G SuryaDev Reddy
14CO214
8123997429
gouru.surya@gmail.com

An Efficient k-Means Clustering Algorithm: Analysis and Implementation

By Tapas Kanungo, David, Nathan, Christine and Angela

Work Finalized:

- Implementation of k-means Llyod's *Filtering* algorithm using BBD-tree data structure.
- Augmenting algorithm with a simple randomized seeding technique for initial selection of k centers.

Portion of Work Completed:

- Building a BBD Tree from the given set of data
- Pruning Function is completed which prunes the centers which are not close to any point lying within the associated cell compared with the center closest to the mean of the cell

Implementation Details:

- All data are placed into a tree where we choose child nodes by partitioning the data along a plane parallel to the axis of longest radius of bounding box.
- We maintain for each node, the bounding box of the data stored at that node.
- To do a k-means iteration, we need to assign data to clusters and calculate the sum and the number of data assigned to each cluster. This can be done using the below logic:
  - *Pruning:* For each node in the tree, we can rule out some cluster centroids as being too far away from every single point in that bounding box.
  - If there are more than one center in candidate centers set, then apply filtering recursively.
  - Once only one center is left in the node, all data in the node can be assigned to that cluster in batch.
- For each of the k-centers, we need to compute the centroid of the set of data points for which this center is closest. We then move this center to the computed centroid and proceed to the next stage.

- **Language:** JAVA
- **Tools:** ECLIPSE

Portion of Work Remaining:

- Initializing of k centroids picked from the data points using careful seeding technique and partitioning the observations into distinct non overlapping groups.
- Clustering is to be completed which is, given k cluster centroids, data points has to be assigned to nearest centroid from the set of candidate centers. In other words, for each iteration, function to assign each data point to the appropriate cluster.
- Implementation of standard k-means algorithm using all the above modules
- Try to learn the value of k , using statistical test as mentioned below

Extra Information:

- With the k-means++ initialization of the cluster centers, the algorithm is guaranteed to find a solution that is O(log k) competitive to the optimal k-means solution.
- This algorithm can still be extended to automatically determine k, the number of clusters based on BIC (Bayesian Information Criterion) scores
- There is another algorithm to learn the value of k which outperforms the above BIC method, is based on a statistical test for the hypothesis that a subset of data follows a Gaussian distribution.

References for Additional Work:

1) D. Arthur and S. Vassilvitskii. "K-means++: the advantages of careful seeding". ACM-SIAM symposium on Discrete algorithms, 1027-1035, 2007
2) G. Hamerly and C. Elkan. Learning the k in k-means. NIPS, 2003
3) Dan Pelleg and Andrew Moore. X-means: Extending K-means with Efficient Estimation of the Number of Clusters. ICML, 2000