CS3345.004

Ved Nigam

Dmitrii Obideiko

Diego Ochoa

Jovanni Ochoa

Samuel Ofiaza

Searching For Similarity Narrative

Throughout the last few weeks in class, we have discussed classification and regression with kNN and decision trees. kNN stands for 'k-Nearest Neighbors' algorithm and it is a supervised machine learning algorithm, but its most significant drawback is its performance on a large dataset. kNN outputs a graph with margins for classification or regression. Decision trees is an unsupervised learning algorithm which outputs a tree/hierarchal structure to show relationships in the data.

The kNN algorithm will find k-closest points and then calculate the conditional probability of the point belonging to a group in regression or classification. What is interesting about this model is that it does not try to fit existing data into a pre-made model: the structure of the model is going to be derived from what the data looks like. To do this, all the data must be stored in memory, making this algorithm very computationally heavy. As the dataset size increases, computational power will have to increase accordingly. Basically, the algorithm will select points (centroids) to classify around, and then calculate the probabilities for the nearest points being including in the group of the centroids. Some application examples for this algorithm are credit ratings, political science, and even in computer vision.

Although this algorithm seems more closely aligned to regression, it is also very efficient for regression since all it does is predict the next point in the line. Based on the surrounding points and probabilities of the point being near a centroid, the algorithm predicts the next point in the regression line. And, since the algorithm doesn't try to fit the data into a specific model, it makes it less biased than a linear regression model, for example.

Decision Trees is one of the most popular methods for classification since the output is very easy to read. The output is displayed in a tree-like structure which is very familiar for the human eye. The model will subset data until it reaches the end, and through such recursion, it will output a tree that shows how certain data belongs to a superset or how it is a subset of a superset. This will help us identify many important relationships in data if the model is accurate. We use the Gini index to evaluate whether a model is accurate or not. The lower the Gini index, the more we should trust the model.

The values for regression using a decision tree are determined through the numerical value assigned to each split of data. As the data is recursively sub-setted, each subset is assigned a numerical value through which we can come up with a regression. Decision Trees are not the most efficient for regression since a slight change between datapoints can cause a whole new branch/node to be made. It is a very sensitive algorithm that could get extremely elaborate for a relatively small dataset.

We also discussed two clustering methods in class: kMeans and hierarchal clustering. kMeans clustering is very similar to the beginning of kNN; the algorithm will select points and cluster data based a point's probability of belonging to a selected point. If the dataset is well balanced and suitable for the task, kMeans will output clusters within which the data have similarities. Hierarchal clustering will use a similar approach to what Decision Trees do, it will subset data recursively based on characteristics and come up with a tree-like structure. This is helpful in seeing how many potential groups the data can be clustered into and how specific we want the clusters to be.

Additionally, we discussed PCA (Principal Components Analysis) and LDA (Linear Discriminant Analysis). PCA is a technique that can help you reduce the dimensions of your data based on the principal component of your data. If there are 10+ variables in a dataset, but PCA believes that those variables only describe 2 main features, it will reduce the dimensions of the data and make 10 variables into 2. It removes features that the method believes are not important to the analysis. This can make the data easier to interpret computationally lighter. LDA considers class and looks for linear combinations of features. Instead of removing features, this method considers all of the features to see how they relate to each other; at the end it comes up with feature(s) that are combinations of existing features in the dataset. These techniques can be very effective for machine learning since they can make the computation easier and faster by decreasing the dimensions of the data.