Clustering: Techniques & Applications

Nguyen Sinh Hoa, Nguyen Hung Son
Agenda

- Introduction
- Clustering Methods
- Applications:
  - Outlier Analysis
  - Gene clustering
- Summary and Conclusions
Clustering vs. Classification

**Clustering:**
Unsupervised learning:
Finds “natural” grouping of instances given un-labeled data

**Classification:**
Supervised learning:
Learns a method for predicting the instance class from pre-labeled (classified) instances
Examples of Clustering Applications

- **Marketing**: discover customer groups and use them for targeted marketing and re-organization
- **Astronomy**: find groups of similar stars and galaxies
- **Earth-quake studies**: Observed earth quake epicenters should be clustered along continent faults
- **Genomics**: finding groups of gene with similar expressions
- **WWW**
  - Document classification
  - Cluster Weblog data to discover groups of similar access patterns
What Is Good Clustering?

- A **good clustering** method will produce high quality clusters with
  - high *intra-class* similarity
  - low *inter-class* similarity

- The **quality of a clustering result** depends on both the similarity measure used by the method and its implementation.

- The **quality of a clustering method** is also measured by its ability to discover some or all of the *hidden* patterns.
Requirements of Clustering in Data Mining

- Scalability
- Ability to deal with different types of attributes
- Discovery of clusters with arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters
- Able to deal with noise and outliers
- Insensitive to order of input records
- High dimensionality
- Incorporation of user-specified constraints
- Interpretability and usability
Agenda

- Introduction
- Clustering Methods
- Techniques for Improving the Efficiency
- Applications:
  - Medical Image Clustering
  - Document Clustering
  - Outlier Analysis
- Summary and Conclusions
Types of Clustering Algorithms

- Hierarchical vs. flat
- For numeric and/or symbolic data
- Deterministic vs. probabilistic
- Exclusive vs. overlapping
- Top-down vs. bottom-up
Clusters: Exclusive vs. Overlapping

Flat, non-overlapping, deterministic

Flat, overlapping, deterministic
Clusters: Hierarchical vs. Flat

Hierarchical, non-overlapping, deterministic

Flat, overlapping, probabilistic
Major Clustering Methods

- **Partitioning algorithms**: Construct various partitions and then evaluate them by some criterion
- **Hierarchy algorithms**: Create a hierarchical decomposition of the set of data (or objects) using some criterion
- **Density-based**: based on connectivity and density functions
- **Grid-based**: based on a multiple-level granularity structure
- **Model-based**: A model is hypothesized for each of the clusters and the idea is to find the best fit of that model to each other
Agenda

- Introduction
- Clustering Methods
  - Partitioning Methods
  - Hierarchical Methods
  - Density-Based Methods
  - Grid-Based Methods
- Applications
- Summary and Conclusions
Partitioning Algorithms: Basic Concept

- **Partitioning method**: Construct a partition of a database $D$ of $n$ objects into a set of $k$ clusters
- Given a $k$, find a partition of $k$ clusters that optimizes the chosen partitioning criterion
  - **k-means** (MacQueen’67): Each cluster is represented by the center of the cluster
  - **k-medoids** or PAM (Partition Around Medoids) (Kaufman & Rousseeuw’87): Each cluster is represented by one of the objects in the cluster
The *K-Means* Clustering Method

- Given *k*, the *k-means* algorithm is implemented in 4 steps:
  
  **Step 1.** Partition objects into *k* nonempty subsets
  **Step 2.** Compute seed points as the centroids of the clusters of the current partition. The centroid is the center (mean point) of the cluster.
  **Step 3.** Assign each object to the cluster with the nearest seed point.
  **Step 4.** Go back to Step 2, stop when no more new assignment.
K-means Example, Step 1

Pick 3 initial cluster centers (randomly)
K-means Example, Step 2

Assign each point to the closest cluster center.
K-means Example, Step 3

Move each cluster center to the mean of each cluster.
K-means Example, Step 4

Reassign points closest to a different new cluster center

Q: Which points are reassigned?
K-means Example, Step 4 …

A: three points with animation
K-means Example, Step 4b

re-compute cluster means
K-means Example, Step 5

move cluster centers to cluster means
Discussion

- Result can vary significantly depending on initial choice of seeds
- Can get trapped in local minimum
  - Example:
- To increase chance of finding global optimum: restart with different random seeds
K-means Clustering Summary

Advantages
- Simple, understandable
- Items automatically assigned to clusters

Disadvantages
- Must pick number of clusters before hand
- All items forced into a cluster
- Too sensitive to outliers
The *K-Medoids* Clustering Method

- Find representative objects, called *medoids*, in clusters
- *PAM* (Partitioning Around Medoids, 1987)
  - starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering
- *CLARA* (Kaufmann & Rousseeuw, 1990)
- *CLARANS* (Ng & Han, 1994): Randomized sampling
PAM (Partitioning Around Medoids)

- PAM (Kaufman and Rousseeuw, 1987)
- Use real object to represent the cluster

**Step 1.** Select $k$ representative objects arbitrarily

**Step 2.** For each pair of non-selected object $h$ and selected object $i$, calculate the total swapping cost $TC_{ih}$

**Step 3.** For each pair of $i$ and $h$, if $(TC_{ih} < 0)$, $i$ is replaced by $h$. Then assign each non-selected object to the most similar representative object

**Step 4.** repeat steps 2-3 until there is no change
PAM Clustering: Total swapping cost

$$TC_{ih} = \sum C_{jih}$$

- $C_{jih} = d(j, h) - d(j, i)$
- $C_{jih} = d(j, t) - d(j, i)$
- $C_{jih} = 0$
Agenda

- Introduction
- Clustering Methods
  - Partitioning Methods
  - Hierarchical Methods
  - Density-Based Methods
  - Grid-Based Methods
- Applications
- Summary and Conclusions
Hierarchical Clustering

- This method does not require the number of clusters $k$ as an input, but needs a termination condition.
Agglomerative Approach

- Start with single-instance clusters
- At each step, join the two closest clusters
- Design decision: distance between clusters
  - E.g. two closest instances in clusters vs. distance between means
Divisive Approach

- Start with one universal cluster
- Find two clusters
- Proceed recursively on each subset
- Can be very fast
A **Dendrogram** Shows How the Clusters are Merged Hierarchically

Decompose data objects into a several levels of nested partitioning (tree of clusters), called a **dendrogram**.

A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster.
Linkage Hierarchies

- Single Linkage
- Complete Linkage
- Average Linkage / Centroid Linkage
Single Linkage

- Distance between clusters (nodes):
  \[ Dist(C_1, C_2) = \min_{p \in C_1, q \in C_2} \{dist(p, q)\} \]

- Merge Step:
  Union of two subset of data points

- A single linkage hierarchy can be constructed using the Minimal Spanning Tree
Complete Linkage

- Distance between clusters (nodes):
  \[ Dist(C_1, C_2) = \max_{p \in C_1, q \in C_2} \{dist(p, q)\} \]

- Merge Step:
  Union of two subset of data points

- Each cluster in a complete linkage hierarchy corresponds to a complete subgraph
Average Linkage / Centroid Method

- Distance between clusters (nodes):
  \[
  Dist_{\text{avg}}(C_1, C_2) = \frac{1}{\#(C_1) \cdot \#(C_2)} \sum_{p \in C_1} \sum_{q \in C_2} dist(p, q)
  \]
  \[
  Dist_{\text{mean}}(C_1, C_2) = dist[\text{mean}(C_1), \text{mean}(C_2)]
  \]

- Merge Step:
  - Union of two subset of data points
  - Construct the mean point of the two clusters
More on Hierarchical Clustering Methods

- Major weakness of agglomerative clustering methods
  - do not scale well: time complexity of at least $O(n^2)$, where $n$ is the number of total objects
  - can never undo what was done previously

- Integration of hierarchical with distance-based clustering
  - BIRCH (1996): uses CF-tree and incrementally adjusts the quality of sub-clusters
BIRCH

- Birch: Balanced Iterative Reducing and Clustering using Hierarchies, by Zhang, Ramakrishnan, Livny (SIGMOD’96)
- Incrementally construct a CF (Clustering Feature) tree, a hierarchical data structure for multiphase clustering
  - Phase 1: scan DB to build an initial in-memory CF tree (a multi-level compression of the data that tries to preserve the inherent clustering structure of the data)
  - Phase 2: use an arbitrary clustering algorithm to cluster the leaf nodes of the CF-tree
BIRCH

- *Scales linearly:* finds a good clustering with a single scan and improves the quality with a few additional scans

- *Weakness:* handles only numeric data, and sensitive to the order of the data record.
Basic Idea of the CF-Tree

• Condensation of the data using CF-Vectors
  • Clustering Feature Vector:
    \[ CF = (N, \vec{LS}, SS) \]
    \[ N: \text{number of objects in the cluster} \]
    \[ \vec{LS} = \sum_{i=1}^{N} \vec{X}_i \]
    \[ SS = \sum_{i=1}^{N} \vec{X}_i^2 \]

• CF-tree uses sum of CF-vectors to build higher levels of the CF-tree

\[ CF = (5, (16,30),(54,190)) \]

\( (3,4) \)
\( (2,6) \)
\( (4,5) \)
\( (4,7) \)
\( (3,8) \)
CF Tree

Root

<table>
<thead>
<tr>
<th>CF_1</th>
<th>CF_2</th>
<th>CF_3</th>
<th>......</th>
<th>CF_6</th>
</tr>
</thead>
<tbody>
<tr>
<td>child_1</td>
<td>child_2</td>
<td>child_3</td>
<td>......</td>
<td>child_6</td>
</tr>
</tbody>
</table>

Non-leaf node

<table>
<thead>
<tr>
<th>CF_1</th>
<th>CF_2</th>
<th>CF_3</th>
<th>......</th>
<th>CF_5</th>
</tr>
</thead>
<tbody>
<tr>
<td>child_1</td>
<td>child_2</td>
<td>child_3</td>
<td>......</td>
<td>child_5</td>
</tr>
</tbody>
</table>

Leaf node

<table>
<thead>
<tr>
<th>prev</th>
<th>CF_1</th>
<th>CF_2</th>
<th>......</th>
<th>CF_6</th>
<th>next</th>
</tr>
</thead>
</table>

Leaf node

<table>
<thead>
<tr>
<th>prev</th>
<th>CF_1</th>
<th>CF_2</th>
<th>......</th>
<th>CF_4</th>
<th>next</th>
</tr>
</thead>
</table>

B = 7
L = 6
Insertion Algorithm for a New Point $x$

**Step 1.** Find the closest leaf $b$

**Step 2.** If $x$ fits in $b$, insert $x$ in $b$;
otherwise split $b$

**Step 3.** Modify the path for $b$

**Step 4.** If tree is too large, condense the tree by merging the closest leaves
Clustering in BIRCH

Phase 1-2 produces a condensed representation of the data (CF-tree).

Phase 3-4 applies a separate cluster algorithm to the leaves of the CF-tree.
Drawbacks of Distance-Based Method

- Drawbacks of square-error based clustering method
  - Consider only one point as representative of a cluster
  - Good only for convex shaped, similar size and density, and if $k$ can be reasonably estimated
Agenda

- Introduction
- Clustering Methods
  - Partitioning Methods
  - Hierarchical Methods
  - Density-Based Methods
  - Grid-Based Methods
- Applications
- Summary and Conclusions
Density-Based Clustering Methods

- **Major features:**
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan
  - Need density parameters as termination condition

- **Several interesting studies:**
  - **DBSCAN:** Ester, et al. (KDD’96)
  - **OPTICS:** Ankerst, et al. (SIGMOD’99).
  - **DENCLUE:** Hinneburg & D. Keim (KDD’98)
  - **CLIQUE:** Agrawal, et al. (SIGMOD’98)
Density-Based Clustering: Background

- **Two parameters:**
  - $Eps$: Maximum radius of the neighbourhood
  - $MinPts$: Minimum number of points in an Eps-neighbourhood of that point

- $N_{Eps}(p)$: \( \{q \text{ belongs to } D \mid dist(p,q) \leq Eps\} \)

- Directly density-reachable: A point $p$ is directly density-reachable from a point $q$ wrt. $Eps$, $MinPts$ if
  - 1) $p$ belongs to $N_{Eps}(q)$
  - 2) core point condition:
    \[ |N_{Eps}(q)| \geq MinPts \]

MinPts = 5
Eps = 1 cm
Density-Based Clustering: Background (II)

- **Density-reachable:**
  - A point \( p \) is density-reachable from a point \( q \) wrt. \( Eps, MinPts \) if there is a chain of points \( p_1, \ldots, p_n, p_1 = q, p_n = p \) such that \( p_{i+1} \) is directly density-reachable from \( p_i \).

- **Density-connected**
  - A point \( p \) is density-connected to a point \( q \) wrt. \( Eps, MinPts \) if there is a point \( o \) such that both, \( p \) and \( q \) are density-reachable from \( o \) wrt. \( Eps \) and \( MinPts \).
DBSCAN: Density Based Spatial Clustering of Applications with Noise

- Relies on a *density-based* notion of cluster: A *cluster* is defined as a maximal set of density-connected points.
- Discovers clusters of arbitrary shape in spatial databases with noise.

**Diagram:**
- **Core**
- **Border**
- **Outlier**

**Parameters:**
- $Eps = 1\, \text{cm}$
- $MinPts = 5$
DBSCAN: The Algorithm

- Arbitrary select a point $p$
- Retrieve all points density-reachable from $p$ wrt $Eps$ and $MinPts$.
- If $p$ is a core point, a cluster is formed.
- If $p$ is a border point, no points are density-reachable from $p$ and DBSCAN visits the next point of the database.
- Continue the process until all of the points have been processed.
Agenda

- Introduction
- Clustering Methods
  - Partitioning Methods
  - Hierarchical Methods
  - Density-Based Methods
  - Grid-Based Methods
- Applications
- Summary and Conclusions
Grid-Based Clustering Method

- Using multi-resolution grid data structure
- Several interesting methods:
  - CLIQUE: Agrawal, et al. (SIGMOD’98)
  - STING (a STatistical INformation Grid approach) by Wang, Yang and Muntz (1997)
  - WaveCluster by Sheikholeslami, Chatterjee, and Zhang (VLDB’98)
- A multi-resolution clustering approach using wavelet method
CLIQUE (Clustering In QUEst)

- Agrawal, Gehrke, Gunopulos, Raghavan (SIGMOD’98).
- Automatically identifying subspaces of a high dimensional data space that allow better clustering than original space
- CLIQUE can be considered as both density-based and grid-based
  - It partitions each dimension into the same number of equal length interval
  - It partitions an m-dimensional data space into non-overlapping rectangular units
  - A unit is dense if the fraction of total data points contained in the unit exceeds the input model parameter
  - A cluster is a maximal set of connected dense units within a subspace
CLIQUE: The Major Steps

- Partition the data space and find the number of points that lie inside each cell of the partition.
- Identify the subspaces that contain clusters using the Apriori principle.
- Identify clusters:
  - Determine dense units in all subspaces of interests
  - Determine connected dense units in all subspaces of interests.
- Generate minimal description for the clusters
  - Determine maximal regions that cover a cluster of connected dense units for each cluster
  - Determination of minimal cover for each cluster
\[ \tau = 3 \]
Strength and Weakness of **CLIQUE**

**Strength**

- It *automatically* finds subspaces of the highest dimensionality such that high density clusters exist in those subspaces.
- It is *insensitive* to the order of records in input and does not presume some canonical data distribution.
- It scales *linearly* with the size of input and has good scalability as the number of dimensions in the data increases.

**Weakness**

- The accuracy of the clustering result may be degraded at the expense of simplicity of the method.
Agenda

- Introduction
- Clustering Methods
- Applications:
  - Outlier Analysis
  - Gene clustering
- Summary and Conclusions
What Is Outlier Discovery?

- What are outliers?
  - The set of objects are considerably dissimilar from the remainder of the data
  - Example: Sports: Michael Jordan, Wayne Gretzky, ...

- Problem
  - Find top n outlier points

- Applications:
  - Credit card fraud detection
  - Telecom fraud detection
  - Customer segmentation
  - Medical analysis
Outlier Discovery: Statistical Approaches

- Assume a model underlying distribution that generates data set (e.g. normal distribution)

- Use discordancy tests depending on
  - data distribution
  - distribution parameter (e.g., mean, variance)
  - number of expected outliers

- Drawbacks
  - most tests are for single attribute

- In many cases, data distribution may not be known
Outlier Discovery: Distance-Based Approach

- Introduced to counter the main limitations imposed by statistical methods
  - We need multi-dimensional analysis without knowing data distribution.
- Distance-based outlier: A DB(p, D)-outlier is an object O in a dataset T such that at least a fraction p of the objects in T lies at a distance greater than D from O.
- Algorithms for mining distance-based outliers
  - Index-based algorithm
  - Nested-loop algorithm
  - Cell-based algorithm
Outlier Discovery: Deviation-Based Approach

- Identifies outliers by examining the main characteristics of objects in a group
- Objects that “deviate” from this description are considered outliers
- Sequential exception technique
  - simulates the way in which humans can distinguish unusual objects from among a series of supposedly like objects
- OLAP data cube technique
  - uses data cubes to identify regions of anomalies in large multidimensional data
Agenda

- Introduction
- Clustering Methods
- Evaluating Clustering Models
- Applications:
  - Outlier Analysis
  - Gene Clustering
- Summary and Conclusions
Expression Vectors

Gene Expression Vectors encapsulate the expression of a gene over a set of experimental conditions or sample types.

Numeric Vector

| -0.8 | 1.5 | 1.8 | 0.5 | -0.4 | -1.3 | 0.8 | 1.5 |

Line Graph

Heatmap

-2 2
Expression Vectors As Points in ‘Expression Space’

<table>
<thead>
<tr>
<th></th>
<th>t1</th>
<th>t2</th>
<th>t3</th>
</tr>
</thead>
<tbody>
<tr>
<td>G1</td>
<td>-0.8</td>
<td>-0.3</td>
<td>-0.7</td>
</tr>
<tr>
<td>G2</td>
<td>-0.4</td>
<td>-0.8</td>
<td>-0.7</td>
</tr>
<tr>
<td>G3</td>
<td>-0.6</td>
<td>-0.8</td>
<td>-0.4</td>
</tr>
<tr>
<td>G4</td>
<td>0.9</td>
<td>1.2</td>
<td>1.3</td>
</tr>
<tr>
<td>G5</td>
<td>1.3</td>
<td>0.9</td>
<td>-0.6</td>
</tr>
</tbody>
</table>

Similar Expression

Experiment 1

Experiment 2

Experiment 3

Clustering
Distance and Similarity

-the ability to calculate a distance (or similarity, it’s inverse) between two expression vectors is fundamental to clustering algorithms

-distance between vectors is the basis upon which decisions are made when grouping similar patterns of expression

-selection of a distance metric defines the concept of distance
Distance: a measure of similarity between gene expression.

<table>
<thead>
<tr>
<th></th>
<th>Exp 1</th>
<th>Exp 2</th>
<th>Exp 3</th>
<th>Exp 4</th>
<th>Exp 5</th>
<th>Exp 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gene A</td>
<td>x_{1A}</td>
<td>x_{2A}</td>
<td>x_{3A}</td>
<td>x_{4A}</td>
<td>x_{5A}</td>
<td>x_{6A}</td>
</tr>
<tr>
<td>Gene B</td>
<td>x_{1B}</td>
<td>x_{2B}</td>
<td>x_{3B}</td>
<td>x_{4B}</td>
<td>x_{5B}</td>
<td>x_{6B}</td>
</tr>
</tbody>
</table>

Some distances: (MeV provides 11 metrics)

1. Euclidean: \( \sqrt{\sum_{i=1}^{6} (x_{iA} - x_{iB})^2} \)

2. Manhattan: \( \sum_{i=1}^{6} |x_{iA} - x_{iB}| \)

3. Pearson correlation
Hierarchical Clustering

Gene 1

Gene 2

Gene 3

Gene 4

Gene 5

Gene 6

Gene 7

Gene 8
Hierarchical Clustering

Gene 1

Gene 2

Gene 4

Gene 5

Gene 3

Gene 8

Gene 6

Gene 7
Hierarchical Clustering
Hierarchical Clustering

Gene 7

Gene 1

Gene 2

Gene 4

Gene 5

Gene 3

Gene 8

Gene 6
Hierarchical Clustering

Gene 7

Gene 1

Gene 2

Gene 4

Gene 5

Gene 3

Gene 8

Gene 6
Hierarchical Clustering

Gene 1
Gene 2
Gene 4
Gene 5
Gene 3
Gene 8
Gene 6
Gene 7
Gene 8
Hierarchical Clustering

Gene 7
Gene 1
Gene 2
Gene 4
Gene 5
Gene 3
Gene 8
Gene 6
Hierarchical Clustering
Hierarchical Clustering

The Leaf Ordering Problem:
- Find ‘optimal’ layout of branches for a given dendrogram architecture
- $2^{N-1}$ possible orderings of the branches
- For a small microarray dataset of 500 genes there are $1.6 \times 10^{150}$ branch configurations
Hierarchical Clustering

The Leaf Ordering Problem:
Agenda

- Introduction
- Clustering Methods
- Applications
- Summary and Conclusions
Problems and Challenges

- Considerable progress has been made in scalable clustering methods
  - Partitioning: k-means, k-medoids, PAM
  - Hierarchical: BIRCH
  - Density-based: DBSCAN
  - Grid-based: CLIQUE

- Current clustering techniques do not address all the requirements adequately

- Constraint-based clustering analysis: Constraints exist in data space (bridges and highways) or in user queries
Summary

- **Cluster analysis** groups objects based on their similarity and has wide applications.

- Measure of similarity can be computed for various types of data.

- Clustering algorithms can be categorized into partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods.

- **Outlier detection** and analysis are very useful for fraud detection, etc. and can be performed by statistical, distance-based or deviation-based approaches.

- There are still lots of research issues on cluster analysis, such as constraint-based clustering.
References (1)

- R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. SIGMOD'98
References (2)

- E. Knorr and R. Ng. Algorithms for mining distance-based outliers in large datasets. VLDB'98.
- R. Ng and J. Han. Efficient and effective clustering method for spatial data mining. VLDB'94.
- G. Sheikholeslami, S. Chatterjee, and A. Zhang. WaveCluster: A multi-resolution clustering approach for very large spatial databases. VLDB’98.
- T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: an efficient data clustering method for very large databases. SIGMOD'96.