Pre-validation and inference in microarrays

Robert Tibshirani [*] and Bradley Efron [†]

June 26, 2002

Abstract

In microarray studies, an important problem is to compare a predictor of disease outcome derived from gene expression levels to standard clinical predictors. Comparing them on the same dataset that was used to derive the microarray predictor can lead to results strongly biased in favor of the microarray predictor. We propose a new technique called "pre-validation" for making a fairer comparison between the two sets of predictors. We study the method analytically and explore its application in a recent study on breast cancer.

1 Introduction

A DNA microarray dataset typically consists of expression measurements on a large number of genes (say 10,000) over a small set of cases (say 100). In addition, a true class label is often available for each case, and the objective is to find a function of the gene expression values that accurately predicts the class membership. A number of techniques have been proposed for this [see e.g. Dudoit et al. (2001)]. Having formed a prediction function from the microarray values, the following problem often arises in a medical context: how do we compare the microarray predictor to an existing clinical predictor

[*] Department of Health Research & Policy, and Department of Statistics, Stanford University, Stanford CA 94305; tibs@stat.stanford.edu

[†] Department of Statistics, and Department of Health Research & Policy, Stanford University, Stanford CA 94305; brad@stat.stanford.edu

of the class membership. Does the new microarray predictor add anything to the clinical predictor?

An example of this problem arose in the paper of van't Veer et al. (2002). Their microarray data has 4918 genes measured over 78 cases, taken from a study of breast cancer. There are 44 cases in the good prognosis group and 34 in the poor prognosis group. The microarray predictor was constructed as follows:

1. 70 genes were selected, having largest absolute correlation with the 78 class labels

2. Using these 70 genes, a nearest centroid classifier (described in detail in Section 6) was constructed.

3. Applying the classifier to the 78 microarrays gave a dichotomous predictor z_j for each case j.

It was of interest to compare this predictor to a number of clinical predictors including tumor grade, estrogen receptor (ER) status, progesteron receptor (PR) status, tumor size, patient age, and angioinvasion. The top part of Table 1, labelled "re-use", shows the result when a multivariate logistic regression was fit to the microarray predictor and clinical predictors. The microarray predictor looks very strong, but this is not surprising as it was derived using the same set of cases that are used in the logistic regression. In the bottom half of the table, we have derived a "pre-validated" version of the microarray predictor, and used it in the logistic regression. Now the microarray predictor is much less significant, and the clinical predictors have strengthened.

The idea of "pre-validation" was used in the supplementary material for van't Veer et al. (2002), and is the topic of this paper. It is also similar to the method of "stacking" due to Wolpert (1992), in the area of machine learning.

2 Pre-validation

In order to avoid the overfitting problem apparent in the top half of Table 1, we might try to use some sort of cross-validation:

1. Divide the cases up into say K approximately equal-sized parts

Table 1: *Results of model fitting to breast cancer data. Top panel: re-using the microarray scores z_j in the logistic model; bottom panel: using pre-validated scores $\tilde{z}_j$. The last column is the change in misclassification rate, when the given predictor is deleted from the full model. The full-model misclassification rates are 0.14 and 0.21 for the re-use and pre-validated models respectively.*

Model	Coef	Stand. Err.	Z score	p-value	Odds ratio	ΔMR
Re-use						
microarray	4.096	1.092	3.753	0.000	60.105	0.12
angio	1.208	0.816	1.482	0.069	3.348	0.01
er	-0.554	1.044	-0.530	0.298	0.575	0.01
grade	-0.697	1.003	-0.695	0.243	0.498	0.01
pr	1.214	1.057	1.149	0.125	3.367	-0.01
age	-1.593	0.911	-1.748	0.040	0.203	0.01
size	1.483	0.732	2.026	0.021	4.406	0.00
Pre-validated						
microarray	1.549	0.675	2.296	0.011	4.706	0.05
angio	1.589	0.682	2.329	0.010	4.898	0.01
er	-0.617	0.894	-0.690	0.245	0.540	0.01
grade	0.719	0.720	0.999	0.159	2.053	0.00
pr	0.537	0.863	0.622	0.267	1.710	0.01
age	-1.471	0.701	-2.099	0.018	0.230	0.03
size	0.998	0.594	1.681	0.046	2.714	0.04

2. Set aside one of parts. Using the other $K-1$ parts, select the 70 genes having the largest absolute correlation with the class labels, and form a nearest centroid classifier.

3. Fit a logistic model to the kth part, using the microarray class predictor and clinical predictors

4. Do steps 2 and 3 for each of the $k = 1, 2, \ldots K$ parts, and average the results from the K resulting logistic models.

The main problem with this idea is step 3, where there will typically be too few cases to fit the model. In the above example,with $K = 10$, the 10th part would consist of only 7 or 8 cases. Using a smaller value of K (say 5) would yield a larger number of cases, but then might make the training sets too small in step 2.

Pre-validation is a variation on cross-validation that avoids these problems. It derives a "fairer" version of the microarray predictor, and then this predictor is fit along side the clinical predictors in the usual way. Here is how pre-validation was used in the bottom half of Table 1:

1. Divide the cases up into $K = 13$ equal-sized parts of 6 cases each.

2. Set aside one of parts. Using only the data from the other 12 parts, select the genes having absolute correlation at least .3 with the class labels, and form a nearest centroid classification rule.

3. Use the rule to the predict the class labels for the 13th part

4. Do steps 2 and 3 for each of the 13 parts, yielding a "pre-validated" microarray predictor $\tilde{z}_j$ for each of the 78 cases.

5. Fit a logistic regression model to the pre-validated microarray predictor and the 6 clinical predictors. Figure 1 illustrates the logic of this computation.

Notice that the cross-validation in steps 1–3 deals only with the microarray predictor: it creates a "fairer" version of this predictor, in which the predictor for case j has not seen the true class label for case j. This pre-validated predictor is then compared to the clinical predictor in the standard way at step 5.

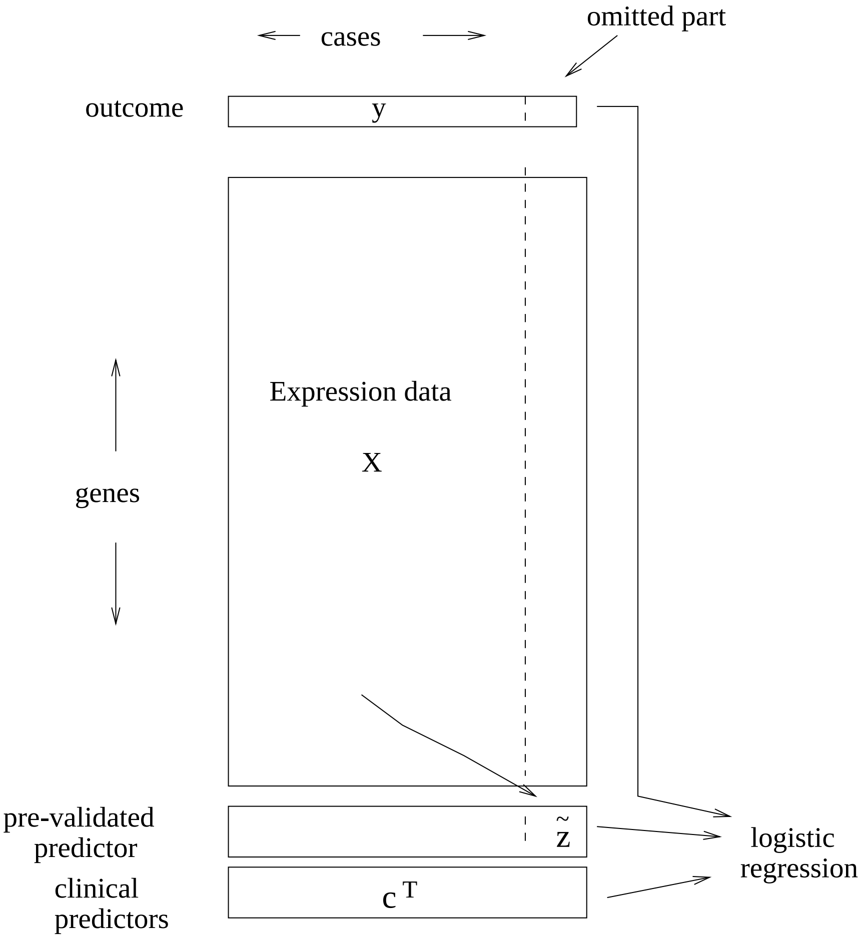

Figure 1: *Schematic of pre-validation process. The cases are divided up into (say) 10 equal-sized groups. The cases in one group are left out, and a microarray predictor is derived from the expression data of the remaining cases. Evaluating this predictor on the left-out cases yields the pre-validated predictor $\tilde{z}$ for those cases. This is done for each of the 10 groups in turn, producing the pre-validated predictor $\tilde{z}$ for all cases. Finally, $\tilde{z}$ can be included along with clinical predictors in a logistic regression model, in order to assess its relative strength in predicting the outcome.*

3 Pre-validation in detail

We have microarray data X, a $p \times n$ matrix of measurements on p genes over n cases. [1] The outcome is an n-vector $\mathbf{y}$, and suppose we also have set of k clinical predictors, represented by a $n \times k$ matrix $\mathbf{c}$, for predicting $\mathbf{y}$. Let x_j denote the jth column of X.

An expression predictor $\mathbf{z} = (z_1, z_2, \dots z_n)$ is adaptively chosen from the training data

$$z_j = f_{X,\mathbf{y}}(x_j). \tag{1}$$

The notation indicates that z_j is a function of the data X and $\mathbf{y}$, and is evaluated at x_j. In our motivating example earlier, the function $f(\cdot)$ consisted of finding the 70 top correlated genes, and using them to form a nearest centroid classifier. Our challenge is to compare $\mathbf{c}$ to $\mathbf{z}$, in terms of their strength in predicting $\mathbf{y}$. The *re-use* method uses the predictor $\mathbf{z}$ as is, and then fits a model to examine the relative contributions of $\mathbf{z}$ and $\mathbf{c}$ in predicting $\mathbf{y}$. In our motivating example, we used a logistic model. Clearly this comparison is biased in favor of $\mathbf{z}$, since the outcomes $\mathbf{y}$ were already used in the construction of $\mathbf{z}$

It is helpful to consider what we would do given an independent test set $(X^0, \mathbf{y}^0)$ with corresponding clinical predictors $\mathbf{c}^0$. We could use the test set to derive the predictor $\mathbf{z}^0 = (z_1^0, z_2^0, \dots z_n^0)$ where $z_j^0 = f_{X,\mathbf{y}}(x_j^0)$ and then use this to fit a model to predict $\mathbf{y}^0$ from $\mathbf{z}^0$ and $\mathbf{c}^0$. This would allow us to directly examine the relative contributions of $\mathbf{z}^0$ and $\mathbf{c}^0$ in predicting $\mathbf{y}^0$.

K-fold *pre-validation* tries to mimic this, without use of a test set. We divide the cases into K roughly equal-sized groups, and let $g(k)$ be the cases composing each part k. For $k = 1, 2, \dots K$, we form the pre-validated predictor

$$\tilde{z}_{g(k)} = f_{X^{-g(k)},\mathbf{y}^{-g(k)}}(x_{g(k)});\ \text{for } k = 1, 2, \dots K \tag{2}$$

the notation indicating that cases $g(k)$ have been removed from X and $\mathbf{y}$. Finally, we fit the model to predict $\mathbf{y}$ from $\tilde{\mathbf{z}}$ and $\mathbf{c}$, and compare their contributions in this prediction.

[1] It is conventional in the microarray area to arrange the data matrix X with genes (predictors) in the rows and cases in the columns. Although this is the transpose of the usual statistical convention, we adopt it here. Accordingly, we index the cases by the subscript j.

4 Does pre-validation work?

The goal of pre-validation is to construct a "fairer" version of the microarray predictor that acts as if it hasn't "seen" the response y. One way to quantify this goal is as follows. When the pre-validated predictor $\tilde{z}$ is included with clinical predictors in a linear or linear logistic model for predicting y, it should spend one degree of freedom. For the usual (non pre-validated) predictor z, we expect more than one degree of freedom to be spent. In this section we will make these notions precise and see if pre-validation works in this sense.

As before we define a microarray predictor as

$$z(x) = f_{X,\mathbf{y}}(x), \tag{3}$$

let $\mathbf{z}$ be the vector of values $(z(x_1), z(x_2), \ldots z(x_n))$ and let $\mathbf{c}$ be an n by k matrix of k clinical predictors. Using these we fit a linear or linear logistic model to predict y, with predicted values

$$\hat{\mu}(x, c;\ \mathbf{z}, \mathbf{c}, \mathbf{y}). \tag{4}$$

The first two arguments indicate the predicted values are evaluated at x (a p-vector of expression values) and c (a k-vector of clinical predictors).

Let $\hat{\mu}_j = \hat{\mu}(x_j, c_j;\ \mathbf{z}, \mathbf{c}, \mathbf{y})$, the predicted values for the training data, and let σ^2 be the variance of each y_j. Following Efron et al. (2002) (see also Stein (1981),Ye (1998)), we define the degrees of freedom in the predicted values $\hat{\mu}_j$ to be

$$\mathrm{df}(\hat{\mu}) = \mathrm{E}(\sum_{j=1}^{n} \frac{\partial \hat{\mu}_j}{\partial y_j})/\sigma^2 = \sum_{j=1}^{n} \mathrm{cov}(\hat{\mu}_j, y_j)/\sigma^2 \tag{5}$$

The leftmost relation defines degrees of freedom by the total change in the predicted values as each y_j is perturbed. On the right, it is defined as the total "self-influence" of each observation on its predicted value. These two notions are equivalent, as shown in Efron (1986). He proves that (5) holds exactly for the Gaussian model, and approximately when $\hat{\mu}$ is an expectation parameter in an exponential family model.

In the special case that f is a linear function of y and the model giving $\hat{\mu}(x, c;\ \mathbf{z}, \mathbf{c}, \mathbf{y})$ is linear least squares, we can derive an explicit expression for $\mathrm{df}(\hat{\mu})$.

Let $\mathbf{z} = A\mathbf{y}$ and let M be the $n \times (k+1)$ matrix $\{\mathbf{z}, \mathbf{c}\}$. Let P project onto the column space of M and P_c project onto the column space of $\mathbf{c}$ alone. Define $A^{\perp} = (I - P)A$, $\mathbf{z}_c = (I - P_c)\mathbf{z}$ and $\hat{\mu} = P\mathbf{y}$. Then we have the following results:

$$\begin{aligned}
\sum_1^n \frac{\partial \hat{\mu}_j}{\partial y_j} &= (k+1) + \frac{\mathbf{y}^T(A^{\perp} + \mathrm{tr}(A^{\perp}) \cdot I)\mathbf{z}_c}{||\mathbf{z}_c||^2} \\
\sum_1^n \frac{\partial \hat{\mu}_j}{\partial y_j}|_{y=\hat{\mu}} &= (k+1) + \frac{\mathbf{y}^T(\mathrm{tr}(A^{\perp}) \cdot I)\mathbf{z}_c}{||\mathbf{z}_c||^2} \qquad (6)
\end{aligned}$$

The proof is given in the Appendix.

The term $(k+1)$ is the degrees of freedom if $\mathbf{z}$ were the usual kind of predictor, constructed independently of the training data. The second term is the additional degrees of freedom due to the possible dependence of $\mathbf{z}$ on $\mathbf{y}$.

If A is a least squares projection matrix of rank p and there are no clinical predictors ($k = 0$), then one can show that the second term in expression (6) is $p - 1$, so that the total degrees of freedom is $(0+1) + (p-1) = p$.

It turns out that leave-one-out pre-validation can also be expressed as a linear operator. Let H be the projection matrix onto the row space of X (recall that X is $p \times n$, with genes in the rows). Let D be a diagonal matrix consisting of the diagonal of H, and let I be the $n \times n$ identity matrix. Then the pre-validated predictors have the form

$$\mathbf{z} = A\mathbf{y}; \text{ with } A = (I - D)^{-1}(H - D) \qquad (7)$$

[using the Sherman-Morrison Woodbury identity; see e.g. Hastie & Tibshirani (1990), chapter 3]. The matrix A has some interesting properties, for example $\mathrm{tr}(A) = 0$.

Our hope here is that with A defined as in (7), the additional degrees of freedom [second term in (6)] will be zero. This is difficult to study analytically, so we turn to a numerical study in the next section.

5 Numerical study of degrees of freedom

In this section we carry out some simulations to study the degrees of freedom in pre-validated predictors.

In our first study, we generated standard independent normal expression measurements on $n = 50$ or 200 cases and p genes, p ranging from 2 to 20. The outcome y was generated $y = x^T\beta + \epsilon$ where $\epsilon \sim N(0, .04)$. The coefficient vector β was set to zero in the null case, and $\beta \sim N(0, 1)$ in the non-null case. Finally, a samples of independent clinical predictors $c \sim N(0, 1)$ was generated.

The linear pre-validation fit (7) was computed from the expression data, and included along with c in a linear least-squares model to predict y. Note that for simplicity, the outcome y was centered and no intercept was included in the linear least squares model. The mean and standard error of the total degrees of freedom [formula (6)] over 50 simulations is shown in Table 2.

While the degrees of freedom tends to be less than $p+1$ (the value without pre-validation), we see it exceeds the ideal value of 2 in the null case, and is less than 2 in the non-null case.

The null case is most bothersome, for in that case treating the microarray predictor as a one degree of freedom predictor will cause us to overestimate its effect. In the null setting with $p = 2$, it is remarkable that the degrees of freedom is actually greater than $p + 1$, which is the value for the non pre-validated predictor.

We think of this increase in degrees of freedom as "leakage". While pre-validation makes each value $\tilde{z}_j$ independent of its outcome value y_j, the outcome y_j does influence other pre-validated values z_k, causing some degrees of freedom to "leak" into the final fit. One might guess that "leakage" will tend to disappear as the sample size n gets large. But that is not borne out in the results for $n = 200$.

The rightmost column of Table 2 gives a parametric bootstrap estimate of degrees of freedom. Fixing the expression data, new response values were generated as $y_j^* \sim \hat{\mu}_j + \epsilon_j^*$ with $\epsilon_j^* \sim N(0, \hat{\sigma}^2)$, $\hat{\sigma}^2$ being the usual unbiased estimate of variance. Using five such sets of bootstrap values, the covariance expression in (5) was estimated. We see that the bootstrap does a reasonable job of estimating the degrees of freedom, although sometimes it underestimates the actual value. The bootstrap method can be applied to general situations where the actual value of degrees of freedom is not available.

In Table 3, we generated a scenario closer to actual microarray experiments. Here $n = 50$ and $p = 1000$, with the expression values again being standard independent Gaussian. Since $p > n$ we cannot use least squares regression to construct the microarray predictor z, so we used pre-validated

Table 2: *Simulation results. Degrees of freedom of pre-validated predictor from formula (6) and parametric bootstrap estimate. Ideal value is 2.*

p	Formula (se)	Parametric bootstrap (se)
	Null case, $n = 50$	
2	4.04 (0.21)	3.25 (0.18)
5	3.40 (0.09)	2.72 (0.12)
10	2.95 (0.05)	2.73 (0.10)
20	2.89 (0.02)	2.64 (0.09)
	Null case, $n = 200$	
2	6.44 (0.63)	3.74 (0.31)
5	3.95 (0.22)	2.98 (0.16)
10	3.39 (0.09)	2.73 (0.14)
20	3.09 (0.06)	2.64 (0.10)
	Non-null case, $n = 50$	
2	1.35 (0.09)	1.32 (0.13)
5	1.19 (0.03)	0.98 (0.09)
10	1.15 (0.01)	0.68 (0.14)
20	1.30 (0.02)	0.56 (0.10)
	Non-null case, $n = 200$	
2	1.61 (0.19)	1.64 (0.19)
5	1.17 (0.02)	1.07 (0.12)
10	1.12 (0.01)	1.01 (0.13)
20	1.11 (0.01)	0.79 (0.19)

Table 3: *Simulation results- large example. As in Table 2, with the ideal value for degrees of freedom equal to 2.*

Formula (se)	Parametric bootstrap (se)
Null case	
6.48 0.4	7.26 0.47
Non-Null case	
2.70 (.34)	3.65 (.50)

ridge regression. As before, the outcome was generated as $y = x^T\beta + \epsilon$ and linear least squares was used for the final model to predict y from $\tilde{z}$ and c, with $\beta \sim N(0, .05^2)$ in the non-null case.

In this setup the mapping $f_{X,y}(x)$ is not linear, so it is not convenient to use formula (6). Hence we computed the covariance expression on the right-hand side of (5) directly. The results in Table 3 show that leakage is again a problem, in the null case.

Our conclusion from these studies is that pre-validation greatly reduces the degrees of freedom of the microarray predictor, but does not reliably reduce it to the ideal value of one. Hence we recommend that for each application of pre-validation, a parametric bootstrap be used to estimate the degrees of freedom. This is illustrated in the breast cancer example in the next section.

6 Further analysis of the breast cancer data

We re-analyzed the breast cancer data from van't Veer et al. (2002). The authors use the following steps:

1. Starting with 24,881 genes, they apply filtering based on fold-change and a p-value criterion, to yield 4936 genes (personal communication from authors).

2. They select the genes have absolute correlation $\geq .3$ with the class labels, giving 231 genes

3. They find the 231-dimensional centroid vector for the 44 good-prognosis cases

4. They compute the correlation of each case with this centroid and choose a cutoff value so that exactly 3 of the poor groups are misclassified. This value turned out to be .38. Finally they classify to the good prognosis group if the correlation with the normal centroid is $\geq .38$, otherwise they classify to the poor prognosis group.

5. Starting with the top $5, 10, 15\ldots$ genes, they carried out this classification procedure with leave-one-out cross-validation, to pick the optimal number of genes. reporting an optimal number of 70

Table 4: *Odds ratios from pre-validated data*

Predictor	Friend *et. al.*	Our analysis
microarray	17.6	4.7
angio	4.7	4.9
ER	1.7	1.9
grade	1.1	2.1
PR	2.1	1.7
age	4.0	4.3
size	3.5	2.7

Even with some help from the authors, we were unable to exactly reproduce this analysis. At stages 2 and 3, we obtained 4918 and 235 genes, respectively. The authors told us they could not release their list of 4936 genes for legal reasons. We fixed the number of genes (70) in step 5.

The authors carry out what we call a pre-validation analysis in the supplementary material to their paper. Table 4 shows the odds ratios from the pre-validated data, both from their analysis and our. The odds ratios for the microarray predictor differ greatly, and we were unable to reproduce their results.

We estimated the degrees of freedom for the full model in Table 4 numerically, using the bootstrap method of the previous section, obtaining an estimate of 9.0, with a standard error of .71. The nominal value would be 8 (7 predictors plus the intercept), so about one degree of freedom has leaked.

In the remainder of this section we apply *full cross-validation* to this dataset Pre-validation is a partial form of cross-validation that is especially convenient when the microarray score "z" might be applied in a wide variety of possible models.

The model illustrated in Figure 1 can be described as follows: for each case j, a score z_j is constructed from the 78 × 4936 microarray data matrix X and the 78 dimensional response vector $\mathbf{y}$, say

$$z_j = f(x_j|X, \mathbf{y}) \tag{8}$$

according to algorithm (1)–(5) at the beginning of this Section. The notation in (8) indicates that $(X, \mathbf{y})$ determines the form of the rule f, which is then evaluated at the 4936-vector x_j for that case's microarray. The final prediction for case j is

$$\widehat{\mu}_j = g(c_j, z_j | \mathbf{c}, \mathbf{z}). \tag{9}$$

Here g is the logistic regression rule based on the 78×6 matrix of covariates $\mathbf{c}$ and the vector $\mathbf{z}$ of 78 z scores, then evaluated at (c_j, z_j), the vector of 6 covariates and z_j (the top panel of Table 1 was based on the predictions $\widehat{\mu}_j$).

Full cross-validation modifies (8)–(9) so that the data for case j is not involved in constructing the form of the rules for its own prediction. Let $X_{(j)}$ indicate the 77×4937 matrix obtained by deleting column x_j from X, and likewise $\mathbf{c}_{(j)}, \mathbf{y}_{(j)}$ etc. The cross-validated predictor $\widetilde{\mu}_j$ is obtained as

$$\widetilde{z}_j = f(x_j | X_{(j)}, \mathbf{y}_{(j)}) \tag{10}$$

and

$$\widetilde{\mu}_j = g(c_j, \widetilde{z}_j | \mathbf{c}_{(j)}, \widetilde{\mathbf{z}}_{(j)}). \tag{11}$$

(By contrast, the pre-validated predictors used in the bottom of Table 1 employed $g(c_j, \widetilde{z}_j | \mathbf{c}, \widetilde{z})$.)

Full cross-validation permits an almost unbiased estimate of the prediction error we would obtain if rule (9) were applied to an independent test set. If $\mathcal{Q}(y_j, \hat{\mu}_j)$ is a measure of error for predicting that outcome y_j by $\widehat{\mu}_j$, then

$$\widehat{\text{Err}} = \frac{1}{n} \sum_{i=1}^{n} \mathcal{Q}(y_j, \hat{\mu}_j) \tag{12}$$

is nearly unbiased for the expected error rate of rule $\widehat{\mu}_j$ applied to an independent set of test cases, see Efron & Tibshirani (1997) and Efron (1983).

Table 5 refers to the error function

$$\begin{array}{rcl} \text{If } y_j & = & 1 \text{ and } \widehat{\mu}_j \le 34/78 \\ \text{or } y_j & = & 0 \text{ and } \widehat{\mu}_j > 34/78 \\ & & \text{then } \mathcal{Q}(y_j, \widehat{\mu}_j) = 1 \end{array} \tag{13}$$

and $\mathcal{Q}(y_j, \widehat{y}_j) = 0$ otherwise. In other words, $\widehat{\mathrm{Err}}$ is the proportion of prediction errors (with the prediction threshold set at 34/78 rather than 1/2 to reflect the 44/34 division of cases in the training cases.)

Table 5: *Estimates of prediction error for two logistic regression models: c alone (only the 6 covariates) and c plus z (6 covariates plus the microarray predictor z.) Naive reuse method suggests that adding z cuts the error rate nearly in half, from 26.9% to 14.1%. Most of the apparent improvement disappears under full cross-validation, where now the comparison is 29.5% versus 28.2%. Bootstrap methods give similar results. The standard errors were obtained from jackknife calculations, and show that this experiment was too small to detect genuine differences of less than about 10%.*

Model	Re-use	cross-val	(sterr)	zero-boot	(sterr)	632+ boot	(sterr)
c alone:	0.269	0.295	(0.052)	0.341	(0.078)	0.320	(0.062)
c plus z:	0.141	0.282	(0.076)	0.342	(0.067)	0.301	(0.068)
difference:	0.128	0.013	(0.091)	-0.001	(0.072)	0.019	(0.066)

Table 5 compares the prediction error rates (13) from two logistic regression models: one based on just the six covariates in **c**, the other using these plus the microarray predictor z. The naive reuse error rates make z look enormously helpful, reducing $\widehat{\mathrm{Err}}$ from 26.9% to 14.1%. Most of z's advantage disappears under cross-validation, giving $\widehat{\mathrm{Err}}(c) = 29.5\%$ versus $\widehat{\mathrm{Err}}(c, z) = 28.2\%$, for a difference of only 1.3%.

Another important point is clear from Table 2: the cross-validated difference of 1.3% has an estimated standard error of ±9.1%. In other words there is not enough data in the van't Veer et al. study to establish a prediction advantage for z of less than say 10% even if it exists (which does not appear to be the case.)

The cross-validation in Table 5 grouped the 78 cases into 13 groups of 6 cases each, so that calculations (11)-(12) produced $\widetilde{z}_j$ and $\widetilde{\mu}_j$ values six at a time. The jackknife was used to calculate standard errors: a typical jackknife pseudo-value deleted one of the 13 groups and repeated the cross-validation calculations using just the data from the other 12 groups, finally obtaining the standard error as in (11.5) or (11.15) of Efron & Tibshirani (1993).

Cross-validation can be overly variable for estimating prediction error (13), see Efron (1983). Table 2 also reports two bootstrap estimates described in Efron & Tibshirani (1997): "zero-boot" and the "632+ rule", equations 17 and 27-29 of that paper, with $\widehat{\gamma} = .50$), the latter having a particularly good track record. The story here does not change much, though 632+ gives a slightly larger difference estimate, 1.9%, with a smaller standard error, ±6.6%. Bootstrap estimates were based on $B = 600$ bootstrap replications, with standard errors estimated by the jackknife-after-bootstrap computations of Efron (1992).

7 Discussion

In this paper we have analyzed pre-validation, a technique for deriving a fairer version of an adaptively chosen predictor. It seems especially well-suited to microarray problems. The promise of pre-validation is that the resulting predictor will act similarly to one that has been derived from an independent dataset. Hence when included in a model with clinical predictors, it should have have one degree of freedom. This is contrast to the usual (non pre-validated) predictor, which has degrees of freedom equal to the total number of parameters fit in each of the two stages of the analysis.

We have found that pre-validation is only partially successful in achieving its goal. Generally it controls the degrees of freedom of the predictor, as compared to the non pre-validated version. However in null situations where the microarray predictor is independent of the response, degrees of freedom can "leak" from one case to another, so that the total degrees of freedom of the pre-validated predictor is more than the ideal value of one. Conversely, in non-null settings, the total degrees of freedom of the pre-validated predictor can be less than expected. Overall we recommend use of the parametric bootstrap, to estimate the degrees of freedom of the pre-validated predictor. With the estimated value of degrees of freedom in hand, one can use the pre-validated predictor along with clinical predictors, in a model to compare their predictive accuracy.

Finally, while pre-validation is a promising method for building and assessing an adaptive predictor on the same set of data, it is no substitute for full cross-validation or test set validation, in situations where there is sufficient data to use these techniques.

Acknowledgments: Tibshirani was partially supported by NIH grant 2 R01 CA72028, and NSF grant DMS-9971405. Efron was partially supported by NIH grant 2R01 CA59039 and NSF grant DMS-0072360.

Appendix: proof of formula (6)

Formula (6) concerns pre-validation in linear model situations. We compute $\mathbf{z} = A\mathbf{y}$ for a fixed $n \times n$ matrix A, and then $\widehat{\mu} = P\mathbf{y}$ where P is the $n \times n$ projection matrix into the linear space spanned by the columns of the $n \times (k+1)$ matrix $M = (\mathbf{c}, \mathbf{z})$, $\mathbf{c}$ being the $n \times k$ matrix of fixed covariates:

$$P = MG^{-1}M^T \qquad (G = M^TM). \tag{$A1$}$$

Notice that $P = P(\mathbf{y})$ is not fixed, being a function of $\mathbf{y}$ though $\mathbf{z}$.

$$\text{Define} \quad P^\perp = I - P, \quad A^\perp = P^\perp A, \quad \text{and} \quad \mathbf{z}_{(c)} = (I - \mathbf{c}(\mathbf{c}^T\mathbf{c})^{-1})\mathbf{z}.$$

An infinitesimal change in the response vector, $\mathbf{y} \to \mathbf{y} + d\mathbf{y}$, changes $\mathbf{z}$ by amount $d\mathbf{z} = Ad\mathbf{y}$, which we can write as

$$d\mathbf{z} = Pd\mathbf{z} + P^\perp d\mathbf{z} \equiv d\widehat{\mathbf{z}} + d\mathbf{z}^\perp. \tag{$A2$}$$

The resulting change in P is calculated as follows.
Lemma

$$dP = (d\mathbf{z}^\perp \mathbf{z}_{(c)}^T + \mathbf{z}_{(c)} d\mathbf{z}^{\perp T})/\|\mathbf{z}_{(c)}\|^2. \tag{$A3$}$$

Proof. Changes in $\mathbf{y}$ affect P only through changes in $\mathbf{z}$. Moreover, the component $d\widehat{\mathbf{z}}$ has no effect on the projection matrix P since it preserves the linear space M, so we can consider dP to be a function of only $d\mathbf{z}^{\perp}$. The change in $G = M^T M$ is zero to first order,

$$G + dG = (\mathbf{c}, \mathbf{z} + d\mathbf{z}^{\perp})^T(\mathbf{c}, \mathbf{z} + d\mathbf{z}^{\perp}) \doteq G, \tag{$A4$}$$

since $\mathbf{c}^T d\mathbf{z}^{\perp} = 0 = \mathbf{z}^T d\mathbf{z}^{\perp}$. Thus

$$\begin{aligned} P + dP \quad &\doteq \quad [(\mathbf{c}, \mathbf{z}) + (\mathbf{0}, d\mathbf{z}^{\perp})]G^{-1}\left[\begin{pmatrix}\mathbf{c}^T \\ \mathbf{z}^T\end{pmatrix} + \begin{pmatrix}\mathbf{0}^T \\ d\mathbf{z}^{\perp}\end{pmatrix}\right] \\ &\doteq \quad P + (\mathbf{0}, d\mathbf{z}^{\perp})G^{-1}\begin{pmatrix}\mathbf{c}^T \\ \mathbf{z}^T\end{pmatrix} + (\mathbf{c}, \mathbf{z})G^{-1}\begin{pmatrix}\mathbf{0} \\ d\mathbf{z}^{\perp T}\end{pmatrix}, \end{aligned}$$

or

$$dP \doteq d\mathbf{z}^{\perp}(G^{21}, G^{22})\begin{pmatrix}\mathbf{c}^T \\ \mathbf{z}^T\end{pmatrix} + (\mathbf{c}, \mathbf{z})\begin{pmatrix}G^{12} \\ G^{22}\end{pmatrix}dz^{\perp T} \tag{$A5$}$$

Here we have partitioned G^{-1} into

$$G^{-1} = \begin{pmatrix}G^{11} & G^{12} \\ G^{21} & G^{22}\end{pmatrix} \qquad \text{(with } G^{21} = G^{21T}\text{).} \tag{$A6$}$$

Let

$$\mathbf{v} = (\mathbf{c}, \mathbf{z})\begin{pmatrix}G^{12} \\ G^{22}\end{pmatrix}.$$

Then, also partitioning G,

$$\mathbf{c}^T\mathbf{v} = (G_{11}, G_{12})\begin{pmatrix}G^{12} \\ G^{22}\end{pmatrix} = 0 \quad \text{and} \quad \mathbf{z}^T\mathbf{v} = (G_{21}, G_{22})\begin{pmatrix}G^{12} \\ G^{22}\end{pmatrix} = 1, \tag{$A7$}$$

(A7) following from

$$\begin{pmatrix}G_{11} & G_{22} \\ G_{21} & G_{22}\end{pmatrix}\begin{pmatrix}G^{11} & G^{12} \\ G^{21} & G^{22}\end{pmatrix} = \begin{pmatrix}I, & 0 \\ 0, & 1\end{pmatrix}.$$

Since $\mathbf{v}$ is a linear combination of $\mathbf{z}$ and the columns of $\mathbf{c}$, $\mathbf{c}^T\mathbf{v} = 0$ shows that $\mathbf{v}$ must lie along the projection of $\mathbf{z}$ orthogonal to $\mathbf{c}$, that is $\mathbf{v} = \gamma\mathbf{z}_{(c)}$.

Also $\mathbf{z}^T\mathbf{v} = 1$ implies $\gamma = 1/\|\mathbf{z}_{(c)}\|^2$, or $\mathbf{v} = \mathbf{z}_{(c)}/\|\mathbf{z}_{(c)}\|^2$. The Lemma follows from (A5).

Lemma (A3) combined with $d\mathbf{z}^\perp = A^\perp d\mathbf{y}$ gives

$$\frac{\partial P_{ij}}{\partial y_i} = (A^\perp_{ii} z_{(c)j} + A^\perp_{ij} z_{(c)i})/\|\mathbf{z}_{(c)}\|^2. \qquad (A7)$$

Finally

$$\frac{\partial \widehat{\mu}_i}{\partial y_i} = \frac{\partial}{\partial y_i} \sum_j P_{ij} y_j = P_{ii} + \sum_j \frac{\partial P_{ij}}{\partial y_i}\, y_j$$

so

$$\begin{aligned}\sum_i \frac{\partial \widehat{\mu}_i}{\partial y_j} &= \sum_i P_{ii} + \sum_j \frac{\partial P_{ij}}{\partial y_i}\, y_j \\ &= (k+1) + \sum_i \sum_j (A^\perp_{ii} z_{(c)j} + A^\perp_{ij} z_{(c)i}) y_j/\|\mathbf{z}_{(c)}\|^2.\end{aligned}$$

which is the top version of (6). The bottom version follows since

$$\widehat{\mu}^T A^\perp \mathbf{z}_{(c)} = (\widehat{\mu}^T P^\perp)(A\mathbf{z}_{(c)}) = \mathbf{0}^T(A\mathbf{z}_{(c)}) = 0$$

Note: it is not necessary for the first mapping $\mathbf{z} = f(\mathbf{y})$ to be linear. Result (6) holds in the nonlinear case if A is defined to be the matrix of partial derivatives

$$A = (\partial z_i/\partial y_j).$$

References

Dudoit, S., Fridlyand, J. & Speed, T. (2001), 'Comparison of discrimination methods for the classification of tumors using gene expression data', *J. Amer. Statist. Assoc* pp. 1151–1160.

Efron, B. (1983), 'Estimating the error rate of a prediction rule: some improvements on cross-validation', *J. Amer. Statist. Assoc.* **78**, 316–331.

Efron, B. (1986), 'How biased is the apparent error rate of a prediction rule?', *J. Amer. Statist. Assoc.* **81**, 461–70.

Efron, B. (1992), 'Jackknife-after-bootstrap standard errors and influence functions (with discussion)', *J. Royal. Statist. Assoc. B.* **54**, 83–111.

Efron, B., Hastie, T., Johnstone, I. & Tibshirani, R. (2002), Least angle regression, Technical report, Stanford University.

Efron, B. & Tibshirani, R. (1993), *An Introduction to the Bootstrap*, Chapman and Hall, London.

Efron, B. & Tibshirani, R. (1997), 'Improvements on cross-validation: the 632+ bootstrap: method', *J. Amer. Statist. Assoc.* **92**, 548–560.

Hastie, T. & Tibshirani, R. (1990), *Generalized Additive Models*, Chapman and Hall, London.

Stein, C. (1981), 'Estimation of the mean of a multivariate normal distribution', *Annals of Statistics* **9**, 1131–1151.

van't Veer, L. J., van de Vijver, H. D. M. J., He, Y. D., Hart, A. A., Mao, M., Peterse, H. L., van der Kooy, K., Marton, M. J., Anke T. Witteveen and, G. J. S., Kerkhoven, R. M., Roberts, C., Linsley, P. S., Bernards, R., & Friend, S. H. (2002), 'Gene expression profiling predicts clinical outcome of breast cancer', *Nature* **415**, 530–536.

Wolpert, D. (1992), 'Stacked generalization', *Neural Networks* **5**, 241–259.

Ye, J. (1998), 'On measuring and correcting the effects of data mining and model selection', *J. Amer. Statist Assoc.* **93**(413), 120–131.