Cross-Validation Error Dynamics in Smaller Datasets

Abstract

Cross-validation (CV) is the de facto standard for estimating a model's generalization performance, but in smaller datasets it exhibits an underappreciated quirk: across folds, training and test errors are strongly negatively correlated, while training and holdout errors show a moderate anticorrelation and test versus holdout errors are essentially uncorrelated. Herein, we document these phenomena empirically—on both real and synthetic datasets under AdaBoost—and introduce a simple generative model that explains them. By viewing each CV split as hypergeometric sampling from a finite population and incorporating an overfitting parameter δ that shifts expected errors on train, test, and holdout sets, we derive closed-form expressions for the covariances among observed error rates. Our analysis shows that sampling-induced anticorrelation dominates in small datasets, while overfitting contributes an additional negative term, thus accounting for the observed error dynamics. We discuss the limitations of our approach and suggest directions for more refined models and extensions to regression settings.

1. Introduction

Cross-validation (CV) is a long-standing widely-used (Kohavi, 1995; Stone, 1974) technique to estimate and optimize a model's generalization performance by repeatedly splitting data into training and test sets (see (Arlot & Celisse, 2010) for a survey). While cross-validation has some known pitfalls (Dietterich, 1998), we herein discuss an underappreciated phenomenon that occurs especially in smaller datasets (containing, say, thousands, not millions of examples): that, across CV splits, training error and test error tend to be negatively correlated—splits where a model fits the training data especially well often yield higher test error, and vice versa. We have also observed a more moderate negative correlation between training and holdout error, as well as almost no correlation between test error and holdout error. These phenomena are of course related to others that have previously been studied, especially in work accounting for the the variance in the test and training splits (Nadeau & Bengio, 2003) and on correlations across splits (Bates et al., 2024; Bayle et al., 2020).

In this short paper, we introduce a simple model that accounts for these patterns. We show how hypergeometric sampling (sampling from a finite population without replacement) of these types induces the observed anti-correlation between training and test splits, and we show how a model's tendency to overfit to its training data can also account for the negative correlation observed between test and holdout error.

2. An empirical finding

In experimenting with cross-validation on smaller datasets, we repeatedly observed a strong anti-correlation between training and test errors across runs. This phenomenon seemed surprising at first, though it is easily explainable (as we discuss in the next section). Moreover, we saw a moderate anti-correlation between training and holdout errors and a negligible anti-correlation between test and holdout.

Our experiments first chose a holdout set and then repeatedly split the remaining data into training and test sets, with the holdout set fixed. Figure 1 shows a graphical display of these phenomena on the 'Adult' UCI dataset, run on AdaBoost. Figure 2 validates this on synthetic data. The aim of this short paper is to discuss this experimental phenomenon and come up with a convincing model to explain it.

3. A theoretical explanation

In this section, we give a simple generative model that explains these experimental results. In addition to explaining the findings, the model illustrates that this is a "small data" issue, as will be explained presently.

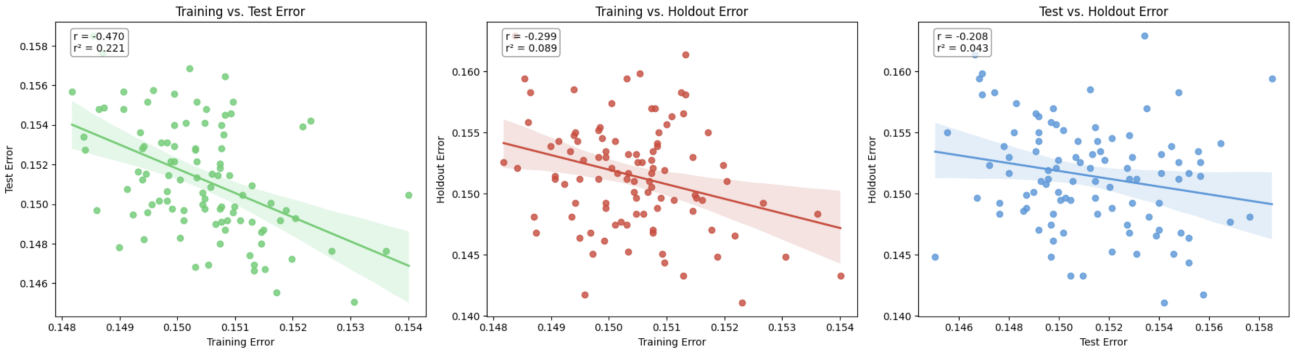

Figure 1. Results from 100-fold AdaBoost Cross-Validation ran on Census data, 'Adult', from UCI's Machine Learning Repository (Kohavi & Becker, 1996). We used decision stumps with 50 base learners in our AdaBoost model. We then performed 100-fold cross validation. There are slightly over 48 thousand instances and 15 features in this data set. The data was partitioned into train, test, and hold-out as follows: initially 10% was reserved as the hold-out set, then 67.5% of the data was devoted to the training set and the remaining 22.5% to the test set.

Figure 2. Synthetic data generated from 10 variables then taking a weighted linear combination of 3 variables making them determine the target label for each instance. Then randomly flipping the bits of 30% of the data, the data contained 1,000 instances. We then performed 100-fold cross validation AdaBoost, with 50 base learners. There were a thousand instances generated and 10 features for this data set. The data was partitioned into train, test, and hold-out as follows: initially 10% was reserved as the hold-out set, then 67.5% of the data was devoted to the training set and the remaining 22.5% to the test set.

3.1. Model

Let N be the total number of data points. Reserve a fixed hold-out set of size H, leaving $M = N - H$ points for cross-validation. On each replicate, draw without replacement n training points from the M, and use the remaining $M - n$ for the CV test fold.

We now let $\delta > 0$ be an over-fitting parameter and μ the baseline error. Define expected (mean) errors for the folds as follows:

$$p_{\text{train}}(\delta) = \mu - \alpha\,\delta, \quad p_{\text{test}}(\delta) = \mu + \beta\,\delta, \quad p_{\text{hold}}(\delta) = \mu + \beta\,\delta,$$

with $\alpha \gg \beta > 0$. Increasing δ lowers train-error by $\alpha\delta$ and raises both test- and holdout-error by $\beta\delta$. Conditioned on δ, the classifier would misclassify $K(\delta) = M\,p_{\text{train}}(\delta) = M\big(\mu - \alpha\delta\big)$ of the M CV points.

We note that the parameters α and β should depend on the training set size—in most machine learning settings β depends on the complexity of the classifier classifier, but also decays as $1/\sqrt{n}$ (Vapnik, 1998). A model's tendency to overfit the training data diminishes as the size of the training dataset increases.

3.2. Expected observed outcomes

We define $\hat{p}_{\text{train}}, \hat{p}_{\text{test}}$, and $\hat{p}_{\text{hold}}$ to represent the observed training, test, and holdout error rates on the respectively.

Let X_{train} be the training sample. Our process draws $X_{\text{train}} \sim \text{Hypergeometric}\big(M,\ K(\delta),\ n\big)$.

$$\hat{p}_{\text{train}} = \frac{X_{\text{train}}}{n},$$

where $\mathbb{E}[\hat{p}_{\text{train}} \mid \delta] = p_{\text{traing}}(\delta) = \mu - \alpha\delta$.

We combine the leftover "hard" points with an explicit bias shift so that $\mathbb{E}[\hat{p}_{\text{test}} \mid \delta] = p_{\text{test}}(\delta) = \mu + \beta\delta$:

$$\hat{p}_{\text{test}} = \underbrace{\frac{K(\delta) - X_{\text{train}}}{M - n}}_{\substack{\text{sampling} \\ \text{noise}}} + \underbrace{\big[p_{\text{test}}(\delta) - p_{\text{train}}(\delta)\big]}_{\substack{\text{bias} \\ \text{shift}}}.$$

Note that by construction of the bias term,

$$\begin{aligned}\mathbb{E}[\hat{p}_{\text{test}} \mid \delta] &= p_{\text{train}}(\delta) + \big[p_{\text{test}}(\delta) - p_{\text{train}}(\delta)\big] \\ &= p_{\text{test}}(\delta) \\ &= \mu + \beta\delta.\end{aligned}$$

On the fixed hold-out set we take

$$\begin{aligned}\mathbb{E}[\hat{p}_{\text{hold}}|\delta] &= p_{\text{hold}}(\delta) \\ &= \mu + \beta\,\delta.\end{aligned}$$

3.3. Analysis of correlations

We can now compute the expected correlations resulting from this model. We do this by examining the covariances of the relevant quantities. By the law of total covariance,

1. Train vs. Test:

$$\text{Cov}\big(\hat{p}_{\text{train}}, \hat{p}_{\text{test}}\big) = \underbrace{\mathbb{E}_\delta\big[\text{Cov}(\hat{p}_{\text{train}}, \hat{p}_{\text{test}} \mid \delta)\big]}_{\text{(I)}} + \underbrace{\text{Cov}\big(\mathbb{E}[\hat{p}_{\text{train}} \mid \delta],\, \mathbb{E}[\hat{p}_{\text{test}} \mid \delta]\big)}_{\text{(II)}}.$$

(I):

$$\begin{aligned}\text{Cov}\big(\hat{p}_{\text{train}}, \hat{p}_{\text{test}} \mid \delta\big) &= \text{Cov}\Big(\tfrac{X_{\text{train}}}{n},\ \tfrac{X_{\text{test}}}{M-n} \,\Big|\, \delta\Big) \\ &= \frac{1}{n\,(M-n)}\ \text{Cov}\big(X_{\text{train}},\ X_{\text{test}} \mid \delta\big) \\ &= -\frac{1}{n\,(M-n)}\ \text{Var}\big(X_{\text{train}} \mid \delta\big).\end{aligned}$$

(II):

$$\begin{aligned}\text{Cov}\big(p_{\text{train}}(\delta),\, p_{\text{test}}(\delta)\big) &= \text{Cov}\big(\mu - \alpha\delta,\ \mu + \beta\delta\big) \\ &= -\,\alpha\beta\ \text{Var}(\delta)\end{aligned}$$

Hence

$$\begin{aligned}\text{Cov}(\hat{p}_{\text{train}}, \hat{p}_{\text{test}}) &= -\mathbb{E}_\delta\Big[\tfrac{1}{n(M-n)}\ \text{Var}(X_{\text{train}} \mid \delta)\Big]\ -\ \alpha\beta\ \text{Var}(\delta) \\ &\ll 0.\end{aligned}$$

We note that the first term is especially large (negative) for smaller dataset sizes and essentially disappears for very large ones.

2. Train vs. Holdout:

$$\begin{aligned}\mathrm{Cov}(\hat{p}_{\mathrm{train}}, \hat{p}_{\mathrm{hold}}) &= \mathrm{Cov}\big(\tfrac{X_{\mathrm{train}}}{n},\ \mu + \beta\delta\big) \\ &= (\mu - \alpha\delta, \mu + \beta\delta) \\ &= -\alpha\beta\ \mathrm{Var}(\delta) \\ &< 0.\end{aligned}$$

3. Test vs. Holdout:

$$\begin{aligned}\mathrm{Cov}(\hat{p}_{\mathrm{test}}, \hat{p}_{\mathrm{hold}}) &= Cov(\mu + \beta\delta, \mu + \beta\delta) \\ &= \beta^2\ \mathrm{Var}(\delta) \\ &> 0 \quad (\approx 0 \text{ for } \beta \text{ small}).\end{aligned}$$

We conclude that under the modeling assumptions in Section 3, we expect 1) the training and test errors to be strongly anticorrelated due to sampling anticorrelation plus overfitting, 2) the training and holdout errors to be moderately anticorrelated due to some degree of overfitting, and finally the test and holdout errors to be slightly correlated. All three rely on an overfitting parameter δ that depends on the training set each round.

4. Conclusions

In this paper, we discussed a phenomenon in cross-validation that we feel should be better known, and we gave a simple mathematical model that partially explains it. This model has some limitations: first the "bias shift" term makes the formulas agree, but fails to account for the "hard examples" that may fall out of the training set and into the test set. A more nuanced approach may be more compelling. Further, our model predicts a mild anti-correlation between test and holdout errors, which is not seen, again pointing to the need for a more refined analysis. Finally, it would be interesting to explore tasks other than classification (e.g. regression) in future work.

References

Arlot, S. and Celisse, A. A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4:40–79, 2010.

Bates, S., Hastie, T., and Tibshirani, R. Cross-validation: What does it estimate and how well does it do it? *Journal of the American Statistical Association*, 119(546):1434–1445, 2024.

Bayle, P., Bayle, A., Janson, L., and Mackey, L. Cross-validation confidence intervals for test error. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Dietterich, T. G. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923, 1998.

Kohavi, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1137–1145, 1995.

Kohavi, R. and Becker, B. Adult data set, 1996. URL `https://archive.ics.uci.edu/ml/datasets/adult`.

Nadeau, C. and Bengio, Y. Inference for the generalization error. *Machine Learning*, 52(3):239–281, 2003.

Stone, M. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society, Series B*, 36(2):111–147, 1974.

Vapnik, V. N. *Statistical Learning Theory*. Wiley-Interscience, New York, 1998. ISBN 978-0-471-03003-4.