CSCE 478/878 Lecture 8: Clustering

Stephen Scott

sscott@cse.unl.edu
Introduction

- If no label information is available, can still perform *unsupervised learning*
- Looking for structural information about instance space instead of label prediction function
- Approaches: density estimation, clustering, dimensionality reduction

*Clustering* algorithms group similar instances together based on a *similarity measure*
Outline

- Clustering background
  - Similarity/dissimilarity measures
- \(k\)-means clustering
- Hierarchical clustering
Clustering Background

- Goal: Place patterns into “sensible” clusters that reveal similarities and differences
- Definition of “sensible” depends on application

(a) How they bear young
(b) Existence of lungs
(c) Environment
(d) Both (a) & (b)
Types of clustering problems:

- **Hard (crisp):** partition data into non-overlapping clusters; each instance belongs in exactly one cluster
- **Fuzzy:** Each instance could be a member of multiple clusters, with a real-valued function indicating the degree of membership
- **Hierarchical:** partition instances into numerous small clusters, then group the clusters into larger ones, and so on (applicable to phylogeny)
  - End up with a tree with instances at leaves
Dissimilarity measure: Weighted $L_p$ norm:

$$L_p(x, y) = \left( \sum_{i=1}^{n} w_i |x_i - y_i|^p \right)^{1/p}$$

Special cases include weighted Euclidian distance ($p = 2$), weighted Manhattan distance

$$L_1(x, y) = \sum_{i=1}^{n} w_i |x_i - y_i| ,$$

and weighted $L_\infty$ norm

$$L_\infty(x, y) = \max_{1 \leq i \leq n} \{ w_i |x_i - y_i| \}$$

Similarity measure: Dot product between two vectors (kernel)
If attributes come from \( \{0, \ldots, k - 1\} \), can use measures for real-valued attributes, plus:

- **Hamming distance**: DM measuring number of places where \( x \) and \( y \) differ
- **Tanimoto measure**: SM measuring number of places where \( x \) and \( y \) are same, divided by total number of places
- Ignore places \( i \) where \( x_i = y_i = 0 \)
  - Useful for ordinal features where \( x_i \) is degree to which \( x \) possesses \( i \)th feature
Might want to measure proximity of point \( x \) to existing cluster \( C \)

Can measure proximity \( \alpha \) by using all points of \( C \) or by using a representative of \( C \)

If all points of \( C \) used, common choices:

\[
\alpha_{ps}^{max}(x, C) = \max_{y \in C} \{ \alpha(x, y) \}
\]

\[
\alpha_{ps}^{min}(x, C) = \min_{y \in C} \{ \alpha(x, y) \}
\]

\[
\alpha_{ps}^{avg}(x, C) = \frac{1}{|C|} \sum_{y \in C} \alpha(x, y)
\]

where \( \alpha(x, y) \) is any measure between \( x \) and \( y \)
Alternative: Measure distance between point $x$ and a representative of the cluster $C$

- **Mean vector** $m_p = \frac{1}{|C|} \sum_{y \in C} y$

- **Mean center** $m_c \in C$:
  \[
  \sum_{y \in C} d(m_c, y) \leq \sum_{y \in C} d(z, y) \quad \forall z \in C,
  \]
  where $d(\cdot, \cdot)$ is DM (if SM used, reverse ineq.)

- **Median center**: For each point $y \in C$, find median dissimilarity from $y$ to all other points of $C$, then take min; so $m_{med} \in C$ is defined as
  \[
  \text{med}_{y \in C} \{d(m_{med}, y)\} \leq \text{med}_{y \in C} \{d(z, y)\} \quad \forall z \in C
  \]

Now can measure proximity between $C$’s representative and $x$ with standard measures
Given sets of instances $C_i$ and $C_j$ and proximity measure $\alpha(\cdot, \cdot)$

- **Max:** $\alpha_{\text{max}}^{ss}(C_i, C_j) = \max_{x \in C_i, y \in C_j} \{\alpha(x, y)\}$
- **Min:** $\alpha_{\text{min}}^{ss}(C_i, C_j) = \min_{x \in C_i, y \in C_j} \{\alpha(x, y)\}$
- **Average:** $\alpha_{\text{avg}}^{ss}(C_i, C_j) = \frac{1}{|C_i| \cdot |C_j|} \sum_{x \in C_i} \sum_{y \in C_j} \alpha(x, y)$
- **Representative (mean):** $\alpha_{\text{mean}}^{ss}(C_i, C_j) = \alpha(m_{C_i}, m_{C_j})$, where $m_{C_i}$ and $m_{C_j}$ are the means of $C_i$ and $C_j$, respectively.
$k$-Means Clustering

- Very popular clustering algorithm
- Represents cluster $i$ (out of $k$ total) by specifying its *representative* $m_i$ (not necessarily part of the original set of instances $\mathcal{X}$)
- Each instance $x \in \mathcal{X}$ is assigned to the cluster with nearest representative
- Goal is to find a set of $k$ representatives such that sum of distances between instances and their representatives is minimized
  - NP-hard in general
- Will use an algorithm that alternates between determining representatives and assigning clusters until convergence (in the style of the EM algorithm)
Choose value for parameter $k$
Initialize $k$ arbitrary representatives $m_1, \ldots, m_k$
  - E.g., $k$ randomly selected instances from $X$
Repeat until representatives $m_1, \ldots, m_k$ don’t change
  1. For all $x \in X$
     - Assign $x$ to cluster $C_j$ such that $\|x - m_j\|$ (or other measure) is minimized
     - I.e., nearest representative
  2. For each $j \in \{1, \ldots, k\}$
    \[ m_j = \frac{1}{C_j} \sum_{y \in C_j} y \]
$k$-Means Clustering

Example with $k = 2$
Hierarchical Clustering

- Useful in capturing hierarchical relationships, e.g., evolutionary tree of biological sequences
- End result is a sequence (hierarchy) of clusterings
- Two types of algorithms:
  - Agglomerative: Repeatedly merge two clusters into one
  - Divisive: Repeatedly divide one cluster into two
Hierarchical Clustering

Definitions

- Let $C_t = \{C_1, \ldots, C_{m_t}\}$ be a level-$t$ clustering of $X = \{x_1, \ldots, x_N\}$, where $C_t$ meets definition of hard clustering.

- $C_t$ is *nested* in $C_{t'}$ (written $C_t \sqsubseteq C_{t'}$) if each cluster in $C_t$ is a subset of a cluster in $C_{t'}$ and at least one cluster in $C_t$ is a proper subset of some cluster in $C_{t'}$.

\[
C_1 = \{\{x_1, x_3\}, \{x_4\}, \{x_2, x_5\}\} \sqsubseteq \{\{x_1, x_3, x_4\}, \{x_2, x_5\}\}
\]

\[
C_1 \not\sqsubseteq \{\{x_1, x_4\}, \{x_3\}, \{x_2, x_5\}\}
\]
Hierarchical Clustering
Definitions (cont’d)

- Agglomerative algorithms start with $C_0 = \{\{x_1\}, \ldots, \{x_N\}\}$ and at each step $t$ merge two clusters into one, yielding $|C_{t+1}| = |C_t| - 1$ and $C_t \sqsubseteq C_{t+1}$

- At final step (step $N-1$) have hierarchy:

$$C_0 = \{\{x_1\}, \ldots, \{x_N\}\} \sqsubseteq C_1 \sqsubseteq \cdots \sqsubseteq C_{N-1} = \{\{x_1, \ldots, x_N\}\}$$

- Divisive algorithms start with $C_0 = \{\{x_1, \ldots, x_N\}\}$ and at each step $t$ split one cluster into two, yielding $|C_{t+1}| = |C_t| + 1$ and $C_{t+1} \sqsubseteq C_t$

- At step $N-1$ have hierarchy:

$$C_{N-1} = \{\{x_1\}, \ldots, \{x_N\}\} \sqsubseteq \cdots \sqsubseteq C_0 = \{\{x_1, \ldots, x_N\}\}$$
Hierarchical Clustering

Pseudocode

1. Initialize $C_0 = \{\{x_1\}, \ldots, \{x_N\}\}$, $t = 0$
2. For $t = 1$ to $N - 1$
   a. Find closest pair of clusters:
      $$(C_i, C_j) = \arg\min_{C_s, C_r \in C_{t-1}} \{d(C_s, C_r)\}$$
      where $r \neq s$
   b. $C_t = (C_{t-1} - \{C_i, C_j\}) \cup \{\{C_i \cup C_j\}\}$ and update representatives if necessary

If SM used, replace $\arg\min$ with $\arg\max$

Number of calls to $d(C_k, C_r)$ is $\Theta(N^3)$
Hierarchical Clustering

Example

\( x_1 = [1, 1]^T, x_2 = [2, 1]^T, x_3 = [5, 4]^T, x_4 = [6, 5]^T, x_5 = [6.5, 6]^T, \) \( DM = \text{Euclidian}/\alpha_{\text{ss}} \min \)

An \((N - t) \times (N - t)\) proximity matrix \( P_t \) gives the proximity between all pairs of clusters at level (iteration) \( t \)

\[
P_0 = \begin{bmatrix}
0 & 1 & 5 & 6.4 & 7.4 \\
1 & 0 & 4.2 & 5.7 & 6.7 \\
5 & 4.2 & 0 & 1.4 & 2.5 \\
6.4 & 5.7 & 1.4 & 0 & 1.1 \\
7.4 & 6.7 & 2.5 & 1.1 & 0
\end{bmatrix}
\]

Each iteration, find minimum off-diagonal element \((i, j)\) in \( P_{t-1} \), merge clusters \( i \) and \( j \), remove rows/columns \( i \) and \( j \) from \( P_{t-1} \), and add new row/column for new cluster to get \( P_t \)
Hierarchical Clustering

Pseudocode (cont’d)

A proximity dendogram is a tree that indicates hierarchy of clusterings, including the proximity between two clusters when they are merged.

Cutting the dendogram at any level yields a single clustering.