Unsupervised Learning: Clustering; K-Means.

W. Wang[1]

[1]Department of Mathematics
University of Houston

MATH 4323

Clustering.

Clustering refers to a broad set of techniques for finding clusters of observations in a data set.

Goal: partition of observations into distinct groups so that the

- observations **within** each cluster are quite **similar** to each other,

  while

- the observations in **different** clusters are quite **different** from each other.

Question: what does it mean for two or more observations to be **similar** or **different**? How do we quantify **similarity** of observations?

Clustering with domain knowledge

Answer: This is often a domain-specific consideration that must be made based on knowledge of the data being studied. For example,

- Suppose we have a set of n observations, each with p features. The n observations could correspond to tissue samples for patients with breast cancer, and the p features could correspond to measurements collected for each tissue sample (these could be clinical measurements, such as tumor stage or grade, or gene expression measurements).
- We may have a reason to believe that there is some heterogeneity among the n tissue samples; for instance, there might be some unknown subtypes of breast cancer.

Clustering vs. PCA

Both clustering and PCA seek to simplify the data via a small number of summaries, but their mechanisms are different:

- 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.

Object Similarity. Market segmentation.

Example. Presume you have access to a number of measurements (e.g. median household income, occupation, distance from nearest urban area, and so forth) for a large customer database.

Goal: market segmentation. Identify subgroups of people who

- might be more receptive to a particular form of advertising, or
- more likely to purchase a particular product.

Solution: *clustering*. In such case we have:

- a set of n customers, $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$,
- each described by a vector $\mathbf{x}_i \in \mathbb{R}^p$ of p measurements (median household income, occupation, etc)

The task of performing market segmentation amounts to **clustering the observations** in that data set.

K-Means and Hierarchical Clustering.

While there exist a great number of clustering methods, we will focus on two most well-known ones:

- K**-means** clustering: we seek to partition the observations into a **pre-specified** number K of clusters.

- **Hierarchical** clustering: we do not know in advance how many clusters we want; we end up with a tree-like visual representation of the observations, called a **dendrogram**, that allows us to view at once the clusterings obtained for each possible number of clusters, from 1 to n, where n - total # of observations).

K-Means Clustering.

We must first specify the desired number of clusters K; then the ***K*-means** clustering algorithm will assign each observation to exactly one of the K clusters.

In other words, ***K*-means** clustering partitions data set into K distinct, non-overlapping clusters. More formally, presume we have

- n total observations, and
- pre-determined desired number K of clusters.

Then let $C_1, \ldots, C_K$ denote sets containing the indices of the observations that belong to the corresponding cluster (i^{th} observation $\in k^{th}$ cluster $\Leftrightarrow i \in C_k$). These sets satisfy:

1. $C_1 \cup C_2 \cup \cdots \cup C_K = \{1, 2, \ldots, n\} \Leftrightarrow$ each observation i belongs to at least one of the K clusters.
2. $C_k \cap C_{k'} = \varnothing \;\; \forall k \neq k' \Leftrightarrow$ clusters are non-overlapping.

Example. Given $n = 5$ and $K = 2$, examples of cluster solutions are

$$C_1 = \{1, 3, 4\}, \quad C_2 = \{2, 5\}; \quad \text{or} \quad C_1 = \{2, 3, 4, 5\}, \quad C_2 = \{1\};$$

K-Means Clustering, different K values.

Example. $n = 150$ observations, K-Means clustering for $K = 2, 3, 4$.

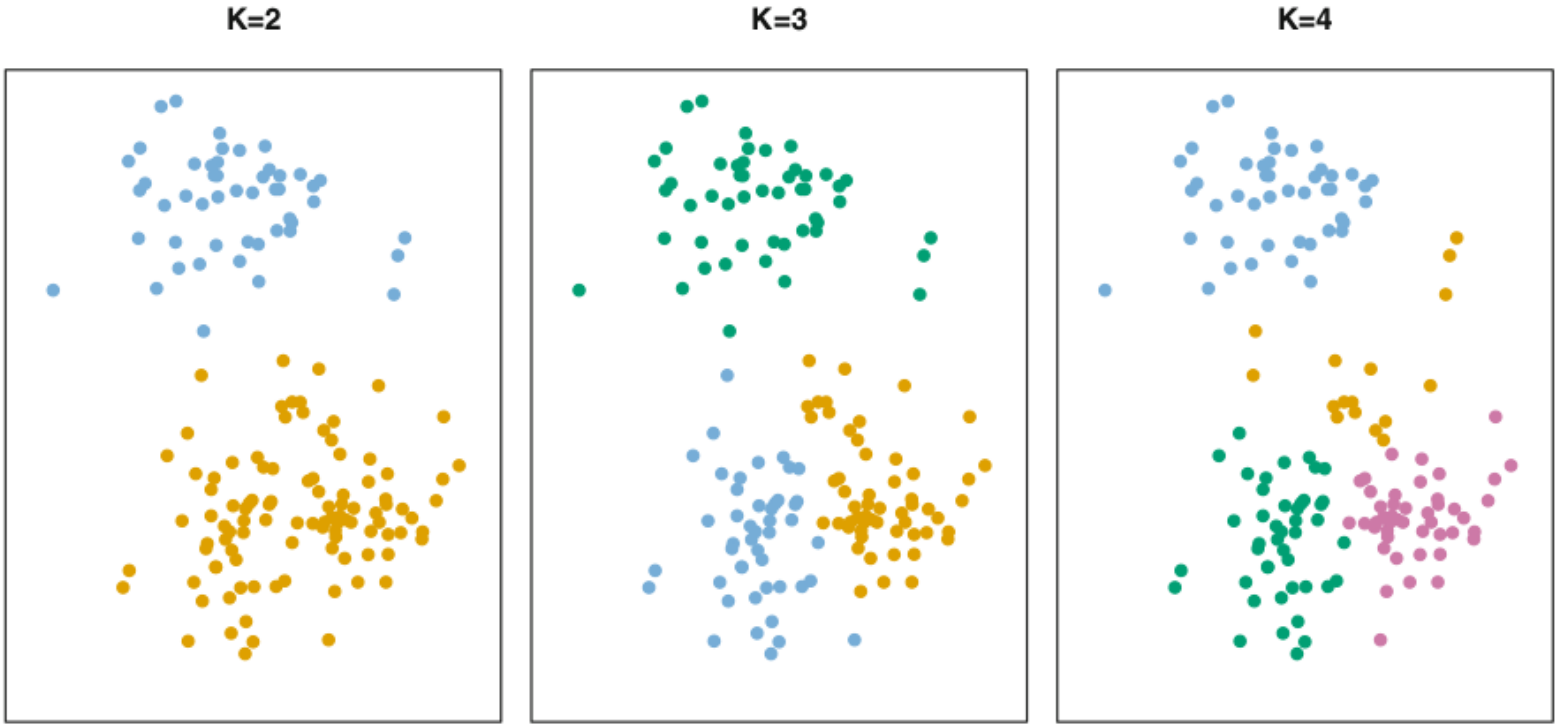


FIGURE 10.5. *A simulated data set with 150 observations in two-dimensional space. Panels show the results of applying K-means clustering with different values of K, the number of clusters. The color of each observation indicates the cluster to which it was assigned using the K-means clustering algorithm. 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: How do we cluster?

Within-cluster variation for cluster C_k is a measure $\boldsymbol{W(C_k)}$ of the amount by which the observations within C_k **differ** from each other.

A good clustering is one for which the **within-cluster** variation is as **small** as possible.

Therefore, we want to find clusters $C_1, \ldots, C_k$ that solve the problem

$$\underset{C_1,\ldots,C_k}{minimize} \{\sum_{k=1}^{K} W(C_k)\} \tag{1}$$

In plain English:

"We want to partition the observations into K clusters such that the total within-cluster variation, summed over all K clusters, is as small as possible."

Within-Cluster Variation: How to measure?

Question: how to define a within-cluster variation of a cluster C_k?

Answer: Most common choice - squared Eucledian distance between all distinct pairs of observations $i, j \in C_k,\ i \neq j$.

E.g. presume $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^\intercal \in \mathbb{R}^p$ and $\mathbf{x}_j = (x_{j1}, \ldots, x_{jp})^\intercal \in \mathbb{R}^p$.
Then squared Eucledian distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ is

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \sum_{r=1}^{p} (x_{ir} - x_{jr})^2$$

Doing it for all distinct pairs $i, j \in C_k$:

$$\sum_{i,j \in C_k,\ i>j} dist(\mathbf{x}_i, \mathbf{x}_j) = \sum_{i,j \in C_k,\ i>j} \sum_{r=1}^{p} (x_{ir} - x_{jr})^2 \tag{2}$$

Within-Cluster Variation: How to measure?

To make (2) into a final within-cluster variation **measure $W(C_k)$**, we also normalize it by the cluster size $|C_k| \equiv \{\#\text{ of observations} \in C_k\}$:

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,j \in C_k,\ i>j} \sum_{r=1}^{p} (x_{ir} - x_{jr})^2 \tag{3}$$

Combining (3) with (1) gives us the main optimization task of **clustering**:

$$\underset{C_1,\ldots,C_k}{\textit{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,j \in C_k,\ i>j} \sum_{r=1}^{p} (x_{ir} - x_{jr})^2 \right\} \tag{4}$$

Question: how do we find the clusters $C_1, \ldots, C_K$ that solve (4)?

K-Means **algorithm.**

K-Means Algorithm for Clustering has the following steps:

1. (**Random Initialization.**) Randomly assign a number, from 1 to K, to each of the observations. These serve as **initial** cluster assignments for the observations.
2. (**Iterative Updates.**) Iterate until the cluster assignments stop changing:
   (a) For each of the clusters $C_1, \ldots, C_K$, compute the cluster **centroid**. The kth cluster centroid is the vector of the p feature means for the observations in the kth cluster:

$$\bar{\mathbf{x}}^k = \frac{1}{|C_k|}\sum_{i\in C_k}\mathbf{x}_i = \frac{1}{|C_k|}\sum_{i\in C_k}(x_{i1}, x_{i2}, \ldots, x_{ip})^\intercal = (\bar{x}^k_{.1}, \bar{x}^k_{.2}, \ldots, \bar{x}^k_{.p})^\intercal$$

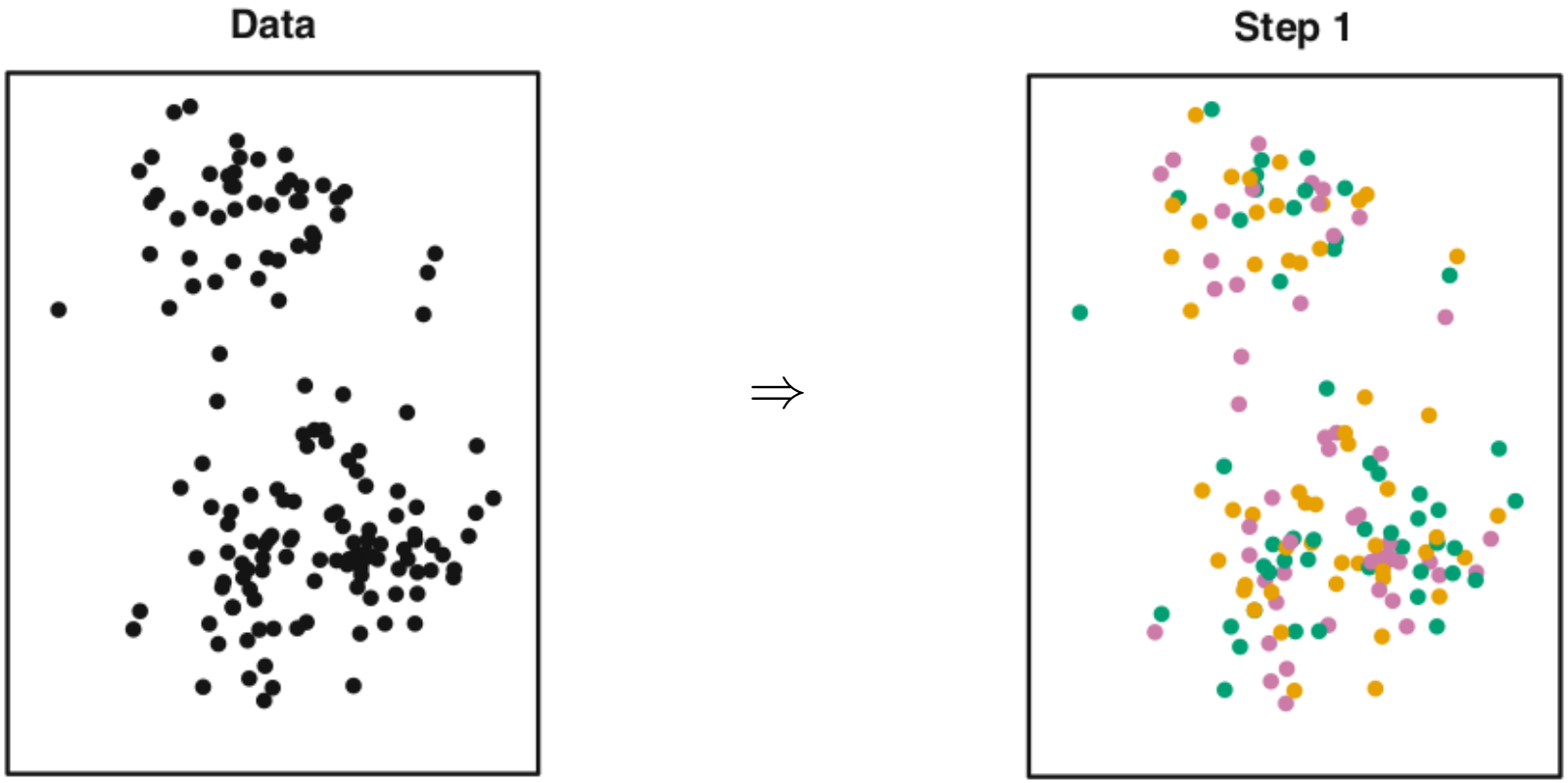
   where $\bar{x}^k_{.r} = \frac{1}{|C_k|}\sum_{i\in C_k} x_{ir}$ is the mean of the r^{th} feature for all observations in k^{th} cluster, $r = 1, \ldots, p$.

   (b) Assign each observation to the cluster whose centroid is **closest** ("closeness" is defined using Euclidean distance).

K-Means algorithm: Toy example demonstration.

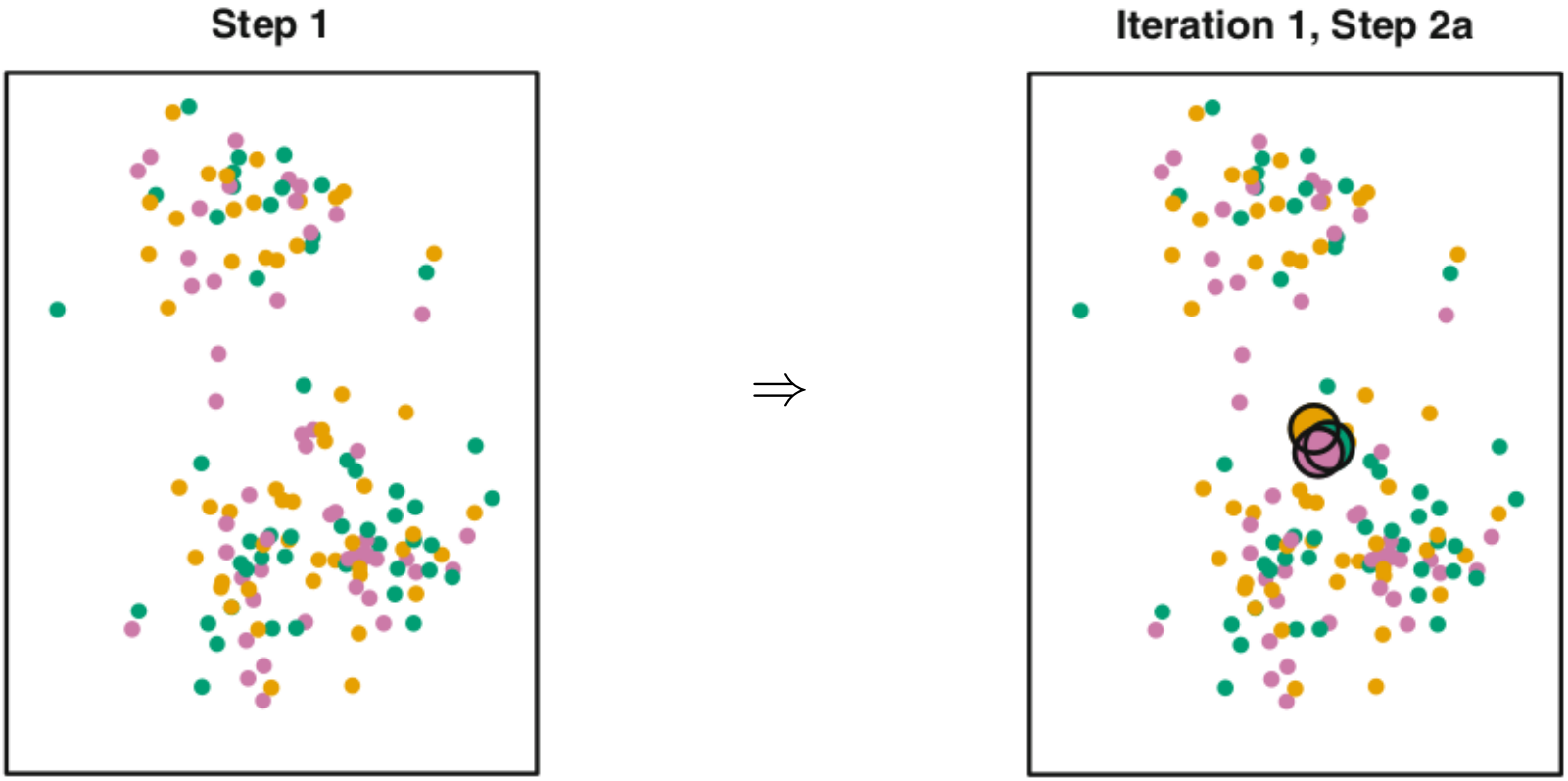
Example (cont'd). To continue with our toy example of $n = 150$ observations and $K = 3$ clusters:

Step 1. Random initialization of cluster assignments (clusters 1, 2, 3) for all $n = 150$ points.



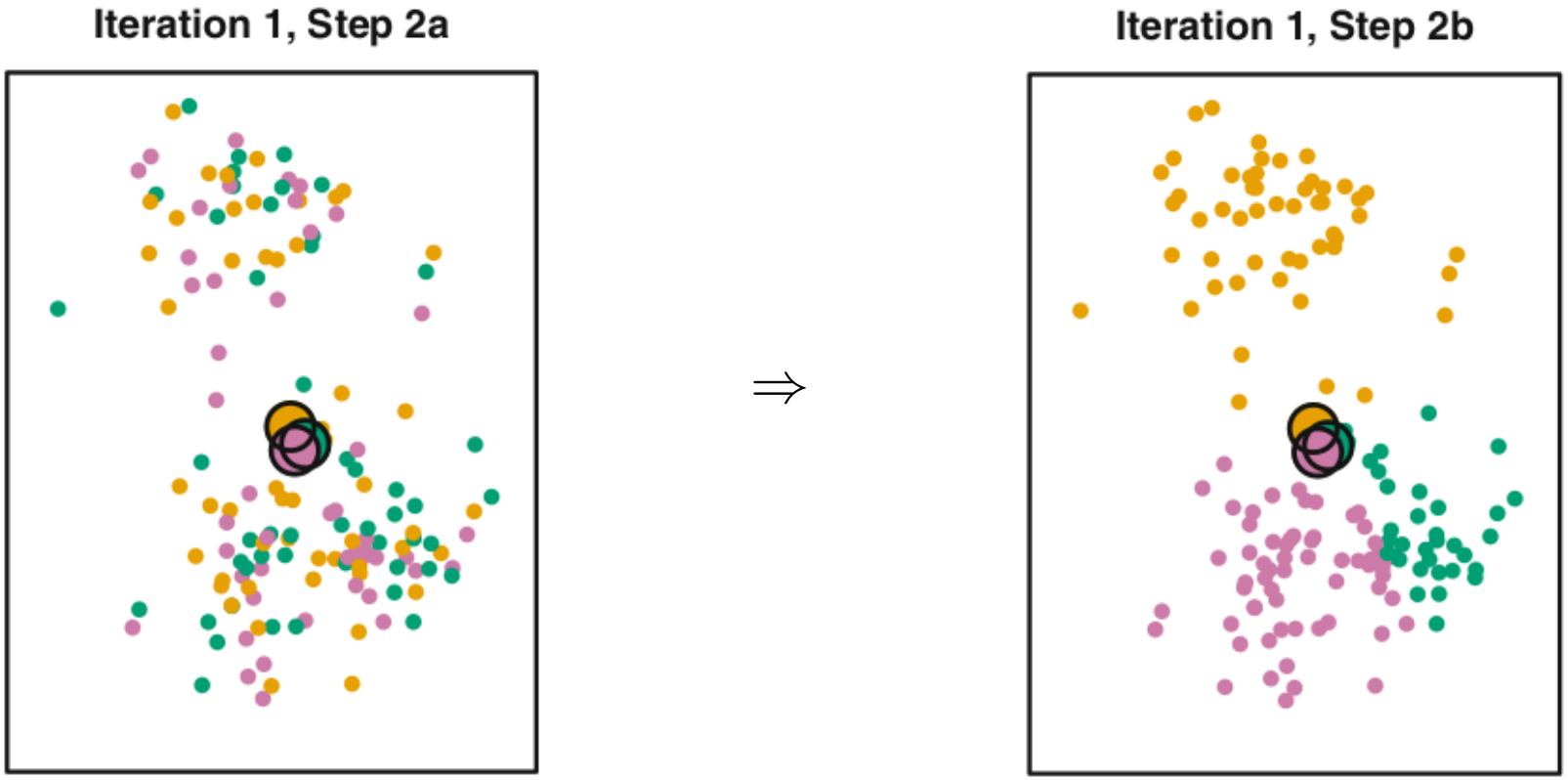
K-Means algorithm: Toy example demonstration.

Iteration #1**, Step** 2(a)**.** Calculating centroids of each cluster.



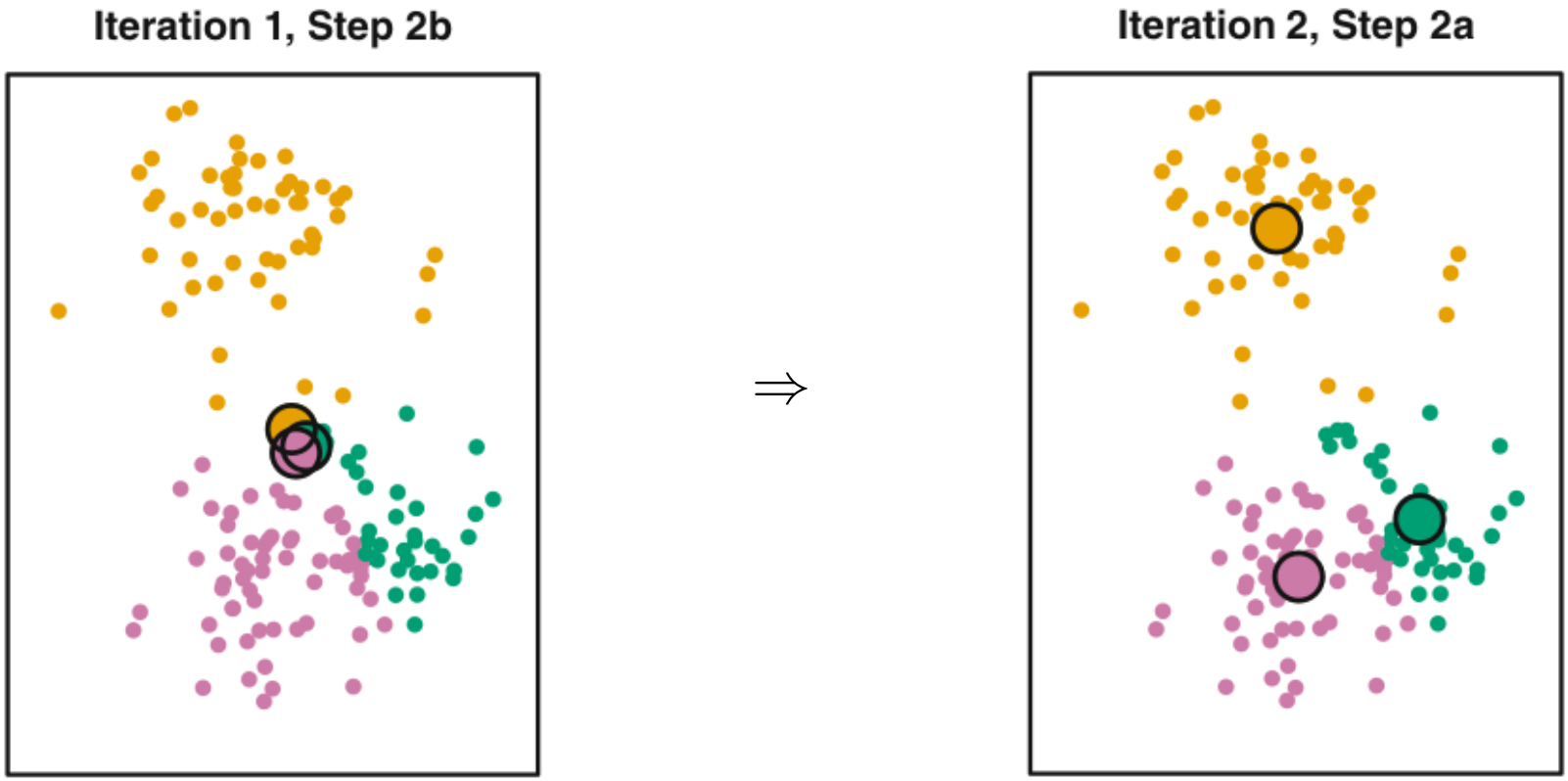
K-Means algorithm: Toy example demonstration.

Iteration #1**, Step** 2(b)**.** Re-assigning each observation to the cluster whose centroid is closest.



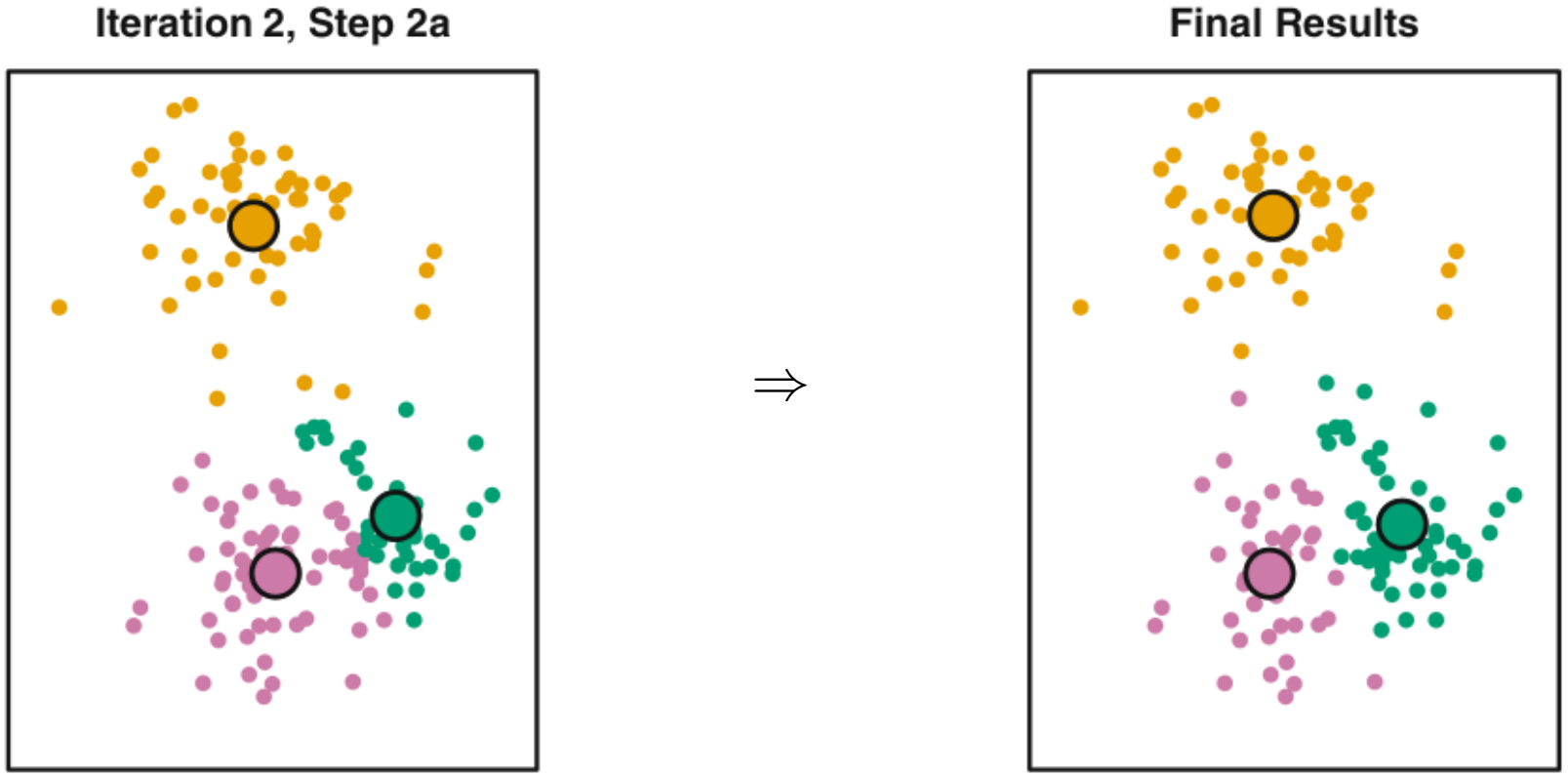
K-Means algorithm: Toy example demonstration.

Iteration #2**, Step** 2(a)**.** Calculating centroids of each cluster.



K-Means algorithm: Toy example demonstration.

Final Result. Final cluster assignment after the algorithm stopped improving the function (4).



K-Means **algorithm.**

K-Means algorithm is guaranteed to decrease the value of the objective (4) at each step, due to the following identity:

$$\frac{1}{|C_k|} \sum_{i,j \in C_k,\ i>j} \sum_{r=1}^{p} (x_{ir} - x_{jr})^2 \equiv 2 \sum_{i \in C_k} \sum_{r=1}^{p} (x_{ir} - \bar{x}_{.r}^{k})^2,$$

Given the logic of step 2(b), where we update the cluster assignment of observation i to the cluster k^{upd} whose centroid is the closest:

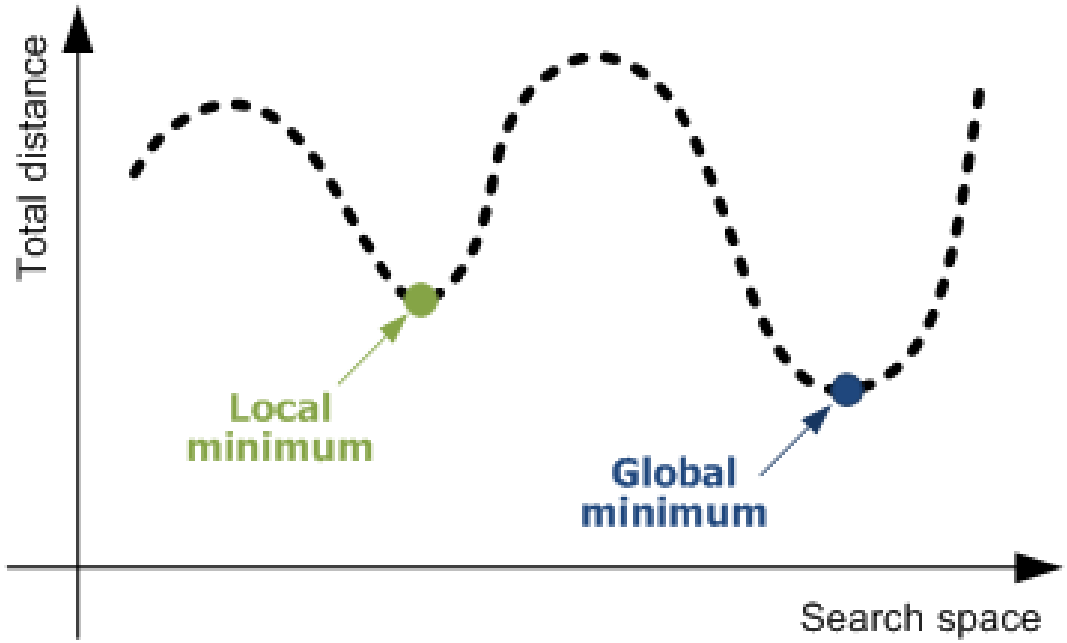
$$k^{upd} = \underset{k}{argmin}\ dist(\mathbf{x}_i, \bar{\mathbf{x}}_k) = \underset{k}{argmin} \sum_{r=1}^{p} (x_{ir} - \bar{x}_{.r}^{k})^2$$

$$\Longrightarrow \sum_{r=1}^{p} (x_{ir} - \bar{x}_{.r}^{k^{upd}})^2 \leq \sum_{r=1}^{p} (x_{ir} - \bar{x}_{.r}^{k})^2,\ i = 1, \ldots, n$$

$$\Longrightarrow \sum_{k=1}^{K} [2 \sum_{i \in C_k^{upd}} \sum_{r=1}^{p} (x_{ir} - \bar{x}_{.r}^{k^{upd}})^2] \leq \sum_{k=1}^{K} [2 \sum_{i \in C_k} \sum_{r=1}^{p} (x_{ir} - \bar{x}_{.r}^{k})^2],$$

Local Optima: Multiple Random Initialization.

Issue: K-means algorithm finds a **local** rather than a **global** optimum $\Longrightarrow$ final solution depends on the initial (random) cluster assignment in Step 1 of K-Means algorithm.



Local Optima: Multiple Random Initialization.

Work-around: run the algorithm multiple times from different random initial configurations.

1. Run K-Means algorithm multiple B times, each time for a new random cluster assignment in Step 1.
2. Selects the best solution out of those B runs, i.e. for which the **within-cluster** variation is smallest.

Local Optima: Multiple Random Initializations.

Example (cont'd). Next slide shows the local optima clustering solutions obtain for our toy example by running K-means clustering $B = 6$ times, using six different initial cluster assignments.

Question: Across 6 runs, what was the optimum **within-cluster** variation value?

Question: Do different initializations necessarily lead to different final clustering solutions? E.g. are the solutions $\#2, 3, 4$ and 5 actually different?

Local Optima: Multiple Random Initializations.

Details of Previous Figure

- K-means clustering performed six times on the data from previous figure 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 (4).
- 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.

Example: Human Tumor Microarray Data

Example. We apply K-means clustering to the aforementioned human tumor microarray data example, with

- $n = 64$ tissue samples,
- $p = 6830$ genes (variables)

We will use K-Means for $K = 3$ to cluster the samples, each of which is a vector of length 6830, representing gene expression measurements.

Also, each sample has a label such as breast (for breast cancer), melanoma, and so on; **BUT** we don't use these labels in the clustering. We will just examine **posthoc** which labels fall into which clusters.

K-Means in *R*: *eclust*() function of *factoextra* library.

In *R*, library *factoextra* contains function *eclust*(), which allows one to conduct a big variety of clustering methods, including *K*-Means and hierarchical clustering.

Usage

```
eclust(x, FUNcluster = c("kmeans", "pam", "clara", "fanny", "hclust", "agnes",
  "diana"), k = NULL, k.max = 10, stand = FALSE, graph = TRUE,
  hc_metric = "euclidean", hc_method = "ward.D2", gap_maxSE = list(method
  = "firstSEmax", SE.factor = 1), nboot = 100, verbose = interactive(),
  seed = 123, ...)
```

Task: Study the arguments of the function on your own (type in ?*eclust* after loading the *factoextra* library). Play around with different values, while applying it to this microarray data set.

K-Means in *R*: *eclust*() function of *factoextra* library.

Example (cont'd). To apply *K*-Means for our microarray data set:

```
> library(ISLR)
> dim(NCI60$data)
>
> library(factoextra)
> km.obj <- eclust(NCI60$data,
                   FUNcluster = "kmeans",
                   k=3,
                   nstart=50)
```

Microarray Example: Cluster Summaries for $K = 3$.

Example (cont'd). Matching up cluster assignments with cancer types:

```
> table(NCI60$labs[km.obj$cluster==1])
  BREAST       CNS MELANOMA    NSCLC  OVARIAN PROSTATE
       3         5         1         7         6         2
   RENAL   UNKNOWN
       9         1
> table(NCI60$labs[km.obj$cluster==2])
     BREAST       COLON K562A-repro K562B-repro    LEUKEMIA
          2           7           1           1           6
MCF7A-repro MCF7D-repro       NSCLC
          1           1           2
> table(NCI60$labs[km.obj$cluster==3])
  BREAST MELANOMA
       2         7
```