Clustering Methods

Yuan YAO

Hong Kong University of Science and Technology

November 21, 2017

1. K-means vs. K-center

2. Hierarchical Agglomerative Clustering

3. Statistical Consistency of K-means and Robust Single Linkage

About this lecture

- K-means vs. K-center methods
  - cover tree: hierarchical/online K-center
- Hierarchical Agglomerative Clustering
  - average linkage
  - complete linkage
  - single linkage

About cluster analysis

- Techniques for finding subgroups or data points, or clusters, in a data set, so that the observations within each group are quite similar to each other.
- Both clustering and PCA seek to simplify the data via a small number of summaries, but their mechanisms are different:
  1. PCA looks to find a low-dimensional representation of the observations that explain a good fraction of the variance;
  2. Clustering looks to find homogeneous subgroups among the observations.

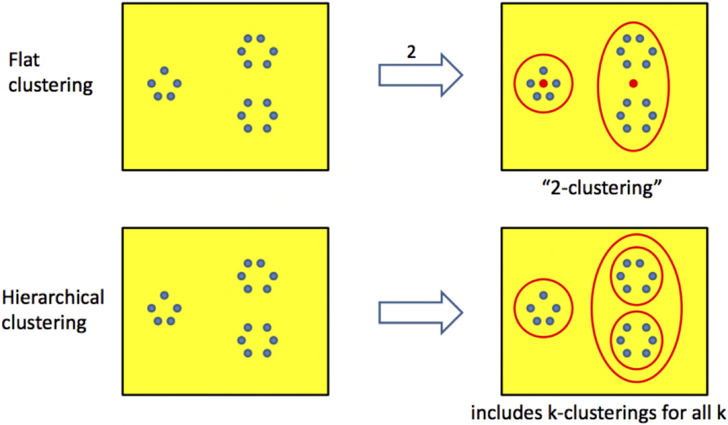


Figure: Two types of clustering

How is the data presented

- Batch
  n data point, all at once
  (can store all of them in memory)
- Online/streaming
  n or endless data point, one by one
  (O(1) or o(n) memory, can NOT store all of them)

K-Means clustering

- Partition the data set of n observations into K distinct, non-overlapping subsets.
  Each set, denoted as C_k, $k = 1, .., K$, is called a cluster.
- Good clustering: the within-cluster variation is as small as possible
- Let $W(C_k)$ be a measure of the within-cluster variation for cluster C_k.
- We wish to minimize the total within-cluster variations

$$\text{minmize}_{C_1,\ldots,C_K} \left\{ \sum_{i=1}^{K} W(C_K) \right\}$$

K-Means clustering

- Several different ways to define $W(C_k)$.
- Using squared Euclidan distance, we define

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,j \in C_k} \|\mathbf{x}_i - \mathbf{x}_j\|^2$$

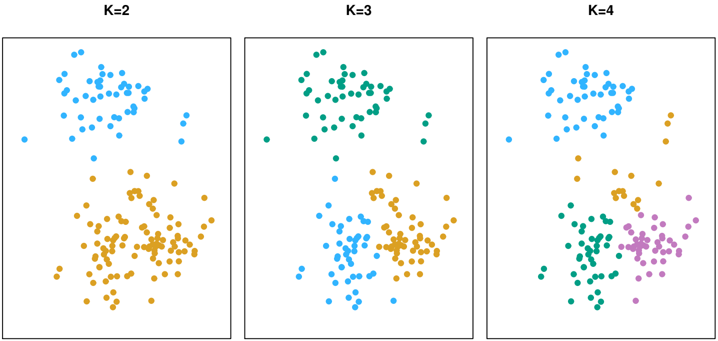


Figure: 10.5. A simulated data set with 150 observations in two-dimensional space. Panels show the results of applying K-means clustering with different values of K, the number of clusters. The color of each observation indicates the cluster to which it was assigned using the K-means clustering algorithm. Note that there is no ordering of the clusters, so the cluster coloring is arbitrary. These cluster labels were not used in clustering; instead, they are the outputs of the clustering procedure.

K-Means cluster algorithm

Algorithm 10.1 K-Means Clustering

- 1. Randomly assign a number, from 1 to K, to each of the observations. These serve as initial cluster assignments for the observations.
- 2. Iterate until the cluster assignments stop changing:
    1. For each of the K clusters, compute the cluster centroid. The kth cluster centroid is the vector of the p feature means for the observations in the kth cluster.
    2. Assign each observation to the cluster whose centroid is closest (where closest is defined using Euclidean distance).

K-Means cluster algorithm

- The objective function always decreases at each step.
- $$\frac{1}{|C_k|}\sum_{i,j\in C_k}\|\mathbf{x}_i-\mathbf{x}_j\|^2 = 2\sum_{i\in C_k}\|\mathbf{x}_i-\bar{\mathbf{x}}\|^2$$
  where $\bar{\mathbf{x}}$ is the sample mean $\mathbf{x}_i$ for $i \in C_k$
- K-means algorithm finds a local minimum.
- The result depends on the initial (random) cluster assignment.
- Try several different initials, and select the best result (the smallest objective function).

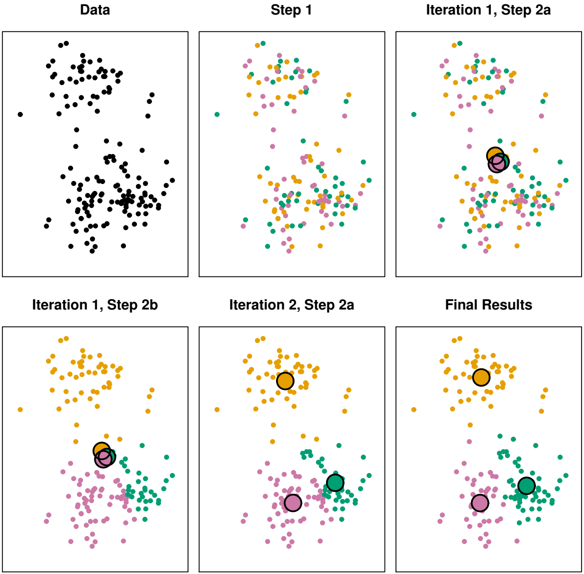
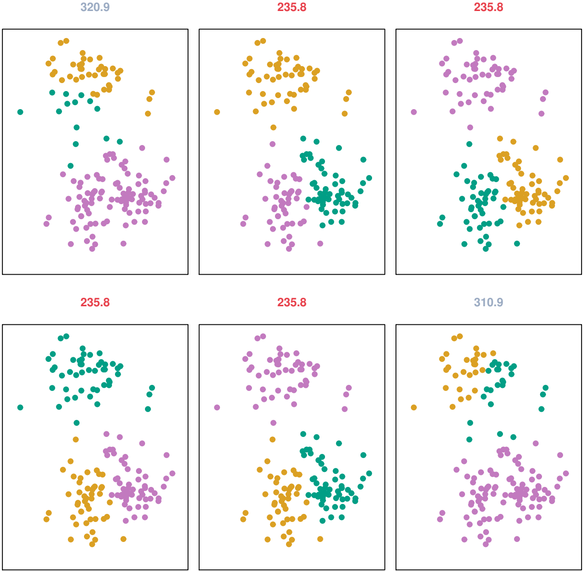

Data
Step 1
Iteration 1, Step 2a
Iteration 1, Step 2b
Iteration 2, Step 2a
Final Results

Figure: 10.6

FIGURE 10.6. The progress of the K-means algorithm on the example of Figure 10.5 with $K = 3$. Top left: the observations are shown. Top center: in Step 1 of the algorithm, each observation is randomly assigned to a cluster. Top right: in Step 2(a), the cluster centroids are computed. These are shown as large colored disks. Initially the centroids are almost completely overlapping because the initial cluster assignments were chosen at random. Bottom left: in Step 2(b), each observation is assigned to the nearest centroid. Bottom center: Step 2(a) is once again performed, leading to new cluster centroids. Bottom right: the results obtained after ten iterations.



320.9 235.8 235.8

235.8 235.8 310.9

Figure: 10.7

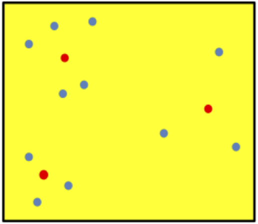
FIGURE 10.7. K-means clustering performed six times on the data from Figure 10.5 with $K = 3$, each time with a different random assignment of the observations in Step 1 of the K-means algorithm. Above each plot is the value of the objective (10.11). Three different local optima were obtained, one of which resulted in a smaller value of the objective and provides better separation between the clusters. Those labeled in red all achieved the same best solution, with an objective value of 235.8.

K-center vs. K-means

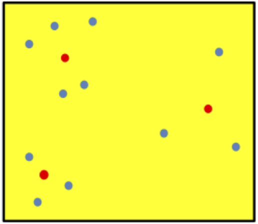
Input: Data set $X \subset R^D$, desired # of clusters k
Goal: Summarize data using a few representatives $C = \{c_1, c_2, \ldots, c_k\} \subset R^D$, to minimize overall distortion.

The *distortion* on a particular x is d(x,C) = min{||x – c||: c in C}

Max distortion (k-center)
max {d(x,C): x in X}



Average distortion (k-means)
sum $\{d(x,C)^2$: x in X}



A Greedy Approximate Algorithm for K-center

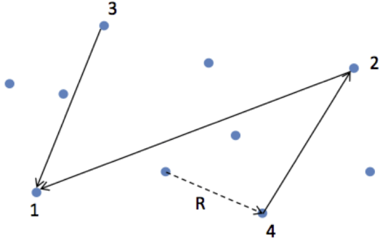
Farthest-first traversal

[Gonzalez, 1985]

Input: data set X, integer k

```
Pick any x in X and set C = {x}
for i = 2 to k:
    find x in X with largest d(x,C)
    add x to C
return centers C
```

eg. k = 4



<u>Claim</u>: cost(C) $\leq$ 2 OPT

Proof:

(i) Let x be the point in X that is farthest from C; and let R = d(x,C). Thus cost(C) = R.

(ii) The k+1 points C $\cup$ {x} are all at distance $\geq$ R from each other.

(iii) Any k-clustering must put two of these points in the same cluster; and this cluster must therefore have radius $\geq$ R/2. Therefore OPT $\geq$ R/2.

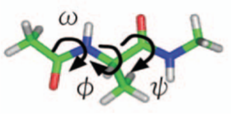
Approximability of K-center

- Upper bounds [Gonzalez, 1985]
  Farthest-first traversal achieves factor 2 approximation for data in a metric space
- Lower bounds [Feder and Greene, 1988]
  Unless P=NP, no polynomial time algorithm achieves a factor
  - better than 2 in a metric space
  - better than 1.82 in Euclidean space
- Open problems:
  - close the gap in the Euclidean space?
  - other algorithms better in practice than the farthest-first traversal?

Geometric Properties of K-center

- K-center gives a R-net of sample set
  - Any two centers in C are R-distance away/seperated
  - Centers in C form a R-cover of samples
- Only depends on metric distance
- K-center is NP-hard, but greedy algorithm is $O(kn)$ of 2-optimal
- K-center is also known as *landmarks* in ISOMAP
- Molecular dynamics application [Sun-Y.-Huang, et al. 2009]
- Shortcomings:
  - sensitive to outliers
  - lacking statistical consistency

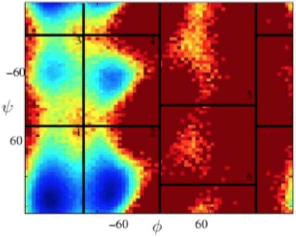
Example: Alanine Dipeptide



[Chodera et al. 2007]

975 trajectories

200 conformations per trajectory



density on $\phi - \psi$ plane

Phi-Psi 3D Torus Embedding

- Reaction coordinates ϕ, ψ of 195,000 points can be embedded on a 3-D torus surface
- $-\log p(x)$ gives free energy on the torus

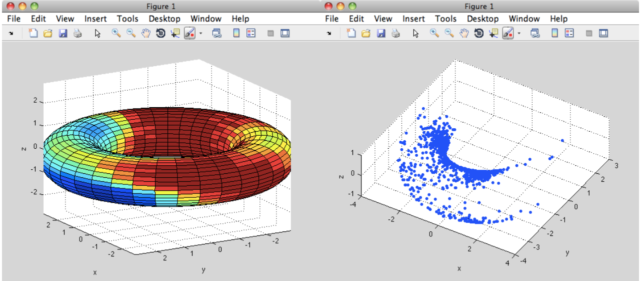
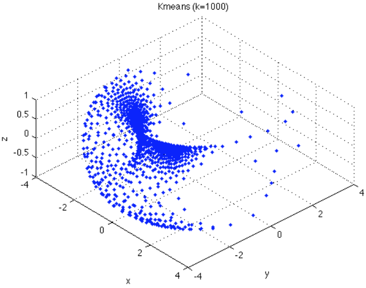
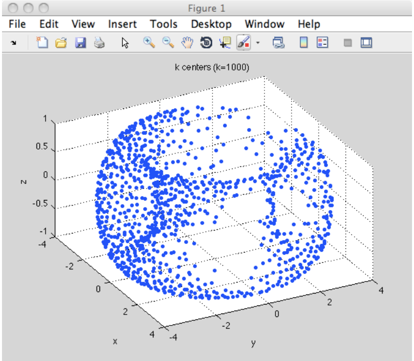


Figure 1
File Edit View Insert Tools Desktop Window Help
2
1
z 0
-1
-2
2
1
0
-1
-2
x
2
1
0
-1
-2
y
Figure 1
File Edit View Insert Tools Desktop Window Help
1
z 0
-1
-4
-2
0
2
4
x
-4
-3
-2
-1
0
1
2
3
y

K-means vs. K-center





A Greedy Approximate Algorithm for K-means

A stochastic farthest-first traversal

kmeans++

[Arthur and Vassilvitskii, 2006]

Input: data set X, integer k

```
Pick x in X at random, set C = {x}
for i = 2 to k:
    pick x in X at random, with
    probability $\propto d(x,C)^2$
    add x to C
return centers C
```

Claim: $E[\text{cost}(C)] \leq O(\log k) \cdot \text{OPT}$

A Constant Approximate Algorithm for K-means

local search

[Kanungo et al, 2003]

Input: data set X, integer k

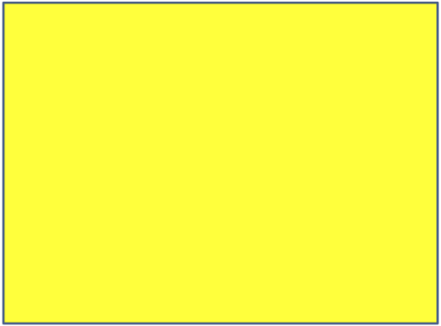
```
Pick initial centers C arbitrarily from X
while ∃ c in C, x in X with
    cost(C – {c} + {x}) < cost(C):
    C = C – {c} + {x}
return C
```

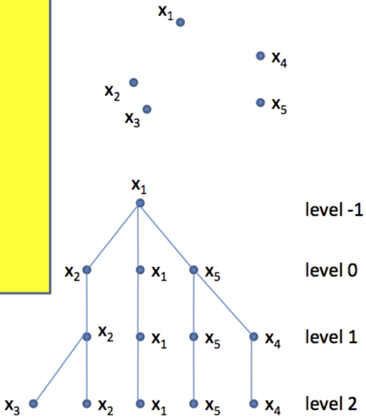
<u>Claim</u>: cost(C) $\leq$ 50 · OPT

Computational Complexity of K-means

- Upper bounds [Inaba et al. 1989]
  Can solve optimally in time $O(n^{kd})$, where
  - n=number of points
  - d=dimension
- Lower bounds [Dasgupta et al. 2009; Mahajan et al. 2009]
  NP-hard in the following cases
  - $k = 2$, arbitrary d
  - $d = 2$, arbitrary k
- Open problems:
  - better approximation algorithms?
  - hardness of approximation results?

Hierarchical K-center

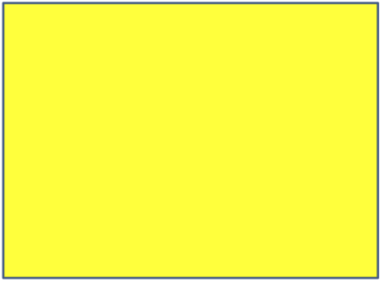


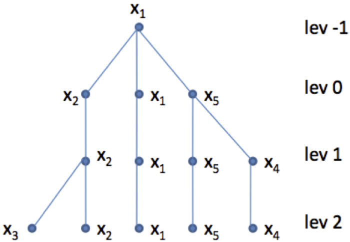


Build online! When new point x arrives:

1. Find largest j such that x is within dist $1/2^j$ of some node p at level j
2. Make x a child of p

Hierarchical K-center: 8-Optimality





Claim: For any k, consider the lowest level with $\leq$ k nodes, and let C_k be those nodes. Then cost(C_k) $\leq$ 8 OPT_k.

Proof: (Suppose it is level j.) C_k's children are within $1/2^j$ of it, and its grandchildren are within $1/2^j + 1/2^{j+1}$ of it, and so on. Therefore:

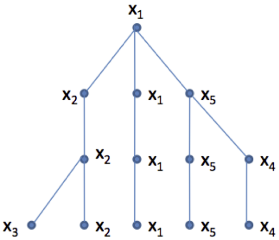
$$\text{cost}(C_k) \leq 1/2^j + 1/2^{j+1} + 1/2^{j+2} + \ldots \leq 1/2^{j-1}$$

Meanwhile, level j+1 has $\geq$ k+1 nodes, at dist $\geq 1/2^{j+1}$ from each other. Any k-clustering puts two of these in the same cluster, and thus has radius $\geq 1/2^{j+2}$.

Hierarchical K-center vs. K-means: Open Problems

- Hierarchical k-center
  Upper bound: 8-approximation. Can we do better?
  Lower bounds: hardness of approximation of *k*-center factor of 2
- Hierarchical k-means
  Any good algorithm for this?

Online K-center



```
For each new point x that arrives:
    Find largest j such that x is within dist $1/2^j$ of some node p at level j
    Make x a child of p
Problem: requires O(n) space – all points are stored
Solution: only maintain levels upto the first level j with $\geq$ k nodes
```

Open problem: online k-means.

Figure: Implemented as *Cover Tree* by Beygelzimer et al. ICML 2006

Hierarchical Agglomerative clustering

- K-means clustering requires pre-specified number of clusters, a disadvantage.
- Hierarchical clustering does not require that number.
- K-means does not have a simple hierarchical or online algorithm.
- K-center has hierarchical/online *cover tree*, top-down.
- Hierarchical Agglomerative Clustering:
  - *bottom-up* or *agglomerative* clustering.
  - dendrogram: a tree-based representation of the observations.

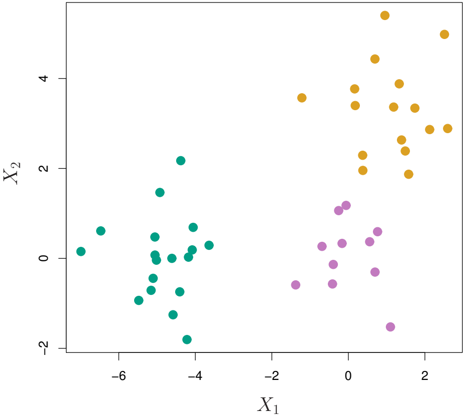


Figure: 10.8. Forty-five observations generated in two-dimensional space. In reality there are three distinct classes, shown in separate colors. However, we will treat these class labels as unknown and will seek to cluster the observations in order to discover the classes from the data.

Dendrogram of Complete Linkage

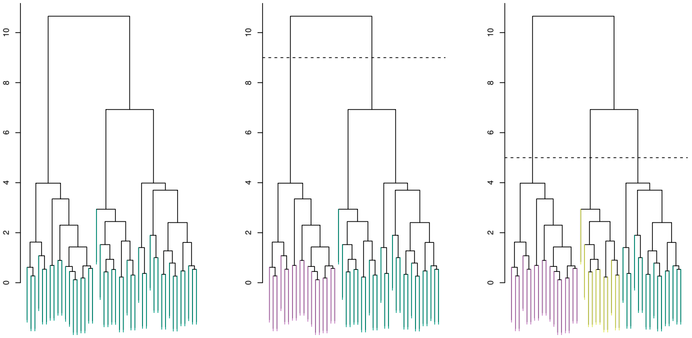


Figure: 10.9. Left: dendrogram obtained from hierarchically clustering the data from Figure 10.8 with complete linkage and Euclidean distance. Center: the dendrogram from the left-hand panel, cut at a height of nine (indicated by the dashed line). This cut results in two distinct clusters, shown in different colors. Right: the dendrogram from the left-hand panel, now cut at a height of five. This cut results in three distinct clusters, shown in different colors. Note that the colors were not used in clustering, but are simply used for display purposes in this figure.

Interpreting a dendrogram

- Each leaf of the dendrogram represents one of the 45 observations in Figure 10.8.
- However, as we move up the tree, some leaves begin to fuse into branches. These correspond to observations that are similar to each other.
- As we move higher up the tree, branches themselves fuse, either with leaves or other branches.
- The earlier (lower in the tree) fusions occur, the more similar the groups of observations are to each other.
- Observations that fuse later (near the top of the tree) can be quite different

A rough closeness measure

- For any two observations, we can look for the point in the tree where branches containing those two observations are first fused. The height of this fusion, as measured on the vertical axis, indicates how different the two observations are
- Observations that fuse at the very bottom of the tree are quite similar to each other, whereas observations that fuse close to the top of the tree will tend to be quite different.

Interpreting dendrogram

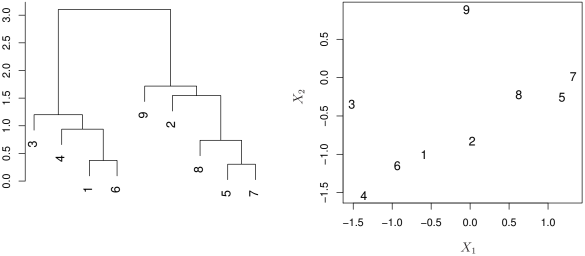


Figure: 10.10. An illustration of how to properly interpret a dendrogram with nine observations in two-dimensional space. Left: a dendrogram generated using Euclidean distance and complete linkage. Observations 5 and 7 are quite similar to each other, as are observations 1 and 6. However, observation 9 is no more similar to observation 2 than it is to observations 8, 5, and 7, even though observations 9 and 2 are close together in terms of horizontal distance. This is because observations 2, 8, 5, and 7 all fuse with observation 9 at the same height, approximately 1.8. Right: the raw data used to generate the dendrogram can be used to confirm that indeed, observation 9 is no more similar to observation 2 than it is to observations 8, 5, and 7. Now

Identifying clusters

- Make a horizontal cut across the dendrogram, as Figure 10.9
- The distinct sets of observations beneath the cut can be interpreted as clusters.
- The lower cuts create more clusters. The higher cuts create less clusters.
- One single dendrogram can be used to obtain any number of clusters.
- Choice of cuts can even be done by visual judgment of the dendrogram.

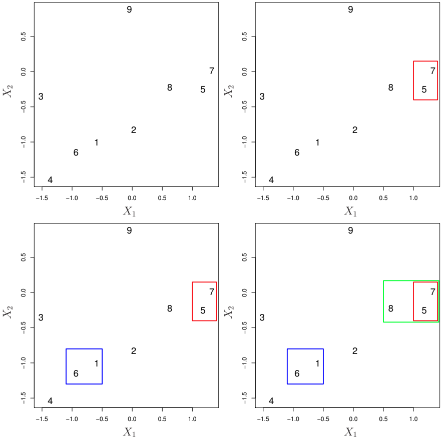


Figure: 10.11. An illustration of the first few steps of the hierarchical clustering algorithm, using the data from Figure 10.10, with complete linkage and Euclidean distance. Top Left: initially, there are nine distinct clusters, $\{1\}, \{2\}, ..., \{9\}$. Top Right: the two clusters that are closest together, $\{5\}$ and $\{7\}$, are fused into a single cluster. Bottom Left: the two clusters that are closest together, $\{6\}$ and $\{1\}$, are fused into a single cluster. Bottom Right: the two clusters that are closest together using complete linkage, $\{8\}$ and the cluster $\{5, 7\}$, are fused into a single cluster.

Hierarchical Agglomerative Clustering

Building a hierarchical clustering:

1. Start with each data point in its own cluster.
2. Repeatedly merge two "closest" clusters.

Notion of distance between clusters:

Single linkage
closest pair of points

Complete linkage
furthest pair of points

Average linkage – several variants
(i) distance between centers
(i) average pairwise distance
(ii) *Ward's method*: increase in k-means cost due to merger

Hierarchical clustering algorithm

- 1. Begin with n observations and a measure (such as Euclidean distance) of all the $\binom{n}{2} = n(n-1)/2$ pairwise dissimilarities. Treat each observation as its own cluster.
- 2. For $i = n, n-1, ...2$:
  1. Examine all pairwise inter-cluster dissimilarities among the i clusters and identify the pair of clusters that are least dissimilar (that is, most similar). Fuse these two clusters. The dissimilarity between these two clusters indicates the height in the dendrogram at which the fusion should be placed.
  2. Compute the new pairwise inter-cluster dissimilarities among the $i-1$ remaining clusters.

Linkage: the dissimilarity measure between two clusters

Linkage	Description
Complete	Maximal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the largest of these dissimilarities.
Single	Minimal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the smallest of these dissimilarities. Single linkage can result in extended, trailing clusters in which single observations are fused one-at-a-time.
Average	Mean intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the average of these dissimilarities.
Centroid	Dissimilarity between the centroid for cluster A (a mean vector of length p) and the centroid for cluster B. Centroid linkage can result in undesirable inversions.

TABLE 10.2. A summary of the four most commonly-used types of linkage

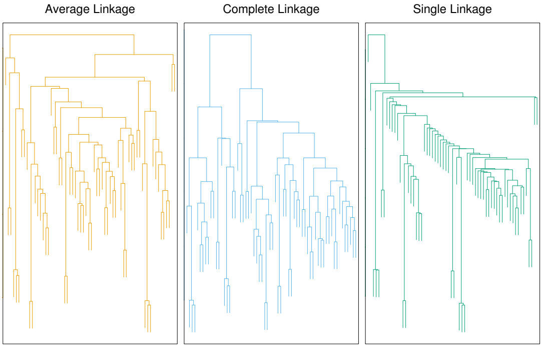


Figure: 10.12. Average, complete, and single linkage applied to an example data set. Average and complete linkage tend to yield more balanced clusters.

Dissimilarity measure

- Very important and can greatly affect the final result.
- Euclidean distrance.
- Correlation based distrance: if two observations have high correlation, the distance is closer.
  (Caution: this is not correlation between two variables, but between two observations.)
- Different problem may need different dissimilarity measure.

Complete Linkage vs. K-center

- Complete Linkage has underlying k-center cost function, any approximability characterization?
- Theorem [Dasgupta 09]: for all k, the induced k-clustering by complete linkage is within factor $\alpha(k)$ of the optimal k-center solution, with $\alpha(k)$ satisfying

$$k \leq \alpha(k) \leq k^{\log 3}$$

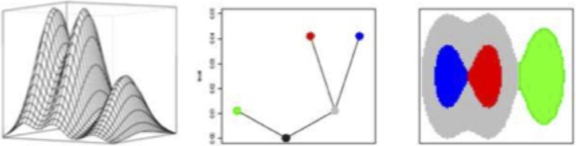
- Open Problem: Wards method of average linkage has the underlying k-means cost function what is its approximation ratio?

Statistical Consistency of K-means

- Suppose data $D_n = \{X_i : i = 1, \ldots, n\}$ is drawn i.i.d. from an underlying distribution P. Let C_k be the optimal k-means centers with respect to P, and $\hat{C}_{nk}$ be the optimal k-means centers for D_n.
- Theorem [Pollard 81]. If μ_j is unique for $1 \leq j \leq k$, then $d(C_k, \hat{C}_{nk}) \to 0$ almost surely, where $d(S,T) = \max_{s\in S} \min_{t \in T} \|s - t\|$.
- Open Issues:
  1. $\hat{C}_{nk}$ is NP-hard to compute
  2. Is C_k something truly useful?

Density Cluster Tree (Reeb Graph)

- For a density function $p(x)$, consider the super-level set $\{x : p(x) \geq r\}$ and let C_r be the connect components of this super-level set.
- Theorem [Hartigan 81]: if $r \leq s$, then $C_s \subseteq C_r$ (functoriality), i.e. hierarchical clustering is with tree structure.



Consistency of Robust Single Linkage

- Robust Single Linkage:
  Build a neighborhood graph G:
  1. node set $V = \{X_i\}$
  2. edge set $E = \{(i,j) : d(X_i, X_j) \leq r\}$
  3. discard nodes with degree $< c \log n$

  Let $\hat{C}_{nr}$ be the connected components of such a graph.
- Theorem [Stuetzle 03; Zhou-Wong 08]: $\hat{C}_{nr}$ converges to the density cluster tree.
- Remark: this is equivalent to 1-skeleton Rips complex with persistent 0-homology, up to a permutation of labels.

Summary

Methods	K-means	K-center	Average	Complete	Single
Complexity	NP	NP	$\approx$ K-means	$\approx$ K-center	Minimal Spanning Tree
Appoximability	50-opt	2-opt. O(kn)	?	$k < \alpha(k) < k^{\log 3}$	?
Online	?	Cover-tree (8-opt)	?	?	Persistent Homology
Hierarchical	?	Cover-tree	Yes	Yes	Yes
Consistency	Pollard'81	No (metric-net)	?	?	Hartigen'81; Stuetzle'03