Similarity Document

The kNN algorithm will take in the training data and store each data point in memory for future reference. When predicting on a new data point in the test data set, the kNN algorithm will find the average value of the target column of the k closest neighbors for that data point by referencing the training data in memory [1]. Classification for the kNN algorithm works similarly to regression in that they both use stored training data and find the k closest data points to the new input. However, unlike where the algorithm finds the average of the target value of the neighbors in classification the kNN algorithm calculates the conditional probability for each class over the number of neighbors that have the class to find the most likely class for the new input [1]. Decision trees are different from kNN in that they are a greedy algorithm that makes the optimal choice at any given step [1]. For regression and classification, a decision tree will recursively split the data along the optimal linear boundaries until a threshold is reached. The difference between regression and classification is found in the condition by which the algorithm partitions the data at each recursive step. In regression the condition for where the linear boundary in the data is based off which boundary will produce the minimum RSS within the regions split by the boundary [1]. In classification the decision tree algorithm instead looks at the number of each class in the regions being split [1]. When find the value of a new test data point the algorithm will go down the branches of the tree looking at the data points predictor values when coming to a split until it reaches the end of a branch, and the value is found [1].

K Means is an unsupervised algorithm that works by find k different centers and assigns observations to different groups based on which of those k centers they are closest to, whether through Euclidian distance or some other form of measurement [1]. To find the centers of the data set first the algorithm assigns centers randomly by picking data points in the set and assigns each data point around those centers. Next the algorithm will iteratively change the centers by moving them to the center of the clusters they are in, until finally none of the centers move anymore and the final clusters are found [1]. The hierarchical clustering technique works by giving each data point in the set its own cluster and calculating the distance between all the data point clusters until it finds the two clusters with the closest distance and combines them together in a hierarchical fashion [1]. When one cluster remains, the algorithm is complete, and a hierarchical tree is created showing the way in which data clusters have been combined [1]. Model based clustering makes use of a model when clustering the data points of the set. The method can have different data models used when clustering the data and will often make use of statistical analysis when deciding, how many clusters there should be and which cluster a given data point should be put into [2].

PCA or principal components analysis reduces the dimensions of a given data set by finding principal components in the data that cover the majority of variance. The number of principal components used is less than the number of predictors in the data set thus reducing the size of the set for a regression or classification model [1]. LDA or linear discriminate analysis is another data reduction technique and it works by looking at the classes of the data set and find the most optimal combination of predictors when modeling to separate those two classes [1]. PCA and LDA are useful for machine learning because they captures the needed columns/components of the data set and get rid of the rest, thus a data set with hundreds of predictors could potential be reduced down to a few with minimal loss in accuracy [1].

References

- Mazidi, Karen. *Machine Learning Handbook Using R and Python.* 2nd edition. 2020 [1]
- Banerjee, A., Shan, H. (2011). Model-Based Clustering. In: Sammut, C., Webb, G.I. (eds) *Encyclopedia of Machine Learning.* Springer, Boston, MA. https://doi.org/10.1007/978-0-387-30164-8_554 [2]