Distributed Clustering Algorithm for Spatial Data Mining

Malika Bendechache, M-Tahar Kechadi
School of Computer Science & Informatics, University College Dublin
Belfield, Dublin 04, Ireland
Outline

- Introduction & Motivation
- Proposed Approach
- Experimental Results
  - Compared Algorithms
  - Evaluations
- Conclusion
Context

- Across a wide variety of fields, datasets are being collected and accumulated at a dramatic pace and massive amounts of data that are being gathered are stored in different sites.
- In this context, data mining (DM) techniques have become necessary for extracting useful knowledge from the rapidly growing large and multi-dimensional datasets.
Spatial Data

- **Size and Importance**
  - About 80% of the data we collect is either spatial or contains spatial dimensions.
  - The size is usually enormous. It needs special handling and management

- **Complex, heterogeneous, & distributed**
  - Spatial dimensions has significant complexity to the analysis
  - High-dimensionality

- **Fast Processing Requirements**
  - Requires fast algorithms for knowledge extraction
  - Requires fast algorithms for visualisation
Distributed Data Mining

- Large-scale distributed platforms
- Datasets geographically distributed and owned by different organisations.

- **Existing techniques:**
  - Distributed association rules
  - Distributed classification.
  - Few research conducted on distributed clustering models: tow main steps:
    1) Perform partial analysis on local data at individual sites
    2) Exchange local results to generate global models by aggregating the local results.

- **Drawbacks**
  - Do not scale well
  - The generation of good global models (e.g., local models do not contain enough information for the merging process)
Distributed DM Techniques

Data Partitioning
- Data Partition 1
- Data Partition 2
- Data Partition N-1
- Data Partition N

Clustering
- K-Means

Reduction
- Reduction
- Reduction
- Reduction

Merging
- Leader 1
- Leader M

Cluster results
Distributed Clustering

**Step 1: Local Clustering**
- LC Algorithm running on each with different number of clusters $k_i$; $k_i$ should be big enough to contain all local clusters
- Our tests are done with the K-Means algorithm
- **Identify the important representatives of each cluster**

**Step 2: Global Clustering**
- Exchange only the representative data of each cluster.
- Execute the process of reduction in each local node.
Data Reduction Techniques

• Reducing data size
  • Reducing data regardless of the knowledge behind them

• Density-based clustering algorithms
  • Choose representative points for each cluster: medoid points, core points, or even specific core points

  • Difficulties in terms of quality and size of representative set
Proposed Approach

- **Objectives**
  - Develop distributed clustering to deal with distributed datasets
  - Exploit the processing power of distributed infrastructure, (grids, clusters, and clouds)
  - The algorithm should be efficient in terms of precision of its results and its computational complexity
  - Exploit (if possible) existing popular algorithms
- **Characteristics**
  - Dynamic number of clusters
  - Efficient data reduction phase
  - Full integration as a service for the current and new technologies such as Grids and Clouds
  - Support visualisation of very large datasets
Parallel Processing

Communication Overheads

Local Result 1
Local Result 2
Local Result 3
Local Result 4
Local Result 5

Aggregation

Global Result
Distributed Clustering

Distributed datasets

Clustered datasets

K-Means Reduction

Aggregation

K-Means Reduction

K-Means Reduction

K-Means Reduction
Proposed Approach (2)

- **Distributed Data Reduction Technique:**
  - Based on shape and the density of local clusters.
  - The shape of a cluster is represented by its boundary points (called contour).

- **Algorithms for extracting the contour from clusters**
  - Triangulation to generate the boundary of the cluster
  - It should allow to construct non-convex boundaries
  - It should have a reasonable complexity: $O(n \log n)$
Proposed Approach (3)

- **Merging**
  - Clusters are represented by their contours.
  - A leader is elected for each group of neighbourhood to collect the contours of the clusters in the same group.
  - The leader merges these contours based on the overlay technique.
  - The process of merging continues until there is no overlapping clusters (well-separated clusters)
Experimental Results

- **Example of execution:**
  - Suppose that the system contains $N = 5$ Nodes
  - Each node executes in parallel the K-Means algorithm with different $K_i$. 
Some Results…

Node1: K=30
Node2: K=60
Node3: K=90
Node4: K=120
Node5: K=150

Results after merging
Border of clusters generated
Comparative Study

- **BIRCH:**
  - BIRCH was proposed by Tian in 1996
  - The implementation performs a pre-clustering and then uses a centroid-based hierarchical clustering algorithm

- **CURE:**
  - CURE was proposed by Sudipto in 2001.
  - The algorithm uses representative points with shrinking towards the mean point
# Experimental Results

## Datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Number of Points in Dataset</th>
<th>Shape of Clusters</th>
<th>Number of Clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset1</td>
<td>14000</td>
<td>Big Oval (Egg Shape)</td>
<td>5</td>
</tr>
<tr>
<td>Dataset2</td>
<td>30350</td>
<td>2 small Circles, 1 big Circle and 2 Ovals linked</td>
<td>4</td>
</tr>
<tr>
<td>Dataset3</td>
<td>17080</td>
<td>4 Circles, 2 Circles linked</td>
<td>5</td>
</tr>
</tbody>
</table>
Experimental Results
Experimental Results (2)
Experimental Results (3)
Observations

- When two clusters are merged, the new cluster is represented by union of the two contours of the two original clusters.

- This speeds up the execution times without impacting the quality of clusters generated.
Algorithm Complexity
Algorithm’s Scalability
Conclusion

- Innovative Distributed Dynamic Clustering Algorithm (D²CA)
- It exploits the processing power of the distributed platform by maximising the parallelism and minimising the communications and mainly the size of the data that is exchanged between the system nodes.

- By its design it can use any local clustering algorithms
- **We need to do more tests with various algorithms**
- **Explore possibilities of extending the techniques to other types of large and distributed datasets.**
Conclusions (2)

- Develop distributed clustering to deal with distributed datasets
- Exploit the processing power of distributed infrastructure, (grids, clusters, and clouds)
- The algorithm should be efficient in terms of precision of its results and its computational complexity
- Exploit (if possible) existing popular algorithms
- Dynamic number of clusters
- Efficient data reduction phase
- Full integration as a service in the current and new technologies such as Grids and Clouds
- Support visualisation of very large datasets
Conclusion

- New approach for distributed clustering on spatial datasets.
- Local models are generated by executing K-Means algorithm in each node, and then the local results are merged to build the global clusters.
- The local models are extracted from the local datasets so that their sizes are small enough to be exchanged through the network.
- Preliminary results of this algorithm showed:
  
  - The effectiveness of proposed approach either on quantity of clusters generated or the execution time comparing to BIRCH and CURE algorithms.
  - The method is different from current distributed clustering models presented so far as it regenerates the global result from mining local dataset using K-means algorithm with different K for each node in the system.
  - The Algorithm scales well for large databases without sacrificing clustering quality.

- **PERSPECTIVE**
  - Execute the Algorithm with different large real world datasets in order to prove its robustness.
Thank you for your attention