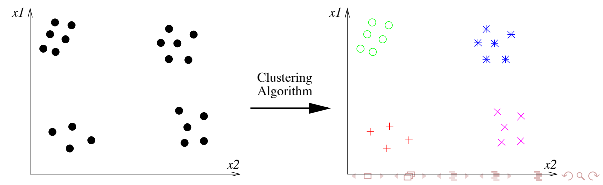
CSCE 478/878 Lecture 8: Clustering

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*

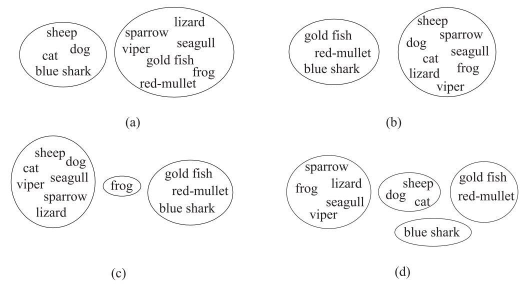


Outline

- 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)

Clustering Background (cont'd)

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

Clustering Background (Dis-)similarity Measures: Between Instances

Dissimilarity measure: Weighted L_p norm:

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

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

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

and weighted L_∞ *norm*

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

Similarity measure: Dot product between two vectors (kernel)

Clustering Background
(Dis-)similarity Measures: Between Instances (cont'd)

If attributes come from $\{0, \ldots, k-1\}$, can use measures for real-valued attributes, plus:

- *Hamming distance*: DM measuring number of places where $\mathbf{x}$ and $\mathbf{y}$ differ
- *Tanimoto measure*: SM measuring number of places where $\mathbf{x}$ and $\mathbf{y}$ 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 $\mathbf{x}$ possesses ith feature

Clustering Background
(Dis-)similarity Measures: Between Instance and Set

- Might want to measure proximity of point $\mathbf{x}$ to existing cluster C
- Can measure proximity α by using *all points* of C or by using a *representative* of C
- If all points of C used, common choices:

$$\alpha^{ps}_{max}(\mathbf{x}, C) = \max_{\mathbf{y} \in C} \{\alpha(\mathbf{x}, \mathbf{y})\}$$
$$\alpha^{ps}_{min}(\mathbf{x}, C) = \min_{\mathbf{y} \in C} \{\alpha(\mathbf{x}, \mathbf{y})\}$$
$$\alpha^{ps}_{avg}(\mathbf{x}, C) = \frac{1}{|C|} \sum_{\mathbf{y} \in C} \alpha(\mathbf{x}, \mathbf{y}) \ ,$$

where $\alpha(\mathbf{x}, \mathbf{y})$ is any measure between $\mathbf{x}$ and $\mathbf{y}$

Clustering Background
(Dis-)similarity Measures: Between Instance and Set (cont'd)

Alternative: Measure distance between point $\mathbf{x}$ and a *representative* of the cluster C

- *Mean vector* $\mathbf{m}_p = \frac{1}{|C|} \sum_{\mathbf{y} \in C} \mathbf{y}$
- *Mean center* $\mathbf{m}_c \in C$:

$$\sum_{\mathbf{y} \in C} d(\mathbf{m}_c, \mathbf{y}) \leq \sum_{\mathbf{y} \in C} d(\mathbf{z}, \mathbf{y}) \quad \forall \mathbf{z} \in C \ ,$$

  where $d(\cdot, \cdot)$ is DM (if SM used, reverse ineq.)
- *Median center*: For each point $\mathbf{y} \in C$, find median dissimilarity from $\mathbf{y}$ to all other points of C, then take min; so $\mathbf{m}_{med} \in C$ is defined as

$$\text{med}_{\mathbf{y} \in C} \{d(\mathbf{m}_{med}, \mathbf{y})\} \leq \text{med}_{\mathbf{y} \in C} \{d(\mathbf{z}, \mathbf{y})\} \quad \forall \mathbf{z} \in C$$

Now can measure proximity between C's representative and $\mathbf{x}$ with standard measures

Clustering Background
(Dis-)similarity Measures: Between Sets

Given sets of instances C_i and C_j and proximity measure $\alpha(\cdot, \cdot)$

- *Max*: $\alpha^{ss}_{max}(C_i, C_j) = \max_{\mathbf{x} \in C_i, \mathbf{y} \in C_j} \{\alpha(\mathbf{x}, \mathbf{y})\}$
- *Min*: $\alpha^{ss}_{min}(C_i, C_j) = \min_{\mathbf{x} \in C_i, \mathbf{y} \in C_j} \{\alpha(\mathbf{x}, \mathbf{y})\}$
- *Average*: $\alpha^{ss}_{avg}(C_i, C_j) = \frac{1}{|C_i|\,|C_j|} \sum_{\mathbf{x} \in C_i} \sum_{\mathbf{y} \in C_j} \alpha(\mathbf{x}, \mathbf{y})$
- *Representative (mean)*: $\alpha^{ss}_{mean}(C_i, C_j) = \alpha(\mathbf{m}_{C_i}, \mathbf{m}_{C_j})$,

k-Means Clustering

- Very popular clustering algorithm
- Represents cluster i (out of k total) by specifying its *representative* $\mathbf{m}_i$ (not necessarily part of the original set of instances $\mathcal{X}$)
- Each instance $\mathbf{x} \in \mathcal{X}$ 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)

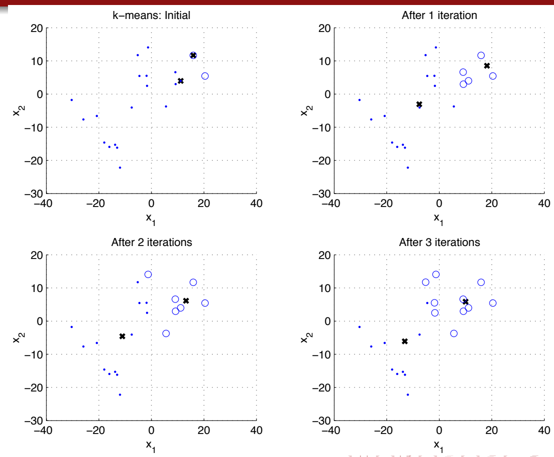
k-Means Clustering
Algorithm

- Choose value for parameter k
- Initialize k arbitrary representatives $\mathbf{m}_1, \ldots, \mathbf{m}_k$
  - E.g., k randomly selected instances from $\mathcal{X}$
- Repeat until representatives $\mathbf{m}_1, \ldots, \mathbf{m}_k$ don't change
  1. For all $\mathbf{x} \in \mathcal{X}$
     - Assign $\mathbf{x}$ to cluster C_j such that $\|\mathbf{x} - \mathbf{m}_j\|$ (or other measure) is minimized
     - I.e., nearest representative
  2. For each $j \in \{1, \ldots, k\}$

$$\mathbf{m}_j = \frac{1}{C_j} \sum_{\mathbf{y} \in C_j} \mathbf{y}$$

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 $\mathcal{C}_t = \{C_1, \ldots, C_{m_t}\}$ be a *level-t clustering* of $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, where $\mathcal{C}_t$ meets definition of hard clustering
- $\mathcal{C}_t$ is *nested* in $\mathcal{C}_{t'}$ (written $\mathcal{C}_t \sqsubset \mathcal{C}_{t'}$) if each cluster in $\mathcal{C}_t$ is a subset of a cluster in $\mathcal{C}_{t'}$ and at least one cluster in $\mathcal{C}_t$ is a proper subset of some cluster in $\mathcal{C}_{t'}$

$$\mathcal{C}_1 = \{\{\mathbf{x}_1, \mathbf{x}_3\}, \{\mathbf{x}_4\}, \{\mathbf{x}_2, \mathbf{x}_5\}\} \sqsubset \{\{\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_4\}, \{\mathbf{x}_2, \mathbf{x}_5\}\}$$
$$\mathcal{C}_1 \not\sqsubset \{\{\mathbf{x}_1, \mathbf{x}_4\}, \{\mathbf{x}_3\}, \{\mathbf{x}_2, \mathbf{x}_5\}\}$$

Hierarchical Clustering

Definitions (cont'd)

- Agglomerative algorithms start with $\mathcal{C}_0 = \{\{\mathbf{x}_1\}, \ldots, \{\mathbf{x}_N\}\}$ and at each step t merge two clusters into one, yielding $|\mathcal{C}_{t+1}| = |\mathcal{C}_t| - 1$ and $\mathcal{C}_t \sqsubset \mathcal{C}_{t+1}$
- At final step (step $N-1$) have hierarchy:

$$\mathcal{C}_0 = \{\{\mathbf{x}_1\}, \ldots, \{\mathbf{x}_N\}\} \sqsubset \mathcal{C}_1 \sqsubset \cdots \sqsubset \mathcal{C}_{N-1} = \{\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}\}$$

- Divisive algorithms start with $\mathcal{C}_0 = \{\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}\}$ and at each step t split one cluster into two, yielding $|\mathcal{C}_{t+1}| = |\mathcal{C}_t| + 1$ and $\mathcal{C}_{t+1} \sqsubset \mathcal{C}_t$
- At step $N-1$ have hierarchy:

$$\mathcal{C}_{N-1} = \{\{\mathbf{x}_1\}, \ldots, \{\mathbf{x}_N\}\} \sqsubset \cdots \sqsubset \mathcal{C}_0 = \{\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}\}$$

Hierarchical Clustering

Pseudocode

1. Initialize $\mathcal{C}_0 = \{\{\mathbf{x}_1\}, \ldots, \{\mathbf{x}_N\}\}$, $t = 0$
2. For $t = 1$ to $N-1$
   - Find closest pair of clusters: $(C_i, C_j) = \operatorname{argmin}_{C_s, C_r \in \mathcal{C}_{t-1}, r \neq s} \{d(C_s, C_r)\}$
   - $\mathcal{C}_t = (\mathcal{C}_{t-1} - \{C_i, C_j\}) \cup \{\{C_i \cup C_j\}\}$ and update representatives if necessary

If SM used, replace argmin with argmax

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

Hierarchical Clustering

Example

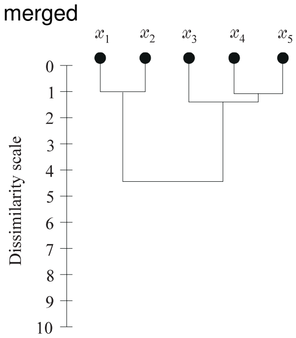
$\mathbf{x}_1 = [1, 1]^T$, $\mathbf{x}_2 = [2, 1]^T$, $\mathbf{x}_3 = [5, 4]^T$, $\mathbf{x}_4 = [6, 5]^T$, $\mathbf{x}_5 = [6.5, 6]^T$, DM = Euclidian/$\alpha_{min}^{ss}$

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

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