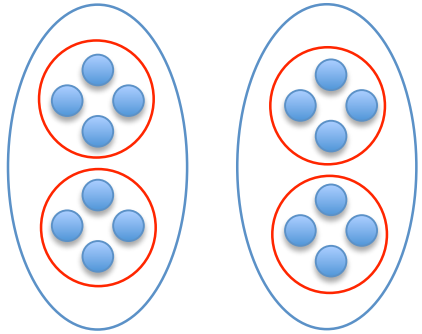
IAML: Hierarchical Clustering

Victor Lavrenko and Nigel Goddard
School of Informatics

Semester 1

Hierarchical clustering

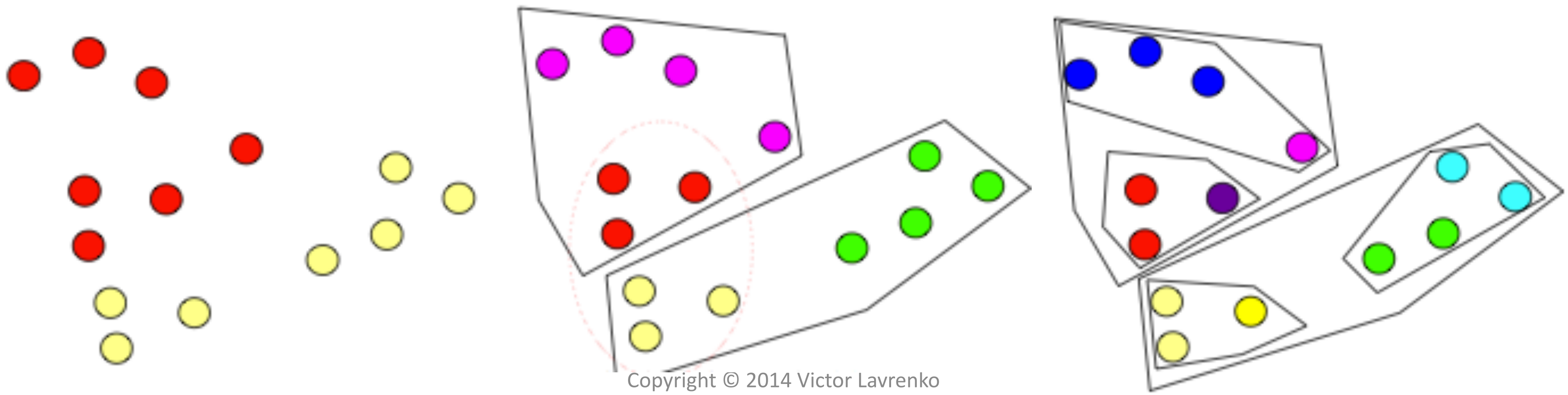


- Selecting K – question of granularity
  - how coarse or fine-grained is the structure in your data?
    - analogy: tidal waves or ripples on the surface?
    - real data: both, and probably everything in-between
  - no clustering algorithm able to pick K (some claim to)
- Instead of picking K – find a hierarchy of structure
  - top levels – coarse effects, low levels – fine-grained
    - topmost cluster – contains every point in the dataset
    - bottom level – set of n singleton clusters, one per data point
  - strategies:
    - top-down: start with all items in one cluster, split recursively
    - bottom-up: start with singletons, merge by some criterion

Copyright © 2014 Victor Lavrenko

Hierarchical K-means

- Top-down approach:
  - run K-means algorithm on the original data $x_1 \dots x_n$
  - for each of the resulting clusters c_i: $i = 1 \dots K$
    - recursively run K-means on points in c_i
- Fast: recursive calls operate on a slice: $O(Knd \log_K n)$
- Greedy: can't cross boundaries imposed by top levels
  - nearby points may end up in different clusters



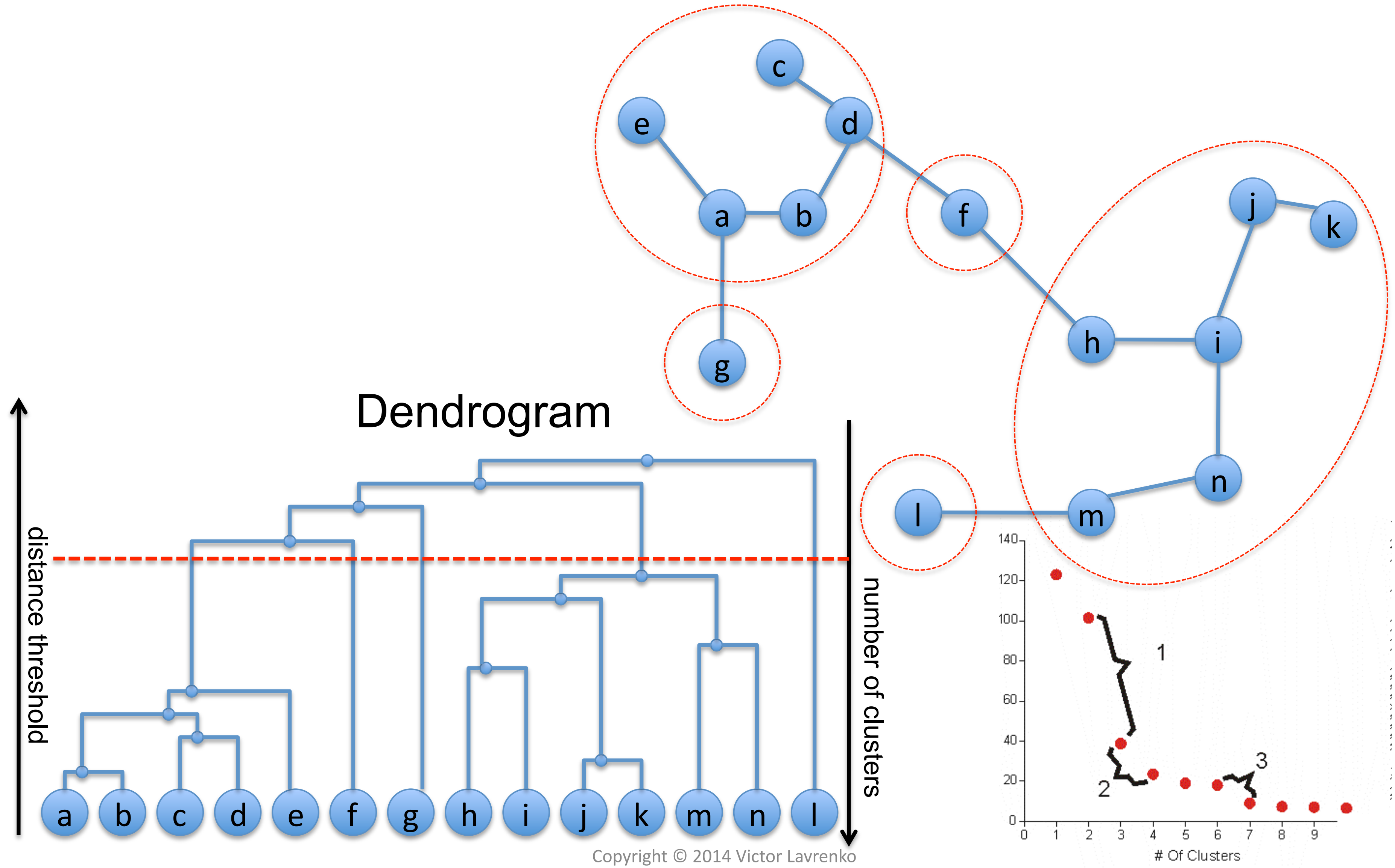
Copyright © 2014 Victor Lavrenko

Agglomerative clustering

- Idea: ensure nearby points end up in the same cluster
- Start with a collection C of n singleton clusters
  - each cluster contains one data point: $c_i=\{x_i\}$
- Repeat until only one cluster is left:
  - find a pair of clusters that is closest: $\min_{i,j} D(c_i, c_j)$
  - merge the clusters c_i, c_j into a new cluster c_{i+j}
  - remove c_i,c_j from the collection C, add c_{i+j}
- Produces a dendrogram: hierarchical tree of clusters
- Need to define a distance metric over clusters
- Slow: $O(n^2d + n^3)$ – create, traverse distance matrix

Copyright © 2014 Victor Lavrenko

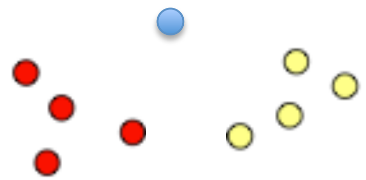
Agglomerative clustering: example



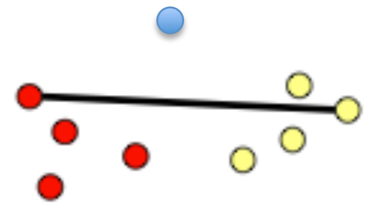
Copyright © 2014 Victor Lavrenko

Cluster distance measures

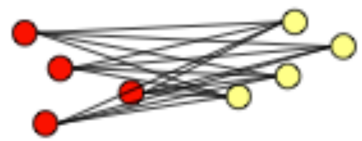
- Single link: $D(c_1, c_2) = \min_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2)$
  - distance between closest elements in clusters
  - produces long chains a→b→c→…→z



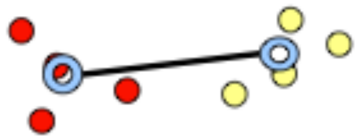
- Complete link: $D(c_1, c_2) = \max_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2)$
  - distance between farthest elements in clusters
  - forces "spherical" clusters with consistent "diameter"



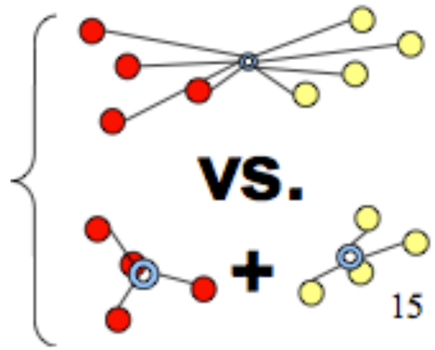
- Average link: $D(c_1, c_2) = \frac{1}{|c_1|} \frac{1}{|c_2|} \sum_{x_1 \in c_1} \sum_{x_2 \in c_2} D(x_1, x_2)$
  - average of all pairwise distances
  - less affected by outliers



- Centroids: $D(c_1, c_2) = D\left(\left(\frac{1}{|c_1|} \sum_{x \in c_1} \vec{x}\right), \left(\frac{1}{|c_2|} \sum_{x \in c_2} \vec{x}\right)\right)$
  - distance between centroids (means) of two clusters



- Ward's method: $TD_{c_1 \cup c_2} = \sum_{x \in c_1 \cup c_2} D(x, \mu_{c_1 \cup c_2})^2$
  - consider joining two clusters, how does it change the total distance (TD) from centroids?



Lance-Williams Algorithm

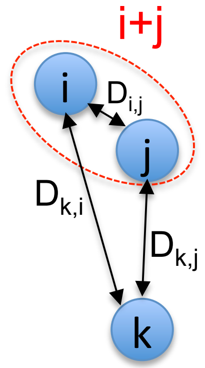
- $D = \{D_{i,j}$: distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ for $i,j=1..N\}$
- for N iterations:

  $i,j =$ **arg min** $D_{i,j}$ … pair of closest clusters

  add cluster: i+j, delete clusters i, j

  for each remaining cluster k:

$$D_{k,i+j} = \alpha_i\, D_{k,i} + \alpha_j\, D_{k,j} + \beta\, D_{i,j} + \gamma\, |D_{k,i} - D_{k,j}|$$



Method	α_i	α_i	β	γ
Single linkage	0.5	0.5	0	-0.5
Complete linkage	0.5	0.5	0	0.5
Group average	$\frac{n_i}{n_i+n_j}$	$\frac{n_j}{n_i+n_j}$	0	0
Weighted group average	0.5	0.5	0	0
Centroid	$\frac{n_i}{n_i+n_j}$	$\frac{n_j}{n_i+n_j}$	$\frac{-n_i \cdot n_j}{(n_i+n_j)^2}$	0
Ward	$\frac{n_i+n_k}{(n_i+n_j+n_k)}$	$\frac{n_j+n_k}{(n_i+n_j+n_k)}$	$\frac{-n_k}{(n_i+n_j+n_k)}$	0

Single link:

$D_{k,i+j}$
$= \frac{1}{2}\,(D_{ki} + D_{kj} - |D_{ki} - D_{kj}|)$
$= \min\,\{D_{ki}\,,\, D_{kj}\}$

$\min_{a,b} = \max_{a,b} - |a-b|$

Copyright © 2014 Victor Lavrenko

Summary

- Clustering: discover underlying sub-populations
- K-means
  - fast, iterative, leads to a local minimum
  - need to pick k: look for unusual reduction in variance
- Mixture models
  - probabilistic version of K-means
  - Expectation Maximization (EM) algorithm
- Hierarchical clustering
  - top-down (K-means) and bottom-up (agglomerative)
  - single / complete / average link variations

Copyright © 2014 Victor Lavrenko