Unsupervised Learning

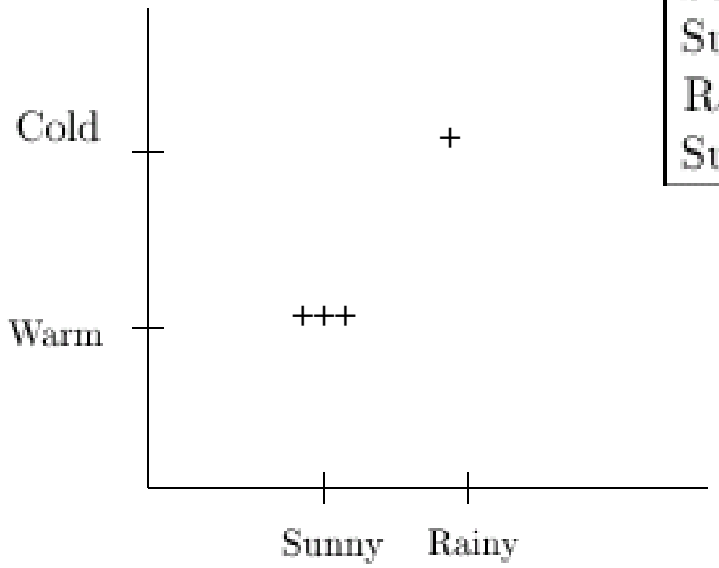
Supervised learning vs. unsupervised learning

- Supervised learning: discover patterns in the data that relate data attributes with a target (class) attribute.
  - These patterns are then utilized to predict the values of the target attribute in future data instances.
- Unsupervised learning: The data have no target attribute.
  - We want to explore the data to find some intrinsic structures in them.

Clustering

- Clustering is a technique for finding similar groups in data, called **clusters**. i.e.,
  - It groups data instances that are similar to (near) each other in one cluster and data instances that are very different (far away) from each other into different clusters.
- Clustering is often called an **unsupervised learning** task as no class values denoting an *a priori* grouping of the data instances are given, which is the case in supervised learning.

An illustration

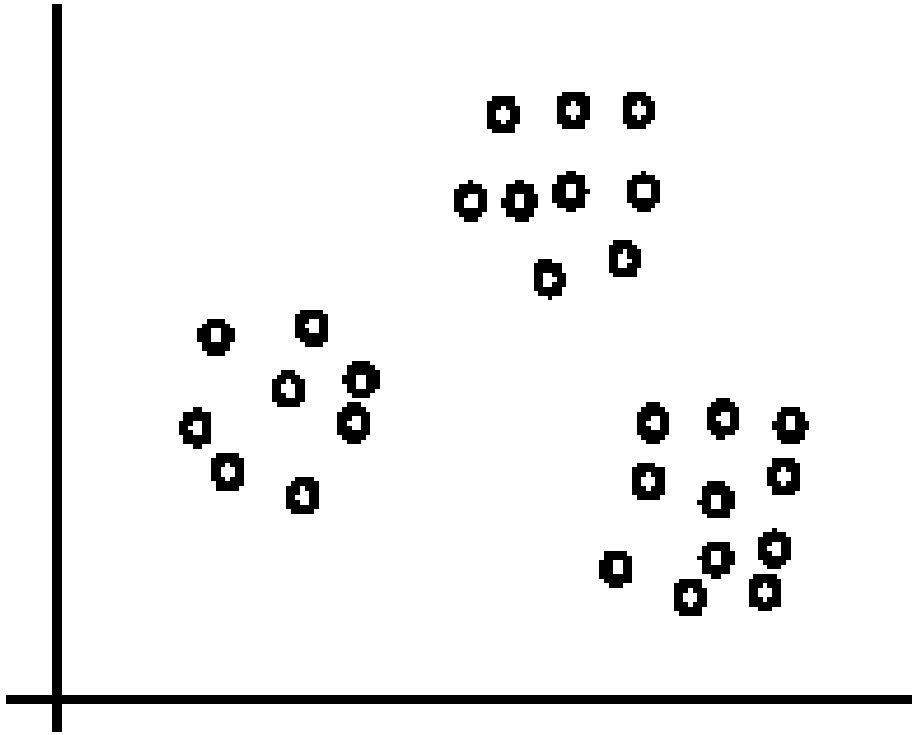


Sky	Temp
Sunny	Warm
Sunny	Warm
Rainy	Cold
Sunny	Warm

We see that there are two natural groups or “clusters”

An illustration

- The data set has three natural groups of data points, i.e., 3 natural clusters.



What is clustering for?

- Let us see some real-life examples
- Example 1: groups people of similar sizes together to make “small”, “medium” and “large” T-Shirts.
  - Tailor-made for each person: too expensive
- Example 2: In marketing, segment customers according to their similarities
- To do targeted marketing.

What is clustering for? (cont…)

- Example 3: Given a collection of text documents, we want to organize them according to their content similarities,
  - To produce a topic hierarchy
- In fact, clustering is one of the most utilized data mining techniques.
  - It has a long history, and used in almost every field, e.g., medicine, psychology, botany, sociology, biology, archeology, marketing, insurance, libraries, etc.
  - In recent years, due to the rapid increase of online documents, text clustering becomes important.

Aspects of clustering

- A clustering algorithm
  - Partitional clustering
  - Hierarchical clustering
  - Density based clustering
  - ---
- A distance (similarity, or dissimilarity) function
- Clustering quality
  - Inter-clusters distance $\Rightarrow$ maximized
  - Intra-clusters distance $\Rightarrow$ minimized

K-means clustering

- K-means is a partitional clustering algorithm
- Let the set of data points (or instances) D be

  $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$,

  where $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ir})$ is a vector in a real-valued space $X \subseteq R^r$, and r is the number of attributes (dimensions) in the data.
- The k-means algorithm partitions the given data into k clusters.
  - Each cluster has a cluster **center**, called **centroid**.
  - k is specified by the user

K-means algorithm

- Given *k*, the *k-means* algorithm works as follows:
  1) Randomly choose *k* data points (seeds) to be the initial centroids, cluster centers
  2) Assign each data point to the closest centroid
  3) Re-compute the centroids using the current cluster memberships.
  4) If a convergence criterion is not met, go to 2).

K-means algorithm – (cont …)

Algorithm *k*-means(*k*, *D*)

1 Choose *k* data points as the initial centroids (cluster centers)
2 **repeat**
3 **for** each data point $\mathbf{x} \in D$ do
4 compute the distance from **x** to each centroid;
5 assign **x** to the closest centroid // a centroid represents a cluster
6 **endfor**
7 re-compute the centroids using the current cluster memberships
8 **until** the stopping criterion is met

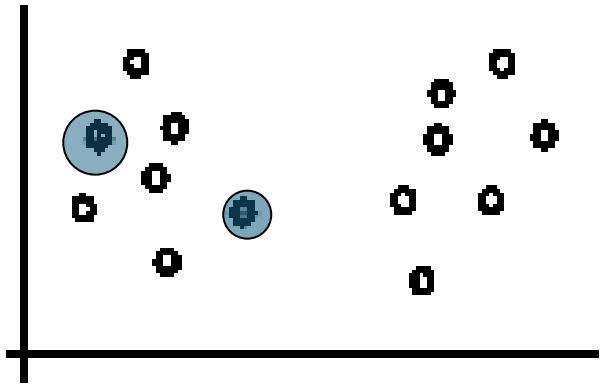
Stopping/convergence criterion

1. no (or minimum) re-assignments of data points to different clusters,
2. no (or minimum) change of centroids, or
3. minimum decrease in the **sum of squared error** (SSE),

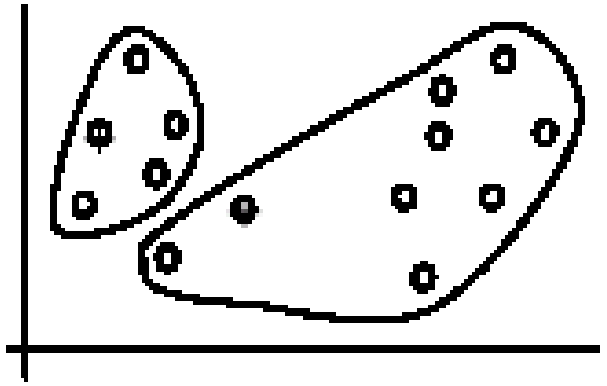
$$SSE = \sum_{j=1}^{k} \sum_{\mathrm{x} \in C_j} dist(\mathrm{x}, \mathrm{m}_j)^2$$

C_j is the j^{th} cluster, $\mathbf{m}_j$ is the centroid of cluster C_j (the mean vector of all the data points in C_j), and $dist(\mathbf{x}, \mathbf{m}_j)$ is the distance between data point **x** and centroid $\mathbf{m}_j$.

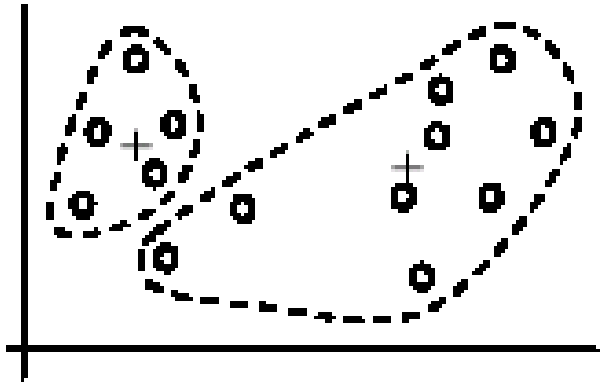
An example



(A). Random selection of k centers

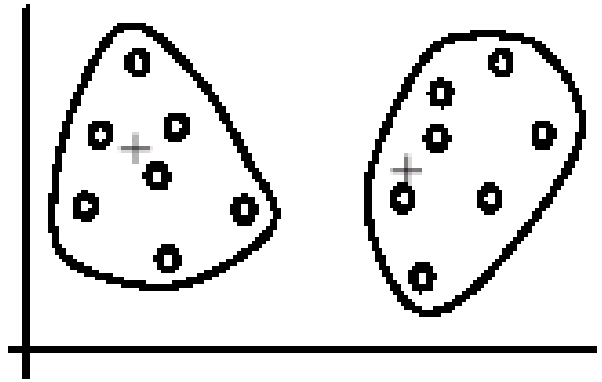


Iteration 1: (B). Cluster assignment

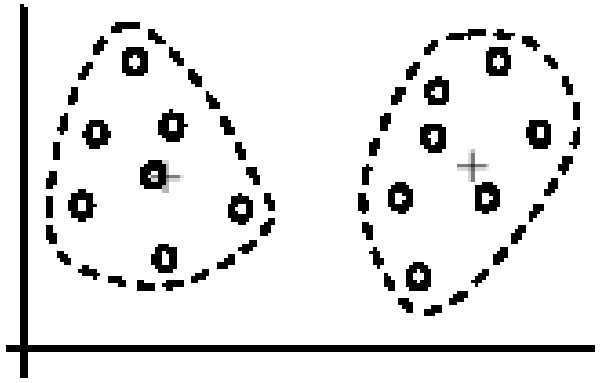


(C). Re-compute centroids

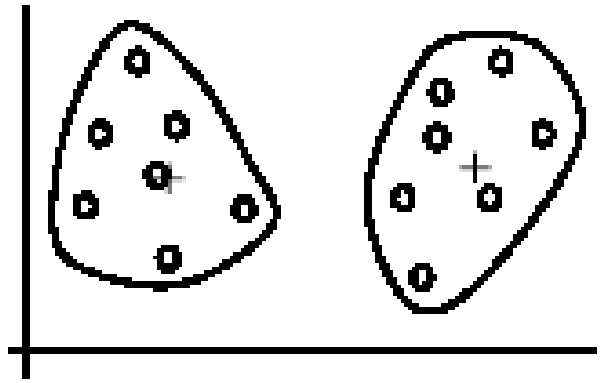
An example (cont …)



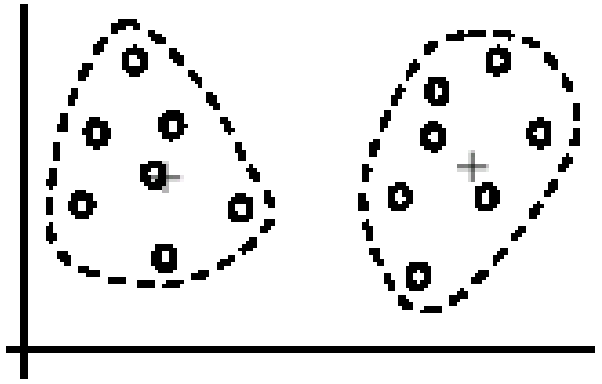
Iteration 2: (D). Cluster assignment



(E). Re-compute centroids



Iteration 3: (F). Cluster assignment



(G). Re-compute centroids

An example distance function

- The k-means algorithm can be used for any application data set where the **mean** can be defined and computed. The mean of cluster is computed with:

$$\mathbf{m}_j = \frac{1}{|C_j|}\sum_{\mathbf{x}_i \in C_j} \mathbf{x}_i$$

- Where $|C_j|$ is the number of points in the cluster C_j. the distance from one point x_i to a mean m_j (centroid) is computed with:

$$dist(\mathbf{x}_i, \mathbf{m}_j) = \sqrt{(x_{i1} - m_{j1})^2 + (x_{i2} - m_{j2})^2 + \ldots + (x_{ir} - m_{jr})^2}$$

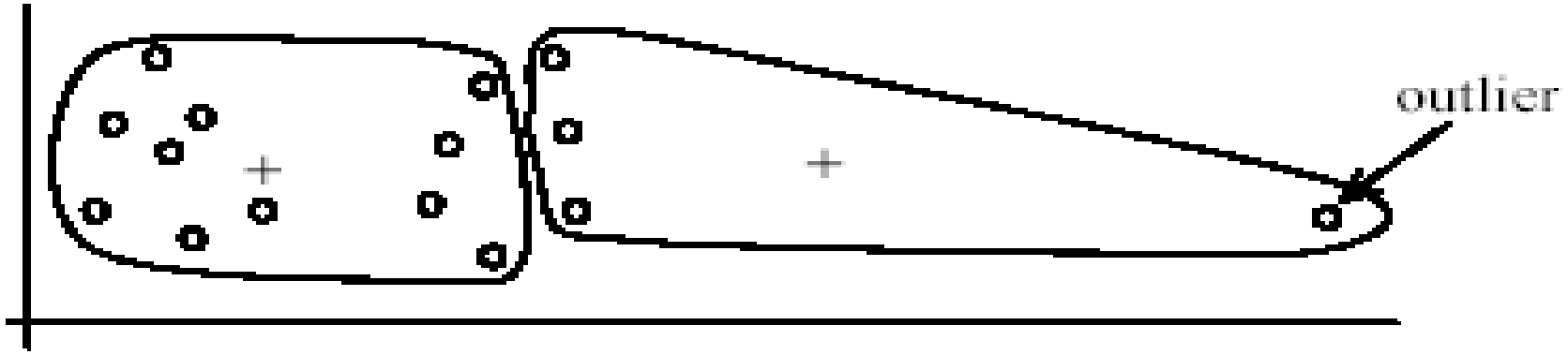
Strengths of k-means

- Strengths:
  - Simple: easy to understand and to implement
  - Efficient: Time complexity: $O(tkn)$,
    where n is the number of data points,
    k is the number of clusters, and
    t is the number of iterations.
  - Since both k and t are small. k-means is considered a linear algorithm.
- K-means is the most popular clustering algorithm.

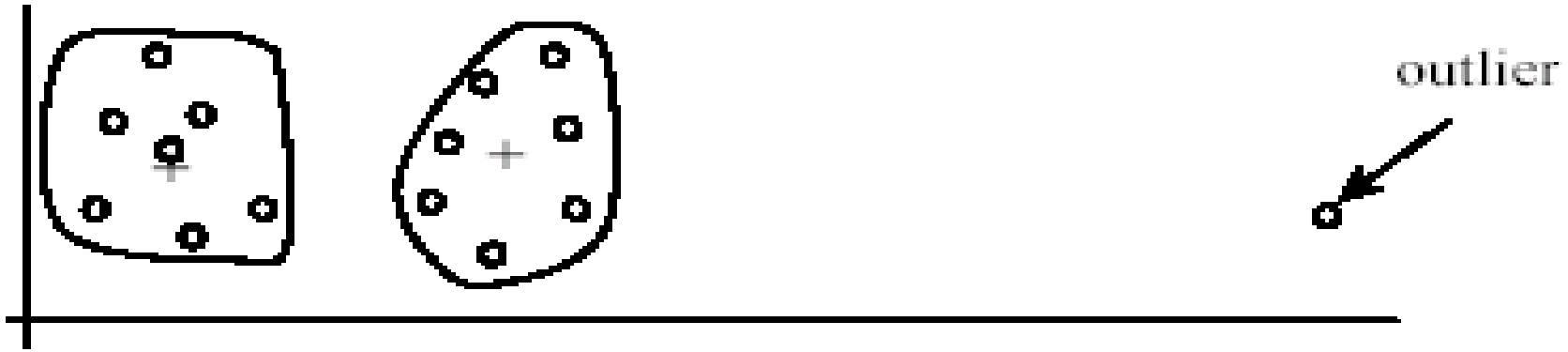
Weaknesses of k-means

- The algorithm is only applicable if the mean is defined.
  - For categorical data, *k*-mode - the centroid is represented by most frequent values.
- The user needs to specify *k*.
- The algorithm is sensitive to **outliers**
  - Outliers are data points that are very far away from other data points.
  - Outliers could be errors in the data recording or some special data points with very different values.

Weaknesses of k-means: Problems with outliers



(A): Undesirable clusters



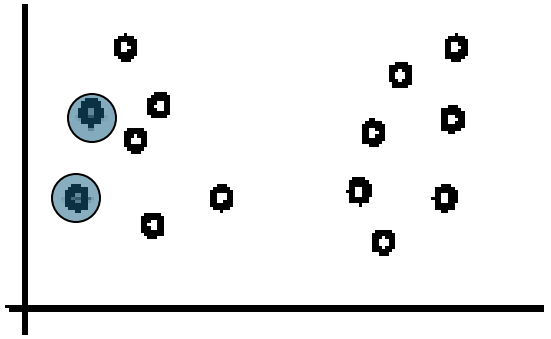
(B): Ideal clusters

Weaknesses of k-means: To deal with outliers

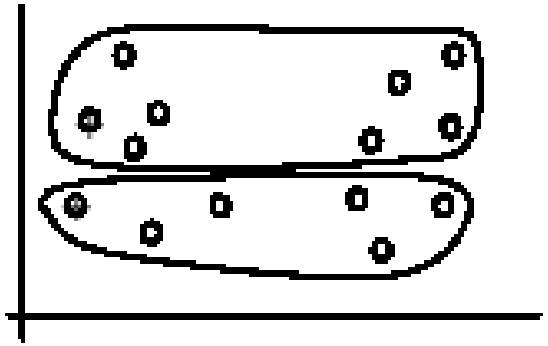
- One method is to remove some data points in the clustering process that are much further away from the centroids than other data points.
  - To be safe, we may want to monitor these possible outliers over a few iterations and then decide to remove them.
- Another method is to perform random sampling. Since in sampling we only choose a small subset of the data points, the chance of selecting an outlier is very small.
  - Assign the rest of the data points to the clusters by distance or similarity comparison

Weaknesses of k-means (cont …)

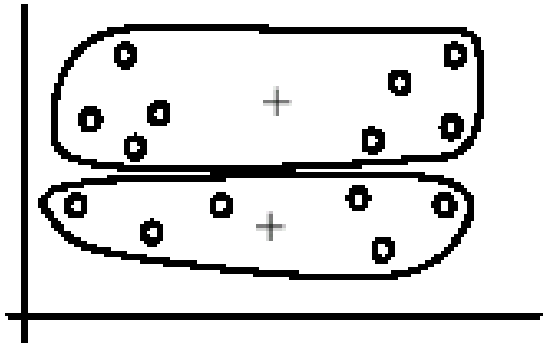
- The algorithm is sensitive to initial seeds.



(A). Random selection of seeds (centroids)



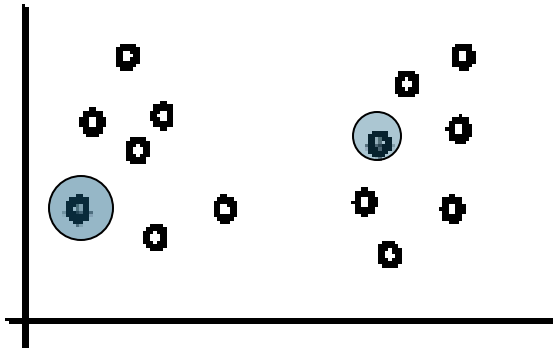
(B). Iteration 1



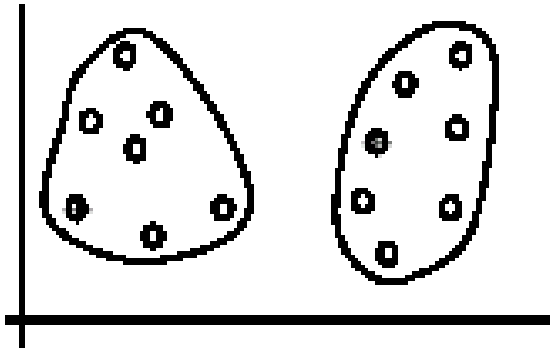
(C). Iteration 2

Weaknesses of k-means (cont …)

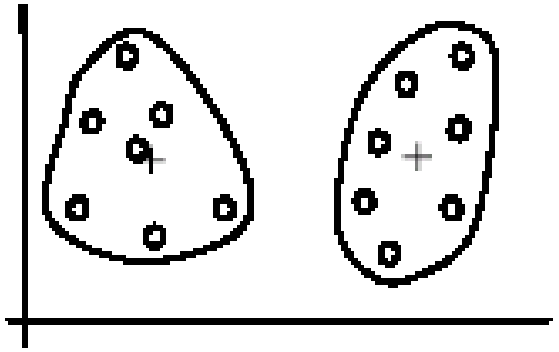
- If we use different seeds: good results



(A). Random selection of k seeds (centroids)



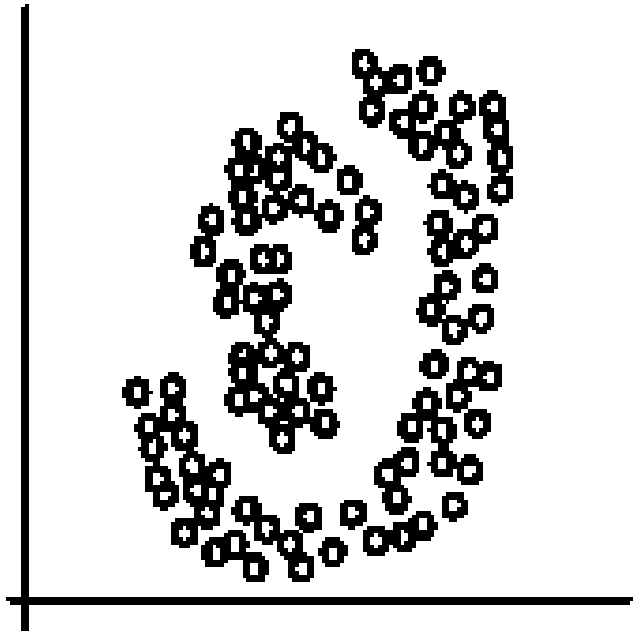
(B). Iteration 1



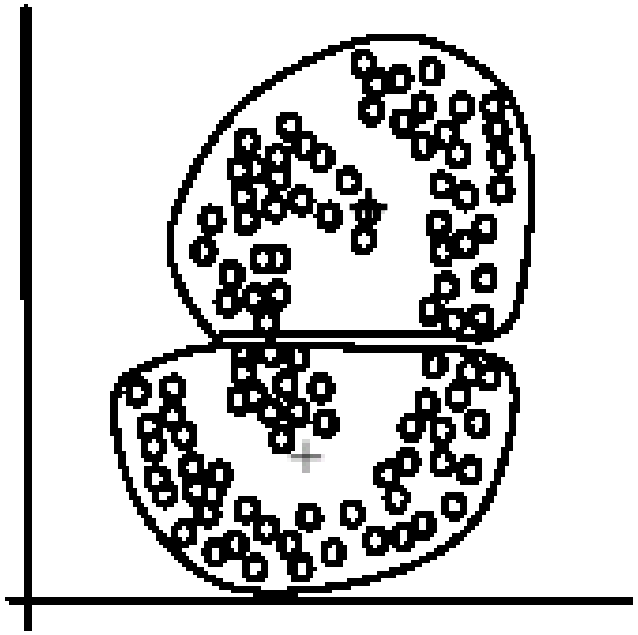
(C). Iteration 2

Weaknesses of k-means (cont …)

- The *k*-means algorithm is not suitable for discovering clusters that are not hyper-ellipsoids (or hyper-spheres).



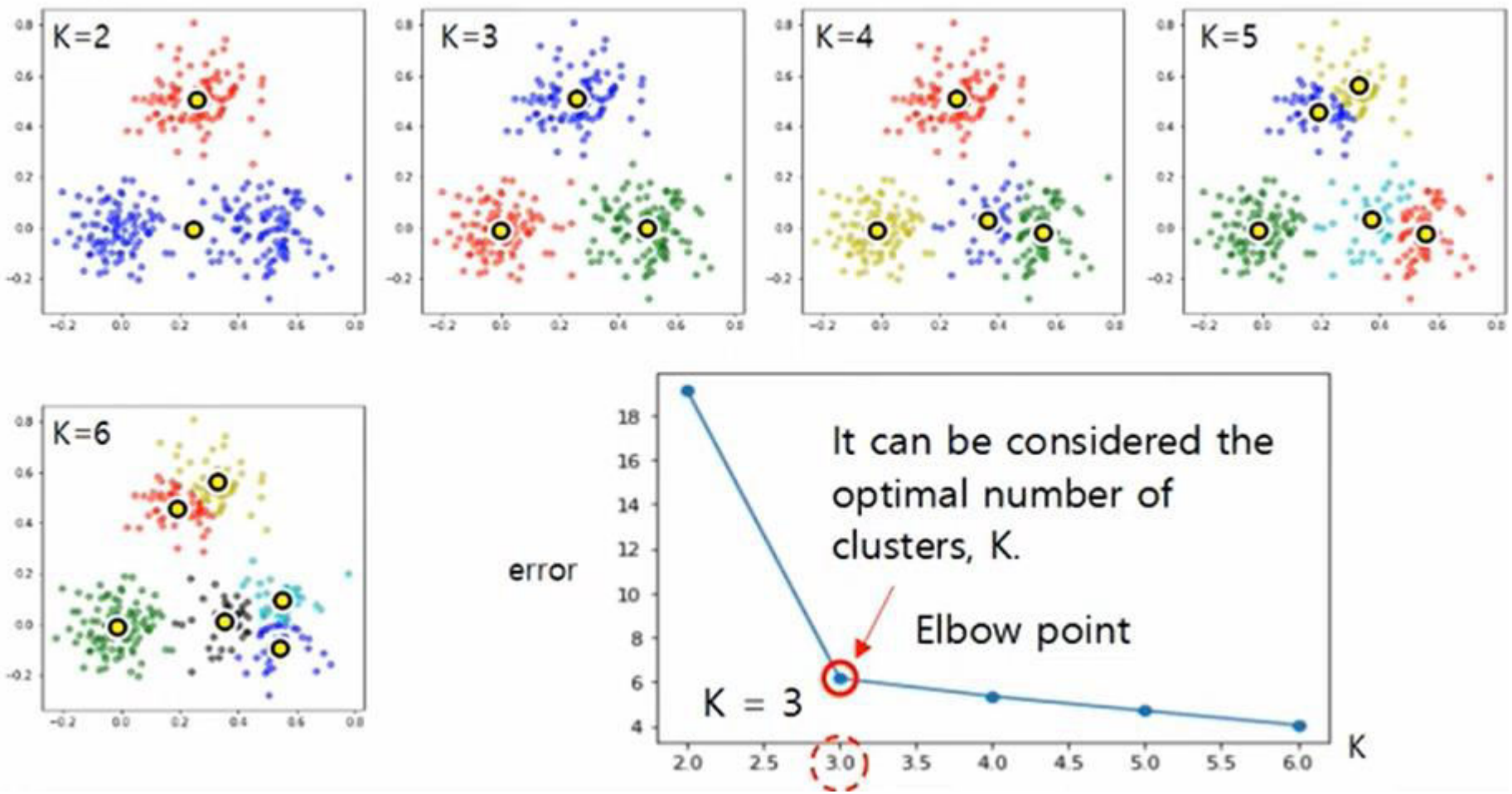
(A): Two natural clusters



(B): *k*-means clusters

Finding the optimal number of clusters – Elbow method

- The elbow method is a graphical method to find the optimal number of clusters, K in K-Means.
- Perform K-Means by varying K value and measure the errors. In the example below, if you set K small at first and gradually increase K, the error will gradually become smaller. At first the error decreases quickly, but later the error decreases slowly. In the error graph below, the point where the graph forms an elbow is likely to be the optimal K value.

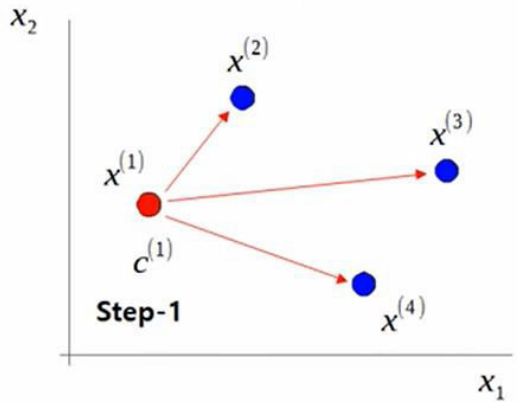


K-means++ Algorithm

- Standard K-Means requires multiple attempts to solve the local minimum problem. This problem typically occurs when randomly set initial centroids are close to each other. Properly distributing the initial centroids can reduce the likelihood of this problem occurring.

- **K-Means++ algorithm**:
- At any given time, let D(x) denote the shortest distance from a data point x to the closest centroid we have already chosen.
  (1) Choose an initial centroid $c^{(1)}$ uniformly at random from X .
  (2) Choose the next centroid $c^{(i)}$, selecting $c^{(i)} = x' \in X$ with probability $\frac{D(x')^2}{\sum_{x \in X} D(x)^2}$
  (3) Repeat Step (2) until we have chosen a total of k centroids.

$$X = [x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)}]$$



$c^{(1)}$ →

i	x_1	x_2	
1	0.1	0.4	
2	0.4	0.6	$D(x)^2$
3	0.8	0.5	
4	0.7	0.2	

Step-2:

$$x' = [x^{(2)}, x^{(3)}, x^{(4)}]$$

$$D(x^{(2)})^2 = d(c^{(1)}, x^{(2)})^2 = (0.1-0.4)^2+(0.4-0.6)^2 = 0.13$$

$$D(x^{(3)})^2 = d(c^{(1)}, x^{(3)})^2 = (0.1-0.8)^2+(0.4-0.5)^2 = 0.5$$

$$D(x^{(4)})^2 = d(c^{(1)}, x^{(4)})^2 = (0.1-0.7)^2+(0.4-0.2)^2 = 0.4$$

probability that $x^{(3)}$ is chosen as $c^{(2)}$.

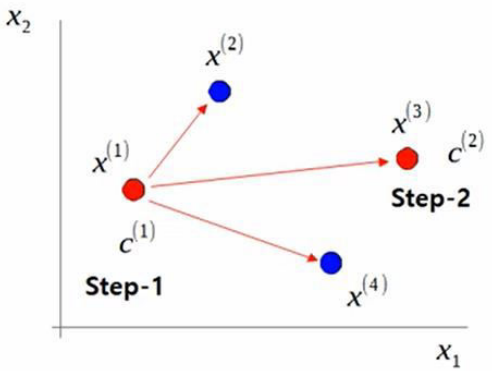
Probability density function: $\frac{D(x')^2}{\sum_{j=[2,3,4]} D(x^{(j)})^2} = \frac{1}{0.13+0.5+0.4} \times [0.13, 0.5, 0.4] = [0.126,\ 0.485,\ 0.388]$

probability that $x^{(4)}$ is chosen as $c^{(2)}$

K-means++ Algorithm

- At any given time, let D(x) denote the shortest distance from a data point x to the closest centroid we have already chosen.
  (1) Choose an initial centroid $c^{(1)}$ uniformly at random from X .
  (2) Choose the next centroid $c^{(i)}$, selecting $c^{(i)} = x' \in X$ with probability $\frac{D(x')^2}{\sum_{x \in X} D(x)^2}$
  (3) Repeat Step (2) until we have chosen a total of k centroids.

$$X = [x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)}]$$



	i	x_1	x_2
$c^{(1)}$ →	1	0.1	0.4
	2	0.4	0.6
$c^{(2)}$ →	3	0.8	0.5
	4	0.7	0.2

d^2

Step-3:

$x' = [x^{(2)}, x^{(4)}]$

closest

$D(x^{(2)})^2 = d(c^{(1)}, x^{(2)})^2 = (0.1-0.4)^2+(0.4-0.6)^2 = 0.13$

$D(x^{(2)})^2 = d(c^{(2)}, x^{(2)})^2 = (0.8-0.4)^2+(0.5-0.6)^2 = 0.17$

$D(x^{(4)})^2 = d(c^{(1)}, x^{(4)})^2 = (0.1-0.7)^2+(0.4-0.2)^2 = 0.4$

$D(x^{(4)})^2 = d(c^{(2)}, x^{(4)})^2 = (0.8-0.7)^2+(0.5-0.2)^2 = 0.1$

probability that $x^{(2)}$ is chosen as $c^{(3)}$.

Probability density function: $\frac{D(x')^2}{\sum_{j=[2,4]} D(x^{(j)})^2} = \frac{1}{0.13+0.1} \times [0.13, 0.1] = [0.565,\ 0.435]$

probability that $x^{(4)}$ is chosen as $c^{(3)}$.

* $x^{(1)}$, $x^{(2)}$, and $x^{(3)}$ are likely to be chosen as initial focuses.