COMS 4771 Machine Learning
Problem Set #5

Jun Hu - jh3846@columbia.edu
Discussants:

July 12, 2017

Problem 1

(a) Compute the Euclidean distance between each observation and the test point:

$$D_{obs.1} = \sqrt{(0-0)^2 + (3-0)^2 + (0-0)^2} = 3$$
$$D_{obs.2} = \sqrt{(2-0)^2 + (0-0)^2 + (0-0)^2} = 2$$
$$D_{obs.3} = \sqrt{(0-0)^2 + (1-0)^2 + (3-0)^2} = \sqrt{10} = 3.162278$$
$$D_{obs.4} = \sqrt{(0-0)^2 + (1-0)^2 + (2-0)^2} = \sqrt{5} = 2.236068$$
$$D_{obs.5} = \sqrt{(-1-0)^2 + (0-0)^2 + (1-0)^2} = \sqrt{2} = 1.414214$$
$$D_{obs.6} = \sqrt{(1-0)^2 + (1-0)^2 + (1-0)^2} = \sqrt{3} = 1.732051$$

(b) Our prediction with $K = 1$ is Green. Because the nearest observation is obs.5, and obs.5 is Green, accordingly, our prediction for $x_1 = x_2 = x_3 = 0$ is Green.

(c) Our prediction with $K = 3$ is Red. Because the nearest 3 observation is obs.5, obs.6 and obs.2, obs.5 is Green, but obs.6 and obs.2 are Red, accordingly, our prediction for $x_1 = x_2 = x_3 = 0$ is Red.

(d) We would expect the best value for K is small. Because larger K takes more observations into account, of which the decision boundary is more fixed for a linear boundary. The smaller K takes less observations, and will be much more flexible based on local observations, which means it is much better for a highly nonlinear boundary.

Problem 2

We would prefer SVM to use for classification of new observations. Because for the 1-nearest neighbors, the training error rate is always 0%. For this 1NN, we know that the average error rate (averaged over both test and training data sets) is 18%, which means the test error rate is actually 36%, higher than the test error rate of SVM, which is 30%. This means SVM is better performed on the test set, so we will use SVM for classification of new observations.

Problem 3

(a) By definition, we have:

$$\begin{aligned} odds_{\text{default}} &= \frac{N_{\text{default}}}{N_{\text{non-default}}} \\ &= \frac{P(\text{default})}{P(\text{non} - \text{default})} \\ &= \frac{P(\text{default})}{1 - P(\text{default})} \end{aligned}$$

Then we get:

$$\begin{aligned} P(\text{default}) &= \frac{odds_{\text{default}}}{1 + odds_{\text{default}}} \\ &= \frac{0.37}{1 + 0.37} \\ &= 0.27 \end{aligned}$$

So the fraction of people with an odds of 0.37 of defaulting on their credit card payment will in fact default is 27%.

(b) As we have proven:

$$\begin{aligned} odds_{\text{default}} &= \frac{P(\text{default})}{1 - P(\text{default})} \\ &= \frac{0.16}{1 - 0.16} \\ &= 0.19 \end{aligned}$$

So the odds that she will default is 0.19.

Problem 4

(a) The statement **iii.** Less flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance. – is true.

The expected squared prediction error decomposes as follows:

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}[\mathbb{E}_Y[(\hat{\beta}x - Y)]] &= \mathrm{Var}(Y) + \mathrm{MSE}(\hat{\beta}x) \\ &= \sigma^2 + \mathrm{Bias}^2(\hat{\beta}x) + \mathrm{Var}(\hat{\beta}x)\end{aligned}$$

Such a decomposition is known as the **bias-variance tradeoff**. We would use lasso when least squares has very hight variance, and when the coefficients shrink toward zero, variance decreases while bias increases, which implies that lasso regression is inherently less flexible than least squares. According to the bias-variance tradeoff above, if the decrease in variance is larger than the increase in bias, accuracy will improve as the coefficients shrink.

(b) As the same reason, the statement **iii.** Less flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance. – is true to the ridge regression, which is just another regularization method as lasso.

Problem 5

Boosting using depth-one trees (or stumps) leads to an additive model – that is, each term involves only a single variable, according to the algorithm:

1. Set $\hat{f}(x) = 0$ and $r_i = y_i$ for all i in the training set.

2. Fit a tree $\hat{f}^b$ with 1 splits (2 terminal nodes) to the training data (x, r).
Let $\hat{f}^1(x) = c_1 I(x_1 < t_1) + c_1' = \frac{1}{\lambda} f_1(x_1)$.
Update $\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^1(x) = \lambda \hat{f}^1(x) = f_1(x_1)$ and $r_i = y_i - \lambda \hat{f}^1(x_i)$ for all i.
Next, we get $\hat{f}^2(x) = c_2 I(x_2 < t_2) + c_2' = \frac{1}{\lambda} f_2(x_2)$
Update $\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^2(x) = \lambda \hat{f}^1(x) + \lambda \hat{f}^2(x) = f_1(x_1) + f_2(x_2)$ and $r_i = y_i - \lambda \hat{f}^1(x_i) - \lambda \hat{f}^2(x_i)$

...

3. Eventually we get:

$$\begin{aligned} \hat{f}(x) &= \lambda \hat{f}^1(x) + \lambda \hat{f}^2(x) + \ldots + \lambda \hat{f}^p(x) \\ &= f_1(x_1) + f_2(x_2) + \ldots + f_p(x_p) \\ &= \sum_{j=1}^{p} f_j(x_j) \end{aligned}$$