Searching for Similarity

By: Ryan Gagliardi, Andrew Gerungan, Ethan Huynh, Meinhard Capucao

Regression

kNN and decision trees are both completely valid algorithms to use when looking to analyze a data set with regression. kNN uses an algorithm to calculate the average between the K closest points in the training set, and attribute that to the new point in the test set. In doing this, the new point is accurately estimated using the data it was trained on that it is nearest to when plotted, and results in accurate predictions based on the data it is given. Decision trees use the standard deviation to estimate the number of branches needed for the graph, and the Coefficient of Deviation to decide there are enough branches. By doing this, the data can be plotted on a graph that partitions it based on similarities in the test data and each branch of the tree makes the data more and more specific until it can reach a conclusion on what the result of the data should be. These predictions that the two algorithms create can be used to determine an accurate estimate of the target based on the data it is trained on. The accuracy of these algorithms mean that they excel in estimating exact numbers for answers, and can do so very efficiently.

Classification

kNN and decision trees work very similarly in their process for classification with a few differences. In the case of kNN, classification predicts a class based off of local probability as opposed to regression that predicts a value based off of local average. Decision trees are different for classification in that it replaces RSS with the count of classes in various regions. Accuracy isn't quite as "accurate" in splitting regions so it also uses entropy or the Gini index instead. Besides these differences, kNN and decision trees function in the same way as in Regression.

PCA & LDA

PCA works by first normalizing the initial variables, then computing a covariance predictor x predictor matrix. Then it computes the eigenvectors and eigenvalues of the covariance matrix to identify principal components, creates a feature vector to choose which principal components to keep, then finally recasts the data along the principal component axes.

LDA works by first computing the mean vectors for the different classes, then commuting the in-between and within-class scatter matrices. After that, it computes the eigenvectors and eigenvalues and sorts the eigenvectors by decreasing eigenvalues and chooses k eigenvectors with the largest eigenvalues. Finally, it recasts the data into the new space.

These techniques are useful as PCA can increase interpretability while minimizing information loss, as it finds the most significant/strongest features; whereas LDA can help to avoid the curse of dimensionality and lower resource costs through simplifying the model.

Classification

K-means clustering works by identifying k-centers (centroids) then grouping other observations based on nearness to the closest centroid. This is an iterative approach. The most important part is random assignment, where one approach is choosing k observations to be the means. Then each observation is assigned to its closest centroids, the centroids are recalculated, then the assignment is repeated until clustering and convergence is complete, or based on the number of starts specified.

Hlerarchical clustering works by creating a dendogram of the clustering. Each observation is placed in its own cluster, and the distance between each cluster and every other cluster is calculated. Then, the two closest clusters are specified. These are repeated until all clusters are linked onto one cluster. There are levels that represent the amount of clusters, which decrease going down until each item has their own individual cluster. The dendogram can be cut to see what visually makes sense for the amount of clusters for the dendogram. Hierarchical clustering does not work well with a lot of data.

Model-based clustering relies on a formal model, and uses statistics to find the most optimal number of clusters. This assumes a data model to find the most likely model component and number of clusters.