

Practical Issues in Classification: Error rates, ROC Curves, and Cross-Validation

Nishant Mehta

September 14, 2010

Several practical issues arise in our quest for optimal classifiers. We will introduce several techniques that can be used to select an appropriate model for a given classification task. The section related to cross-validation also applies to general estimation tasks, such as regression, but for simplicity let us restrict our attention to classification.

Training and Test Error

Assume that for a sample $\mathcal{D} = (x^{(1)}, y^{(1)}, \dots, (x^{(n)}, y^{(n)}))$, each $(x^{(i)}, y^{(i)})$ is drawn iid from a distribution $p(x, y)$. We have some loss function $L(y, \hat{y})$ such as the 0-1 loss $L(y, \hat{y}) = 1$ if $y \neq \hat{y}$ and 0 otherwise.

Let us consider a classifier $f : \mathcal{X} \rightarrow \mathcal{Y}$.

For the fixed sample above, the *empirical risk* is

$$\mathcal{R}_{\mathcal{D}}(f) = \mathbb{E}_{(X, Y) \sim \mathcal{D}} [L(Y, f(X))] = \frac{1}{n} \sum_{i=1}^n L(y^{(i)}, f(x^{(i)})). \quad (1)$$

The empirical risk also commonly is called the *training error*, since it is the error suffered by the classifier f on a particular training set.

In contrast to the empirical risk, the true risk (commonly called the *generalization error*) is the expected loss $L(X, f(X))$ for (X, Y) drawn from the true distribution $p(x, y)$:

$$\mathcal{R}(f) = \mathbb{E}_{(X, Y) \sim p} [L(X, f(X))] \quad (2)$$

For learning tasks, we want f to perform well on some new test sample of examples that are drawn from $p(x, y)$, and hence, our end goal should be to minimize the true risk.

In practice, the training sample (and the test sample) will of course be finite, and the training error and test error may be wildly different. An easy way to see this is from two simple constructions: a ridiculously simple classifier and an incredibly complicated one.

- 1) Let f be the constant function $f(x) = 1$. Regardless of the training set, it classifies all examples as positive. In this case, the training error and the test error should be as close as possible; to see this, we can observe that the training error is simply the fraction of negative examples in the training sample, the test error is the fraction of negative examples in the test sample, and finally that the elements of the training and test set are drawn iid from the same distribution.
- 2) Suppose f was chosen from a training sample to be a “memorizing” function such that $f(x) = 1$ only if we observed x labeled as 1 in the training sample (i.e., $f(x) = 1$ if $(x, 1) \in \{(x^{(1)}, y^{(1)}), \dots, (x^{(n)}, y^{(n)})\}$; otherwise $f(x) = 0$). Let $\mathcal{X} \subset \mathbb{R}$, and suppose the marginal distribution $p(x)$ has bounded density everywhere (we observe any single point with probability zero). In this case, the training error is 0 but the test error is the fraction of negative examples in the test sample, since we will see the positively labeled points in the training set again with probability 0.

In 2), we selected a function f according to the training sample. At this point, we might think that f is just some function. It is a curious and somewhat deep question as to why the empirical risk, which is the expectation of the loss with respect to a training sample drawn iid from $p(x, y)$, is not an unbiased estimator of a function f . The answer is that the empirical risk (with respect to the training sample) of f is not actually an estimator of the true risk of f , but rather, it is an estimate of the true risk of the best function (with respect to a particular training sample) within a function class F . This is different than saying that the empirical risk (with respect to the training sample) of f as a fixed function is an estimate of the true risk of f . This point is somewhat subtle, but it will be revisited later in the course.

Suppose we wish to select among a set of classifiers that classifier which has the minimum true risk. The above two examples would indicate that the empirical risk of each classifier is not a good estimate of the true risk (in fact, it is an optimistic estimate); instead, after estimating the classifiers from the training sample, the correct approach is to compute the test error of each classifier on a separate test set to obtain an unbiased estimate of the true risk. We will explore this point further in the section on cross-validation.

ROC Curves and Error Tradeoffs

Seeking the minimum true risk (and in particular, seeking the minimum empirical risk) for 0-1 loss can be problematic when we have imbalanced classes. If $p(y = 1) = 0.01$ and $p(y = 0) = 0.99$, then the constant function $f(x) = 0$ has a true risk of only 0.01. In such settings, it often is the case that the label $y = 1$ corresponds to a rare but very important event, such as cancer. For these classification tasks, it can be desirable to have a parameter that lets us gain some correctness on positive examples by giving up some of our correctness on negative examples.

Suppose we can vary the ability of f to discriminate two classes. For example, if we have an estimator $g(x) : \mathcal{X} \rightarrow [0, 1]$ of class probabilities for the two classes given a data point, then f could be the decision rule

$$f(x) = 1 \text{ if } g(x) \geq \beta \text{ and } 0 \text{ otherwise,} \quad (3)$$

for a free parameter β . Each choice of β induces a different f , so that β induces a family of estimators. The usual choice of β would be 0.5 if the cost of misclassifying a truly positive example is equal to the cost of misclassifying a truly negative example and the marginal probabilities of the classes are equal. However, if the task is to classify toys as toxic or safe, then the cost of classifying a perfectly safe toy as toxic is far outweighed by cost of letting a toxic toy go to market. In the former case, an additional check can be done on the supposedly toxic toy, while in the latter case, (little) people may die.

Before going further, we need a few definitions:

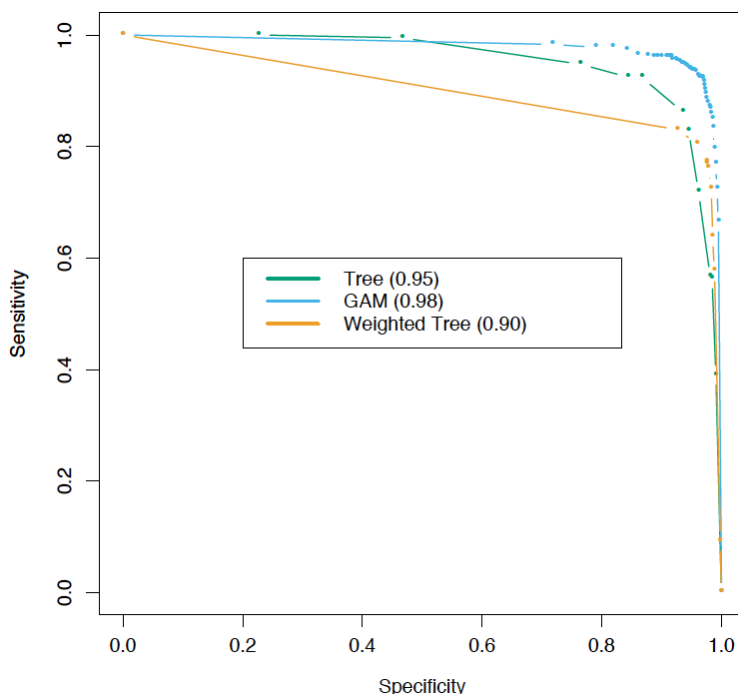
- True positives: Examples that f correctly labels as positive.
- True negatives: Examples that f correctly labels as negative.
- False positives: Examples that f labels positive but which are truly negative.
- False negatives: Examples that f labels negative but which are truly positive.

$$\text{true positive rate} = \frac{\# \text{ true positives}}{\text{total } \# \text{ positive}} \quad \text{true negative rate} = \frac{\# \text{ true negatives}}{\text{total } \# \text{ negative}} \quad (4)$$

The true positive rate also is called sensitivity, since it is a measure of how sensitive you are to the target (positive) class. The true negative rate also is called specificity, since it measures how specific you are about what you label positive.

A receiver operating characteristic curve (ROC curve) can be used to summarize tradeoffs between the true positive rate and the true negative rate. After creating an ROC curve, we can identify for a given curve if there is some value of β that provides an acceptable balance between the true positive rate and $(1 - \text{the true negative rate})$. By demanding more recall (more true positives among the total positives), we may encounter a higher false positive rate $(1 - \text{the true negative rate})$.

Consider the 3 ROC curves describing performance on a spam classification task below.



For any true negative rate, GAM¹ dominates or is competitive with the true negative rate of the other two models. When we demand that the true positive rate be exactly 1 (that we always label spam as spam), all of the models have horrible true negative rates (they label everything as spam!). If we however give up just a little on the true positives (by letting just a few spam emails into your inbox), we improve massively on the true negatives (we let most good emails into your inbox). The ROC curve shows this nicely.

The line from the top left corner to the bottom right corner ((TPR = 1, TNR = 0) to (TPR = 0, TNR = 1)) sometimes is called the line of no discrimination. If an ROC curve is below this line, it is doing no better than chance.

We also can use ROC curves for model selection. One metric is the area under the ROC curve (AUC). It is more common to see ROC curves plotted as the true positive rate versus (1 - the true negative rate).

A natural tradeoff to consider in fields like information retrieval is precision versus recall. Precision refers to the (true positives) / (true positives + false positives). It is a measure of how precise you are about what you label positive. Recall is equivalent to sensitivity. It is a measure of the rate at which you sense/recall/(correctly label) positive examples compared to the overall number of positive examples.

Example:

User → Search query

Search results ← Google

Precision: percent of returned search results that are relevant to the query

Recall: percent of overall relevant search results that the query returned

¹GAM is a generalized additive model. You can think of this and the other two models just as *some* models.

Cross-validation

When the true model is known to us, we can estimate/optimize the model parameters on a training set. The true risk can then be estimated by computing the empirical risk on a test set drawn identically from the same distribution as the training set. However, suppose that the complexity of the model is not known to us. Rather than using a single model f , we instead can consider a collection of classifiers (f_1, \dots, f_M) ; perhaps each classifier f_i corresponds to setting a value of a tuning parameter to α_i .

We need a standard way of selecting the best model (classifier) from this set. The models with higher complexity are likely to have lower training error, since they can better fit arbitrary samples; hence, it is not sufficient to use the training error as a criterion for model selection, as this typically leads to selection of overly complicated models.

Ideally, we would select the model that has the lowest true risk. If we are in possession of a copious amount of data (a large iid sample), one strategy is to

1. Divide the sample into 3 (not necessarily equal) parts: a training set, a validation set, and a test set.
2. Fit each f_i to the training set to yield an estimator \hat{f}_i .
3. Obtain an estimate of the true risk by computing the error of each estimator on the validation set.
4. Select the estimator \hat{f}^* that has the lowest error on the validation set. Note that it is not sufficient to report the empirical risk of \hat{f}_{i^*} as an unbiased estimate of the true risk (can you explain why?), so we are not done yet.
5. Fit f_{i^*} on the union of the training set and the validation set, obtaining \hat{f}^* .
6. Estimate the true risk of \hat{f}^* by computing its error on the test set.

The problem with this strategy is that if we divide our data into three parts, we may not have enough data to properly fit the models and obtain good estimates of the test error on the validation and test sets.

k -fold cross-validation is a powerful technique which can more efficiently make use of the data to help us accomplish our task. The sample is split into a training set $A = ((x_1, y_1), \dots, (x_n, y_n))$ and a test set B . The training set then is partitioned into k folds A_1, \dots, A_k of equal or close to equal size. For each j , we obtain an estimate \hat{f}^{-j} of the optimal classifier by minimizing the empirical risk over $A \setminus A_j$ (the full training set A minus the j^{th} fold A_j). We can estimate the true risk of \hat{f}^{-j} by measuring its empirical risk on A_j . The mean of the empirical risks

$$\hat{\mathcal{R}}_{\text{cv}}(f) = \frac{1}{k} \sum_{j=1}^k \frac{1}{|A_j|} \sum_{(x_i, y_i) \in A_j} L(\hat{f}^{-j}(x_i), y_i) \quad (5)$$

is the cross-validation estimate of the true risk of f .

This estimate can be used as a criterion for selecting among multiple classifiers f_1, f_2, \dots, f_M , by choosing

$$f^* = \arg \min f \in \{f_1, \dots, f_m\} \hat{\mathcal{R}}_{\text{cv}}(f). \quad (6)$$

The selected classifier f^* can then be trained on the entire training set, and we can estimate the true error via the error on the test set. Note that the test set should not be touched at all until this very moment!

Two of the most commonly used varieties of k -fold cross validation are leave-one-out cross-validation (LOOCV), where $k = n$, and 10-fold cross-validation. Both seem to better estimate the true error of a classifier (i.e., with an expectation over the random variable that is the training set), rather than the true error of a classifier conditional on the given training set. In practice, 10-fold cross-validation tends to provide a better estimate of the true risk than LOOCV. This may seem counter-intuitive as LOOCV uses the largest training sets during cross-validation; indeed, from using the largest training sets, LOOCV is an approximately unbiased estimator of the true prediction error; however, since the training sets for each fold are all highly similar, the LOOCV estimator of the true risk may also have high variance.

In contrast, in 10-fold cross-validation the training set used in each fold differs by roughly 10% from the training sets in the other folds. As a result, the cross validation score for 10-fold cross-validation tends to have lower variance; although it may have a pessimistic bias since it deals with smaller training sets in each fold, it appears to provide a good balance between bias and variance.