Clustering

Milos Hauskrecht
milos@cs.pitt.edu
5329 Sennott Square
Clustering

Groups together “similar” instances in the data sample

**Basic clustering problem:**

- distribute data into $k$ different groups such that data points similar to each other are in the same group
- Similarity between data points is defined in terms of some distance metric (can be chosen)

Clustering is useful for:

- **Similarity/Dissimilarity analysis**
  Analyze what data points in the sample are close to each other
- **Dimensionality reduction**
  High dimensional data replaced with a group (cluster) label
Clustering example

• We see data points and want to partition them into groups
• Which data points belong together?
Clustering example

- We see data points and want to partition them into the groups
- Which data points belong together?
Clustering example

- We see data points and want to partition them into the groups
- Requires a distance measure to tell us what points are close to each other and are in the same group

Euclidean distance
Clustering example

- A set of patient cases
- We want to partition them into groups based on similarities

<table>
<thead>
<tr>
<th>Patient #</th>
<th>Age</th>
<th>Sex</th>
<th>Heart Rate</th>
<th>Blood pressure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patient 1</td>
<td>55</td>
<td>M</td>
<td>85</td>
<td>125/80</td>
</tr>
<tr>
<td>Patient 2</td>
<td>62</td>
<td>M</td>
<td>87</td>
<td>130/85</td>
</tr>
<tr>
<td>Patient 3</td>
<td>67</td>
<td>F</td>
<td>80</td>
<td>126/86</td>
</tr>
<tr>
<td>Patient 4</td>
<td>65</td>
<td>F</td>
<td>90</td>
<td>130/90</td>
</tr>
<tr>
<td>Patient 5</td>
<td>70</td>
<td>M</td>
<td>84</td>
<td>135/85</td>
</tr>
</tbody>
</table>
Clustering example

- A set of patient cases
- We want to partition them into the groups based on similarities

<table>
<thead>
<tr>
<th>Patient #</th>
<th>Age</th>
<th>Sex</th>
<th>Heart Rate</th>
<th>Blood pressure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patient 1</td>
<td>55</td>
<td>M</td>
<td>85</td>
<td>125/80</td>
</tr>
<tr>
<td>Patient 2</td>
<td>62</td>
<td>M</td>
<td>87</td>
<td>130/85</td>
</tr>
<tr>
<td>Patient 3</td>
<td>67</td>
<td>F</td>
<td>80</td>
<td>126/86</td>
</tr>
<tr>
<td>Patient 4</td>
<td>65</td>
<td>F</td>
<td>90</td>
<td>130/90</td>
</tr>
<tr>
<td>Patient 5</td>
<td>70</td>
<td>M</td>
<td>84</td>
<td>135/85</td>
</tr>
</tbody>
</table>

How to design the distance metric to quantify similarities?
Clustering example. Distance measures.

In general, one can choose an arbitrary distance measure.

Properties of distance metrics:
Assume 2 data entries $a$, $b$

**Positiveness:** $d(a, b) \geq 0$

**Symmetry:** $d(a, b) = d(b, a)$

**Identity:** $d(a, a) = 0$
Distance measures.

Assume pure real-valued data-points:

12  34.5  78.5  89.2  19.2
23.5  41.4  66.3  78.8  8.9
33.6  36.7  78.3  90.3  21.4
17.2  30.1  71.6  88.5  12.5
...

What distance metric to use?
Distance measures

Assume pure real-valued data-points:

\[
\begin{array}{cccccc}
12 & 34.5 & 78.5 & 89.2 & 19.2 \\
23.5 & 41.4 & 66.3 & 78.8 & 8.9 \\
33.6 & 36.7 & 78.3 & 90.3 & 21.4 \\
17.2 & 30.1 & 71.6 & 88.5 & 12.5 \\
\ldots
\end{array}
\]

What distance metric to use?

**Euclidian:** works for an arbitrary k-dimensional space

\[
d(a, b) = \sqrt{\sum_{i=1}^{k} (a_i - b_i)^2}
\]
Distance measures

Assume pure real-valued data-points:

```
12  34.5  78.5  89.2  19.2
23.5 41.4  66.3  78.8  8.9
33.6 36.7  78.3  90.3  21.4
17.2 30.1  71.6  88.5  12.5
```

What distance metric to use?

**Squared Euclidian:** works for an arbitrary k-dimensional space

\[
d^2 (a, b) = \sum_{i=1}^{k} (a_i - b_i)^2
\]
Distance measures.

Assume pure real-valued data-points:

\[
\begin{array}{cccccc}
12 & 34.5 & 78.5 & 89.2 & 19.2 \\
23.5 & 41.4 & 66.3 & 78.8 & 8.9 \\
33.6 & 36.7 & 78.3 & 90.3 & 21.4 \\
17.2 & 30.1 & 71.6 & 88.5 & 12.5 \\
\end{array}
\]

**Manhattan distance:**

works for an arbitrary k-dimensional space

\[
d(a, b) = \sum_{i=1}^{k} |a_i - b_i|
\]

Etc. ..
Distance measures

Generalized distance metric:

\[ d^2 (\mathbf{a}, \mathbf{b}) = (\mathbf{a} - \mathbf{b}) \Gamma^{-1} (\mathbf{a} - \mathbf{b})^T \]

\( \Gamma \) semi-definite positive matrix
\( \Gamma^{-1} \) is a matrix that weights attributes proportionally to their importance. Different weights lead to a different distance metric.

If \( \Gamma = I \) we get squared Euclidean

\( \Gamma = \Sigma \) (covariance matrix) – we get the Mahalanobis distance that takes into account correlations among attributes
Distance measures.

Assume pure binary values data:

```
0 1 1 0 1
1 0 1 0 1
0 1 1 0 1
1 1 1 1 1
...
```

What distance metric to use?
Distance measures.

Assume pure binary values data:

```
0 1 1 0 1
1 0 1 0 1
0 1 1 0 1
1 1 1 1 1
```

...  

What distance metric to use?  

**Hamming distance:** The number of bits that need to be changed to make the entries the same  

How about Euclidean distance?
Distance measures.

Assume pure categorical data:

```
0 1 1 0 0
1 0 3 0 1
2 1 1 0 2
1 1 1 1 2
...```

What distance metric to use?

**Hamming distance:** The number of values that need to be changed to make them the same
Distance measures.

Combination of real-valued and categorical attributes

<table>
<thead>
<tr>
<th>Patient #</th>
<th>Age</th>
<th>Sex</th>
<th>Heart Rate</th>
<th>Blood pressure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patient 1</td>
<td>55</td>
<td>M</td>
<td>85</td>
<td>125/80</td>
</tr>
<tr>
<td>Patient 2</td>
<td>62</td>
<td>M</td>
<td>87</td>
<td>130/85</td>
</tr>
<tr>
<td>Patient 3</td>
<td>67</td>
<td>F</td>
<td>80</td>
<td>126/86</td>
</tr>
<tr>
<td>Patient 4</td>
<td>65</td>
<td>F</td>
<td>90</td>
<td>130/90</td>
</tr>
<tr>
<td>Patient 5</td>
<td>70</td>
<td>M</td>
<td>84</td>
<td>135/85</td>
</tr>
</tbody>
</table>

What distance metric to use?
Distance measures.

Combination of real-valued and categorical attributes

<table>
<thead>
<tr>
<th>Patient #</th>
<th>Age</th>
<th>Sex</th>
<th>Heart Rate</th>
<th>Blood pressure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patient 1</td>
<td>55</td>
<td>M</td>
<td>85</td>
<td>125/80</td>
</tr>
<tr>
<td>Patient 2</td>
<td>62</td>
<td>M</td>
<td>87</td>
<td>130/85</td>
</tr>
<tr>
<td>Patient 3</td>
<td>67</td>
<td>F</td>
<td>80</td>
<td>126/86</td>
</tr>
<tr>
<td>Patient 4</td>
<td>65</td>
<td>F</td>
<td>90</td>
<td>130/90</td>
</tr>
<tr>
<td>Patient 5</td>
<td>70</td>
<td>M</td>
<td>84</td>
<td>135/85</td>
</tr>
</tbody>
</table>

What distance metric to use?

A weighted sum approach: e.g. a mix of Euclidian and Hamming distances for subsets of attributes
Clustering

Clustering is useful for:

- **Similarity/Dissimilarity analysis**
  Analyze what data points in the sample are close to each other

- **Dimensionality reduction**
  High dimensional data replaced with a group (cluster) label

- **Data reduction**: Replaces many datapoints with the point representing the group mean

Problems:

- Pick the correct similarity measure (problem specific)
- Choose the correct number of groups
  - Many clustering algorithms require us to provide the number of groups ahead of time
Clustering algorithms

- **K-means algorithm**
  - suitable only when data points have continuous values; groups are defined in terms of cluster centers (also called means). Refinement of the method to categorical values: K-medoids

- **Probabilistic methods (with EM)**
  - Latent variable models: class (cluster) is represented by a latent (hidden) variable value
  - Every point goes to the class with the highest posterior
  - Examples: mixture of Gaussians, Naïve Bayes with a hidden class

- **Hierarchical methods**
  - Agglomerative
  - Divisive
K-means

**K-Means algorithm:**

Initialize randomly $k$ values of means (centers)
Repeat two steps until no change in the means:

- Partition the data according to the current set of means (using the similarity measure)
- Move the means to the center of the data in the current partition

Stop when no change in the means

**Properties:**

- Minimizes the sum of squared center-point distances for all clusters
  
  $$
  \arg \min_s \sum_{i=1}^{k} \sum_{x_j \in S_i} \| x_j - u_i \|^2
  $$

- The algorithm always converges (to the local optima).
K-means algorithm

• **Properties:**
  – converges to centers minimizing the sum of squared center-point distances (still local optima)
  – The result is sensitive to the initial means’ values

• **Advantages:**
  – Simplicity
  – Generality – can work for more than one distance measure

• **Drawbacks:**
  – Can perform poorly with overlapping regions
  – Lack of robustness to outliers
  – Good for attributes (features) with continuous values
    • Allows us to compute cluster means
    • k-medoid algorithm used for discrete data
Probabilistic (EM-based) algorithms

- **Latent variable models**
  
  Examples: Naïve Bayes with hidden class  
  Mixture of Gaussians

- **Partitioning:**
  
  - the data point belongs to the class with the highest posterior

- **Advantages:**
  
  - Good performance on overlapping regions  
  - Robustness to outliers  
  - Data attributes can have different types of values

- **Drawbacks:**
  
  - EM is computationally expensive and can take time to converge  
  - Density model should be given in advance
Hierarchical clustering.

Uses an arbitrary similarity/dissimilarity measure.

Typical similarity measures $d(a,b)$:

Pure real-valued data-points:
- Euclidean, Manhattan, Minkowski distances

Pure binary values data:
- Hamming distance - Number of matching values
- the same as Euclidean

Pure categorical data:
- Number of matching values

Combination of real-valued and categorical attributes
- Weighted, or Euclidean
Hierarchical clustering

Approach:

• Compute dissimilarity matrix for all pairs of points
  – uses standard or other distance measures

• Construct clusters greedily:
  – Agglomerative approach
    • Merge pair of clusters in a bottom-up fashion, starting from singleton clusters
  – Divisive approach:
    • Splits clusters in top-down fashion, starting from one complete cluster

• Stop the greedy construction when some criterion is satisfied
  – E.g. fixed number of clusters
Cluster merging

• Construction of clusters through greedy agglomerative approach
  – Merge pair of clusters in a bottom-up fashion, starting from singleton clusters
  – Merge clusters based on cluster (or linkage) distances. Defined in terms of point distances. Examples:

Min distance
\[ d_{\text{min}}(C_i, C_j) = \min_{p \in C_i, q \in C_j} d(p, q) \]

Max distance
\[ d_{\text{max}}(C_i, C_j) = \max_{p \in C_i, q \in C_j} d(p, q) \]

Mean distance
\[ d_{\text{mean}}(C_i, C_j) = \left| d \left( \frac{1}{|C_i|} \sum_i p_i ; \frac{1}{|C_j|} \sum_j q_j \right) \right| \]
Hierarchical clustering example
Hierarchical clustering example

- dendogram
Hierarchical clustering

- **Advantage:**
  - Smaller computational cost; avoids scanning all possible clusterings

- **Disadvantage:**
  - Greedy choice fixes the order in which clusters are merged; cannot be repaired

- **Partial solution:**
  - Combine hierarchical clustering with iterative algorithms like k-means
Other clustering methods

• **Spectral clustering**
  – Uses similarity matrix

• **Multidimensional scaling**
  – techniques often used in data visualization for exploring similarities or dissimilarities in data.