CS – 4390 Machine Learning

Searching For Similarity

Subash Chandra, Derrick Martin, Abigail Solomon, and Aditi Chaudhari

SXC200027 DRM180001 TSM190000 APC180001

A) kNN is a fairly simple algorithm which, upon receiving a data point, will identify the 'k' Nearest Neighbors to this data point. It then polls those neighbors, and chooses which class that data point belongs to. A good number for K is the square root of the number of Data Points that you have. K should usually be odd. Because of the simplicity of the algorithm, KNN is easily misled by noisy data. For regression, knn will poll the k Nearest Neighbor for a predicted target value. A K value of 1 will be highly granular, but will probably be overfitted. Larger values of K can perform better than linear regression, but because of kNN's strong assumptions, it has a high bias.

Decision Trees are Binary trees that recursively split the data along a parameter until all the data is split. At the root node, the entire population is represented, and the Nodes are given a series of binary, True/False questions. All data that is 'True' is split down one path, and all data that is ' false' down the other. A node that does not split is called a Terminal Node, and the goal of Decision Trees is to have only Terminal Nodes as leaves. To do this effectively, the Decision tree needs to be able to calculate 2 things.

1) How 'splittable' or 'impure' a Leaf node is, through the Gini Impurity score

2) How much information is lost or gained by a specific T/F question.

After finding the Information Gain score of the T/F questions, a question can be chosen which maximizes the splitting of the data. A decision tree with too many splits will tend to overfit, so introducing a maximum number of splits, or using multiple decision trees which find the answer through consensus will provide more accurate, stable results.

B) K Means Clustering is an algorithm that splits the given data into K clusters. It starts by selecting random data points as the 'core' of 3 hypothetical clusters, and then slowly filters all the remaining data points into these clusters by distance. Then the means of these clusters are used as new 'cores' of the clusters, and each points is reevaluated to these new clusters. If there are no changes, then the K Means Clustering algorithm is complete.

Heirarchical clustering works by recursively finding the 2 closest points, and putting them into clusters. If the closest point to a new data point is already in a cluster, this new point is added to that cluster. The order that these clusters are made in can be shown as a dendrogram/Heirarchy chart where the shortest branches are the oldest.

Both these models are heuristics, and K Means clustering is even worse because it is completely randomized. A good alternative is Model Based Clustering.

Model – Based Clustering works on the principle that the data was created  from a finite combination of component models of probability distributions. Each of these component models is a parametric, distribution of any kind, i.e. Gaussian, Pareto, Bernoulli, etc. 2 Observations that come from the same component model are determined to be in the same cluster.

C) Both PCA (Principal component analysis) and LDA (Linear discriminant analysis) are dimension reduction algorithms, they reduce the number of random variables , design a new feature  from the existing data. PCA is a data reduction technique that can help you reduce the dimensions of your dataset that captures as much of the variation in the data as possible. PCA transforms the data into a new coordinate space by combining linearly correlated variables into the set of new variables that are called principal component variables, it retains the most valuable parts of all the variables but drops the "least important" variables. This transformation fits the data into a coordinate system where the most significant variance is found on the first coordinate, and each subsequent coordinate is orthogonal to the last and has a lesser variance than the previous. PCA is applied to data without regard to class since it's unsupervised learning. Similarity LDA reduces the number of features, it also compressed the dataset, it is a method of finding features that maximize the variability between 2 or more classes of objects. The significant difference between PCA and LDA is that LDA is supervised learning that it considers the class.

Both are useful techniques to machine learning because they make data visualization easy when it comes to higher dimension datasets. Datasets with big number of variables are hard to analyze and may also be unnecessary variables (noise features) mixed in that could hinder your analyses. If we simply delete some data, we could lose the accuracy of our model, here PCA and LDA technically reduce the number of dimensions, or even extract more useful features from the ones we already have, that could greatly improve our analyses and model building. Another issue with higher dimension is that as the number of features grow, the space greatly increases, but using  PCA and LDA, the more features they reduce, the larger space they save.