On Bias, Variance, 0/1 - Loss, and the Curse of Dimensionality

RK

April 13, 2014

Abstract

The purpose of this document is to summarize the main points from the paper, “On Bias, Variance, 0/1 - Loss, and the Curse of Dimensionality”, written by Jerome H.Friedman(1997).

1 Introduction

It is intuitively obvious that if you overfit a model to the training data, training error goes down but the test error might increase. This is the classic bias-variance trade off that is kept in mind by any modeler. There are obviously methods that sacrifice bias for decrease in variance of the estimator. In regression framework, one usually puts in constraints on parameters and thus achieves an estimate of lesser variance. LASSO, Smoothing Spline methods , Principal component regression etc fall in this category. In the classification framework, the math is slightly different as the the loss function is not L2 penalty but a 0/1 loss function.

In a typical classification problem, a function of covariates serves as a probability estimate of the observation belonging to a certain class. Based on these estimates, the forecast of the most likely categorical value of the output is made. The paper in the introduction mentions an important point that,

> Much research in classification has been devoted to achieving higher accuracy probability estimates under the presumption that this will generally lead to more accurate predictions. This need not always be the case.

So, the objective of the paper is to show that accurate probability estimates do not necessarily lead to better classification performance and often can make it worse. The paper has implication on the kind of methods one might end up choosing for the classification problem. Instead of choosing a fancy method that improves probability estimate, simple models such as naive Bayes and nearest neighbor methods might yield lower prediction error. Even though such methods produce poor probability estimates, it is ok as long as the variance of the estimator is low.

2 Classification

This section gives the generic math behind of the classification. It deals with the 2 class problem, i.e $y \in \{0, 1\}$. The goal of a classification procedure is to predict the output value given the value of a set of input variable $\mathbf{x}$. It is often the case that at a $\mathbf{x} \in R^n$, the value of y is not uniquely determinable. It can assume both its values with respective probabilities that depend on the location of the point $\mathbf{x}$ in the n dimensional space.

$$Pr(y = 1|\mathbf{x}) = f(x) = 1 - Pr(y = 0|\mathbf{x})$$

The role of classification procedure is to produce a rule that makes a prediction $\hat{y} \in \{0,1\}$ for the correct class label y at every input $\mathbf{x}$. The goal is to choose $\hat{y} \in \{0,1\}$ to minimize inaccuracy as characterized by the misclassification risk

$$r(\mathbf{x}) = l_1 f(\mathbf{x}) 1(\hat{y}(\mathbf{x}) = 0) + l_0(1 - f(\mathbf{x}))1(\hat{y}(\mathbf{x}) = 1)$$

The above expression can be further simplified as follows

$$\begin{aligned} r(\mathbf{x}) &= l_1 f(\mathbf{x})(1 - 1(\hat{y}(\mathbf{x}) = 1)) + l_0(1 - f(\mathbf{x}))1(\hat{y}(\mathbf{x}) = 1) \\ &= l_1 f(\mathbf{x}) - l_1 f(\mathbf{x}) 1(\hat{y}(\mathbf{x}) = 1)) - l_0 1(\hat{y}(\mathbf{x}) = 1) + l_0 f(\mathbf{x})) \\ &= l_1 f(\mathbf{x}) - (l_1 + l_0) f(\mathbf{x}) 1(\hat{y}(\mathbf{x}) = 1)) + l_0 f(\mathbf{x})) \\ &= (l_1 + l_0) \left\{ \frac{l_1}{l_1 + l_0} f(\mathbf{x}) - 1(\hat{y}(\mathbf{x}) = 1)) \cdot \left(f(\mathbf{x}) - \frac{l_0}{l_0 + l_1} \right) \right\} \end{aligned}$$

The risk function can be minimized by the Bayes rule

$$y_B(x) = 1 \left(f(\mathbf{x}) \geq \frac{l_0}{l_0 + l_1} \right)$$

This achieves the lowest possible risk

$$r_B(\mathbf{x}) = \min(l_1 f(\mathbf{x}), l_0(1 - f(\mathbf{x})))$$

The above Bayes boundary is unknown for any system. Modeling involves using training data to learn a classification rule $\hat{y}(\mathbf{x}|T)$ for future prediction. The usual paradigm for accomplishing this is to be use the training data T to form an approximation $\hat{f}(\mathbf{x}|T)$ to $f(\mathbf{x})$ and get

$$\hat{y}(\mathbf{x}|T) = 1 \left(\hat{f}(\mathbf{x}|T) \geq \frac{l_0}{l_0 + l_1} \right)$$

When $\hat{f}(x) \neq f(\mathbf{x})$, this rule may be different from the Bayes rule and thus not achieve the minimal Bayes risk. The paper takes $l_0 = l_1 = 1$ for expository purpose. Hence the Bayes boundary

$$y_B(x) = 1 \left(f(\mathbf{x}) \geq \frac{1}{2} \right)$$

$$\hat{y}(\mathbf{x}|T) = 1 \left(\hat{f}(\mathbf{x}|T) \geq \frac{1}{2} \right)$$

3 Function estimation

The usual approach is to assume a relation between

$$y = f(\mathbf{x}) + \epsilon$$

and an estimate of

$$\hat{f}(\mathbf{x}|T) = \hat{E}(y|\mathbf{x}, T)$$

using the training set is made.

In the case of logit function, the expectation is assumed to be a sigmoid function and a logistic model is fit to obtain $\hat{f}(\mathbf{x}|T)$. This is plugged in to

$$\hat{y}(\mathbf{x}|T) = 1\left(\hat{f}(\mathbf{x}|T) \geq \frac{1}{2}\right)$$

to get an estimate of class label.

4 Density estimation

There are many methods such as LDA, QDA, Mixture models, learning vector quantization techniques, etc where the classification is done vial Bayes theorem. Separate densities are fitted for each class and based on the priors of each class, the posterior probabilities of each class is computed and a voting framework is adopted to estimate class label.

5 Bias, variance and estimation error

There is an obvious association between training data and the function estimate. You change the training data and you get a different estimate. This dependency and the performance of the model on test data is aptly summarized by the following bias-variance tradeoff relation

$$E_T(y - \hat{f}(\mathbf{x}|T))^2 = [f(\mathbf{x}) - E_T\hat{f}(\mathbf{x}|T)]^2 + E_T[\hat{f}(\mathbf{x}|T) - E_T\hat{f}(|T)]^2 + E_\epsilon(\epsilon|\mathbf{x})^2$$

$$\text{Prediction error} \;=\; \text{Bias squared} \;+\; \text{Variance} + \;\text{Irreducible error}$$

For problems with large training samples, the bias can be dominant contributor to estimation error. Hence most of the methods with big data that desire accurate probabilistic estimates naturally focus on decreasing estimation bias. For classification however this strategy has been less successful in improving performance. Some highly biased procedures such as naive Bayes and KNN remain competitive and sometimes outperform more sophisticated ones, even with moderate to large training samples. Why is that ? Answering this question is the heart of the paper and the next section delves in to the math behind it

6 Bias, variance and classification error

This section derives the bias variance framework for classification problems. The basic idea is that is this : Given a training sample T , the error rate $Pr(\hat{y}(\mathbf{x}|T) \neq y)$ (averaged over all future predictions at $\mathbf{x}$) depends on whether or not the decision agrees with that of Bayes rule. If it agrees then the error rate is the irreducible error associated with the Bayes rule, i.e $\min(1 - f(\mathbf{x}), f(\mathbf{x}))$. If it does not there is additional error component, i.e $\max(1 - f(\mathbf{x}), f(\mathbf{x}))$.

$$Pr(\hat{y}(\mathbf{x}|T) \neq y) = |2f(\mathbf{x}) - 1| \cdot 1[\hat{y}(\mathbf{x}|T) \neq y_B(\mathbf{x})] + Pr(y \neq y_B(\mathbf{x}))$$

Suppressing the dependency on $\mathbf{x}$it is

$$Pr(\hat{y} \neq y) = |2f - 1| \cdot 1[\hat{y} \neq y_B] + Pr(y \neq y_B)$$

By considering a normal distribution for $\hat{f}$, the author derives an approximate expression for the classification error

$$Pr(\hat{y} \neq y_B) = \overline{\psi}\left[\text{sign}\,(f - 1/2)\frac{E\hat{f} - 1/2}{\sqrt{\text{var}\hat{f}}}\right]$$

7 Discussion

This is the crucial section of the paper. The author uses the following expression

$$b(f, E\hat{f}) = \text{sign}\,(1/2 - f)(E\hat{f} - 1/2)$$

and explains that the bias variance effect for a classification problem.The above expression is termed as “boundary bias”. What’s the thing to note in this expression ? - There is no explicit $E\hat{f} - f$ expression.

- For a given var $\hat{f}$, so long as the boundary bias remains negative, the classification error decreases with increasing $|E\hat{f} - 1/2|$ irrespective of the estimation bias $(f - E\hat{f})$. For positive boundary bias, the classification increases with the distance of $E\hat{f}$ from 1/2
- For a given $E\hat{f}$, so long as the boundary bias remains negative, the classification error decreases with decrease in variance. For a positive boundary bias , the error increases with decrease in variance.

The key thing to note is that our estimate $E\hat{f}$ may be off from f by a huge margin. It does not matter as long as you take care of the fact that you lie on the appropriate side of 1/2 and cut down your variance.

> The bias variance trade off is clearly very different for classification error than estimation error on the probability function f itself. The dependency of squared estimation error on $E\hat{f}$ and var$\hat{f}$ is additive whereas for classification error, there is a strong multiplicative interaction effect. The effect of boundary bias on classification error can be mitigated by low variance.

What’s the big takeaway from this section

Certain methods that are inappropriate for function estimation because of their very high bias may perform well for classification when their estimates are used in the context of a classification rule

$$\hat{y}(\mathbf{x}|T) = 1\left(\hat{f}(\mathbf{x}|T) \geq 1/2\right)$$

All that is required is a negative boundary bias and small enough variance. The procedures where the bias is caused by over smoothing have negative boundary bias. Let’s say your over smoothed estimate is

$$\hat{f}(\mathbf{x}) = (1 - \alpha(\mathbf{x}))f(\mathbf{x}) + \alpha(\mathbf{x})\overline{y}$$

where $\overline{y}$ is 1/2, i.e there are equal number of training and test cases. The boundary bias can be seen to be always negative

$$\begin{aligned}\text{sign}\,(1/2 - f)(E\hat{f} - 1/2) &= \text{sign}\,(1/2 - f)\Big((1-\alpha)f + \alpha/2 - 1/2\Big)\\ &= \text{sign}\,(1/2 - f)\Big((1-\alpha)f - (1-\alpha)/2\Big)\\ &= \text{sign}\,(1/2 - f)\Big((1-\alpha)(f - 1/2)\Big)\end{aligned}$$

The term $\text{sign}\,(1/2 - f)\Big((1-\alpha)(f - 1/2)\Big)$ has always a negative bias and thus any procedure which over-smoothes and has a low variance is a good candidate for classification problem.

R excursion

Until this point in the paper, the author provides a sound math argument behind how bias variance conspire to make methods of high bias a good candidate for classification problems. Let me take a break here and simulate some data and compute the classification error. Let

$$
\begin{aligned}
P(Y = 1) &= 0.9 \quad \text{if } 0 \leq x \leq 0.5 \\
P(Y = 0) &= 0.1 \quad \text{if } 0.5 \leq x \leq 1
\end{aligned}
$$

Let me create a master data set containing the true values, sample 100 observations from each class, fit a linear reg model and evaluate the estimate at $x = 0.48$. This will be done 5000 times to get the distribution of $p(y|x)$

```r
set.seed(1)
n         <- 10000
x         <- runif(n)
y         <- ifelse( x<0.5, rbinom(1,1,prob=0.9),rbinom(1,1,prob=0.1))
master    <- data.frame(x = x, y = y)

yhat      <- replicate(5000,{
lab0      <- sample(which(master$y==0),50)
lab1      <- sample(which(master$y==1),50)
training  <- master[rbind(lab0,lab1),]
fit       <- with(training,lm(y~x))
y.est     <- predict(fit,newdata = data.frame(x = 0.48))
y.est
})
```

The mean value of the estimate is

```r
mean(yhat)

## [1] 0.5337
```

The probability that the estimate is not equal to the Bayes rule is

```r
mean(yhat<0.5)

## [1] 0.0602
```

```
hist(yhat, breaks = 100, xlim = c(0.4,0.7), main = "P(y|x=0.48)",
     probability=T, col="blue", cex.main = 0.8, ylab ="Density" , cex.lab =0.8)
abline(v= 0.5, col = "red", lwd = 2, lty = "dashed")
text(0.46,10,"error region")
```

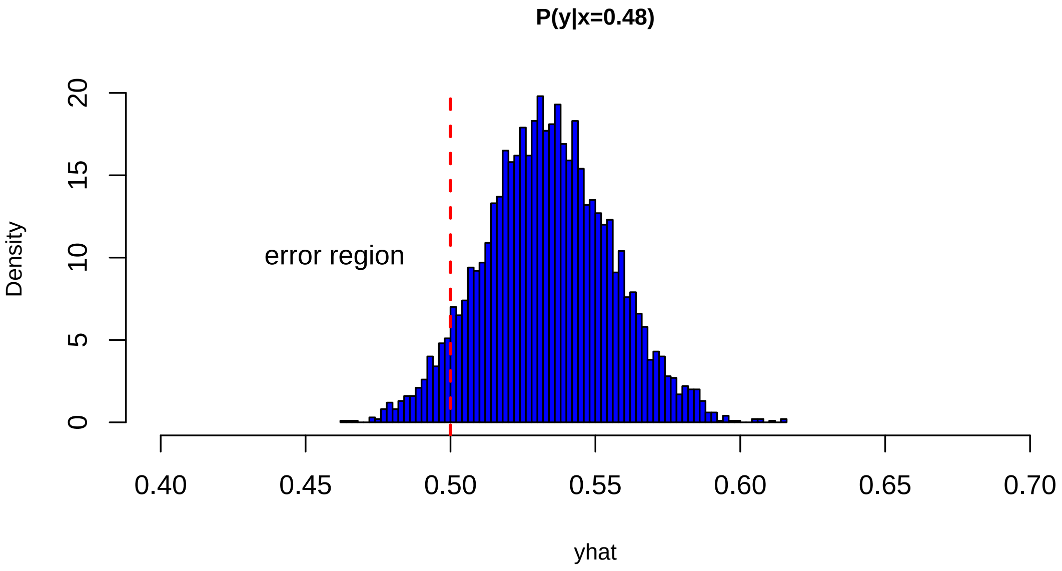

Figure 7.1: Probability estimate

Hence the classification error is

$$\begin{aligned} Pr(\hat{y} \neq y) &= |2f - 1| \cdot 0.0602 + Pr(y \neq y_B) \\ &= |2(0.9) - 1| \cdot 0.0602 + 0.1 \\ &= 0.1482 \end{aligned}$$

8 Naive Bayes methods

Naive Bayes approximates each class conditional probability density by the product of marginal densities on each input variable. Data from each class is used to estimate the density and used in Naive Bayes. Estimates obtained in this manner are clearly biased but this bias is one that results in smoothing and hence results in a negative boundary bias and hence works well for classification purpose.

9 K nearest neighbor methods

For K-nearest neighbor procedures, the bias-variance trade-off associated with estimation error is generally is driven by the bias in high dimensional setting. Like naive Bayes, the bias associated with K-nearest neighbor procedures by over-smoothing and hence decreasing the variance can have impact on reducing boundary error.

R excursion

The paper illustrates an example where the input space is n dimensional hypercube. $\mathbf{x} \in [0,1]^n$ with

$$p_0(\mathbf{x}) = 2 \cdot 1(x_1 < 1/2)$$
$$p_1(\mathbf{x}) = 2 \cdot 1(x_1 \geq 1/2)$$

so that the target probability function is

$$f(\mathbf{x}) = 1(x_1 \geq 1/2)$$

One can write a code similar to this to check out the error rates and Optimal neighbors for varying levels of training sample sizes. The following code is relevant to classification. A similar code can be written for estimation error and the results from the paper can be replicated.

```
set.seed(1)
p           <- 20
N           <- c(100,200,400,800,1600,3200,6400,12800)
n           <- 1000000
x           <- matrix(runif(n), ncol = 20, nrow = n/p)
ytrue       <- ifelse(x[,1]<1/2,0,1)
colnames(x) <- paste("x",1:p,sep="")
master      <- data.frame(x = x, y =ytrue)
i           <- 1
lab0        <- sample(which(master$y==0),20000)
lab1        <- sample(which(master$y==1),20000)
test        <- master[rbind(lab0,lab1),]
i           <- 1
results.classify <- matrix(0,nrow =length(N), ncol = 2)

for(i in seq_along(N)){
lab0        <- sample(which(master$y==0),N[i])
lab1        <- sample(which(master$y==1),N[i])
training    <- master[rbind(lab0,lab1),]
kparams     <- round(seq(from = 1,to = N[i], length.out = 100))
tune.obj    <- tune.knn(x=training[,1:20], y=factor(training[,21]), k = kparams,
                     tunecontrol = tune.control(sampling = "boot",cross=20))
kselected   <- summary(tune.obj)$best.parameters
knn.pred    <- knn(training[,1:20],test[,1:20],training[,21],k=kselected)
results.classify[i,] <- c(as.numeric(kselected), mean(knn.pred!=test[,21]) )
}
```

The results of the study show that classification error decreases at a much faster rate than squared estimator error as N increases. Also as one goes to higher dimensional data, the estimation error behaves far more badly than classification error. In that sense, high dimensionality is a problem for classification but a curse for estimation error. Also the section shows that the choice of number of nearest neighbors is less critical for classification error so long as the K is neither too small nor too large.

10 Boundary bias

The presence of negative boundary bias is crucial for high bias low variance classification methods to work. However there are cases when the negative boundary bias is violated. This section mentions a trick, i.e. bias adjustment mechanism which ensures that negative boundary bias is restored and methods like KNN classifier works well in low as well as high dimensional setting.

11 Bias plus variance in classification, Aggregated Classifiers, Limitations and future work

The last three sections go in to a little more depth. There is a section that introduces Kohavi and Wolpert decomposition and goes through the math in detail. A comment is made on the aggregated classifiers. The paper ends with a list of limitations of generalizing this framework.

Takeaway

This paper throws light on the way bias and variance conspire to make some of the so called highly biased methods perform well on test data. Naive Bayes works, KNN works and so do many such classifiers that are highly biased. This paper gives the actual math behind and shows that the additive nature of bias and variance that holds good for estimation error cannot be generalized to classification error. There is a multiplier effect, which the author calls it as "boundary bias" that makes a biased method perform well. Also this paper provides the right amount of background to explore Domingos framework that provides a nice solution to the misclassification loss function decomposition, consistent with concepts of bias and variance.