Charles Cui, Forest Elliott, Lucky Lai, Zach Steuer

Professor Roberto Hoyle

CS241 Final Project Report

December 15th, 2017

Machine Learning and Data Mining: *k*-NN and K-Means

In this project, we used R to write a program that clusters data based on two attributes. In doing so, we used different machine learning algorithms to sort data. Machine learning algorithms can be classified into either supervised machine learning, which infers a function from labeled data, or unsupervised machine learning, which looks for a structure within unlabeled data, and reinforcement learning, which aims to maximize long-run reward. Only the first two were studied in this project. We met weekly to work on this project, and in total spent about 50 hours. Throughout the project, we used a number of data sets from the University of California, Irvine's digital library to test and debug our code. These data sets were conveniently labeled with the number of factors, the number of data points, and whether they were best tackled with classification or clustering algorithms.

We decided to do this project because machine learning is very useful in a lot of different situations. Supervised machine learning has applications in organizing data, for example, speech recognition or ranking search results, unsupervised machine learning is more useful for detecting anomalies, such as in fraud detection.

The code which uses *k*-nearest neighbors also makes use of cross validation, which is a semi-supervised machine learning algorithm, even though *k*-NN is supervised, as are classification algorithms in general. However, another algorithm we used to sort data is

K-means, which, as a clustering algorithm, is unsupervised. Clustering algorithms, as opposed to classification algorithms, group a set of data points into clusters which are similar to each other in some way. K-means, specifically, is not sensitive to probability, but otherwise attempts to converge to a local optimum.

Machine learning started to flourish in the 1990s. The field changed its goal from achieving artificial intelligence to tackling solvable problems of a practical nature. It shifted focus away from the symbolic approaches it had inherited from AI, and toward methods and models borrowed from statistics and probability theory. It also benefited from the increasing availability of digitized information, and the possibility to distribute that via the Internet.

Before beginning this project, we had to learn about the algorithms we were going to implement. In order to do this we researched and found two textbooks, which are referenced in the Bibliography at the end. Next we had to become fluent in R, as R allows for very easy data manipulation and graphing. We used R tutorial along with trial and error to learn the language. Finally we were ready to start implementation.

The first two weeks of this project were spent studying the algorithms behind *k*-nearest neighbors in order to write our code. We also had to learn the mathematics, including statistics, behind this algorithm. We decided to implement our algorithms in R, as R allows for easy data manipulation and graphing. The *k*-nearest neighbors algorithm, given a training set of points with labels and a test point which we want to label, computes distances from the test point to all points in the training set. It then finds the nearest *k* neighbors and labels the test point based on a "majority vote" of its *k* neighbors, or which point the nearest *k* neighbors have. If a point has a tie between several labels, we picked one of the labels randomly. To test our *k*-nearest Neighbors

implementation in R, we used a dataset from the University of California, Irvine's digital library titled "Iris Data Set", which had 4 attributes (of which we could only use two), and three classes with 50 variables each. This data set was particularly good for testing our code because only one data set was linearly separable from the other two. Nonetheless, our code performed well, producing well classified graphs.

The next thing we had to implement was cross validation for our *k*-nearest neighbors algorithm. Cross validation takes a dataset and randomly partitions it into k equal sized subsets. One sample is retained as validation, the rest are used as training sets. Then *k*-NN is done again on the new validation and training set, and a misclassification rate is calculated. The process is then repeated k times, where each of the of the subsamples is used once. In our implementation, we used 10 Folds, however, it is possible to use any integer that is smaller than the number of data points.

In the next week we implemented our clustering algorithm, K-means. K-means initializes points to K clusters randomly. It then takes the average of each point in a cluster and finds the point closest to the center. After that, it relabels all points based on their closest center and repeats the process until all current labels are the same as all labels from the previous iteration. We checked the accuracy of our K-means algorithm by computing the average distance from each point to its cluster center. Our implementation of K-means had great performance, producing well clustered graphs with very low run time.

For our both our *k*-nearest neighbor and K-means algorithms, we used two more datasets with only two classes, neither of which were linearly separable. These were the "Banknote

Authentication Dataset," and the "Blood Transfusion Dataset," both with five attributes of which, again, we could only use two in our code.

In this project, we learned R, *k*-nearest neighbors algorithm, K-means algorithm, and cross validation. We learned these skills throughout this project by implementing the later two algorithms in R and testing the same three data sets on both of them. We then used cross validation to check our results. Hence, we learned a great deal about machine learning, mathematics, statistics, and programming.

Bibliography:

- Geron, Aurelien. *Hands-on Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly, 2017.
- Rogers, Simon, and Mark Girolami. *A First Course in Machine Learning*. CRC Press, Taylor & Francis Group, a Chapman & Hall Book, 2017.

Github: https://github.com/yccui/cs241_final_project