Large Scale Clustering Algorithm Design

DS 5460 Big Data Scaling - Final Project

Team Members: Hongyu Dai, Jingyuan Wu, Zihan Fang

Overview

Problem

- The problem we are trying to solve is to cluster a dataset using cluster algorithms and identify patterns and similarities within the dataset.

Method

- To accomplish that, we will use KMeans and BFR algorithms to cluster the data points, evaluate the results using the Adjusted Rand Score metric and compare two cluster solutions based on peak memory

Data

Overview

This dataset appears to represent a genomic sequence. The first column 'contigname' is likely the name of the sequence or contig, and the subsequent columns (feat_0 to feat_31) are various features associated with that sequence.

Shape

- Data.csv: 1512372 rows × 33 columns
- Label.csv: 1074858 rows × 2 columns

VANDERBILT UNIVERSITY

Method - Kmeans

- Unsupervised machine learning algorithm
- Cluster similar data points together by minimizing the sum of squared distances between each point and its centroid.

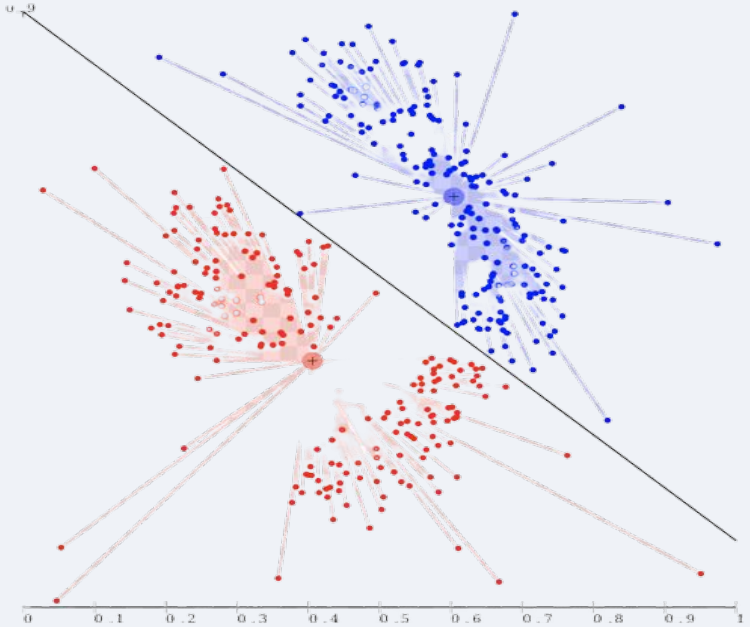


Figure Source: https://commons.wikimedia.org/wiki/File:KMeans-density-data.svg

VANDERBILT UNIVERSITY

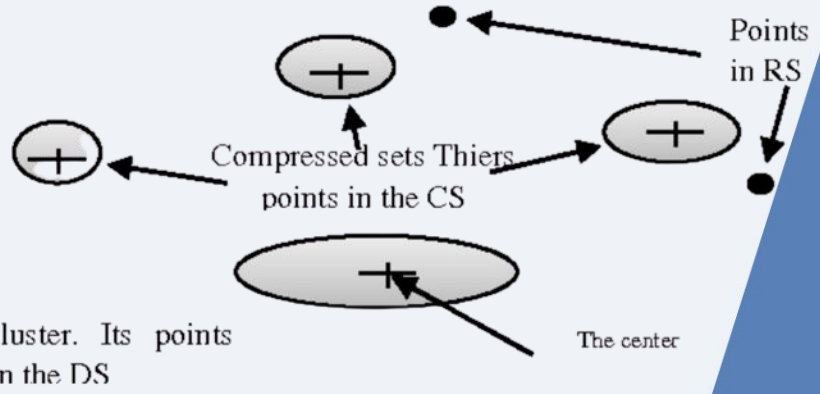
Milestone 1

- Left join the Data.csv and Label.csv
- Separate the target variable Y and the features X using numpy arrays.
- Create a KMeans object with n_clusters=745
- Fit the KMeans model to the feature matrix X
- Obtain the predicted label for each data point
- Remove data points who don't have label
- Compute the ARI score using the ground truth labels Y and the predicted value
- ARI: 0.26.

VANDERBILT UNIVERSITY

Method - BFR

- BFR is a variant of KMeans designed to handle very large (disk-resident) data sets.
- It assumes that clusters are normally distributed around a centroid in a Euclidean space.
  - Mean and Standard deviations in different dimensions may vary



- Goal: to find cluster centroids; point assignment can be done in a second pass through the data.

- Idea: Rather than keeping points, BFR keeps summary statistics of groups of points
  - Points are read one main-memory-full at a time.
  - Most points from previous memory loads are summarized by simple statistics.

VANDERBILT UNIVERSITY

Method - BFR

- Overview of the algorithm:

  1. Initialize K clusters/centroids
  2. Load in a bag points from disk
  3. Assign new points to one of the K original clusters, if they are within some distance threshold of the cluster
  4. Cluster the remaining points, and create new clusters
  5. Try to merge new clusters from step 4 with any of the existing clusters
  6. Repeat steps 2-5 until all points are examined

Milestone 2

1. Reads in a distributed data file using Spark.
2. Processes the data by splitting it into smaller partitions for clustering.
3. Uses the KMeans algorithm to cluster data points into two sets: Dense Set (DS) and Remaining Set (RS).
4. In subsequent rounds, checks if data points belong to DS or RS and updates cluster statistics accordingly.
5. Compressed Set (CS) merges small or close clusters and is repeated until no more merges are possible.
6. Merges final CS and DS and any clusters that are close enough.
7. Finally, the resulting clustering assignments for each data point are outputted.

VANDERBILT UNIVERSITY

Milestone 3 - Result

	ARI	Peak Memory
KMeans	0.26	10517988K
BFR	0.04	9808216K

Conclusion

- KMeans
  - Strengths
    - Simple and easy to implement
    - Fast and efficient for small datasets with low dimensions
    - Works well with spherical clusters
    - Can handle large datasets with parallelization
  - Weaknesses
    - Requires the number of clusters to be specified beforehand
    - Sensitive to initial cluster centroids
    - May converge to local optima, resulting in suboptimal solutions
    - Not suitable for non-spherical clusters or clusters with varying sizes and densities

Conclusion

- BFR
  - Strengths
    - Can handle large datasets efficiently and is capable of processing data that cannot be accommodated in memory
    - Can handle non-spherical clusters and clusters with varying sizes and densities
  - Weaknesses
    - Can be sensitive to the choice of the split point
    - Requires more implementation effort compared to KMeans
    - May be slower than KMeans, as it needs multiple iterations and can create many intermediate clusters
    - May not perform well on datasets with overlapping clusters

Challenge

- Improve the performance of BFR algorithm
  - Chunk size
  - Number of clusters in the first/intermediate round

VANDERBILT UNIVERSITY

THANKS!