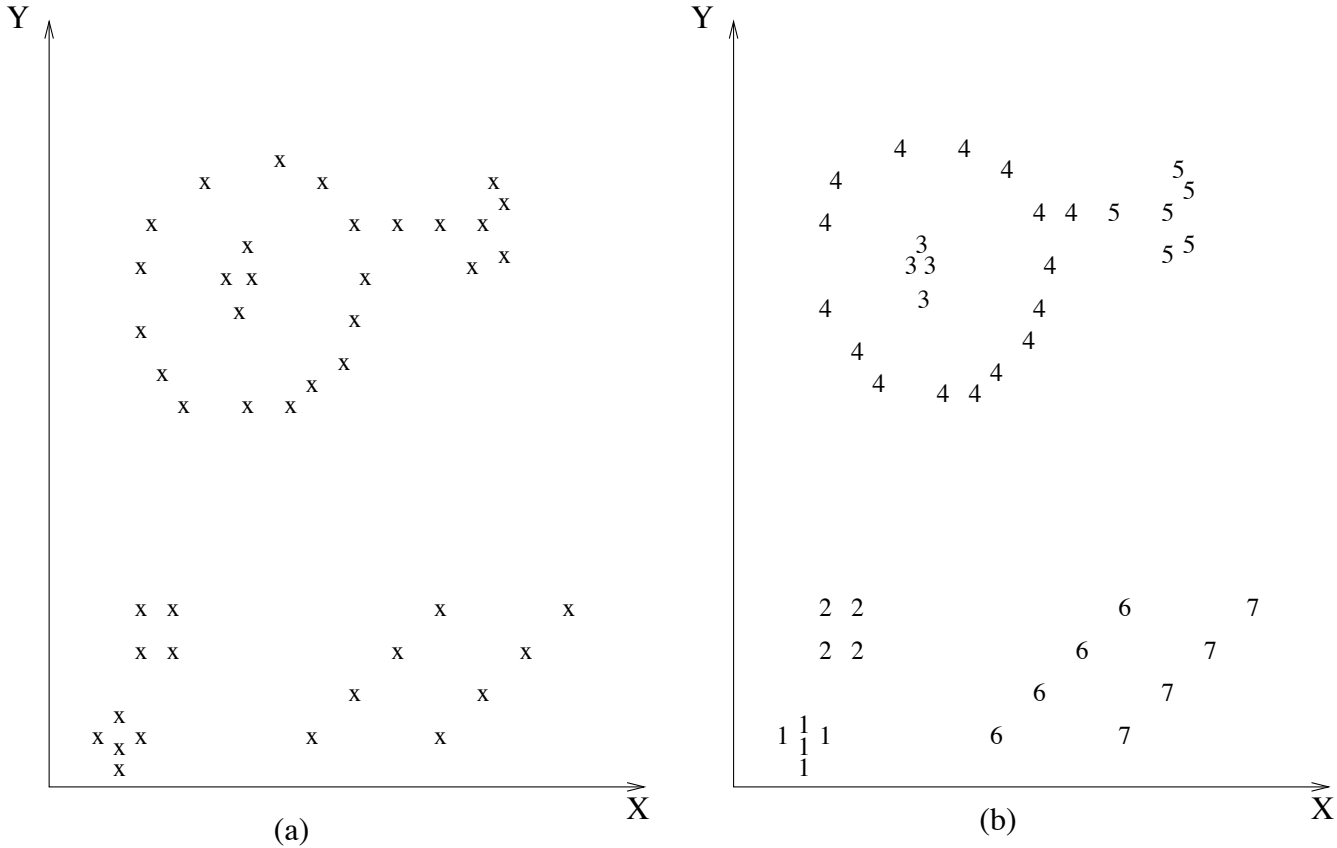
SMLO-4: Clustering: Slides based on *Data Clustering: A Review*, Jain-Murty-Flynn

Douglas Aberdeen

Machine Learning Program
National ICT Australia
Canberra, 0200 ACT

Clustering Problem Definition



Data Clustering Questions

- Feature selection
- Data normalisation
- Similarity/distance measure
- Integration of domain knowledge
- Scaling to very large data sets
  - $O(n)$ algorithms
  - On-line algorithms
- Number of clusters
- Probabilistic or Hard
- Cluster evaluation

Clustering evaluation

- What is the correct clustering?
- $\mathbf{c}_i$ is centroid of cluster i, $i \in \{1, \ldots, k\}$
- $\mathbf{x}_j$ is data point j, $j \in \{1, \ldots, m\}$
- Mean square error (small is good)

$$E(X, k, \mathbf{c}) = \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in X_i} d(\mathbf{x}_j, \mathbf{c}_i)^2$$

- Cluster separation metric (big is good)

$$\frac{\sum_{i=1}^{k} \sum_{i'=i}^{k} d(\mathbf{c}_i, \mathbf{c}_{i'})^2}{\sum_{i=1}^{k} \sum_{\mathbf{x}_j \in X_i} d(\mathbf{x}_j, \mathbf{c}_i)^2}$$

- Domain independent evaluation undecidable?

Other Names For Clustering

- Unsupervised learning (no labels)
- Numerical taxonomy
- Vector quantization (assymetric discretisation)
- Learning by observation

Distance/Similarity Metrics

- We will assume euclidean

$$d(\mathbf{x}, \mathbf{x}') = \sum_d (x_d - x'_d)^2 = \mathbf{x}^\top \mathbf{x}$$

- Mahanalobis distance,

$$d(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \Sigma \mathbf{x}'$$

- l_p metrics, KL-divergence, Hamming
- Distance metric changes results a lot
- Metrics on structures: strings, graphs
- Combine feature selection and similarity: kernels

$$k(\mathbf{x}, \mathbf{x}')$$

Agglomerative or Divisive?

- Agglomerative merges clusters, starting with individual points
- Divisive starts with one big cluster and splits
- Agglomerative yields more information
- Agglomerative is naturally hierarchical
- Divisive is faster
- Can do online divisive

Agglomerative Clustering

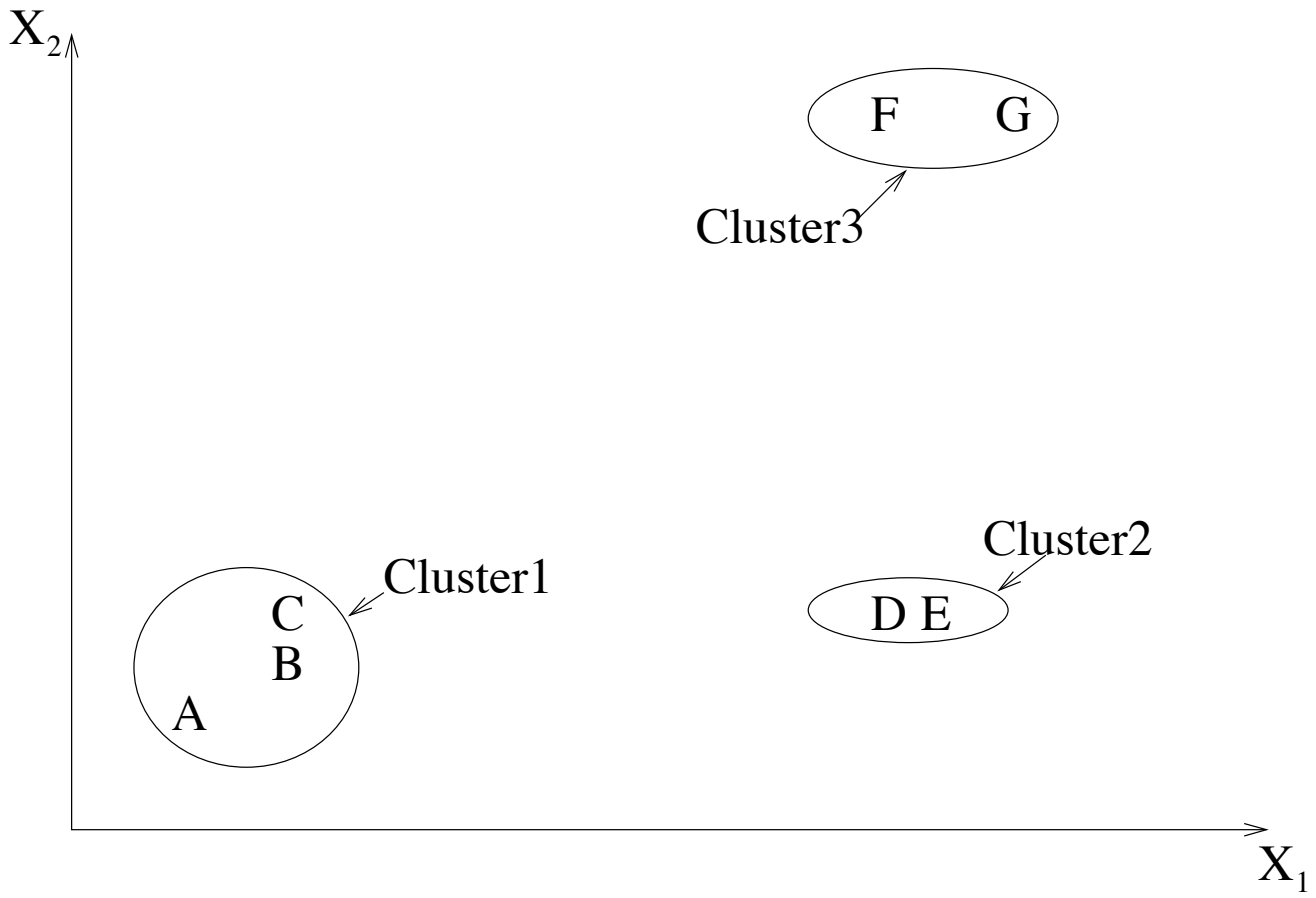
```
All points form an initial cluster
Computer all pairwise distances
while more than one cluster left do
    Merge closest two clusters
    Update dendrogram tree
    Update distances for merged cluster
end while
```

Inter cluster distances:

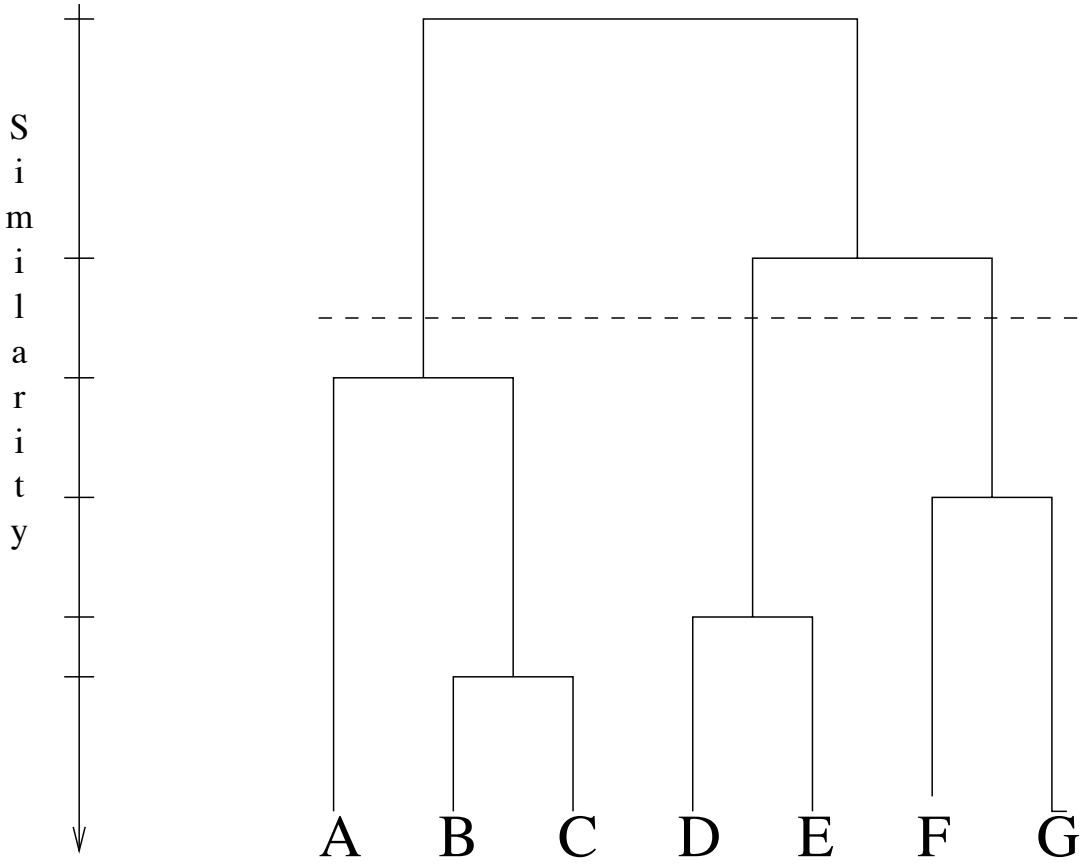
- max pairwise=Complete Link Algorithm
  (Single link learns complex clusters)
- min pairwise=Single Link Algorithm
  (More useful hierarchies)

Prune tree by cluster distance...

Agglomerate Example



Dendrogram



k-means

- Simple and fast, therefore common
- An instance of EM (we saw probabilistic k-means)
- Requires input k: number of clusters
- Moves cluster centers until no change
- Local convergence, depends initial cluster centres
- Usually produces hyper-spheric clusters

k-means cont.

```
Initialise k cluster centres randomly
while Any x changes cluster do
    for each point x_j do
        assign x_j to arg min_i d(x_j, c_i)
    end for
    for each cluster center c_i do
        Let X_i = set of points assigned to cluster i
        c_i = 1/|X_i| Σ_{x_j ∈ X_i} x_j
    end for
end while
```

Clever ways to initialise cluster centres are good

Choosing k

- k-means will find clusters even if they don't exist
- Choice of k is important
- Can choose automatically
  - Split if variance inside cluster is too high
  - Merge if centroids too close

kd-trees

- Split data along chosen dimension
- Recurse into each split, different dimension
- Hierarchical
- Produces a tree of splits
- Efficiently add new points

kd-trees cont.

Initial partition is all data points

```
while More than one data point in current partition do
    Pick next dimension to split (max variance?)
    Split at median data point in that dimension
    Recurse into each half
end while
```

- Result is successively smaller hypercubes
- No way to prune by distance

Minimum Spanning Tree

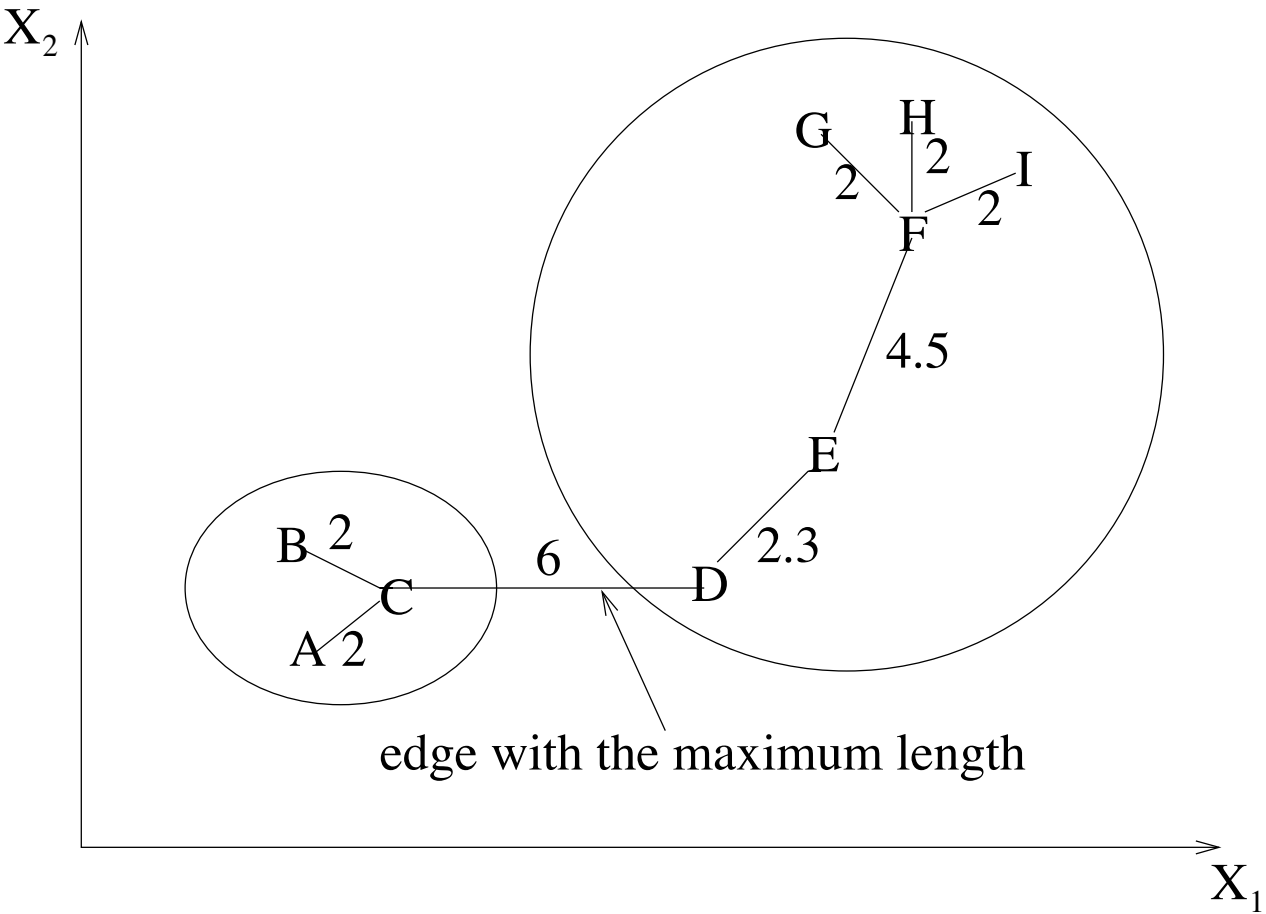
- Graph theoretic clustering
- Minimum spanning tree:
  Shortest set of links to connect all verticies
- Combination of a distance measure and MST algorithm
- Each data point is treated as a vertex in the graph
- Runs in $O(m^2)$ time
- Hierarchical
- Produces a tree with distance pruning interpretation

MST Cont.

```
while Any unconnected data points do
    Join two closest points x, x′ with edge
    while x——x′ induces a loop do
        Try next two closest points
    end while
end while
Remove longest edges to produce clusters
```

- Related to agglomerate algorithms
- ... but no cluster distances
- Related to Nearest neighbour algorithms

MST Example



Simulated Annealing

- Perform greedy search most of the time
- With probability

$$\Pr[suboptimal|\lambda, T] = \frac{exp(1/T)}{exp(\lambda/T) + exp(1/T)}$$

  choose a random update
- Reduce T over time
- Analogy to nature finding lowest energy solution ...cooling metal crystalises as it cools
- Idea is that a local optimisation can turned into a global optimisation with random component.
- Periodically reduce T to ensure convergence
- Many machine learning algorithms reduce a parameter over time to slowly make algorithm more "conservative"

k-means with simulated annealing

- Done by Bandyopadhyay, Maulik, and Pakhira
- While assigning points to closest centroid, occasionally allocate a centroid randomly
- Benefit: hopefully less sensitive to initial centroids
- Drawback: slow convergence, hard to tune

Other Clustering Methods

- Neural Networks (gradient descent)
  - Combine feature selection and cluster assignment
  - Kohonen's Self Organising Feature Map
- Genetic Algorithms
  - Swap cluster point assignments
  - Mutate cluster centroids
  - Make sure you have appropriate fitness function
- Nearest Neighbour Clustering...

Nearest Neighbour Clustering

```
Select distance threshold t
All x start unassigned
for each x ∈ X do
    Find neighbour x_NN = arg min_{x'∈X} d(x, x')
    if d(x, x_NN) > t then
        assign x to a new cluster
    else
        assign x to the same cluster as x_NN
    end if
end for
```

Good for online implementation

Automatically chooses number of clusters k

End of Clustering

k-nearest neighbour classification

- Labelled training data $\{\mathbf{x}, y\}$
- Given labelled instances, classify a new example $\mathbf{x}'$

1: Compute distances $d(\mathbf{x}, \mathbf{x}')$ for all $\mathbf{x} \in X$
2: Keep k nearest $\mathbf{x}$
3: Check labels of k nearest $\mathbf{x}$
4: Class of new sample $\mathbf{x}'$ is majority label of k nearest $\mathbf{x}$

Non-parametric method: classifier records training data