ERM

The goal of ERM is to find the optimal model fitting the data.

Suppose we have a set of data **X** and a set of corresponding label Y, therefore we have paired training data (**x**, y). Each paired data (**x**, y) is drawn from a distribution P. We pre-determine a loss function L which measures the disagreement between its arguments and outputs non-negative result. We can have a lot of estimator functions, which may or may not be the best model. All these possible estimator functions build a set *F*.

For any estimator function *f*, the risk is defined as the expectation of loss function for the data from distribution P:

If we can find a function *f\** which gives the minimum *R(f, P)*, *f\** will our best estimator. However, we do not know the underlying distribution P, so we have to use an approximation, the empirical risk, to calculate loss for our training data given f:

Now the question becomes finding a function from *F* to achieve the minimum empirical loss:

The process of finding based on empirical loss is called empirical loss minimization (ERM). Several things need to be pointed out. First, our hypothesis set *F* may not contain the “real” estimator function, which leads to approximation error; our data may not reflect the actual distribution of P, which leads to sample error. Second, ERM can be an NP-hard problem, for example when it’s a classification problem with 0-1 loss function.

To prevent overfitting, the empirical risk is often regularized to penalize complex estimators.

Coding Problem:

Data, codes and images can be found on the webpage: <http://www.unc.edu/~zyu/ml134/hw2/hw2.html>  
Observance and Explanation:

1. It is not a good strategy to choose an even k, which leads to undeterminable data.
2. Validation error and test error decreases with increasing k, but training error increases with increasing k. Too small k value indicates an over-fitting problem, which leads to high testing error as well as validation error.
3. Training error is 0 when k = 1; however, it is meaningless to use k = 1 since it is almost certain an over-fitting case.
4. When k >= 3, training error does not increase too much as k increases; however, validation error and test error drops quickly.
5. Increasing k cannot bring test error down to optimal bayes error (0.1)
6. In this case, simpler model (larger k <=10) did not increase test error. It is more likely that kNN is a good fitting strategy for this particular data set.
7. Test error of linear regression is generally higher than kNN method, indicating kNN is better at estimating the distribution than linear regression when choosing the right k value.
8. Generally speaking, increasing sample size brings down test error. This effect is more obvious when k is large.
9. When k is large, the error curve w.r.t. sample size becomes less smooth. It is reasonable considering that when k is large, you need more data points to give good estimation.