Cross-validation: what does it estimate?

A leisurely review of Bates et al. 2021

MIGUEL DE BENITO DELGADO, appliedAI

June 11th, 2021

In Cross-validation: what does it estimate and how well does it do it? *Bates et al. focus on cross validation for error estimation and show in detail that it approximates a different error than the one typically of interest in applications. The authors prove this for a class of linear models, then introduce a nested procedure which targets the quantity of interest for practitioners, and leads to tighter variance estimates and confidence intervals.*

Contents

A supervised machine learning application predicts values $Y \in \mathcal{Y}$ from values $X \in \mathcal{X}$, by fitting a function $\hat{f}$ in some model class $\mathcal{F}$, using data $D := \{(X_i, Y_i)\}_{i=1}^n$. A learning algorithm $\mathcal{A}$ maps $D \mapsto \hat{f} = \hat{f}_D$. Now, given $\hat{f}_D$ and a loss function $l: \mathcal{X} \times \mathcal{Y} \longrightarrow \mathbb{R}^+$, one wishes to estimate its **expected prediction error**

$$\operatorname{Err}(\hat{f}) := \mathbb{E}[l(\hat{f}_D(X), Y)], \tag{1}$$

but here one must be careful with the meaning of the expectation. The function $\hat{f}_D$ itself is random, being the image by $\mathcal{A}$ of a random quantity, $\hat{f}_D = \mathcal{A}(D)$, so the above spelled out is:

$$\operatorname{Err}(\hat{f}) = \mathbb{E}_{D \sim \mathbb{P}^n_{X,Y}}[\mathbb{E}_{X,Y}[l(\hat{f}_D(X), Y) | D]].$$

To be more precise, $\operatorname{Err}(\hat{f})$ is a function of the tuple $(\mathcal{A}, l, n, \mathbb{P}_{XY})$. It is the expected error on future test points, when training on any data set *of size* n drawn i.i.d. from the same distribution.[1] This quantity is of interest e.g. when designing learning algorithms, but it is not what a practitioner usually wants. Instead, for applications one is interested in the error that some specific $\hat{f}_D$, trained on a fixed dataset $D = \{(x_i, y_i)\}_{i=1}^n$, will incurr when deployed on unseen data, or what is usually called **generalization error** or **prediction error**:

$$\operatorname{Err}_{XY}(\hat{f}) := \mathbb{E}_{X,Y}[l(\hat{f}(X), Y) | D] \tag{2}$$

(here and in the sequel we will ommit the subindex in $\hat{f}_D$, as well as the dependency on $\mathcal{A}, n, l$ and the distribution of X, Y for brevity). Note

[1] The number of samples matters: estimators using subsets of the data, e.g. K-fold cross-validation, will be biased for the errors of algorithms trained on data sets of full size n.

that we assume $\hat{f}_D$ to be the outcome of a model fitting procedure, and not of model selection. More on this below.

Because the true distribution for X, Y is unknown, estimates for these quantities are required. Most methods can be roughly classified as **resampling based** or **analytic**, see Figure 1. Here we focus on CV, but the results of Section 2 extend to most of them (for linear models).

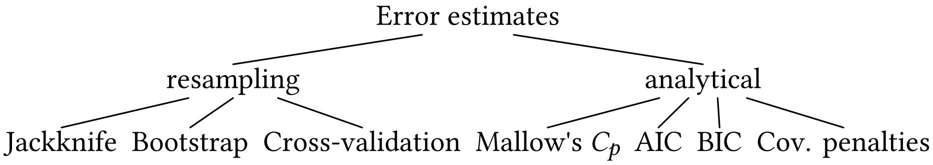

Figure 1. Common approaches to the estimation of different prediction errors.

1 A note on model selection

It is important to note that we are always assuming that no model selection procedure is performed using the training data, i.e. no feature selection, hyperparameter tuning nor selection of an imputation procedure. Otherwise, there would be correlations between the selected model, which is a random variable itself, and any statistics used in inference later, e.g. confidence intervals, hypothesis tests or error estimates.[2] Additionally, it is well known that the CV estimate of the error *of the model chosen by a selection procedure* is strongly biased downwards, because the statistics of the extreme value $\min\{\widehat{\mathrm{Err}}_1, \ldots, \widehat{\mathrm{Err}}_k\}$ are different from those of the $\widehat{\mathrm{Err}}_j$.

The simplest approach for simultaneous model selection and error estimation is a simple train - test split of the data. But this comes at cost of both bias and variance because of the reduced data on which the models are chosen and the winner finally evaluated. If computational resources allow for it, it is common to use a nested CV procedure, which has great computational cost, and for which there are no known theoretical guarantees. There is some work attempting to debias the minimum in [TT09], and more recently [Gua18], as well as a growing body of research on valid *post-selection inference* for specific algorithms which enable inference at lower computational cost, and with guarantees.[3] More on these topics will follow on our website.

We focus then on CV for error estimation, but what error are we talking about?

2 What cross validation estimates

K-fold CV first splits D into equally sized subsets I_j, $j \in \{1, \ldots, K\}$. We assume that K divides n for simplicity, and write $i \in I_j$ for $(x_i, y_i) \in I_j$.

[2] Consider a regression problem in which the data is used to do feature selection. Intuitively, in a significance test for the chosen variables, these are necessarily going to be significant for the data that was used to select them in the first place.

[3] **Post-selection inference** refers to a set of techniques enabling statistical inference (e.g. computing confidence intervals) after doing model selection using the data.

Then it trains a function $\hat{f}^{(-j)}$ on all but I_j, and evaluates it on I_j with:

$$\mathrm{CV}_j(\hat{f}) := \frac{K}{n} \sum_{i \in I_j} l(\hat{f}^{(-j)}(x_i), y_i).$$

The CV error is:

$$\widehat{\mathrm{Err}}^{\mathrm{cv}}(\hat{f}) := \frac{1}{K} \sum_{j=1}^{K} \mathrm{CV}_j(\hat{f}). \tag{3}$$

A subtle question is what exactly does $\widehat{\mathrm{Err}}^{\mathrm{cv}}$ estimate, and the widely accepted answer for over two decades is that it is Err and not Err_{XY}. Intuitively, CV_j, the inner sum in (3), is an estimate for Err_{XY} for fixed I_j, and the outer sum estimates Err, with the I_j being different samples from the data set distribution.[4] Despite this being known, detailed analyses of the mismatch between $\widehat{\mathrm{Err}}^{\mathrm{cv}}$ and Err_{XY} have been lacking, and one of the results of this paper is a rigorous treatment of this question in the linear case. Specifically, they show that CV for a certain type of problem has the property that $\widehat{\mathrm{Err}}^{\mathrm{cv}}$ *is independent of* Err_{XY} *given* $\boldsymbol{X}$ [BHT21, Theorem 1]. In other words:

THEOREM 1. *For certain model classes,*[5] *given two datasets* $D = \{(X_i, Y_i)\}_{i=1}^n, D' = \{(X_i, Y_i')\}_{i=1}^n$ *sampled from the same distribution and with the same feature matrix* $\boldsymbol{X}$*, and* Err_{XY} *and* $\mathrm{Err}_{XY'}$ *being the true errors and* $\widehat{\mathrm{Err}}^{\mathrm{cv}}_{XY}$ *and* $\widehat{\mathrm{Err}}^{\mathrm{cv}}_{XY'}$ *their CV estimates, one has:*

$$(\mathrm{Err}_{XY}, \widehat{\mathrm{Err}}^{\mathrm{cv}}_{XY}) \overset{d}{=} (\mathrm{Err}_{XY}, \widehat{\mathrm{Err}}^{\mathrm{cv}}_{XY'}).$$

> *«This means that for the purpose of estimating* Err_{XY}*, we have no reason to prefer using the cross-validation estimate with* $(\boldsymbol{X}, \boldsymbol{Y})$ *to using the cross-validation estimate with a different data set* $(\boldsymbol{X}, \boldsymbol{Y}')$*, even though we wish to estimate the error of the model fit on* $(\boldsymbol{X}, \boldsymbol{Y})$*»*

The consequence derived from this is that $\widehat{\mathrm{Err}}^{\mathrm{cv}}$ is a better estimate of an intermediate quantity Err_X than of Err_{XY}.

COROLLARY 2. *Under the conditions of Theorem 1,*

$$\mathbb{E}[(\widehat{\mathrm{Err}}^{\mathrm{cv}} - \mathrm{Err}_{XY})^2] \geqslant \mathbb{E}[(\widehat{\mathrm{Err}}^{\mathrm{cv}} - \mathrm{Err}_X)^2],$$

where $\mathrm{Err}_X := \mathbb{E}_Y[\mathrm{Err}_{XY} | X]$.

[4] The estimate for Err is for data sets of size $n - n/K$, and typically slightly biased upwards for data sets of size n.

[5] *Linear Gaussian with constant noise variance, fitted with squared loss.*

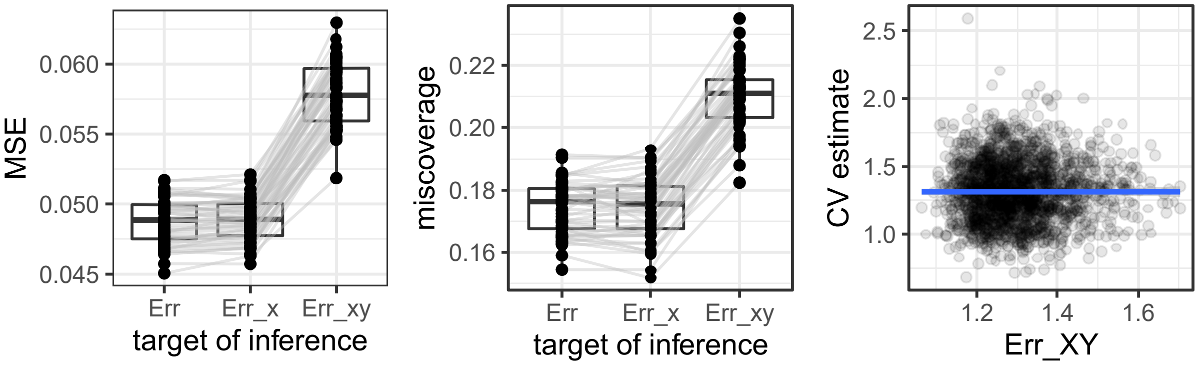

Figure 2. [BHT21, Figure 3]. Left: mean squared error of the CV point estimate of prediction error relative to three different estimands: Err, Err_X, and $\mathrm{Err}_{\mathrm{XY}}$. Center: coverage of Err, Err_X, and $\mathrm{Err}_{\mathrm{XY}}$ by the naive cross-validation intervals in a homoskedastic Gaussian linear model. The nominal miscoverage rate is 10%. Each pair of points connected by a line represents 2000 replicates with the same feature matrix $\boldsymbol{X}$. Right: 2000 replicates with the same feature matrix and the line of best fit (blue).

Finally, two additional corollaries yield the conclusions of the whole section, namely:[6]

- $\widehat{\mathrm{Err}}^{\mathrm{cv}}$ is uncorrelated with Err_{XY}, in a certain asymptotic sense.
- **CV has larger error for estimating Err_{XY} than for estimating Err or Err_X.**

Other estimators. The theory developed in Sections 2 and 3 of the paper applies not only to CV but to other methods as well: data splitting with refitting (train on one half of the data, evaluate on the other, then refit on the whole data set), Mallow's C_p and bootstrap.

3 Computing standard errors

Besides a point estimate for the error one always wants an indication of the trust that can be placed in it, i.e. a confidence interval. In order to do so, an estimate of the standard error is required. For applications one wants a confidence interval for Err_{XY} and for algorithm design or benchmarking, one for Err. Note that even if $\mathrm{Err} = \mathbb{E}[\mathrm{Err}_{XY}]$ and $\widehat{\mathrm{Err}}^{\mathrm{cv}}$ is (almost) unbiased for Err, whether it estimates one or the other matters for what it is that its sampling variance informs us about.

Let $e_i := l\left(f^{(-I_{j(i)})}(X_i), Y_i\right)$ be the error for sample X_i, Y_i when trained on the $K-1$ folds not containing it. Then

$$\mathrm{Var}(\widehat{\mathrm{Err}}^{\mathrm{cv}}) = \mathrm{Var}\left(\frac{1}{n}\sum_{i=1}^{n} e_i\right) \approx \frac{1}{n} e_1, \tag{4}$$

[6] These results hold only for a linear problem with no regularization. **The authors expect CV to be closer to Err_{XY} when regularization is applied.**

where equality would hold *only if the e_i where i.i.d.* So the sample estimate for the standard error of the point estimate $\widehat{\mathrm{Err}}^{\mathrm{cv}}$ is:

$$\widehat{\mathrm{se}}^{\mathrm{cv}} = \frac{1}{\sqrt{n}} \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (e_i - \widehat{\mathrm{Err}}^{\mathrm{cv}})^2}. \tag{5}$$

With this, the standard confidence interval would be

$$(\widehat{\mathrm{Err}}^{\mathrm{cv}} - z_{1-\alpha/2}\,\widehat{\mathrm{se}}^{\mathrm{cv}}, \widehat{\mathrm{Err}}^{\mathrm{cv}} + z_{1-\alpha/2}\,\widehat{\mathrm{se}}^{\mathrm{cv}}). \tag{6}$$

where $z_{1-\alpha/2}$ is the $1-\alpha/2$ quantile of the standard normal distribution, e.g. 1.96 for a 95% CI.[7]

However, since every point in D is used both for training and testing, the independence assumption in (4) does not hold and the closer K is to its maximum n, the higher the correlations will be, hence the total true variance. Any confidence interval built using (4) and (5) will be too small and have poor coverage.

Even if one does not do CV, but is in the optimal situation of having K independent training sets and L independent test sets, because for each of the training sets many errors will be computed, these errors will not be conditionally independent, given one training set. This will make the naive estimate (5) too optimistic. In the extreme case of leave-one-out CV, every sample is used n times for training and testing, leading to even higher sampling variance.

This was shown in the fundamental paper [BG04], where the main result was a proof that any estimate of the variance of CV is biased and the nature of the bias is neither determined by the number of folds nor the sample size. The key is the structure of the covariance matrix of the errors e_i, see Figure 3. [BG04] show that it is only parametrised by three numbers,

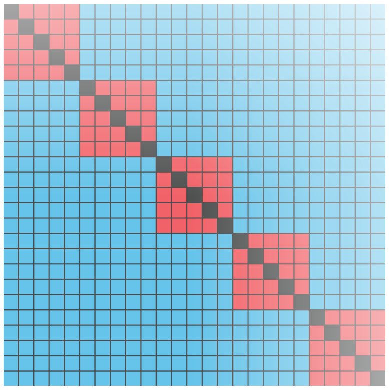

Figure 3. [BHT21, Figure 7]. «Covariance structure of the CV errors. Red entries correspond to the covariance between points in the same fold, and blue entries correspond to the covariance between points in different folds.»

$$\begin{aligned} \mathrm{Var}(\widehat{\mathrm{Err}}^{\mathrm{cv}}) &= \frac{1}{n^2} \sum_{i,j=1}^{n} \mathrm{Cov}(e_i, e_j) \\ &= \frac{1}{n}\sigma^2 + \left(\frac{1}{K} - \frac{1}{n}\right)\omega + \left(1 - \frac{1}{K}\right)\gamma, \end{aligned} \tag{7}$$

where σ^2 is the variance of e_i when D has size $n - n/K$, ω is the **in-block covariance** of errors due to a common training set, and γ is the **between-blocks covariance** due to the dependence between training sets I_k. Note how increasing K deteriorates the third term, and increasing n the second one. There are no constraints as to how ω and γ behave, other than being at most $|\sigma^2|$, and in practice both are seen to be positive. Therefore, typically

$$\mathrm{Var}(\widehat{\mathrm{Err}}^{\mathrm{cv}}) > \frac{1}{n}\mathrm{Var}(e_1),$$

[7] Because in (4) we assumed that the e_i were (roughly) independent, $\widehat{\mathrm{Err}}^{\mathrm{CV}}$ is asymptotically normal by the CLT and this confidence interval is the natural one.

and (5) will underestimate the standard error. Furthermore, γ is seen to be of order σ^2, especially in the presence of outliers, irrespective of the number of folds [BG04, Section 7], see Figure 4. So using (6) for decisions is fraught with dangers in practical cases.

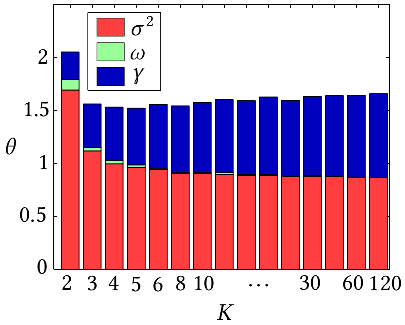

Figure 4. [BG04] Figure 5b. contributions of σ^2, ω and γ to $\mathrm{Var}(\widehat{\mathrm{Err}}^{\mathrm{CV}})$ vs number of folds K, in the presence of outliers.

In order to address the issue, one must then either modify CV with new sampling / splitting schemes, or add distributional assumptions. Recent work has proposed splitting the data in half, then doing CV in different variants, but this typically proves to be either too costly or too conservative in the estimates. Instead, [BHT21] modify CV with a nested procedure.

The definition of $\widehat{\mathrm{se}}$. In practice, instead of (5), another rougher approximation is typically done. Once the K values $\mathrm{CV}_j(\hat{f})$ have been computed, one can argue that

$$\begin{aligned}\mathrm{Var}(\widehat{\mathrm{Err}}^{\mathrm{cv}}) &= \mathrm{Var}\left(\frac{1}{K}\sum_{j=1}^{K}\mathrm{CV}_j(\hat{f})\right)\\ &\approx \frac{1}{K}\mathrm{Var}(\mathrm{CV}_j(\hat{f})),\end{aligned} \tag{8}$$

where equality would hold if the CV_j where i.i.d. The sample estimate for the standard error is

Equation (9) is just `scores.std()/sqrt(K)`

$$\widehat{\mathrm{se}}^{\mathrm{cv}} = \frac{1}{\sqrt{K}}\sqrt{\frac{1}{K-1}\sum_{j=1}^{K}(\mathrm{CV}_j - \widehat{\mathrm{Err}}^{\mathrm{cv}})^2}, \tag{9}$$

and the standard confidence interval:

$$(\widehat{\mathrm{Err}}^{\mathrm{cv}} - z_{1-\alpha/2}\,\widehat{\mathrm{se}}^{\mathrm{cv}}, \widehat{\mathrm{Err}}^{\mathrm{cv}} + z_{1-\alpha/2}\,\widehat{\mathrm{se}}^{\mathrm{cv}}). \tag{10}$$

Now, this is a different interval than (6) and given the relatively low value of K, the normality assumption is less justified than before. But, crucially it has the same problem of not taking dependencies into account and therefore having necessarily poor coverage.

4 Nested cross validation

A word on nomenclature. Nested CV is typically used in machine learning for simultaneous model selection and error estimation, as suggested in [VS06]. This is *not* what is done here.

As we explained above, the problem with the confidence interval (6) is that the true variance of $\widehat{\mathrm{Err}}^{\mathrm{cv}}$ is higher than the one in (5) because of correlations in the errors introduced by using samples both for training and testing. In the naive approximation, one assumes that $\omega = 0$ and $\gamma = 0$ in (7), but we have seen in Figure 4 that in particular the covariances γ arising from dependencies across folds, can be of the same magnitude as the variance σ^2.

The goal of the authors is to estimate the Mean Squared Error of CV:

$$\mathrm{MSE}_{K,n} := \mathbb{E}[(\widehat{\mathrm{Err}}^{\mathrm{cv}} - \mathrm{Err}_{XY})^2],$$

which because of the usual bias - variance decomposition and the low bias of $\widehat{\text{Err}}^{\text{cv}}$ is a reasonable (and conservative) proxy for $\text{Var}(\widehat{\text{Err}}^{\text{cv}})$. Note that despite the results of Section 2, the authors pick MSE wrt. Err_{XY}. This is both because of technical reasons, and the fact that Err_{XY} is what typically matters for practical applications.

In order to introduce the algorithm we consider first a single split of the data into $D_{\text{train}} = \{(X_i, Y_i)\}_{i \in I_{\text{train}}}$ and $D_{\text{out}} = \{(X_i, Y_i)\}_{i \in I_{\text{out}}}$. For $i \in I_{\text{train}}$ we write $(\tilde{X}_i, \tilde{Y}_i) = (X_i, Y_i)$ and $\hat{f}$ is obtained after training on D_{train}. Exactly as in (2) we define, considering D_{train} as a r.v.

$$\text{Err}_{\tilde{X}\tilde{Y}}(\hat{f}) := \mathbb{E}_{X,Y}[l(\hat{f}(X), Y) | D_{\text{train}}],$$

and the key is then the following decomposition, for any estimate $\widehat{\text{Err}}_{\tilde{X}\tilde{Y}}$ of this quantity, which is obtained only using D_{train}:

Lemma 3. ([BHT21, Holdout MSE])

$$\begin{aligned}\mathbb{E}_{D_{\text{train}}}[(\widehat{\text{Err}}_{\tilde{X}\tilde{Y}} - \text{Err}_{\tilde{X}\tilde{Y}})^2] &= \underbrace{\mathbb{E}_D[(\widehat{\text{Err}}_{\tilde{X}\tilde{Y}} - \bar{e}_{\text{out}})^2]}_{(a)} \\ &\quad - \underbrace{\mathbb{E}_D[(\bar{e}_{\text{out}} - \text{Err}_{\tilde{X}\tilde{Y}})^2]}_{(b)}.\end{aligned}$$

It is possible to estimate (a) and (b) without bias, hence the left hand side as well.

Algorithm [BHT21, Nested Cross Validation]

1. Split D into K folds with $K-1$ building D_{train} and the remaining one being D_{out}, with indices I_{out}.
2. For each split j:
   a. Compute $\varepsilon_j := \bar{e}_{\text{in}} = \widehat{\text{Err}}_{\tilde{X}\tilde{Y}}$ with $(K-2)$-fold CV over the $K-1$ folds in D_{train}.
   b. Train model on D_{train}. Compute errors e_i for all samples $(x_i, y_i) \in D_{\text{out}}$.
   c. Compute $\bar{e}_{\text{out}} :=$ mean of $\{e_i\}_{i \in I_{\text{out}}}$.
   d. Set $a_j := (\widehat{\text{Err}}_{\tilde{X}\tilde{Y}} - \bar{e}_{\text{out}})^2$ (estimate of (a)).
   e. Set $b_j :=$ empirical variance of $\{e_i\}_{i \in I_{\text{out}}}$ (estimate of (b)).
3. Output $\widehat{\text{MSE}} := \text{mean}(a_j) - \text{mean}(b_j)$.
4. Output $\widehat{\text{Err}}^{\text{ncv}} := \text{mean}(\varepsilon_j)$.

The key result is that Algorithm 3 is a good approximation of the MSE of CV (for a smaller sample size):

Theorem 4. ([BHT21, Estimand of nested CV]) *For a nested CV with a sample of size n,*

$$\mathbb{E}[\widehat{\text{MSE}}] = \text{MSE}_{K-1,n-n/K}.$$

As noted before, the fact that estimation happens on $K-1$ folds introduces some bias for the actual quantity of interest $\text{MSE}_{K,n}$. One can rescale $\widehat{\text{MSE}}$ with the factor $\frac{K-1}{K}$, although is just a heuristic with no theoretical guarantee.

For the same reason, $\widehat{\text{Err}}^{\text{ncv}}$ is unbiased for Err with data set of size $n\,(K-2)/K$ but biased for size n. A debiasing strategy is suggested with the estimator:

$$\hat{b}^{\text{ncv}} := \left(1 + \frac{K-2}{K}\right)(\widehat{\text{Err}}^{\text{ncv}} - \widehat{\text{Err}}^{\text{cv}}),$$

and finally, the confidence interval for Err becomes

$$(\widehat{\text{Err}}^{\text{ncv}} - \hat{b}^{\text{ncv}} - z_{1-\alpha/2}\,\widehat{\text{se}}^{\text{ncv}}, \widehat{\text{Err}}^{\text{ncv}} - \hat{b}^{\text{ncv}} + z_{1-\alpha/2}\,\widehat{\text{se}}^{\text{ncv}}),$$

with $\widehat{\text{se}}^{\text{ncv}} := \sqrt{\frac{K-1}{K}\widehat{\text{MSE}}}$.

On computational cost. Obviously, the amount of computation required to produce $\widehat{\text{Err}}^{\text{ncv}}$ can be orders of magnitude higher than simple CV, depending on the number of folds. Because of the embarrassingly parallel nature of the problem this can be a minor issue for applications where CV itself runs rather quickly, but makes nested CV inpracticable when it does not.

5 Results and conclusions

The authors test Nested CV for classification and regression with synthetic and real data sets. For binary classification, they use a sparse logistic data generating process $\mathbb{P}(Y_i = 1|X_i = x_i) = \sigma(-x_i^{\top}\theta)$, for $i \in \{1,..., n\}$, $X_i \sim \mathcal{N}(0, I_p)$, and $\theta := c\,(1,1,1,1,0,\ldots,0) \in \mathbb{R}^p$, $c > 0$ chosen to obtain different Bayes risks.[8] These are optimal lower bounds for Err.

In the low dimensional regime ($p = 20, n = 100$), with c chosen to have a Bayes error of 33% and an unregularized logistic regression model, they obtain miscoverage rates for Err and Err_{XY} close to the nominal values of 10%, as seen in Table 1.

Table 1. ([BHT21, Table 1, excerpt]) Performance of cross-validation (CV) and nested cross-validation (NCV) for low-dim logistic regression. *«A "Hi" miscoverage is one where the confidence interval is too large and the point estimate falls below the interval; conversely for a "Lo" miscoverage. The standard error in each coverage estimate reported is about 0.5%. The "Target" column indicates the target of coverage. The intervals are always generated identically(…)»*

Target	Point estim.		Miscoverage			
			CV		NCV	
	CV	NCV	Hi	Lo	Hi	Lo
Err_{XY}	39.6%	39%	10%	8%	**5%**	**3%**
Err	”	”	9%	8%	**3%**	**4%**

[8] After some computation one finds the Bayes risk to be $2\int_{\{\theta\cdot x\leqslant 0\}}\sigma(-\theta\cdot x)\,\mathcal{N}(x;0,I_p)\,\mathrm{d}x$. Solving for θ provides the required value of c.

Similar experiments in the high dimensional setting ($n \in \{90, 200\}$, $p = 1000$), with ℓ_1 regularization shows that the intervals obtained through NCV show miscoverage much closer to nominal with "high" and "low" rates of 3 to 6% each for NCV as opposed to 10 to 20% for CV.

In the linear regression setting, with $p = 20$ and $n > 100$ and a model fitted with ordinary least squares, NCV provides again better coverage than CV up to dimension around 400. From here on, both methods perform similarly. As a matter of fact, several recent asymptotic results in the literature make it expected that the violations in coverage vanish as $n \longrightarrow \infty$ for fixed p. As a general rule of thumb, **one can expect CV to perform well when n/p is large and regularization is used**, so NCV should be used in high dimensional settings or low data regimes.

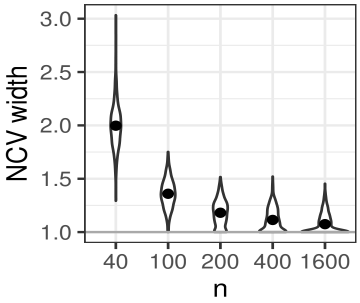

Figure 5. ([BHT21, Fig. D.2]) *«Width of nested CV intervals relative to the width of the naive CV intervals»*

Finally, a high dimensional sparse linear model with a lasso estimator repeats the conclusions, although both methods fail at very low sample numbers.

Further experiments with real data sets use demographic and radar measurements again with similar results. Please see section 6 of the paper.

BIBLIOGRAPHY

[BG04] Yoshua Bengio and Yves Grandvalet. No Unbiased Estimator of the Variance of K-Fold Cross-Validation. *The Journal of Machine Learning Research*, 5:1089–1105, dec 2004.

[BHT21] Stephen Bates, Trevor Hastie, and Robert Tibshirani. Cross-validation: what does it estimate and how well does it do it? Apr 2021.

[Gua18] Leying Guan. Test Error Estimation after Model Selection Using Validation Error. *ArXiv:1801.02817 [stat]*, feb 2018.

[TT09] Ryan J. Tibshirani and Robert Tibshirani. A bias correction for the minimum error rate in cross-validation. *The Annals of Applied Statistics*, 3(2):822–829, jun 2009.

[VS06] Sudhir Varma and Richard Simon. Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics*, 7(1):91, feb 2006.