On Cross-Validation and Stacking: Building seemingly predictive models on random data

Claudia Perlich
Media6°
New York, NY 10012
claudia@media6degrees.com

Grzegorz Świrszcz
IBM T.J. Watson Research Center
Yorktown Heights, NY 10598
swirszcz@us.ibm.com

ABSTRACT

A number of times when using cross-validation (CV) while trying to do classification/probability estimation we have observed surprisingly low AUC's on real data with very few positive examples. AUC is the area under the ROC and measures the ranking ability and corresponds to the probability that a positive example receives a higher model score than a negative example. Intuition seems to suggest that no reasonable methodology should ever result in a model with an AUC *significantly* below 0.5. The focus of this paper is not on the estimator properties of CV (bias/variance/significance), but rather on the properties of the 'holdout' predictions based on which the CV performance of a model is calculated. We show that CV creates predictions that have an 'inverse' ranking with AUC well below 0.25 using features that were initially entirely unpredictive and models that can only perform monotonic transformations. In the extreme, combining CV with bagging (repeated averaging of out-of-sample predictions) generates 'holdout' predictions with perfectly *opposite* rankings on random data. While this would raise immediate suspicion upon inspection, we would like to caution the data mining community against using CV for stacking or in currently popular ensemble methods. They can reverse the predictions by assigning negative weights and produce in the end a model that appears to have close to perfect predictability while in reality the data was random.

1. INTRODUCTION

Cross-validation (CV) is a widely studied and commonly used approach to evaluate modeling methodologies on finite datasets. Rather than setting a fixed amount of data aside for testing and thereby limiting the training set, CV splits the datasets into N equal size folds and uses N-1 such folds to build a model and one to estimate the performance on the remaining one. It finally averages the performances across all N folds as an estimate of the out-of sample performance. CV is particularly appealing when the data set is small or has only few positive examples. In the extreme case of leave-one-out every datapoint can be its own fold. The conventional wisdom is that CV may be computationally expensive for large N, but is generally reliable ([1]). 10 folds perform well and appear to provide a good tradeoff between evaluation variance and estimation variance. CV is known to have notable variance[12; 4] and a body of literature points at pitfalls when selecting methods based on maximum CV performance [13; 3; 9; 11; 5]. Only one (rather extreme) case is known where accuracy estimation using CV fails completely. If $p(y = 1) = 0.5$ and x is constant, it is easily seen that leave-one-out will produce an accuracy estimate of 0 [8] for almost all modeling techniques. The reason is a slight shift of the base-rate. If the held out datapoint was positive, the training set of all other data points would have a baserate slightly below 0.5 and assign a negative class label. A simple solution to avoid such shifts is *stratified* CV that requires that each fold has the exact same number of positive examples and thereby maintains the baserate in both training and holdout. But is this really the only case when things can go wrong? We identify a similar effect leading to inverse rankings and very low AUC's in much less restricted scenarios. Stacking and ensemble methods are very popular approaches that use CV to generate pseudo 'holdout' predictions which serve as input to the next modeling step[15; 6; 2]. Our findings suggest that this practice can produce very misleading and optimistic results when the number of positives is small.
So far we have not mentioned the actual classifier or modeling technique. As a matter of fact, it is close to irrelevant. As long as the algorithm is fairly sensitive to the underlying distribution and produces somewhat well-calibrated probability estimates, the results will be misleading. In particular, we experimented with Naive Bayes and logistic regression and both exhibit the same artifact. The cause is not the model but the shift in the underlying data. One final word on the evaluation metrics: The artifact is most obvious in AUC due to its sensitivity to minor relative shifts that can change the ranking significantly. For accuracy to be affected, the shifts need to occur close to the cutoff point of 0.5. As this is a special issue on unexpected results we will keep the discussion on a somewhat applied level without going into the theoretical results we have obtained [10].

2. HOLDOUT PREDICTIONS FROM CV

We will start by describing the initial setting of a real world project. Keep in mind that it serves only as demonstration and is meant to tell a tale. We have observed the exact same phenomena on other real domains as well as simulated data using different implementations. In the later parts of the paper we will abstract from the particular settings and explore on simulated data the theoretical reason and drivers of the artifacts. The goal of the motivating project was to identify *causal* drivers of failures in a production process. The number of confirmed failures was only 18 out of a total number of 3000 examples. The number of potential causes/features

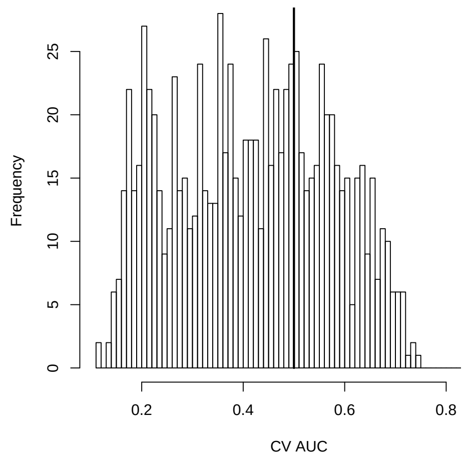

Figure 1: Distribution of AUC in 18-fold CV

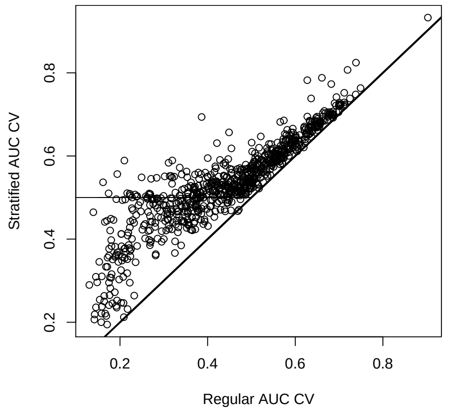

Figure 2: Comparing the AUC of stratified and regular CV

was approximately 900. Since some of them were categorical, we started our initial investigation by calculating for each of the features independently the holdout AUC using logistic regression and 18-fold CV. We choose 18 to allow for a stratification. A *causal* driver would be expected to show up with a very high AUC. Contrary to the usually performed average of the 18 AUC's we are calculating the AUC on the union of the predictions. This mirrors directly the stacking scenario. Originally CV was necessitated by the categorical features. However, we only use the numerical features in the remainder of the paper.

Figure 1 shows the distribution of the CV AUC's. We find only one feature with very high AUC > 0.9 — indicating some potential causal dependence. But alas, we have a lot more features with AUC<0.2. Does this mean that these features have a lot of signal indeed? Would flipping them around make a good model? While it is possible that logistic regression picks the for AUC 'wrong' sign of the parameter because it is optimizing likelihood, it still seems surprising to occur that often. Maybe this is an artifact from the shift in base-rate as reported previously? Figure 2 shows the effect of a stratification. Assigning one positive per fold

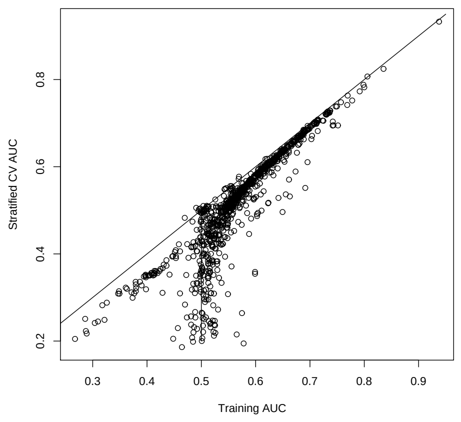

Figure 3: Comparing stratified CV AUC and training AUC

always improves the AUC. The effect is most notable for features with low signal. In particular, the gains are the largest for features with AUC below 0.5. However, even after stratification we are still observing a large number of AUC's below 0.5, some as low as 0.2. If true, this still means that the CV prediction has a lot of opposite signal.

One question we have not really discussed so far is what is the 'true' AUC for a given feature? There is a subtle distinction between the predictiveness of the feature and the predictiveness of the realization of the feature in a finite dataset. So far we have intuitively argued that any true AUC has to be larger or equal to 0.5. But even if that is the case, the realization in a small sample will obviously have some variance. While this discussion is interesting from a scientific standpoint - we care more to observe how the ranking ability of the holdout prediction has changed relative to the ranking ability of the original feature. The natural ranking could be measured as the better of two natural orders (decreasing or increasing). Instead of trying which is the 'better' direction (which would introduce a positive bias), we will use the in sample AUC of a logistic model to keep things comparable and account for the occasional 'wrong' sign of the logistic parameter. We do not think that this will introduce an overfitting problem since we still only apply a monotonic transformation which can worst case only flip the AUC'=1-AUC. Figure 3 shows this comparison of training AUC and 18-fold stratified CV AUC. These results are really interesting: For features with a significant signal (AUC>0.5) the training and CV AUC's are very similar. We also observe a number of features with AUC<<0.5 both on the training set and in cross-validation. These are the cases where the likelihood optimization of logistic regression picks the wrong sign. But, we see that the majority of features with very small CV AUC tend to have a training AUC around 0.5. So for these features, the holdout predictions are notably different from the original features. We would like to reinforce that the CV holdout predictions indeed provide a good inverse ranking on the dataset. This is not a measurement issue, but it is a property of the CV predictions. Unfortunately it is not true signal that could be exploited to predict failures on new data, instead the inverse ranking was *introduced* by the cross-validation process. To illustrate the persistence of this effect let us take a closer look at some of the 'bad' features with large differences. The largest difference between training (AUC = 0.57) and CV (AUC = 0.19) occurs for feature 304. One question we would like to address here is whether it is just accidentally a very low AUC due to some particular random assignment of the negatives into the folds (remember that the positives are stratified to one per fold). We re-estimate the CV AUC in 100 experiments where the negatives are assigned randomly. The average AUC is 0.196 and apparently rather typical for this feature and NOT caused by some random negative assignment. The conclusion is that the low AUC is indeed consistent and not related to random folds. To summarize our empirical observations:

- The CV predictions for some unpredictive features have strong negative predictive information.
- The 'spurious' information is introduced by the CV process (some explanation later).
- The effect remains even after stratification.
- The effect appears dominantly for features with no natural ranking ability (AUC 0.5).
- The effect is not a variance effect from the random assignment to folds.

3. SOME THEORETICAL THOUGHTS AND EXPLANATIONS

So what is causing the appearance of an inverse signal? The effect is fundamentally the same that happened in the accuracy failure of leave-one-out for balanced datasets. In some sense CV is a zero-sum setting. When you take from a fixed set of datapoints one (or more) examples out of the training and assign them to the test set, the two sets are no longer independent. An example that is in one cannot be in the other. This argument holds for the shift in base-rate when assigning more positives to the training and therefore less to the test. In fact, it is easy to generalize the accuracy case to failure of AUC for arbitrary baserates: Consider the artificial case of one-holdout with a constant feature $x = 1$ for all observations. Let N be the number of examples and S the number of positive examples. The predictions of most modeling approaches including a logistic model (or Naive Bayes) $M(x) = M(1)$ are:

$$M(x)|(y = 1) = \frac{S-1}{N} \tag{1}$$

$$M(x)|(y = 0) = \frac{S}{N-1} \tag{2}$$

It is easily seen that the predictions for all positive observations is smaller than the prediction for all negative observations leading to an AUC of 0. But this can again be fixed by stratification. However, similar shifts also happen to the distribution of the independent variable x.

Without going into the formal proof, lets consider stratified CV with one positive example per fold for a non-predictive binary $x \in 0, 1$ with $P(y|x) = P(y)$. Intuitively, assigning a positive example $(x, y) = (1, 1)$ to a fold implies that the union of the training folds has slightly reduced probability for $P(y = 1|x = 1)$ whereas the probability of $P(y|x = 0)$ remains largely (ignoring the random assignment of negatives for now) unaffected. As a result, the positive example along

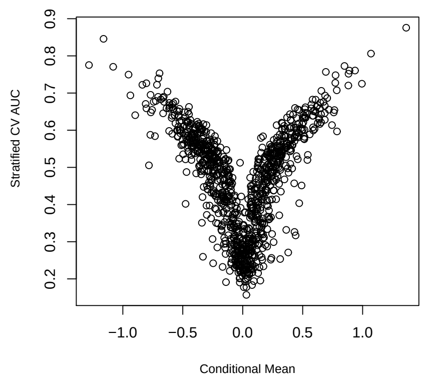

Figure 4: CV AUC as a function of the conditional mean

with all other negatives with $x = 1$ in the holdout fold gets a slightly smaller prediction than (negative) all examples with $x = 0$. The definition of AUC is the probability of a positive example having a higher prediction than a negative. For all examples with $x = 0$ this is 0.5 (they have the same prediction) and for the other half of examples with $x = 1$ this probability is 0. So we can expect an AUC close to 0.25. And indeed we have shown that 0.25 is a theoretical upper bound for the AUC in this scenario [10].

So what is happening practically? Whenever the positive example in the holdout fold has a value x that is large, the x's for the positives in the training are on average lower. If the conditional mean of $x|y = 1$ is lower than $x|y = 0$ logistic regression will estimate a negative parameter on x. As a result, the positive example in the holdout will receive a relatively *low* prediction. The same holds for the alternative case of a small x in the holdout. In essence, as the conditional mean $x|y = 1$ in the training switches sides with the conditional mean $x|y = 0$ (which is moving randomly as a result of the sampling), the sign of the parameter is alternating and due to the zero sum nature it tends to indicate the wrong direction. This also explains why the effect is dominant for truly uninformative features where the true parameter is 0.

The next experiment uses simulated data where x is sampled from a normal distribution N(0,1) and y is sampled independently (so x is not predictive) with a baserate of 0.01. We generate 1000 datasets with each having 100 observations. We use stratified CV where the number of folds equals the number of positives (on average 10 folds).

Figure 4 illustrates our intuition very well. It shows the CV AUC as a function of the conditional mean $x|y = 1$. The wide range of AUC (up to 0.9) is driven by the sample variance. We also observe many experiments with small AUC's. These low AUC's occur whenever the conditional mean is close to 0 and switches sides depending on the fold. At this point we would like to briefly comment on the properties of CV as an estimator of AUC and how it relates to our observations. While initially it appears odd that there are that many small AUC's, it makes perfect sense from an estimation perspective. If the sample from an independent variable in a given experiment can be such that the target and the feature happen to be strongly correlated (and the estimator will rightly measure a high AUC), there need to be instances where the estimator results in AUC << 0.5. Otherwise it cannot be unbiased and have an expectation of 0.5. So in some sense CV *has* to create the inverse signal for uncorrelated samples in order to be consistent. Table 1 shows the statistics of the distribution of the AUC estimates across the 1000 experiments. The first observation is that when used in the traditional sense of averaging the AUC across folds the mean is indeed very close to 0.5. However, this is not the case when calculating the AUC on the union of all folds as it is for instance implemented in WEKA[14]. The mean of 0.464 is far from the truth and the non-stratified case is even worse at 0.362. We are not aware whether the biased nature of the union has been explicitly discussed in previous literature. We are only aware of the discussion of inconsistencies between the two version by Forman and Scholz [7]. They however do not observe any bias, but only suggest that the union case would "punish models with badly calibrated probability estimates".

Stratification does not affect the traditional CV estimation since the shift in base-rate does not affect AUC on a single fold. The picture in Figure 4 is only slightly different when plotting the mean AUC in the traditional setting of averaging folds. The V shape remains identical; the only difference is that the points are slightly shifted up. In fact, the AUC on the union is always slightly lower than the mean AUC, which is consistent with some of our theoretical results [10]. But to reinforce our previous message: the holdout predictions based on which the AUC is calculated are the same and they are inversely affected.

4. PRACTICAL CONSIDERATIONS

For the remainder of this paper we are looking at some implications as well as different scenarios and how they affect the properties of the holdout predictions.

4.1 Number of Folds

How does the number of folds affect the AUC? In an attempt to allow for as many training examples as possible our previous experiments used as many folds as there were positive examples. Here we explore what happens if we reduce the number of folds. Intuitively, this might reduce the model performance in general, but could help against the inverse CV effect since we no longer single out one positive example. We expect to see the highest variance in the conditional mean for the largest number of folds. This is a direct result of the variance of the estimate of the mean of a distribution to be decreasing in the number of observations. The estimate of a mean from a single observation (the one positive in the holdout) by definition has the highest variance and therefore the highest chance of deviating far from the true conditional mean. To avoid the effect of changes of the baserate we only consider stratified 18,9,6,3 and 2-fold CV for our worst feature. Figure 5 shows indeed the largest AUC for 2 folds and the smallest for number of folds equal to the number of positive examples. The implication would suggest using as few fold as possible in the CV stacking scenario. However, for informative features there will be a tradeoff between the reduction of the inverse CV effect and the decrease of a model performance.

Method	Min	Median	Mean	Max
Union No Stratify	0.0941	0.3560	0.3621	0.8514
Union Stratify	0.1570	0.4894	0.4641	0.8757
Mean No Stratify	0.1289	0.5404	0.5018	0.8770
Mean Stratify	0.1156	0.5566	0.4980	0.8751

Table 1: Properties of the AUC estimation using CV. Union indicates that the predictions are pooled prior to calculation the AUC (stacking scenario) whereas Mean indicates that the AUC is calculated on each fold independently and finally averaged.

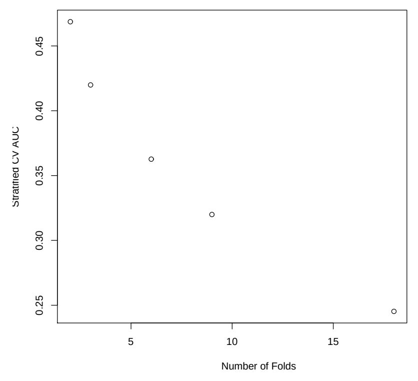

Figure 5: CV AUC as a function of the number of folds

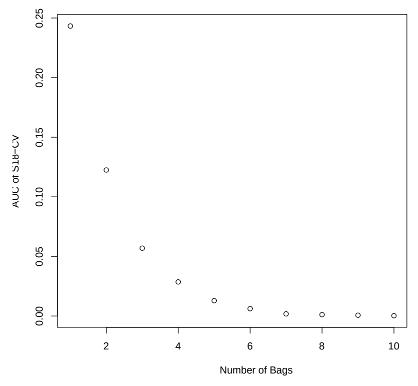

Figure 6: Bias as a function of the number of bagging iterations of stratified CV on a bad feature

4.2 Bagging

Bagging has the reputation to never really hurt. Here it does. Kohavi[8] even suggested that repeated cross-validation should reduce the variance of the CV estimate of model performance. So instead of running stratified CV only once, we ran it multiple times and averaged the holdout predictions prior to calculating the AUC.

Figure 6 shows that for a feature with moderately low AUC, it very quickly degrades and reaches completely inverse predictions (AUC = 0) within 8-10 bagging iterations. The smaller the initial AUC, the quicker was the decrease. In the light of our theoretical explanation, this degradation is very intuitive. In a given CV split, all examples in one fold with the same x value have identical predictions. Positive examples always have lower or equal predictions. If you average predictions from multiple fold splits the positives

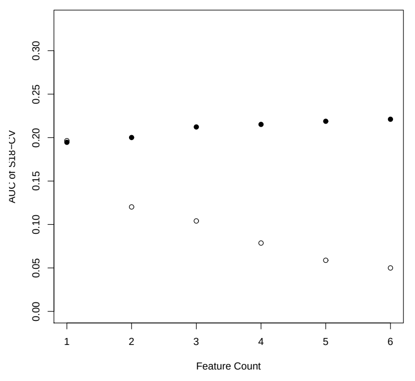

Figure 7: AUC under stratified CV as a function of the number of bad features in the model

always get the low predictions while negatives sometimes have a high predictions (when the positive in the fold had the opposite x value) and sometimes a low prediction. After a number of experiments the positive cases all have the lowest predictions.
On the good side, for a feature without a reasonable initial AUC repeated cross-validation does not systematically affect the AUC. Nevertheless these results are alarming. If this averaged holdout prediction was used in a stacking setting, the model would perfectly predict the target as long as it is evaluated on some split of the original dataset. Only completely new data (which is typically not the case for stacking) would show the truly random performance of this model and only new data can prevent the incorporation of such feature into an ensemble.

4.3 Multivariate Models

So far we have only considered single features. What happens in the normal case of a multivariate model? To investigate this we start with the infamous worst feature (most prone to low CV AUC) and add one-by-one six more features in reverse order of their CV AUC. Figure 7 shows how the CV performance deteriorates as more and more such features are added to the model. The solid circles above show the single AUC of the last added feature and light circles the performance of the model including all of the previous features. This result is alarming in the sense that in the more realistic setting with multiple features, the holdout predictions can have an AUC close to 0.

5. CONCLUSION

The main message of this paper is to beware when using CV in the context of stacking and when the number of positives is very low. If the nature of the problem requires CV, it is advisable to make the number of folds as small as possible. In addition, an intermediate test should reject all holdout predictions with AUC < 0.5 as inputs to the next modeling layer. On the upside, a low AUC in the context of CV is not nearly as surprising as it might appear on first sight (part of the estimation process) and typically only indicates that the feature/approach under study is NOT predictive. So rejecting low AUC is not likely to cause type two errors.

6. ACKNOWLEDGEMENTS

We would like to thank Saharon Rosset and Alexandru Niculescu-Mizil for very valuable discussions.

7. REFERENCES

[1] A. Blum, A. Kalai, and J. Langford. Beating the hold-out: Bounds for k-fold and progressive cross-validation. In *Computational Learing Theory*, pages 203–208, 1999.

[2] L. Breiman. Stacked regressions. *Machine Learning*, 24:49–64, 1996.

[3] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and regression trees*. Wadsworth International Group, Belmont, 1984.

[4] B. Efron. Estimating the error rate of a prediction rule: Improvement on cross-validation. *Journal of the American Statistical Association*, 78(382):316–331, 1983.

[5] B. Efron and R. Tibshirani. Cross-validation and the bootstrap: Estimating the error rate of a prediction rule. Technical report, Stanford University, 1995.

[6] A. Fast and D. Jensen. Why stacked models perform effective collective classification. In *Proceedings of the International Conference on Data Mining (ICDM)*, pages 785–790, 2008.

[7] G. Forman and M. Scholz. Apples-to-apples in cross-validation studies: Pitfalls in classifier performance measurement. *SIGKDD Explorations*, 12(1):49–57, 2010.

[8] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1137–1143, 1995.

[9] A. Y. Ng. Preventing overfitting of crossvalidation data. In *Proceedings of the 14th International Conference on Machine Learning (ICML)*, pages 245– 253, 1997.

[10] C. Perlich and G. Świrszcz. Cross-validation: Bias warning - proceed with care. Technical report, IBM Research, 2010.

[11] R. B. Rao, G. Fung, and R. Rosales. On the dangers of cross-validation. Overfitting and Overestimation of performance: An experimental evaluation. In *Proceedings of the SIAM International Conference on Data Mining (SDM)*, pages 588–596, 2008.

[12] M. Stone. Asymptotics for and against crossvalidation. *Biometrika*, 1(64):29–35, 1977.

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

[14] I. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, 1999.

[15] D. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.