Data Mining

4. Cluster Analysis

4.4 Hierarchical Methods

Spring 2010

Instructor: Dr. Masoud Yaghini
Outline

- Introduction
- BIRCH Algorithm
- References
Introduction
Introduction

- A **hierarchical clustering** method works by grouping data objects into a tree of clusters.

- **Types of hierarchical clustering methods:**
  - **Agglomerative:** the hierarchical decomposition is formed in a bottom-up (merging) fashion.
  - **Divisive:** the hierarchical decomposition is formed in a top-down (splitting) fashion.
Introduction

- **Agglomerative hierarchical clustering**
  - This bottom-up strategy starts by placing each object in its own cluster and then merges these atomic clusters into larger and larger clusters, until all of the objects are in a single cluster or until certain termination conditions are satisfied.
  - Most hierarchical clustering methods belong to this category.
  - They differ only in their definition of intercluster similarity.
Introduction

- **Divisive hierarchical clustering**
  - This top-down strategy starts with all objects in one cluster.
  - It subdivides the cluster into smaller and smaller pieces, until each object forms a cluster on its own or until it satisfies certain termination conditions,
  - Termination conditions can be
    - a desired number of clusters is obtained or
    - the diameter of each cluster is within a certain threshold.
Example

- Example: Agglomerative versus divisive hierarchical clustering
  - the application of AGNES (AGglomerative NESting), an agglomerative hierarchical clustering method,
  - and DIANA (DIlusive ANAlysis), a divisive hierarchical clustering method, to a data set of five objects, \{a, b, c, d, e\}.
Example

- Agglomerative and divisive hierarchical clustering on data objects \( \{a, b, c, d, e\} \).
Dendrogram

- **Dendrogram**
  - A tree structure which is commonly used to represent the process of hierarchical clustering.
  - It shows how objects are grouped together step by step.
Dendrogram representation for hierarchical clustering of data objects \{a, b, c, d, e\}.
Measures for Distance Between Clusters

- Common measures for distance between clusters are as follows:
  - Minimum distance
  - Maximum distance
  - Mean distance
  - Average distance
Measures for Distance Between Clusters

**Notation**

- $|p - p'|$: is the distance between two objects or points, $p$ and $p'$
- $m_i$ is the mean for cluster, $C_i$
- $n_i$ is the number of objects in $C_i$
- $m_j$ is the mean for cluster, $C_j$
Measures for Distance Between Clusters

- **Minimum distance**

\[ d_{\text{min}}(C_i, C_j) = \min_{p \in C_i, p' \in C_j} |p - p'| \]

- When an algorithm uses the **minimum distance**, it is sometimes called a nearest-neighbor clustering algorithm.

- If the clustering process is terminated when the distance between nearest clusters exceeds an **arbitrary threshold**, it is called a single-linkage algorithm.
Measures for Distance Between Clusters

- **Maximum distance**

\[ d_{\text{max}}(C_i, C_j) = \max_{p \in C_i, p' \in C_j} |p - p'| \]

- When an algorithm uses the maximum distance, it is sometimes called a **farthest-neighbor clustering algorithm**.

- If the clustering process is terminated when the maximum distance between nearest clusters exceeds an arbitrary threshold, it is called a **complete-linkage algorithm**.

- Farthest-neighbor algorithms tend to minimize the increase in diameter of the clusters at each iteration as little as possible.
Measures for Distance Between Clusters

- **Mean distance**

\[ d_{\text{mean}}(C_i, C_j) = |m_i - m_j| \]

- The minimum and maximum measures tend to be overly sensitive to outliers or noisy data.
- The use of **mean or average distance** is a compromise between the minimum and maximum distances and overcomes the outlier sensitivity problem.
Measures for Distance Between Clusters

- **Average distance**

\[
d_{\text{avg}}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{p \in C_i} \sum_{p' \in C_j} |p - p'|\]

- Whereas the mean distance is the simplest to compute, the average distance is advantageous in that it can handle categoric as well as numeric data.
- The computation of the mean vector for categoric data can be difficult or impossible to define.
The Difficulties with Hierarchical Clustering

- The quality of a pure hierarchical clustering method suffers from its inability to perform adjustment once a merge or split decision has been executed.
- That is, if a particular merge or split decision later turns out to have been a poor choice, the method cannot backtrack and correct it.
- Recent studies have emphasized the integration of hierarchical agglomeration with iterative relocation methods.
The Difficulties with Hierarchical Clustering

Three such methods are introduced in this chapter, including:

- **BIRCH**,
  - begins by partitioning objects hierarchically using tree structures, where the leaf or low-level nonleaf nodes can be viewed as “microclusters” depending on the scale of resolution.
  - It then applies other clustering algorithms to perform macroclustering on the microclusters.

- **ROCK**
  - Merges clusters based on their interconnectivity.

- **Chameleon**,  
  - Explores dynamic modeling in hierarchical clustering.
BIRCH Algorithm
BIRCH Algorithm

- **BIRCH: Balanced Iterative Reducing and Clustering Using Hierarchies**
  - BIRCH is designed for clustering a large amount of numerical data
  - It integrates the hierarchical clustering (at the initial microclustering stage) and other clustering methods such as iterative partitioning (at the later macroclustering stage).
  - It overcomes the two difficulties of agglomerative clustering methods:
    - (1) scalability and
    - (2) the inability to undo what was done in the previous step.
BIRCH Algorithm

- BIRCH introduces two concepts:
  - Clustering Feature (CF)
  - Clustering feature tree (CF tree)
- They are used to summarize cluster representations.
- These structures help the clustering method achieve good speed and scalability in large databases and also make it effective for incremental and dynamic clustering of incoming objects.
BIRCH Algorithm

- Given \( n \) \( d \)-dimensional data objects or points in a cluster, we can define the centroid \( x_0 \), radius \( R \), and diameter \( D \) of the cluster as follows:

\[
\begin{align*}
    x_0 &= \frac{1}{n} \sum_{i=1}^{n} x_i \\
    R &= \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - x_0)^2} \\
    D &= \sqrt{\frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j=1}^{n} (x_i - x_j)^2}
\end{align*}
\]

- Where \( R \) is the average distance from member objects to the centroid, and \( D \) is the average pairwise distance within a cluster.

- Both \( R \) and \( D \) reflect the tightness of the cluster around the centroid.
BIRCH Algorithm

- **Clustering Feature (CF)**
  - CF is a three-dimensional vector summarizing information about clusters of objects.
  - Given $n$ $d$-dimensional objects or points in a cluster, $\{x_i\}$, then the CF of the cluster is defined as:
    
    $$ \text{CF} = \langle n, LS, SS \rangle $$
    
    - where $n$ is the number of points in the cluster,
    - $LS$ is the linear sum of the $n$ points, i.e.,
      
      $$ \sum_{i=1}^{n} x_i $$
    - $SS$ is the square sum of the data points, i.e.,
      
      $$ \sum_{i=1}^{n} x_i^2 $$
Clustering features are additive.

For example, suppose that we have two disjoint clusters, $C_1$ and $C_2$, having the clustering features, $CF_1$ and $CF_2$, respectively.

The clustering feature for the cluster that is formed by merging $C_1$ and $C_2$ is simply $CF_1 + CF_2$.

Clustering features are sufficient for calculating all of the measurements that are needed for making clustering decisions in BIRCH.
Example: Clustering feature.

- Suppose that there are three points, (2, 5), (3, 2), and (4, 3), in a cluster, $C_1$. The clustering feature of $C_1$ is:

$$CF_1 = \langle 3, (2 + 3 + 4, 5 + 2 + 3), (2^2 + 3^2 + 4^2, 5^2 + 2^2 + 3^2) \rangle$$

$$= \langle 3, (9, 10), (29, 38) \rangle.$$

- Suppose that $C_1$ is joint to a second cluster, $C_2$, where $CF_2 = \langle 3, (35, 36), (417, 440) \rangle$.

- The clustering feature of a new cluster, $C_3$, that is formed by merging $C_1$ and $C_2$, is derived by adding $CF_1$ and $CF_2$. That is:

$$CF_3 = \langle 3 + 3, (9 + 35, 10 + 36), (29 + 417, 38 + 440) \rangle$$

$$= \langle 6, (44, 46), (446, 478) \rangle.$$
BIRCH Algorithm

- A **CF tree** is a height-balanced tree that stores the clustering features for a hierarchical clustering.

```
  CF_1    CF_2    -----    CF_k
   |       |       |        |
  CF_{11}  CF_{12}  -----  CF_{1k}
   |       |       |        |
```

Root level

First level
BIRCH Algorithm

- By definition, a nonleaf node in a tree has children.
- The nonleaf nodes store sums of the CFs of their children, and thus summarize clustering information about their children.
- A CF tree has two parameters:
  - **Branching factor, B**
    - specifies the maximum number of children per nonleaf node.
  - **Threshold, T**
    - specifies the maximum diameter of subclusters stored at the leaf nodes of the tree.
- These two parameters influence the size of the resulting tree.
BIRCH Algorithm Phases

The primary phases of BIRCH are:

**Phase 1:**
- BIRCH scans the database to build an initial in-memory CF tree

**Phase 2:**
- BIRCH applies a (selected) clustering algorithm to cluster the leaf nodes of the CF tree, which removes sparse clusters as outliers and groups dense clusters into larger ones.
BIRCH Algorithm Phases

- **Phase 1:**
  - the CF tree is built dynamically as objects are inserted.
  - Thus, the method is **incremental**.
  - An object is inserted into the closest leaf entry (subcluster).
  - If the diameter of the subcluster stored in the leaf node after insertion is larger than the threshold value, then the leaf node and possibly other nodes are split.
  - After the insertion of the new object, information about it is passed toward the root of the tree.
  - The size of the CF tree can be changed by modifying the threshold.
BIRCH Algorithm Phases

- Phase 2:
  - Once the CF tree is built, any clustering algorithm, such as a typical partitioning algorithm, can be used with the CF tree in Phase 2.
Computation Complexity of the Algorithm

- The computation complexity of the algorithm is $O(n)$, where $n$ is the number of objects to be clustered.
- Experiments have shown the linear scalability of the algorithm with respect to the number of objects and good quality of clustering of the data.

Hierarchical Methods
Weakness of BIRCH

- However, since each node in a CF tree can hold only a limited number of entries due to its size, a CF tree node does not always correspond to what a user may consider a natural cluster.

- Moreover, if the clusters are not spherical in shape, BIRCH does not perform well, because it uses the notion of radius or diameter to control the boundary of a cluster.
References

- J. Han, M. Kamber, *Data Mining: Concepts and Techniques*, Elsevier Inc. (2006). (Chapter 7)
The end