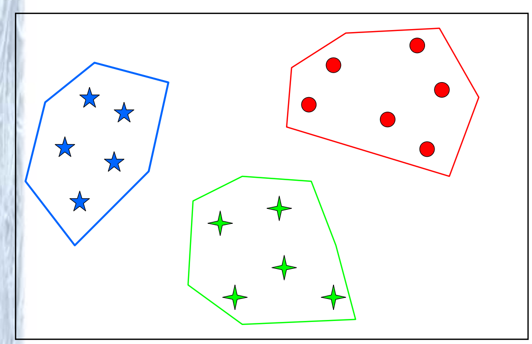
Clustering

- Unsupervised learning
- Generating “classes”
- Distance/similarity measures
- Agglomerative methods
- Divisive methods

What is Clustering?

- Form of *unsupervised* learning - no information from teacher
- The process of partitioning a set of data into a set of meaningful (hopefully) sub-classes, called *clusters*
- Cluster:
  – collection of data points that are “similar” to one another and collectively should be treated as group
  – as a collection, are sufficiently different from other groups

Clusters



Characterizing Cluster Methods

- Class - label applied by clustering algorithm
  – hard versus fuzzy:
    - hard - either is or is not a member of cluster
    - fuzzy - member of cluster with probability
- Distance (similarity) measure - value indicating how similar data points are
- Deterministic versus stochastic
  – deterministic - same clusters produced every time
  – stochastic - different clusters may result
- Hierarchical - points connected into clusters using a hierarchical structure

Basic Clustering Methodology

Two approaches:

Agglomerative: pairs of items/clusters are successively linked to produce larger clusters

Divisive (partitioning): items are initially placed in one cluster and successively divided into separate groups

Cluster Validity

- One difficult question: how *good* are the clusters produced by a particular algorithm?
- Difficult to develop an objective measure
- Some approaches:
  – external assessment: compare clustering to *a priori* clustering
  – internal assessment: determine if clustering intrinsically appropriate for data
  – relative assessment: compare one clustering methods results to another methods

Basic Questions

- Data preparation - getting/setting up data for clustering
  - extraction
  - normalization
- Similarity/Distance measure - how is the distance between points defined
- Use of domain knowledge (prior knowledge)
  - can influence preparation, Similarity/Distance measure
- Efficiency - how to construct clusters in a reasonable amount of time

Distance/Similarity Measures

- Key to grouping points
  distance = inverse of similarity
- Often based on representation of objects as feature vectors

An Employee DB

ID	Gender	Age	Salary
1	F	27	19,000
2	M	51	64,000
3	M	52	100,000
4	F	33	55,000
5	M	45	45,000

Term Frequencies for Documents

	T1	T2	T3	T4	T5	T6
Doc1	0	4	0	0	0	2
Doc2	3	1	4	3	1	2
Doc3	3	0	0	0	3	0
Doc4	0	1	0	3	0	0
Doc5	2	2	2	3	1	4

Which objects are more similar?

Distance/Similarity Measures

Properties of measures:

based on feature values $x_{instance\#,feature\#}$

for all objects x_i,B, dist(x_i, x_j) ≥ 0, dist(x_i, x_j)=dist(x_j, x_i)

for any object x_i, dist(x_i, x_i) = 0

dist(x_i, x_j) $\leq$ dist(x_i, x_k) + dist(x_k, x_j)

Manhattan distance: $\sum_{f=1}^{|features|} |x_{i,f} - x_{j,f}|$

Euclidean distance: $\sqrt{\sum_{f=1}^{|features|} (x_{i,f} - x_{j,f})^2}$

Distance/Similarity Measures

Minkowski distance (p): $\sqrt[p]{\sum_{f=1}^{|features|} (x_{i,f} - x_{j,f})^p}$

Mahalanobis distance: $(x_i - x_j)\nabla^{-1}(x_i - x_j)^T$

where ∇^{-1} is covariance matrix of the patterns

More complex measures:

Mutual Neighbor Distance (MND) - based on a count of number of neighbors

Distance (Similarity) Matrix

- Similarity (Distance) Matrix
  - based on the distance or similarity measure we can construct a symmetric matrix of distance (or similarity values)
  - (i, j) entry in the matrix is the distance (similarity) between items i and j

$$
\begin{array}{c|cccc}
 & I_1 & I_2 & \cdots & I_n \\
\hline
I_1 & \bullet & d_{12} & \cdots & d_{1n} \\
I_2 & d_{21} & \bullet & \cdots & d_{2n} \\
\vdots & \vdots & \vdots & \bullet & \vdots \\
I_n & d_{n1} & d_{n2} & \cdots & \bullet
\end{array}
$$

Note that $d_{ij} = d_{ji}$ (i.e., the matrix is symmetric). So, we only need the lower triangle part of the matrix.

The diagonal is all 1's (similarity) or all 0's (distance)

d_{ij} = similarity (or distance) of D_i to D_j

Example: Term Similarities in Documents

	T1	T2	T3	T4	T5	T6	T7	T8
Doc1	0	4	0	0	0	2	1	3
Doc2	3	1	4	3	1	2	0	1
Doc3	3	0	0	0	3	0	3	0
Doc4	0	1	0	3	0	0	2	0
Doc5	2	2	2	3	1	4	0	2

$$sim(T_i, T_j) = \sum_{k=1}^{N} (w_{ik} \cdot w_{jk})$$

Term-Term Similarity Matrix

	T1	T2	T3	T4	T5	T6	T7
T2	7						
T3	16	8					
T4	15	12	18				
T5	14	3	6	6			
T6	14	18	16	18	6		
T7	9	6	0	6	9	2	
T8	7	17	8	9	3	16	3

Similarity (Distance) Thresholds

– A similarity (distance) threshold may be used to mark pairs that are "sufficiently" similar

	T1	T2	T3	T4	T5	T6	T7
T2	7						
T3	16	8					
T4	15	12	18				
T5	14	3	6	6			
T6	14	18	16	18	6		
T7	9	6	0	6	9	2	
T8	7	17	8	9	3	16	3

Using a threshold value of 10 in the previous example

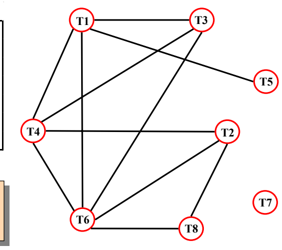
	T1	T2	T3	T4	T5	T6	T7
T2	0						
T3	1	0					
T4	1	1	1				
T5	1	0	0	0			
T6	1	1	1	1	0		
T7	0	0	0	0	0	0	
T8	0	1	0	0	0	1	0

Graph Representation

- The similarity matrix can be visualized as an undirected graph
  – each item is represented by a node, and edges represent the fact that two items are similar (a one in the similarity threshold matrix)

	T1	T2	T3	T4	T5	T6	T7
T2	0						
T3	1	0					
T4	1	1	1				
T5	1	0	0	0			
T6	1	1	1	1	0		
T7	0	0	0	0	0	0	
T8	0	1	0	0	0	1	0

If no threshold is used, then matrix can be represented as a weighted graph



Agglomerative Single-Link

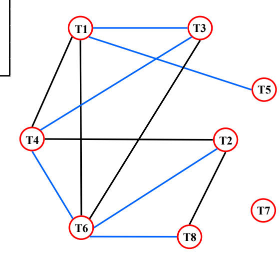
- Single-link: connect all points together that are within a threshold distance
- Algorithm:
  1. place all points in a cluster
  2. pick a point to start a cluster
  3. for each point in current cluster
     add all points within threshold not already in cluster
     repeat until no more items added to cluster
  4. remove points in current cluster from graph
  5. Repeat step 2 until no more points in graph

Example

	T1	T2	T3	T4	T5	T6	T7
T2	7						
T3	16	8					
T4	15	12	18				
T5	14	3	6	6			
T6	14	18	16	18	6		
T7	9	6	0	6	9	2	
T8	7	17	8	9	3	16	3

	T1	T2	T3	T4	T5	T6	T7
T2	0						
T3	1	0					
T4	1	1	1				
T5	1	0	0	0			
T6	1	1	1	1	0		
T7	0	0	0	0	0	0	
T8	0	1	0	0	0	1	0

All points except T7 end up in one cluster



Agglomerative Complete-Link (Clique)

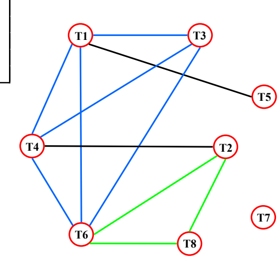
- Complete-link (clique): all of the points in a cluster must be within the threshold distance
- In the threshold distance matrix, a clique is a complete graph
- Algorithms based on finding maximal cliques (once a point is chosen, pick the largest clique it is part of)
  – not an easy problem

Example

	T1	T2	T3	T4	T5	T6	T7
T2	7						
T3	16	8					
T4	15	12	18				
T5	14	3	6	6			
T6	14	18	16	18	6		
T7	9	6	0	6	9	2	
T8	7	17	8	9	3	16	3

	T1	T2	T3	T4	T5	T6	T7
T2	0						
T3	1	0					
T4	1	1	1				
T5	1	0	0	0			
T6	1	1	1	1	0		
T7	0	0	0	0	0	0	
T8	0	1	0	0	0	1	0

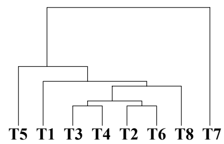
Different clusters possible based on where cliques start



Hierarchical Methods

- Based on some method of representing hierarchy of data points
- One idea: hierarchical dendogram (connects points based on similarity)

	T1	T2	T3	T4	T5	T6	T7
T2	7						
T3	16	8					
T4	15	12	18				
T5	14	3	6	6			
T6	14	18	16	18	6		
T7	9	6	0	6	9	2	
T8	7	17	8	9	3	16	3



Hierarchical Agglomerative

- Compute distance matrix
- Put each data point in its own cluster
- Find most similar pair of clusters
  - merge pairs of clusters (show merger in dendogram)
  - update proximity matrix
  - repeat until all patterns in one cluster

Partitional Methods

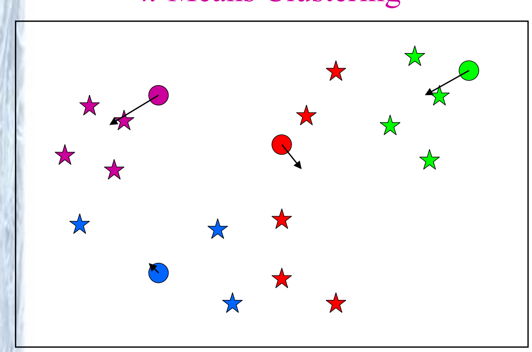
- Divide data points into a number of clusters
- Difficult questions
  - how many clusters?
  - how to divide the points?
  - how to represent cluster?
- Representing cluster: often done in terms of centroid for cluster
  - centroid of cluster minimizes squared distance between the centroid and all points in cluster

k-Means Clustering

1. Choose *k* cluster centers (randomly pick k data points as center, or randomly distribute in space)
2. Assign each pattern to the closest cluster center
3. Recompute the cluster centers using the current cluster memberships (moving centers may change memberships)
4. If a convergence criterion is not met, goto step 2

Convergence criterion:
- no reassignment of patterns
- minimal change in cluster center

k-Means Clustering



k-Means Variations

- What if too many/not enough clusters?
- After some convergence:
  - any cluster with too large a distance between members is split
  - any clusters too close together are combined
  - any cluster not corresponding to any points is moved
  - thresholds decided empirically

An Incremental Clustering Algorithm

1. Assign first data point to a cluster
2. Consider next data point. Either assign data point to an existing cluster or create a new cluster. Assignment to cluster based on threshold
3. Repeat step 2 until all points are clustered

Useful for efficient clustering

Clustering Summary

- Unsupervised learning method
  - generation of "classes"
- Based on similarity/distance measure
  - Manhattan, Euclidean, Minkowski, Mahalanobis, etc.
  - distance matrix
  - threshold distance matrix
- Hierarchical representation
  - hierarchical dendogram
- Agglomerative methods
  - single link
  - complete link (clique)

Clustering Summary

- Partitional method
  - representing clusters
    - centroids and "error"
  - k-Means clustering
    - combining/splitting k-Means
- Incremental clustering
  - one pass clustering