

# Generalization Error Bounds Using Unlabeled Data

Matti Kääriäinen

Department of Computer Science,  
P.O. Box 68, FIN-00014 University of Helsinki, Finland  
`matti.kaariainen@cs.helsinki.fi`

**Abstract.** We present two new methods for obtaining generalization error bounds in a semi-supervised setting. Both methods are based on approximating the disagreement probability of pairs of classifiers using unlabeled data. The first method works in the realizable case. It suggests how the ERM principle can be refined using unlabeled data and has provable optimality guarantees when the number of unlabeled examples is large. Furthermore, the technique extends easily to cover active learning. A downside is that the method is of little use in practice due to its limitation to the realizable case.

The idea in our second method is to use unlabeled data to transform bounds for randomized classifiers into bounds for simpler deterministic classifiers. As a concrete example of how the general method works in practice, we apply it to a bound based on cross-validation. The result is a semi-supervised bound for classifiers learned based on all the labeled data. The bound is easy to implement and apply and should be tight whenever cross-validation makes sense. Applying the bound to SVMs on the MNIST benchmark data set gives results that suggest that the bound may be tight enough to be useful in practice.

## 1 Introduction

We study an extension of the (*supervised*) *statistical learning model* to a model for semi-supervised learning. In the semi-supervised model, the learner gets a *labeled learning sample*  $(X_1, Y_1), \dots, (X_n, Y_n)$  and an *unlabeled learning sample*  $(X_{n+1}, \dots, X_{n+m})$ . Here, the labeled examples  $(X_i, Y_i) \in \mathcal{X} \times \mathcal{Y}$ ,  $1 \leq i \leq n$ , are independent copies of a random element  $(X, Y)$  having distribution  $P$  on  $\mathcal{X} \times \mathcal{Y}$ , and the unlabeled examples  $X_{n+j} \in \mathcal{X}$ ,  $1 \leq j \leq m$ , are independent copies of  $X$ , whose distribution (the marginal distribution of  $P$  on  $X$ ) we denote by  $P_X$ . Based on the (labeled and unlabeled) learning samples, the learner is supposed to pick a classifier  $f: \mathcal{X} \rightarrow \mathcal{Y}$  with small *generalization error*  $\epsilon(f) = P(f(X) \neq Y)$ . In addition to the classifier, we are interested in a *generalization error bound* for it, that is, a random variable that upper bounds  $\epsilon(f)$  for the learned classifier  $f$  with at least probability  $1 - \delta$ . The setting extends easily to learning *randomized classifiers* which will be defined formally in Section 2.

The usual motivation for studying semi-supervised learning is that in practice getting unlabeled data is often considerably easier or cheaper than getting labeled data. We are tempted to go even further and claim that in cases where the model of statistical learning theory makes sense, unlabeled data should be almost free. The reason is that if examples distributed according to  $P_X$  are hard to get, stating the goal of learning in terms of generalization performance — the expected loss on such examples — is peculiar. In such cases, it would probably be better to resort to transduction (in case the unlabeled sample to be labeled is known at the time of learning) or to state the goal of learning in terms other than generalization error. The semi-supervised model thus seems to be applicable in most of the cases in which the model of statistical learning theory is sensible. An exception to this rule is the case in which unlabeled data will be available but only after learning has taken place.

If we take the sample of unlabeled data for granted, the next question is whether and how access to it can help in learning and/or generalization error analysis. These questions have been subject to intensive research that has produced many semi-supervised learning algorithms that can be used in practice. The theoretical aspects of semi-supervised learning have received less attention, although some interesting results have been published recently [1, 2]. The value of unlabeled data to learning has been studied in restricted settings [3, 4], but to our knowledge the general question of whether unlabeled data provably helps in classifier learning has not been answered.

We prove that unlabeled data is useful in the *realizable case*, that is, when the learner is given access to a set  $F$  of classifiers that contains a *target function*  $f_0$  for which  $Y = f_0(X)$  (always or at least with probability 1). More specifically, we show that we can improve on the best results obtainable for *empirical risk minimization (ERM)* [5] provided we have access to a sufficiently large sample of unlabeled examples. In our second method for obtaining semi-supervised generalization error bounds we drop the assumption of the existence of a target function. The method is based on derandomizing generalization error bounds for randomized classifiers using unlabeled data. As an example of a concrete bound that can be proved using the proposed method, we transform the cross-validation estimate into a true generalization error bound for the hypothesis learned based on all the labeled data. Our empirical experiments indicate that the resulting bound applied to SVMs on the MNIST benchmark data set gives bounds comparable to cross-validation estimates. Thus, even though our second method lacks theoretical a priori optimality guarantees, it seems to provide bounds that are extremely tight in practice.

Our bounds for both the realizable and the general case are based on using the *disagreement probability*  $d(f, g) = \mathbb{P}(f \neq g) = \mathbb{P}(f(X) \neq g(X))$  as a metric in the space of randomized classifiers. Variants of  $d$  have been used earlier as a basis for model selection criteria [6, 7], in providing lower bounds and estimates of the variance of the error of a hypothesis produced by a learning algorithm in a co-validation setting [1], and as an example of a distance measure that can be used in the learning by distances model [8]. To our knowledge, using  $d$  in

proving generalization error bounds is original to our work. The disagreement probability  $d$  is very natural in this context, since the generalization error of a classifier is its probability of disagreeing with the target. The reason  $d$  fits the semi-supervised setting particularly well is that it can be approximated using  $\hat{d}$  given by  $\hat{d}(f, g) = \frac{1}{m} \sum_{j=1}^m \mathbb{1}[f(X_{n+j}) \neq g(X_{n+j})]$ , where the notation  $\mathbb{1}[\phi]$  means the function that has value 1 if  $\phi$  is true and 0 otherwise. Note that  $m\hat{d}(f, g)$  is the number of times  $f$  and  $g$  disagree on the unlabeled sample, so its distribution is binomial with parameters  $m$  and  $d(f, g)$ . Thus, one can derive confidence intervals for  $d(f, g)$  given  $\hat{d}(f, g)$  using the familiar techniques for binomial distributions.

## 2 Randomized Classifiers

In addition to standard deterministic classifiers, we work with *randomized classifiers*, also referred to as *Gibbs classifiers* in the literature. A randomized classifier  $f$  is simply a  $\mathcal{Y}$ -valued random variable that may depend on  $X$  but is independent of other randomized classifiers given  $X$ . In particular, the target  $Y$  is viewed as a randomized classifier. To classify an example  $x \in \mathcal{X}$ , a randomized classifier  $f$  chooses a label  $f(x) \in \mathcal{Y}$  from the conditional distribution of  $f$  given  $X = x$ . A new copy of  $f$  is used each time it is applied.

In practice, a randomized classifier  $f$  is usually specified by a set of classifiers  $\{f_\theta: \mathcal{X} \rightarrow \mathcal{Y}\}$ , where the parameter  $\theta$  is a realization of a random variable  $\Theta = \Theta_f$  that specifies the underlying classifier to use. It is assumed that the parameters  $\Theta_f$  are independent of each other and everything else. The randomized classifier corresponding to a deterministic classifier  $f: \mathcal{X} \rightarrow \mathcal{Y}$  is simply  $f(X)$ . It is admittedly a bit unnatural to incorporate the distribution of  $X$  in the definition of a randomized classifier, but this choice will be technically convenient in the following. The definition of generalization error is extended to randomized classifiers  $f$  by setting  $\epsilon(f) = \mathbb{P}(f \neq Y) = \mathbb{P}(f_\Theta(X) \neq Y)$ .

The definition of disagreement probability  $d(f, g) = \mathbb{P}(f \neq g)$  and its empirical approximation  $\hat{d}(f, g)$  extend automatically to randomized classifiers. The fact that  $d$  really is a (pseudo-)metric on the space of randomized classifiers can be easily verified (also for loss functions other than the 0-1 loss). A key property of  $d$  we will take advantage of is that  $\epsilon(f) = d(f, Y)$  for all randomized classifiers  $f$ , a fact first noted by Schuurmans and Southey for the special case of deterministic classifiers [6]. Thus, we can embed all the classifiers and the target into a metric space, state the goal of learning in terms of this metric, and use the metric structure of the space both in the learning process and its analysis.

Note that the distance  $\hat{d}(f, g)$  between randomized classifiers  $f$  and  $g$  depends on the unlabeled data points  $X_{n+j}$ ,  $1 \leq j \leq m$ , and the random classifications of  $f$  and  $g$  only, so it can be computed without knowing the labels for the unlabeled points. Our strategy will be to use  $\hat{d}$  to relate the generalization error of a learned classifier to that of a (randomized) classifier for which it is either known or can be tightly upper bounded. We will show how to do this in the realizable and in the general case in Sections 3 and 4, respectively.

### 3 The Realizable Case

In this section we present our bounds for the realizable case, discuss their properties, and outline extensions to active learning.

#### 3.1 General Bound

Our bound for the realizable case is based on relating the learned classifier to (other) *consistent* classifiers — classifiers  $f$  for which the *empirical error*  $\hat{\epsilon}(f) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}[f(X_i) \neq Y_i]$  is zero. The idea is that even though the target function is unknown, we know that it is among the consistent classifiers and that its generalization error is zero. Thus, if we can show that the learned classifier is not too far from any of the consistent classifiers, then it has to be close to the target function, too. Because the metric  $d$  we use for measuring distances is the disagreement probability, this implies that the generalization error of the learned classifier is bound to be close to zero as well. We will next show how to make this idea precise.

Let  $F_0 = \{f \in F \mid \hat{\epsilon}(f) = 0\}$  be the set of consistent classifiers, also known as the *version space*. Here,  $F$  is the given class of classifiers that is known to contain the target  $f_0$  by assumption, so  $f_0$  is always in the version space. Let  $f$  be any classifier. The generalization error of  $f$  can be written as

$$\epsilon(f) = d(f, Y) = d(f, f_0).$$

Thus, the generalization error of  $f$  is simply its  $d$ -distance to  $f_0$ .

The only thing we know about  $f_0$  is that it is by assumption consistent with the labeled data and in  $F_0$ , so the best imaginable upper bound for  $d(f, f_0)$  based on the knowledge at hand is  $\sup \{d(f, g) \mid g \in F_0\}$ . But we do not know  $d$ , so we have to replace it by the empirical approximation  $\hat{d}$  to get

$$\epsilon(f) \leq \sup_{g \in F_0} \left[ \hat{d}(f, g) + (d - \hat{d})(f, g) \right] \leq \sup_{g \in F_0} \hat{d}(f, g) + \sup_{g \in F_0} (d - \hat{d})(f, g).$$

Here, only the term  $\sup_{g \in F_0} (d - \hat{d})(f, g)$  depends directly on the unknown distribution  $P$  (through  $d$ ). This far nothing has been assumed about  $f$ , but in order to bound the error introduced by replacing  $d$  by  $\hat{d}$ , one has to introduce some restrictions. A natural choice (suggested by the ERM principle) is to restrict  $f$  to be a consistent classifier chosen from  $F$ , that is, to assume  $f \in F_0$ . This gives

$$\epsilon(f) \leq \sup_{g \in F_0} \hat{d}(f, g) + \sup_{g', g \in F_0} (d - \hat{d})(g', g).$$

Optimizing this bound over  $f$  suggests choosing the  $\hat{f} \in F_0$  whose (empirical) distance to the farthest point in  $F_0$  is minimal (for simplicity we assume such a minimizer exists). We will call this  $\hat{f}$  the *empirical center of the version space* for obvious reasons.

Putting the above reasoning together, we get the following bound for the empirical center. A similar bound without the infimum holds for any  $f \in F_0$  and can be useful, e.g., if finding the empirical center is computationally hard.

**Theorem 1.** *Let  $\hat{f}$  be the empirical center of  $F_0$ . It is always true that*

$$\epsilon(\hat{f}) \leq \inf_{f \in F_0} \sup_{g \in F_0} \hat{d}(f, g) + \sup_{g', g \in F_0} (d - \hat{d})(g', g).$$

This bound still depends on the unknown distribution  $P_X$  through the term  $\sup \{(d - \hat{d})(g', g) \mid g', g \in F_0\}$ . We will next show how to get rid of this dependency by using Rademacher penalization (other uniform convergence techniques familiar from generalization error analysis could have been used as well). If unlabeled data is abundant, one can also take a course similar to the hold-out bounds and use an independent sample of unlabeled data to test how close  $d$  and  $\hat{d}$  really are to each other on  $F_0$ .

### 3.2 Concrete Bound Based on Rademacher Penalization

The idea here is to apply standard Rademacher penalization bounds to the class  $\{x \mapsto \llbracket g'(x) \neq g(x) \rrbracket \mid g', g \in F_0\}$  and the sample of unlabeled data to show that the empirical expectations of these indicators (in our notation  $\hat{d}$ ) are with high probability close to their true expectations (in our notation  $d$ ). This yields an upper bound for  $\sup \{(d - \hat{d})(g', g) \mid g', g \in F_0\}$ , the quantity we are interested in.

Following [9], we define the Rademacher penalty  $R_m(H)$  of a class  $H$  of functions from  $\mathcal{X}$  to  $\{0, 1\}$  as follows:

$$R_m(H) = \sup_{h \in H} \left| \frac{1}{m} \sum_{j=1}^m \sigma_j (1 - 2h(X_{n+j})) \right|.$$

Here, the random elements  $X_{n+j}$  are independent copies of  $X$  and  $\sigma_1, \dots, \sigma_m$  is a sequence of symmetrical  $\{\pm 1\}$ -valued random signs independent of each other and everything else. With this definition, we have the following:

**Theorem 2 ([9]).** *Let  $H$  be any set of functions from  $\mathcal{X}$  to  $\{0, 1\}$ . With probability at least  $1 - \delta$  (over the choice of the random signs and the  $X_j$ s), it is true that*

$$\sup_{h \in H} \left| \frac{1}{m} \sum_{j=1}^m h(X_{n+j}) - \mathbb{E}h(X) \right| \leq R_m(H) + \frac{3}{\sqrt{2}} \sqrt{\frac{\ln 2/\delta}{m}}.$$

To use this bound,  $H$  has to be independent of  $X_{n+1}, \dots, X_{n+m}$ . In our case  $H$  depends on the labeled sample through  $F_0$ , but is independent of the unlabeled sample. Hence, the previous theorem can be applied, which together with Theorem 1 gives the following.

**Theorem 3.** *Let  $\hat{f}$  be the empirical center of  $F_0$ . For all labeled learning samples, it is true with probability at least  $1 - \delta$  (over the choice of the unlabeled learning sample and the Rademacher signs) that*

$$\epsilon(\hat{f}) \leq \inf_{f \in F_0} \sup_{g \in F_0} \hat{d}(f, g) + R_m(\{\llbracket g' \neq g \rrbracket \mid g', g \in F_0\}) + \frac{3}{\sqrt{2}} \sqrt{\frac{\ln(2/\delta)}{m}}.$$

This bound depends on the observed data only. Thus, the bound can be evaluated in practice if the computational problems related to evaluating the Rademacher penalty term can be overcome. If not or if one is only interested in how the bound behaves in the worst case as a function of  $m$ , one can resort to further upper bounds based on (upper bounds) for the VC dimension of  $\{\llbracket g' \neq g \rrbracket \mid g', g \in F_0\}$  to get the following corollary [9, 10].

**Corollary 1.** *Let  $\hat{f}$  be the empirical center of  $F_0$  and let  $D$  be an upper bound for the (data-dependent) VC dimension of  $\{\llbracket g' \neq g \rrbracket \mid g', g \in F_0\}$ . Then with probability at least  $1 - \delta$  (over the choice of the unlabeled sample) we have*

$$\epsilon(\hat{f}) \leq \inf_{f \in F_0} \sup_{g \in F_0} \hat{d}(f, g) + \sqrt{2} \sqrt{\frac{D(\ln(m/D) + 1) + \ln(2/\delta)}{m}} + \frac{3}{\sqrt{2}} \sqrt{\frac{\ln(4/\delta)}{m}}.$$

When the corollary is applicable, it implies that the error introduced by approximating  $d$  by  $\hat{d}$  vanishes as the size of the unlabeled sample increases. Even though tighter bounds could be desired in practical applications, this is all we need to know in the discussion that follows.

### 3.3 Properties of the Bound for the Realizable Case

In this section we analyze how our bounds for the realizable case behave as a function of  $n$  and  $m$ . The general intuition is as follows. The term  $\sup_{g \in F_0} \hat{d}(f, g)$  measures the amount of uncertainty about the target that remains after seeing the labeled sample. The remaining terms measure the inaccuracy introduced by approximating  $d$  by  $\hat{d}$ , that is, the remaining uncertainty about  $P_X$ . These two kinds of uncertainty depend on each other: The less labeled data, the larger the version space  $F_0$ , and thus the more complex the task of approximating  $d$  on  $F_0$ .

Let us first see what happens in the limit  $m \rightarrow \infty$ . In case the loss class  $\{\llbracket g' \neq g \rrbracket \mid g', g \in F_0\}$  has finite VC dimension, we know by Corollary 1 that  $(d - \hat{d})$  goes uniformly to zero on  $F_0$ . In this case, for large  $m$ , the bound reduces essentially to

$$\inf_{f \in F_0} \sup_{g \in F_0} \hat{d}(f, g) = \inf_{f \in F_0} \sup_{g \in F_0} d(f, g).$$

The best possible bound for ERM would be  $\sup \{d(f, g) \mid f, g \in F_0\}$ : Any smaller bound would be violated by some combinations of a consistent hypothesis  $f$  and a target  $g$ . As ERM views all  $f \in F_0$  equivalently and the target may be any of the consistent functions, such a worst case situation can be realized.

In geometric terms, the lower bound for bounds for ERM is the true diameter of the version space, whereas our upper bound for the empirical center is its true radius. This simple observation immediately yields the following:

**Theorem 4.** *Suppose the Rademacher penalty term in Theorem 3 converges to 0 as  $m \rightarrow \infty$ . Then, for sufficiently large  $m$ , the bound for the empirical center is at least as good as the best possible bound for ERM and cannot be improved without additional assumptions or labeled data. The bound of Theorem 3 improves upon the best possible bound for ERM by a factor of 2 if the radius of  $F_0$  is only half its diameter, but in case the radius equals the diameter the bounds may be equal.*

The other limiting case is when no uncertainty about the labeling remains, whence  $F_0$  reduces to  $\{f_0\}$ . In this case the bound of Theorem 1 reduces to zero, irrespectively of the unlabeled learning sample. This limiting case is probably not too interesting, but it is still nice that the bound gives the correct answer.

Of course, the most interesting cases are the ones in between the extremes outlined above. Here, the exact values of  $\sup\{(d - \hat{d})(g', g) \mid g', g \in F_0\}$  and its upper bounds become important. The trade-off is that the more complex  $F_0$  is, the more unlabeled data is needed to reveal its structure, that is, to make  $\sup\{(d - \hat{d})(g', g) \mid g', g \in F_0\}$  small. If  $F_0$  is simple enough (e.g., the related class of pairs of classifiers has finite VC dimension), we know that this supremum vanishes as  $m$  increases with a speed depending on the complexity of  $F_0$ . In practice it is impossible to get or use arbitrarily large samples of unlabeled data, which makes the non-asymptotic behavior of the penalty terms important. The quest for tightest possible finite sample bounds on the deviation between  $\hat{d}$  and  $d$  resembles a lot the analogous task for generalization error bounds based on uniform convergence. Unlike in the case of generalization error analysis, it seems that uniform convergence is really required here — the approximation has to be good uniformly on  $F_0$  and not only when the distances are small.

### 3.4 Extensions Towards Active Learning

The only assumption on the labeled learning sample we actually used in deriving our bounds is that the labels of the examples are assigned according to the target  $f_0$ . This is enough to guarantee that  $f_0 \in F_0$ , which is all we need in the proofs. Hence, the bounds will remain true even if we drop the assumption that the points  $X_i$ ,  $1 \leq i \leq n$ , are sampled from  $P_X$ . The unlabeled examples  $X_{n+j}$ ,  $1 \leq j \leq m$ , have to be distributed according to  $P_X$ , though, since otherwise the approximation  $\hat{d}$  would not necessarily converge to  $d$ .

A version of the semi-supervised model where only the unlabeled data is distributed according to  $P_X$  may be quite natural in many settings. For example, the set of examples to be labeled might be chosen by stratified sampling or in some other complex way, because one wants to focus labeling efforts to a set of points that is in some sense as informative as possible. With respect to our bounds, the efficiency of such sampling schemes can be measured in terms of the radius of the resulting version space. The less data is needed to make the radius of  $F_0$  small, the better.

Our bounds can be used in deriving new criteria for actively selecting the points in  $\mathcal{X}$  to be labeled, also. Namely, one can try to optimize the bound by selecting points to be labeled so that the *empirical radius* — the distance from the empirical center of  $F_0$  to the farthest classifier in  $F_0$  — decreases as much as possible when the labels are revealed. There are many variants of this active learning setting even if only label queries are considered: The learner can be forced to select all the points to be labeled before seeing any labeled examples, the learner may be allowed to query labels of points one by one in

an online fashion, or the active part of learning can start only after the learner has first obtained a (randomly chosen) labeled sample as in the non-active semi-supervised setting. From a technical point of view, the choice of the setting affects the bounds only through the set on which we have to be able to guarantee that  $\hat{d}$  is a good approximation to  $d$ . In the first two settings we have to have guarantees on the whole of  $F$ , while in the last setting it is enough that  $\hat{d}$  and  $d$  are close to each other on the version space related to the initial non-actively chosen sample. This last case is interesting because it models a situation in which the learner is not satisfied with the bound it got with the (randomly chosen) labeled data, and tries to improve on it by querying new labels.

## 4 Bounds for the General Case

The results obtained in the realizable case are interesting mostly from a theoretical point of view, since the assumption that the target lies in a (simple) hypothesis class known to the learner in advance is hardly ever justifiable in practice. This limitation is not a problem of our setting only, but affects, e.g., all results obtained in the original PAC model introduced by Valiant [11]. In this section, we drop all assumptions about the existence of a target, which makes our results applicable in all situations covered by the semi-supervised learning model.

Our bounds for the general case build on bounds for randomized classifiers. The idea is to use a randomized classifier for which a good generalization bound exists as an anchoring point for the generalization error of the learned deterministic classifier. The randomized classifier together with its bound thus plays the role the target function was in in the realizable case. Randomized bounds that can be used here include, e.g., the PAC-Bayesian bounds [12], the recent bounds for ensembles of classifiers created by an online learning algorithm [13], and the progressive validation bound [14]. Test set bounds can be interpreted as a special case of this setting in which the randomized classifier is actually deterministic. Also bagging and cross-validation can be used as bases for generalization error bounds. We have worked through instances of all the above mentioned bounds, but will cover only the bound based on cross-validation in this paper.

There are many reasons for being interested in deterministic classifiers even though the bounds for randomized classifiers are often tighter. First, deterministic classifiers are nicer to work with since the predictions they give do not change randomly over time. Second, using a randomized classifier often requires storing all the underlying deterministic classifiers in memory or otherwise at hand, although at times it is possible to represent the randomized classifier in a more concise form (e.g., as a distribution of perturbations to a single deterministic classifier). In many cases the deterministic classifiers are huge and so is their number, so the memory requirements may be enormous. Third, the randomization is often introduced only to facilitate (the analysis of) generalization performance, while the underlying learning algorithm is originally designed to learn single deterministic classifiers. This is the case, e.g., with the online

bound (when applied to batch algorithms) and the cross-validation bound. In such cases, aiming at bounds for the deterministic classifier learned based on all labeled data is very natural indeed.

#### 4.1 Derandomization by Voting

Suppose  $f$  is an arbitrary randomized classifier. Let  $f_{\text{vote}}$  be the deterministic *voting classifier* related to  $f$  given by  $f_{\text{vote}}(x) = \arg \max\{\mathbb{P}(f(x) = y) \mid y \in \mathcal{Y}\}$  (ties are broken arbitrarily). Replacing  $f$  by  $f_{\text{vote}}$  is a standard method of getting rid of randomness. The drawbacks of using  $f_{\text{vote}}$  instead of  $f$  are that (1) in the worst case,  $\epsilon(f_{\text{vote}}) = 2\epsilon(f)$  (this is the case if  $f$  is based on fair coin tosses and the target is  $1 - f_{\text{vote}}$ ), (2) using  $f_{\text{vote}}$  as a classifier requires storing all the classifiers underlying  $f$  in memory, and (3) one has to evaluate them all when classifying an instance. Given these,  $f_{\text{vote}}$  is probably not the classifier we are looking for. We know (1) is in general unavoidable if a deterministic approximation to  $f$  is desired, but we will show that the complexity issues (2) and (3) can be circumvented by accepting a small loss in generalization performance.

In the following theorems, the randomized classifiers  $f$  and  $g$  and the random variables  $\alpha$  and  $\beta$  may depend on the labeled and unlabeled data in any way. We leave the choice of  $\alpha$  and  $\beta$  intentionally open for the sake of generality. All probabilities are over the choice of data and the randomness in the classifiers.

The next Theorem is in a key role in all that follows.

**Theorem 5.** *Let  $f$  and  $g$  be randomized classifiers. If  $\mathbb{P}(\epsilon(f) \leq \alpha) \geq 1 - \delta/2$  and  $\mathbb{P}(d(f, g) \leq \beta) \geq 1 - \delta/2$ , then  $\mathbb{P}(\epsilon(g) \leq \alpha + \beta) \geq 1 - \delta$ , where the probabilities are over the labeled and unlabeled data as well as the randomness in the classifiers  $f$  and  $g$ .*

*Proof.* If  $f$  agrees with  $Y$  and  $g$  agrees with  $f$ , then  $g$  agrees with  $Y$ . Thus,  $g$  errs only if either  $f$  errs or  $g$  disagrees with  $f$ . By the assumptions, the definition of  $d$ , and the union bound, the probability for this event is at most  $\alpha + \beta$ .

Alternatively, one can use the triangle inequality for  $d$  and write

$$\epsilon(g) = d(g, Y) \leq d(g, f) + d(f, Y) = d(f, Y) + d(f, g) \leq \alpha + \beta,$$

where the last inequality is true with probability at least  $1 - \delta$  by the assumptions. □

As a simple corollary we get the following:

**Corollary 2.** *Let  $f$  be a randomized classifier. If  $\mathbb{P}(\epsilon(f) \leq \alpha) \geq 1 - \delta/2$  and  $\mathbb{P}(d(f, f_{\text{vote}}) \leq \beta) \geq 1 - \delta/2$ , then  $\mathbb{P}(\epsilon(f_{\text{vote}}) < \alpha + \beta) \geq 1 - \delta$ .*

In words, derandomizing  $f$  by replacing it with  $f_{\text{vote}}$  incurs a loss of at most  $d(f, f_{\text{vote}})$ . If  $f$  depends only on the labeled data, then  $d(f, f_{\text{vote}})$  is simply the probability of the event that the classifiers  $f$  and  $f_{\text{vote}}$  (fixed after seeing the labeled data) disagree. Thus, we can use  $\hat{d}(f, f_{\text{vote}})$  to obtain  $\beta$ . The same can be done in case of Theorem 5 if neither  $f$  nor  $g$  depend on the unlabeled data.

The next theorem shows that in case the bound for  $f$  is good,  $d(f, f_{\text{vote}})$  has to be small.

**Theorem 6.** *For any randomized classifier  $f$ , it is true that  $d(f, f_{\text{vote}}) \leq \epsilon(f)$ .*

*Proof.* Consider the learning problem  $P'$  defined by  $f$  as follows: Choose an  $X$  according to  $P_X$ , and let  $Y' = f(X)$ . It is easy to see that  $f_{\text{vote}}$  is the Bayes classifier for this problem and thus has the minimal probability of misclassifying  $(X, Y')$  [15]. By the definition of  $Y'$ , this probability is  $d(f, f_{\text{vote}})$ . Now  $d(f, f_{\text{vote}}) \leq \inf_g d(f, g) \leq d(f, Y) = \epsilon(f)$ , since  $Y$  can be viewed as a potential choice of  $g$ .  $\square$

Combining this theorem with Corollary 2 gives the known result that transforming a randomized classifier to a voting classifier at most doubles the generalization error. However, it may be that  $d(f, f_{\text{vote}})$  is much smaller than  $\epsilon(f)$  and at least much smaller than the best bounds for  $\epsilon(f)$ , so Corollary 2 may provide significant improvements over the factor 2 bound.

Another interesting consequence of Theorem 6 is that if a randomized classifier  $f$  does well, it is almost deterministic in the sense that its probability of disagreeing with the deterministic classifier  $f_{\text{vote}}$  — that is,  $d(f, f_{\text{vote}})$  — is small. In other words good randomized classifiers are almost deterministic on the parts of  $\mathcal{X}$  with significant probability. More exactly, the expected margin of a good randomized classifier has to be large.

The classifier  $f_{\text{vote}}$  is the best deterministic approximation to  $f$  in the sense that its probability of disagreeing with  $f$  is minimal. As a corollary it also always optimizes the bound of Theorem 5. However, optimizing the distance to  $f$  (equivalently, the bound in Theorem 5) is equivalent to optimizing the generalization error only if  $f_{\text{vote}}$  happens to be the Bayes classifier for  $P$ , which needs not be the case. This and the complexity of  $f_{\text{vote}}$  motivates us to look for other choices of  $g$  in Theorem 5.

One evident choice would be the classifier  $g^*$  that minimizes  $d(g, f)$  over the classifiers underlying  $f$ . By the Markov inequality, we have  $d(g^*, f_{\text{vote}}) \leq d(f, f_{\text{vote}})$ . Combining this with the triangle inequality, we get

$$d(f, g^*) \leq d(f, f_{\text{vote}}) + d(f_{\text{vote}}, g^*) \leq 2d(f, f_{\text{vote}}),$$

so at most a factor of 2 is lost in  $\beta$  by resorting to the (simple)  $g^*$  instead of the (complex)  $f_{\text{vote}}$ . A drawback is that  $g^*$  depends on the unknown  $d$  and thus has to be approximated by the classifier that optimizes  $\hat{d}(g, f)$  instead. Hence, to get good bounds, we have to be able to guarantee that  $\hat{d}(f, g)$  is close to  $d(f, g)$  over all  $g$  underlying  $f$  that we optimize over. This can be easily accomplished by using the union bound in case the set of these  $g$  is small (at least finite), but in general one may have to resort to more complicated uniform convergence techniques.

Some bounds for randomized classifiers suggest other choices for deterministic approximations. For example, in case of the bound for the ensembles of classifiers produced by an online algorithm and the bound for cross-validation, the most

natural choice is to use the classifier  $f_{\text{final}}$  learned based on all the labeled data in the role of  $g$  in Theorem 5. The next subsection is devoted to deriving such a bound for  $f_{\text{final}}$  starting from a cross-validation estimate.

#### 4.2 A Concrete Bound Based on Cross-Validation

Cross-validation works as follows. First, the labeled data is split into  $k$  subsets or folds of equal size, where  $k$  is a parameter of the method (for simplicity, we assume that  $n$  is divisible by  $k$ ). The  $i$ th fold thus consists of the points  $(X_{(i-1)n/k+1}, Y_{(i-1)n/k+1}), \dots, (X_{in/k}, Y_{in/k})$ , where  $i = 1, \dots, k$ . Then, the learning algorithm is run  $k$  times. In the  $i$ th run the examples not in fold  $i$  are used for learning and the examples in fold  $i$  for testing the learned hypothesis. This way, one gets  $k$  classifiers  $f_1, \dots, f_k$  and unbiased estimates  $\hat{\epsilon}_1, \dots, \hat{\epsilon}_k$  for their generalization errors  $\epsilon(f_1), \dots, \epsilon(f_k)$ . Here,

$$\hat{\epsilon}_i = \frac{k}{n} \sum_{j=1}^{n/k} \mathbb{I}[f_i(X_{(i-1)n/k+j}) \neq Y_{(i-1)n/k+j}].$$

The average of these estimates is often used in assessing the performance of the classifier  $f_{\text{final}}$  — the classifier learned by the same learning algorithm that produced  $f_1, \dots, f_k$ , but this time based on all the labeled data. Cross-validation is widely used in practice, even though there are no guarantees that the estimates it gives are meaningful.

We now show how to transform the heuristic cross-validation estimate into a generalization error bound for  $f_{\text{final}}$  by the method presented in the previous section. Let  $f$  be the randomized classifier obtained by choosing a classifier among the classifiers  $f_i$  uniformly at random. That is, let the set of classifiers underlying  $f$  be  $\{f_1, \dots, f_k\}$  and let  $\Theta_f$  have uniform distribution in  $\{1, \dots, k\}$ .

The following generalization error bound for  $f$  that builds on tight test set bounds [16] for the underlying classifiers  $f_i$  is to our knowledge new.

**Theorem 7.** *Let  $f$  be the randomized classifier obtained by cross-validation as explained above. Then with probability at least  $1 - \delta$  (over the choice of the labeled sample), we have*

$$\epsilon(f) \leq \frac{1}{k} \sum_{i=1}^k \overline{\text{Bin}}(\hat{\epsilon}_i, n/k, \delta/k),$$

where the inverse binomial tail [16]  $\overline{\text{Bin}}(\hat{p}, m, \delta)$  is the  $p$  for which

$$\sum_{i=0}^{\lceil \hat{p}m \rceil} \binom{m}{i} p^i (1-p)^{m-i} = \delta.$$

*Proof.* For each  $i = 1, \dots, k$ , we have  $\frac{n}{k}\hat{\epsilon}_i \sim \text{Bin}(\epsilon(f_i), \frac{n}{k})$ . Thus, by definition, it is true for each  $i$  that  $\epsilon(f_i) \leq \overline{\text{Bin}}(\hat{\epsilon}_i, n/k, \delta/k)$  with probability at least  $1 - \delta/k$ .

Using the definitions and the union bound, we get

$$\begin{aligned}\epsilon(f) &= P(f(X) \neq Y) = \sum_{i=1}^k P(f(X) \neq Y | \Theta_f = i) P(\Theta_f = i) \\ &= \frac{1}{k} \sum_{i=1}^k \epsilon(f_i) \leq \frac{1}{k} \sum_{i=1}^k \overline{\text{Bin}}(\hat{\epsilon}_i, n/k, \delta/k)\end{aligned}$$

with probability at least  $1 - \delta$ .  $\square$

Combining the above bound with Theorem 5 gives the following.

**Theorem 8.** *Let  $f_{\text{final}}$  be the classifier learned based on all the data and let  $f$  be as above. With probability at least  $1 - \delta$  (over the choice of labeled and unlabeled data and the randomization in  $f$ ), we have*

$$\epsilon(f_{\text{final}}) \leq \frac{1}{k} \sum_{i=1}^k \overline{\text{Bin}}(\hat{\epsilon}_i, n/k, \delta/(2k)) + \overline{\text{Bin}}(\hat{d}(f, f_{\text{final}}), m, \delta/2).$$

*Proof.* Use the result of Theorem 7 (with  $\delta/2$  in place of  $\delta$ ) to get  $\alpha$  and choose  $\beta = \overline{\text{Bin}}(\hat{d}(f, f_{\text{vote}}), m, \delta/2)$ . The result then follows from Theorem 5.  $\square$

The bound of Theorem 8 assumes nothing about the learning algorithm. For the bound to be tight, the algorithm has to produce hypotheses  $f_i$  with good generalization error. This is reflected to the bound by the (expectations of the) estimates  $\hat{\epsilon}_i$ . In addition, the algorithm has to be stable in two senses: First, the hypotheses  $f_i$  have to be relatively close to each other. Otherwise, transforming the bound for  $f$  into a bound for any deterministic classifier will incur a large loss by Theorem 6. Second, the classifier  $f_{\text{final}}$  has to be close to  $f$ , too. There are no guarantees for this in general. However, if  $d(f, f_{\text{final}})$  is large, using the cross-validation estimate to assess the performance of  $f_{\text{final}}$  would have been on shaky grounds anyway. Thus, the conditions required for our bound to be tight have to be true anyway in order for cross-validation to make sense.

The notion of stability required by our cross-validation bound to be tight resembles the various notions of stability studied in the learning theory literature. The connection of these notions to the generalization performance of an algorithm has received lots of attention recently. It has been shown that *training stability* (one notion of algorithmic stability) implies that the empirical and generalization errors of a learned classifier are close to each other. If the algorithm is ERM, then training stability is also necessary and sufficient for successful generalization. For this line of research, see [17] and the references therein.

The notions of algorithmic stability measure how much the error of a learned hypothesis (on a point or over the whole of  $\mathcal{X}$ ) may change when the labeled learning sample is perturbed slightly. Estimating these stability parameters based on the observed data only may be hard, since they are often defined in terms of expectations involving the unknown distribution  $P$ . This seems to seriously limit

the applicability of these stability concepts in practice. In contrast, the quantities in our bounds depend on unlabeled data only and no a priori assumptions of stability are needed. Of course, we can have no a priori guarantees on the quality of the bound either, but we hypothesize that if the algorithm is, e.g., training set stable and the bound of Theorem 7 is small, then our cross-validation bound is small, too.

### 4.3 Empirical Experiments with the Cross-Validation Bound

In this section we present results of experiments with the bound of Theorem 8 applied to SVMs and the MNIST dataset. The MNIST dataset consists of 60 000 labeled training examples and 10 000 labeled test examples of  $28 \times 28$  gray scale images of handwritten digits from 0 to 9. We combined the training and test sets, permuted the data randomly, and used the 60 000 first examples of the permuted data set as the labeled data and forgot the labels of the remaining 10 000 examples to get a set of unlabeled data. The only preprocessing was scaling the pixel intensities to  $[-1, 1]$ .

As the learning algorithm, we used `svmlight` [18], a standard implementation of the  $C$ -SVM learning algorithm. The algorithm is capable of solving binary problems only, so we transformed the original learning problem into ten 1 vs rest problems. That is, for each  $i \in \{0, \dots, 9\}$ , classifier  $i$  was provided with training examples that were labeled  $+1$  if the class was  $i$  and  $-1$  otherwise. The predictions were combined by choosing the class corresponding to the classifier whose output was the largest. All this is done internally, so that as far as the bound is concerned, the classifiers appear to be multi-class classifiers. As a kernel we used a degree 4 polynomial kernel, and chose the default value for  $C$ . The computation time required to transform the cross-validation estimate into a generalization error bound is only a few seconds on a standard PC (in addition to the time taken by the SVMs to classify the unlabeled data). In this sense transforming a cross-validation estimate into a semi-supervised bound is almost free.

Table 1 summarizes the bounds obtained for various hypotheses. The classifier  $f_{\text{final}}$  is the final multi-class SVM learned based on all labeled data. The bound

**Table 1.** Semi-supervised and test set bounds for (combinations of) SVMs on the MNIST data set with  $n = 60\,000$ ,  $m = 10\,000$ ,  $k = 10$ , and  $\delta = 0.01$

Bound for the randomized $f$ (Theorem 7)	2.16%
Bound for $f_{\text{vote}}$	2.74%
Bound for $f_{\text{final}}$ (Theorem 8)	2.84%
Bound for the best $f_i$ underlying $f$	2.89%
Empirical error of $f_{\text{final}}$ (on the “unlabeled” data)	1.49%
Test set bound for $f_{\text{final}}$ (on the “unlabeled” data)	1.80%

for  $f_{\text{final}}$  is almost as good as the bound for  $f_{\text{vote}}$ , and neither overshoots the exact test set bound [16] (computed by cheating and looking at the labels of the 10 000 “unlabeled” examples) by more than about a percent. This is in striking contrast with the training set bounds for  $f_{\text{final}}$  in the standard supervised setting without unlabeled data. The tightest of these bounds are applicable to two class problems only and usually even then quite loose. We did not experiment with any of these alternative bounds, but feel that it is safe to claim that they all would have been way above 100% on the multi-class learning problem at hand (for a survey on some of these bounds and their looseness, see [19]). Also note that the bound for the best  $f_i$  underlying  $f$  is worse than the bound for  $f_{\text{final}}$ . The intuitive explanation is that  $f_{\text{final}}$  has an advantage because it is learned based on all the labeled data, but it is surprising that this advantage shows up in the bounds. The test set performance of  $f_{\text{vote}}$  is slightly better than the test set performance of  $f$ , showing that derandomization may actually increase the accuracy although its effect on our bound is negative.

In summary, our initial empirical experiments seem to indicate that the proposed cross-validation bound that uses unlabeled data is considerably tighter than earlier bounds that do not require a separate labeled test set. The bound is not quite as tight as the test set bound one could use if the labels of the unlabeled sample were known, but this is to be expected as the semi-supervised bound has access to less information. In a sense, the unlabeled sample can be viewed as a cheap but still good replacement for a labeled test set.

## 5 Future Work

Besides applying the method presented in Section 4 to other bounds for randomized classifiers, we plan to investigate other loss functions than the 0-1 loss studied in this paper. We also plan to study the use of the bounds for model selection and other tasks. A problem with the cross-validation bound is that even though it can be used to tell how well an algorithm did on a dataset, it gives little guidance in designing algorithms that would do better. This is because the bound views the algorithm as a black box and hence cannot identify directly which of its properties are important for generalization. We hope that other bounds like the bound for ensembles of classifiers learned by an online algorithm and the PAC-Bayesian bounds may be more useful in this respect. In the realizable case, the most interesting direction seems to be pursuing the extensions to active learning and their connections to query learning and other active learning approaches.

Approximating  $d$  is only one of the possible uses of unlabeled data one can think of, and the use of  $d$  is not restricted to derandomizing classifiers. Same kinds of arguments can be used if one, e.g., wants to switch from a classifier  $f$  with a non-intelligible representation but good generalization performance (a neural network or SVM) to a classifier  $g$  with a more understandable representation (a rule set or decision tree). If  $d(f, g)$  is small, then the good generalization

performance of  $f$  will be inherited by the more comprehensible  $g$ . Of course, similar things can be done within a representation scheme, e.g., to find good decision tree prunings. We plan to continue working on these and other uses of unlabeled data in the near future.

**Acknowledgments.** I wish to thank John Langford, Jyrki Kivinen, Anssi Kääriäinen, and Taneli Mielikäinen for helpful discussions.

## References

1. Madani, O., Pennock, D.M., Flake, G.W.: Co-validation: Using model disagreement to validate classification algorithms. In: NIPS 2004 Preproceedings. (2004)
2. Balcan, M.F., Blum, A.: A PAC-style model for learning from labeled and unlabeled data (2004) Draft.
3. Castelli, V., Cover, T.M.: On the exponential value of labeled samples. *Pattern Recognition Letters* **16** (1995) 105–111
4. Ratsaby, J., Venkatesh, S.S.: Learning from a mixture of labeled and unlabeled examples with parametric side information. In: Proceedings of the 8th Annual Conference on Computational Learning Theory (COLT'95), New York, NY, USA, ACM Press (1995) 412–417
5. Vapnik, V.N.: *Statistical Learning Theory*. John Wiley and Sons, New York (1998)
6. Schuurmans, D., Southey, F.: Metric-based methods for adaptive model selection and regularization. *Machine Learning* **42** (2002) 51–84
7. Bengio, Y., Chapados, N.: Extensions to metric-based model selection. *Journal of Machine Learning Research* **3** (2003) 1209–1227
8. Ben-David, S., Itai, A., Kushilevitz, E.: Learning by distances. *Information and Computation* **117** (1995) 240–250
9. Bartlett, P.L., Mendelson, S.: Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research* **3** (2002) 463–482
10. Kääriäinen, M.: Relating the Rademacher and VC bounds. Technical Report Report C-2004-57, Department of Computer Science, Series of Publications C (2004)
11. Valiant, L.G.: A theory of the learnable. *Communications of the ACM* **27** (1984) 1134–1142
12. McAllester, D.A.: PAC-Bayesian stochastic model selection. *Machine Learning* **51** (2003) 5–21
13. Cesa-Bianchi, N., Gentile, C.: Improved risk tail bounds for on-line algorithms (2004) A presentation in the (Ab)use of Bounds workshop.
14. Blum, A., Kalai, A., Langford, J.: Beating the hold-out: bounds for k-fold and progressive cross-validation. In: Proceedings of the 12th Annual Conference on Computational Learning Theory, New York, NY, ACM Press (1999) 203–208
15. Devroye, L., Györfi, L., Lugosi, G.: *A Probabilistic Theory of Pattern Recognition*. Volume 31 of Applications of Mathematics. Springer, Berlin Heidelberg New York (1996)
16. Langford, J.: Practical prediction theory for classification (2003) A tutorial presented at ICML 2003. Available at [http://hunch.net/~jl/projects/prediction\\_bounds/tutorial/tutorial.pdf](http://hunch.net/~jl/projects/prediction_bounds/tutorial/tutorial.pdf).

17. Kutin, S., Niyogi, P.: Almost-everywhere algorithmic stability and generalization error. In: *Proceedings of Uncertainty in AI. (2002)* 275–282
18. Joachims, T.: Making large-scale SVM learning practical. In Schölkopf, B., Burges, C., Smola, A., eds.: *Advances in Kernel Methods – Support Vector Learning*. MIT-Press (1999)
19. Seeger, M.: *Bayesian Gaussian Process Models: PAC-Bayesian Generalisation Error Bounds and Sparse Approximations*. PhD thesis, University of Edinburgh (2003)