KNN and Condensed KNN

Amrith Deepak

Abstract:

The ecoli data to classify localization sites of proteins in ecoli cells, image segmentation data to classify pixels in an image, computer hardware dataset to predict performance, and the forest fire dataset to predict the burned area of forest fires were analyzed using KNN classification(ecoli, image segmentation) and KNN regression(Computer Hardware, Forest Fire). For the classification tasks, condensed KNN was also used and compared to KNN. The performance in condensed KNN was very similar to that of KNN and there was no significant difference. For the computer hardware data, the best performance after tuning k was the same, and for the image segmentation data, performance was slightly better for regular KNN, but the difference was insignificant. For all the values of k, the performance between KNN and condensed KNN was within a couple percentage points of each other.

Problem Statement:

This project is examining how well KNN works for classification and regression and examining how condensed KNN performs compared to KNN. These algorithms are being evaluated on how they perform on different datasets to classify data. 7 attributes are being used to classify localization sites of proteins in ecoli cells, 19 attributes are being used to classify pixels of an image, 12 attributes are being used to predict forest fire area, and 6 attributes are being used to predict relative performance of a machine.

Hypothesis:

KNN when tuned properly should work well for both classification and regression when the data does not have too many dimensions. In the data here, the segmentation data has more dimensions, so the performance there is expected to be slightly lower. In general, the performance should be fairly high for the other datasets. Condensed KNN will reduce the number of observations, which can be helpful since it removes noise and irrelevant data points, but also can be harmful if it removes useful data points. Hence, the performance is predicted to be approximately the same as KNN.

Algorithms:

For KNN, the Euclidean distance between the test point and every point in the training data is calculated. Depending on the chosen k value, the k closest points to the testing data point are selected. The class value predicted is the most frequently occurring class among those k points. The accuracy is the proportion of correctly classified points. For regression, the average value of the k closest points to the testing data is calculated and selected as the value. The accuracy is measured by the mean squared error.

For tuning the k, values 1 to 9 were examined. For each k value, 5-fold cross validation was performed, and the k-value with best performance was chosen. The accuracy of each k value is printed out for each of the 5 folds of cross validation.

For condensed KNN, the list of values(Z) starts out with the first entry of the training data. Then for each value in the training data, the Euclidean distance is calculated with all values in Z. The class of the closest point in the training data to the class of the point in Z are compared and if they are not equal, the point in the training data is added to Z. This process continues until Z does not change. After that, Z is used as the training data and normal KNN is performed.

Experimental Approach:

For each of the three datasets, the data was split into training and testing data using 5-fold cross validation. The algorithms described above were followed.

For the forest fires dataset, the months and days of week were changed to numerical variables with month from 1-12(January-December) and the days of week 1-7(Monday-Sunday).

For the computer hardware data, the variable being predicted is published relative performance rather than the estimated relative performance from the original article. The estimated relative performance is also not being used for training.

For the classification, accuracy rate is measured and for the regression, mean squared error is measured.

Results:

Dataset	**Type**	**KNN Performance**	**Condensed KNN Performance**
Ecoli	Classification	0.5969	0.5785
Segmentation	Classification	0.3857	0.3857
Machine	Regression	26885	-
Forest Fire	Regression	45073	-

KNN and condensed KNN performed best with k=1 for the segmentation data with performance 0.3857. For the ecoli dataset, the best performance for both KNN and condensed KNN was with k=1, but the performance for KNN was 0.5969 and 0.5785 for condensed KNN. For the machine data, the mean squared error was 26885 and for the forest fire data, the mean squared error was 45073.

Both algorithms performed fairly well on the ecoli data and moderately on the segmentation data, but neither seemed significantly difference in performance.

Discussion:

They both performed very similarly, but KNN performed slightly better in the ecoli data than the condensed KNN. Both the algorithms performed moderately well, as for the ecoli data they got close to 60% accuracy when there were 5 classes and for the segmentation data close to 40% accuracy where there were 7 classes. The performance also came down when more neighbors were included. The training data had moderately high number of dimensions which probably caused the performance to not be excellent. For low-dimensional data, using more than 1 neighbor might help the performance and yield higher accuracy.

The mean squared error for both the machine data and forest fire data was high. This is due to a few outlier values, even though most predictions were not too inaccurate. Some of the forest fires had very high areas and some of the machines had very high relative performance.

Summary:

KNN performed slightly better than condensed KNN on the ecoli dataset, but the difference was not significant. Both algorithms performed identically on the segmentation data. For the regression problems, mean squared error was calculated and for the classification, the accuracy was calculated. For the classification datasets, KNN and condensed KNN performed moderately. For the regression, there was high mean square error due to some predictions particularly for very high values being off. The code trained using 5-fold cross validation where each time 20% of the data was used for testing and 80% for training.

Works Cited

Guo, Gongde, et al. “KNN Model-Based Approach in Classification.” *School of Computing and Mathematics, University of Ulster*, Queen's University Belfast, citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.2.815&rep=rep1&type=pdf.

Meneses, Jamie Salvador, et al. “Compressed KNN: K-Nearest Neighbors with Data Compression.” *Entropy*, 28 Feb. 2019, www.mdpi.com/1099-4300/21/3/234/pdf.

Datasets:

https://archive.ics.uci.edu/ml/datasets/Ecoli

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

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

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