PQHS 471
Lecture 3: Unsupervised Learning (2)
Clustering

Clustering

- *Clustering:* A very broad set of techniques for finding *subgroups*, or *clusters*, in a dataset.
- We seek a partition of the data into distinct groups so that the observations within each group are homogeneous.
- To make this concrete, we must define what it means for two or more observations to be *similar* or *different*.
- This is often a domain-specific consideration that must be made based on knowledge of the data being studied.

Similar or Different?

The most common choice for the *measurement of similarity* between two random vectors: **Squared Euclidean Distance**
Suppose we have two $n \times 1$ vectors $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$, the *Squared Euclidean Distance* is defined as:

$$d(\boldsymbol{x_1}, \boldsymbol{x_2}) = ||\boldsymbol{x_1} - \boldsymbol{x_2}||^2 = \sum_{i=1}^{n} (x_{1i} - x_{2i})^2$$

This requires **all the features be standardized** in advance.

pop quiz

$$\boldsymbol{a} = (0, 3, 1, 4)^{\mathrm{T}}$$

$$\boldsymbol{x_1} = (1, 2, 1, 5)^{\mathrm{T}}$$

$$\boldsymbol{x_2} = (-1, 3, 3, 2)^{\mathrm{T}}$$

For $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$, who is closer to $\boldsymbol{a}$?

pop quiz

$$\boldsymbol{a} = (0, 3, 1, 4)^{\mathrm{T}}$$

$$\boldsymbol{x_1} = (1, 2, 1, 5)^{\mathrm{T}}$$

$$\boldsymbol{x_2} = (-1, 3, 3, 2)^{\mathrm{T}}$$

For $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$, who is closer to $\boldsymbol{a}$?

$$d(\boldsymbol{a}, \boldsymbol{x_1}) = 1^2 + 1^2 + 0^2 + 1^2 = 3$$

$$d(\boldsymbol{a}, \boldsymbol{x_2}) = 1^2 + 0^2 + 2^2 + 2^2 = 9$$

$\boldsymbol{x_1}$ and $\boldsymbol{a}$ are more similar.

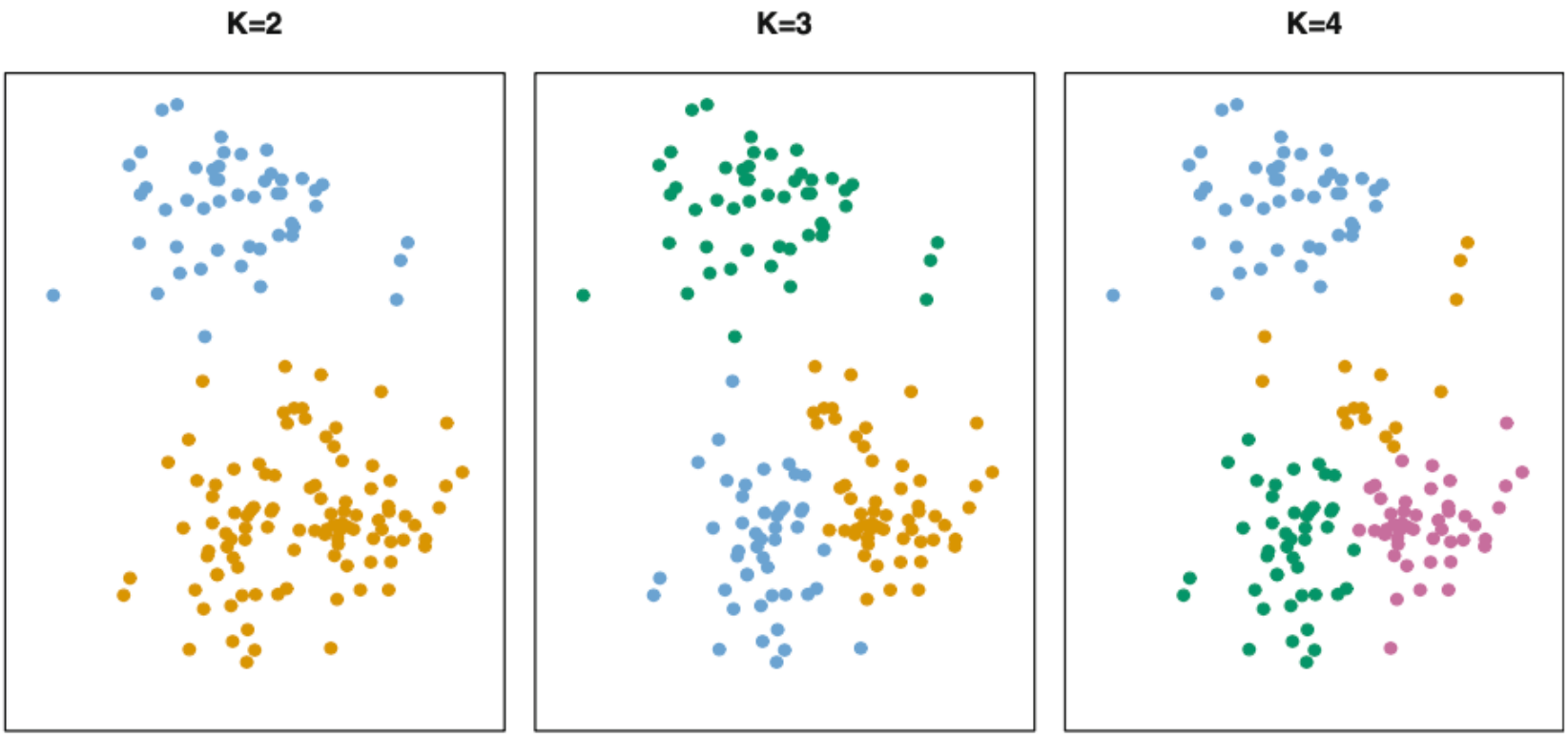
Aside: PCA vs Clustering

- PCA looks for a low-dimensional representation of the observations that explains a good fraction of the variance.
- Clustering looks for homogeneous subgroups among the observations.

Two classic clustering methods

- K-means clustering: an iterative method to partition the observations into a pre-specified number (K) of clusters.
- hierarchical clustering: unknown number of clusters in advance, we end up with a tree-like visualization of the observations.
  - A tree-like visualization of the observations is called a dendrogram.
  - Allow us to view at once the clusterings obtained for each possible number of clusters, from 1 to n.

K-means clustering



$n = 150, p = 2$

The colored labels were not used in clustering; instead, they are the outputs of the K-means algorithm.

centroid

Definition: A **centroid** of a cluster is a point (location) $\boldsymbol{x_0}$ that minimize

$$\frac{1}{|C|}\sum_{i\in C}||\boldsymbol{x_i}-\boldsymbol{x_0}||^2$$

, its average squared Euclidean distance from all the observations in the cluster.
Here, $|C|$ denotes the number of observations in the cluster C.
The **centroid** is often defined as the average $\frac{1}{|C|}\sum_{i\in C}\boldsymbol{x_i}$

K-means clustering

For the K-means clustering method, we need to have:

1. a measure of **dissimilarity**, $d(\boldsymbol{x_i}, \boldsymbol{x_j}$ between any two points, $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$;
2. a definition of **centroid** for any set of data points;
3. a number **K**, the target number of subsets.

Given a cluster, the within-cluster heterogeneity can be defined as

$$W(C) = \frac{1}{|C|} \sum_{i,j \in C} d(\boldsymbol{x_i}, \boldsymbol{x_j})$$

Goal: Given K, find a partition of data so that the total within-cluster heterogeneity $W(C)$ is minimized.

K-means clustering

For the K-means clustering method, we need to have:

1. a measure of **dissimilarity**, $d(\boldsymbol{x_i}, \boldsymbol{x_j}$ between any two points, $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$;
2. a definition of **centroid** for any set of data points;
3. a number **K**, the target number of subsets.

Given a cluster, the within-cluster heterogeneity can be defined as

$$W(C) = \frac{1}{|C|} \sum_{i,j \in C} d(\boldsymbol{x_i}, \boldsymbol{x_j})$$

Goal: Given K, find a partition of data so that the total within-cluster heterogeneity $W(C)$ is minimized.

Goal (more rigorously): Given a partition of data, let $\{C_1, ..., C_K\}$ be the sets of indices for the partition. Then the goal is:

$$minimize_{\{C_1,...,C_K\}} \sum_{k=1}^{K} W(C_k)$$

K-means clustering

K-means algorithm:

1. Randomly assign the observations to K classes (every class needs to have at least one observation).

K-means clustering

K-means algorithm:

1. Randomly assign the observations to K classes (every class needs to have at least one observation).
2. Iterate the following two steps:
   - Update centroids: Calculate the centroids for the K classes.
   - Update membership: Re-assign every observation to the class represented by the centroid "closest" to it (i.e., that centroid that has the smallest d from the observation).

K-means clustering

K-means algorithm:

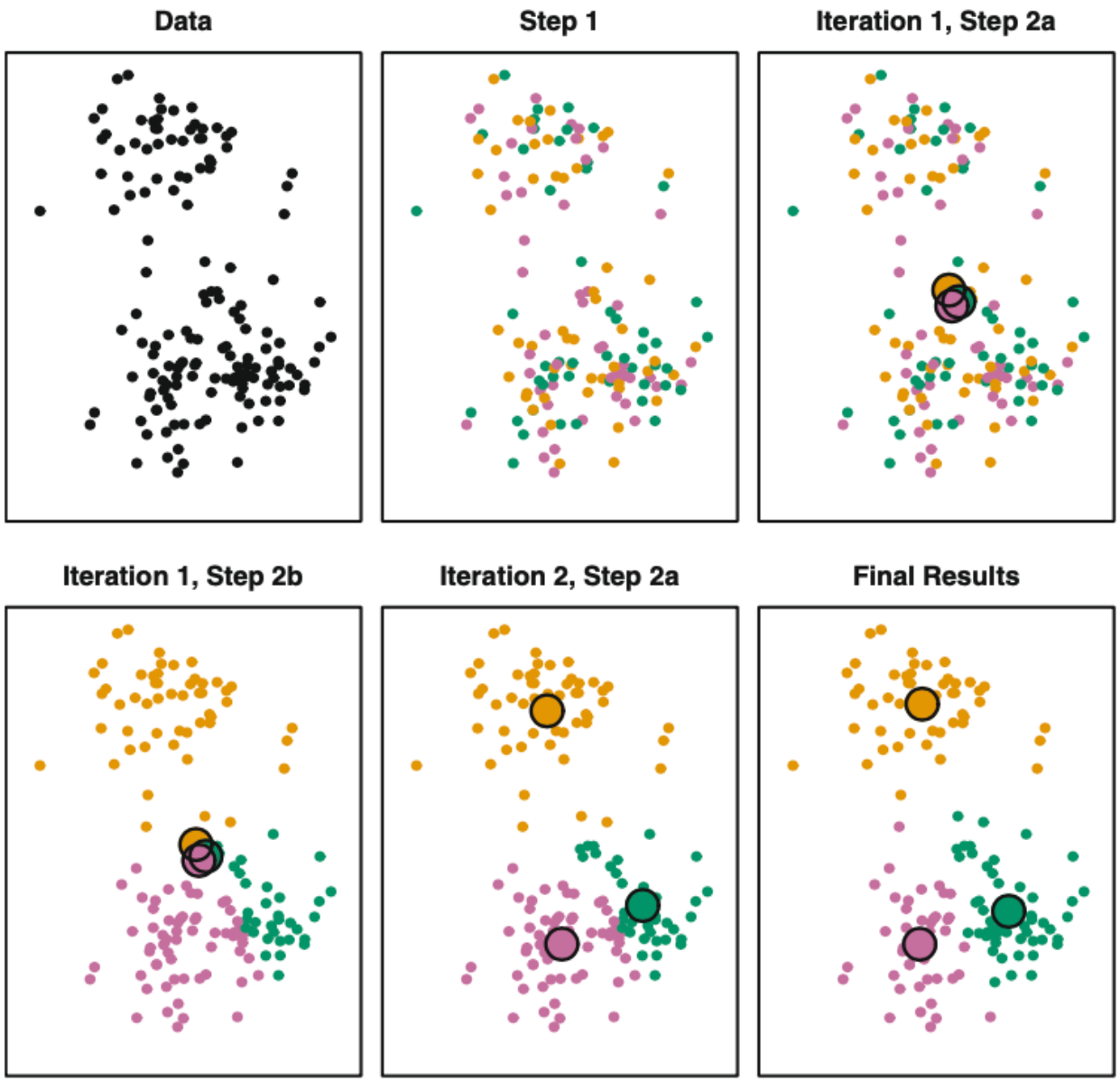
1. Randomly assign the observations to K classes (every class needs to have at least one observation).
2. Iterate the following two steps:
   - Update centroids: Calculate the centroids for the K classes.
   - Update membership: Re-assign every observation to the class represented by the centroid "closest" to it (i.e., that centroid that has the smallest d from the observation).
3. Once converged (i.e., there is no change in membership or centroids), record the partition and its $\sum_k W(C_k)$.

K-means clustering

K-means algorithm:

1. Randomly assign the observations to K classes (every class needs to have at least one observation).
2. Iterate the following two steps:
   - Update centroids: Calculate the centroids for the K classes.
   - Update membership: Re-assign every observation to the class represented by the centroid "closest" to it (i.e., that centroid that has the smallest d from the observation).
3. Once converged (i.e., there is no change in membership or centroids), record the partition and its $\sum_k W(C_k)$.
4. Repeat 1–3 multiple times. Select the partition with the smallest $\sum_k W(C_k)$.

K-means example



Algorithm progress for the example above

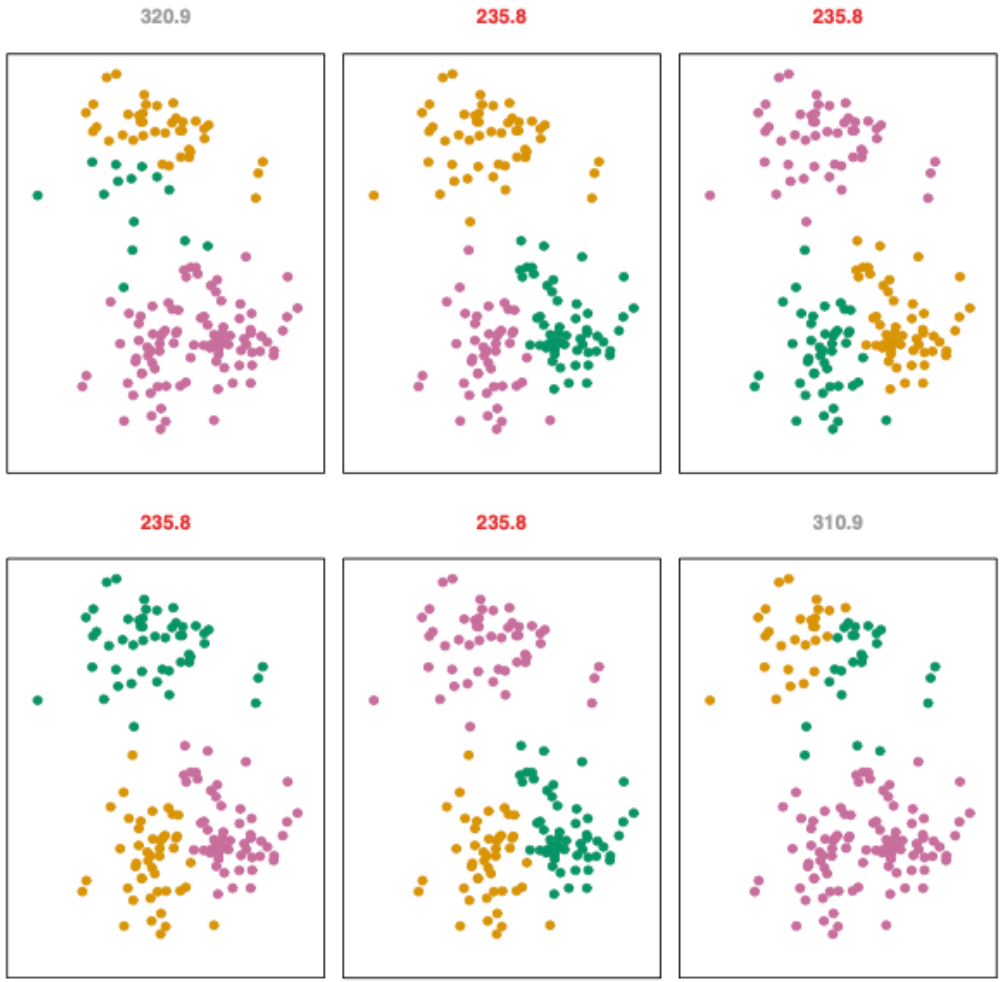
The progress of the K-means algorithm with $K=3$.

- 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 10 iterations.

Properties of K-means

- K-means is a greedy algorithm: it is guaranteed to decrease the value of the objective function AT EACH STEP.
- However it is not guaranteed to give the global minimum.
- Algorithm step 4 increase our chance to hit the global minimum.
- Adding features may bring more information (if they are informative) or dilute information (if they are noisy).
- K is a hyperparameter based upon domain knowledge.

Example: different starting values



Red numbers above each panel are objective function values, the overall within-cluster heterogeneity.

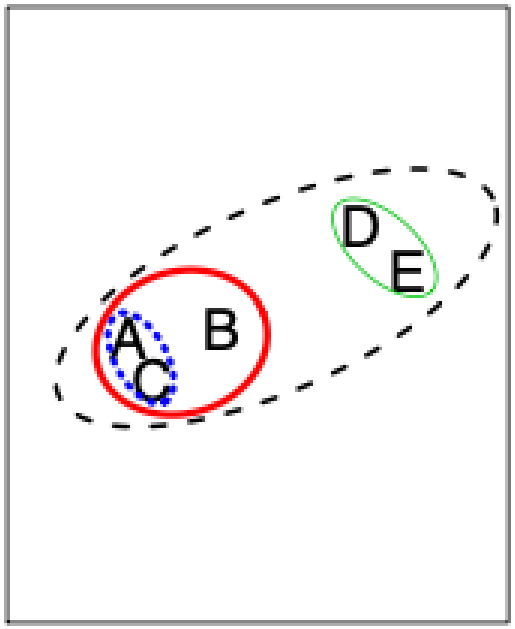
Hierarchical Clustering

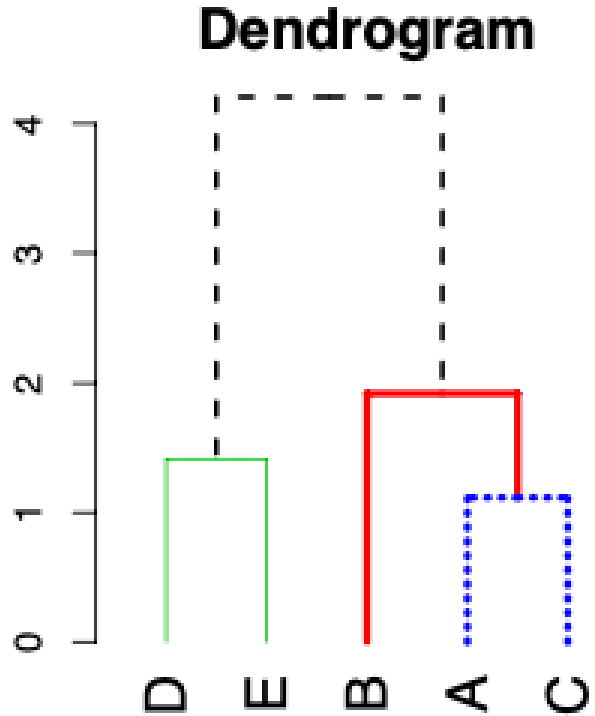
Hierarchical Clustering

- A disadvantage of K-means: requires pre-specify the number of clusters, K.
- Hierarchical clustering offers an alternative approach that does not require a particular choice of K.
- A tree that is upside-down: dendrogram is built starting from the leaves and combining clusters up to the trunk.
- It is a *bottom-up* or *agglomerative* clustering method.

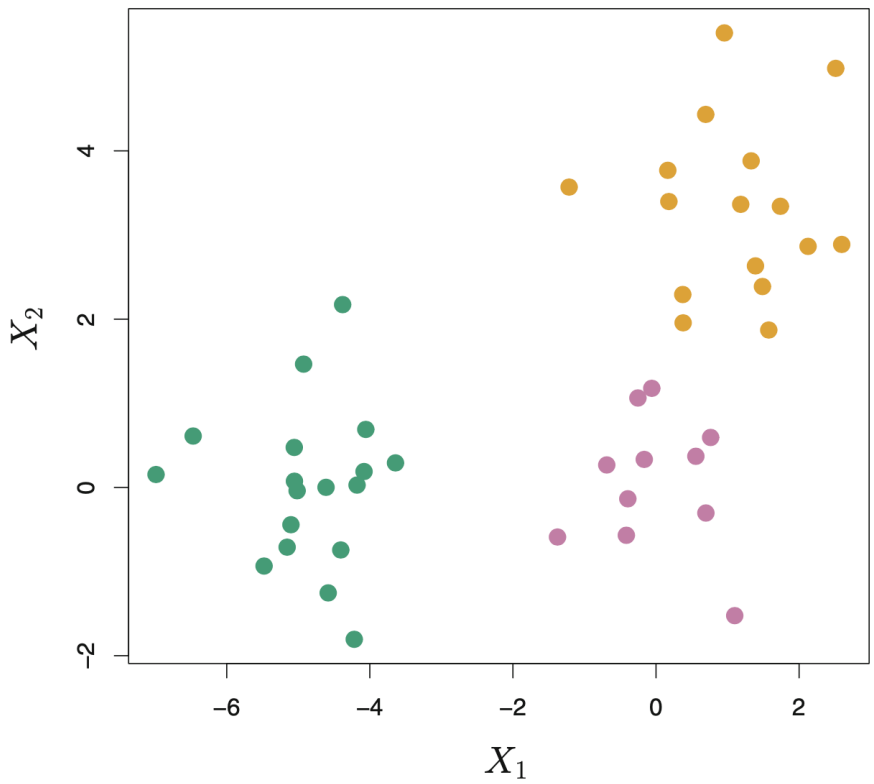
Hierarchical Clustering Algorithm

1. Start with each point in its own cluster.
2. Identify the closest two clusters and merge them.
3. Repeat.
4. Ends when all points are in a single cluster.



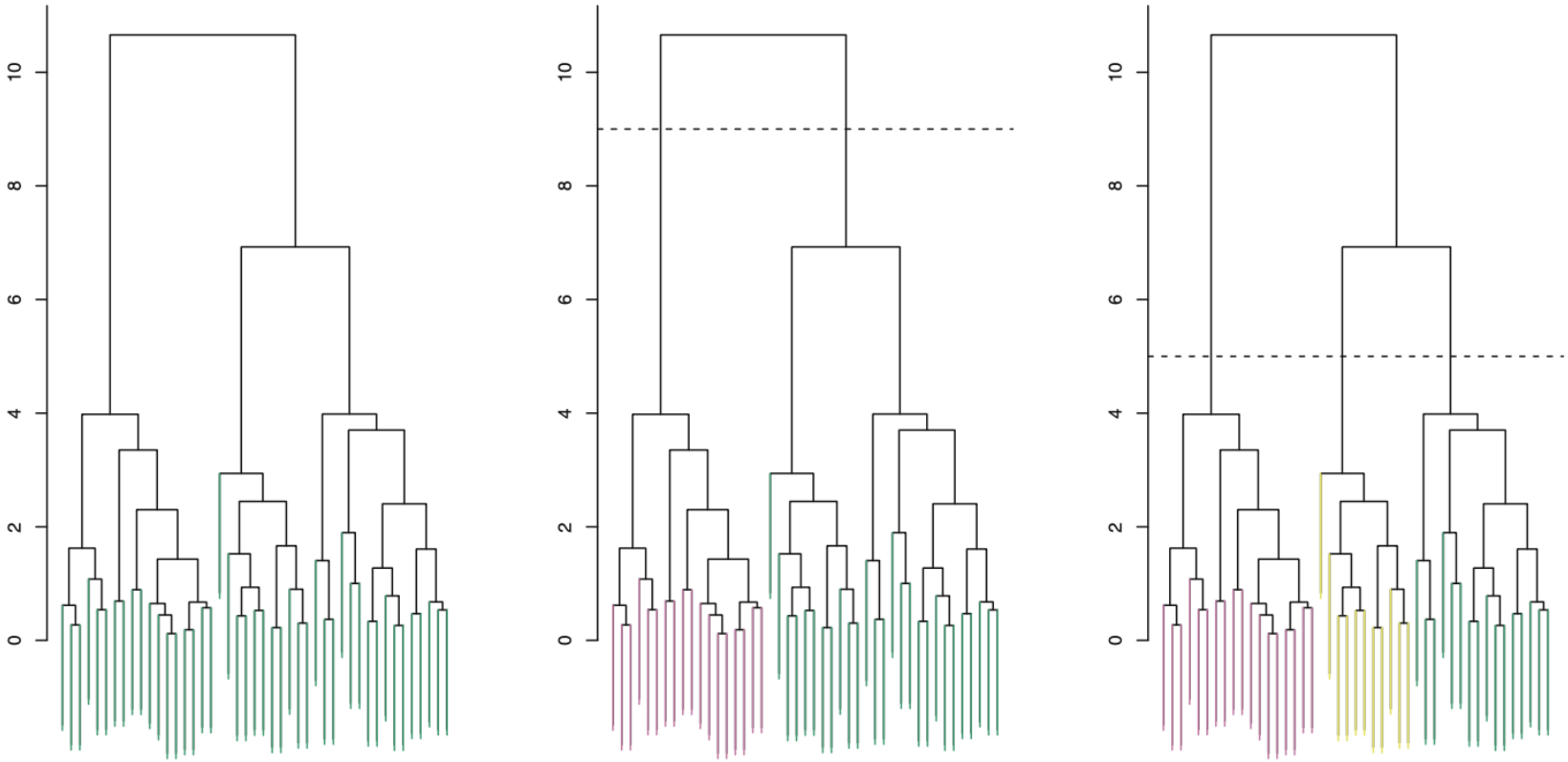


Hierarchical Clustering Algorithm(Intuitive)



- 45 observations in 2D space.
- Data generated using 3 distinct clusters, in separate colors.
- We'll treat the clusters AS UNKNOWN for hierarchical clustering.

where to ‘cut’ the tree



define the cluster-wise dissimilarity

We already have a concept of the dissimilarity between two observations, but we haven't define that for two clusters.
We need to define the dissimilarity between two groups of observations.
We call $d(C_i, C_j)$ as **linkage**.

define the cluster-wise dissimilarity

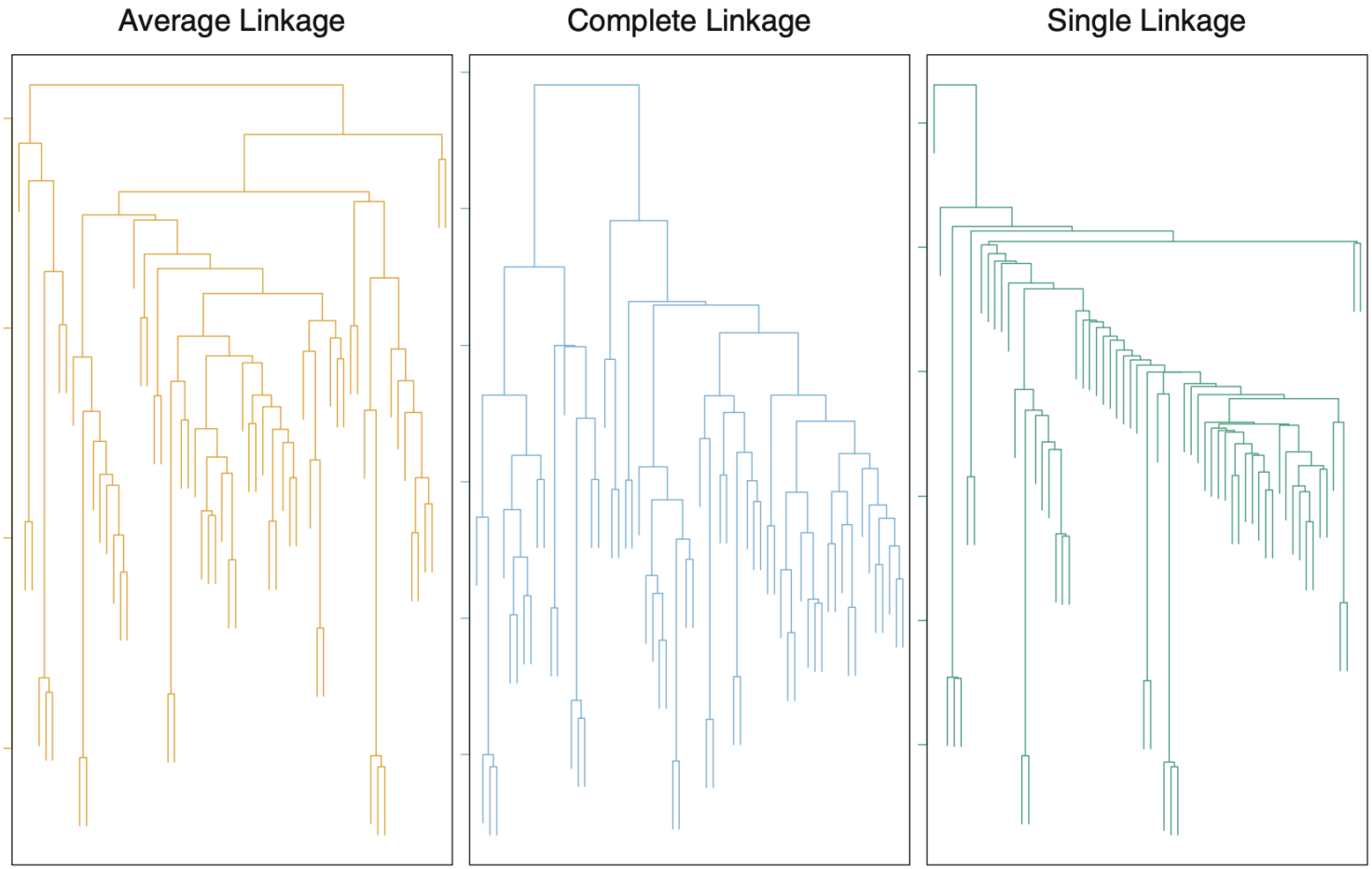
We already have a concept of the dissimilarity between two observations, but we haven't define that for two clusters.
We need to define the dissimilarity between two groups of observations.
We call $d(C_i, C_j)$ as **linkage**.

- $d(C_1, C_2) = \max_{i \in C_1, j \in C_2} d(x_i, x_j)$ (**complete** linkage)
- $d(C_1, C_2) = \min_{i \in C_1, j \in C_2} d(x_i, x_j)$ (**single** linkage)
- $d(C_1, C_2) = \text{average}_{i \in C_1, j \in C_2} d(x_i, x_j)$ (**average** linkage)
- $d(C_1, C_2) = d(\text{centroid}_1, \text{centroid}_2)$ (**centroid** linkage)

define linkage

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*.

choice of linkage



choice of linkage

- Average, complete, and single linkage are most popular among statisticians.
- Average and complete linkage tend to yield more balanced dendrograms.
- Centroid is often used in genomics, but can suffer from *inversion* (two clusters are fused at a height below either of the individual clusters in the dendrogram.)
- The resulting dendrogram depends quite strongly on the type of linkage used.

Exercise

Suppose we have 5 observations, for which we have already computed a similarity (distance) matrix as follows:

	A	B	C	D	E
A	0				
B	9	0			
C	3	7	0		
D	6	5	9	0	
E	11	10	2	8	0

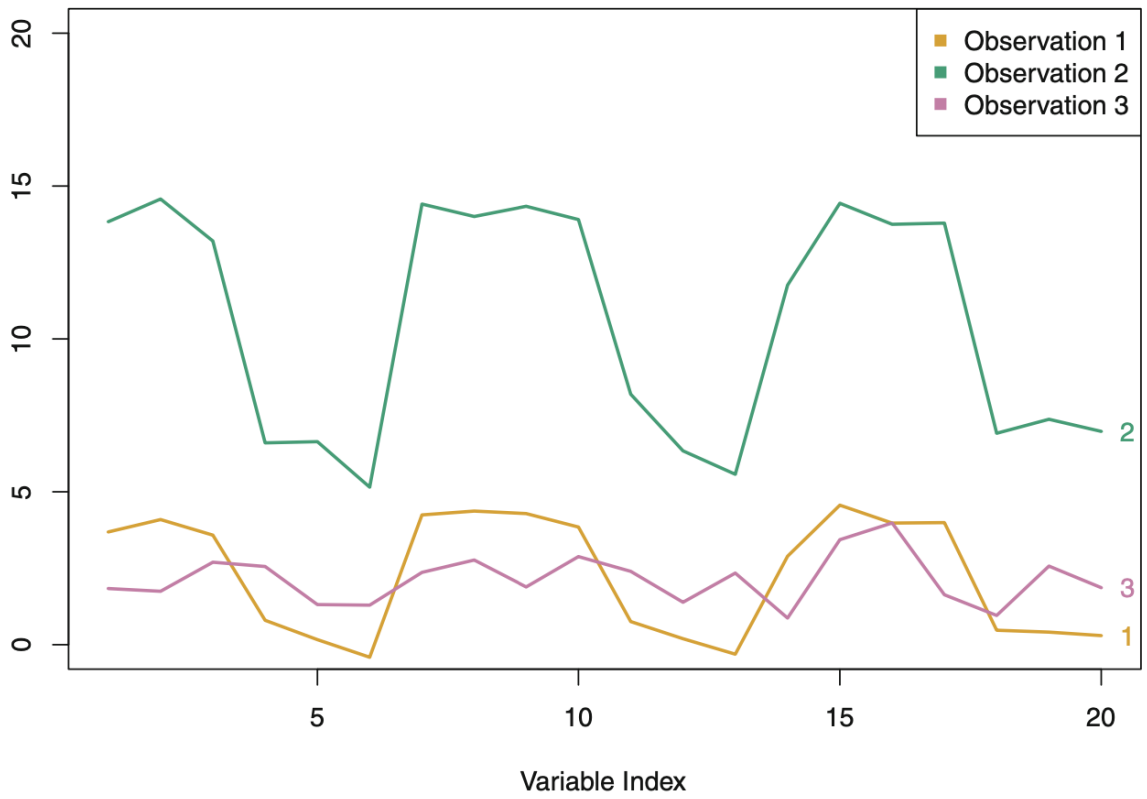
Now, using this similarity matrix, sketch the dendogram that results from hierarchically clustering these 5 observations using complete linkage.

Hierarchical Clustering Algorithm

1. Start from n singletons (a singleton is a subset containing a single observation).
2. Repeat the following until a single set is reached:
   1. Among all the current subsets, merge the two subsets that have the smallest $d(C_i, C_j)$.
   2. Record that $d(C_i, C_j)$.
3. Use a threshold value for $d(C_i, C_j)$ to cut the tree to obtain branches as clusters.

Choice of Dissimilarity Measure

- So far have used Euclidean distance.
- An alternative is correlation-based distance: two observations to are similar if their features are highly correlated.
- Correlation-based distance focuses on the 'shape' of the observation profiles rather than their magnitudes

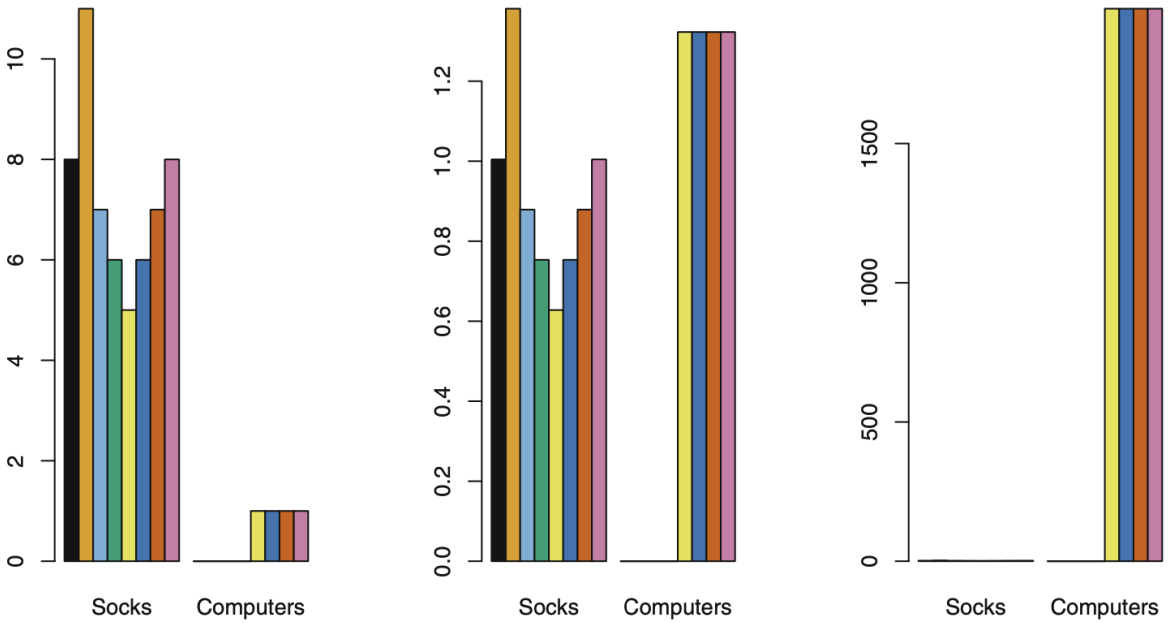


Choice of Dissimilarity Measure example

- An online retailer interested in clustering shoppers based on their past shopping histories.
- To identify subgroups of similar shoppers, for targeted advertisements.
- Data matrix:
  - rows are the shoppers and the columns are the items available for purchase
  - elements of the data matrix = the number of times a given shopper has purchased a given item
- What type of dissimilarity measure should be used?

Choice of Dissimilarity Measure example

- Euclidean distance? Then infrequent shoppers will be clustered together.
- Correlation might be good, except when # of items are large.
- Use price $ instead of purchased items #?



Practical Issues in Clustering

Small Decisions with Big Consequences

- Feature scaling matters
- Which linkage should we use?
- Which dissimilarity measure should we use?
- Where to cut the tree(dendrogram)?
- In K-means, how many clusters should we look for?

Practical Issues in Clustering

Small Decisions with Big Consequences

- Feature scaling matters
- Which linkage should we use?
- Which dissimilarity measure should we use?
- Where to cut the tree(dendrogram)?
- In K-means, how many clusters should we look for?

There is no single right answer for all cases.

Practical Issues in Clustering

Small Decisions with Big Consequences

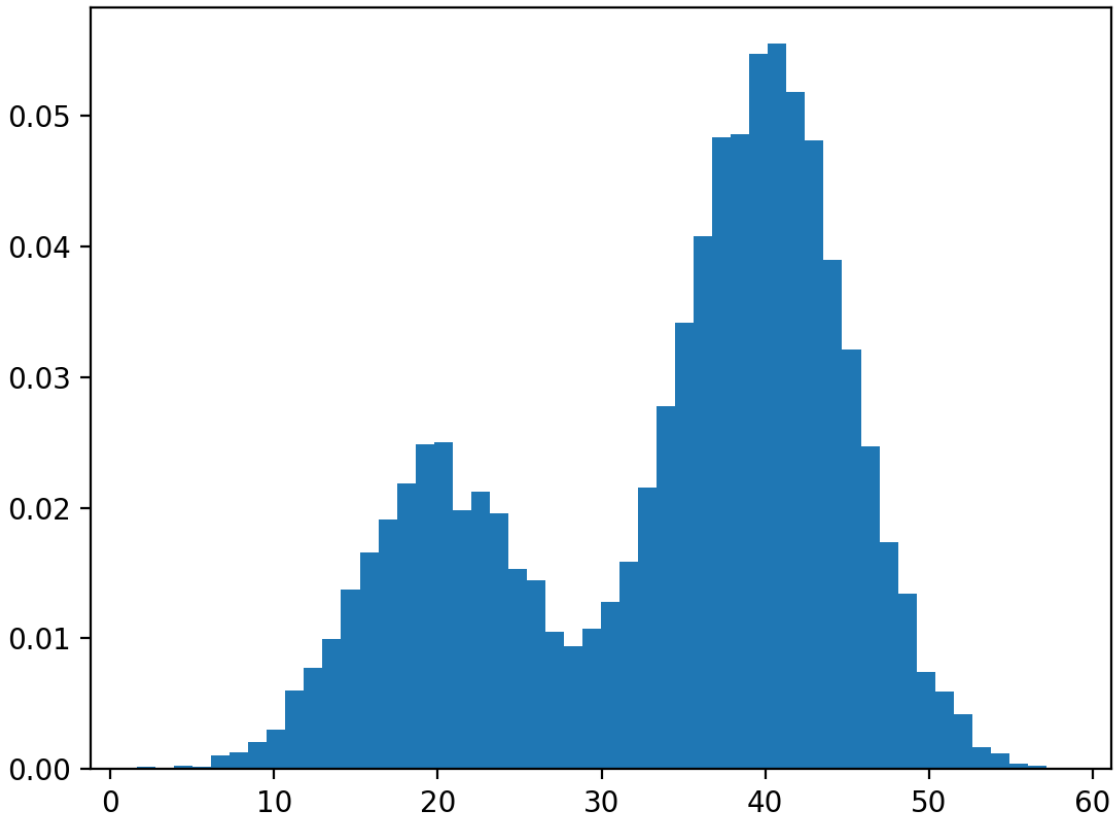
- Feature scaling matters
- Which linkage should we use?
- Which dissimilarity measure should we use?
- Where to cut the tree(dendrogram)?
- In K-means, how many clusters should we look for?

There is no single right answer for all cases.
In practice we can always try several different choices and try to search for the one that's most useful and interpretable.

Clustering using a mixture model

Clustering using a mixture model

Real interview question from an IT company: I accidentally pooled observations from two different normal distributions together, how do I split them?



Clustering using a mixture model

For the mixture model approach, we need to have:

- a family of distributions (with parameters θ) for the clusters.
- a number K, the number of underlying components.

Clustering using a mixture model

In a mixture model, we assume the data are from a mixture of K distributions with unknown parameters θ_k and unknown mixture probabilities π_k:

$$h(y|\phi) = \sum_{k=1}^{K} \pi_k f(y|\theta_k)$$

subject to $\pi_k \geq 0, \sum_{k=1}^{K} \pi_k = 1$
Here, $\phi = (\pi_1, \pi_2, ..., \pi_K, \theta_1, \theta_2, ..., \theta_K)^T$

Clustering using a mixture model

Suppose we can estimate all the parameters (details later), we can calculate the posterior probability for each observation to belong to a class k as:

$$\hat{p}_k = P(k|y, \hat{\phi}) = \frac{\hat{\pi}_k f(y|\hat{\theta}_k)}{\sum_k \hat{\pi}_k f(y|\hat{\theta}_k)}$$

and assign the observation to the underlying class(membership) with the largest posterior probability.

Parameter estimation in mixture model

For parameter estimations in the problem above, an **expectation-maximization (EM) algorithm** can be used.

Parameter estimation in mixture model

For parameter estimations in the problem above, an **expectation-maximization (EM) algorithm** can be used.
EM algorithm is an iterative algorithm, looping through an **E-step** and an **M-step** to update parameter estimates until convergence.
Specifically, after initializing the parameters ϕ, we iterate between the following two steps:

- E-step: Given current $\hat{\phi}$, calculate $\hat{p}_{ik} = P(k|y_i, \hat{\phi})$ for $k = 1, 2, ..., K$ and each observation y_i. Then, update $\pi_k = \frac{1}{n} \sum_{i=1}^{n} \hat{p}_{ik}$
- M-step: Maximize likelihoood $l_k(\theta_k) = \sum_i \hat{p}_{ik} log f(y_i|\theta_k)$ to obtain new $\hat{\theta}_k$

Parameter estimation in mixture model

More math details:
Consider the full data $\{(y_i, g_i)\}$, where g_i is the unobserved class/membership for observation i. The log-likelihood for the full data is $l = \sum_i log f(y_i|\theta_{gi})$.

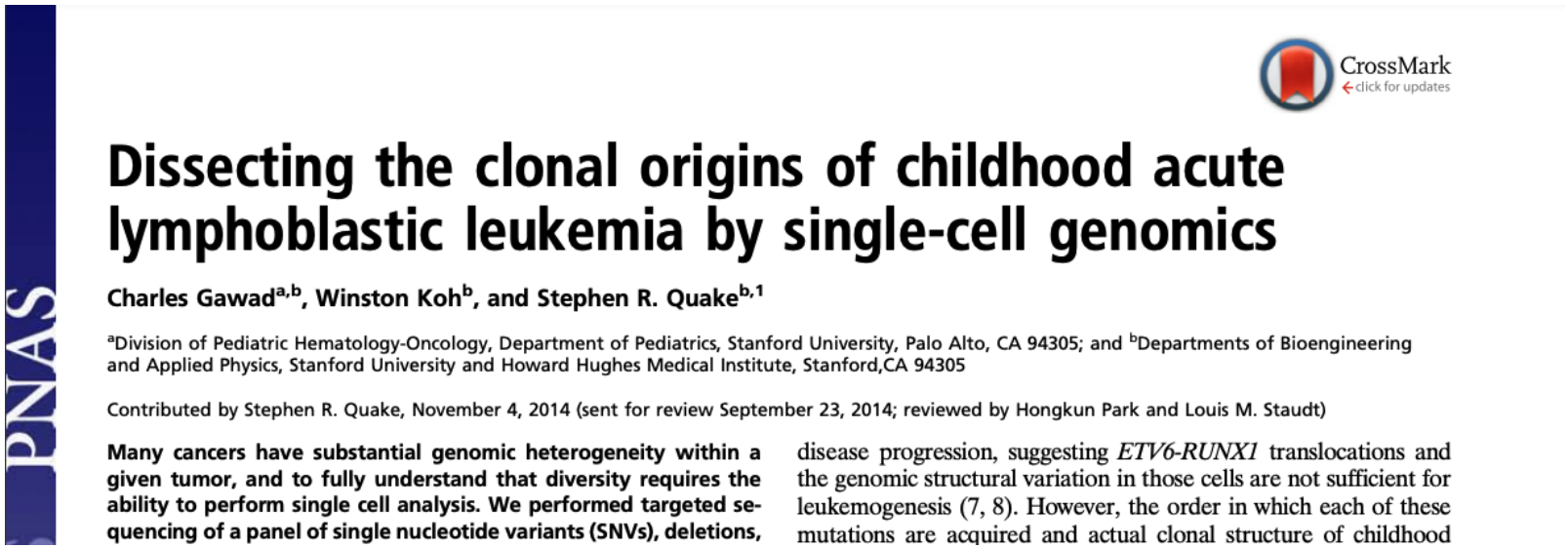
The E-step is to calculate the expectation
$l_E = \sum_i \sum_k \hat{p}_{ik} log f(y_i|\theta_k) = \sum_k l_k(\theta_k)$.

The M-step is to maximize $l_E = \sum_k l_k(\theta_k)$ with respect to all the parameters, which is equivalent to maximizing $l_k(\theta_k)$ for all k.

A Gentle Introduction to Expectation Maximization ▸ Photo by valcker, some rights reserved.

EM algorithm example

Gawad et al. Dissecting the clonal origins of childhood acute lymphoblastic leukemia by single-cell genomics. PNAS. 2014 111:17947–17952.



PNAS

CrossMark click for updates

Dissecting the clonal origins of childhood acute lymphoblastic leukemia by single-cell genomics

Charles Gawad[a,b], Winston Koh[b], and Stephen R. Quake[b,1]

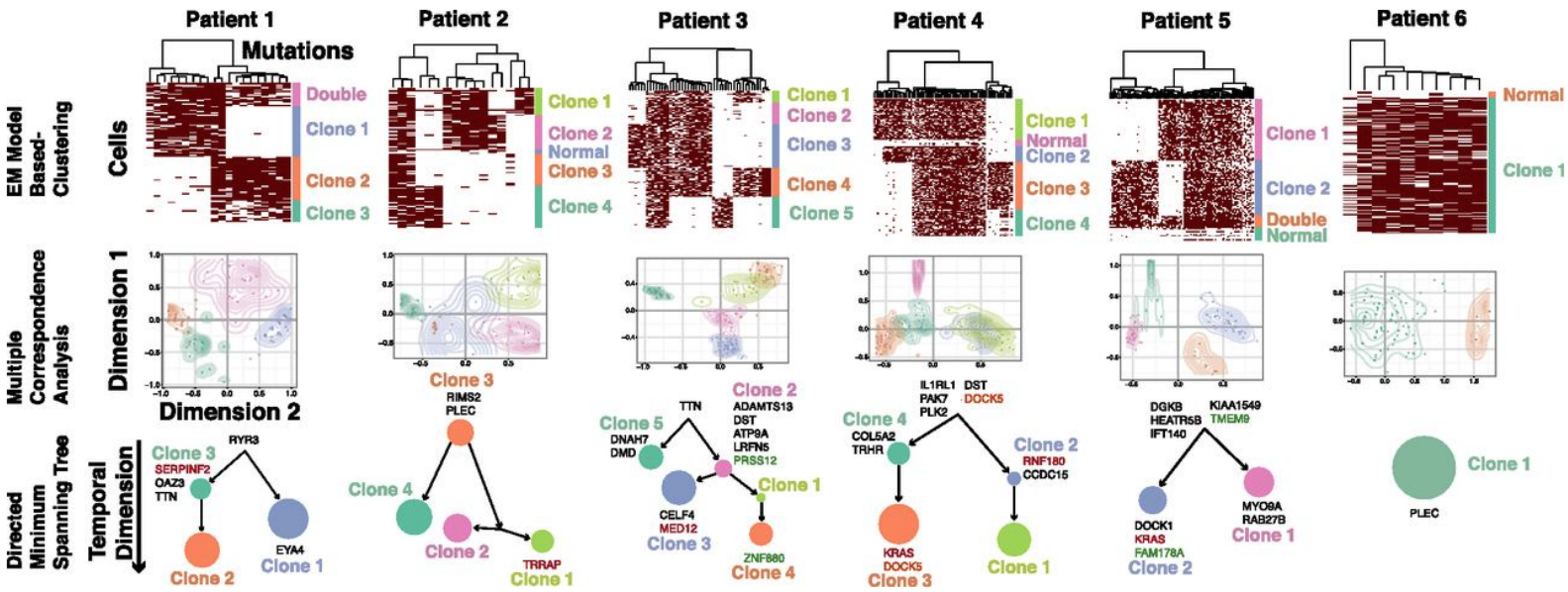
[a]Division of Pediatric Hematology-Oncology, Department of Pediatrics, Stanford University, Palo Alto, CA 94305; and [b]Departments of Bioengineering and Applied Physics, Stanford University and Howard Hughes Medical Institute, Stanford,CA 94305

Contributed by Stephen R. Quake, November 4, 2014 (sent for review September 23, 2014; reviewed by Hongkun Park and Louis M. Staudt)

Many cancers have substantial genomic heterogeneity within a given tumor, and to fully understand that diversity requires the ability to perform single cell analysis. We performed targeted sequencing of a panel of single nucleotide variants (SNVs), deletions,

disease progression, suggesting *ETV6-RUNX1* translocations and the genomic structural variation in those cells are not sufficient for leukemogenesis (7, 8). However, the order in which each of these mutations are acquired and actual clonal structure of childhood

EM algorithm example



EM algorithm on the multivariate Bernoulli distribution for genome mutation data.

Textbook chapters

K-means, Hierarchical clustering:

- ISLR chapter 12: 12.4

EM algorithm:

- Original publication: Dempster, A.P.; Laird, N.M.; Rubin, D.B. (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm". Journal of the Royal Statistical Society, Series B. 39 (1): 1–38. JSTOR 2984875. MR 0501537.