K-Nearest Neighbors and Condensed K-Nearest Neighbors   
(CS605.449 – Project 2)

By Max Robinson

# Abstract

# Problem

# Algorithm Implementation

## k-Nearest Neighbor

K-Nearest Neighbors (k-NN) is a lazy, instance based, algorithm that can do either classification or regression. Being instance based and lazy means that k-NN computes a prediction for a query instance directly from the training data that k-NN was supplied, and at the time of querying the model. The actual model for k-NN is the training data that is supplied along with the metric used for calculating distance.

K-NN works by finding k, where k is some number, instances in the model that are the k closest points in the model to the query instance. The distance calculation for the implementation for this experiment is Euclidean distance. Euclidean distance is defined as , for the distance from q to p.

After finding the k-nearest points in the model, a function is applied to retrieve the predicted value for the query instance, based on the labels or values of the k-nearest points.

For the implementation for regression in this experiment, which will be referred to as k-NN regression, the function used is a simple average function. This sums the values of the k-nearest data points together and then divides by the number of data points, k. This average is then the predicted value for the query instance.

For the implementation for classification in this experiment, referred to as k-NN classification, a majority vote of the data points is taken. This means, of the k-nearest data points, which ever label is of the k-nearest occurs the most, is the value that is given to the query instance.

## Condensed k-Nearest Neighbor

Condensed k-Nearest Neighbor uses k-NN at query time, but prior to that a smaller model of data points is constructed from the original training. This smaller data set, , is then used as the training set for k-NN when a query instance is passed to condensed k-NN.

The condensed data set is constructed as follows.

1. Start with an empty set Z.
2. While Z continues to change, repeat the following:
   1. For each data point in the training set , find a point in Z that is the closest point in Z to .
   2. If the labels of and are not the same, add to Z

In this implementation of condensed k-NN, the k-NN algorithm with a is used with the data points in Z to calculate the closest point in Z to a given instance when creating the set of condensed points.

This condensed set of data points is calculated every time a new set of training data is used, but is only calculated once. This condensed set is then used for any later query instance. This means that for instance, in cross validation, every time a new training set of data is created from the folds a condensed model is calculated.

## Scoring

### Mean Squared Error

Mean Squared error is a metric that is used to describe the average squared error for a test set of data when compared to a model. Mean Squared error can be described as the sum of the squared errors for all test data points, divided by the total number of test data points.

This error is used to determine how close a predicted regression value is to the actual value when using Euclidean distance as a distance metric.

### Error Rate

Error rate is used in this experiment to calculate how many misclassifications occurred when compared to the total number of data points classified. Error rate can be described as

# Experimental Approach

The approach for this experiment was twofold. The first part was done through data preprocessing, and the second part was done through how the experiments were run. The metric used for evaluation are MSE and Error rate for the regression and classification problems respectively.

## Data Pre-processing

Each data set used had to be preprocessed to fit the constraints of the algorithms and provide useful data for running the classifications.

For the E coli data set, there were three classes in the data set that had very few examples in the data set. To not skew our results because of a lack of data, these classes were removed from the data set. The classes removed were, omL, imL, and imS. In addition, the data was transformed into csv format with the class label as the last value in the feature vector.

The segmentation data set was originally two different data sets, a training and test set. For this experiment, the two data sets were merged together into one data set so that cross validation could be done on the entire data set. In addition, all class labels were moved to be the last feature in the feature vector.

The Forest Fire data set as manipulated to have all numeric data types. There were two specific features that were non-numeric that were transformed to be numeric, month and day of the week. The month value was replaced with an integer value corresponding to the month number on the calendar instead of a string representation of the month name. For example, “mar” was transformed to “3”. For day of the week, i.e. mon, tue, etc., was given an integer value for each day of the week where Monday was given the value “1”, and Sunday “7”. These transformations allowed all of the data to be handled in the same way and to use the same distance metric.

In the Computer Hardware data set, three feature values were removed. Vendor Name, Model Name, and ERP were all removed. Model Name was removed because the values were highly unique, and as such likely had little effect on prediction. ERP was removed because the field is actually the estimated relative performance that was calculated by the original uses of the data set. These predictions were the result of running their own linear regression algorithms. As such, these values were not the true goal values to estimate and were thus removed.

The Vendor Name was removed for consistency in how distance is calculated for this implementation of k-NN. Distance is calculated using Euclidian distance of numeric values in this implementation. Having non-numeric values would require a different distance calculation. This distance calculation could have repercussions on how the overall distance is calculated and have undesired side effects on the predicted values.

## k-NN and Choosing K

There are two types of problems k-Nearest Neighbor can solve, classification and regression. For this experiment two of the data sets used for classification, E coli and Image Segmentation, and two data sets used for regression, Computer Hardware and Forest Fire. When running k-NN for the data sets, the appropriate error function was used. Classification used error rate, while regression used MSE, as described above under the “scoring” section.

For this experiment, cross validation was used to provide robust and averaged results across different distributions of the data. Five-fold cross validation was used for this experiment. This means that from the entire set of the data, five partitions are created. For each run of k-NN the five folds are rotated through, were four of the folds are merged to create training set of 80% of the data, and one fold is used as the test set, 20%. The partition that is used for the test set is rotated so that every partition is used as the test set once.

This has the effect of running k-NN on five different training and test sets, and producing results for each. These results are then averaged and the standard deviation of the five trials is also calculated. This provides information about how well the algorithm did on average as well has how much it varied between the different test and training sets.

The performance of k-NN relies heavily on what value of k is selected. The procedure for finding the best k value for each data set and each version of k-NN, k-NN or condensed k-NN, was done as follows. For each data set, k-NN with cross validation was run with a selected value for k. K was allowed to range for k=1 to k=10. This provided results for how well a given k did on a single cross validation run. This was then repeated n number of times, where n varied per data set.

This process provided an average of the error from the cross validations for all k’s from one to ten. From this data, charts were created for each combination of data set and k-NN or condensed k-NN that was used to find the best k. The best K was then selected from the charts, where lowest error is best.

The figures below show how the error rate correlates to the number of k’s used for each data set with each algorithm, k-NN or condensed k-NN. N in this context is the number of times cross validation was run to get an average of the errors from cross validation.

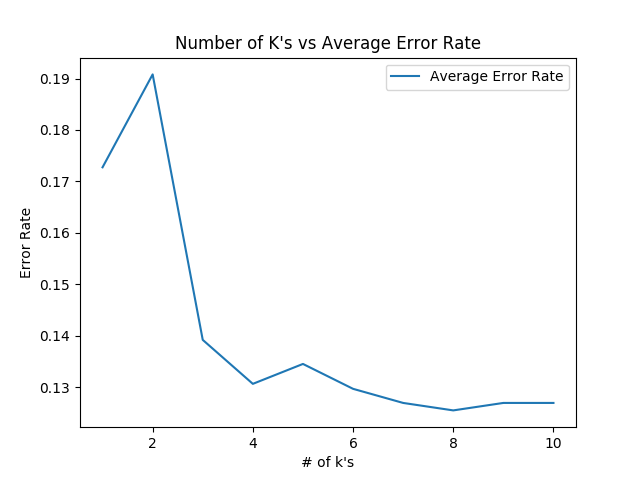
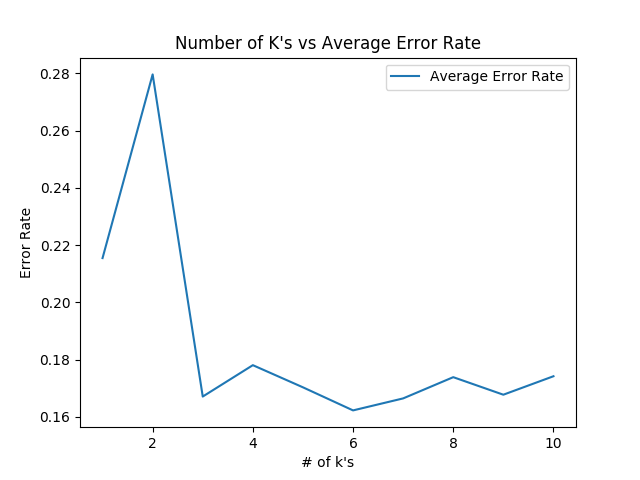


Figure 1 E coli k-NN, k selection, n=20 Figure 2: E coli Condensed k-NN, k Selection, n=20

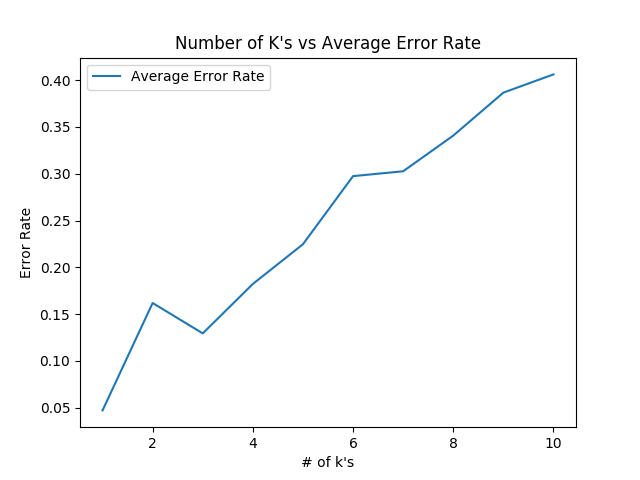
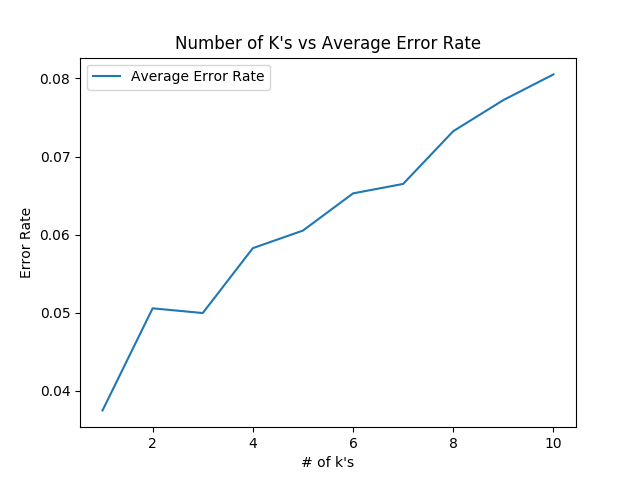


Figure 3 Segmentation Condensed k-NN, k Selection Figure 4 Segmentation Condensed k-NN, k Selection

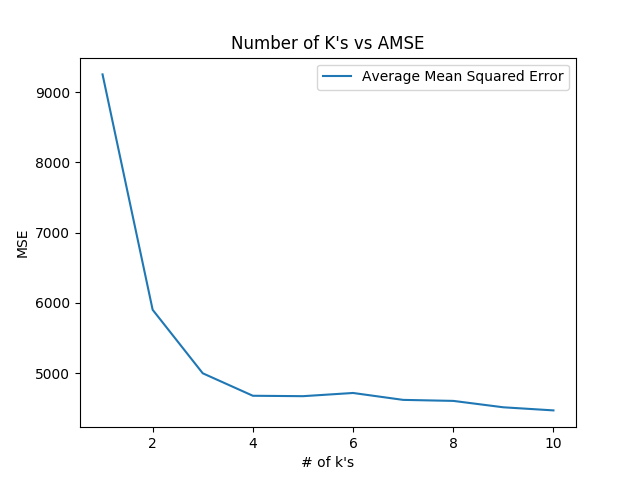
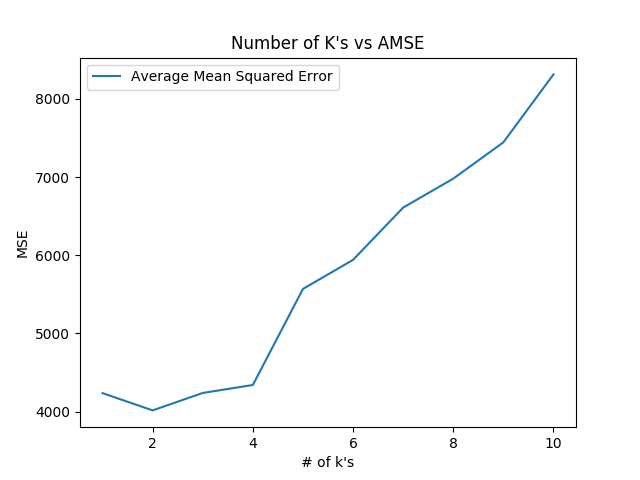


Figure 5 Computer Hardware k-NN, k selection Figure 6 Forest Fire k-NN, k selection

These figures also show the average error of each k-NN algorithm on each data set for any given k. Table 1 show for which value for k performed best for each data set and algorithm.

Table 1 k value chosen for each data set and algorithm

|  |  |  |
| --- | --- | --- |
| Dataset | Algorithm | K value chosen |
| E coli | k-NN classification | K = 3 |
| E coli | Condensed k-NN classification | K = 4 |
| Segmentation | k-NN classification | K = 1 |
| Segmentation | Condensed k-NN classification | K = 1 |
| Computer Hardware | k-NN regression | K = 2 |
| Forest Fire | k-NN regression | K = 10 |

To calculate the final results, the value for k on each algorithm and dataset was then taken and run directly for the k value with cross validation to generate the output results shown in the results section.

# Results

After the running the experiment as described in the above sections, the resulting MSE or error rate for each data set in Table 1 and Table 2 respectively. The error rate for k-NN and condensed k-NN are showed alongside each other to compare the results. These results show the MSE or error rate for k-NN or condensed k-NN with the specified best performing ‘k’ as shown in the k-NN and Choosing K section.

Table 2 MSE per Data Set with Standard Deviation for Cross Validation

|  |  |  |
| --- | --- | --- |
| Data Set | Average Mean Squared Error | Standard Deviation of Cross Validation |
| Forest Fire, k=10 | 4225.461914368932 | 4477.235155437042 |
| Computer Hardware, k=2 | 5062.4390243902435 | 4597.6086020176635 |

Table 3 Error Rate to six decimal places per Data Set for k-NN

|  |  |  |
| --- | --- | --- |
| Data Set | k-NN Error Rate | Standard Deviation of CV |
| E coli, k = 3 | 0.129032 | 0.014426 |
| Image Segmentation, k = 1 | 0.035931 | 0.009445 |

Table 4 Error Rate to six decimal places per Data Set for condensed k-NN

|  |  |  |
| --- | --- | --- |
| Data Set | Condensed k-NN Error rate | Condensed k-NN  Standard Deviation of CV |
| E coli, k = 8 | 0.190323 | 0.041310 |
| Image Segmentation, k = 1 | 0.049784 | 0.009285 |

# Behavior

# Summary

# References

## Data Sources

E coli — <https://archive.ics.uci.edu/ml/datasets/Ecoli>

Image Segmentation — <https://archive.ics.uci.edu/ml/datasets/Image+Segmentation>

Computer Hardware — <https://archive.ics.uci.edu/ml/datasets/Computer+Hardware>

Forest Fires — <https://archive.ics.uci.edu/ml/datasets/Forest+Fires>