Homework 4

Daniel Hartig

April 11, 2017

1 d.

The observed cross-validation errors are low, in the range of 0.2. This is much lower than would be expected. Since the feature values are randomly generated, and the labels are randomly applied, the expected accuracy of any model is 0.5. Any model accuracy greater than this is purely ramdom and does not have any predictive value.

What has happened is that we used a very large feature set of 10000 compared to only 100 samples. By selecting a subset of the features that had the highest magnitude, we were selecting the features that had either the highest postive or negative correlation with the randomly assigned labels. For most of the randomly built features, since the normal distribution is evenly split between positive and negative values, the expected magnitude of c_j would be zero, as both the positive and negative feature values would have an equal probability of being multiplied by positive or negative labels in the formula for c_j.

However, with a feature set 100 times larger than the sample set, we are bound to find some features where most of the negative feature values line up with negative y-labels, and the positive feature values with positive y-labels to produce mostly postitive products $X_{ij}y_i$. Alternately, the converse could happen, where the product of feature values and y-labels is mostly negative. In either case, the magnitude will be larger if there is a random correlation between features values and y-labels, and these highly correlated features are precisely what we are selecting by taking the five features with the largest magnitude in part b.

As a result, these five features *appear* to have high predictive power for the training dataset, even when the dataset is split using cross-validation. In conclusion, with a relatively small sample set and relatively large feature space, there is a danger of random predictive associations appearing that are not justified when applied to a larger test set.

2 a.

The best feature subset is the 0-indexed feature set $(3, 8, 9)$. These three features produce a cross validation error of 0.2.

2 b.

The error on the test set is 0.497, which is not significantly different from random assignment of classifications, which would be expected to produce a test error of 0.5.

If we investigate the 15 features of this data by calculating the 'error' between the sign of the feature and the value of `ytrain`, we find that most of of them have an error of about 0.5. However, the feature with 0-index of 9 has an error of 0.725. For a sample size of 40, this means that the value of feature 9 has the opposite sign as `ytrain` for 29 of the 40 samples. The probability of this happening is is binomial with $p = 0.5$ chance of both this feature and `ytrain` having the same sign, $n = 40$ and $k = 29$, and so has probability 0.002. The probability that a single one of 15 features will so closely reflect the sample classification is 0.031; not likely, but not particularly unlikely.

If feature 9 is weighted negatively, it would provide about a 0.275 error rate by itself, no matter how the sample was divided for cross validation. As the test set showed, this feature is not generally predictive, yet the biologists had the misfortune of having a feature appear to be predictive due to a low probability alignment between that feature and the training classification. In

the light of the test set, I would tell the biologists that their prediction system was the unfortunate product of a too-small training set.

2 c.

If we randomly reassigned the values of `ytrain`, we can test to see if the 80% accuracy that we found is exceptional, or if it is likely given the size of our sample and the number of features. For example, if we fit a model to a random classification, and we can still find a feature set that has about 80%, we will have a good reason to doubt that the model we have is as good as claimed.

2 d.

Rerunning the model 20 times with a randomly shuffled `ytrain` give the following output:

```
Subset 1 complete
Indices: (1, 5, 7, 8, 11, 14)
Cross Validation Error: 0.15

Subset 2 complete
Indices: (2, 4, 11, 14)
Cross Validation Error: 0.3

Subset 3 complete
Indices: (2, 3, 6, 8, 10)
Cross Validation Error: 0.2

Subset 4 complete
Indices: (2, 9)
Cross Validation Error: 0.175

Subset 5 complete
Indices: (0, 1, 5, 8, 10, 12)
Cross Validation Error: 0.175

Subset 6 complete
Indices: (0, 5, 8, 14)
Cross Validation Error: 0.325

Subset 7 complete
Indices: (0, 1, 2, 3, 4, 6, 7, 8, 10, 13)
Cross Validation Error: 0.175

Subset 8 complete
Indices: (0, 1, 2, 4, 5, 7, 8, 10, 11, 13)
Cross Validation Error: 0.25

Subset 9 complete
Indices: (0, 1, 5, 6, 7, 12)
Cross Validation Error: 0.175

Subset 10 complete
Indices: (3, 4, 5, 6, 9)
Cross Validation Error: 0.2

Subset 11 complete
Indices: (1, 7, 9, 11, 14)
```

```
Cross Validation Error: 0.275

Subset 12 complete
Indices: (0, 3, 5, 11, 13)
Cross Validation Error: 0.25

Subset 13 complete
Indices: (7, 8, 10)
Cross Validation Error: 0.3

Subset 14 complete
Indices: (3, 4, 9, 11)
Cross Validation Error: 0.325

Subset 15 complete
Indices: (3, 5, 6, 7, 8, 10, 11)
Cross Validation Error: 0.225

Subset 16 complete
Indices: (1, 3, 5, 11)
Cross Validation Error: 0.3

Subset 17 complete
Indices: (0, 9, 11)
Cross Validation Error: 0.35

Subset 18 complete
Indices: (7,)
Cross Validation Error: 0.45

Subset 19 complete
Indices: (9,)
Cross Validation Error: 0.325

Subset 20 complete
Indices: (4, 5, 7, 8, 10, 12)
Cross Validation Error: 0.225
```

As we can see, a cross-validation error of 0.2 is not unusually high; 7 of 20 random permutations of `ytrain` produced a model that was this accurate. Performing this permutation test on a model would be a good way to determine if the model is likely to be effective, or if its good results are random noise.