Convexity, classification, and risk bounds," *Journal of the American Statistical Association, 101*, 2006.

[16] J. Langford and J. Shawe-Taylor, "PAC-Bayes and margins," in *NIPS*, 2002.

[17] E. Parrado-Hernandez A. Ambroladze and J. Shawe-Taylor, "Learning the prior for the PAC-Bayes bound," Tech. Rep., Southampton, UK, 2004.

[18] K. Kilanski, J. Malkin, X. Li, R. Wright, and J. Bilmes, "The Vocal Joystick data collection effort and vowel corpus," in *Interspeech*, 2006.

[19] Y. LeCun, F. J. Huang, and L. Bottou, "Learning methods for generic object recognition with invariance to pose and lighting," in *CVPR*, 2004.