Predicting Diabetes using K-Nearest Neighbor and Naïve Bayes Algorithms

By Scott Onestak

Aim

Using data from the National Institute of Diabetes and Digestive and Kidney Diseases, the objective of this project was to implement both the K-Nearest Neighbor and Naïve Bayes algorithms to predict whether a patient will show signs of diabetes based on eight numeric attributes provided for each individual. In order to analyze how well these algorithms perform, 10-fold cross-validation is utilized on the data set.

The 10-fold cross-validation technique worked by splitting the dataset into 10 folds where no fold differed by more than one instance from every other fold and still contains the same proportion of instances belonging to each class as the entirety of the dataset. After broken into the 10-folds, the classifiers were built off nine of the folds and tested on the one remaining fold. This was continued until each fold had been tested by the classifiers built by the other folds. From this, the analysis of the accuracy of the classifier could be seen as the percentage of the number of instances correctly classified as true (a true positive result) plus the number of instances correctly classified as false (a true negative result) over the total number of instances in the data set.

The K-Nearest Neighbor algorithm was built by finding the K number of instances closest to the instance trying to be predicted. In order to calculate the distance, a Euclidean distance function was employed. After the closest K number of instances had been selected, whichever class was in the majority was selected as the predicted class for that instance. If there was a tie between the number of "yes" and "no" instances among the K nearest neighbors, the "yes" class was selected.

The Naïve Bayes algorithm was constructed as a probabilistic based algorithm. The algorithm operated under the assumption that each attribute was dependent from the other attributes. Additionally, it was assumed that all attributes were as equally important as all other attributes in predicting the diabetes class. As a result, the probability of the testing instance belong to the "yes" class as well as the probability of the testing instance belonging to the "no" class were both calculated to the point where the two classes were comparable. The attribute was then classified based on whichever class was more likely, and in the result of equal likelihood, the "yes" class was chosen.

This research is two-fold in helping advance diabetes diagnosis. The first is that developing better classifiers allows us to better diagnose individuals correctly with the disease of diabetes. In addition to also diagnosing better, the study also analyzes which attributes are most predictive of diabetes. By being able to distinguish the most known signs of diabetes, doctors will be able to better distinguish individuals of diabetes quicker, and both doctors and individuals will be able to understand if their patients or they themselves are at a greater risk of having diabetes, which will most likely lead to faster diagnosis.

Data

To train the classifiers, modified data from the Pima Indians Diabetes Database was used. There were 768 instances, which have all been constrained to females over the age of 21 with Pima Indian heritage. In addition, the observations with missing attributes were replaced with the averages, and the classes were changed to the nominal values of "yes" and "no" to represent whether a person does or does not have diabetes.

The classifiers were built on eight different attributes collected in the study: (1) number of times pregnant, (2) plasma glucose concentration a 2 hours in an oral glucose tolerance test, (3) diastolic blood pressure in mm Hg, (4) triceps skin fold thickness in mm, (5) 2-hour serum insulin in mu U/ml, (6) body mass index, (7) diabetes pedigree function, and (8) age. Of these eight attributes, attributes 2, 5, 6, 7, and 8 were used in the correlation-based feature selection analysis.

This correlation-based feature selection (CFS) selected a subset of the original attributes based on a heuristic of how well the attribute predicts the appropriate diagnosis class as well as how correlated the attribute was with the other attributes. The best CFS subsets are highly correlated with the class, but uncorrelated with each other. Using this subset of features, the same analysis using the K-Nearest Neighbors and Naïve Bayes algorithms could be conducted and analyzed in comparison to the use of all eight attributes.

Results and Discussion

The results of both the accuracy of algorithms using Weka and my own can be seen in the tables below.

	ZeroR	1R	1NN	5NN	NB	DT	MLP
No feature Selection	65.10%	70.83%	67.84%	74.48%	75.13%	71.88%	75.39%
CFS	65.10%	70.83%	69.01%	74.48%	76.30%	73.31%	75.78%

	My1NN	My5NN	MyNB
No feature selection	68.88%	75.52%	74.61%
CFS	68.62%	74.61%	76.30%

As can be seen, the correlation-based feature selection always performed equivalent to or better than no feature selection by Weka, but that was not the case for my classifiers. This may be due to the fact that by eliminating some attributes, the new calculations were able to flip a few classifications from "yes" to "no" or vice versa. Overall though, my nearest neighbor algorithms appeared relatively unaffected, differing by five classification changes or less.

On the other hand, the CFS improved both Weka's and my Naïve Bayes algorithms' accuracies. This was most likely because the elimination of some features reduced the dependency of some of the attributes on the others because even though the algorithm operated under the assumption that each attribute was independent from all others that was not the case in the actual data. One example of dependency may be triceps skinfold thickness and body mass index. Since both are techniques to measure body fat, there should be a correlation between the measurements. Therefore, by only using one of these measures, the Naïve Bayes calculation may have been able to reduce the double calculation of a person's body fat in the prediction of the class.

In comparing my results to Weka's results, it could be seen that the accuracies across the algorithm implementations are very close, especially among the subset of the feature selection. On the overall dataset, my K-Nearest Neighbor slightly outperformed Weka's implementation, and Weka's Naïve Bayes edged out mine. This most likely occurred based on our assigning instances to the folds. Simply having different folds may have slightly changed the calculations enough for a few instances that resulted in the accuracy differentials present between the two 10-fold cross-validation accuracies.

One final analysis I consider important to this study was the choice of a proper K for the K-Nearest Neighbor algorithm. As seen by the results, the average of five nearest neighbors outperformed the closest neighbor quite substantially. However, implementing this algorithm on my algorithm for the 10 nearest neighbors resulted in a worse accuracy than five nearest neighbors did. This is most likely because more information about the neighbors was good in the beginning because they were all huddled so close together. However, once you start expanding further out, the information becomes less relevant to the instance being analyzed and less predictive of the class the instance actually belongs to. Therefore, when choosing a K-value for the K-Nearest Neighbor algorithm, it is important to choose one that gets a good feel for the grouping around the instance, but not so far as the number of neighbors becomes so numerous that information is no longer as valuable.

Conclusion

In conclusion, both the K-Nearest Neighbor and Naïve Bayes algorithms were able to classify approximately 75% of the instances correctly, proving that even very simplistic algorithms can have extremely powerful predictive properties. In addition, it could be seen that reducing the number of features did not aid the K-Nearest Neighbor algorithm in better predicting the appropriate class; however, it did improve the accuracy of the Naïve Bayes algorithm.

Future work in this area could include a weighting system for the K-Nearest Neighbor algorithm, so that as the number of instances expands in the analysis, the weight of those instances further away decline in the importance of predicting the appropriate class. Furthermore, research on other data sets could provide a better understanding of the predictive power of these algorithms since the scope of the instances were so limited in the study.

Reflection

Throughout the implementation of this assignment, there has been much that I have learned. The first of these is planning when implementing algorithms. It is important to read the whole assignment in its entirety and understand exactly what to do before simply starting to code. Otherwise, this will only result in having to correct aspects missed later.

Additionally, I learned of the power of simple algorithms. Both algorithms, but especially the K-Nearest Neighbor algorithm, surprised me in how accurately they were able to predict the proper class with very little information about the attributes. It showed that even algorithms of a simplistic nature could prove very powerful when applied to actual data.