k-Means Clustering

부산대학교 정보·의생명공학대학
정보컴퓨터공학부

개요

- Clustering
- k-means clustering
- Example code

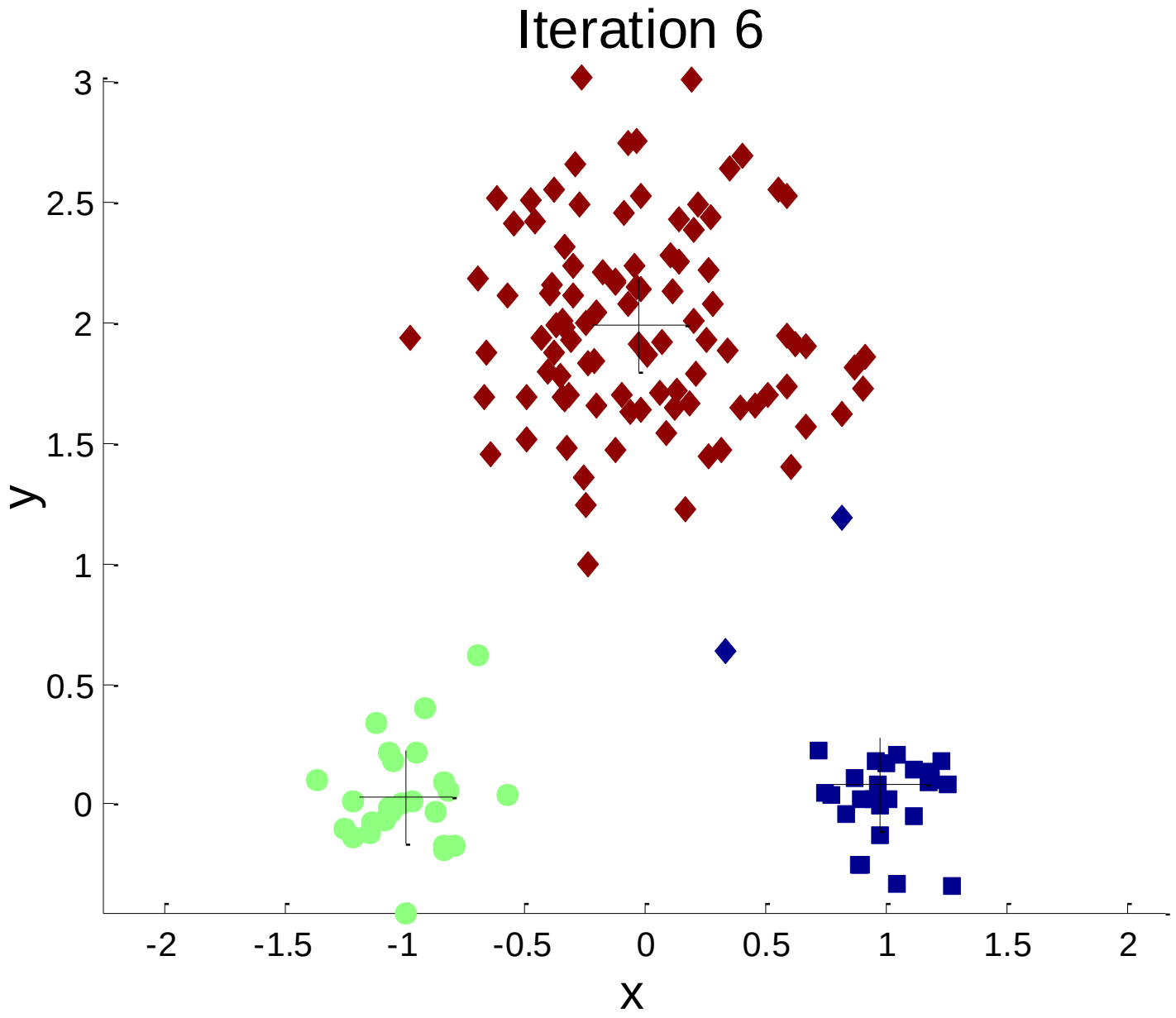
K-MENAS CLUSTERING

참조:

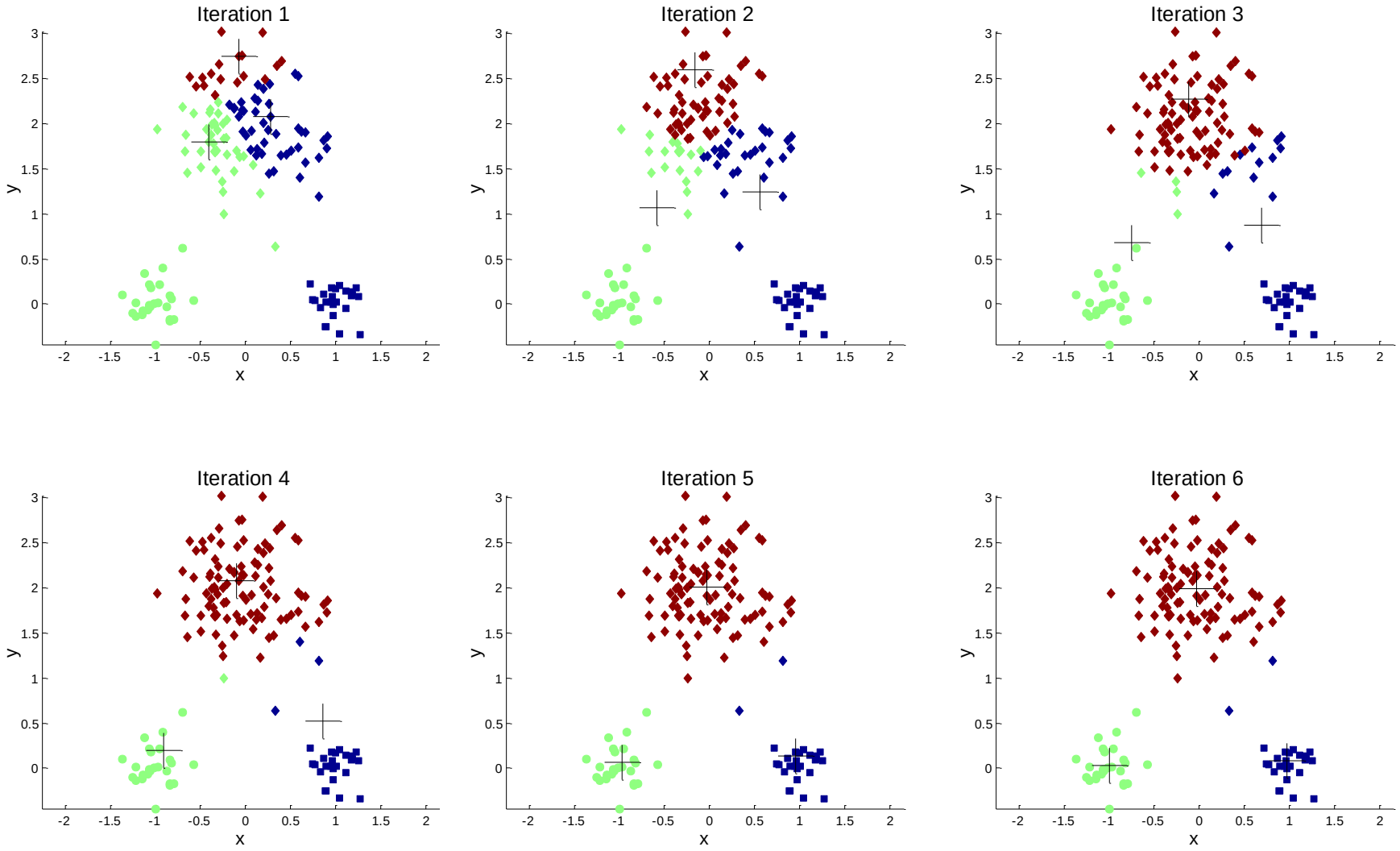
Data Mining: Concepts and Techniques, 3rd Edition, Han et al.

Introduction to Data Mining, 2nd Edition, Tan et al.

Example of K-means Clustering



Example of K-means Clustering



K-Means Clustering

- Partitioning method: Discovering the groupings in the data by optimizing a specific objective function and iteratively improving the quality of partitions
- Number of clusters, k, must be specified
- Each cluster is associated with a centroid (center point)
- Each point is assigned to the cluster with the closest centroid
- The basic algorithm is very simple

1: Select K points as the initial centroids.
2: **repeat**
3: Form K clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: **until** The centroids don't change

K-Means Clustering

- Simple iterative algorithm.
  - Choose initial centroids;
  - repeat {assign each point to a nearest centroid; re-compute cluster centroids}
  - until centroids stop changing.
- Initial centroids are often chosen randomly.
  - Clusters produced can vary from one run to another
- The centroid is (typically) the mean of the points in the cluster, but other definitions are possible
- K-means will converge for common proximity measures with appropriately defined centroid
- Most of the convergence happens in the first few iterations.
  - Often the stopping condition is changed to ‘Until relatively few points change clusters’
- Complexity is O(n * K * I * d)
  - n = number of points, K = number of clusters,
    I = number of iterations, d = number of attributes

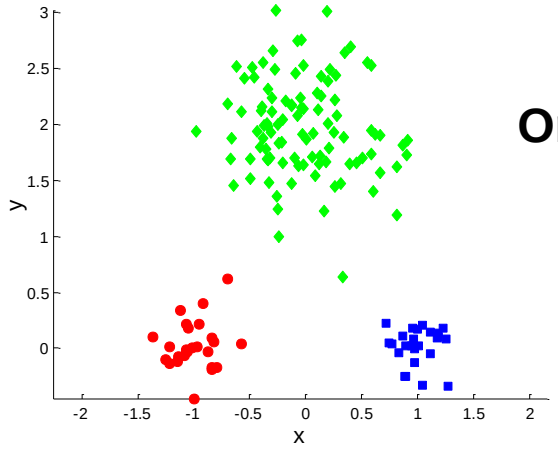
K-means Objective Function

- A common objective function (used with Euclidean distance measure) is Sum of Squared Error (SSE)
  - For each point, the error is the distance to the nearest cluster center
  - To get SSE, we square these errors and sum them.

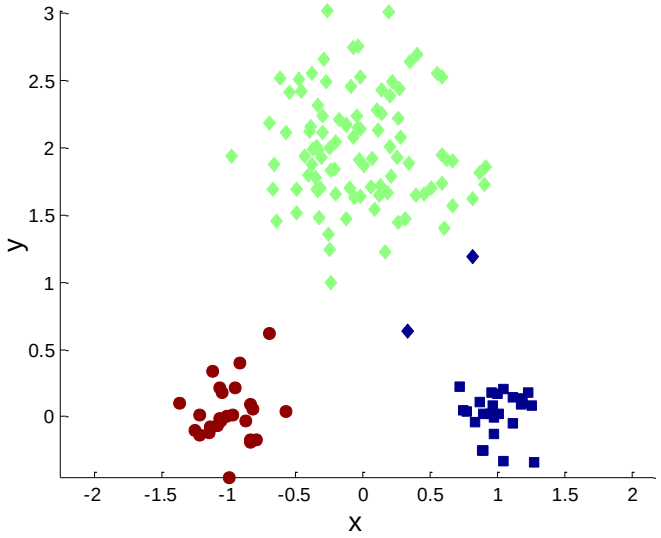
$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

  - x is a data point in cluster C_i and m_i is the centroid (mean) for cluster C_i
  - SSE improves in each iteration of K-means until it reaches a local or global minima.

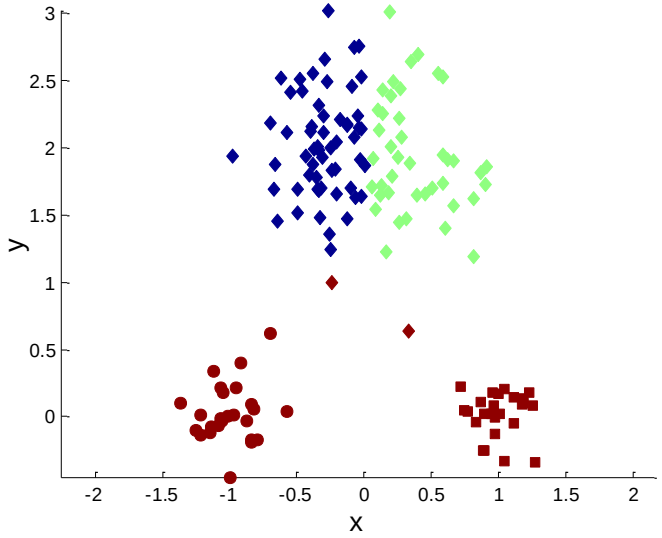
Two different K-means Clusterings



Original Points

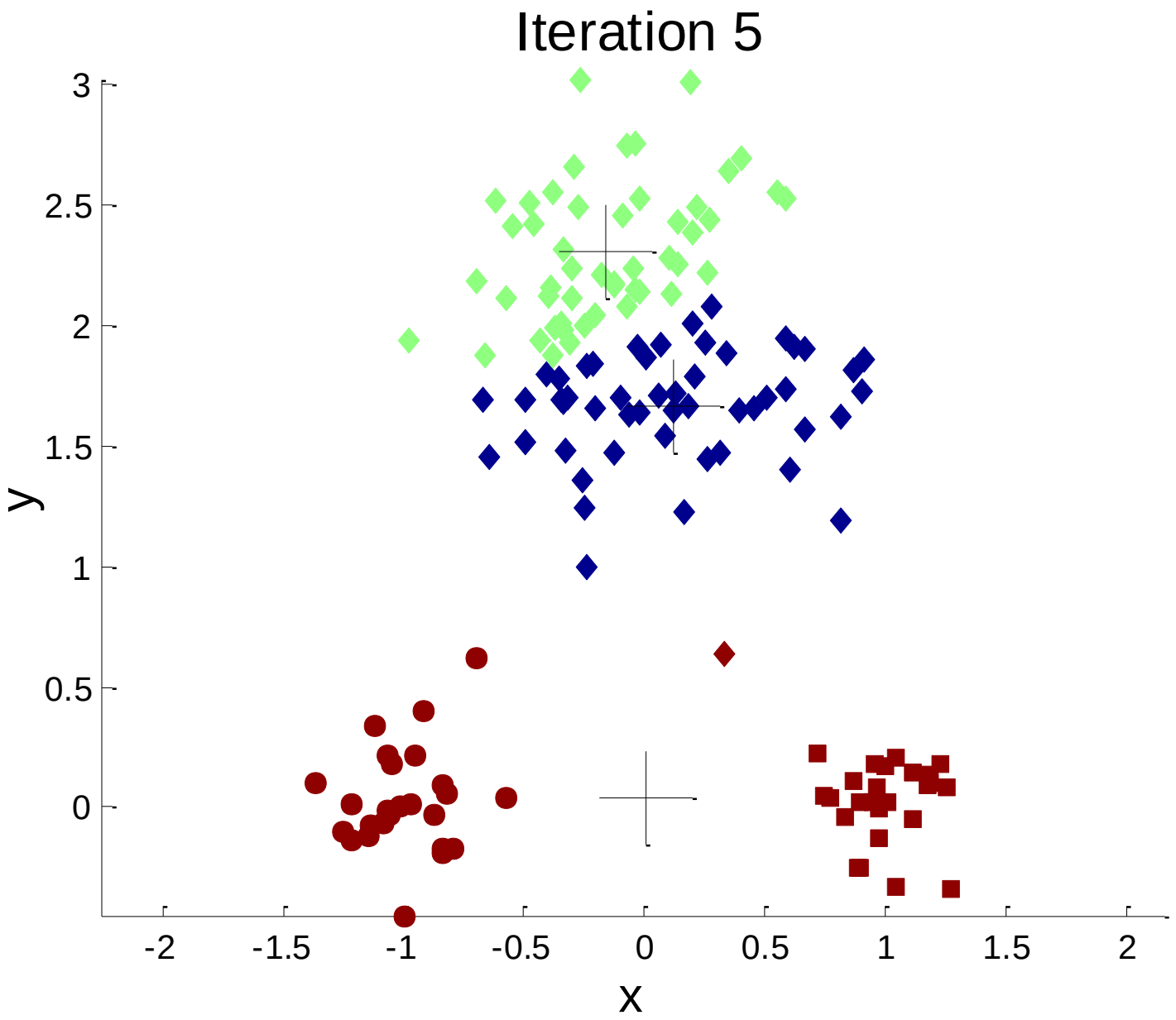


Optimal Clustering

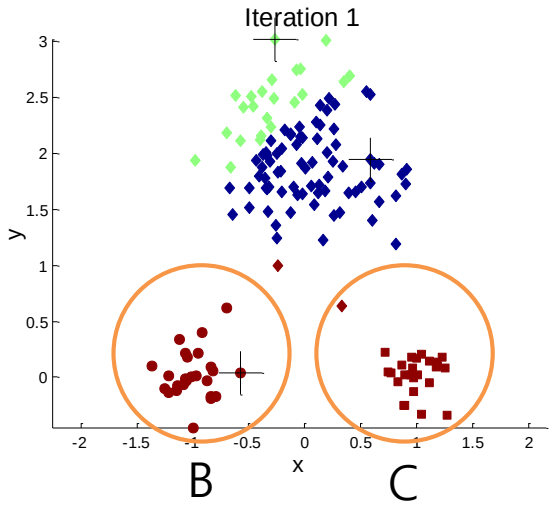


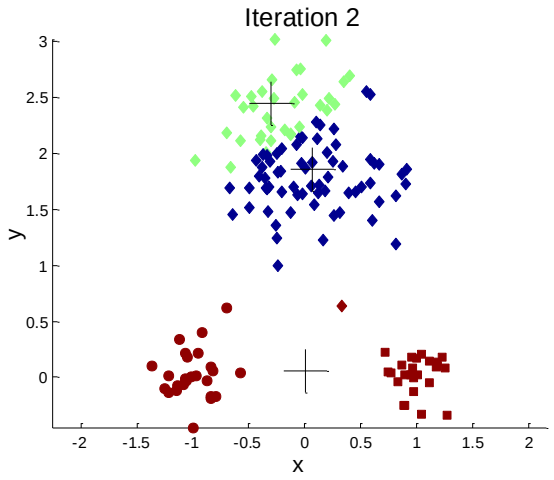
Sub-optimal Clustering

Importance of Choosing Initial Centroids …

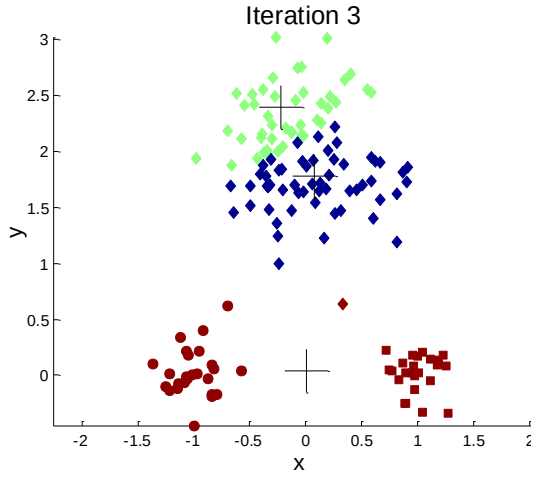


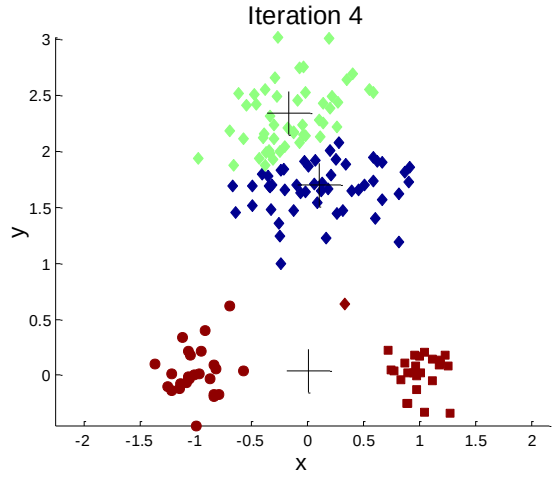
Importance of Choosing Initial Centroids …

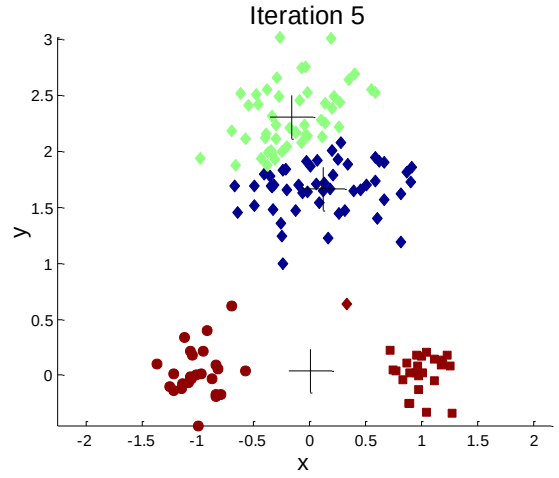




Depending on the choice of initial centroids, B and C may get merged or remain separate

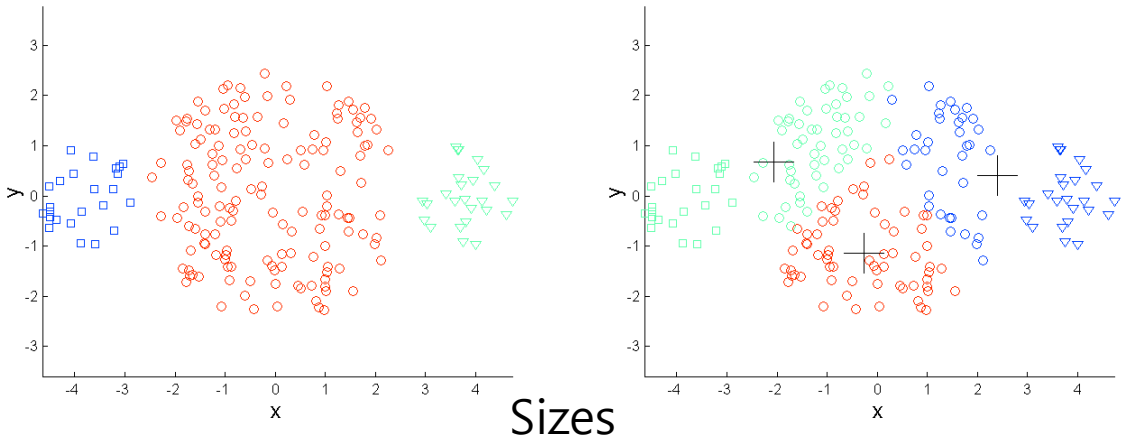




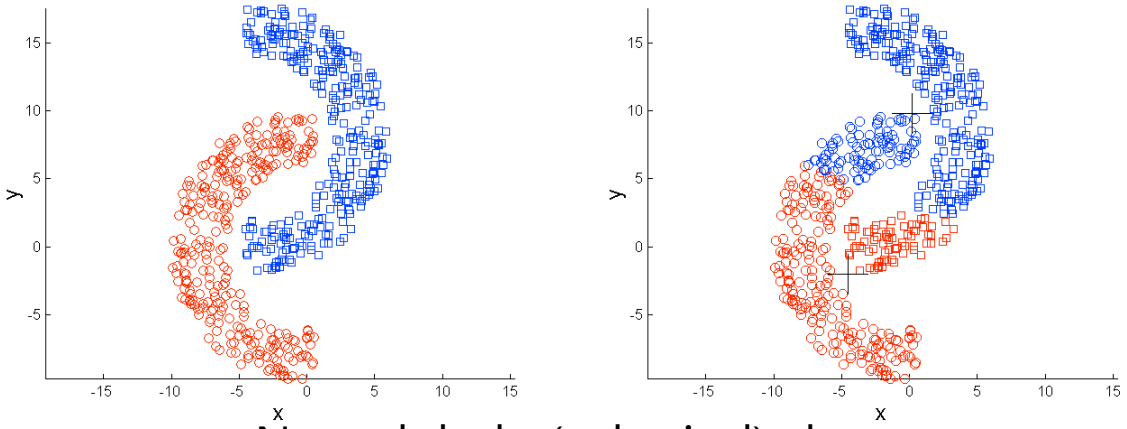


Limitation of k-means

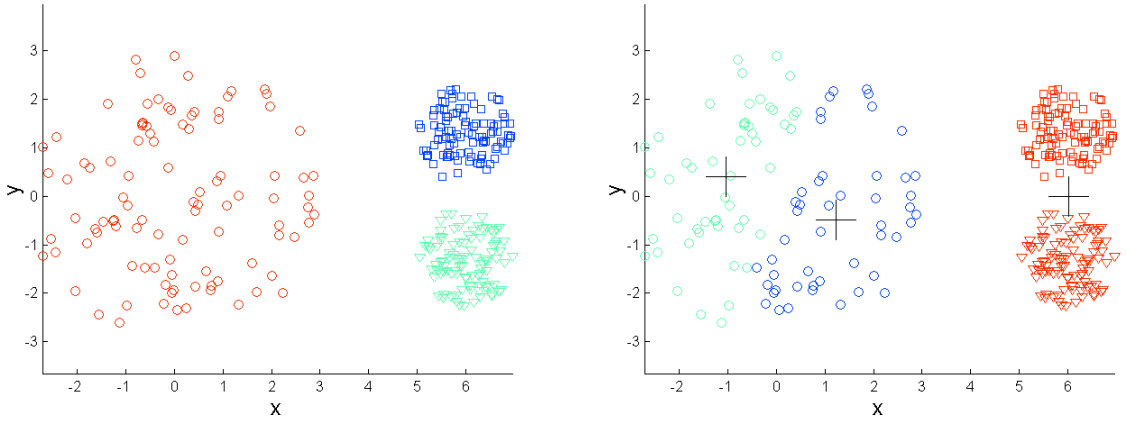
- K-means has problems when clusters are of differing size, densities, non-globular shapes



Sizes



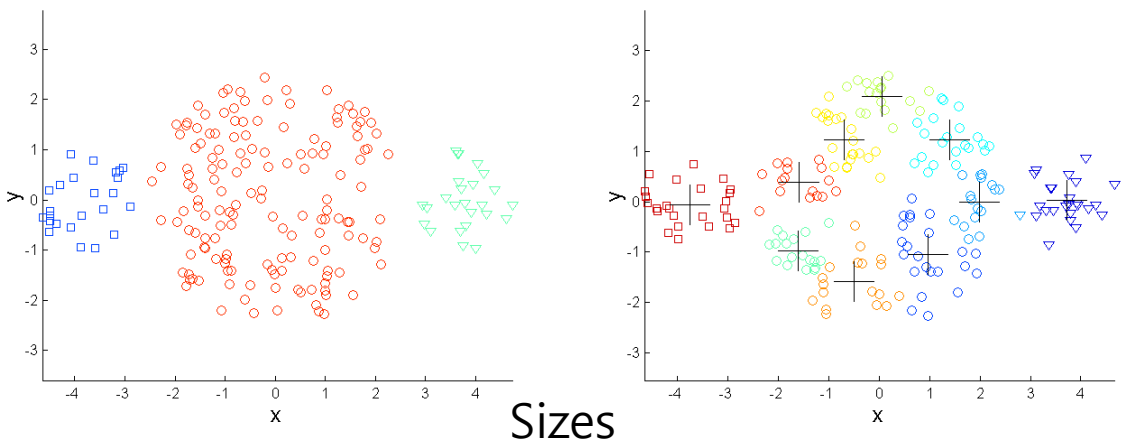
Non-globular(spherical) shapes



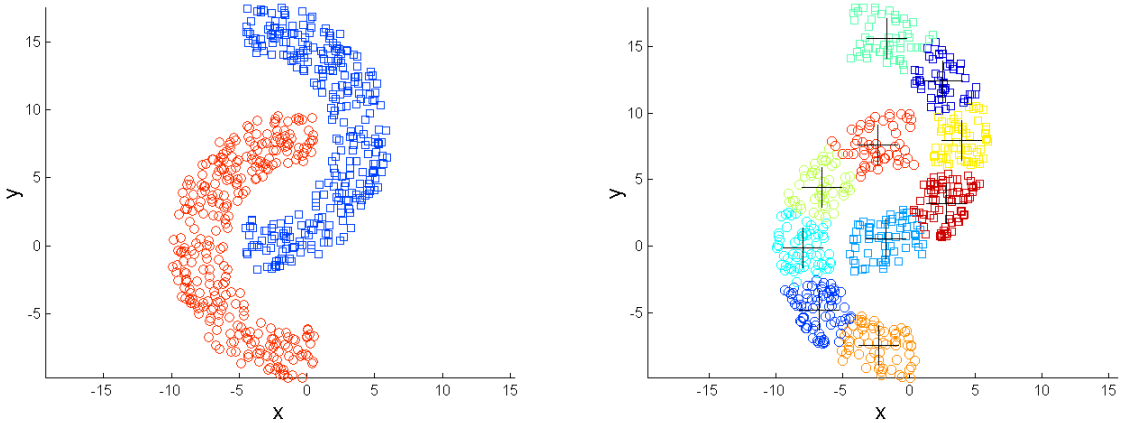
Densities

Limitation of k-means

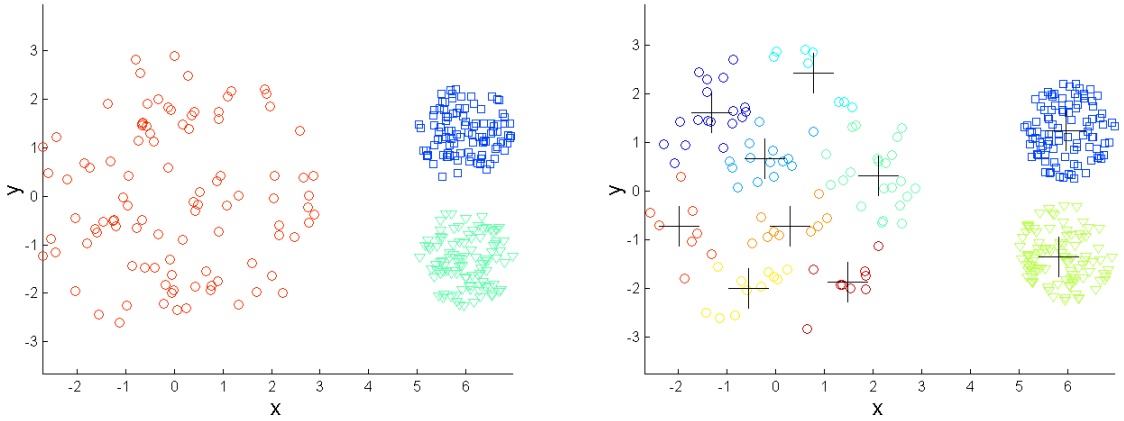
- One solution is to find a large number of clusters such that each of them represents a part of a natural clu ster. But these small clusters need to be put together in a post-processing step.



Sizes



Non-globular(spherical) shapes



Densities

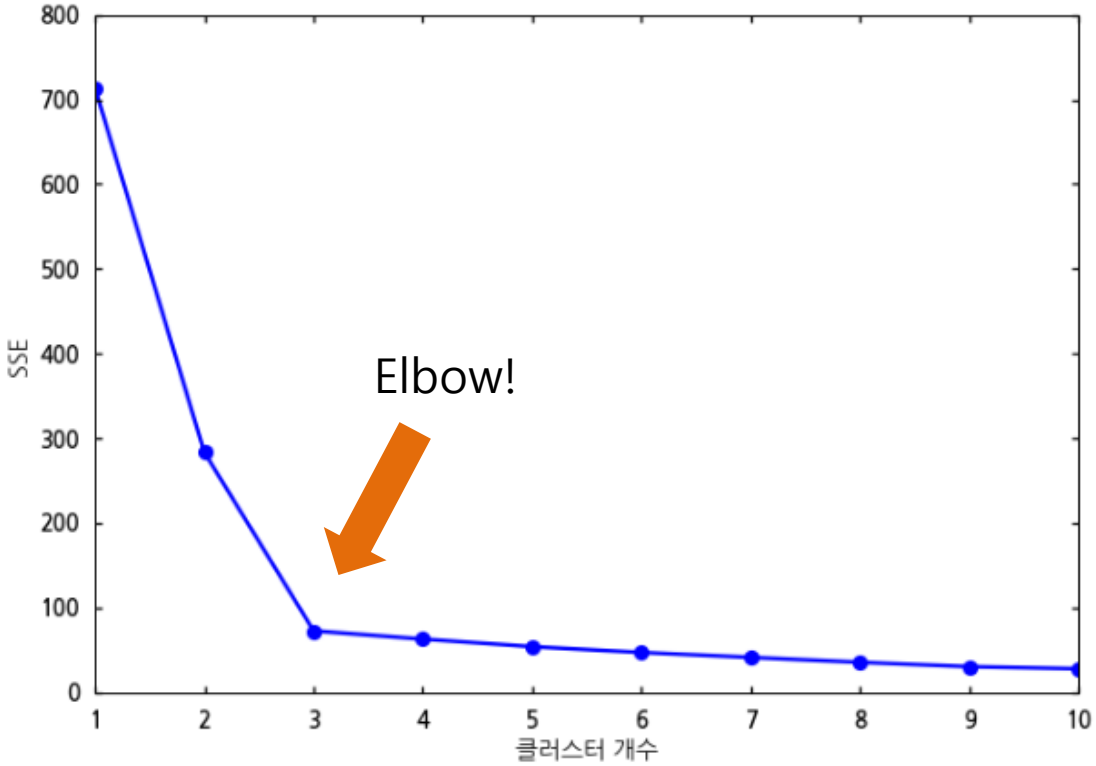
Issues with k-means method

- K-means clustering often terminates at a local optimal
  - Initialization can be important to find high-quality clusters
- Need to specify K, the number of clusters, in advance
  - There are ways to automatically determine the “best” K
  - In practice, one often runs a range of values and selected the “best” K value
- Sensitive to noisy data and outliers
  - Variations: Using K-medians, K-medoids, etc.
- K-means is applicable only to objects in a continuous n-dimensional space
  - Using the K-modes for categorical data
- Not suitable to discover clusters with non-convex shapes
  - Using density-based clustering, kernel K-means, etc.
- Variations of k-means
  - Choosing better initial centroid estimates: k-means++, intelligent k-means, generic k-means
  - Choosing different representative prototypes for the clusters: k-medoids, k-medians, k-modes
  - Applying feature transformation techniques: weighted k-means, kernel k-means

Finding best k

- Elbow method: using SSE to determine k

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$



Evaluation: Classifier

❖ **Confusion Matrix:**

Actual class\Predicted class	C_1	¬ C_1
C_1	**True Positives (TP)**	**False Negatives (FN)**
¬ C_1	**False Positives (FP)**	**True Negatives (TN)**

❖ **Example of Confusion Matrix:**

Actual class\Predicted class	buy_computer = yes	buy_computer = no	Total
buy_computer = yes	**6954**	**46**	7000
buy_computer = no	**412**	**2588**	3000
Total	7366	2634	10000

Evaluation: Classifier

A\P	C	¬C	
C	**TP**	**FN**	**P**
¬C	**FP**	**TN**	**N**
	P'	**N'**	**All**

❖ **Classifier accuracy,** or recognition rate

- Percentage of test set tuples that a re correctly classified

Accuracy = (TP + TN)/All

❖ **Error rate:** *1 – accuracy*, or

Error rate = (FP + FN)/All

❖ **Class imbalance problem**

- One class may be *rare*
  - E.g., fraud, or HIV-positive
- Significant *majority of the negative class* and minority of the positive class
- Measures handle the class imbalance problem
  - **Sensitivity** (recall): True positive recognition rate
    - **Sensitivity = TP/P**
  - **Specificity**: True negative recognition rate
    - **Specificity = TN/N**

Evaluation: Classifier

A\P	C	¬C	
C	**TP**	**FN**	**P**
¬C	**FP**	**TN**	**N**
	P'	**N'**	**All**

- **Precision**: Exactness: what % of tuples that the classifier labeled as positive are actually positive?

$$\mathrm{P} = \mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

- **Recall:** Completeness: what % of positive tuples did the classifier label as positive?

$$\mathrm{R} = \mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

  - Range: [0, 1]

- ***F* measure (**or ***F*-score):** harmonic mean of precision and recall
  - In general, it is the weighted measure of precision & recall

$$\mathrm{F}_\beta = \frac{1}{\alpha \cdot \frac{1}{\mathrm{P}} + (1 - \alpha) \cdot \frac{1}{\mathrm{R}}} = \frac{(\beta^2 + 1)\mathrm{PR}}{\beta^2 \mathrm{P} + \mathrm{R}}$$

Assigning β times as much weig ht to recall as to precision)

  - ***F1-measure (balanced F-measure)***
    - » That is, when β = 1, $\mathrm{F}_1 = \frac{2\mathrm{PR}}{\mathrm{P} + \mathrm{R}}$

Evaluation: Classifier

❖ Use the same confusion matrix, calculate the measure just introduced

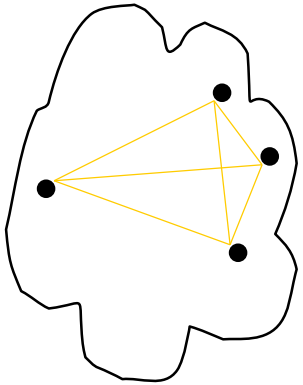
Actual Class\Predicted class	cancer = yes	cancer = no	Total	Recognition(%)
cancer = yes	**90**	**210**	300	30.00 (*sensitivity*)
cancer = no	**140**	**9560**	9700	98.56 (*specificity*)
Total	230	9770	10000	96.50 (*accuracy*)

- Sensitivity = TP/P = 90/300 = 30%
- Specificity = TN/N = 9560/9700 = 98.56%
- Accuracy = (TP + TN)/All = (90+9560)/10000 = 96.50%
- Error rate = (FP + FN)/All = (140 + 210)/10000 = 3.50%
- Precision = TP/(TP + FP) = 90/(90 + 140) = 90/230 = 39.13%
- Recall = TP/ (TP + FN) = 90/(90 + 210) = 90/300 = 30.00%
- F1 = 2 P × R /(P + R) = 2 × 39.13% × 30.00%/(39.13% + 30%) = 33.96%

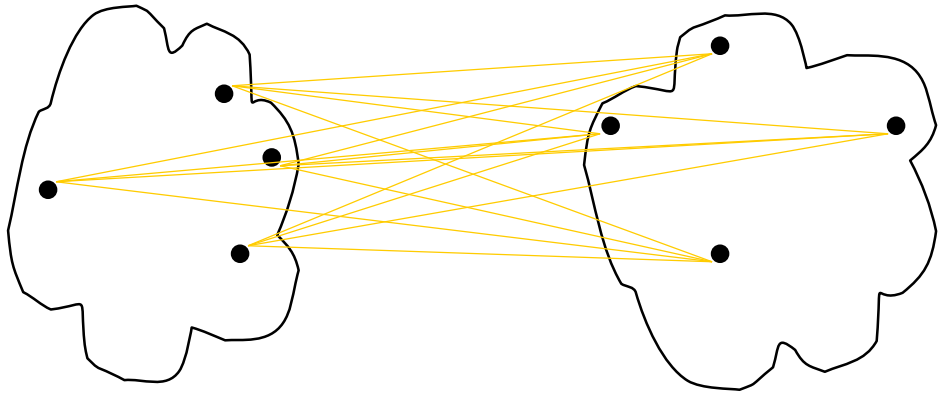
Evaluation: Clustering

- Supervised (external evaluation) : used to measure the extent to which cluster labels match externally supplied class labels.
  - Jaccard index = TP/(TP+FN+FP)
  - Rand index = (TP + TN)/N

- Unsupervised (internal evaluation) : used to measure the goodness of a clustering structure without respect to external information.
  - Sum of Squared Error (SSE)
  - Sum of squares between (SSB)
  - Silhouette coefficient



cohesion



separation

Unsupervised Measures: Cohesion and Separation

- Cluster Cohesion: Measures how closely related are objects in a cluster
  - Example: SSE
- Cluster Separation: Measure how distinct or well-separated a cluster is from other clusters

Example: Squared Error

- Cohesion is measured by the within cluster sum of squares (SSE)

$$SSE = \sum_i \sum_{x \in C_i} (x - m_i)^2$$

- Separation is measured by the between cluster sum of squares

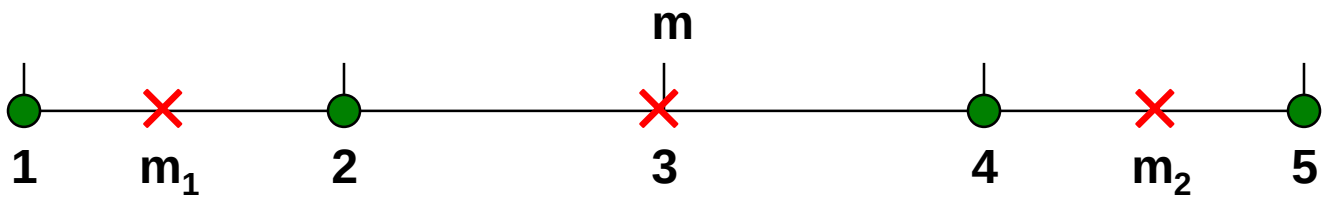
$$SSB = \sum_i |C_i|(m - m_i)^2$$

Where $|C_i|$is the size of cluster *i*

Unsupervised Measures: Cohesion and Separation

❖ Example: SSE

- SSB + SSE = constant



K=1 cluster:

$$SSE = (1-3)^2 + (2-3)^2 + (4-3)^2 + (5-3)^2 = 10$$

$$SSB = 4 \times (3-3)^2 = 0$$

$$Total = 10 + 0 = 10$$

K=2 clusters:

$$SSE = (1-1.5)^2 + (2-1.5)^2 + (4-4.5)^2 + (5-4.5)^2 = 1$$

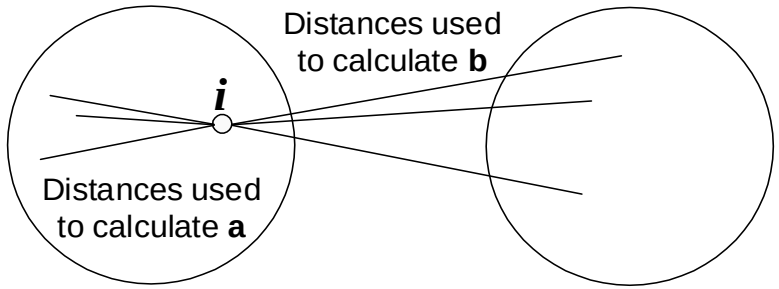
$$SSB = 2 \times (3-1.5)^2 + 2 \times (4.5-3)^2 = 9$$

$$Total = 1 + 9 = 10$$

Unsupervised Measures: Silhouette Coefficient

- Silhouette coefficient combines ideas of both cohesion and separation, but for individual points, as well a s clusters and clusterings
- For an individual point, ***i***
  - Calculate ***a*** = average distance of ***i*** to the points in its cluster
  - Calculate ***b*** = min (average distance of ***i*** to points in another cluster)
  - The silhouette coefficient for a point is then given by

    s = (b – a) / max(a,b)



  - Value can vary between -1 and 1
  - Typically ranges between 0 and 1.
  - The closer to 1 the better.

EXAMPLE CODES

참조:

Data Mining: Concepts and Techniques, 3rd Edition, Han et al.

Introduction to Data Mining, 2nd Edition, Tan et al.