Thoughts on prediction and cross-validation.

Ronald Christensen

Department of Mathematics and Statistics

University of New Mexico

May 21, 2015

Abstract

KEY WORDS:

1. Cross-validation of R^2.

Typical application: Five fold cross-validation of normal models by using exact copies of X.

Suppose we have a random vector (y, x') where y is a scalar random variable and want to use x to predict y. We do this by defining some predictor function $f(x)$. We also have a prediction loss function $L[y, f(x)]$ that allows us to evaluate how well a predictor does. Want f that minimizes

$$\mathrm{E}_{y,x}\{L[y, f(x)]\}$$

which is called the *expected prediction loss* or the *expected prediction error.* Also, for whatever predictor we end up using, we want to be able to estimate the expected prediction error.

Two examples:

The loss function determines the best predictor. These problems are equivalent to Bayesian decision problems if we just think of y as θ, the marginal distribution of y as the prior of θ, and the prediction loss as the decision loss. In this context,

$$\mathrm{E}_{y,x}\{L[y, f(x)]\}$$

is the Bayes risk of the decision problem and the optimal predictor will be the Bayes decision rule.

The most common loss function for prediction is squared error, see PA Section 6.3,

$$L[y, f(x)] = [y - f(x)]^2$$

from which it follows that the optimal estimator is the "posterior" mean

$$m(x) \equiv E(y|x).$$

For the special case when $y \sim \mathrm{Bern}(p)$, write $p(x) \equiv E(y|x)$. The use of squared error loss leads to estimates of the expected prediction error called Brier scores. Another option called Hamming loss is

$$L[y, f(x)] = 1 - I_{\{0\}}[y - f(x)].$$

In other words, for Hamming loss if you predict y correctly there is no loss and if you predict it incorrectly the loss is 1. The expected prediction error is just the probability of mispredicting y, i.e.,

$$\mathrm{E}_{y,x}\{L[y, f(x)]\} = \mathrm{Pr}_{y,x}\{y \neq f(x)]\}$$

Note that with Hamming loss, it makes no sense to predict a value other than 0 or 1, so these will be referred to as valid predictions. Hamming loss is equivalent to the Bayes test procedure with $y = 1$ the alternative hypothesis and $y = 0$ the null. The optimal prediction is equivalent to rejecting when the posterior probability of the alternative is greater that 0.5, i.e., the optimal rule δ has

$$\delta(x) = \begin{cases} 1 & \text{if } p(x) > 0.5 \\ 0 & \text{if } p(x) < 0.5 \end{cases}.$$

We don't care which valid prediction we make (action we take) when $p(x) = 0.5$. This rule clearly minimizes the loss for each x but using Bayes Theorem one can also show that it has the form of the N-P Lemma so is a most powerful test. □

Note that

$$\mathrm{E}_{y,x}\{L[y, \delta(x)]\} = \mathrm{Pr}_{y,x}\{y \neq \delta(x)]\} = \int_{\{x|p(x)\geq 0.5\}} [1 - p(x)]f(x)dx + \int_{\{x|p(x)<0.5\}} p(x)f(x)dx.$$

These rules depend on knowing the joint distribution of (y, x'), which is generally unknown in prediction problems. We want to use data to estimate both $E(y|x)$ and $\mathrm{E}_{y,x}\{L[y, m(x)]\}$. Suppose $(y, x'), (y_1, x_1'), \ldots, (y_n, x_n')$ are iid. Let Y be the vector of y_is and let X is the matrix with x_i' as its ith row.

Estimate $E(y|x)$**.**

A nonparametric approach to estimating $E(y|x)$ is to identify x_i values that are close to x and estimate $E(y|x)$ by taking a weighted mean of the y_is that correspond to close x_is.

Obviously, the weights on the y_i might well depend on how far the x_is are from x. This is called a nearest-neighbor approach.

Quite generally, one can assume that $E(y|x)$ is a member of a parametric family, say $m(x;\theta)$ and use a maximum likelihood estimate of θ, say $\hat{\theta}$. In this set-up, the x_i are treated as fixed and the distributions of y_i given x_i are assumed independent and to be in a parametric family of distributions (largely) determined by its mean. This is already in the form of nonlinear regression but standard generalized linear models also fit this paradigm. Nonparametric regression techniques based on basis functions such as polynomials, wavelets, or sines and cosines also fit into the generalized linear model paradigm.

In general, we end up with an estimate

$$\hat{m}(x) \equiv m(x;\hat{\theta}).$$

.

Estimate $\mathrm{E}_{y,x}\{L[y, m(x)]\}$**.**

If we know m, an unbiased estimate is

$$\frac{1}{n}\sum_{i=1}^{n} L[y_i, m(x_i)]. \tag{1}$$

Generally, we have to estimate m, so we might use

$$\frac{1}{n}\sum_{i=1}^{n} L[y_i, \hat{m}(x_i)]. \tag{2}$$

Since $\hat{m}$ is a complex function of the data, the expected value of this function is hard to find. Conventional wisdom is that (2) underestimates the true expected prediction error, e.g.,

$$\mathrm{E}\left\{\frac{1}{n}\sum_{i=1}^{n} L[y_i, \hat{m}(x_i)]\right\} \leq \mathrm{E}\left\{\frac{1}{n}\sum_{i=1}^{n} L[y_i, m(x_i)]\right\}.$$

I wonder if Eaton's methods might be able to show this? (We will show not only that this is true for linear models but that cross-validation can be even more biased upwards.)

To "fix" this problem, people try Cross-Validation. Life is much easier if we have one set of (training) data from which to estimate $\mathrm{E}(y|x)$ and a different set of (test) data from which to estimate $\mathrm{E}_{y,x}\{L[y, m(x)]\}$. In such a case, $\hat{m}$ based on the training data is a fixed predictor with regard to the test data so equation (1) gives an unbiased estimate of expected prediction error for $\hat{m}$ given the training data. One might call this procedure, *validation*.

Cross-validation is based on using the validation idea repeatedly with the same data. For example, k-fold cross-validation randomly divides the data into k subsets of roughly equal size. First identify one subset as the test data and combine the other $k-1$ subsets into the training data. Estimate the best predictor from the training data and then use that estimate with the test data to estimate the expected prediction error. So far, this is just validation and the estimate of the expected prediction error should be conditionally unbiased.

However, in k-fold cross-validation there are k possible choices for the test data, so one goes through all k validation processes and averages the k estimates of the expected prediction error to give an overall estimate of the expected prediction error. With n data points, the largest choice for k is n, which is known as *leave one out cross-validation*.

Let's look at how all of this works in the most tractable case, linear models with squared prediction error loss. In linear models and more generally in nonparametric regression the model is typically taken as

$$y_i = m(x_i) + \varepsilon_i, \quad \mathrm{E}(y_i) = 0, \quad \mathrm{Var}(\varepsilon) = \sigma^2$$

with independent ε_is, or alternatively

$$y_i|X \quad \text{indep.} \quad \mathrm{E}(y_i|X) = m(x_i), \quad \mathrm{Var}(y_i|X) = \sigma^2.$$

This model will not work for $y \sim \mathrm{Bern}(p)$ because the constant variance condition cannot hold except in degenerate cases. Under squared error loss

$$\mathrm{E}_{y,x}\{L[y, m(x)]\} = \mathrm{E}_x \mathrm{E}_{y|x}\{L[y, m(x)]\} = \mathrm{E}_x \sigma^2 = \sigma^2.$$

In a linear model

$$m(x) = x'\beta$$

.

It is not hard to see that (1) leads to

$$\frac{1}{n}\sum_{i=1}^{n} L[y_i, m(x_i)] = \frac{1}{n}\sum_{i=1}^{n}[y_i - x_i'\beta]^2$$

which is an unbiased estimate of σ^2. However, with least squares estimation and $\hat{m}(x) = x'\hat{\beta}$,

$$\mathrm{E}_{Y|X}\left\{\frac{1}{n}\sum_{i=1}^{n}[y_i - x_i'\hat{\beta}]^2\right\} = \frac{n - r(X)}{n}\sigma^2,$$

which underestimates σ^2. Of course, what we really do in linear models is use the mean squared error, i.e.,

$$\mathrm{E}_{Y|X}\left\{\frac{1}{n - r(X)}\sum_{i=1}^{n}[y_i - x_i'\hat{\beta}]^2\right\} = \sigma^2.$$

Finally, for leave one out cross-validation, the estimate uses the well known Press statistic, see PA Chapter 13. In the following, let $p \equiv r(X)$. With M the perpendicular projection operator onto the model matrix space, I believe

$$\begin{aligned}
\mathrm{E}(Press/n) &= \frac{1}{n}\mathrm{E}\left[Y'(I-M)D^2\left(\frac{1}{(1-m_{ii})}\right)(I-M)Y\right] \\
&= \frac{1}{n}\mathrm{tr}\left[D^2\left(\frac{1}{(1-m_{ii})}\right)(I-M)\sigma^2 I(I-M)\right] \\
&= \frac{\sigma^2}{n}\mathrm{tr}\left[D\left(\frac{1}{(1-m_{ii})}\right)(I-M)D\left(\frac{1}{(1-m_{ii})}\right)\right] \\
&= \frac{\sigma^2}{n}\sum_{i=1}^{n}\frac{1}{(1-m_{ii})}.
\end{aligned}$$

For a one sample problem (intercept only model),

$$\mathrm{E}(Press/n) = \frac{n}{n-1}\sigma^2$$

which is biased up. In fact, this is a lower bound for the expected value among models that include an intercept. Moreover, for $x = 0, 0.5, 1, 10, 19, 19.5, 20$ and fitting a cubic polynomial, I believe $\mathrm{E}(Press/n) > 5\sigma^2$.

In fact, since

$$\sum_{i=1}^{n} \frac{(1-m_{ii})}{n} = \frac{n-p}{n}$$

Jensen's Inequality gives

$$\frac{1}{n}\sum_{i=1}^{n} \frac{1}{(1-m_{ii})} \geq \frac{n}{n-p},$$

so it looks like Leave One Out CV is multiplicatively more biased upward than the naive estimator is biased downward.

I remember from talking to Rick Picard about his thesis years ago that he claimed Press really sucked. I wonder if this is why he said that.

Other Loss Functions

Using the formal equivalence between prediction and Bayesian decision theory, we can draw conclusions about other loss functions. For example, if for a positive weighting function $w(\cdot)$,

$$L[y, f(x)] = w(y)[y - f(x)]^2,$$

the BP is

$$\frac{\mathrm{E}[yw(y)|x]}{\mathrm{E}[w(y)|x]}.$$

If

$$L[y, f(x)] = |y - f(x)|,$$

the BP is

$$\mathrm{med}(y|x).$$

Moreover, if we use the absolute loss function with y Bernoulli, we get the same result as using Hamming loss, i.e., the BP is

$$\delta(x) = \begin{cases} 1 & \text{if } p(x) > .5 \\ 0 & \text{if } p(x) < .5. \end{cases}$$

Wikipedia:Cross-validation

Statistical properties

Suppose we choose a measure of fit F, and use cross-validation to produce an estimate F* of the expected fit EF of a model to an independent data set drawn from the same population as the training data. If we imagine sampling multiple independent training sets following the same distribution, the resulting values for F* will vary. The statistical properties of F* result from this variation.

The cross-validation estimator F* is very nearly unbiased for EF. **** The reason that it is slightly biased is that the training set in cross-validation is slightly smaller than the actual data set (e.g. for LOOCV the training set size is n - 1 when there are n observed cases). In nearly all situations, the effect of this bias will be conservative in that the estimated fit will be slightly biased in the direction suggesting a poorer fit. In practice, this bias is rarely a concern.

The variance of F* can be large.[9][10] For this reason, if two statistical procedures are compared based on the results of cross-validation, it is important to note that the procedure with the better estimated performance may not actually be the better of the two procedures (i.e. it may not have the better value of EF). Some progress has been made on constructing confidence intervals around cross-validation estimates,[9] but this is considered a difficult problem.