

A Permutation Approach to Validation*

Malik Magdon-Ismail
magdon@cs.rpi.edu

Computer Science Department
Rensselaer Polytechnic Institute
110 8th Street, Troy, NY 12180, USA

Konstantin Mertsalov
mertsk2@cs.rpi.edu

Computer Science Department
Rensselaer Polytechnic Institute
110 8th Street, Troy, NY 12180, USA

February 8, 2011

Abstract

We give a permutation approach to validation (estimation of out-sample error). One typical use of validation is model selection. We establish the legitimacy of the proposed permutation complexity by proving a uniform bound on the out-sample error, similar to a VC-style bound. We extensively demonstrate this approach experimentally on synthetic data, standard data sets from the UCI-repository, and a novel diffusion data set. The out-of-sample error estimates are comparable to cross validation (CV); yet, the method is more efficient and robust, being less susceptible to overfitting during model selection.

1 Introduction

The holy grail when learning from data is an in-sample estimate of the out-sample error, i.e. *model validation*. Assume a standard setting with data

$$D = \{\mathbf{x}_i, y_i\}_{i=1}^n,$$

where (\mathbf{x}_i, y_i) are sampled *iid* from the joint distribution $p(\mathbf{x}, y)$ on $\mathbb{R}^d \times \mathbb{R}$ (for regression) or on $\mathbb{R}^d \times \{\pm 1\}$ (for binary classification). Let \mathcal{H} be a learning model (e.g. decision tree, k -nearest neighbor, linear regression) which produces a hypothesis $g \in \mathcal{H}$ when given D . Our discussion, though mostly generalizable, will assume that \mathcal{H} is closed under negation, and that the risk is the squared error (the misclassification error rate and the least squares regression error can both be written as the squared error). Denote by e_{in} the in-sample error,

$$e_{\text{in}}(h) = \frac{1}{n} \sum_{i=1}^n (h(\mathbf{x}_i) - y_i)^2,$$

and by e_{out} the out-sample error,

$$e_{\text{out}}(h) = \mathbb{E}[(h(\mathbf{x}) - y)^2].$$

*A preliminary version of this paper appears in the Siam Data Mining Conference, 2010

The expectation is over the joint distribution $p(\mathbf{x}, y)$. We wish to estimate $e_{\text{out}}(g)$, and typically $e_{\text{in}}(g)$ is not an unbiased estimate of e_{out} . For example, when g minimizes the in-sample error over \mathcal{H} , then for a small data set, e_{in} will typically have a large optimistic bias precisely because you are minimizing e_{in} . Instead of e_{out} , it is equally good to get an estimate of the *generalization error*¹

$$e_{\text{gen}}(g) = e_{\text{out}}(g) - e_{\text{in}}(g),$$

which is typically positive and explicitly measures the optimism of the in-sample error.

Our goal is to present a method for estimating the generalization error, in particular, we present a permutation estimate for the generalization error. Loosely speaking, it considers learning problems related to permutations of the realized data. These permutation problems can be explicitly generated from the realized data set. For each permutation-transformed problem, we compute the expected optimism of the in-sample error. The average of this optimism over permutations is the permutation estimate for $e_{\text{gen}}(g)$. To make the idea more concrete, we will illustrate on a toy vision problem of learning to distinguish between male and female faces, using a labeled data set of images. Assume some algorithm which produces a classifier, given a data set. A small data set is shown in Figure 1. After learning on the data, suppose the learned hypothesis (hypothetically)

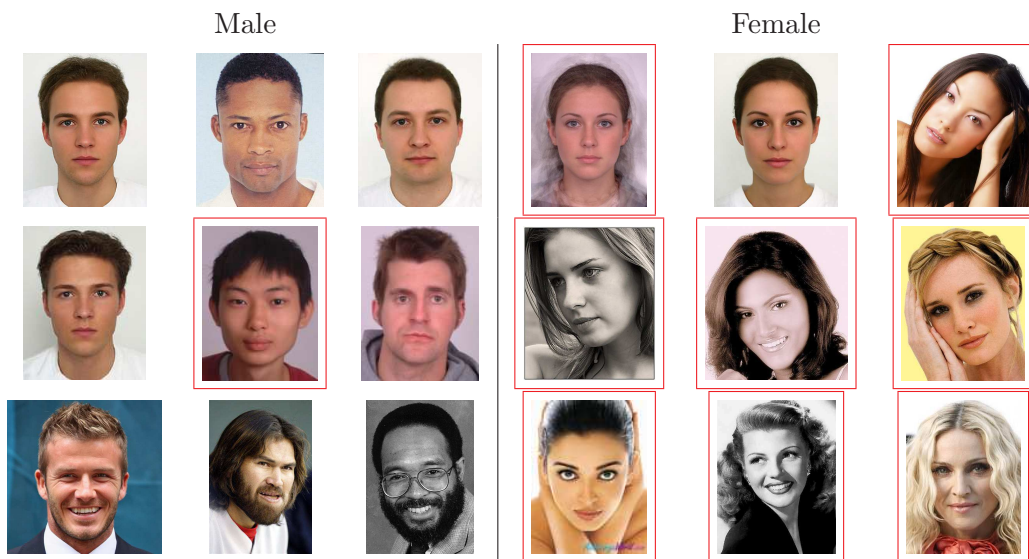


Figure 1: A sample data set for learning to distinguish male from female faces.

classified the images in red boxes as female (in this example, the learned rule is “roundish face or long hair is female”) – after learning the rule, you can classify the in-sample data. If your rule were perfect in-sample, all the faces on the right would be boxed in red, and none of the faces on the left would be boxed. This rule is not quite perfect, and your in-sample classification error on this example data set is about 11% (2 errors). How reliable is this error of 11%? Do we expect that this rule will generalize well to out-sample, and achieve $e_{\text{out}} \approx 11\%$?

The permutation estimate would provide one estimate of the reliability. To apply the permutation estimate, we first generate a random permutation of the data, i.e. permute the labels (male or

¹some authors use generalization error to denote what we call the out-sample error.

female) randomly, to obtain a permuted data set. One such realized random permutation is shown in Figure 2. This data with the randomly permuted labels is a very confusing data set. Clearly one

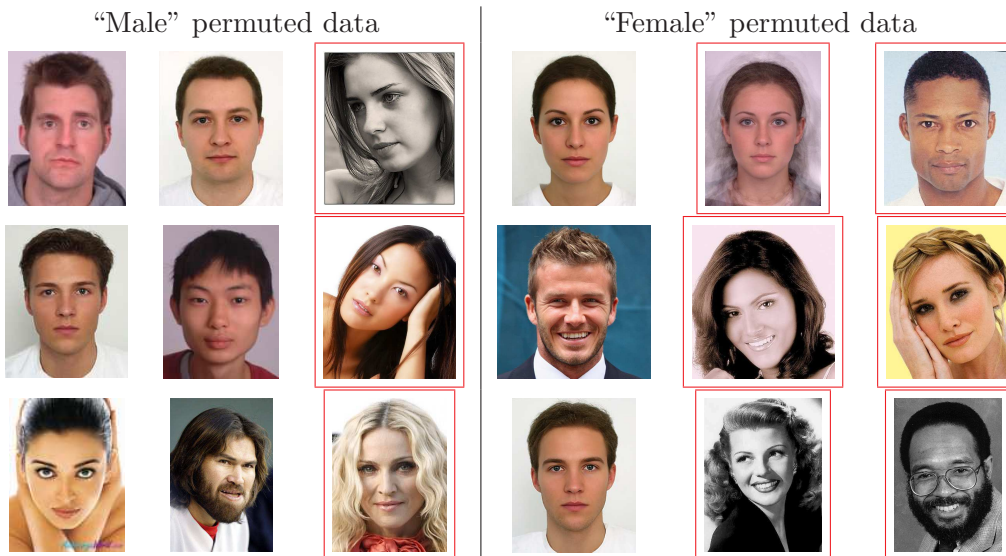


Figure 2: A permuted version of the same sample data in Figure 1. Learning on this data set should not yield anything meaningful.

should not expect better than 50% performance from any algorithm out-of-sample, if the real data distribution was like this permuted data. Nevertheless, we can go ahead and attempt to learn with the same learning algorithm. Since, in-sample, your learning algorithm is fitting the data, it may overfit even this randomly permuted data. In our example, suppose the new learned rule is “dark skin or long hair is female”; this gives the in-sample classifications of females shown above (red boxes). In this case, on the randomly permuted data, we have made 6 errors (the 3 boxed images on the left and three un-boxed images on the right), or roughly 33% classification error rate. As we already argued, for such randomly permuted data distribution, no algorithm can give out-sample error better than 50% (assuming that there are equal numbers of male and female faces), and so this achieved in-sample error rate of 33% is optimistic by about 17%. That is, the learning algorithm has overfit the permuted data by 17% (the generalization error is 17%). This level of overfitting is a single sample estimate of the overfitting capability of the model on data “of this type” – for another type of data, say digit classification, the level of overfitting on permuted data could be different. Thus, this permutation estimate of generalization error is *data dependent*, and certainly depends on the complexity of the model, and the learning algorithm. It seems reasonable to guess that this same level of overfitting might have affected the actual fitting on the actual data, since the actual data is, at least on the surface, of this “type”. We thus apply this 17% optimism of the in-sample error obtained from looking at the permuted data to the learning on the unpermuted data; we conclude that our original classifier “roundish face or long hair is female”, which had in-sample error of 11% would have an out-sample error more like 28%. The goal of this paper is to first define the permutation estimate for both regression and classification; provide some theoretical justification for its use, and experimentally compare it with some other methods for validation.

Our empirical comparisons will use leave-one-out cross validation, LOO-CV, as the strawman

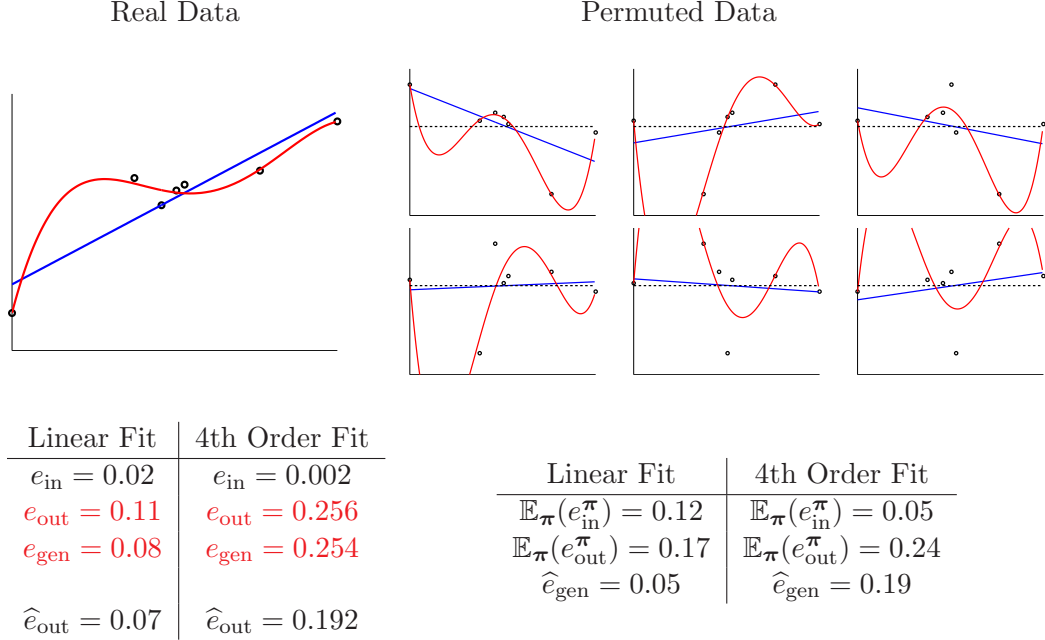


Table 1: On the left, a linear and 4th order fit to some data. We estimate e_{in} and e_{gen} and e_{out} are not available. On the right, 6 example permuted data sets. For the permuted data, we can also get e_{in} , but now we can also compute e_{out} (see Theorem 1); thus, for the permuted data distribution, we can estimate \hat{e}_{gen} ; we use this estimate, together with e_{in} on the real data to estimate e_{out} on the real data.

benchmark. This is a general validation method, which is in common use. It has also been compared with most other validation methods, and hence is a valid benchmark. Our permutation estimate (as with LOO-CV) can be applied to any learning model or error metric, requiring only the ability to run the model. However, it is more efficient than LOO-CV and suffers less from the potential to be overfit during model selection, especially in regression.

1.1 Our Results.

We give a permutation estimate for validation. Validation is one of the most important tasks when learning from data. We quantitatively illustrate the permutation estimate using a regression problem in Table 1 using a linear versus a 4th order polynomial model. The algorithm begins by fitting the real data to obtain e_{in} . Next one permutes the y values (shown are 6 permutations) using a random permutation π ; one then uses each model to fit the permuted data sets, to compute the average in-sample error and the average out-sample error (Theorem 1 shows how to compute the average out-sample error for the permuted data distribution). The permutation estimate $\hat{e}_{\text{gen}} = \mathbb{E}_{\pi}[e_{\text{out}}^{\pi} - e_{\text{in}}^{\pi}]$ (the difference between the average in-sample and out-sample errors on the permuted data). The out-sample error estimate is then $\hat{e}_{\text{out}} = e_{\text{in}} + \hat{e}_{\text{gen}}$. In this particular instance, model selection with respect to \hat{e}_{out} would select the linear model over the quartic model, and it would be correct. We show empirically that this method works well, and is superior to other methods for model selection, in particular the leave-one-out method.

Corresponding to the permutation estimate, we define a “permutation complexity” measure for

the complexity of a learning model, which is data dependent:

$$\mathcal{P}_{\text{in}}(\mathcal{H}|D) = \mathbb{E}_{\boldsymbol{\pi}} \left[\max_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n y_{\pi_i} h(\mathbf{x}_i) \right].$$

Here, $\boldsymbol{\pi}$ is a random permutation on $[1, n]$. We theoretically justify the permutation estimate in the context of empirical error minimization in classification problems by proving a *uniform bound* for the out-sample error in terms of this *data dependent* permutation complexity. Specifically, we prove that for the empirical risk minimizer g , with probability at least $1 - \delta$,

$$e_{\text{out}}(g) \leq e_{\text{in}}(g) + 2\widehat{e}_{\text{gen}}(\mathcal{H}|D) + 4\bar{y} \mathbb{E}_{\boldsymbol{\pi}} [\bar{g}^{\boldsymbol{\pi}}] + O\left(\sqrt{\frac{1}{n} \ln \frac{1}{\delta}}\right).$$

For balanced distributions ($\bar{y} = O\left(\sqrt{\frac{1}{n} \ln \frac{1}{\delta}}\right)$), and so this result provides direct theoretical justification for the permutation estimate. When a distribution is not balanced, the theoretical bound is weaker, nevertheless we have found that the permutation estimate still works well empirically. This is because for the empirical risk minimizer on a permutation, $g^{\boldsymbol{\pi}}$ is attempting to fit the permuted data and so $\bar{g}^{\boldsymbol{\pi}} \approx \bar{y}$; this term is then approximately \bar{y}^2 which is close to a (small) constant for all the models, and hence will not significantly affect the model selection. From the practical, implementation point of view, we show that the permutation estimate (expectation over permutations) concentrates around its expectation, meaning that a few random permutations suffice to compute it. To our knowledge, this is the first theoretical result within this setting for a dependent sampling process (both for the generalization bound and for the concentration result). The novel aspects of the proof technique are the introduction of a second ghost data set to handle the dependent sampling in the permutation complexity, and the linking of the permutation estimate to the bootstrap estimate to obtain the concentration result; to our knowledge, these are the first use of such techniques.

On the practical side, we give a detailed experimental investigation to support the permutation estimate. We show that it is more efficient than the leave-one-out cross validation method, with comparable or better performance. We use LOO-CV as our strawman benchmark, since most validation methods have been compared to this benchmark, including the leave-K-out methods, and other statistical estimators. leave-K-out methods are also more efficient than LOO CV, but they will sacrifice on accuracy.

Note to Practitioners. The theoretical bound applies only to classification, and is “loose” in the sense that there is an factor of 2 multiplying \widehat{e}_{gen} , together with the \bar{y} bias. The concentration result, however, has real practical significance. It essentially says that a very small number of learning episodes (on different permutations) suffices to get a good estimate of \widehat{e}_{gen} . Contrast this with K -fold cross validation where one needs K episodes of learning on data sets of size $n(K-1)/K$. Ideally, one wants to choose K as large as is computationally feasible, and typically, computation is a real bottle neck, which means that in practice, one often never has the luxury of running LOO-CV. In practice, probably 10-fold validation is the most commonly used.

The existence of the uniform bound is comforting, however it shouldn't be taken too literally². Though the bound is loose, it is found in practice that criteria for which uniform bounds are available (eg. VC-bound, permutation estimate, Rademacher complexity) work well for model selection, even if these bounds are weak. This highlights the role of variance versus bias when it comes to model selection. A model selection criterion which has a large but systematic bias (with respect to the true out-sample error), but very low variance is perfectly fine; in fact all one needs is that the model selection criterion be monotonic in the true out-sample error, and have low variance. On the other hand, a model selection criterion which has very low bias, but very high variance can lead to serious problems, especially when one is selecting among many models; by chance, a very bad model can be selected due to the large variance. In practice, it is often found that criteria, for which one can obtain uniform bounds that hold for all data distributions, tend to display significant bias, but with low variance; this bias also tends to be somewhat systematic across different models, and so the decreased variance more than compensates for the large bias, when it comes to model selection.

Our experimental results will illustrate this general bias-variance tradeoff for model selection. LOO-CV has low bias, but its relatively high variance causes serious problems in practice, resulting in excellent performance for out-sample error estimation, but poor performance for model selection. On the other hand, the permutation based method tends to have a relatively higher bias, but this bias is systematic across models (so it does not affect the model selection), and the variance is much lower. This much lower variance (with potentially higher but generally systematic bias), when coupled with the fact that the permutation estimate is more computationally efficient (via the concentration result) means that the permutation estimate is a serious alternative to LOO-CV for model selection in both regression and classification. If bias were the only concern (for example, when it comes to out-sample error estimation for a single model), it is very hard to beat the LOO-CV estimate (assuming computation is not an issue).

Paper Outline. Next, we review some relevant literature, followed by a description of the permutation method and the uniform bound which justifies its use. We then give a detailed experimental investigation and conclusions.

1.2 Related Work

Out-sample error estimation has extensive coverage in the literature, both in the statistics community and the learning community. Broadly speaking there are three approaches:

(i) *Statistical methods* which try to estimate the out-sample error asymptotically in n , giving consistent estimates under certain model assumptions, for example: final prediction error (FPE) (Akaike, 1974); Generalized Cross Validation (GCV) (Craven and Wahba, 1979); or, covariance-type penalties (Efron, 2004; Wang and Shen, 2006). Statistical methods tend to work well when the model has been well specified. Such methods are not our primary focus. However, we do show that for linear models in the statistical regression setting, the permutation estimate reduces to an AIC-type prediction error estimate with the noise estimated by the in-sample variance in the targets.

²A similar comment would apply to the VC-generalization bound: it is very loose.

(ii) *Bounds.* The most celebrated uniform bound on generalization error is the distribution independent bound of Vapnik-Chervonenkis (VC-bound) of Vapnik and Chervonenkis (1971). Since the VC-dimension may be hard to compute, empirical estimates have been suggested (Vapnik *et al.*, 1994). The VC-bound is optimal among distribution independent bounds; however, for a particular distribution, it could be sub-optimal. Several data dependent bounds have been proposed, which can typically be estimated in-sample via optimization: maximum discrepancy (Bartlett *et al.*, 2002); Rademacher-style penalties (Bartlett and Mendelson, 2002; Fromont, 2007; Kääriäinen and Elomaa, 2003; Koltchinskii, 2001; Koltchinskii and Panchenko, 2000; Lozano, 2000; Lugosi and Nobel, 1999; Massart, 2000); margin based bounds, for example (Shawe-Taylor *et al.*, 1998). Relevant to this work are Rademacher penalties. Let \mathbf{r} be a sequence of *iid* binary random variables with $\mathbb{P}[+1] = \frac{1}{2}$. The Rademacher complexity is

$$\mathcal{R}(\mathcal{H}|D) = \mathbb{E}_{\mathbf{r}} \left[\frac{1}{n} \max_{h \in \mathcal{H}} \sum_{i=1}^n r_i h(\mathbf{x}_i) \right].$$

Generalizations to Gaussian and symmetric, bounded variance \mathbf{r} have also been suggested, (Bartlett and Mendelson, 2002; Fromont, 2007). The permutation estimate is related to a “permutation complexity” which is a Rademacher-like complexity where the r_i are obtained via a random permutation of the observed target values in D , instead of via independent uniform Bernoulli trials. Thus, the permutation estimate more closely mimics the distribution supported by the data. It also has the advantage that the estimate can be used with multi-class and regression problems, and can accomodate regularized learning algorithms.

(iii) *Sampling methods*, such as leave-one-out cross validation (LOO-CV), try to estimate the out-sample error directly. Cross validation is perhaps the most used validation method, dating as far back as 1931 (Larson, 1931; Wherry, 1931, 1951; Katzell, 1951; Cureton, 1951; Mosier, 1951; Stone, 1974). The permutation estimate uses a “sampled” data set on which to run the model and obtain the estimate; other than this superficial similarity, the estimates are inherently different. The permutation estimate is more like a Rademacher penalty in spirit.

Permutation Methods are not new to statistics (Good, 2005). They have been suggested as tests of significance for specific model selection tasks (Golland *et al.*, 2005; Wiklund *et al.*, 2007). In (Golland *et al.*, 2005), the authors give a concentration inequality for such a test which involves the Rademacher complexity. We directly give a uniform bound for the out-sample error in terms of a permutation complexity, which answers a question posed in Golland *et al.* (2005), where the authors suggest that there should be a direct link between permutation statistics and generalization errors. There have also been some early studies of using permutation tests for model validation, though none are specifically of our form, and there are no uniform bounds on out-sample error proven (Lindgren *et al.*, 1996; Golland and Fischl, 2003; Carmack *et al.*, 2002).

2 The Permutation Estimate

Consider a new learning problem, for which the input space is exactly the data examples, and the outputs are a random permutation of the observed target values. This problem mimics the learning problem at hand, in that the targets have the same joint distribution but for which there

is no input-output relationship. For this new permutation-learning problem, we can compute the “out-sample” error for a test example drawn from this same random permutation distribution. If we fit the model to the data, we obtain the in-sample error of the learned function which will generally be lower than this computed out-sample error. The expected difference between the in-sample error and the computed out-sample error for the learned function is the estimate of the optimistic bias. We use this optimistic bias as the estimate of the generalization error for the particular data set D (after fitting the model to it). There is a leap of faith here: we are using the average level of optimism for this random class of problems as the measure of optimism for the *single* actual realized problem. We will provide justification for this leap of faith by relating the permutation estimate (for binary classification) to a uniform upper bound on the out-sample error. We now describe the details.

In order to specify the random permutation learning problem, $p^\pi(\mathbf{x}, y)$, we will take an operational route. (We use the superscript $(\cdot)^\pi$ for quantities relevant to the random permutation learning problem.) Let

$$D^\pi = (\mathbf{x}_1, y_{\pi_1}), \dots, (\mathbf{x}_n, y_{\pi_n}),$$

be a random permutation of the data, where π is a random permutation of $1, \dots, n$. This is a new learning problem in which the \mathbf{x} values are unchanged but the target function f^π is a random function with $\mathbb{P}[f^\pi(\mathbf{x}_i) = y_j] = \frac{1}{n}$ for $j = 1, \dots, n$, independent of the particular \mathbf{x}_i . Though the target values are independent of \mathbf{x}_i , the target values at two different inputs $\mathbf{x}_i, \mathbf{x}_j$ are *not* independent. This target function has the same joint distribution of outputs on the data as the true target function, but otherwise it is independent of the input, and so there is no input-output relationship to be learned. Let g^π be the function output by the model (e.g. via empirical risk minimization). If the model “learns” a relationship, it has overfit the permuted data. In order to compute the level to which it has overfit the data, we need to first compute the out-sample error for this permuted learning problem.

2.1 Out-Sample Error for the Random Permutation Problem

The out-sample error of any function $h \in \mathcal{H}$ can be computed because we know the target function f^π , which is specified by $p^\pi(y|\mathbf{x})$ (uniform on $\{y_i\}$). Let \bar{y} and s_y^2 be the sample mean and sample variance of the target values.

Theorem 1.

$$e_{\text{out}}^\pi(h) = s_y^2 + \frac{1}{n} \sum_{i=1}^n (h(\mathbf{x}_i) - \bar{y})^2.$$

Proof.

$$\begin{aligned} e_{\text{out}}^\pi(h) &= \frac{1}{n} \sum_{i=1}^n \mathbb{E}_{p^\pi} [(h(\mathbf{x}_i) - f^\pi(\mathbf{x}_i))^2] \\ &= \frac{1}{n} \sum_{i=1}^n \frac{1}{n} \sum_{j=1}^n (h(\mathbf{x}_i) - y_j)^2. \end{aligned}$$

Adding and subtracting \bar{y} in the summand, we have:

$$\begin{aligned} e_{\text{out}}^{\pi}(h) &= \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n (h(\mathbf{x}_i) - \bar{y} + \bar{y} - y_j)^2 \\ &= s_y^2 + \frac{1}{n} \sum_{i=1}^n (h(\mathbf{x}_i) - \bar{y})^2. \end{aligned}$$

■

Notice that for this random learning problem, the best one can do is achieve an out-sample error of s_y^2 which is attained by the function which predicts the mean target value for all data points.

2.2 Estimating the Generalization Error

We now compute the bias of the in-sample error (typically obtained via empirical risk minimization) on a random permutation problem. On the permuted data, the in-sample error is:

$$e_{\text{in}}^{\pi}(g^{\pi}) = \frac{1}{n} \sum_{i=1}^n (g^{\pi}(\mathbf{x}_i) - y_i^{\pi})^2.$$

For $h \in \mathcal{H}$ let \bar{h} be the average value of h over the data set, $\bar{h} = \frac{1}{n} \sum_{i=1}^n h(\mathbf{x}_i)$. Since the generalization error is $e_{\text{gen}}^{\pi}(g^{\pi}) = e_{\text{out}}^{\pi}(g^{\pi}) - e_{\text{in}}^{\pi}(g^{\pi})$, we have the following result.

Theorem 2.

$$\begin{aligned} e_{\text{gen}}^{\pi}(g^{\pi}) &= \frac{2}{n} \sum_{i=1}^n (y_i^{\pi} - \bar{y}) g^{\pi}(\mathbf{x}_i) \\ &= \frac{2}{n} \sum_{i=1}^n y_i^{\pi} g^{\pi}(\mathbf{x}_i) - 2\bar{y}\bar{g}^{\pi}. \end{aligned}$$

Proof. Using Theorem 1,

$$e_{\text{gen}}^{\pi}(g^{\pi}) = s_y^2 + \frac{1}{n} \sum_{i=1}^n (g^{\pi}(\mathbf{x}_i) - \bar{y})^2 - \frac{1}{n} \sum_{i=1}^n (g^{\pi}(\mathbf{x}_i) - y_i^{\pi})^2,$$

and the result follows after some elementary algebra. ■

Note that e_{gen}^{π} is twice the covariance between the learned hypothesis and the randomly permuted target values. We wish to estimate $\mathbb{E}_{\pi}[e_{\text{gen}}^{\pi}(g^{\pi})]$. Naturally we have to do this by sampling. Unfortunately, y_i^{π} are dependent (sampled without replacement), and so it is not easy to obtain a concentration result around the expectation; nevertheless, it is possible (see Section 3.3). For *iid* sampling with replacement, it is immediate that the generalization error on the random problem concentrates about its expectation (via McDiarmid's Inequality). Thus, one can compute the generalization penalty for sampling with or without replacement using just a single random sample (asymptotically). Since one is always in the finite regime, we recommend to average over M random permutations, for some reasonably sized M – the real value of validation estimates is for small

n when overfitting is a real concern. We compared $M = 1000$ and $M = 10$ in our experimental results. The results were not significantly different from each other of from $M = 1$ when n , the number of data points, is large. Summarizing,

$$\widehat{e}_{\text{gen}}(\mathcal{H}|D) = \mathbb{E}_{\boldsymbol{\pi}}[e_{\text{gen}}^{\boldsymbol{\pi}}(g^{\boldsymbol{\pi}})] \approx \frac{1}{M} \sum_{m=1}^M e_{\text{gen}}^{\boldsymbol{\pi}_m}(g^{\boldsymbol{\pi}_m}). \quad (1)$$

(we use hat notation $\widehat{(\cdot)}$ for estimates of quantities relevant to the data D .) As with cross validation, the drawback with using more samples (larger M) is that one has to learn a final hypothesis for each permuted data set. Luckily, small M is enough for most practical purposes. The benefit over cross-validation is that for each iteration of training, one gets n estimates of the generalization error.

We now estimate the out-sample error for the distribution supported by the realized data D by

$$\widehat{e}_{\text{out}}(g) = e_{\text{in}}(g) + \widehat{e}_{\text{gen}}(\mathcal{H}|D). \quad (2)$$

This generalization error estimate is data dependent, because the randomly permuted learning problems were modeled after the original data set D . This is one of the advantages of such a method over distribution independent methods (such as VC) which may not be optimal for a particular problem.

2.2.1 Permutation Estimate for Linear Ridge Regression.

Ridge regression (regression with weight decay) is popular in statistics, and many statistical estimates of the out-sample error exist. Our permutation estimate closely resembles the AIC criterion (Akaike, 1974) in this setting.

A linear model has the form

$$\mathcal{H} = \left\{ \mathbf{w}^T \mathbf{z} \mid \mathbf{w} \in \mathbb{R}^{d+1} \right\},$$

where $\mathbf{z}^T = [1, \mathbf{x}^T]$ is the original input prepended by a 1 for the constant. Let $X^T = [\mathbf{z}_1, \dots, \mathbf{z}_n]$ be the input data matrix and $\mathbf{y}^T = [y_1, \dots, y_n]$ be the target values. Assume for simplicity that X has full column rank. If λ is the ridge regression parameter, then the optimal regularized fit minimizing $e_{\text{in}}(\mathbf{w}) + \lambda \mathbf{w}^T \mathbf{w}$ is given by

$$\mathbf{w}_{\text{in}} = (X^T X + \lambda I)^{-1} X^T \mathbf{y},$$

and the in-sample predictions are $\hat{\mathbf{y}} = S(\lambda) \mathbf{y}$, where,

$$S(\lambda) = X(X^T X + \lambda I)^{-1} X^T.$$

(We will typically suppress the dependence on λ .)

For linear models, we can obtain the permutation estimate without sampling, via an analytic computation. We want $\mathbb{E}_{\boldsymbol{\pi}}[e_{\text{gen}}^{\boldsymbol{\pi}}(g^{\boldsymbol{\pi}})]$. The predictions on the permuted data are $\hat{\mathbf{y}}^{(\boldsymbol{\pi})} = S \mathbf{y}^{(\boldsymbol{\pi})}$, where S is independent of the permutation $\boldsymbol{\pi}$ because it only depends on the \mathbf{x} values in the data.

Using Theorem 2:

$$e_{\text{gen}}^{\pi}(g^{\pi}) = \frac{2}{n} \sum_{i,j=1}^n S_{ij} (y_i^{\pi} y_j^{\pi} - \bar{y} y_i^{\pi}) \quad (3)$$

To compute $\mathbb{E}_{\pi}[e_{\text{gen}}^{\pi}(g^{\pi})]$ for Theorem 3, we will need the next result on the correlations.

Lemma 1.

$$\mathbb{E}_{\pi}[y_i^{\pi}] = \bar{y}, \quad \mathbb{E}_{\pi}[y_i^{\pi} y_j^{\pi}] = \begin{cases} \bar{y}^2 + s_y^2 & i = j, \\ \bar{y}^2 - \frac{1}{n-1} s_y^2 & i \neq j. \end{cases}$$

Proof. The first equality follows immediately because y_i^{π} is uniform over \mathbf{y} . To prove the second, note that

$$\begin{aligned} \mathbb{E}[y_i^{\pi}, y_j^{\pi}] &= \sum_{i=1}^n \sum_{j \neq i} \frac{1}{n(n-1)} y_i y_j, \\ &= \frac{1}{n(n-1)} \sum_{i=1}^n \left(\sum_{j=1}^n y_i y_j - y_i^2 \right), \\ &= \frac{n}{n-1} \bar{y}^2 - \frac{1}{n-1} \bar{y}^2, \end{aligned}$$

from which the result follows, because $s_y^2 = \bar{y}^2 - \bar{y}^2$. ■

Define $\hat{\sigma}_y^2 = \frac{n}{n-1} s_y^2$, the unbiased variance estimate for y , and let $\mathbf{1}$ be the n -vector of ones.

Theorem 3.

$$\hat{e}_{\text{out}}(g) = e_{\text{in}}(g) + \frac{2\hat{\sigma}_y^2}{n} \left(\text{trace}(\mathbf{S}) - \frac{\mathbf{1}^T \mathbf{S} \mathbf{1}}{n} \right).$$

Proof. Using (3),

$$\begin{aligned} \hat{e}_{\text{gen}}(\mathcal{H}|D) &= \mathbb{E}_{\pi}[e_{\text{gen}}^{\pi}(g^{\pi})] \\ &= \frac{2}{n} \sum_{i,j=1}^n S_{mn} (\mathbb{E}_{\pi}[y_i^{\pi} y_j^{\pi}] - \bar{y} \mathbb{E}_{\pi}[y_i^{\pi}]). \end{aligned}$$

The result follows using Lemma 1 after some algebra. ■

With no regularization ($\lambda = 0$), \mathbf{S} is a projection matrix (projecting onto the columns of \mathbf{X}). Since the first column of \mathbf{X} is a column of ones (representing a constant term in the regression), then $\mathbf{S} \mathbf{1} = \mathbf{1}$ and the permutation estimate becomes

$$\hat{e}_{\text{out}} = e_{\text{in}} + \frac{2\hat{\sigma}_y^2 d}{n}.$$

Thus, with no regularization, the permutation estimate reduces to an AIC-like criterion, and the term $2\hat{\sigma}_y^2 d/n$ is a penalty for model complexity. Observe that $(\text{trace}(\mathbf{S}) + 1 - \frac{1}{n} \mathbf{1}^T \mathbf{S} \mathbf{1})$ is an “effective number of parameters”. The choice $d_{\text{eff}} = \text{trace}(\mathbf{S})$ has been proposed in the literature (see for example Bishop (2006)).

Kernel Based Classification. For classification in a linear space, in general it is not possible to compute the empirical risk minimizer. However, there are many algorithms which could be used to approximate this; one is the support vector machine algorithm; another approach is to simply use the linear regression solution for classification, for which case the generalization estimate is

$$\widehat{e}_{\text{gen}} = \frac{2}{n} \mathbb{E}_{\boldsymbol{\pi}} \sum_{i=1}^n (y_i^{\boldsymbol{\pi}} - \bar{y}) \text{sign} \left(\sum_{j=1}^n S_{ij} y_j^{\boldsymbol{\pi}} \right).$$

Since the permutation estimate concentrates about its expectation, a few sample random permutations should suffice to compute \widehat{e}_{gen} .

2.3 Extensions

Sampling with Replacement. A simpler alternative to the permutation estimate (sampling the y_i without replacement) is to sample the y_i independently with replacement according to the bootstrap distribution B to generate a sampled data

$$D^B = \{(\mathbf{x}_i, y_i^B)\}.$$

Computationally, this is slightly simpler, but it does not preserve the joint distribution of the target values (they are now independent at two different points $\mathbf{x}_i, \mathbf{x}_j$). There are some advantages to this approach, for example one can obtain easy concentration results for e_{gen}^B about its expectation. This implies that even one random learning problem could be enough for estimating the optimism penalty (the concentration result for the permutation estimate is much harder to obtain).

Theorems 1 to 2 are unchanged, as they simply rely on $\mathbb{E}_B[y_i^B] = \bar{y}$. However, the derivation of the bootstrap estimate for linear ridge regression does change as now y_i and y_j are independent. Our entire discussion will generalize easily to this setting, and Theorem 3 for linear ridge regression becomes

$$e_{\text{out}}^B(g) = e_{\text{in}}(g) + \frac{2s_y^2 \text{trace}(\mathbf{S})}{n}.$$

Note that $\text{trace}(\mathbf{S})$ plays the role of an effective number of parameters. When $\lambda = 0$, $e_{\text{out}}^B(g) = e_{\text{in}}(g) + \frac{2s_y^2(d+1)}{n}$, again, similar to the AIC estimate.

Multi-Class. Suppose you have K classes c_1, \dots, c_K , so $y_i \in [1, K]$. Assume a loss matrix $L(y, y')$ which quantifies the loss of classifying class y' when the true class is y . So,

$$e_{\text{in}}(g) = \frac{1}{n} \sum_{i=1}^n L(y_i, g(\mathbf{x}_i)),$$

and

$$e_{\text{in}}^{\boldsymbol{\pi}}(g^{\boldsymbol{\pi}}) = \frac{1}{n} \sum_{i=1}^n L(y_i^{\boldsymbol{\pi}}, g^{\boldsymbol{\pi}}(\mathbf{x}_i)), \quad \text{and} \quad e_{\text{out}}^{\boldsymbol{\pi}}(g^{\boldsymbol{\pi}}) = \frac{1}{n^2} \sum_{i,j=1}^n L(y_j, g^{\boldsymbol{\pi}}(\mathbf{x}_i)).$$

The permutation estimate can be extended to different risk metrics for regression problems in an analogous way. The analogues of Theorems 1 and 2 would need to be computed; their form may

not be as simple, but nonetheless it is a straightforward task. It is not immediate how to extend the Rademacher penalty to regression, since, for example, the maximizer of covariance for a linear model is not well defined.

3 Uniformly Bounding Out-Sample Classification Error

For classification ($y \in \{\pm 1\}$), the estimate \widehat{e}_{gen} is closely related to a Rademacher-like “permutation complexity”. We now give a bound for the out-sample error using this permutation complexity for empirical risk minimization. We will adapt the standard ghost sample approach in VC-type proofs and the symmetrization trick in (Giné and Zinn, 1984) which has greatly simplified VC-style proofs. In general, high probability results are with respect to the distribution over data sets. Our main bounding tool will be McDiarmid’s inequality:

Lemma 2 (McDiarmid (1989)). *Let $X_i \in A_i$ be independent; suppose $f : \prod_i A_i \mapsto \mathbb{R}$ satisfies*

$$\sup_{\substack{\mathbf{x} \in \prod_i A_i \\ z \in A_j}} |f(\mathbf{x}) - f(x_1, \dots, x_{j-1}, z, x_{j+1}, \dots, x_n)| \leq c_j,$$

for $j = 1, \dots, n$. Then, for all $t > 0$,

$$\mathbb{P}[f(X_1, \dots, X_n) - \mathbb{E}f(X_1, \dots, X_n) \geq t] \leq e^{-2t^2 / \sum_{i=1}^n c_i^2}.$$

Corollary 1. *With probability at least $1 - \delta$,*

$$f(X_1, \dots, X_n) \leq \mathbb{E}f(X_1, \dots, X_n) + \sqrt{\frac{1}{2} \sum_{i=1}^n c_i^2 \ln \frac{1}{\delta}}.$$

Note that the reverse inequality $\mathbb{E}f \leq f + \sqrt{\frac{1}{2} \sum_{i=1}^n c_i^2 \ln \frac{1}{\delta}}$ can also be obtained by applying McDiarmid’s inequality to $-f$. So, by applying the union bound, with probability at least $1 - \delta$, $|f - \mathbb{E}f| \leq \sqrt{\frac{1}{2} \sum_{i=1}^n c_i^2 \ln \frac{1}{\delta}}$.

3.1 Permutation Complexity

The out-sample permutation complexity of a model is:

$$\mathcal{P}_{\text{out}}(\mathcal{H}) = \mathbb{E}_{D, \pi} \left[\max_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n y_i^\pi h(\mathbf{x}_i) \right],$$

where the expectation is over the data $D = (\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)$ and a random permutation π . For a particular sample D , the in-sample permutation complexity is

$$\mathcal{P}_{\text{in}}(\mathcal{H}|D) = \mathbb{E}_\pi \left[\max_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n y_i^\pi h(\mathbf{x}_i) \right].$$

We note that the above definitions can be extended to models not closed under negation by considering $\max_{h \in \mathcal{H}} \left| \frac{1}{n} \sum_{i=1}^n y_i^\pi h(\mathbf{x}_i) \right|$. Let D' differ from D only in one example, $(\mathbf{x}_j, y_j) \rightarrow (\mathbf{x}'_j, y'_j)$.

Lemma 3. $|\mathcal{P}_{\text{in}}(\mathcal{H}|D) - \mathcal{P}_{\text{in}}(\mathcal{H}|D')| \leq \frac{4}{n}$.

Proof. Consider any permutation π ; the sum $\sum_{i=1}^n y_i^\pi h(\mathbf{x}_i)$ changes by at most 4 (only two points are affected) for every $h \in \mathcal{H}$. Thus, the maximum over $h \in \mathcal{H}$ changes by at most 4 and the lemma follows. \blacksquare

Lemma 3 together with McDiarmid's inequality implies a concentration of \mathcal{P}_{in} about \mathcal{P}_{out} , which means we can work with \mathcal{P}_{in} instead of the unknown \mathcal{P}_{out} .

Corollary 2. *With probability at least $1 - \delta$,*

$$\mathcal{P}_{\text{out}}(\mathcal{H}) \leq \mathcal{P}_{\text{in}}(\mathcal{H}|D) + \sqrt{\frac{8}{n} \ln \frac{1}{\delta}}.$$

Let g^π be the empirical risk minimizer for the permuted data. Since $e_{\text{in}}(h) = 2 - \frac{2}{n} \sum_{i=1}^n y_i h(\mathbf{x}_i)$, it follows that the empirical risk minimizer maximizes the correlation, and so:

$$\max_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n y_i^\pi h(\mathbf{x}_i) = \frac{1}{n} \sum_{i=1}^n y_i^\pi g^\pi(\mathbf{x}_i).$$

Thus, combining with Theorem 2, we have that for empirical risk minimization, \hat{e}_{gen} and \mathcal{P}_{in} are related:

Theorem 4. *For empirical risk minimization, $\mathcal{P}_{\text{in}}(\mathcal{H}|D) = \frac{1}{2} \hat{e}_{\text{gen}}(\mathcal{H}|D) + \bar{y} \mathbb{E}_\pi [\bar{g}^\pi]$.*

We will now prove our uniform bound on the generalization error, with respect to \mathcal{P}_{in} .

3.2 Bound for the Out-Sample Error

We are interested in the worst case generalization error, $\sup_{h \in \mathcal{H}} \{e_{\text{out}}(h) - e_{\text{in}}(h)\}$. We will need some regularity property of the error e_{in} . Generally, $e_{\text{in}}(h) = \frac{1}{n} \sum_{i=1}^n e(y_i, h(\mathbf{x}_i))$, and we will suppose that e_{in} satisfies a ‘‘continuity condition’’ similar to the assumptions in McDiarmid's lemma. Specifically, for data sets D and D' which differ on only one point, $(\mathbf{x}_i, y_i) \rightarrow (\mathbf{x}'_i, y'_i)$,

$$|e_{\text{in}}(h|D) - e_{\text{in}}(h|D')| \leq \frac{c}{n}.$$

Most of our discussion has been based on the squared error function $e(y, y') = (y - y')^2$; in this case, the regularity condition holds with $c = 4$ for classification. For regression, one would assume a bounded range for the target and hypothesis functions to ensure such a regularity condition.

The first step in the proof of our bound uses the standard ghost sample and symmetrization arguments typical of modern generalization error proofs (see for example Bartlett and Mendelson (2002); Shawe-Taylor and Cristianini (2004)). Let $\mathbf{r} = [r_1, \dots, r_n]$ be an arbitrary ± 1 sequence.

Lemma 4. *With probability at least $1 - \delta$:*

$$\sup_{h \in \mathcal{H}} \{e_{\text{out}}(h) - e_{\text{in}}(h)\} \leq \mathbb{E}_{D, D'} \left[\sup_{h \in \mathcal{H}} \left\{ \frac{1}{n} \sum_{i=1}^n r_i (e(y'_i, h(\mathbf{x}'_i)) - e(y_i, h(\mathbf{x}_i))) \right\} \right] + \sqrt{\frac{c^2}{2n} \ln \frac{1}{\delta}}.$$

Proof. The following sequence of inequalities establishes the result.

$$\begin{aligned}
\sup_{h \in \mathcal{H}} \{e_{\text{out}}(h) - e_{\text{in}}(h)\} &\stackrel{(a)}{\leq} \mathbb{E}_D \left[\sup_{h \in \mathcal{H}} \left\{ \mathbb{E}_{\mathbf{x}, y} \left[e(y, h(\mathbf{x})) - \frac{1}{n} \sum_{i=1}^n e(y_i, h(\mathbf{x}_i)) \right] \right\} \right] + \sqrt{\frac{c^2}{2n} \ln \frac{1}{\delta}}, \\
&= \mathbb{E}_D \left[\sup_{h \in \mathcal{H}} \left\{ \mathbb{E}_{D'} \left[\frac{1}{n} \sum_{i=1}^n e(y'_i, h(\mathbf{x}'_i)) - e(y_i, h(\mathbf{x}_i)) \right] \right\} \right] + \sqrt{\frac{c^2}{2n} \ln \frac{1}{\delta}}, \\
&\stackrel{(b)}{\leq} \mathbb{E}_{D, D'} \left[\sup_{h \in \mathcal{H}} \left\{ \frac{1}{n} \sum_{i=1}^n e(y'_i, h(\mathbf{x}'_i)) - e(y_i, h(\mathbf{x}_i)) \right\} \right] + \sqrt{\frac{c^2}{2n} \ln \frac{1}{\delta}}, \\
&\stackrel{(c)}{=} \mathbb{E}_{D, D'} \left[\sup_{h \in \mathcal{H}} \left\{ \frac{1}{n} \sum_{i=1}^n r_i (e(y'_i, h(\mathbf{x}'_i)) - e(y_i, h(\mathbf{x}_i))) \right\} \right] + \sqrt{\frac{c^2}{2n} \ln \frac{1}{\delta}}.
\end{aligned}$$

(a) follows from the regularity condition on e_{in} and McDiarmid's inequality because $e_{\text{out}}(h) - e_{\text{in}}(h)$ changes by at most $\frac{c}{n}$ if one data point changes, and so the maximum changes by at most that much; (b) follows from convexity (supremum of expectation is at most expectation of supremum); finally, (c) follows because $r_i = -1$ corresponds to exchanging $\mathbf{x}_i, \mathbf{x}'_i$ in the expectation which does not change the expectation (it amounts to relabeling of random variables). ■

For classification and the squared error, $y^2 = h^2 = 1$ and so we have

Corollary 3. *For classification and squared error, with probability at least $1 - \delta$:*

$$\sup_{h \in \mathcal{H}} \{e_{\text{out}}(h) - e_{\text{in}}(h)\} \leq \mathbb{E}_{D, D'} \left[\max_{h \in \mathcal{H}} \left\{ \frac{2}{n} \sum_{i=1}^n r_i (y_i h(\mathbf{x}_i) - y'_i h(\mathbf{x}'_i)) \right\} \right] + \sqrt{\frac{8}{n} \ln \frac{1}{\delta}}.$$

Lemma 4 holds for an *arbitrary* sequence \mathbf{r} which is independent of D, D' . An immediate corollary is that we can take the expectation with respect to \mathbf{r} , for *arbitrarily* distributed \mathbf{r} , as long as \mathbf{r} is independent of D, D' .

Fix \mathbf{y} . For a given permutation $\boldsymbol{\pi}$, we will define a corresponding sequence $\mathbf{r}^\boldsymbol{\pi}$ as follows: $r_i^\boldsymbol{\pi} = y_i^\boldsymbol{\pi} y_i$; then, because $y_i^2 = 1$, $y_i^\boldsymbol{\pi} = r_i^\boldsymbol{\pi} y_i$. Thus, given \mathbf{y} , for each of the $n!$ permutations $\boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_{n!}$, we have a corresponding ± 1 sequence; we thus obtain a multiset of sequences $S_{\mathbf{y}} = \{\mathbf{r}^{\boldsymbol{\pi}_1}, \dots, \mathbf{r}^{\boldsymbol{\pi}_{n!}}\}$ (there may be repetitions in this set as two different permutations may result in the same sequence of values); we have a mapping from permutations to the ± 1 sequences in $S_{\mathbf{y}}$. If \mathbf{r} is a random vector of ± 1 s which is selected uniformly from $S_{\mathbf{y}}$, then $\mathbf{r} \cdot \mathbf{y}$ (componentwise product) is uniform over the permutations of \mathbf{y} . We say that $S_{\mathbf{y}}$ generates the permutations on \mathbf{y} . Similarly, we can define $S_{\mathbf{y}'}$, the generator of permutations on the ghost target vector \mathbf{y}' . Unfortunately, $S_{\mathbf{y}}, S_{\mathbf{y}'}$ depend on D, D' , and so we can't take the expectation uniformly over (for example) $\mathbf{r} \in S_{\mathbf{y}}$. We can overcome this by introducing a second ghost data set D'' to “approximately” generate the permutations for \mathbf{y}, \mathbf{y}' , ultimately allowing us to prove the main theorem.

Theorem 5. *With probability at least $1 - \delta$,*

$$\sup_{h \in \mathcal{H}} \{e_{\text{out}}(h) - e_{\text{in}}(h)\} \leq 4\mathcal{P}_{\text{out}}(\mathcal{H}) + 3\sqrt{\frac{8}{n} \ln \frac{6}{\delta}}.$$

3.2.1 Proof of Theorem 5

Let D'' be a *second, independent* ghost data set, and let $S_{\mathbf{y}''}$ be the generator of permutations for \mathbf{y}'' . We will use Corollary 3, and take the expectation with respect to the ± 1 sequences $\mathbf{r}'' \in S_{\mathbf{y}''}$. The first term on the RHS of Corollary 3 becomes

$$\mathbb{E}_{D, D'} \frac{1}{n!} \sum_{\boldsymbol{\pi}} \left[\sup_{h \in \mathcal{H}} \frac{2}{n} \sum_{i=1}^n r_i''(\boldsymbol{\pi})(y_i h(\mathbf{x}_i) - y_i' h(\mathbf{x}_i')) \right], \quad (4)$$

where the first summation is over the $n!$ permutations, each permutation $\boldsymbol{\pi}$ inducing a particular sequence $\mathbf{r}''(\boldsymbol{\pi})$. Consider the sequences of ± 1 vectors $\{\mathbf{r}_1, \dots, \mathbf{r}_{n!}\}$ and $\{\mathbf{r}'_1, \dots, \mathbf{r}'_{n!}\}$, corresponding to the permutations $\{\boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_{n!}\}$. The next lemma will ultimately relate the expectation over permutations in the second ghost data set to the permutations over D, D' .

Lemma 5. *With probability at least $1 - \delta$, there is a one-to-one mapping from the sequences in $S_{\mathbf{y}''} = \{\mathbf{r}''(\boldsymbol{\pi}_i)\}_{i=1}^{n!}$ to $S_{\mathbf{y}} = \{\mathbf{r}(\boldsymbol{\pi}_j)\}_{j=1}^{n!}$ such that*

$$\left| \frac{1}{n} \sum_{i=1}^n (r_i'' - r_i(\mathbf{r}'')) y_i h(\mathbf{x}_i) \right| \leq \sqrt{\frac{8}{n} \ln \frac{2}{\delta}},$$

for every $\mathbf{r}'' \in S_{\mathbf{y}''}$ and every $h \in \mathcal{H}$ (we write $\mathbf{r}(\mathbf{r}'')$ to denote the sequence $\mathbf{r} \in S_{\mathbf{y}}$ to which \mathbf{r}'' is mapped). Similarly, there exists such a mapping from $S_{\mathbf{y}''}$ to $S_{\mathbf{y}'}$.

Proof. We can (without loss of generality) reorder the points in D'' so that the first k'' are $+1$, so $y_1'' = \dots = y_{k''}'' = +1$, and the remaining are -1 . Similarly, we can order the points in D so that the first k are $+1$, so $y_1 = \dots = y_k = +1$. We now construct the mapping from $S_{\mathbf{y}''}$ to $S_{\mathbf{y}}$ as follows. For a given permutation $\boldsymbol{\pi}$, we map $\mathbf{r}''(\boldsymbol{\pi}) \in S_{\mathbf{y}''}$ to $\mathbf{r}(\boldsymbol{\pi}) \in S_{\mathbf{y}}$. This mapping is clearly bijective since every permutation corresponds uniquely to a sequence in $S_{\mathbf{y}}$ (and $S_{\mathbf{y}''}$).

By definition, $r_i = y_i^{\boldsymbol{\pi}} y_i$ and $r_i'' = y_i''(\boldsymbol{\pi}) y_i''$ (we use the notation $y^{\boldsymbol{\pi}}$ and $y(\boldsymbol{\pi})$ interchangeably). If $r_i \neq r_i''$, it implies that either $y_i^{\boldsymbol{\pi}} \neq y_i''(\boldsymbol{\pi})$ or $y_i \neq y_i''$. Since \mathbf{y} and \mathbf{y}'' disagree on exactly $|k - k''|$ locations (and similarly for $y^{\boldsymbol{\pi}}$ and $y''(\boldsymbol{\pi})$), the number of locations where \mathbf{r} and \mathbf{r}'' disagree is therefore at most $2|k - k''|$. Thus, for any \mathbf{r}'' and any $h \in \mathcal{H}$,

$$\begin{aligned} \left| \frac{1}{n} \sum_{i=1}^n (r_i'' - r_i(\mathbf{r}'')) y_i h(\mathbf{x}_i) \right| &\leq \frac{1}{n} \sum_{i=1}^n |r_i'' - r_i(\mathbf{r}'')| |y_i h(\mathbf{x}_i)| \\ &= \frac{1}{n} \sum_{i=1}^n |r_i'' - r_i(\mathbf{r}'')| \\ &\leq \frac{4|k - k''|}{n}, \end{aligned}$$

where the last inequality follows because there are at most $2|k - k''|$ locations where \mathbf{r}'' and $\mathbf{r}(\mathbf{r}'')$ disagree, and when they disagree, the difference is ± 2 . We observe that $\frac{1}{2} \sum_{i=1}^n y_i - y_i'' = k - k''$ and so,

$$\left| \frac{1}{n} \sum_{i=1}^n (r_i'' - r_i(\mathbf{r}'')) y_i h(\mathbf{x}_i) \right| \leq \left| \frac{2}{n} \sum_{i=1}^n y_i - y_i'' \right| = \left| \frac{2}{n} \sum_{i=1}^n z_i \right|,$$

where $z_i = y_i - y_i''$. Since \mathbf{y} and \mathbf{y}'' are identically distributed, z_i are independent and zero mean. We consider the function $f(z_1, \dots, z_n) = \frac{1}{n} \sum_{i=1}^n z_i$. Since $z_i \in \{0, \pm 1\}$, if you change one of the z_i , f changes by at most $\frac{2}{n}$, and so the conditions hold to apply McDiarmid's inequality to f . Thus, with probability at least $1 - \delta$,

$$\frac{1}{n} \sum_{i=1}^n z_i \leq \sqrt{\frac{2}{n} \ln \frac{1}{\delta}}.$$

By the symmetry of z_i , it follows that with probability at least $1 - 2\delta$,

$$\left| \frac{1}{n} \sum_{i=1}^n z_i \right| \leq \sqrt{\frac{2}{n} \ln \frac{1}{\delta}}.$$

To conclude, multiply by 2 and rescale $2\delta \rightarrow \delta$. ■

Given D, D', D'' , assume the mappings which are known to exist by the previous lemma are $\mathbf{r}(\mathbf{r}'')$ and $\mathbf{r}'(\mathbf{r}'')$. We can rewrite the internal summand in the expression of Equation (4) using the equality

$$r_i''(y_i h(\mathbf{x}_i) - y_i' h(\mathbf{x}_i')) = (r_i'' - r_i(\mathbf{r}'') + r_i(\mathbf{r}''))y_i h(\mathbf{x}_i) - (r_i'' - r_i'(\mathbf{r}'') + r_i'(\mathbf{r}''))y_i' h(\mathbf{x}_i').$$

Using Lemma 5, we can, with probability at least $1 - \delta$, bound the term which involves $(r_i'' - r_i(\mathbf{r}''))$ in Equation (4); and, similarly, with probability at least $1 - \delta$, we bound the term involving $(r_i'' - r_i'(\mathbf{r}''))$. We can apply this bound inside the sup because the bound from Lemma 5 applies to every $h \in \mathcal{H}$ for the given mapping. Thus, with probability at least $1 - 2\delta$, the expression in Equation (4) is bounded by:

$$\mathbb{E}_{D, D'} \frac{1}{n!} \sum_{\pi} \left[\sup_{h \in \mathcal{H}} \frac{2}{n} \sum_{i=1}^n (r_i(\mathbf{r}'')y_i h(\mathbf{x}_i) - r_i'(\mathbf{r}'')y_i' h(\mathbf{x}_i')) \right] + 2\sqrt{\frac{8}{n} \ln \frac{2}{\delta}},$$

where $\mathbf{r}''(\pi)$ cycles through the sequences in $S_{\mathbf{y}''}$. Since the mappings $\mathbf{r}(\mathbf{r}'')$ and $\mathbf{r}'(\mathbf{r}'')$ are one-to-one, $\mathbf{r}(\mathbf{r}'') \cdot \mathbf{y}$ cycles through the permutations of \mathbf{y} , and similarly for $\mathbf{r}'(\mathbf{r}'') \cdot \mathbf{y}'$. Since \mathcal{H} is closed under negation, we finally obtain the bound

$$\mathbb{E}_D \frac{1}{n!} \sum_{\pi} \left[\sup_{h \in \mathcal{H}} \frac{2}{n} \sum_{i=1}^n y_i^{\pi} h(\mathbf{x}_i) \right] + \mathbb{E}_{D'} \frac{1}{n!} \sum_{\pi} \left[\sup_{h \in \mathcal{H}} \frac{2}{n} \sum_{i=1}^n y_i^{\pi} h(\mathbf{x}_i') \right] + 2\sqrt{\frac{8}{n} \ln \frac{2}{\delta}}.$$

Since D, D' are identically distributed, this reduces to $2\mathbb{E}_{D, \pi} \left[\sup_{h \in \mathcal{H}} \frac{2}{n} \sum_{i=1}^n y_i^{\pi} h(\mathbf{x}_i) \right] = 4\mathcal{P}_{\text{out}}(\mathcal{H}) + 2\sqrt{\frac{8}{n} \ln \frac{2}{\delta}}$. To summarize, with probability at least $1 - 2\delta$, the first term on the RHS of Corollary 3 is bounded by this expression. We thus conclude that with probability at least $1 - 3\delta$,

$$\sup_{h \in \mathcal{H}} \{e_{\text{out}}(h) - e_{\text{in}}(h)\} \leq 4\mathcal{P}_{\text{out}}(\mathcal{H}) + \sqrt{\frac{8}{n} \ln \frac{1}{\delta}} + 2\sqrt{\frac{8}{n} \ln \frac{2}{\delta}}.$$

After redefining $3\delta \rightarrow \delta$, the RHS is bounded by $4\mathcal{P}_{\text{out}}(\mathcal{H}) + 3\sqrt{\frac{8}{n} \ln \frac{6}{\delta}}$. ■

The bound, being uniform, applies to the empirical risk minimizer g . Combining Theorem 5 with Corollary 2 and Theorem 4, we obtain a bound for the generalization error.

Corollary 4. *With probability at least $1 - \delta$,*

$$e_{\text{out}}(g) \leq e_{\text{in}}(g) + 2\widehat{e}_{\text{gen}}(\mathcal{H}|D) + 4\bar{y} \mathbb{E}_{\pi} [\bar{g}^{\pi}] + 4\sqrt{\frac{8}{n} \ln \frac{12}{\delta}}.$$

Remarks. Corollary 4 justifies using the permutation estimate $\widehat{e}_{\text{gen}}(\mathcal{H}|D)$; one could use the bound in Corollary 4, which can be computed just as easily. Typically, \bar{y} is close to 0 (balanced data) and $\bar{g} \sim \bar{y}$, so the third term is approximately \bar{y}^2 which is small and approximately constant (hence it will not affect model selection much). Hence the bound, up to small correction terms is $2\widehat{e}_{\text{gen}}$ for empirical risk minimization. Since the bound in Theorem 5 is uniform, it applies to any algorithm; the bound, as with all other data dependent bounds, only looks at the hypothesis set – the permutation complexity is a complexity measure for the model. However, as the example with linear ridge regression illustrates, the permutation estimate uses a penalty which is algorithm specific, and can directly account for regularization in the learning algorithm. We show empirically that the permutation estimate works even for regularized algorithms. This can be justified because there are two ways to view a regularized algorithm. The practical approach is to consider a regularization penalty term and minimize the penalized in-sample error. An alternative view is to consider regularization as a constraint on the hypothesis set, in which case one performs constrained in-sample error minimization. In this second view, since one is doing empirical risk minimization, our results apply, and the permutation complexity uniform bound justifies the use of the permutation generalization estimate.

The analysis can accommodate models not closed under negation by using absolute values in the permutation complexity. All the analysis can be extended to sampling with replacement rather than permutations (sampling without replacement). One could use techniques for bounding Rademacher complexity to bounding permutation complexity on specific domains (for example, see Bartlett and Mendelson (2002) for decision trees, neural networks, kernel methods).

Any approximate estimate of the out-sample error (which satisfies some bound of this form) can be used for model selection, after adding a (small) penalty for the “complexity of model selection” (see Bartlett *et al.* (2002)). In practice, this penalty for the complexity of model selection is ignored (as in Bartlett *et al.* (2002)).

3.3 Concentration around $\mathcal{P}_{\text{in}}(\mathcal{H}|D)$

The goal here is to show that one can essentially estimate $\mathcal{P}_{\text{in}}(\mathcal{H}|D)$, which is an average over all permutations of the data, using one permutation of the data – i.e., for a randomly selected permutation π , $\sup_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n y_i^{\pi} h(\mathbf{x}_i)$ concentrates around its expectation over all permutations. We prove:

Theorem 6. *For a random permutation π , with probability at least $1 - \delta$,*

$$\mathcal{P}_{\text{in}}(\mathcal{H}|D) \leq \sup_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n y_i^{\pi} h(\mathbf{x}_i) + c\sqrt{\frac{2}{n} \ln \frac{2}{\delta}}.$$

for some $c \leq 2 + \frac{1}{\sqrt{2 \ln 4}}$.

To prove Theorem 6, we will link the permutation complexity to the *bootstrap* distribution

(sampling targets uniformly *with replacement*). Specifically, we will show that the permutation and the bootstrap estimates are close. We will then show that the bootstrap estimate is concentrated via McDiarmid's inequality, which will ultimately allow us to conclude that the permutation estimate is also concentrated.

The bootstrap sampling process B constructs an independent, random sequence of targets \mathbf{y}^B , where each y_i^B is sampled independently and uniformly from y_1, \dots, y_n ; the key requirement is that y_i^B are independent samples. We define the *bootstrap complexity*

$$\mathcal{B}_{\text{in}}(\mathcal{H}|D) = \mathbb{E}_B \left[\sup_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n y_i^B h(\mathbf{x}_i) \right],$$

which is very similar to the permutation complexity, except the y_i are sampled with replacement. It is convenient to define some basic processes which will help in relating the bootstrap process to the permutation process. We can view the bootstrap process as the composition of two processes:

1. Sample κ , the number of +1s in the sequence \mathbf{y}^B ; κ has a binomial distribution.
2. Among all sequences with κ +1s, sample one of them uniformly. We denote this process $B|\kappa$.

Note that if $\kappa = k$, where k is the number of +1s in \mathbf{y} , then $B|\kappa$ is exactly the permutation process. Infact, one way to generate a sample according to $B|\kappa$ is to first generate a random permutation π . Now, if $\kappa < k$, randomly select $|k - \kappa|$ +1s and flip them to -1 ; similarly, if $\kappa > k$, randomly select $|k - \kappa|$ -1 s and flip them to $+1$. We call this random flipping process $F_{k,\kappa}$. More formally, $F_{k,\kappa}$ takes as input a sequence with k +1s and returns a random sequence with κ +1s. We have that

$$B|\kappa = F_{k,\kappa} \circ \pi,$$

where when the context is clear, we use π to denote the random sampling process which generates permutations as well as a specific random permutation.

Lemma 6. $|\mathcal{B}_{\text{in}}(\mathcal{H}|D) - \mathcal{P}_{\text{in}}(\mathcal{H}|D)| \leq \frac{1}{\sqrt{n}}$.

Proof. Let k be the number of y_i which are +1. Based on the discussion above, B is the composition of two processes, so

$$\mathcal{B}_{\text{in}}(\mathcal{H}|D) = \mathbb{E}_{\kappa} \mathbb{E}_{B|\kappa} \left[\sup_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n y_i^B h(\mathbf{x}_i) \middle| \kappa \right].$$

We also have that

$$\mathbb{E}_{B|\kappa} \left[\sup_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n y_i^B h(\mathbf{x}_i) \middle| \kappa \right] = \mathbb{E}_{F_{k,\kappa}} \mathbb{E}_{\pi} \left[\sup_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n y_i^{\pi, F} h(\mathbf{x}_i) \right],$$

where we use the notation $\mathbf{y}^{\pi, F}$ to denote the random vector which results from generating a random permutation π and then applying the random flipping process $F_{k,\kappa}$. Since $\mathbf{y}^{\pi, F}$ differs from \mathbf{y}^{π} in exactly $|k - \kappa|$ positions, and when they differ, the difference is ± 2 ,

$$\sup_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n y_i^{\pi} h(\mathbf{x}_i) - \frac{2|k - \kappa|}{n} \leq \sup_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n y_i^{\pi, F} h(\mathbf{x}_i) \leq \sup_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n y_i^{\pi} h(\mathbf{x}_i) + \frac{2|k - \kappa|}{n}.$$

Now, first take the expectation with respect to $\boldsymbol{\pi}$; then with respect to $F_{k,\kappa}$; and, finally with respect to κ to obtain:

$$\mathcal{P}_{\text{in}}(\mathcal{H}|D) - \frac{2}{n} \mathbb{E}_{\kappa} [|k - \kappa|] \leq \mathcal{B}_{\text{in}}(\mathcal{H}|D) \leq \mathcal{P}_{\text{in}}(\mathcal{H}|D) + \frac{2}{n} \mathbb{E}_{\kappa} [|k - \kappa|].$$

By convexity, $\mathbb{E}_{\kappa} [|k - \kappa|] \leq \sqrt{\mathbb{E}_{\kappa} [(k - \kappa)^2]} = \sqrt{\text{Var}[k - \kappa]} \leq \frac{1}{2}\sqrt{n}$ (the last inequality is because κ has a binomial distribution), the result follows. \blacksquare

In addition to furthering our cause toward the proof of Theorem 6, Lemma 6 is interesting in its own right, because it says that permutation and bootstrap sampling are similar (for classification). The nice thing about the bootstrap estimate is that the expectation is over independent y_1^B, \dots, y_n^B , and so we can easily get a concentration result.

Lemma 7. *For a random bootstrap sample B , with probability at least $1 - \delta$,*

$$\mathcal{B}_{\text{in}}(\mathcal{H}|D) \leq \sup_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n y_i^B h(\mathbf{x}_i) + \sqrt{\frac{2}{n} \ln \frac{1}{\delta}}.$$

Proof. Since the function inside the expectation changes by at most $\frac{2}{n}$ if you change one of the samples, the result follows by application of McDiarmid's inequality, \blacksquare

We are now ready to prove concentration for \mathcal{P}_{in} .

Proof. (Theorem 6.) As in the proof of Lemma 6, we generate \mathbf{y}^B in two steps. First generate κ , the number of +1's in \mathbf{y}^B ; κ has a binomial distribution with expectation k . Now, generate a random permutation $\boldsymbol{\pi}$, and flip (as appropriate) a randomly selected $|k - \kappa|$ entries using the random flipping process $F_{k,\kappa}$ (remember, k is the number of +1s in \mathbf{y}). Consider the function $f(\mathbf{y}^B) = \kappa$, the number of +1s in \mathbf{y}^B . Changing a single entry of \mathbf{y}^B changes f by at most 1. We can thus apply McDiarmid's inequality f to obtain that with probability at least $1 - \delta$,

$$|\kappa - k| \leq \sqrt{\frac{n}{2} \ln \frac{2}{\delta}}.$$

Thus, with probability at least $1 - \delta$, \mathbf{y}^B differs from $\mathbf{y}^{\boldsymbol{\pi}}$ in at most $\sqrt{\frac{n}{2} \ln \frac{2}{\delta}}$ positions. Hence, with probability at least $1 - \delta$,

$$\sup_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n y_i^B h(\mathbf{x}_i) \leq \sup_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n y_i^{\boldsymbol{\pi}} h(\mathbf{x}_i) + 2\sqrt{\frac{1}{2n} \ln \frac{1}{\delta}}.$$

Combining with Lemma 7, we conclude that for a random permutation $\boldsymbol{\pi}$, with probability at least $1 - 2\delta$,

$$\mathcal{B}_{\text{in}}(\mathcal{H}|D) \leq \sup_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n y_i^{\boldsymbol{\pi}} h(\mathbf{x}_i) + 2\sqrt{\frac{2}{n} \ln \frac{2}{\delta}}.$$

By Lemma 6, $\mathcal{P}_{\text{in}}(\mathcal{H}|D) \leq \mathcal{B}_{\text{in}}(\mathcal{H}|D) + \frac{1}{\sqrt{n}}$; setting $\delta \rightarrow \frac{\delta}{2}$, and using $\delta \leq 1$, we obtain Theorem 6 after a little algebra. \blacksquare

Remarks. We have not only established that \mathcal{P}_{in} is concentrated, but we have also established a general connection between the permutation and bootstrap based estimates. In this particular case, we see that sampling with and without replacement are very closely related. In practice, sampling without replacement can be very different, because one is never in the truly asymptotic regime. Along that vein, even though we have concentration, it pays in practice to take the average over a few permutations. Since $\widehat{e}_{\text{gen}} = 2\mathcal{P}_{\text{in}} - 2\bar{y} \mathbb{E}_{\boldsymbol{\pi}} [\bar{g}^{\boldsymbol{\pi}}]$, it follows that \widehat{e}_{gen} also concentrates when $\bar{g}^{\boldsymbol{\pi}}$ concentrates about its expectation. This will not be the case for all learning algorithms; but, for learning algorithms which minimize an error function which satisfies a lipshitz-like constraint (such as empirical risk minimization where the risk is lipshitz, as in our case of empirical minimization of misclassification error), then we do get concentration. The proof is similar to the concentration around \mathcal{P}_{in} , and we omit the details. First one shows concentration for the bootstrap sampling statistic \bar{g}^B ; one then links the bootstrap and permutation estimates to finally arrive at the destination.

3.4 The Permutation Complexity for VC-Classes

The growth function $m_{\mathcal{H}}(n)$ is the maximum number of dichotomies a hypothesis set can implement on a data set D with n points. A VC-class has finite VC-dimension d_{VC} which means that $m_{\mathcal{H}}(n) = O(n^{d_{\text{VC}}})$. When $\sum_i y_i = 0$, the bootstrap complexity is equivalent to the Rademacher penalty, which can be bounded by the VC-dimension (Shawe-Taylor and Cristianini, 2004). Using the arguments in the previous section which relate sampling without replacement to sampling with replacement, one can thus bound the permutation complexity as well. The asymptotics will not change if instead of $\bar{y} = 0$, one has $\mu = \mathbb{E}[y] = 0$, because then $\bar{y} = \frac{1}{n} \sum_i y_i$ is close to zero (to within $O(\sqrt{\frac{1}{n} \ln \frac{1}{\delta}})$) with probability at least $1 - \delta$.

When $\mu \neq 0$, the theoretical bounds here get loose, and one needs to make a more careful study. Nevertheless the permutation estimate shows good empirical performance.

4 Experiments

We compared the permutation estimate in a variety of settings to test both its validation capability, and its performance during model selection. For regression, we used linear models and conducted an extensive comparison of several methods, including several statistical estimates on simulated data (we averaged results over more than 1 million experiments). For classification on real data, we used a held out test set for evaluation (we averaged results over 1,000 random test-training splits). In general we compare LOO-CV and the permutation estimate, which are both generally applicable, requiring only the ability to learn with a model. For classification, we also compared the Rademacher penalty, used in a similar vein to the permutation complexity by applying the algorithm to the Rademacher variables (it is not possible to compute the empirical Rademacher complexity as that would entail a full empirical risk minimization). We use LOO-CV as the strawman benchmark because it is widely used; it should be noted that some of the pitfalls of the LOO-CV are its computation complexity, which is addressed by the leave- K -out CV estimate (with $K < n$); we also compared with 10-fold cross validation when learning was costly. The tradeoffs between leave- K -out and LOO have been studied, and in general most validation methods have been compared

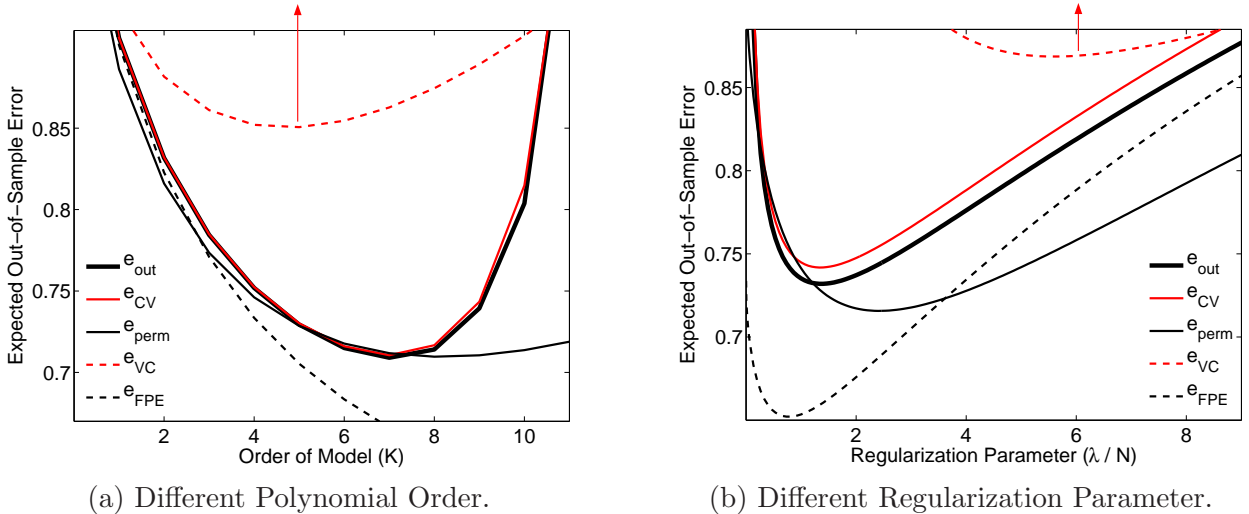


Figure 3: Performance of validation measures as compared to the true expected out-sample error. LOO-CV gives very good estimates of the out-sample error on average.

to LOO. Hence LOO is a valid benchmark. For linear regression, we compared some statistical estimates (eg. Akaike’s final prediction error (FPE)), as well as the VC penalty measure.

4.1 Data

We considered a number of data sets:

Simulated Data – Regression. The input \mathbf{x} is uniform on $[-1, 1]$. The target function is a polynomial of degree d_f , which we write as $f(x) = \sum_{i=0}^{d_f} a_i L_i(x)$, where $L_i(x)$ are the Legendre polynomials. The regression model is

$$y_i = f(x_i) + \sigma \epsilon_i,$$

where ϵ_i are *iid* standard Normals and σ^2 , the noise variance, was chosen uniformly from $(0, 1]$. We consider polynomial models of different order (order selection) and different regularization parameters λ in the ridge regression (regularization parameter selection). For order selection, we used $n = 100$ and for regularization parameter selection, we used $n = 15$.

Simulated Data – Classification. We considered the 2-D problem shown in Figure 4, where the green region is $+1$. We used k -nearest neighbor (complexity parameter is k) and decision trees with greedy splitting according to information gain (complexity controlled by number of leaves). More precisely, we used variation of the ID3 decision tree learning algorithm Quinlan (1986), where the tree is constructed by splitting the node which produces the maximum information gain over all leaf node splits; the algorithm starts with a single root node.

In the simulated setting we are able to generate data points controlling the amount of noise and the proportion of positive and negative examples in the data. This allows us to conduct a systematic comparison of the different validation methods in different settings.

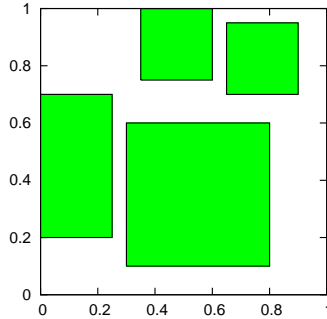


Figure 4: Simulated classification problem.

Data	$ D $	Pos./Neg.	Dim.
Abalone	4,177	0.50/0.50	7
Ionosphere	351	0.64/0.36	34
M.Mass	830	0.51/0.49	5
Parkinsons	195	0.75/0.25	22
P.I.D.	768	0.35/0.75	8
Spambase	4,601	0.39/0.61	57
Transfusion	748	0.23/0.77	4
WDBC	569	0.37/0.63	30
Diffusion	3,554	0.22/0.78	16

Table 2: Real datasets used in experiments.

Real Data – Classification. We evaluated the methods on various UCI ML repository data sets (Asuncion and Newman, 2007), using decision trees (DT) and k -nearest neighbors (k -NN). See Table 2 for some statistics on the data sets we used. For Abalone data the decision task was to predict $[\text{age} \geq 10]$. We also used a novel large data set for predicting the diffusion of YouTube videos in the LiveJournal blogosphere – the task is to determine if a video will spread to at least M blogs, based on features of the early diffusion (e.g. the in/out-degrees of initially affected bloggers). This is an extremely hard and noisy prediction problem. For a detailed description of the data set, see Magdon-Ismail *et al.* (2009)

4.2 Results on Regression

In the simulated linear ridge regression setting, we compared the permutation and LOO-CV estimates together with Akaike’s FPE Akaike (1970) (a statistical estimate from information theory) and a VC-penalty for regression ((Cherkassky and Mulier, 2007, (4.27b)) . These penalties are defined in terms of an effective model complexity, d_{eff} (we used $d_{\text{eff}} = \text{trace}(S(\lambda))$). Define $p = \frac{n}{d_{\text{eff}}}$; the other two estimates are:

$$\begin{aligned} \text{VC penalty (Cherkassky and Mulier, 2007, (4.27b))} \quad e_{\text{out}}(g) &\leq \frac{\sqrt{p}}{\sqrt{p} - \sqrt{1 + \ln p + \frac{\ln N}{2d_{\text{eff}}}}} \cdot e_{\text{in}}(g) \\ \text{Akaike’s FPE (Akaike, 1970)} \quad e_{\text{out}}(g) &\approx \frac{p+1}{p-1} \cdot e_{\text{in}}(g) \end{aligned}$$

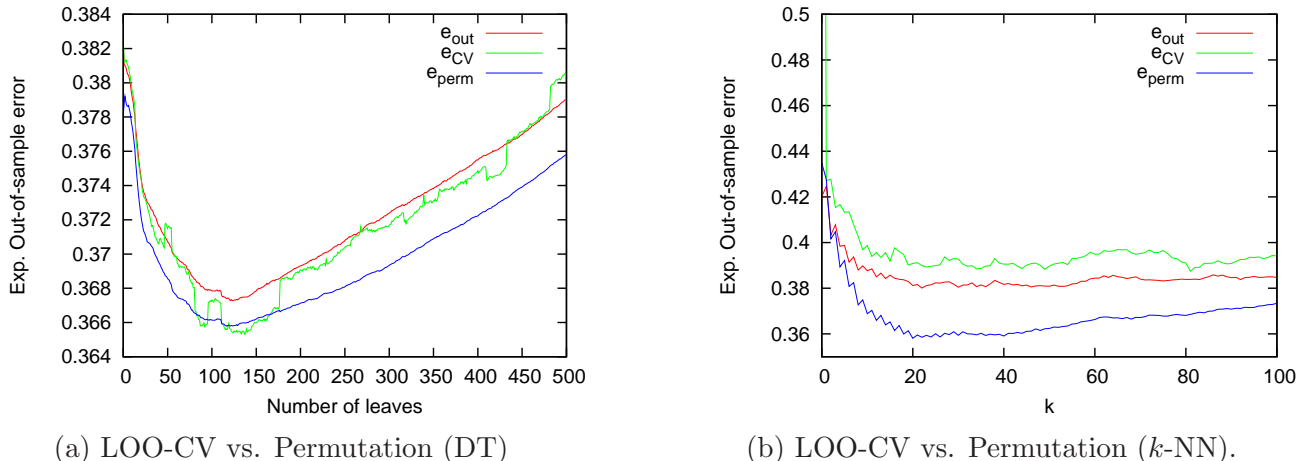


Figure 5: Comparison of the permutation and LOO-CV estimates on a diffusion prediction data set with the decision tree and nearest neighbor learning models. The permutation estimate is well behaved and smooth whereas the LOO-CV estimate is very unstable.

The graphs in Figure 3 compare the validation estimates as the model varies along two dimensions: (a) polynomial regression models with different polynomial order and no regularization; (b) polynomial ridge regression with 5th order polynomials, and different regularization parameters λ . For the permutation estimate, we use the analytical formula given in Section 2. Similarly, for the LOO estimate, we use the analytical formula

$$E_{\text{LOO-CV}} = \frac{1}{n} \sum_{i=1}^n \frac{(y_i - (\mathbf{S}\mathbf{y})_i)^2}{(1 - S_{ii})^2}.$$

We observe that LOO-CV is very hard to beat as an estimate of the out-sample error. However, for model selection, by far the most important use of validation, the LOO-CV estimate performs poorly. We tabulate the performance of model selection below. We evaluate the validation estimates using the fractional regret (as compared to the optimal model), where the fractional regret is:

$$\text{regret} = \frac{e_{\text{out}}(\text{selected model}) - e_{\text{out}}(\text{optimal model})}{e_{\text{out}}(\text{optimal model})}.$$

Table 3 summarizes the performance of model selection using each of the validation estimates. LOO-CV experiences huge regret. This is because in practice it pays to be conservative: LOO-CV pays too big a price when it is wrong (erring on the side of too complex a model). It is already known that LOO-CV can suffer from overfitting during model selection, and several approaches to “regularizing” cross validation have been suggested (see for example Ng (1997), (Hastie *et al.*, 2001, 1- σ rule, pg 216)).

We take a simple approach to regularizing model selection by removing the choice $\lambda = 0$ from the set of models. This corresponds to “regularizing” the model selection methods, because by removing the $\lambda = 0$ model, we are not allowing any validation method to be too aggressive. This makes practical sense, because whenever noise is present in the data, which is the typical case, one

Validation Estimate	Order Selection		λ Selection	
	Regret	Avg Order	Regret	Avg. $\frac{\lambda}{N}$
LOO-CV	540	9.29	18.8	23.1
Perm.	185	7.21	5.96	9.57
VC	508	5.56	3.50	125
FPE	9560	11.42	51.3	18.1

Table 3: Comparison of validation estimates for polynomial order selection and regularization parameter selection. (Regret is the % loss versus the optimal selection which minimizes e_{out} .)

Validation Estimate	λ Selection (no $\lambda = 0$) Regret
LOO-CV	0.44
Perm.	0.39
VC	0.42
FPE	0.87

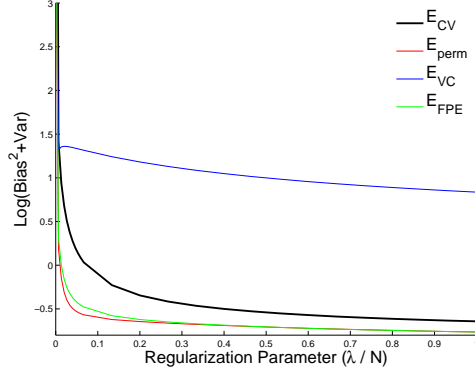
Table 4: Comparison of validation methods for selecting the regularization parameter when the unstable case $\lambda = 0$ is excluded from the set of possible models.

expects some regularization to help. The results of model selection among models not containing $\lambda = 0$ is shown in the Table 4. Note that all methods benefit from a more regularized model selection which excludes the $\lambda = 0$ model. The permutation estimate is now the best performer.

The main purpose for this experiment of removing the $\lambda = 0$ model and then doing selection is to illustrate how the bias and variance of the validation estimators trades off in model selection. For the $\lambda = 0$ model, the LOO-CV has a pathologically high variance; this means that LOO-CV can accidentally select the $\lambda = 0$ model which is a disaster out-sample. When you remove this high variance model, now the LOO-CV estimate looks much better, though it is still not best. We show the bias and variance numbers for the out-sample error estimators in the table of Figure 6. The message is that the variance of the estimators has a role to play in model selection. The $\text{bias}^2 + \text{var}$ governs the expected error in the out-sample error estimate. If your method has unusually high variance for a particularly bad model, then this can lead to serious problems in model selection. The permutation estimate has a generally decent bias with much lower variance than the LOO-CV estimate; when the model selection task is extremely adversarial (eg. when one has the $\lambda = 0$ model), very conservative methods are called for, such as the VC-penalty. As one can observe in Figure 6 and the associated table, the VC penalty generally has large $\text{bias}^2 + \text{var}$, but for the highly unstable $\lambda = 0$ model, the very conservative VC-bound pays off.

4.3 Results on Classification

We compared LOO-CV, the permutation, and the Rademacher approaches to validation. When learning is relatively costly (decision trees), all validation estimates had a fixed number of learning episodes to make computation times reasonable (1000 in our case). We also considered another extreme where only a very few learning episodes are allowed (10 in our case); in such a setting,



Val.	λ/N				Val.	λ/N			
Est.	0	0.02	0.04	0.06	Est.	0	0.02	0.04	0.06
LOO-CV	29400	0.0051	0.0017	0.0009	LOO-CV	3.9×10^9	3.89	1.75	1.17
Perm.	14100	0.221	0.062	0.025	Perm.	0.33	0.32	0.31	0.30
VC	13300	6.13	6.47	6.36	VC	10.9	9.1	8.3	7.8
FPE	14100	0.422	0.180	0.11	FPE	0.33	0.31	0.30	0.29

bias^2
variance

Figure 6: Bias and Variance of the out-sample error estimates. The figure shows the bias² + var in the region of small λ . The VC penalty is generally inferior, except in the $\lambda \approx 0$ region. When there is large instability in the models, a stable estimator which has low variance is called for.

10-fold cross validation is a common alternative to LOO-CV, so we also compared with 10-fold cross validation. When learning is cheap (e.g. k -NN), we do not limit the number of cross validation points. The plots in Figure 5 illustrate the validation estimates for the diffusion prediction data set.

In general, we observe that the CV estimate is more accurate but less stable than the permutation estimate. Our experimental results when using the respective estimates for model selection are shown in Table 5. The permutation and Rademacher approaches are comparable, and generally better than LOO-CV in decision tree learning. In case of k -NN classification LOO-CV showed best results and permutation estimate was a close second. One reason that LOO-CV performs well with k -NN is because there was no limit on the number of learning episodes. For simulated data, we used a balanced data set with 300 data points and 40% noise. We used the simulated example as a testbed to study how the noise level affected the validation estimates. For small noise, the LOO-CV estimate suffers most, and for large noise, approaching 50%, all methods equalize with LOO-CV slightly better. The permutation estimate slightly dominates the Rademacher estimate. These results are shown in Table 7. We also did a systematic study of how the positive to negative example imbalance affects the estimates. We found that when the imbalance in the data increases, the permutation estimate dominates the Rademacher estimate.

For DTs, learning is costly (relative to k -NN), so we fixed the number of learning episodes, and LOO-CV is generally inferior. For k -NNs, when learning is not costly, LOO-CV works well. For both DTs and k -NNs, the permutation estimate generally dominates the Rademacher approach. We have also tried using the uniform bounds for model selection instead of the estimates, and that

Data	Decision Trees			k -Nearest Neighbor		
	LOO-CV	Perm.	Rad.	LOO-CV	Perm.	Rad.
Abalone	0.05	0.02	0.02	0.04	0.04	0.04
Ionosphere	0.17	0.16	0.17	0.17	0.70	0.83
M.Mass	0.09	0.05	0.05	0.09	0.11	0.11
Parkinsons	0.24	0.34	0.41	0.25	0.33	0.43
Pima Indians Diabetes	0.09	0.07	0.07	0.11	0.11	0.14
Spambase	0.07	0.06	0.07	0.19	0.43	0.55
Transfusion	0.10	0.08	0.09	0.09	0.12	0.19
WDBC	0.20	0.23	0.34	0.21	0.34	0.51
Diffusion	0.04	0.03	0.02	0.04	0.06	0.03
Simulated	0.16	0.15	0.15	0.21	0.21	0.21

Table 5: Average regret after model selection when using various validation estimates. For decision trees, the permutation estimate performs best, and for k -NN, LOO-CV seems best, with the permutation estimate a clear second. For decision trees, 1000 learning episodes were available to all estimates. For k -NN, no limit was placed on the number of learning episodes as learning is “free”. One can compare these results to Table 6, where only 10 learning episodes were available to each method.

is generally inferior.

Table 6 shows the performance of model selection when the number of learning episodes was limited to 10. The permutation and Rademacher results are similar to using a large number of learning episodes (Table 5), a by-product of the concentration of the complexity measures about their expectation. The performance of CV significantly deteriorated. This is to be expected, because the validation estimate is computed using only 10 test points in total; naturally it will be a very unstable estimate. The alternative to LOO with 10 episodes of learning 10-fold cross validation, a very popular technique; this has the advantage of using all the data as test points. The tradeoff is that this 10-fold cross validation estimates the out-sample performance of learning with $\frac{9}{10}$ th of the data, which is only an approximation to learning with all the data, and this introduces a systematic bias. From Table 6, it is not clear which of the two CV measures is better, but the permutation estimate is now dominant. The ability of a method to perform under such limited conditions (very few learning episodes) is important when model construction is expensive and model application is cheap (e.g. Decision Trees). The permutation method works well with just one additional learning episode, producing an estimate using all the data. The permutation estimate also seems to dominate the Rademacher approach because it takes more properties about the distribution into account, without sacrificing on stability.

5 Discussion

Cross validation is general, but may be unstable and computationally intensive, which has led to a variety of alternate validation methods. We presented a permutation estimate (Equations (1), (2) and Theorem 2), which, as with cross validation, is generally applicable, requiring only the ability to learn with a model. It can be applied to: classification or regression; general models and error

Data	n	Decision Trees				k -Nearest Neighbor		
		LOO-CV	10-fold	Perm.	Rad.	LOO-CV	Perm.	Rad.
Abalone	3,132	0.12	0.13	0.02	0.02	0.24	0.09	0.12
Ionosphere	263	0.24	0.21	0.18	0.19	0.49	0.75	0.84
M.Mass	667	0.23	0.13	0.06	0.06	0.15	0.11	0.12
Parkinsons	144	0.25	0.31	0.34	0.40	0.34	0.32	0.44
Pima Indians Diabetes	576	0.18	0.18	0.07	0.07	0.16	0.12	0.15
Spambase	3,450	0.28	0.09	0.07	0.07	0.44	0.43	0.54
Transfusion	561	0.19	0.13	0.08	0.09	0.17	0.12	0.19
WDBC	426	0.31	0.40	0.24	0.37	0.55	0.33	0.50
Diffusion	2,665	0.13	0.04	0.03	0.02	0.09	0.06	0.04

Table 6: Average regret for model selection with number of learning episodes limited to 10. The permutation estimate is now clearly dominant. There is no clear pattern between 10-fold CV (10 learning episodes on $\frac{9}{10}$ th of the data, each being evaluated on the left out $\frac{1}{10}$ th of the data), and 10 learning episodes on $n - 1$ points, each being evaluated on the single left out point.

Noise(%)	LOO-CV	Perm.	Rad.
5	0.30	0.28	0.28
10	0.28	0.27	0.27
15	0.28	0.25	0.25
20	0.28	0.26	0.26
25	0.26	0.25	0.25
30	0.24	0.24	0.24
35	0.24	0.23	0.24
40	0.25	0.26	0.26
45	0.24	0.26	0.26

Table 7: Regret for model selection on simulated data as a function of noise level. For small noise, the permutation and Rademacher methods dominate, with the permutation method slightly better. For large noise, all methods significantly deteriorate to large regret with LOO-CV slightly better.

metrics; any data distribution; and, most importantly to any learning algorithm for a given model. Other data dependent estimates, such as the Rademacher penalty, penalize the complexity of a hypothesis set; the permutation complexity which we also introduced to justify the permutation estimate for empirical error minimization is similar to the Rademacher complexity in that it is a complexity measure for a hypothesis set. The permutation estimate, however, also captures the properties of the learning algorithm (for example regularized empirical risk minimization). The permutation estimate is general, whereas the Rademacher complexity only applies to classification and empirical error minimization (our uniform bound involving the permutation complexity is also similarly restricted).

For classification, with empirical risk minimization, the permutation estimate gives similar guarantees as the Rademacher penalty, and indeed the empirical performances are comparable, though the permutation estimate appears slightly better. We gave a detailed empirical comparison of the permutation estimate with other estimates, including statistical estimates which are tailored for the linear regression setting. The permutation estimate, in most cases, was either the best or comparable to the best for model selection.

When learning is costly, cross validation is prohibitively expensive; we have demonstrated on simulated data, the diffusion data and some standard UCI data sets that the permutation approach gives better results for a fixed (small) number of costly learning episodes. Another way to improve upon LOO-CV when learning is too costly is the K -fold CV procedure; 10-fold is a typical choice. We explicitly compared the 10-fold CV with LOO-CV and with the permutation estimate (also restricted to 10 learning episodes). The permutation estimate dominates. Naturally, our experimental study is nowhere near exhaustive. The tradeoffs between K -fold CV and LOO-CV have been extensively studied elsewhere; the conclusions of these studies are not cut and dry; for example, the choice of K is a sticky point, and there are no clear guidelines. We see here that it is not even clear whether a LOO-CV estimate from only 10 left out test points (corresponding to 10 learning episodes) is dominated by 10-fold CV. Fortunately, with the permutation estimate, we can use all the data in constructing the validation estimate. Further, we gave theoretical guarantees, and in experiments, the permutation estimate is comparable or better than CV, especially when model construction is costly. Conclusion: the permutation estimate provides a more stable alternative to CV which is as generally applicable, more efficient, and (for classification) comes with a uniform bound. There are numerous directions for further investigation, such as a more detailed comparison with K -fold validation methods, the dependence on n , etc. These are directions for future work.

References

- Akaike, H. (1970). Statistical predictor identification. *Annals Inst. Stat. Math.*, **22**, 203–217.
- Akaike, H. (1974). A new look at the statistical model identification. *IEEE Trans. Aut. Cont.*, **19**, 716–723.
- Asuncion, A. and Newman, D. (2007). UCI machine learning repository.
- Bartlett, P. L. and Mendelson, S. (2002). Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, **3**, 463–482.

- Bartlett, P. L., Boucheron, S., and Lugosi, G. (2002). Model selection and error estimation. *Machine Learning*, **48**, 85–113.
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
- Carmack, P., Sain, S., and Schucany, W. (2002). Permutation testing in multivariate regression trees. Technical Report SMU-TR-304, Southern Methodist University.
- Cherkassky, V. and Mulier, F. (2007). *Learning From Data: Concepts, Theory, and Methods*. Wiley.
- Craven, P. and Wahba, G. (1979). Smoothing noisy data with spline functions. *Numerische Mathematik*, **31**, 377–403.
- Cureton, E. E. (1951). Symposium: The need and means of cross-validation: II approximate linear restraints and best predictor weights. *Education and Psychology Measurement*, **11**, 12–15.
- Efron, B. (2004). The estimation of prediction error: Covariance penalties and cross-validation. *Journal of the American Statistical Association*, **99**(467), 619–632.
- Fromont, M. (2007). Model selection by bootstrap penalization for classification. *Machine Learning*, **66**(2-3), 165–207.
- Giné, E. and Zinn, J. (1984). Some limit theorems for empirical processes. *Annals of Prob.*, **12**, 929–989.
- Golland, P. and Fischl, B. (2003). Permutation tests for classification: Towards statistical significance in image-based studies. In *Information Processing in Medical Imaging*, volume 2732 of *Lecture Notes in Computer Science*, pages 330–341. Springer Berlin / Heidelberg.
- Golland, P., Liang, F., Mukherjee, S., and Panchenko, D. (2005). Permutation tests for classification. *Learning Theory*, pages 501–515.
- Good, P. (2005). *Permutation, parametric, and bootstrap tests of hypotheses*. Springer.
- Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer.
- Kääriäinen, M. and Elomaa, T. (2003). Rademacher penalization over decision tree prunings. In *In Proc. 14th European Conference on Machine Learning*, pages 193–204.
- Katzell, R. A. (1951). Symposium: The need and means of cross-validation: III cross validation of item analyses. *Education and Psychology Measurement*, **11**, 16–22.
- Koltchinskii, V. (2001). Rademacher penalties and structural risk minimization. *IEEE Transactions on Information Theory*, **47**(5), 1902–1914.
- Koltchinskii, V. and Panchenko, D. (2000). Rademacher processes and bounding the risk of function learning. In E. Giné, D. Mason, and J. Wellner, editors, *High Dimensional Prob. II*, volume 47, pages 443–459.

- Larson, S. C. (1931). The shrinkage of the coefficient of multiple correlation. *Journal of Education Psychology*, **22**, 45–55.
- Lindgren, F., Hansen, B., Karcher, W., Sjöström, and Eriksson, L. (1996). Model validation by permutation tests: Applications to variable selection. *Journal of Chemometrics*, **10**(5–6), 521–532.
- Lozano, F. (2000). Model selection using Rademacher penalization. In *Proc. 2nd ICSC Symp. on Neural Comp.*
- Lugosi, G. and Nobel, A. (1999). Adaptive model selection using empirical complexities. *Annals of Statistics*, **27**, 1830–1864.
- Magdon-Ismaïl, M., Mertsalov, K., and Goldberg, M. (2009). Learning to predict diffusion in the blogosphere. (submitted).
- Massart, P. (2000). Some applications of concentration inequalities to statistics. *Annales de la Faculté des Sciences de Toulouse*, **X**, 245–303.
- McDiarmid, C. (1989). On the method of bounded differences. In *Surveys in Combinatorics*, pages 148–188. Cambridge University Press.
- Mosier, C. I. (1951). Symposium: The need and means of cross-validation: I problem and designs of cross validation. *Education and Psychology Measurement*, **11**, 5–11.
- Ng, A. (1997). Preventing "overfitting" of cross-validation data. In *Proc. 14th International Conference on Machine Learning (ICML)*, pages 245–253.
- Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, **1**(1), 81–106.
- Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel Methods for Pattern Analysis*. Camb. Univ. Press.
- Shawe-Taylor, J., Bartlett, P. L., Williamson, R. C., and Anthony, M. (1998). Structural risk minimization over data dependent hierarchies. *IEEE Transactions on Information Theory*, **44**, 1926–1940.
- Stone, M. (1974). Cross validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society*, **36**(2), 111–147.
- Vapnik, V. N. and Chervonenkis, A. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, **16**, 264–280.
- Vapnik, V. N., Levin, E., and Le Cun, Y. (1994). Measuring the VC-dimension of a learning machine. *Neural Computation*, **6**(5), 851–876.
- Wang, J. and Shen, X. (2006). Estimation of generalization error: random and fixed inputs. *Statistica Sinica*, **16**, 569–588.
- Wherry, R. J. (1931). A new formula for predicting the shrinkage of the multiple correlation coefficient. *Annals of Mathematical Statistics*, **2**, 440–457.

Wherry, R. J. (1951). Symposium: The need and means of cross-validation: III comparison of cross validation with statistical inference of betas and multiple r from a single sample. *Education and Psychology Measurement*, **11**, 23–28.

Wiklund, S., Nilsson, D., Eriksson, L., Sjoström, M., Wold, S., and Faber, K. (2007). A randomization test for PLS component selection. *Journal of Chemometrics*, **21**(10-11), 427–439.