DATA MINING 2
Transactional Clustering

Riccardo Guidotti

a.a. 2021/2022
Clustering

- **Clustering**: Grouping of objects into different sets, or more precisely, the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait - often proximity according to some defined distance measure.

- Common distance functions:
  - Euclidean distance, Manhattan distance, ...

- This kind of distance functions are suitable for **numerical data**.
Not Only Numerical Data

### Numerical Data

<table>
<thead>
<tr>
<th>Acceleration 0-100 (s)</th>
<th>Length (m)</th>
<th>Width (m)</th>
<th>Height (m)</th>
<th>Price (€)</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>4</td>
<td>1.6</td>
<td>1.7</td>
<td>20'000</td>
</tr>
<tr>
<td>14</td>
<td>3.7</td>
<td>1.5</td>
<td>1.65</td>
<td>16'000</td>
</tr>
<tr>
<td>15</td>
<td>3.5</td>
<td>1.5</td>
<td>1.6</td>
<td>12'000</td>
</tr>
<tr>
<td>9.4</td>
<td>4.2</td>
<td>1.8</td>
<td>1.7</td>
<td>24'000</td>
</tr>
</tbody>
</table>

### Categorical Data

<table>
<thead>
<tr>
<th>Hairs</th>
<th>Eyes</th>
</tr>
</thead>
<tbody>
<tr>
<td>brown</td>
<td>black</td>
</tr>
<tr>
<td>blond</td>
<td>blue</td>
</tr>
<tr>
<td>black</td>
<td>green</td>
</tr>
<tr>
<td>red</td>
<td>brown</td>
</tr>
</tbody>
</table>
Boolean and Categorical Attributes

• A **boolean** attribute corresponding to a single item in a transaction, if that item appears, the boolean attribute is set to ‘1’ or ‘0’ otherwise.

• A **categorical** attribute may have several values, each value can be treated as an item and represented by a boolean attribute.
Market Basket Data

• A transaction represents one customer, and each transaction contains set of items purchased by the customer.
• Clustering customers reveals customers with similar buying patterns putting them into the same cluster.
• It is useful for
  • Characterizing different customer groups
  • Targeted Marketing
  • Predict buying patterns of new customers based on profile
• A market basket database: A scenario where attributes of data points are non-numeric, transaction viewed as records with boolean attributes corresponding to a single item (TRUE if transaction contain item, FALSE otherwise).
• **Boolean** attributes are special case of **Categorical** attributes.
Shortcomings of Traditional Clustering

• For categorical data we:
  • Define new criterion for *neighbors* and/or *similarity*
  • Define the ordering criterion

• Consider the following 4 market basket transactions

<table>
<thead>
<tr>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
</tr>
</thead>
<tbody>
<tr>
<td>{1, 2, 3, 4}</td>
<td>{1, 2, 4}</td>
<td>{3}</td>
<td>{4}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1, 1, 1, 1)</td>
<td>(1, 1, 0, 1)</td>
<td>(0, 0, 1, 0)</td>
<td>(0, 0, 0, 1)</td>
</tr>
</tbody>
</table>

• using Euclidean distance to measure the closeness between all pairs of points, we find that \(d(P1,P2)\) is the smallest distance: it is equal to 1
Shortcomings of Traditional Clustering

• If we use a hierarchical algorithm then we merge P1 and P2 and get a new cluster (P12) with (1, 1, 0.5, 1) as a centroid.

• Then, using Euclidean distance again, we find:
  • $d(p_{12},p_3) = \sqrt{3.25}$
  • $d(p_{12},p_4) = \sqrt{2.25}$
  • $d(p_3,p_4) = \sqrt{2}$

• So, we should merge P3 and P4 since the distance between them is the shortest.

• However, T3 and T4 don't have even a single common item.

• So, using distance metrics as similarity measure for categorical data is not appropriate.
Clustering Algorithms for Categorical/Transactionial Data

• K-Modes
• ROCK
• CLOPE
• TX-Means
K-Modes

Minimise \[ P(W, Q) = \sum_{l=1}^{k} \sum_{i=1}^{n} w_{i,l} d(X_i, Q_l) \]

subject to \[ \sum_{l=1}^{k} w_{i,l} = 1, \quad 1 \leq i \leq n \]
\[ w_{i,l} \in \{0, 1\}, \quad 1 \leq i \leq n, \; 1 \leq l \leq k \]

• \( X = \{ X_1, \ldots, X_n \} \) is the dataset of objects.
• \( X_i = [x_1, \ldots, x_m] \) is an object i.e., a vector of \( m \) categorical attributes
• \( W \) is a matrix \( n \times k \), with \( w_{i,l} \) equal to 1 if \( X_i \) belongs to Cluster \( l \), 0 otherwise.
• \( Q = \{ Q_1, \ldots, Q_k \} \) is the set of representative objects (mode) for the \( k \) clusters.
• \( d(X_i, Q_l) \) is a distance function for objects in the data
K-Modes: Distance

- K-Means as distance uses Euclidean distance
- K-Modes as distance uses the number of mismatches between the attributes of two objects.

\[ d(X, Y) = \sum_{i=1}^{m} (x_i - y_i)^2 \]

\[ d_1(X, Y) = \sum_{j=1}^{m} \delta(x_j, y_j) \]

\[ \delta(x_j, y_j) = \begin{cases} 
0 & (x_j = y_j) \\
1 & (x_j \neq y_j) 
\end{cases} \]
K-Modes: Mode

• K-Modes uses the mode as representative object of a cluster
• Given the set of objects in the cluster $C_l$ the mode is get computing the max frequency for each attribute

$$f_r(A_j = c_{l,j} \mid X_l) = \frac{n_{c_{l,k}}}{n}$$
K-Modes: Algorithm

1. Randomly select the initial objects as modes
2. Scan of the data to assign each object to the closer cluster identified by the mode
3. Re-compute the mode of each cluster
4. Repeat the steps 2 and 3 until no object changes the assigned cluster
K-Modes: Algorithm

1. Randomly select the initial objects as modes
2. Scan of the data to assign each object to the closer cluster identified by the mode
3. Re-compute the mode of each cluster
4. Repeat the steps 2 and 3 until no object changes the assigned cluster
ROCK: RObust Clustering using linK

• ROCK is a **hierarchical** algorithm for clustering transactional data (market basket databases)
• ROCK uses **links to cluster** instead of the classical distance notion
• ROCK uses the notion of **neighborhood** between pair of objects to identify **the number of links** between two objects
ROCK: The Neighbors Concept

• It captures a notion of similarity
  • A and B are neighbors if \( \text{sim}(A, B) \geq \theta \)

• ROCK uses the Jaccard coefficient
  • \( \text{sim}(A, B) = \frac{|A \cap B|}{|A \cup B|} \)

\[
A = \{1, 3, 4, 7\} \\
B = \{1, 2, 4, 7, 8\}
\]

\[
\text{sim}(A,B) = \frac{3}{6} = \frac{1}{2} = 0.5
\]
A link defines the number of common neighbors between two objects. 

\[ \text{link}(A, B) = |\text{neighbor}(A) \cap \text{neighbor}(B)| \]

Higher values of \( \text{link}(A, B) \) means higher probability that \( A \) and \( B \) belong to the same cluster.

Similarity is local while link is capturing global information.

A point is considered a neighbor of itself.

There is a link from each neighbor of the “root” point back to itself through the root.

Therefore, if a point has \( n \) neighbors, then \( n^2 \) links are due to it.
ROCK: Example

- Data consisting of 6 Attributes:
  - {Book}
  - {Water, Sun, Sand, Swimming}
  - {Water, Sun, Sand, Reading}
  - {Reading, Sand}

- Resulting Jaccard Coefficient Matrix

- Set Threshold = 0.2. Neighbors:
  - N(A)={A}; N(B)={B,C,D}
  - N(C)={B,C,D}, N(D)={B,C,D}

- Number of Links Table
  - Link (B, C) = |{B,C,D}| = 3

- Resulting Clusters after applying ROCK: {A}, {B,C,D}
\[ E_i = \sum_{i=1}^{k} n_i \times \sum_{p_q, p_r \in C_i} \frac{\text{link}(p_q, p_r)}{n_i^{1+2f(\theta)}} \]

Maximize \[ f(\theta) = \frac{1-\theta}{1+\theta} \]

This goodness measure helps to identify the best pair of clusters to be merged during each step of ROCK.

\[ g(C_i, C_j) = \frac{\text{link}[C_i, C_j]}{(n_i + n_j)^{1+2f(\theta)} - n_i^{1+2f(\theta)} - n_j^{1+2f(\theta)}} \]

Where \( C_i \) denotes cluster \( i \)

- \( n_i \) is the number of points in \( C_i \)
- \( k \) is the number of clusters
- \( \theta \) is the similarity threshold

Dividing by the number of expected links between pairs of objects in the cluster \( C_i \), we avoid that objects with a low number of links are assigned all to the same cluster.

Number of expected cross-links between two clusters
ROCK: Clustering Algorithm

Input:
- A set $S$ of data points
- Number of $k$ clusters to be found
- The similarity threshold

Output:
- Groups of clustered data

The ROCK algorithm is divided into three major parts:
1. Draw a random sample from the data set
2. Perform a hierarchical agglomerative clustering algorithm
3. Label data
ROCK: Clustering Algorithm

Draw a random sample from the data set:

• Sampling is used to ensure scalability to very large data sets
• The initial sample is used to form clusters, then the remaining data on dataset is assigned to these clusters
Perform a hierarchical agglomerative clustering algorithm:

- ROCK performs the following steps which are common to all hierarchical agglomerative clustering algorithms, but with different definition to the similarity measures:
  1. Places each single data point into a separate cluster
  2. Compute the similarity measure for all pairs of clusters
  3. Merge the two clusters with the highest similarity (goodness measure)
  4. Verify a stop condition. If it is not met, then go to step 2.
ROCK: Clustering Algorithm

Label data

• Finally, the remaining data points are assigned to the clusters.
• This is done by selecting a random sample $L_i$ from each cluster $C_i$, then we assign each point $p$ to the cluster for which it has the strongest linkage with $L_i$. 
ROCK Summary

Input: dataset, number of clusters.
1. Draw a random sample from the data set
2. Places each data point into a separate cluster
3. Compute the similarity measure for all pairs of clusters
4. Merge the two clusters with the highest similarity
5. Verify a stop condition. If it is not met, then go to step 2.
6. Assign not used points to clusters using linkage similarity with respect to selected samples from each cluster
CLOPE: Clustering with sLOPE

- Transactional clustering efficient for high dimensional data
- Uses a **global criterion function** that tries to increase the intra-cluster overlapping of transaction items **by increasing the height-to-width ratio of the cluster histogram**.

**Example**: 5 transactions \{a,b\} \{a,b,c\} \{a,c,d\} \{d,e\} \{d,e,f\}

\[
\begin{align*}
D(C) &= \text{set of items in } C \\
S(C) &= \sum_{i \in C} |t_i| \\
W(C) &= |D(C)| \\
H(C) &= S(C) / W(C)
\end{align*}
\]

Clustering 1

- \{a, b, c\} \{d, e, f\}
- \(H=2.0, W=4\) \(H/W=0.5\)
- \{a, b, c\} \{a, c, d\}
- \(H=1.67, W=3\) \(H/W=0.55\)

Clustering 2

- \{a, b, c\} \{d, e, f\}
- \(H=1.67, W=3\) \(H/W=0.55\)
- \{a, c, d\} \{d, e, f\}
- \(H=1.6, W=5\) \(H/W=0.32\)

Higher \(H/W\) means higher item overlapping
CLOPE: Criterion Function

• For evaluating the goodness of a clustering the gradient of a cluster is
• $G(C) = H(C)/W(C) = S(C)/W(C)^2$

Repulsion.
When $r$ is large, transactions within the same cluster must share a large portion of common items.
CLOPE: Algorithm

/* Phrase 1 - Initialization */
1: while not end of the database file
2:  read the next transaction \( \langle t, \text{unknown} \rangle \);
3:  put \( t \) in an existing cluster or a new cluster \( C_i \)
    that maximize profit;
4:  write \( \langle t, i \rangle \) back to database;

/* Phrase 2 - Iteration */
5: repeat
6:  rewind the database file;
7:  \( moved = \text{false} \);
8:  while not end of the database file
9:  \( \text{read} \ \langle t, i \rangle \);
10: move \( t \) to an existing cluster or new cluster \( C_j \)
    that maximize profit;
11: if \( C_i \neq C_j \) then
12:  write \( \langle t, j \rangle \);
13:  \( moved = \text{true} \);
14: until not \( moved \);
CLOPE Summary

Input: dataset, repulsion, maximum number of clusters

• Phase 1
  1. For each transaction, add it to a new cluster or to an existing one such that the profit is maximized

• Phase 2
  1. For each transaction, try to move it to another cluster and do it if this maximizes the profit
  2. Repeat 1. until all the transactions remain in the same cluster
TX-MEANS

• A parameter-free clustering algorithm able to efficiently partitioning transactional data automatically

• Suitable for the case where clustering must be applied on a massive number of different datasets
  • E.g.: when a large set of users need to be analyzed individually and each of them has generated a long history of transactions

• TX-Means automatically estimates the number of clusters

• TX-Means provides the representative transaction of each cluster, which summarizes the pattern captured by that cluster.
How It Works 1/3
How It Works 2/3
How It Works 3/3

- Clusters

- Representative Baskets
TX-Means Algorithm

**TXMEANS**(*B*: baskets):

- \( r \leftarrow \text{GETREPR}(B) \);  
- \( Q\).push\((B,r)\);  
- **While** there is a cluster \( B,r \) to split in \( Q \):
  - Remove common items from \( B \);  
  - \( B1, B2, r1, r2 \leftarrow \text{BISECTBASKET}(B) \);  
  - **If** \( \text{BIC}(B1,B2,r1,r2) > \text{BIC}(B,r) \) **Then:**
    - add \( B1,B2,r1,r2 \) to the clusters to split \( Q \);  
  - **Else**
    - add \( B,r \) to the clustering result \( C \);  
- Return \( C \);
Bisecting Schema

**BISECTBASKET**(B: baskets):

- SSE \text{ <-- inf};
- r1, r2 \text{ <-- select random initial baskets in B as representative};
- **While** True:
  - C1, C2 \text{ <-- assign baskets in B with respect to r1, r2};
  - r1\text{\_new} \text{ <-- GETREPR(C1); r2\text{\_new} <-- GETREPR(C2)};
  - SSE\text{\_new} \text{ <-- SSE(C1,C2,r1\text{\_new},r2\text{\_new})};
  - **If** SSE\text{\_new} >= SSE **Then**:
    - Return C1, C2, r1, r2;
    - r1, r2 \text{ <-- r1\text{\_new},r2\text{\_new}};
Get Representative Baskets

\textbf{GETREPR}(B: \text{baskets}):

\begin{itemize}
  \item I \leftarrow \text{not common items in } B;
  \item r \leftarrow \text{common items in } B;
  \item While I is not empty:
    \begin{itemize}
      \item Add to r the items with maximum frequency in I;
      \item Calculate the distance between r and the baskets in B;
      \item If the distance no longer decreases Then:
        \begin{itemize}
          \item Return r;
        \end{itemize}
      \item Else
        \begin{itemize}
          \item remove from I the items with maximum frequency;
        \end{itemize}
    \end{itemize}
  \item Return r;
\end{itemize}

overlap-based distance function (Jaccard coefficient)
Dealing with Big Datasets

• Clustering of a big individual transactional dataset $B$.
• TX-Means is scalable thanks to the following sampling strategy.

• Sampling strategy:
  • Random selection of a subset $S$ of the baskets in $B$;
  • Run of TX-Means on the subset $S$ and obtain clusters $C$ and representative baskets $R$;
  • Assign the remaining baskets $B/S$ to the clusters $C$ using a nearest neighbor approach with respect to the representative baskets $R$. 
References


Exercises Transactional Clustering
Rock – Exercise 1

- Suppose we have four verses contains some subjects, as follows:
  - P1 = {judgment, faith, prayer, fair}
  - P2 = {fasting, faith, prayer}
  - P3 = {fair, fasting, faith}
  - P4 = {fasting, prayer, pilgrimage}
- the similarity threshold = 0.3, and number of required cluster is 2.

Using Jaccard coefficient as a similarity measure, we obtain the following similarity table:

<table>
<thead>
<tr>
<th></th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>1</td>
<td>0.4</td>
<td>0.4</td>
<td>0.17</td>
</tr>
<tr>
<td>P2</td>
<td></td>
<td>1</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>P3</td>
<td></td>
<td></td>
<td>1</td>
<td>0.2</td>
</tr>
<tr>
<td>P4</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>
Rock – Exercise 1

• Since we have a similarity threshold equal to 0.3, then we derive the adjacency table:

<table>
<thead>
<tr>
<th></th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>1</td>
<td>0.4</td>
<td>0.4</td>
<td>0.17</td>
</tr>
<tr>
<td>P2</td>
<td>1</td>
<td>0.5</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>P3</td>
<td>1</td>
<td></td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>P4</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

• By multiplying the adjacency table with itself, we derive the following table which shows the number of links (or common neighbors):

<table>
<thead>
<tr>
<th></th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>P2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>P3</td>
<td>1</td>
<td></td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>P4</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>-</td>
<td>3</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>P2</td>
<td>-</td>
<td></td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>P3</td>
<td>-</td>
<td></td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>P4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Rock – Exercise 1

• we compute the goodness measure for all adjacent points, assuming that

\[ f(\theta) = \frac{1-\theta}{1+\theta} = \frac{1-0.3}{1+0.3} = 0.54 \]

• we obtain the following table →

<table>
<thead>
<tr>
<th>Pair</th>
<th>Goodness measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1,P2</td>
<td>1.35</td>
</tr>
<tr>
<td>P1,P3</td>
<td>1.35</td>
</tr>
<tr>
<td>P1,P4</td>
<td>0.45</td>
</tr>
<tr>
<td>P2,P3</td>
<td>1.35</td>
</tr>
<tr>
<td>P2,P4</td>
<td>0.90</td>
</tr>
<tr>
<td>P3,P4</td>
<td>0.45</td>
</tr>
</tbody>
</table>

• we have an equal goodness measure for merging ((P1,P2), (P2,P3), (P3,P1))
Rock – Exercise 1

• Now, we start the hierarchical algorithm by merging, say P1 and P2.

• A new cluster (let’s call it C(P1,P2)) is formed.

• It should be noted that for some other hierarchical clustering techniques, we will not start the clustering process by merging P1 and P2, since Sim(P1,P2) = 0.4, which is not the highest. But, ROCK uses the number of links as the similarity measure rather than distance.
Now, after merging P1 and P2, we have only three clusters. The following table shows the number of common neighbors for these clusters:

<table>
<thead>
<tr>
<th></th>
<th>C(P1,P2)</th>
<th>P3</th>
<th>P4</th>
</tr>
</thead>
<tbody>
<tr>
<td>C(P1,P2)</td>
<td>-</td>
<td>3+3</td>
<td>2+1</td>
</tr>
<tr>
<td>P3</td>
<td>-</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>P4</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Then we can obtain the following goodness measures for all adjacent clusters:

<table>
<thead>
<tr>
<th>Pair</th>
<th>Goodness measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>C(P1,P2),P3</td>
<td>1.31</td>
</tr>
<tr>
<td>C(P1,P2),P4</td>
<td>0.66</td>
</tr>
<tr>
<td>P3,P4</td>
<td>0.45</td>
</tr>
</tbody>
</table>
Rock – Exercise 1

• Since the number of required clusters is 2, then we finish the clustering algorithm by merging $C(P_1,P_2)$ and $P_3$, obtaining a new cluster $C(P_1,P_2,P_3)$ which contains $\{P_1,P_2,P_3\}$ leaving $P_4$ alone in a separate cluster.
Rock – Exercise 2

• Given the following similarity matrix find the clustering result knowing that the similarity threshold = 0.4, and number of required cluster is 2.

<table>
<thead>
<tr>
<th></th>
<th>p1</th>
<th>p2</th>
<th>p3</th>
<th>p4</th>
<th>p5</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>1</td>
<td>0.7</td>
<td>0.2</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>p2</td>
<td>1</td>
<td>0.6</td>
<td>0.8</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>p3</td>
<td></td>
<td>1</td>
<td>0.5</td>
<td>0.4</td>
<td></td>
</tr>
<tr>
<td>p4</td>
<td></td>
<td></td>
<td>1</td>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td>p5</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
# Rock – Exercise 2 – Solution

$$
\begin{array}{cccccc}
\text{p1} & \text{p2} & \text{p3} & \text{p4} & \text{p5} \\
\text{p1} & 1 & 0.7 & 0.2 & 0.5 & 0.5 \\
\text{p2} & 1 & 0.6 & 0.8 & 0.1 & 1 \\
\text{p3} & 1 & 0.5 & 0.4 & 1 & 1 \\
\text{p4} & 1 & 0.3 & 1 & 0.5 \\
\text{p5} & 1 & 1 & 1 & 0.7 & 1 \\
\end{array}
$$

$$
\begin{array}{cccccc}
\text{p1} & \text{p2} & \text{p3} & \text{p4} & \text{p5} \\
\text{p1} & 1 & 1 & 0 & 1 & 1 \\
\text{p2} & 1 & 1 & 1 & 1 & 0 \\
\text{p3} & 0 & 1 & 1 & 1 & 1 \\
\text{p4} & 1 & 1 & 1 & 1 & 0 \\
\text{p5} & 1 & 0 & 1 & 0 & 1 \\
\end{array}
$$
## Rock – Exercise 2 – Solution

<table>
<thead>
<tr>
<th></th>
<th>p1</th>
<th>p2</th>
<th>p3</th>
<th>p4</th>
<th>p5</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>p2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>p3</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>p4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>p5</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>p1</th>
<th>p2</th>
<th>p3</th>
<th>p4</th>
<th>p5</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>-</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>p2</td>
<td>-</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>p3</td>
<td>-</td>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>p4</td>
<td>-</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-</td>
</tr>
</tbody>
</table>
Rock – Exercise 2 – Solution

• \( f(\theta) = \frac{1 - \theta}{1 + \theta} = \frac{1 - 0.4}{1 + 0.4} = 0.43 \)
• \( 1 + 2f(\theta) = 1.86 \)

\[
g(P_i, P_j) = \frac{\text{link}[P_i, P_j]}{(n + m)^{1 + 2f(\theta)} - n^{1 + 2f(\theta)} - m^{1 + 2f(\theta)}}
\]
Rock – Exercise 2 – Solution

• \( f(\theta) = \frac{1-\theta}{1+\theta} = \frac{1-0.4}{1+0.4} = 0.43 \)
• \( 1 + 2f(\theta) = 1.86 \)

\[
g(P_i, P_j) = \frac{\text{link}[P_i, P_j]}{(n + m)^{1+2f(\theta)} - n^{1+2f(\theta)} - m^{1+2f(\theta)}}
\]

<table>
<thead>
<tr>
<th></th>
<th>p1</th>
<th>p2</th>
<th>p3</th>
<th>p4</th>
<th>p5</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>-</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>p2</td>
<td>-</td>
<td>-</td>
<td>3</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>p3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>p4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td>p5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>p1</th>
<th>p2p4</th>
<th>p3</th>
<th>p5</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>-</td>
<td>6</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>p2p4</td>
<td>-</td>
<td>6</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>p3</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>p5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>p1</th>
<th>p2p4</th>
<th>p3</th>
<th>p5</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>-</td>
<td>1.94</td>
<td>1.84</td>
<td>1.22</td>
</tr>
<tr>
<td>p2p4</td>
<td>-</td>
<td>1.94</td>
<td>1.29</td>
<td></td>
</tr>
<tr>
<td>p3</td>
<td>-</td>
<td>-</td>
<td>1.22</td>
<td></td>
</tr>
<tr>
<td>p5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

• Final Clusters: p1234 p5
Clope Exercise 1

Split1:
• 4 transactions: abc, abc, ab, a
  • a: 4, b:3, c: 2 -> sol: S=9; W=3; H=9/3=3; H/W=1
• 3 transactions: def, de, de
  • d: 3, e:3, f: 1 -> sol: S=7; W=3; H=7/3=2.33; H/W=0.77

Split2:
• 2 transactions: abcd, ab
  • a: 2, b:2, c: 1, d:1 -> sol: S=6; W=4; H=6/4=1.5; H/W=0.37
• 2 transactions: ec, ec
  • e:2, c: 2 -> sol: S=4; W=2; H=4/2=2; H/W=1

Split1 is the best clustering considering r=2
Profit(Split1) = (9/3^2 * 4 + 7/3^2 * 3) / 7 = 0.90
Profit(Split2) = (6/4^2 * 2 + 4/2^2 * 2) / 4 = 0.16