Innovations in Computer Science 2011

Cross-Validation and Mean-Square Stability

Satyen Kale Ravi Kumar Sergei Vassilvitskii
Yahoo! Research, 701 First Ave, Sunnyvale, CA 94089, USA
{skale, ravikumar, sergei}@yahoo-inc.com

Abstract: A popular practical method of obtaining a good estimate of the error rate of a learning algorithm is *k-fold cross-validation*. Here, the set of examples is first partitioned into k equal-sized folds. Each fold acts as a test set for evaluating the hypothesis learned on the other $k-1$ folds. The average error across the k hypotheses is used as an estimate of the error rate. Although widely used, especially with small values of k (such as 10), the cross-validation method has heretofore resisted theoretical analysis due to the fact that the k distinct estimates have inherent correlations between them. With only sanity-check bounds known, there is no compelling reason to use the k-fold cross-validation estimate over a simpler holdout estimate.
Conventional wisdom is that the averaging in cross-validation leads to a tighter concentration of the estimate of the error around its mean. In this paper, we show that the conventional wisdom is essentially correct. We analyze the reduction in variance of the gap between the cross-validation estimate and the true error rate, and show that for a large family of stable algorithms cross-validation achieves a near optimal variance reduction factor of $(1 + o(1))k$. In these cases, the k different estimates are essentially *independent* of each other.
To proceed with the analysis, we define a new measure of algorithm stability, called *mean-square stability*. This measure is weaker than most stability notions described in the literature, and encompasses a large class of algorithms including bounded SVM regression and regularized least-squares regression. For slightly less stable algorithms such as t-nearest-neighbor, we show that cross-validation leads to an $O(1\sqrt{k})$ reduction in the variance of the generalization error.

Keywords: cross-validation, stability, generalization error.

1 Introduction

The primary goal of many machine learning algorithms is to produce a hypothesis that has a low error rate on unseen examples. However, in many situations (such as parameter tuning, algorithm selection, etc.), obtaining a good estimate of the error rate is just as important. This can be easily accomplished by splitting the set of input examples into two parts: a training set, which is used as input to the learning algorithm, and a holdout test set, which is used to evaluate the hypothesis. Since the learning algorithm does not "see" the examples in the test set before the evaluation, it is easy to prove that this results in an unbiased estimator of the error rate. The difference between the estimator and the true error rate is called the generalization error. Getting good bounds on the generalization error is integral to understanding the performance of a learning algorithm or comparing two different learning algorithms.

Since labeled data is expensive, there is typically a tension between allocating enough data to the training set so that the algorithm can learn a better hypothesis, and allocating data to the test set so that one can produce a better estimate of the error rate. As a compromise, cross-validation is often employed to provide several (dependent) estimates of the error rate. Here, the input is first randomly partitioned into k parts, called *folds*, of n examples each. Then, the algorithm is repeatedly trained on the data in $k-1$ of the folds, and evaluated on the remaining n examples. If one uses a classifier randomly chosen among the k classifiers found on different folds, then its error rate can be estimated using the cross-validation estimate: this is simply the average of the errors obtained in the k trials.

Blum, Kalai, and Langford [2] showed that for all algorithms satisfying some mild non-degeneracy conditions, the variance (and other higher moments) of the error of the cross-validation estimate is always at most that of the holdout estimate. They also show that under a very mild non-degeneracy condition, all the moments are strictly smaller as well, although they do not specify by how much. Quantifying the improvement gained by cross-validation has remained an open problem for the past decade. Were the k different clas-

sifiers trained on independently chosen data sets, one could hope for an $O(1/k)$ reduction in variance. However, cross-validation introduces intricate correlations between the hypotheses learned on the different folds, and the examples on which they are evaluated. Unraveling these dependencies is key to understanding the power of cross-validation, and a formal analysis even for the special case of $k = 2$ has remained elusive.

1.1 Algorithm stability

In the special case of leave-one-out cross-validation, algorithm stability has been a useful tool to prove bounds on the generalization error. In this approach, the algorithm is trained on all but one of the available examples and tested on the remaining one. Rogers and Wagner [8] and Devroye and Wagner [5] were the first to show that for some specific learning algorithms, the leave-one-out cross-validation estimate is close to the true error rate of the classifier trained on the *entire* data set. Implicit in their argument was a notion of stability of learning algorithms, which quantifies the change in the algorithm's performance if a single training example is changed.

Kearns and Ron [6] (and later, Anthony and Holden [1] for k-fold cross-validation) prove the so-called *sanity-check* bounds. Assuming some stability of the learning algorithm, they show that the generalization error between the cross-validation estimate and the true error rate of the classifier trained on the entire data set is almost as small as the error for the resubstitution estimate, i.e., the empirical error rate obtained by running the learned classifier on the training set. In a similar line of work, Bousquet and Elisseeff [3] and Kutin and Niyogi [7] showed that a notion of training stability is necessary and sufficient to obtain good bounds on the generalization error.

Unfortunately, these results do not seem to generalize easily to k-fold cross-validation for small values of k. In practice, leave-one-out cross-validation is very expensive when the number of training examples runs into millions and five- or ten-fold cross-validation may be the only feasible choice. Such k-fold cross-validation estimates are widely used to claim superiority of one algorithm over another. However, there is no theoretical justification for why the k-fold cross-validation estimate would be *much* better than simply using one holdout estimate, since the sanity-check bounds and the bounds of [2] only show that it is no worse. The intuition is that the averaging step in k-fold cross-validation makes the estimate more concentrated around its mean, and one might even naïvely expect a k-fold reduction in variance (as would happen for averaging k independently trained classifiers). Unlike previous work which focused on improving generalization error bounds, in this paper, we focus on showing that this intuition is essentially correct: *k-fold cross-validation can significantly reduce the variance of the generalization error, even for small k.*

Our work is based on developing a new notion of algorithmic stability, called the *mean-square stability*. This is weaker than most of the existing stability notions and is implied by both the weak-hypothesis stability of Devroye and Wagner [5] and the weak-L_1 stability of Kutin and Niyogi [7]. Intuitively, this notion limits how much a change of one example in a typical training set affects the loss on a randomly chosen test example. Since the notion is weaker than the previous two, we can leverage the known literature on the stability of learning algorithms and show that wide classes of algorithms have low mean-square stability. For example, both SVMs [3] and empirical risk minimization (ERM) [7] fall into this category.

Although this notion of stability holds only with respect to changing a single example, we show that it is enough to obtain near optimal variance reduction even for a small number of folds ($k = 2, 3, \ldots$). At first glance this is surprising, since we do not restrict the algorithm to give stable results across different folds directly, rather we show how the stability allows us to reason about the correlation between the different runs.

1.2 Contributions

We analyze the variance reduction of the generalization error due to cross-validation and show that assuming mean-square stability, one can prove a near-optimal reduction in variance. Namely, for many learning algorithms, the variance of the generalization error obtained using k-fold cross-validation scales as a $(\frac{1+o(1)}{k})$-fraction of the variance on a single holdout test set. This extends the theory of stability of learning algorithms, which was previously only used to prove generalization error bounds, to showing variance reduction. Note that the current techniques can only be used to show variance reduction; such reductions for higher moments are beyond the scope of this paper.

Our second contribution is the notion of mean-square stability used to obtain the results above.

Mean-square stability is weaker than most of the previously defined stability notions and yet is sufficient to obtain variance reduction.

2 Preliminaries

We have a space of points $\mathcal{X}$ and a set of labels $\mathcal{Y}$. Labeled examples are drawn from an unknown distribution $\mathcal{D}$ over $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. Whenever we refer to an example $z \in \mathcal{Z}$, we implicitly assume that it is drawn from $\mathcal{D}$, and, unless specified otherwise, all probabilities, expectations, and variances are computed when examples are chosen i.i.d. from $\mathcal{D}$. A *hypothesis* is a function[1] $h : \mathcal{X} \to \mathcal{Y}$. A *learning algorithm* $\mathcal{A}$ is a function that takes as input a data set S of examples from $\mathcal{Z}$, and outputs a hypothesis $h_{\mathcal{A}(S)}$.

We assume that the learning algorithm $\mathcal{A}$ under consideration is *symmetric*, i.e., its output is the same if the positions of two examples in the input data set are flipped. Such an assumption can be made without loss of generality since any algorithm can be made symmetric simply by randomly permuting the input data set prior to running the algorithm on it.

Error rate and generalization error. To measure the performance of a hypothesis h, we introduce a *loss function* $\ell : \mathcal{Z} \to [0, M]$, where M is a known upper bound on the loss. The *loss* of a hypothesis h on an example $z = (x, y)$ is defined to be $\ell_h(z) := \ell(x, h(x))$. The *error rate* of a hypothesis h is defined to be the expected loss of h on a random example, viz.

$$\bar{\ell}_h \;:=\; \mathop{\mathbb{E}}_{z}[\ell_h(z)].$$

To estimate the error rate of a hypothesis, we may use a test set T of examples. The error of h on T is defined to be

$$\ell_h(T) \;:=\; \frac{1}{|T|} \sum_{z \in T} \ell_h(z).$$

Note that $\ell_h(T)$ is an unbiased estimator of $\bar{\ell}_h$. We measure the closeness of estimation by the *generalization error* or *generalization defect* for the hypothesis h w.r.t. a test set T, which is defined to be

$$\mathtt{gen}_h(T) \;:=\; \ell_h(T) - \bar{\ell}_h.$$

[1]This function could be probabilistic, in which case the output of h on some point $x \in \mathcal{X}$ is some probabilistically chosen label $y \in \mathcal{Y}$. To simplify our exposition, we assume that our hypotheses are deterministic. Our results hold without change in the probabilistic case as well. Here, the loss of the hypothesis on an example is defined to be the expected loss of the label predicted by the hypothesis.

Cross-validation. We now introduce k-fold *cross-validation* (CV for short) for some algorithm $\mathcal{A}$. We start with a set U of $m = nk$ examples drawn from $\mathcal{D}$. We think of k as being a fixed parameter, and n increasing to infinity. We split up the m examples randomly into k folds $T_1, T_2, \ldots, T_k$ of equal size n.

The ith classifier is defined to be the output of $\mathcal{A}$ when supplied with all examples in U except for those in T_i, viz. $h_i := \mathcal{A}(U \setminus T_i)$. We measure the performance of h_i on the test set T_i, and define the ith holdout estimate to be

$$\mathtt{ho}_i \;:=\; \ell_{h_i}(T_i).$$

The generalization error of ith holdout estimate is then $\mathtt{gen}_i := \mathtt{gen}_{h_i}(T_i) = \mathtt{ho}_i - \bar{\ell}_{h_i}$. We are interested in the variance of $\mathtt{gen}_i$. Since $\mathbb{E}_U[\mathtt{gen}_i] = \mathbb{E}_{U \setminus T_i}[\mathbb{E}_{T_i}[\ell_{h_i}(T_i) - \bar{\ell}_{h_i} \mid U \setminus T_i]] = 0$, we have $\mathbf{var}_U[\mathtt{gen}_i] = \mathbb{E}_U[\mathtt{gen}_i^2]$.

The *cross-validation classifier* is defined to be one that, for every example x, chooses one of the k classifiers h_i uniformly at random, and uses it to label x. The error rate of the CV-classifier is thus $\bar{\ell}_{\mathtt{cv}} = \frac{1}{k}\sum_{i=1}^{k} \bar{\ell}_{h_i}$. The *cross-validation estimate* for the error rate of the CV-classifier is defined to be

$$\mathtt{cv} \;:=\; \frac{1}{k} \sum_{i=1}^{k} \mathtt{ho}_i.$$

The generalization error for the CV-estimate is $\mathtt{gen}_{\mathtt{cv}} = \mathtt{cv} - \bar{\ell}_{\mathtt{cv}} = \frac{1}{k}\sum_{i=1}^{k} \mathtt{gen}_i$. As before, we have $\mathbf{var}_U[\mathtt{gen}_{\mathtt{cv}}] = \mathbb{E}_U[\mathtt{gen}_{\mathtt{cv}}^2]$.

Since the CV-estimate averages k other estimates, one might expect it to capture the error rate of the CV-classifier better than any holdout estimate can capture the error rate of the corresponding classifier. To quantify this, we would like to show that $\mathbf{var}_U[\mathtt{gen}_{\mathtt{cv}}]$ is much smaller than $\mathbf{var}_U[\mathtt{gen}_1]$ (we choose the first classifier without loss of generality, since by symmetry we have $\mathbf{var}_U[\mathtt{gen}_1] = \mathbf{var}_U[\mathtt{gen}_i]$ for any other i).

What variance reduction might one hope for? If one averages k classifiers trained on k independent training sets, then the variance of the generalization error of the resulting classifier drops down by a factor of k compared to the variance of the generalization error of any one classifier. Clearly, this is the best variance reduction one can achieve, since the k classifiers available after cross-validation are not trained on independently drawn training sets: in fact, the training sets are highly correlated. We will show that

for many well-known learning algorithms, such near-optimal variance reduction is indeed possible.

3 Mean-square stability

To aid in our analysis we define a new notion of stability of a learning algorithm:

Definition 1(Mean-square stability). *A learning algorithm $\mathcal{A}$ is* β-mean-square stable *w.r.t. ℓ if for any $i \in \{1, 2, \ldots, m\}$,*

$$\mathop{\mathbb{E}}_{S,z,z'}[(\ell_{\mathcal{A}(S)}(z) - \ell_{\mathcal{A}(S^{i,z'})}(z))^2] \leqslant \beta.$$

Here, S, z, z' are sampled i.i.d. from $\mathcal{D}$, and $S^{i,z'}$ is the data set formed by replacing the ith element of S by z'.

Remark 1. The choice of the index i in the definition of mean-square stability is not important since the algorithm is assumed to be symmetric. Hence, without loss of generality, we will assume that $i = m$, i.e., only the last example is replaced by a new one.

Remark 2. Mean-square stability can also be viewed as requiring that on average (over a randomly chosen test example z, and a randomly chosen training set S' of size $m-1$), the variance of the loss on the test example when the mth training example is chosen randomly is small: this is because

$$\begin{aligned}&\mathop{\mathbf{var}}_{z'}[\ell_{\mathcal{A}(S' \cup z')}(z) | S', z] \\ &= \frac{1}{2} \mathop{\mathbb{E}}_{z', z''}[(\ell_{\mathcal{A}(S' \cup z')}(z) - \ell_{\mathcal{A}(S' \cup z'')}(z))^2 | S', z],\end{aligned}$$

and so

$$\begin{aligned}&\mathop{\mathbb{E}}_{S,z,z'}[(\ell_{\mathcal{A}(S)}(z) - \ell_{\mathcal{A}(S^{i,z'})}(z))^2] \\ &= 2 \mathop{\mathbb{E}}_{S',z}\left[\mathop{\mathbf{var}}_{z'}[\ell_{\mathcal{A}(S' \cup z')}(z) | S', z]\right].\end{aligned}$$

Kutin and Niyogi [7] have defined several notions of stability of learning algorithms and established implications between them indicating which notion is weaker than which other. While the goal of their work is to understand the weakest notions of stability that are sufficient to show good generalization error bounds, unlike our goal of showing variance reduction, it is nonetheless worthwhile to understand where our mean-square stability notion fits in the hierarchy of stability notions. We now give several such implications indicating that mean-square stability is a very weak notion, although not as weak as the weakest notion of stability (defined by Kearns and Ron [6]). However, since mean-square stability is weaker than weak-L_1 stability (see Lemma 4 below), and it is known (see [7]) that weak-L_1 stability is too weak to show generalization error bounds, it follows that mean-square stability is also too weak to show generalization error bounds. Even so, it is strong enough to show variance reduction bounds, as we prove later.

The implications between the various stability notions will allow us to harness the stability results already proven for a wide variety of learning algorithms and show how we get near-optimal variance reduction in these cases.

We begin by defining four more notions of stability, in increasing order of weakness, using the same notation as in Definition 1. Note that our naming convention for these notions of stability mirrors those of Kutin and Niyogi [7]. To understand these definitions, it is best to think of the loss function ℓ as being a continuous valued function like the hinge loss or square loss, rather than a discrete function like the $\{0, 1\}$ loss. This is because we measure stability in terms of closeness between the losses of two hypotheses that are obtained by training on data sets that differ in only one example. For a discrete loss function like the $\{0, 1\}$ loss, we end up with either trivialities (closeness becomes equivalent to identity for some notions of stability) or degeneracies (different notions of stability become identical).

The strongest notion of stability is the uniform stability notion defined by Bousquet and Elisseeff [3]:

Definition 2(Uniform stability). *A learning algorithm $\mathcal{A}$ is* β-uniform stable *w.r.t. ℓ if for any $i \in \{1, 2, \ldots, m\}$, we have*

$$\max_{S,z,z'} |\ell_{\mathcal{A}(S)}(z) - \ell_{\mathcal{A}(S^{i,z'})}(z)| \leqslant \beta.$$

The next one is the weak-hypothesis stability notion defined by Devroye and Wagner [5]:

Defintion 3(Weak-hypothesis stability). *A learning algorithm $\mathcal{A}$ is* (β, δ)-weak-hypothesis stable *w.r.t. ℓ if for any $i \in \{1, 2, \ldots, m\}$, with probability at least $1 - \delta$ over the choice of S, z', we have*

$$\max_{z} |\ell_{\mathcal{A}(S)}(z) - \ell_{\mathcal{A}(S^{i,z'})}(z)| \leqslant \beta.$$

The third one is the weak-L_1 stability notion defined by Kutin and Niyogi [7]:

Definition 4 (Weak-L_1 stability). *A learning algorithm $\mathcal{A}$ is* (β, δ)-weak-L_1 stable *w.r.t. ℓ if for any*

$i \in \{1, 2, \ldots, m\}$, with probability at least $1 - \delta$ over the choice of S, z', we have

$$\mathop{\mathbb{E}}_{z}[|\ell_{\mathcal{A}(S)}(z) - \ell_{\mathcal{A}(S^{i,z'})}(z)|] \leqslant \beta.$$

The only stability notion weaker than weak-L_1 stability notion is the weak-error stability notion defined by Kearns and Ron [6]:

Definition 5 (Weak-error stability). *A learning algorithm $\mathcal{A}$ is (β, δ)-*weak-error stable *w.r.t. ℓ if for any $i \in \{1, 2, \ldots, m\}$, with probability at least $1 - \delta$ over the choice of S, z', we have*

$$|\bar{\ell}_{\mathcal{A}(S)} - \bar{\ell}_{\mathcal{A}(S^{i,z'})}| \leqslant \beta.$$

The following lemma shows that mean-square stability is a weaker notion than weak-hypothesis stability and weak-L_1 stability. The proof follows directly from the definitions of the notions of stability, and using the fact that $|\ell_{\mathcal{A}(S)}(z) - \ell_{\mathcal{A}(S^{i,z'})}(z)| \leqslant M$, and is hence omitted.

Lemma 1.

1. *If $\mathcal{A}$ is β-uniform stable, then it is $(\beta, 0)$-weak-hypothesis stable.*
2. *If $\mathcal{A}$ is (β, δ)-weak-hypothesis stable, then it is $(\beta^2 + M^2\delta)$-mean-square stable.*
3. *If $\mathcal{A}$ is (β, δ)-weak-L_1 stable, then it is $(M\beta + M^2\delta)$-mean-square stable.*

The relation between mean-square stability and weak-error stability is somewhat unclear, although it seems that weak-error stability is slightly weaker, as the following lemma shows:

Lemma 2. *If $\mathcal{A}$ is β-mean-square stable, then for any $\beta' > 0$, it is $(\beta', \frac{\beta}{\beta'^2})$-weak-error stable.*

Proof. Using the β mean-square stability of $\mathcal{A}$ and Jensen's inequality we get

$$\begin{aligned}
&\mathop{\mathbb{E}}_{S,z'}[(\bar{\ell}_{\mathcal{A}(S)} - \bar{\ell}_{\mathcal{A}(S^{i,z'})})^2] \\
&= \mathop{\mathbb{E}}_{S,z'}[(\mathop{\mathbb{E}}_{z}[\ell_{\mathcal{A}(S)}(z)] - \mathop{\mathbb{E}}_{z}[\ell_{\mathcal{A}(S^{i,z'})}(z)])^2] \\
&\leqslant \mathop{\mathbb{E}}_{S,z,z'}[(\ell_{\mathcal{A}(S)}(z) - \ell_{\mathcal{A}(S^{i,z'})}(z))^2] \leqslant \beta.
\end{aligned}$$

Using the fact that $\mathbb{E}_{S,z'}[\bar{\ell}_{\mathcal{A}(S)} - \bar{\ell}_{\mathcal{A}(S^{i,z'})}] = 0$, we thus get that

$$\begin{aligned}
&\mathop{\mathbf{var}}_{S',z'}[\bar{\ell}_{\mathcal{A}(S)} - \bar{\ell}_{\mathcal{A}(S^{i,z'})}] \\
&= \mathop{\mathbb{E}}_{S,z'}[(\bar{\ell}_{\mathcal{A}(S)} - \bar{\ell}_{\mathcal{A}(S^{i,z'})})^2] \leqslant \beta,
\end{aligned}$$

and hence by Chebyshev's inequality we get

$$\Pr_{S',z'}[|\bar{\ell}_{\mathcal{A}(S)} - \bar{\ell}_{\mathcal{A}(S^{i,z'})}| > \beta'] \leqslant \frac{\beta}{\beta'^2}.$$

□

4 Variance reduction via cross-validation

In this section, we show our main result regarding the variance reduction possible using k-fold cross-validation.

We first state a key tool that will be used in our analysis.

Theorem 1(Steele's inequality [9]). *Let $F : X^n \to \mathbb{R}$ be a function defined on data sets of size n. Let T be a data set of n i.i.d. examples drawn from some unknown distribution $\mathcal{D}$. Let T^i be the data set formed from T by replacing the ith element of T by a new independently drawn example z'. Then we have*

$$\mathop{\mathbf{var}}_{T}[F(T)] \leqslant \frac{1}{2}\sum_{i=1}^{n} \mathop{\mathbb{E}}_{T,z'}[(F(T) - F(T^i))^2].$$

Note that if F is a symmetric function of T, then the above inequality can be simplified to

$$\mathop{\mathbf{var}}_{T}[F(T)] \leqslant \frac{n}{2} \mathop{\mathbb{E}}_{T,z'}[(F(T) - F(\tilde{T}))^2],$$

where $\tilde{T}$ is formed by replacing a random element of T by z'. Since all learning algorithms we will consider are symmetric, we can directly use the simplified form of Steele's inequality.

We next state and prove our main result.

Theorem 2. *Suppose the learning algorithm $\mathcal{A}$ is β-mean-square stable w.r.t ℓ. Then*

$$\begin{aligned}
\mathop{\mathbf{var}}_{U}[\mathtt{gen}_{\mathtt{cv}}] &\leqslant \frac{1}{k}\mathop{\mathbf{var}}_{U}[\mathtt{gen}_1] \\
&\quad + \left(1 - \frac{1}{k}\right)\sqrt{\frac{\beta \mathop{\mathbf{var}}_{U}[\mathtt{gen}_1]}{2}}.
\end{aligned}$$

Proof. We start with massaging $\mathbf{var}_U[\mathtt{gen}_{\mathtt{cv}}]$ into a more manageable form. The analysis which follows allows decoupling of the correlations that arise due to the averaging over the k-folds without much loss. We have

$$\mathop{\mathbf{var}}_{U}[\mathtt{gen}_{\mathtt{cv}}] \tag{1}$$

$$= \operatorname*{\mathbf{var}}_U\left[\frac{1}{k}\sum_{i=1}^{k}\mathtt{gen}_i\right] = \frac{1}{k^2}\operatorname*{\mathbf{var}}_U\left[\sum_{i=1}^{k}\mathtt{gen}_i\right]$$

$$= \frac{1}{k^2}\sum_{i=1}^{k}\sum_{j=1}^{k}\operatorname*{\mathbf{cov}}_U(\mathtt{gen}_i,\mathtt{gen}_j)$$

$$= \frac{1}{k^2}\left(\sum_{i=1}^{k}\operatorname*{\mathbf{var}}_U(\mathtt{gen}_i) \;+ \sum_{i=1}^{k}\sum_{j\neq i}\operatorname*{\mathbf{cov}}_U(\mathtt{gen}_i,\mathtt{gen}_j)\right)$$

$$= \frac{1}{k}\operatorname*{\mathbf{var}}_U(\mathtt{gen}_1) + \left(1-\frac{1}{k}\right)\operatorname*{\mathbf{cov}}_U(\mathtt{gen}_1,\mathtt{gen}_2). \tag{2}$$

Here we used the facts that for any i we have $\mathbf{var}_U[\mathtt{gen}_i] = \mathbf{var}_U[\mathtt{gen}_1]$, and for any $i \neq j$, $\mathbf{cov}_U(\mathtt{gen}_i,\mathtt{gen}_j) = \mathbf{cov}_U(\mathtt{gen}_1,\mathtt{gen}_2)$, by symmetry. The first term, $\frac{1}{k}\mathbf{var}_U(\mathtt{gen}_1)$ is the optimal variance reduction one can expect to get by averaging k classifiers trained on independent data sets. To bound the excess variance, we need to bound $\mathbf{cov}_U(\mathtt{gen}_1,\mathtt{gen}_2)$.

We first note that at this stage we can recover the first variance reduction result of Blum, Kalai, and Langford [2]. By the Cauchy–Schwarz inequality, we have

$$\operatorname*{\mathbf{cov}}_U(\mathtt{gen}_1,\mathtt{gen}_2) \leqslant \sqrt{\operatorname*{\mathbf{var}}_U[\mathtt{gen}_1]\operatorname*{\mathbf{var}}_U[\mathtt{gen}_2]}$$
$$= \operatorname*{\mathbf{var}}_U[\mathtt{gen}_1], \tag{3}$$

implying, in conjunction with (2) the result of [2] that $\mathbf{var}_U[\mathtt{gen}_{\mathtt{cv}}] \leqslant \mathbf{var}_U[\mathtt{gen}_1]$.

We now turn to a more nuanced analysis. To do this, we introduce some notation. Let T be the first fold, T' the second, and let $S = U \setminus (T \cup T')$. Using the law of total covariance, conditioning on S, T we get

$$\operatorname*{\mathbf{cov}}_U(\mathtt{gen}_1,\mathtt{gen}_2) \tag{4}$$

$$= \mathop{\mathbb{E}}_{S,T}\left[\operatorname*{\mathbf{cov}}_{T'}(\mathtt{gen}_1,\mathtt{gen}_2|S,T)\right]$$
$$+ \operatorname*{\mathbf{cov}}_{S,T}(\mathop{\mathbb{E}}_{T'}[\mathtt{gen}_1|S,T], \mathop{\mathbb{E}}_{T'}[\mathtt{gen}_2|S,T])$$
$$= \mathop{\mathbb{E}}_{S,T}\left[\operatorname*{\mathbf{cov}}_{T'}(\mathtt{gen}_1,\mathtt{gen}_2|S,T)\right]$$
$$(\because \mathop{\mathbb{E}}_{T'}[\mathtt{gen}_1|S,T] = 0)$$
$$\leqslant \mathop{\mathbb{E}}_{S,T}\left[\sqrt{\operatorname*{\mathbf{var}}_{T'}[\mathtt{gen}_1|S,T]\operatorname*{\mathbf{var}}_{T'}[\mathtt{gen}_2|S,T]}\right]$$
(By Cauchy–Schwarz)
$$\leqslant \sqrt{\mathop{\mathbb{E}}_{S,T}\left[\operatorname*{\mathbf{var}}_{T'}[\mathtt{gen}_1|S,T]\right]\mathop{\mathbb{E}}_{S,T}\left[\operatorname*{\mathbf{var}}_{T'}[\mathtt{gen}_2|S,T]\right]}$$
(By Cauchy–Schwarz)
$$= \sqrt{\mathop{\mathbb{E}}_{S,T}\left[\operatorname*{\mathbf{var}}_{T'}[\mathtt{gen}_2|S,T]\right]\operatorname*{\mathbf{var}}_U[\mathtt{gen}_1]}. \tag{5}$$

The final equality follows by the law of total variance:

$$\operatorname*{\mathbf{var}}_U(\mathtt{gen}_1) = \mathop{\mathbb{E}}_{S,T}[\operatorname*{\mathbf{var}}_{T'}[\mathtt{gen}_1|S,T]] + \operatorname*{\mathbf{var}}_{S,T}[\mathop{\mathbb{E}}_{T'}[\mathtt{gen}_1|S,T]]$$
$$= \mathop{\mathbb{E}}_{S,T}[\operatorname*{\mathbf{var}}_{T'}[\mathtt{gen}_1|S,T]].$$

It only remains to bound $\mathbb{E}_{S,T}[\mathbf{var}_{T'}[\mathtt{gen}_2|S,T]]$ appropriately.

To do this we use Steele's inequality. Fix S, T, and define the function $F : \mathcal{Z}^n \to \mathbb{R}$ as $F(T') = \mathtt{gen}_{\mathcal{A}(S\cup T')}(T) = \mathtt{gen}_2$. We use the notation $\tilde{T}'$ to represent the data set obtained from T by replacing a random element in it with a new independently sampled example z'. By Steele's inequality, we have

$$\frac{2}{n}\cdot\mathop{\mathbb{E}}_{S,T}\left[\operatorname*{\mathbf{var}}_{T'}[\mathtt{gen}_2|S,T]\right]$$
$$\leqslant \mathop{\mathbb{E}}_{S,T}\left[\mathop{\mathbb{E}}_{T',z'}[(F(T')-F(\tilde{T}'))^2 \mid S,T]\right]$$
$$= \mathop{\mathbb{E}}_{S,T,T',z'}\left[\left(\left(\frac{1}{n}\sum_{z\in T}\ell_{\mathcal{A}(S\cup T')}(z) - \bar{\ell}_{\mathcal{A}(S\cup T')}\right)\right.\right.$$
$$\left.\left. - \left(\frac{1}{n}\sum_{z\in T}\ell_{\mathcal{A}(S\cup\tilde{T}')}(z) - \bar{\ell}_{\mathcal{A}(S\cup\tilde{T}')}\right)\right)^2\right]$$
$$= \mathop{\mathbb{E}}_{S,T',z'}\left[\mathop{\mathbb{E}}_{T}\left[\left(\left(\frac{1}{n}\sum_{z\in T}\ell_{\mathcal{A}(S\cup T')}(z) - \bar{\ell}_{\mathcal{A}(S\cup T')}\right)\right.\right.\right.$$
$$\left.\left.\left. - \left(\frac{1}{n}\sum_{z\in T}\ell_{\mathcal{A}(S\cup\tilde{T}')}(z) - \bar{\ell}_{\mathcal{A}(S\cup\tilde{T}')}\right)\right)^2 \;\middle|\; S,T',z'\right]\right]$$

$$= \mathop{\mathbb{E}}_{S,T',z'}\left[\operatorname*{\mathbf{var}}_{T}\left[\left(\frac{1}{n}\sum_{z\in T}\ell_{\mathcal{A}(S\cup T')}(z)\right.\right.\right.$$
$$\left.\left.\left. -\frac{1}{n}\sum_{z\in T}\ell_{\mathcal{A}(S\cup\tilde{T}')}(z)\right) \;\middle|\; S,T',z'\right]\right],$$

because $\mathbb{E}_T\left[\frac{1}{n}\sum_{z\in T}\ell_{\mathcal{A}(S\cup T')}(z)\right] = \bar{\ell}_{\mathcal{A}(S\cup T')}$.

Continuing, since the examples in T are i.i.d., we get that the RHS of above equals (for a new randomly

chosen example z)

$$\begin{aligned}
&\mathop{\mathbb{E}}_{S,T',z'}\left[\frac{1}{n}\mathop{\mathbf{var}}_{z}\left[\left(\ell_{\mathcal{A}(S\cup T')}(z)-\ell_{\mathcal{A}(S\cup\tilde{T}')}(z)\right)\;\middle|\;S,T',z'\right]\right]\\
&\leqslant \mathop{\mathbb{E}}_{S,T',z'}\left[\frac{1}{n}\mathop{\mathbb{E}}_{z}\left[\left(\ell_{\mathcal{A}(S\cup T')}(z)-\ell_{\mathcal{A}(S\cup\tilde{T}')}(z)\right)^2\;\middle|\;S,T',z'\right]\right]\\
&= \frac{1}{n}\cdot\mathop{\mathbb{E}}_{S,T',z',z}\left[\left(\ell_{\mathcal{A}(S\cup T')}(z)-\ell_{\mathcal{A}(S\cup\tilde{T}')}(z)\right)^2\right]\\
&\leqslant \frac{\beta}{n}
\end{aligned}$$

where the last inequality follows because the mean-square stability of $\mathcal{A}$ is β. This, combined with inequalities (2) and (5), gives us the desired variance reduction result. □

5 Discussion of the variance reduction bound

As we mentioned before, in a previous work, Blum, Kalai, and Langford [2] showed that $\mathbf{var}_U[\mathtt{gen}_{\mathtt{cv}}] \leqslant \mathbf{var}_U[\mathtt{gen}_1]$. This falls out of our analysis as well, see (3). They also gave some very mild conditions under which the inequality is strict.

Given this bound, we now wish to understand situations where our results along with the assumptions on the mean-square stability give better quantitative variance reduction bounds. Generally to get a quantitative variance reduction bound from Theorem 2, we must get a handle on $\mathbf{var}_U[\mathtt{gen}_1]$. First, note that

$$\begin{aligned}
\mathop{\mathbf{var}}_{U}[\mathtt{gen}_1] &= \mathop{\mathbb{E}}_{S,T}[\mathop{\mathbf{var}}_{T'}[\mathtt{gen}_1|S,T]]\\
&= \frac{1}{n}\mathop{\mathbb{E}}_{S,T}[\mathop{\mathbf{var}}_{z}[\ell_{\mathcal{A}(S\cup T)}(z)-\bar{\ell}_{\mathcal{A}(S\cup T)}]]\\
&= \frac{1}{n}\mathop{\mathbf{var}}_{U,z}[\ell_{\mathcal{A}(S\cup T)}(z)].
\end{aligned}$$

We distinguish between two cases: the noisy setting, where the above quantity is bounded away from 0, and the noise-free (or realizable) setting.

Definition 6 (Volatility). *An instance of the cross-validation problem composed of the algorithm $\mathcal{A}$, the loss function ℓ, and the distribution $\mathcal{D}$ is δ-volatile* if $\mathop{\mathbf{var}}_{U,z}[\ell_{\mathcal{A}(S\cup T)}(z)] = \Omega(\delta)$.

We note that the $\Omega(\cdot)$ notation is allowed to have dependence on complexity parameters of the hypothesis space (such as the VC dimension) as long as they are fixed and not growing with m, the number of samples chosen.

5.1 Variance reduction under the noisy setting

In this setting we consider instances that are δ-volatile for some $\delta > 0$. This is a mild assumption, and easily satisfied under various models of noise that corrupt the label y of an example (or its feature vector x). Consider the classification setting, where the loss of a hypothesis h on an example $z = (x, y)$ is 0 if $h(x) = y$ and 1 otherwise. If the labels of every example are flipped due to noise with some constant probability, then the setting is $\Omega(1)$-volatile. Similarly, in the linear regression setting, if examples (x, y) are generated as $y = w^\star \cdot x + \varepsilon$, where $w^\star$ is an unknown parameter vector, and ε is zero-mean noise (such as the unit Gaussian), then most loss functions such as squared loss, $(w \cdot x - y)^2$, lead to $\Omega(1)$-volatile instances.

The work of Bousquet and Elisseeff [3] and Kutin and Niyogi [7] showed that several learning algorithms satisfy notions of stability that are stronger than the mean-square stability. Using Lemma 1 and Theorem 2, we can directly translate those stability conditions into variance reduction bounds for the learning algorithms.

Near-optimal variance reduction from uniform stability. In their seminal work on stability and generalization bounds, Bousquet and Elisseeff [3] focused on uniform stability and proved that several learning algorithms are $O(1/m)$-uniform stable. By Lemma 1, all of these algorithms are $\beta = O(1/m^2)$-mean-square stable. We then get the following variance reduction bound from Theorem 2:

Lemma 3. *If $\mathcal{A}$ is $O(1/m)$-uniform stable, then in an $\omega(1/n)$-volatile setting,*

$$\mathop{\mathbf{var}}_{U}[\mathtt{gen}_{\mathtt{cv}}] \;\leqslant\; (1+o(1))\cdot\frac{1}{k}\mathop{\mathbf{var}}_{U}[\mathtt{gen}_1].$$

This shows the surprising result that the CV estimate is almost as tightly concentrated as the average of k holdout estimates of classifiers trained on *independent* data sets. This is the strongest general form of variance reduction that one can expect to get.

We highlight the fact that this bound holds for all k. For $\Omega(1)$-volatile settings, the $o(1)$ term grows as $O(1/\sqrt{n})$ and allows us to achieve almost optimal variance reduction even for small k such that $3, 4$, etc., a setting that is widely used in practice.

Bousquet and Elisseeff [3] have shown that many regularized learning algorithms are $O(1/m)$-uniform

stable. Such algorithms include (see [3] for details) risk minimization in Hilbert spaces with Tikhonov regularization (bounded SVM regression, regularized least-squares regression, etc.), and risk minimization with relative entropy regularization. Note that the loss functions used here are regularized losses (see [3]), not the $\{0,1\}$ loss.

Variance reduction from weak-hypothesis or weak-L_1 stability. Uniform stability is a very restrictive concept; most classification algorithms can only have the trivial uniform stability of M. For this reason, weaker notions of stability such as weak hypothesis and weak-L_1 stability were defined and algorithms shown to satisfy these weaker notions. For instance, Devroye and Wagner [5] show that t-local rules (such as t-nearest neighbor) are $(O(\sqrt{t}/m), O(\sqrt{t}/m))$-weak-$L_1$ stable. Lemma 1, implies that an $(O(1/m), O(1/m))$-weak-L_1 stable algorithm is $(\beta = O(1/m))$-mean-square stable. We get the following variance reduction bound from Theorem 2.

Lemma 4. *If $\mathcal{A}$ is either $(O(1/m), O(1/m))$-weak-hypothesis or weak-L_1 stable, then in an $\Omega(1)$-volatile setting,*

$$\underset{U}{\mathbf{var}}[\mathtt{gen}_{\mathtt{cv}}] \;\leqslant\; \frac{O(1)}{\sqrt{k}} \underset{U}{\mathbf{var}}[\mathtt{gen}_1].$$

This variance reduction bound shows that the cross-validation estimate is almost as tightly concentrated as the average of approximately $\sqrt{k}$ holdout estimates of classifiers trained on independent data sets. Thus, this reduction is not as strong as the one we get via uniform stability. Furthermore, the $O(1)$ constant in the variance reduction bound depends on the problem parameters (although it is fixed even if m grows); hence, we get variance reduction only for moderately large k.

5.2 Variance reduction in the noise-free setting

In the noise-free (or realizable) setting, examples are generated to be consistent with a fixed hypothesis $h_0 : \mathcal{X} \to \mathcal{Y}$. We assume now that $\mathcal{Y} = \{0,1\}$, and we have the $\{0,1\}$-loss, i.e., if $z = (x,y)$ is an example, and h is a hypothesis, then $\ell_h(z) = |h(x) - y|$.

These instances may be 0-volatile and the results above would not apply. However, if the hypothesis class $\mathcal{H}$ has low complexity (measured by its VC-dimension), then the empirical risk minimization (ERM) procedure converges to a good hypothesis extremely fast. This yields some form of variance reduction as well; this time with additive error (over the average of k independent holdout estimates) rather than multiplicative.

Kutin and Niyogi [7, Corollary 7.4] show that if the VC dimension of $\mathcal{H}$ is finite, then ERM over $\mathcal{H}$ is $(0, e^{-\Omega(m)})$-CV stable[2], which translates it to being $O(e^{-\Omega(m)}, e^{-\Omega(m)})$-weak-$L_1$ stable using their Theorem 5.9 [7] with $\alpha(m) = e^{-\Omega(m)}$. Lemma 1 implies that the algorithm has mean-square stability $e^{-\Omega(m)}$. Then, using Theorem 2, we get the following variance reduction bound:

Lemma 5. *Let VC-dimension of $\mathcal{H}$ be finite. If $\mathcal{A}$ does ERM over $\mathcal{H}$, then we have*

$$\underset{U}{\mathbf{var}}[\mathtt{gen}_{\mathtt{cv}}] \;\leqslant\; \frac{1}{k} \underset{U}{\mathbf{var}}[\mathtt{gen}_1] + \sqrt{e^{-\Omega(m)} \underset{U}{\mathbf{var}}[\mathtt{gen}_1]}.$$

Let us interpret the various quantities in this bound intuitively to get an understanding of how much variance reduction we obtain. Since we are in the realizable setting, ERM over $S \cup T$ returns a hypothesis $\mathcal{A}(S \cup T)$ that has no training error. Let d be the VC-dimension of $\mathcal{H}$. By Vapnik's uniform convergence theorem [10], $\mathcal{A}(S \cup T)$ is expected to have error rate about $\bar{\ell}_{\mathcal{A}(S \cup T)} \approx O(\sqrt{d \ln(m/d)/m})$, with very high probability. Since we have $\{0,1\}$-loss, $\mathbf{var}_z[\ell_{\mathcal{A}(S \cup T)}(z) | S, T] = \bar{\ell}_{\mathcal{A}(S \cup T)}(1 - \bar{\ell}_{\mathcal{A}(S \cup T)}) \approx O(\sqrt{d \ln(m/d)/m})$, and hence $\mathbf{var}_U[\mathtt{gen}_1] \approx O(\frac{\sqrt{d \ln(m/d)}}{n\sqrt{m}})$. In comparison, the second term on the RHS of Lemma 5 is negligibly small. The overall effect is again of averaging k independent holdout estimates.

We note here that finite VC-dimension is not always necessary for the bound of Lemma 5 to hold. Kutin and Niyogi [7] give an example (see Example 9.18) of a language learning algorithm whose hypothesis class has infinite VC-dimension, and yet it has strong (and hence, weak) hypothesis stability $O(0, e^{-\Omega(m)})$. This gives the same variance reduction bound as in Lemma 5.

6 Conclusions and open problems

In this paper, we defined the new notion of mean-squared stability and showed that for mean-square

[2] Not defined in this paper, refer to [7] for the definition.

stable learning algorithms, the variance of the k-fold cross-validation estimate drops dramatically compared to the variance of the holdout estimate. We showed that many widely used learning algorithms have the requisite stability to almost achieve the optimal factor k reduction in variance. Our results hold for small values of k as well, justifying the wide use of k-fold cross-validation in practice, and lending theoretical support to conventional wisdom.

Several questions remain open. Is our notion of mean-square stability the most general one that can be used to prove near-optimal variance reduction bounds? There is evidence that the weaker notion of weak-error stability of Kearns and Ron is too weak to imply near-optimal variance reduction, but a formal proof is lacking. Similarly, there is evidence that the $O(\sqrt{\beta\,\mathbf{var}_U[\mathtt{gen}_1]})$ term in the bound of Theorem 2 can be tightened further. Both of these improvements would be significant steps towards finding the necessary and sufficient stability notion for variance reduction. Another open question is whether it is possible to obtain similar reduction for higher moments. Such a reduction would be useful in obtaining strong tail bounds on the tightness of the cross-validation estimate.

Finally, for unstable algorithms, bagging [4] has been shown to markedly improve algorithm performance. Extending the techniques in this work to unravel the dependencies created by bagging predictors remains an interesting open problem.

References

[1] M. Anthony and S. B. Holden.Cross-validation for binary classification by real-valued functions: theoretical analysis. In *Proc. 11th COLT*, pages 218-229, 1998.

[2] A. Blum, A. Kalai, and J. Langford.Beating the hold-out: Bounds for -fold and progressive cross-validation. In *Proc. 12th COLT,* pages 203-208, 1999.

[3] O. Bousquet and A. Elisseeff. Stability and generalization. *JMLR*, 2:499-526, 2002.

[4] L. Breiman.Bagging predictors. *Mach. Learn.*, 24(2):123-140, 1996.

[5] L. P. Devroye and T. J. Wagner.Distribution-free performance bounds. *IEEE TOIT,* 25:601-604, 1979.

[6] M. J. Kearns and D. Ron.Algorithmic stability and sanity-check bounds for leave-one-out cross-validation. *Neural Computation,* 11(6):1427-1453, 1999.

[7] S. Kutin and P. Niyogi.Almost-everywhere algorithmic stability and generalization error. Technical Report TR-2002-03, University of Chicago, Computer Science Department, 2002.

[8] W.Rogers and T. Wagner. Afinite sample distribution-free performance bound for local discrimination rules. *Annals of Statistics,* 6(3):506-514, 1978.

[9] J.M. Steele. An Efron-Stein inequality for nonsymmetric statistics. *Annals of Statistics*, 14(2):753-758, 1986.

[10] V. Vapnik.*Estimation of Dependences Based on Empirical Data*. Springer-Verlag, 1982.