

Homework 3

Due: Tuesday, October 30

1. Text, Ex 12.10
2. Text, Ex 4.4, Ex 7.7
3. Text, Ex 7.2
4. Learning the Kernel

The point of the kernel trick is two-fold

- It makes computation faster when the number of features is larger than the number of examples.
- It means that the features can be replaced with a generic inner product kernel. Typically, a gaussian kernel is used

$$K(x, z) = e^{-\gamma\|x-z\|^2}$$

for some $\gamma > 0$.

However, the success of the kernel trick depends on the choice of the kernel, or its parameters. When a radially symmetric kernel, such as the one above, is used, there is no way to ignore components of x that do not contribute to prediction.

We will investigate kernelization with the Support Vector Machine. You can find SVM code in the 'e1071' library for R, using the function `svm`.

- (a) Use a support vector machine to predict on the spam data set from last homework with a Gaussian kernel. Choose the value of γ by performance on the test set. How does performance compare with your results from Homework 2. It is often useful to consider $\gamma = e^\tau$ and search over τ .
- (b) Generate 50 "nuisance" dimensions to add as features, each independently distributed as a uniform random variable on $[0,1]$. Add these as features to the spam data and repeat the exercise above.
 - i. Compare the processing time of adding the extra dimensions.
 - ii. How does performance change, how does your optimal value of γ change?
- (c) Produce a decision tree, pruned with the 1 SE rule, for the spam data, both for the original data, and the data with the new variables. CART is implemented by the `rpart` function in the 'rpart' library. Compare the relative change in error for CART with that for SVM.

5. Models and Sensitivity

Here we will investigate the sensitivity of classification performance to model specification and outliers for LDA, logistic regression and SVMs, all using linear features.

We will model the data as a mixture of two standard bivariate normal distributions. Class 1 will have mean $(1,1)$ and class 2 will have mean $(-1,-1)$. We will assume equal probabilities (but not necessarily equal numbers) for each class. It will be useful for you to write a program to generate these data automatically.

- (a) Generate a test set of size 10,000. Graph average misclassification rates over 20 simulations for the three methods using training set sizes 10, 20, 40, 80, 160, 320, and 640. Does there appear to be an advantage to specifying a full model when it is correct? How could these results be best illuminated?
- (b) Using a training set of size 20, pick one point at random. Plot test-set misclassification error as you move its value for x_1 from -100 to +100 for each of the three methods. Use a larger range (or a different point) if you cannot see a tangible change for at least one method.