Programming Project #02 – K-Nearest Neighbor

Jacob M. Lundeen JLUNDEE1@JHU.EDU

Department of Data Science
Johns Hopkins University
Baltimore, MD 21218

Abstract

This paper describes multiple experiments with the k-nearest neighbors algorithm across six data sets (three regression and three classification). All experiments were done utilizing 5x2 cross validation and the R^2, MSE, and F_1 scores were reported for their respective data sets. Additionally, Condensed Nearest Neighbors and Edited Nearest Neighbors were used to reduce the training sets to improve the accuracy and resource requirements of the KNN algorithm. A Gaussian Kernel function is used as a weighting function with the KNN algorithm.

1. Introduction

The K-Nearest Neighbor (KNN) algorithm is a non-parametric supervised learning method. The KNN algorithm can be used for both classification and regression. For classification, the output is the class membership determined by a plurality vote of the k-nearest neighbors. For regression, the output is the mean of the k-nearest neighbors. KNN (and two reduction functions, Edited and Condensed Nearest Neighbors (ENN and CNN, respectively)) will be used in this experiment.

Problem Statement

In this experiment, we are going to see how well the KNN algorithm performs in classification and regression when compared to the Null Model from our previous experiment. A Kernel function will be used as weighting function for regression, and CNN and ENN will be to provide further comparison. Kx2 Cross Validation (CV) is used. My hypothesis is that the standard KNN function will outperform the Null Model, along with the CNN and ENN reduction methods providing further improvements in both Mean Square Error (MSE) and F_1 scores.

2. Algorithms and Experimental Methods

Data Sets

The data sets used are the same six used previously: Abalone, Breast Cancer, Car, Forest Fires, House Votes, and Machine.

The abalone data set is a regression set used to predict the age of an abalone from its physical characteristics.

The breast cancer set is a classification set used to classify a tumor as benign or malignment based on characteristics of the tumor.

The car dataset is a classification problem to determine acceptability of a car based on certain specifications.

The forest fire data set is a regression problem looking to predict the burned area of forest fires utilizing meteorological and other data.

The house votes data set is a classification problem attempting to classify congressmen as republican or democrat.

The machine data set is a regression problem to estimate performance of computer hardware.

Pre-Processing

Pre-processing of the data is handled the same as in the first programming assignment. The data is read in, missing values are handled, and categorical variables are encoded (both ordinal and nominal). The regression data sets are normalized.

Experimental Approach

As mentioned previously, KNN is a non-parametric supervised learning method. Non-parametric meaning that we make no assumptions about the shape of the underlying structure of the data. Supervised learning is the task of learning a model that maps an input to an output. KNN works by taking a test example and calculating the distance from that example to every example in the training set (for this experiment we utilized the Euclidean distance, or L_2 Norm). With those distances calculated, the 'k' Nearest Neighbors (NN) are found, and the test point is predicted (a plurality vote for classification and the mean of the 'k' NN for regression). As with the previous experiment, 5x2 CV is used.

To improve performance of the KNN algorithm, we utilize two reduction functions, CNN and ENN. ENN utilizes the principal of "Helpful Teacher" by carefully selecting examples that can more accurately represent the classes than an independent, identically distributed (iid) sample. It can improve computational performance as well as the accuracy of the KNN regressor. ENN is very similar in principle to Stepwise Backwards Selection (SBS) where examples that are misclassified or correctly classified (this up to the user) are removed from the training set.

CNN is essentially the opposite of ENN and closely resembles Stepwise Forward Selection (SFS). An empty set is created, the first example from the training set is added to the empty set, and then each remaining example in the training set is predicted off the new set. If the example is misclassified, it is added to the new set and once complete, that new set is used for the KNN regressor.

A gaussian kernel function (or Radial Basis Function (RBF)) is used as a weighting function when doing regression. Those neighbors closer to the query point, x_q, will be weighted heavier than those farther away. The kernel used is as follows:

$$K(x_q, x^t) = \exp\left[-\frac{(\|x_q - x^t\|)^2}{2\sigma^2}\right]$$

Once the NN are found, their distances are run through the Kernel, and then that value is multiplied by the target value acting as a weight to penalize examples that are farther away from the test example.

Tuning

For this experiment, three hyperparameters needed to be tuned: k, σ, and ε (σ and ε were only utilized in regression). 'k' being the number of NN, σ being the variance used in the Kernel function, and ε being a range parameter for determining if the predicted value for regression was "correct". All three hyperparameters were tuned utilizing a greedy grid search method on the 20% validation set. 'k' was tuned between 1 and 10 in steps of 1, σ between 0.001 and 5, and ε between -1.0 and 1.0, both over 20 equally spaced values.

Metrics

For regression, the primary metric to evaluate the KNN model is MSE, but R^2 and Pearson's Correlation are both reported. For classification: Precision, Recall, and F_1 scores are reported. One data set (the Cars data set) is multivariate classification, so a weighted F_1 score is reported (the F_1 for each of the four classes is calculated, weighted by the percentage of examples of that class over the total number of examples) and then the F_1 scores are averaged.

3. Results

Tables 1, 2, and 3 shows the results of the KNN classifier/regressor without reduction, CNN, and ENN reduction, respectively:

Table 1 – KNN with no Reduction

	Parameters			**Metrics**				
	k	**ε**	**σ**	**R^2**	**MSE**	**Precision**	**Recall**	**F_1**
Abalone	9	1.0	4.474	0.652	0.348			
Breast Cancer	6					0.591	0.564	0.311
Cars	7					0.777	0.808	0.229
Forest	1	0.778	0.001	0.091	0.945			
House	6					0.448	0.943	0.179
Machine	2	0.447	5.0	0.035	0.958			

Table 1 – KNN Results Using CNN

	Parameters			Reduction		Metrics				
	k	**ε**	**σ**	**Start Size**	**End Size**	**R^2**	**MSE**	**Precision**	**Recall**	**F_1**
Abalone	9	1.0	4.474	3342	1101	0.384	0.887			
Breast Cancer	6			560	7			0.188	0.75	0.161
Cars	7			1384	14			0.0	0.292	0.0
Forest	1	0.778	0.001	414	121	-0.016	3.106			
House	6			349	5			0.125	0.417	0.083
Machine	2	0.447	5.0	168	37	-0.225	3.464			

Table 2 - KNN Results Using ENN

	Parameters			Reduction		Metrics				
	k	**ε**	**σ**	**Start Size**	**End Size**	**R^2**	**MSE**	**Precision**	**Recall**	**F_1**
Abalone	9	1.0	4.474	3342	2021	0.714	0.123			
Breast Cancer	6			560	282			0.245	0.5	0.242
Cars	7			1384	1036			0.714	0.888	0.454
Forest	1	0.778	0.001	414	366	0.84	0.004			
House	6			349	329			0.468	0.961	0.236
Machine	2	0.447	5.0	168	87	-0.578	0.152			

The metrics are mean of the respective score over the 5x2 CV.

4. Discussion

No Reduction

When no reduction algorithm was used, KNN did not perform well. For the three regression sets, only the Abalone data set produced an R^2 over 0.5. The Forest and Machine data sets both had an R^2 below 0.1, which would indicate the KNN model does not provide a good fit. For the three classification data sets, the KNN's highest score was 0.311, which indicates it was not accurate in classifying the examples.

CNN Reduction

For the regression data sets, the CNN algorithm reduced each data set between 70-80% of the original data set. The is reduction in the data set did not improve the KNN's ability to predict the outcome, as in all three cases the R^2 got worse. In the case of the Forest and Machine data sets, the R^2 went negative. For the three classification problems, CNN reduced the data sets to an even

further extreme with all three being reduced by over 90%. Which such a large reduction, the F_1 scores all dropped dramatically.

ENN Reduction

The ENN reduction seemed to have a more moderate effect on reducing the data sets, even improving the KNN's accuracy in some cases. For the regression sets, the most a data set was reduced was by 49% (Machine data set, 87 out of 168 examples). For the Abalone and Forest data sets, both the KNN's R^2 and MSE improved when using ENN (the Machine data set's scores declined). For the classification data sets, the largest reduction was with the Cars data set (~25% reduction in examples). Just like with the regression data sets, the KNN's accuracy improved with two of the data sets (Cars and House) but declined with the Cancer data set.

5. Conclusion

The results show that my original hypothesis to not be true. The KNN algorithm by itself (outside of one example) did a poor job of prediction. As well, the CNN algorithm reduced the KNN's accuracy across the board. With the three classification data sets, the drop in accuracy can mostly be assigned to the large reduction in examples which simply led to not enough data to properly train the KNN algorithm.

The ENN algorithm performed better than CNN, but the KNN algorithm still performed worse on two of the six data sets even though the reduction was less severe than with CNN. This oddity requires further investigation into, as well as the CNN's extreme reduction for classification.

Acknowledgements

I would like to acknowledge the love and support of my wife, Dr. Jordan S. Lundeen, and our wonderful family: Logan, Sydney, Mavis, Thea, Hermione, Ron, and Ginny. Without their unyielding support of my endeavors, I am not sure I would have made it this far. I would also like to thank Johns Hopkins for taking a chance on me.