Cluster Analysis in Data Mining

Part I

Andrew Kusiak
Intelligent Systems Laboratory
2139 Seamans Center
The University of Iowa
Iowa City, IA 52242 - 1527
andrew-kusiak@uiowa.edu
http://www.icaen.uiowa.edu/~ankusiak
Tel. 319-335 9934

Cluster Analysis

Decomposition
Aggregation
Grouping

Models

- Matrix formulation
- Mathematical programming formulation
- Graph formulation

Cluster Representation 1

Mutually exclusive regions

Cluster Representation 2

Overlapping regions

Cluster Representation 3

Hierarchy
Cluster Representation 4

Clusters

<table>
<thead>
<tr>
<th>Clusters</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>.6</td>
<td>.3</td>
<td>.1</td>
</tr>
<tr>
<td>B</td>
<td>.5</td>
<td>.2</td>
<td>.3</td>
</tr>
<tr>
<td>C</td>
<td>.6</td>
<td>.3</td>
<td>.1</td>
</tr>
<tr>
<td>D</td>
<td>.3</td>
<td>.6</td>
<td>.1</td>
</tr>
<tr>
<td>Objects</td>
<td>.2</td>
<td>.1</td>
<td>.7</td>
</tr>
<tr>
<td>F</td>
<td>.1</td>
<td>.2</td>
<td>.7</td>
</tr>
<tr>
<td>G</td>
<td>.2</td>
<td>.6</td>
<td>.2</td>
</tr>
<tr>
<td>H</td>
<td>.3</td>
<td>.6</td>
<td>.1</td>
</tr>
</tbody>
</table>

Fuzzy clusters

Two types of data

- Object with decisions
- Object without decisions

Clustering Data with Decisions

Method 1

- Ignore decisions
- Treat the objects as data without decisions

Method 2

- Group objects according to the decisions
- Treat each group of objects with decisions as a separate data set

Matrix Formulation

Feature

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Red</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Mutually Separable Clusters

GO-1

<table>
<thead>
<tr>
<th>Feature</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>GO-1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>GO-2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Partially Separable Clusters

GO-1

<table>
<thead>
<tr>
<th>Feature</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>GO-1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

GO-2

<table>
<thead>
<tr>
<th>Feature</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>GO-2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Overlapping feature
Solving the Clustering Problem: Binary Matrix Formulation

- Similarity coefficient methods
- Sorting based algorithms
- Bond energy algorithm
- Cost-based method
- Cluster identification algorithm
- Extended cluster identification algorithm

Clustering Algorithms

Desired Properties

- Low computational time complexity
- Structure enhancement
- Enhancement of human interaction and simplification of interface with software
- Generalizability

Similarity Coefficient Method

\[ s_{ij} = \frac{\sum_{k=1}^{n} \delta_1(a_{ik}, a_{jk})}{\sum_{k=1}^{n} \delta_2(a_{ik}, a_{jk})} \]

where \( \delta_1(a_{ik}, a_{jk}) = \begin{cases} 
1 & \text{if } a_{ik} = a_{jk} \\
0 & \text{otherwise}
\end{cases} \)

\( \delta_2(a_{ik}, a_{jk}) = \begin{cases} 
0 & \text{if } a_{ik} = a_{jk} \\
1 & \text{otherwise}
\end{cases} \)

Generalization

\( n = \) the number of features

Example

Consider

<table>
<thead>
<tr>
<th>GO-1</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Yes</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Yes</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GO-2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

\( s_{12} = s_{34} = \frac{1}{3} = 0.33 \)

\( s_{13} = \frac{1}{4} = 0.25 \)

\( s_{14} = s_{24} = s_{23} = \frac{0}{5} = 0 \)

\( s_{15} = s_{25} = \frac{0}{5} = 0 \)

\( \delta_1(a_{ik}, a_{jk}) = \begin{cases} 
1 & \text{if } a_{ik} = a_{jk} \\
0 & \text{otherwise}
\end{cases} \)

\( \delta_2(a_{ik}, a_{jk}) = \begin{cases} 
0 & \text{if } a_{ik} = a_{jk} \\
1 & \text{otherwise}
\end{cases} \)
Decomposition Problem

Decompose an object-feature incidence matrix into mutually separable submatrices (groups of objects and groups of features) with the minimum number of overlapping objects (or features) subject to the following constraints:

Constraint C1: Empty groups of objects or features are not allowed.
Constraint C2: The number of objects in a group does not exceed an upper limit, $b$ (or alternatively, the number of features in a group does not exceed, $d$).

Cluster Identification Algorithm

Step 0. Set iteration number $k = 1$.
Step 1. Select row $i$ of incidence matrix $[a_{ij}]^{(k)}$ and draw a horizontal line $h_i$ through it ($[a_{ij}]^{(k)}$ is read: matrix $[a_{ij}]$ at iteration $k$).
Step 2. For each entry of * crossed by the horizontal line $h_i$ draw a vertical line $v_j$.
Step 3. For each * entry crossed-once by the vertical line $v_j$ draw a horizontal line $h_k$.
Step 4. Repeat steps 2 and 3 until there are no more crossed-once entries of * in $[a_{ij}]^{(k)}$. All crossed-twice entries * in $[a_{ij}]^{(k)}$ form row cluster $RC-k$ and column cluster $CC-k$.
Step 5. Transform the incidence matrix $[a_{ij}]^{(k)}$ into $[a_{ij}]^{(k+1)}$ by removing rows and columns corresponding to the horizontal and vertical lines drawn in steps 1 through 4.
Step 6. If matrix $[a_{ij}]^{(k+1)} = 0$ (where 0 denotes a matrix with all empty elements), stop; otherwise set $k = k + 1$ and go to step 1.

Example 1

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>*</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>*</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>*</td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>
Next Step
Delete all double-crossed elements

Resultant Matrix

Final decomposition result

Three types of matrices
(a) decomposable matrix
(b) non-decomposable matrix with overlapping features
(c) non-decomposable matrix with objects
Extended CI Algorithm

Step 0. Set iteration number \( k = 1 \).
Step 1. Select those objects (rows of matrix \([aij]\)) \((k)\) that based on the user's expertise, are potential candidates for inclusion in machine cell \( GO-k \). Draw a horizontal line \( h_i \) through each row of matrix \([aij]^{(k)}\) corresponding to these objects. In the absence of the user's expertise any machine can be selected.

Step 2. For each column in \([aij]^{(k)}\) corresponding to entry 1, single-crossed by any of the horizontal lines \( h_i \), draw a vertical line \( v_j \).

Step 3. For each row in \([aij]^{(k)}\) corresponding to the entry 1, single-crossed by the vertical line \( v_j \), drawn in Step 2, draw a horizontal line \( h_i \). Based on the objects corresponding to all the horizontal lines drawn in Step 1 and Step 3, a temporary machine cell \( GO-k \), is formed.

Extended CI Algorithm

If the user's expertise indicates that some of the objects cannot be included in the temporary group \( GO-k \), erase the corresponding horizontal lines in the matrix \([aij]^{(k)}\).

Removal of these horizontal lines results in group \( GO'-k \).

Delete from matrix \([aij]^{(k)}\) features (columns) that are contained in at least one of the objects already included in \( GO-k \).

Place these features on a separate list.

Draw a vertical line \( v_j \) through each single-crossed entry 1 in \([aij]^{(k)}\) which does not involve any other objects than those included in \( GO-k \).

Step 4. For all the double-crossed entries 1 in \([aij]^{(k)}\), form a cluster \( GO-k \) and a group \( GF-k \).

Step 5. Transform the incidence matrix \([aij]^{(k)}\) into \([aij]^{(k+1)}\) by removing all the rows and columns included in \( GO-k \) and \( GF-k \), respectively.

Step 6. If matrix \([aij]^{(k+1)}\) = 0 (where 0 denotes a matrix with all elements equal to zero), stop; otherwise set \( k = k + 1 \) and go to step 1.

Example

Extended CI Algorithm

Constraints:
Max \(|GO| = 4

Objects 1 and 4 in the cluster

Step 2. For columns 1, 2, 3, 6, and 7 crossed by the horizontal lines \( h_1 \) and \( h_4 \), five vertical lines, \( v_1, v_2, v_3, v_6, \) and \( v_7 \) are drawn.
Step 3. Three horizontal lines, $h_2$, $h_6$, and $h_7$ are drawn through rows 2, 5, and 7 corresponding of the single-crossed elements 1.

<table>
<thead>
<tr>
<th>Feature number</th>
<th>v1</th>
<th>v2</th>
<th>v3</th>
<th>v4</th>
<th>v5</th>
<th>v6</th>
<th>v7</th>
<th>v8</th>
<th>v9</th>
<th>v10</th>
<th>v11</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

• A temporary machine cell GO'-1 with objects {1, 2, 4, 6, 7} is formed.
• objects 2 and 6 are excluded from in GO'-1 as they include less double-crossed (committed) elements than object 7. The horizontal lines $h_2$ and $h_6$ are erased.

Step 4. The double-crossed entries 1 of matrix indicate:
• Object cluster GO-1 = {1, 4, 7} and
• Feature cluster GF-1 = {2, 3, 6, 7}

Step 5. Matrix is transformed into the matrix below.

\[
\begin{bmatrix}
2 & 1 & 1 \\
3 & 1 & 1 \\
5 & 1 & 1 \\
6 & 1 & 1 \\
\end{bmatrix}
\]

Step 6. Set $k = k + 1 = 2$ and go to Step 1.
The second iteration ($k = 2$) results in:
• Object cluster GO-2 = {2, 3, 5, 6} and
• Feature cluster GF-2 = {5, 8, 10, 11}

The final result

\[
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
3 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
4 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
5 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
6 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\end{bmatrix}
\]

Mathematical Programming Formulation

dij: distance measure between objects $i$ and $j$

Axioms
• Reflexivity $d_{ii} = 0$
• Symmetry $d_{ij} = d_{ji}$
• Triangular inequality $d_{ij} = d_{ip} + d_{pq}$
1. Minkowski distance measure
\[ d_{ij} = \left( \sum_{k=1}^{m} |a_{ki} - a_{kj}|^r \right)^{1/r} \]
where \( r \) = a positive integer
\( m \) = the number of objects
Two special cases of the above measure are widely used:
- Absolute metric (for \( r = 1 \))
- Euclidean metric (for \( r = 2 \))

2. Weighted Minkowski distance measure
\[ d_{ij} = \left( \sum_{k=1}^{m} w_k |a_{ki} - a_{kj}|^r \right)^{1/r} \]
There are two special cases:
- Weighted absolute metric (for \( r = 1 \))
- Weighted Euclidean metric (for \( r = 2 \))

3. Hamming distance
\[ d_{ij} = \sum_{k=1}^{m} d(a_{ki}, a_{kj}) \]
where \( d(a_{ki}, a_{kj}) = \{ \)
\( 1 \) if \( a_{ki} \neq a_{kj} \)
\( 0 \) otherwise

Notation
\( m \) = number of objects
\( n \) = number of features
\( p \) = number of clusters
\( x_{ij} \) = \{ for feature \( i \) belongs to cluster \( j \)
\( 0 \) otherwise
\( d_{ij} \) = distance between objects \( i \) and \( j \)

The \( p \)-Median Model

\[ \min \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} \]
subject to:
\[ \sum_{j=1}^{n} x_{ij} = 1 \text{ for all } i = 1, \ldots, n \]
\[ \sum_{j=1}^{n} x_{jj} = p \]
\[ x_{ij} \leq x_{jj} \text{ for all } i = 1, \ldots, n \text{, } j = 1, \ldots, n \]
\[ x_{ij} = 0, 1 \text{ for all } i = 1, \ldots, n \text{, } j = 1, \ldots, n \]

Example

Hamming Distance
\[ d_{ij} = \sum_{k=1}^{m} d(a_{ki}, a_{kj}) \]
Two objects
\[ Q_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \]
\[ Q_k = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \]
Hamming distance
\( d_{15} = 1 + 0 + 1 + 1 + 0 + 1 = 4 \)

Decision variable
\( x_{ij} \)
The University of Iowa
Intelligent Systems Laboratory

Example

Given

\[ p = 2 \]

\[ \sum_{i=1}^{m} d_{i} = \sum_{k=1}^{n} d(ak_i, ak_j) \]

Feature number

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>0</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Object number

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Does the solution to the \( p \)-median problem indicate overlapping objects (features)?

Solution: \( x_{11} = 1, x_{31} = 1 \)

\( x_{24} = 1, x_{44} = 1, x_{54} = 1 \)

Based on the definition of \( x_{ij} \), two groups of features are formed:

- \( GF-1 = \{1, 3\} \)
- \( GF-2 = \{2, 4, 5\} \)
- \( GO-1 = \{2, 4\} \)
- \( GO-2 = \{1, 3\} \)

Solution: \( x_{13}=1, x_{23}=1 \)

\( x_{35}=1, x_{45}=1, x_{55}=1 \)
Cluster Analysis in Data Mining
Part II

Two generic modes of learning:
- **Supervised learning**
- **Unsupervised learning (clustering)**

Supervised Learning

**Goal:**
Determine class boundaries (design a classifier)

Supervised Learning: Classifier Design

- Types of classifiers:
  - linear
  - piecewise linear
  - nonlinear
  - nearest neighbor(s)
  - ...

- Implementation:
  - decision trees
  - regression analysis
  - neural networks
  - ...

Unsupervised Learning with Clustering Methods

(Kaufman, Rousseeuw, 1990)
... cluster analysis is the art of finding groups in data.

Unsupervised Learning: Main Classes of Clustering Methods

- Hierarchical clustering
- Model-based clustering
Hierarchical Clustering

Aggregate and Divisive Methods

**Aggregate methods**
Treat each example (object) as a separate cluster and aggregate them successively based upon its relationship with other objects. The aggregation process is to optimize a performance measure.

**Divisive methods**
Work in an opposite direction by beginning with a single cluster and splitting it into smaller clusters.

Distance Function: Definition

Objects x and y

1. \( ||x - y|| = 0 \) if and only if \( x = y \) (the objects are identical)
2. \( ||x - y|| \geq 0 \) (Nonnegativity)
3. \( ||x - y|| = ||y - x|| \) (Symmetry)
4. \( ||x - z|| \leq ||x - y|| + ||y - z|| \) (Triangular inequality)

Distance Functions (Proximity of Data)

- **Minkowski distance measure**
  \[ ||x - y|| = \left( \sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p} \]
  - \( p = 1 \): Absolute
  - \( p = 2 \): Euclidean
  - \( p = \infty \): Tchebyshev

Mahalanobis Distance

- Represents relationships between individual features
- Scaling is possible

\[ ||x - y|| = (x - y)^T M^{-1} (x - y) \]

Mahalanobis Distance

For matrix \( M \) with \( \sigma_i \) as diagonal elements, the inverse of \( M \)

\[ M^{-1} = \begin{bmatrix} \sigma_1^-2 & 0 & \cdots & 0 \\ 0 & \sigma_2^-2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n^-2 \end{bmatrix} \]

\[ ||x - y|| = \sqrt{\sum_{i=1}^{n} \frac{(x_i - y_i)^2}{\sigma_i^2}} \]

Bhattacharyya distance

For \( M = I \): Euclidean distance
Mahalanobis Distance and Data

Hierarchical Clustering: An Aggregation Approach

Given: Data set and a distance function.
1. Begin with “N” clusters by assigning each object to a separate cluster.
2. Proceed with this initial configuration of the clusters and merge the closest clusters. In other words, if S and T are the two closest clusters, form a single cluster \{S, T\} and reduce the number of clusters by one.
3. Repeat Step 2 until the desired clusters have been reached.
Result: Clusters of data (partitions).

Hierarchical Clustering: Single Linkage Approach

S, T = two clusters

Similarity between S and T is computed based on the minimum distance between the objects belonging to the corresponding clusters.

\[ ||T - S|| = \min_{x \in T, y \in S} ||x - y|| \]

Hierarchical Clustering: Complete Linkage Approach

Maximum distance between the objects in the analyzed clusters is considered.

\[ ||T - S|| = \max_{x \in T, y \in S} ||x - y|| \]

Hierarchical Clustering: Average Linkage Approach

Two clusters are formed based on their average distance between the objects in the clusters.

\[ ||T - S|| = \frac{1}{\text{card}(T) \cdot \text{card}(S)} \sum_{x \in T} \sum_{y \in S} ||x - y|| \]
Hierarchical Clustering: Computational Aspects

- Number of clusters
- Storage of distance matrix
- Non-iterative optimization

Model-based Clustering

- \( p \)-median model
- Generalized \( p \)-median model

The \( p \)-median Model

\[
\begin{align*}
\text{min} & \quad \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} \\
\text{s.t.} & \quad \sum_{j=1}^{n} x_{ij} = 1 \text{ for all } i = 1, \ldots, n \\
& \quad \sum_{j=1}^{n} x_{jj} = p \\
& \quad x_{ij} \leq x_{jj} \text{ for all } i = 1, \ldots, n, \quad j = 1, \ldots, n \\
& \quad x_{ij} = 0,1 \text{ for all } i = 1, \ldots, n, \quad j = 1, \ldots, n
\end{align*}
\]