DATA MINING 1
Centroid-based Clustering

Dino Pedreschi, Riccardo Guidotti

Revisited slides from Lecture Notes for Chapter 7 “Introduction to Data Mining”, 2nd Edition by Tan, Steinbach, Karpatne, Kumar

UNIVERSITÀ DI PISA

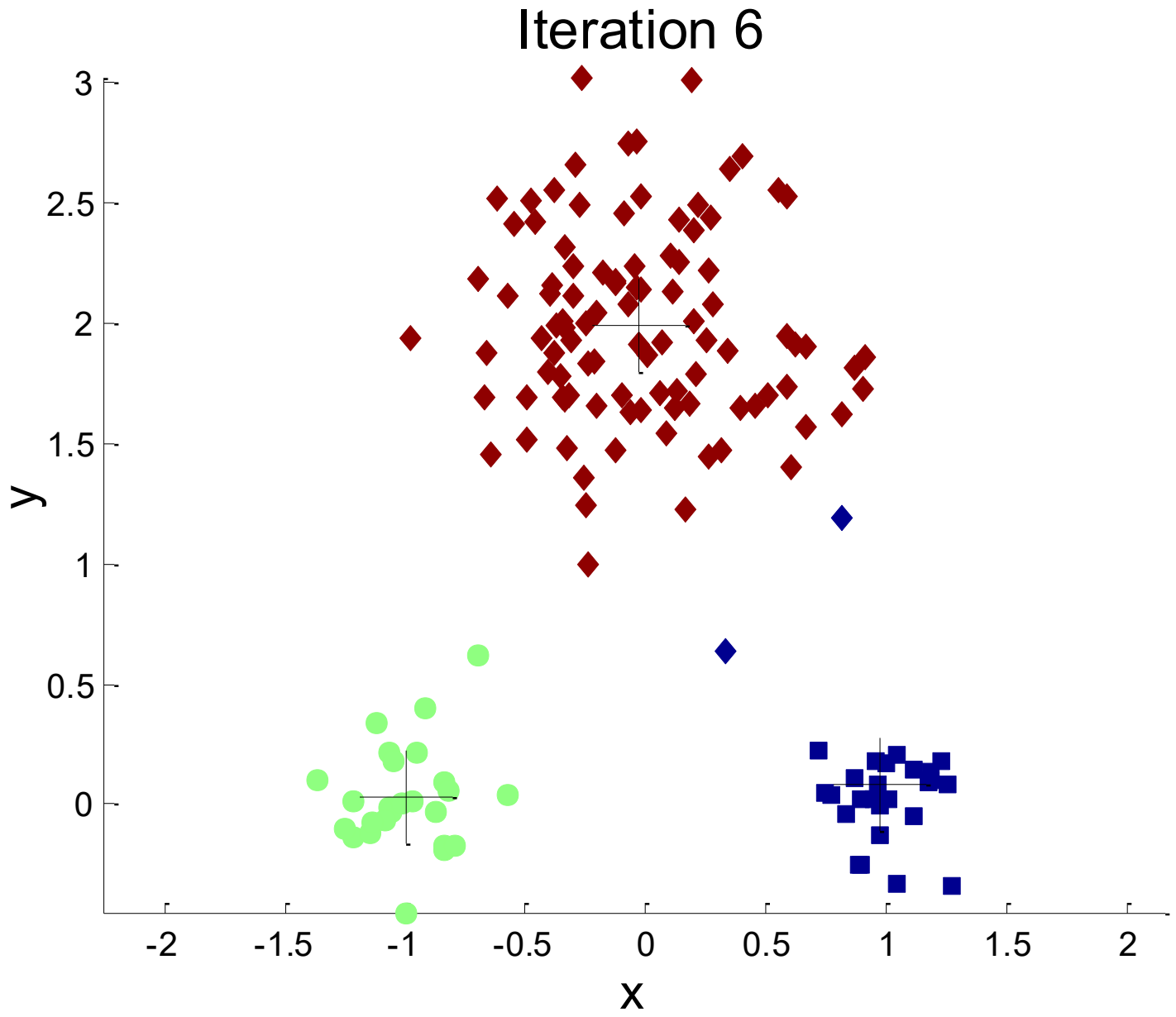
K-Means

K-Means Clustering

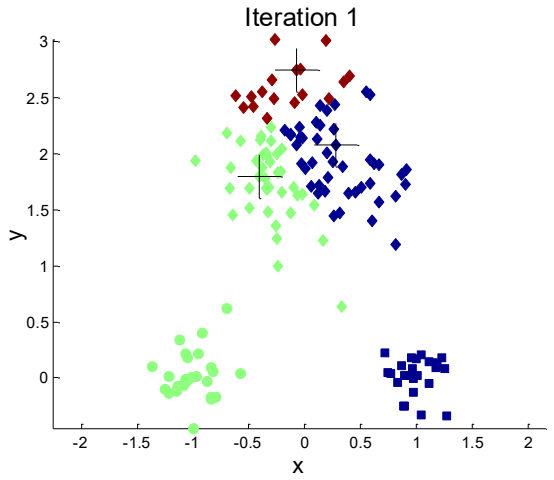
- Partitional clustering approach
- Number of clusters, K, must be specified
- Each cluster is associated with a **centroid** (center point)
- Each point is assigned to the cluster with the **closest centroid**
- The basic algorithm is very simple

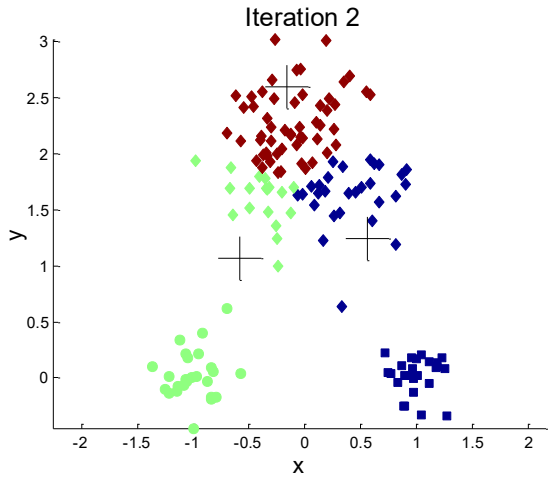
1: Select K points as the initial centroids.
2: **repeat**
3: Form K clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: **until** The centroids don't change

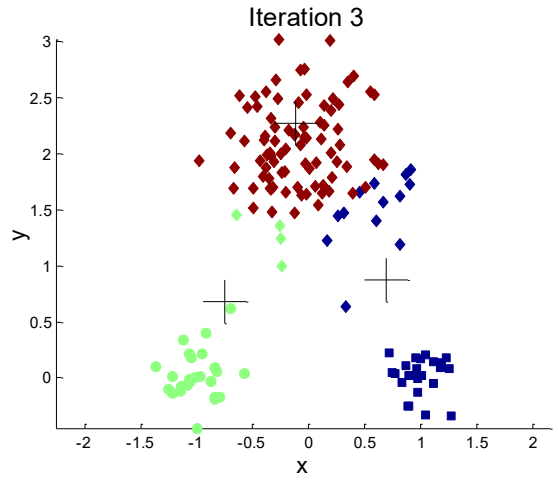
Example of K-Means Clustering

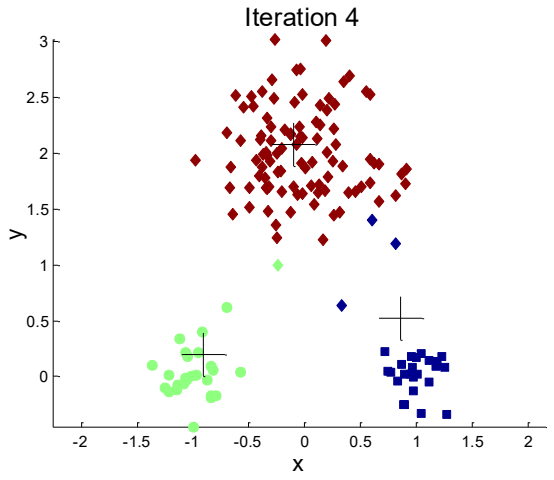


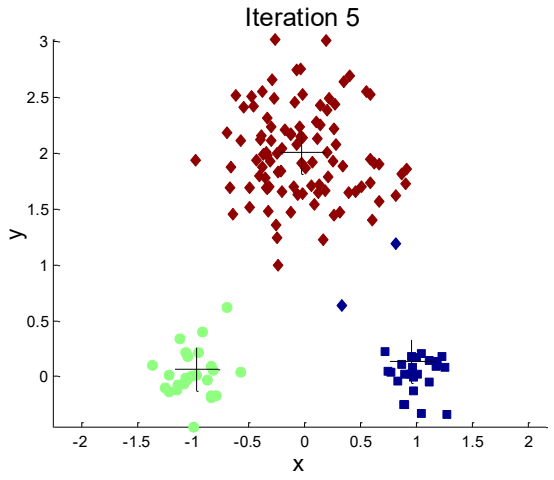
Example of K-Means Clustering

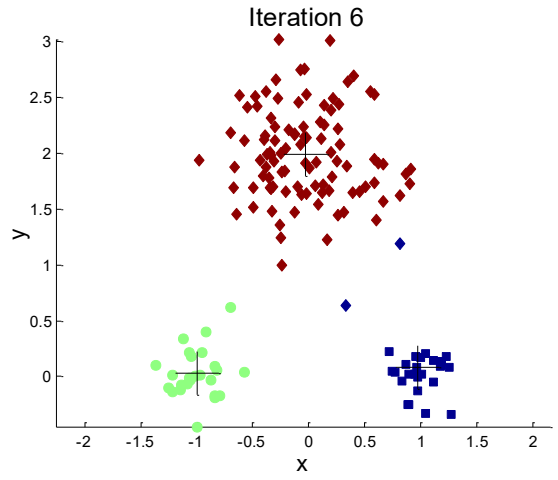








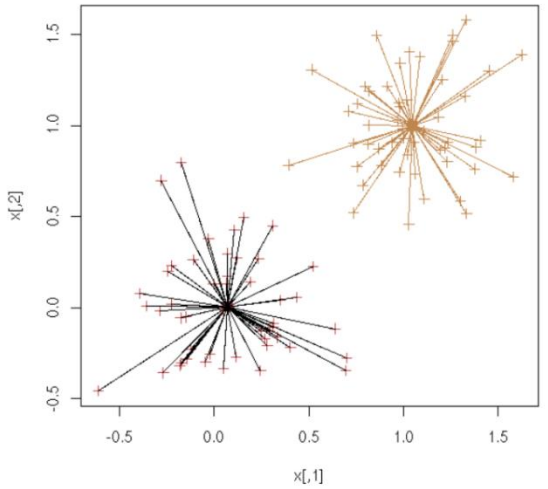




K-Means Clustering – Details

- Initial centroids are often chosen randomly.
  - Clusters produced vary from one run to another.
- The centroid is (typically) the mean of the points in the cluster.
- ‘Closeness’ is measured by Euclidean distance, cosine similarity, correlation, etc.
- K-means will converge for common similarity measures mentioned above.
- 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

Evaluating K-Means Clusters



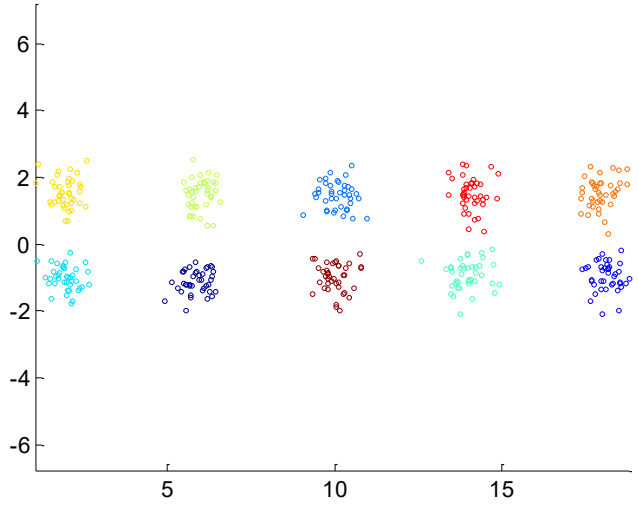
- Most common measure is Sum of Squared Error (SSE)
  - For each point, the error is the distance to the nearest cluster
  - To get SSE, we square these errors and sum them.

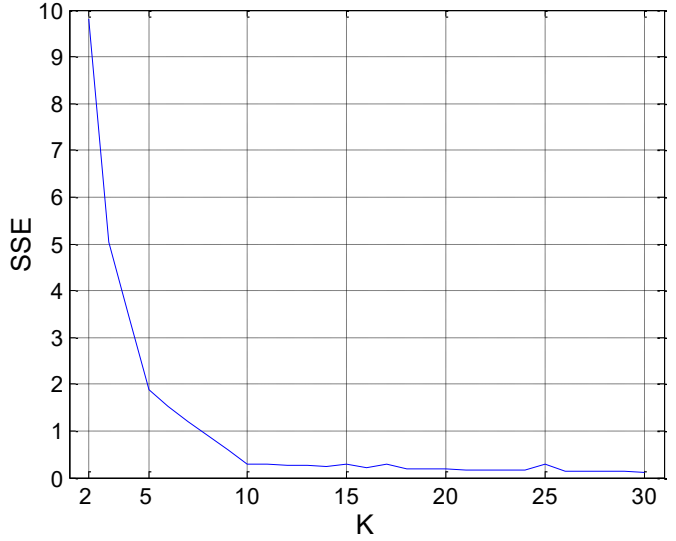
$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

- x is a data point in cluster C_i and m_i is the representative point for cluster Ci
  - can show that mi corresponds to the center (mean) of the cluster
- Given two sets of clusters, we prefer the one with the smallest error
- One easy way to reduce SSE is to increase K, the number of clusters
- A good clustering with smaller K can have a lower SSE than a poor clustering with higher K

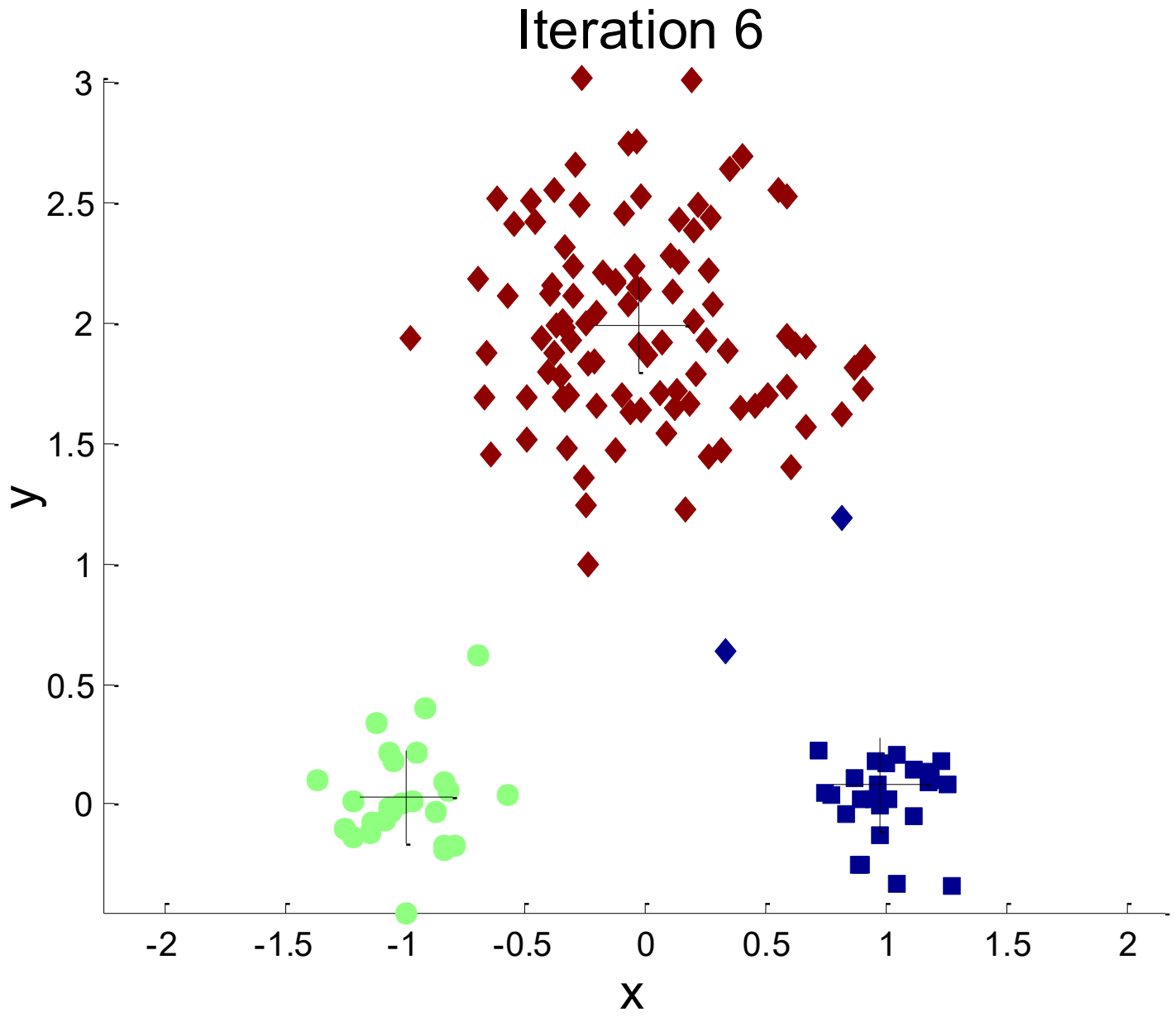
Selecting The Number of Clusters K

- SSE Used to measure the goodness of a clustering structure without respect to external information
- SSE is good for comparing two clusterings or two clusters (average SSE).
- Can also be used to estimate the number of clusters (elbow method).

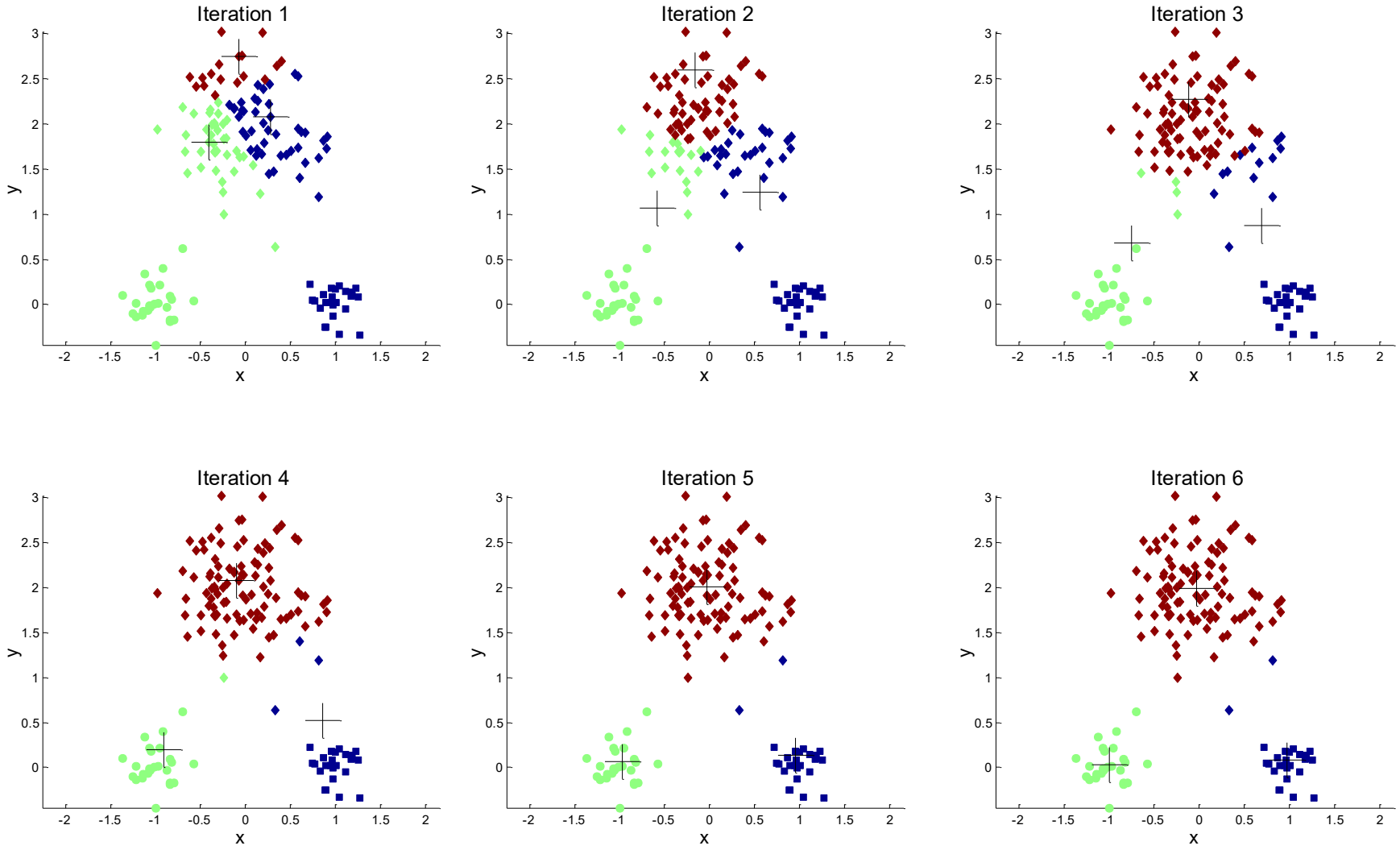




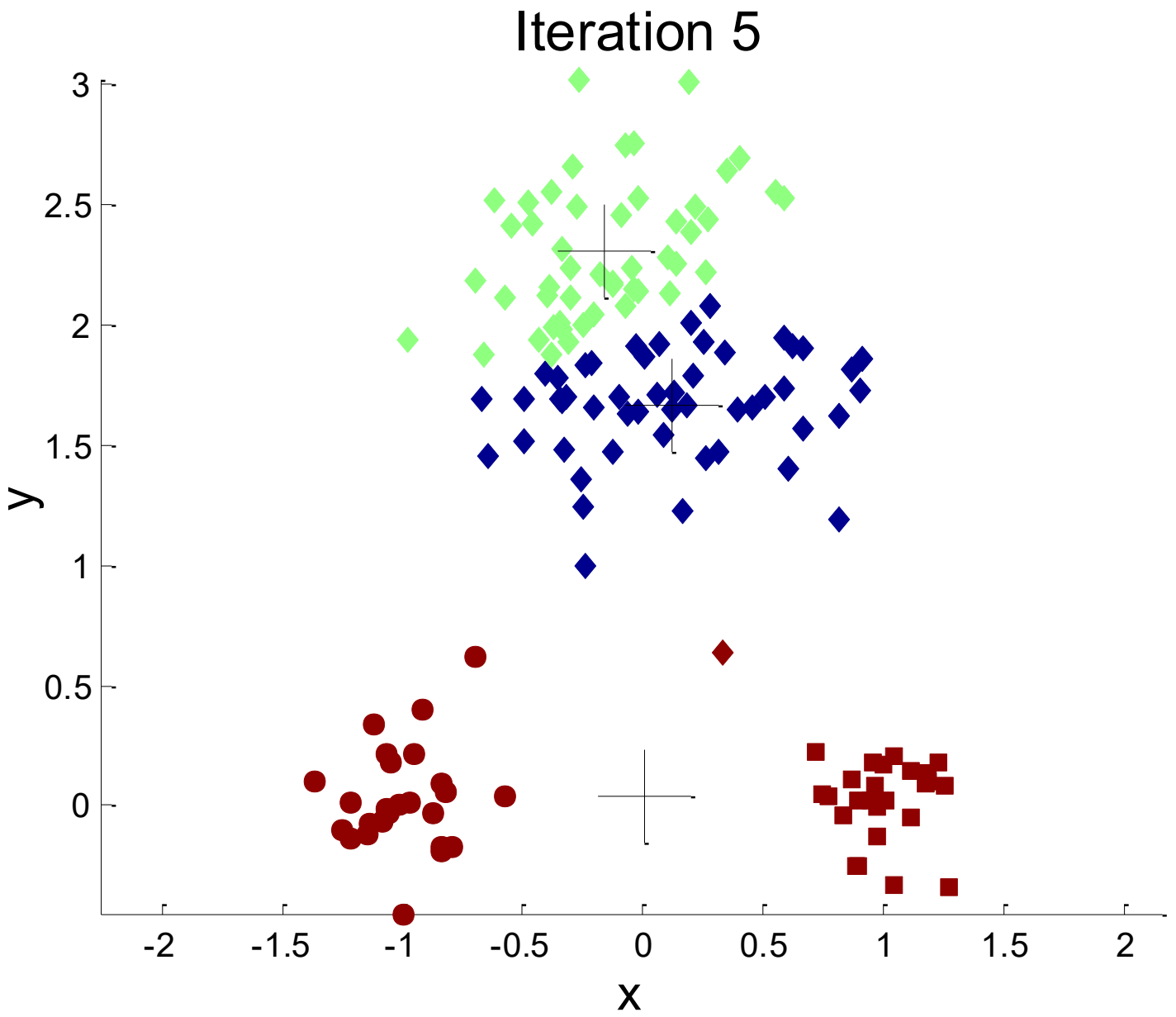
Importance of Choosing Initial Centroids



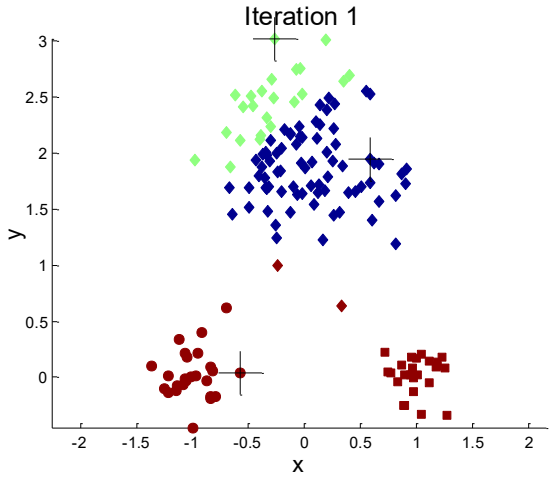
Importance of Choosing Initial Centroids

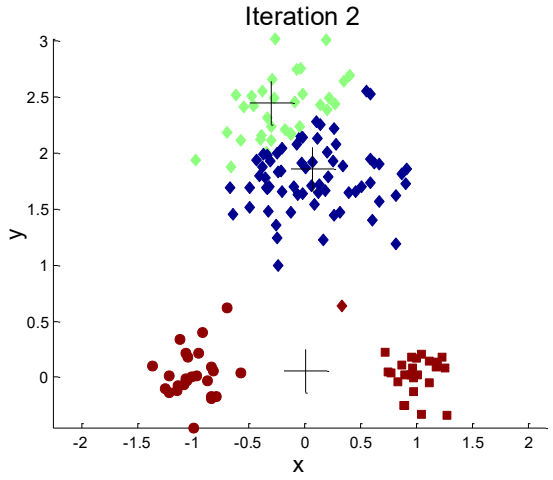


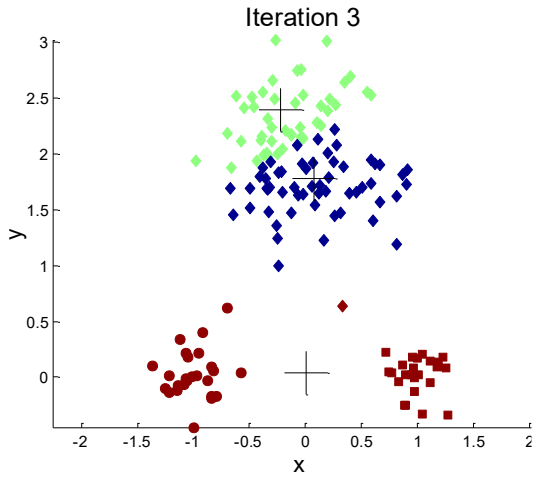
Importance of Choosing Initial Centroids …

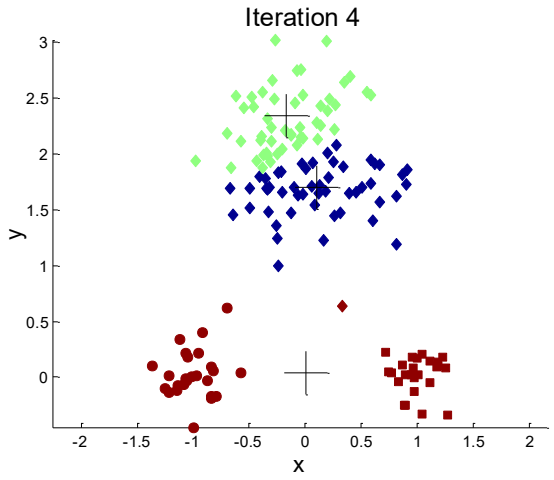


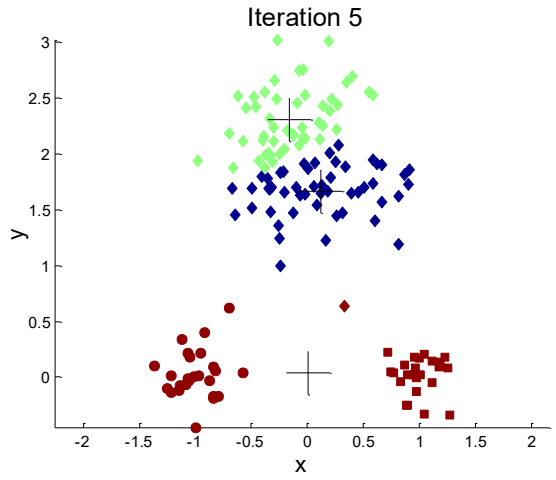
Importance of Choosing Initial Centroids …



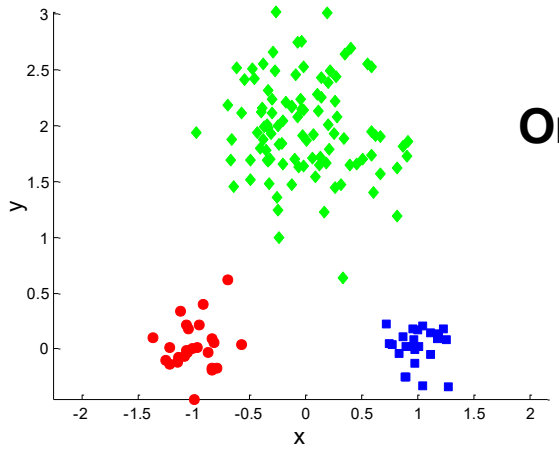




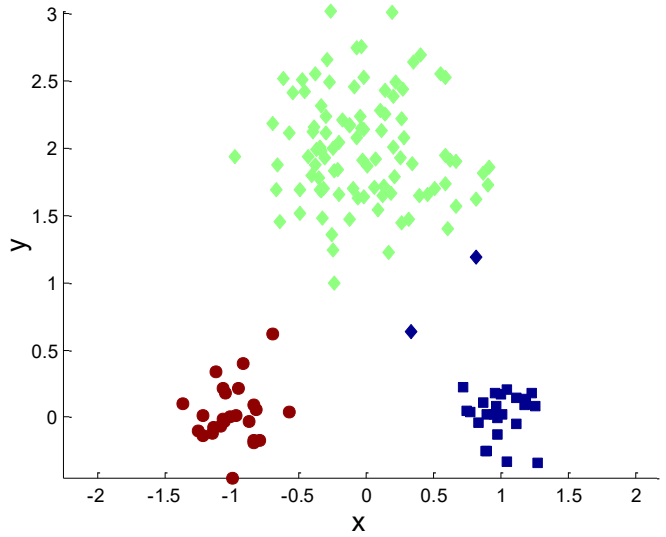




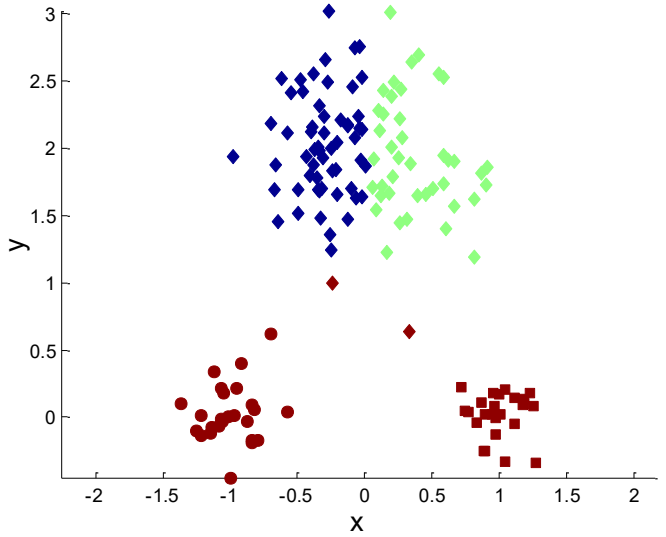
Two different K-Means Clusterings



Original Points

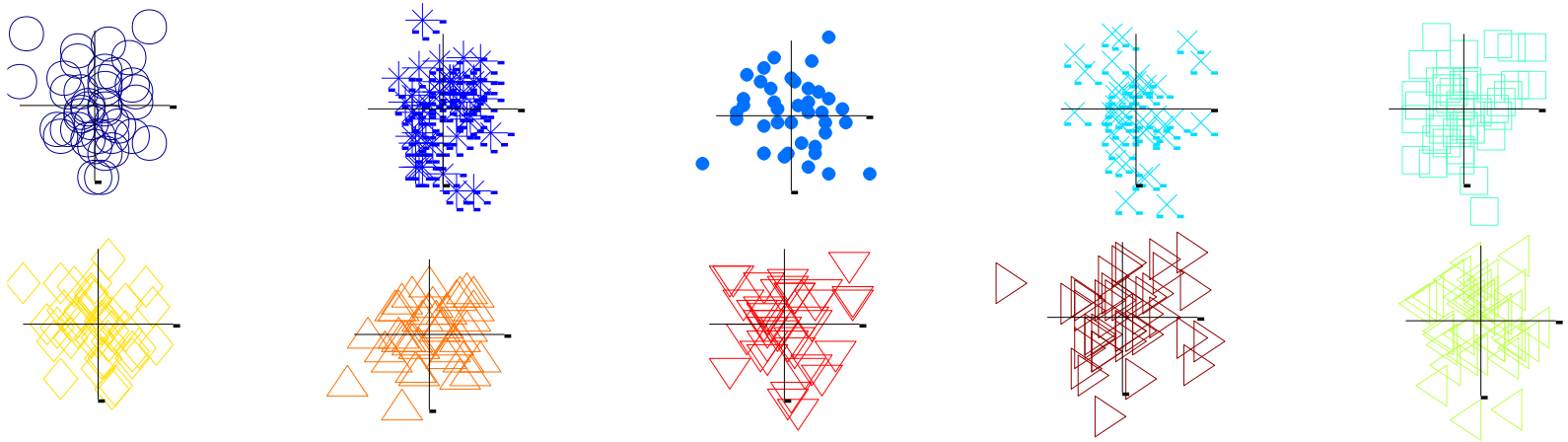


Optimal Clustering



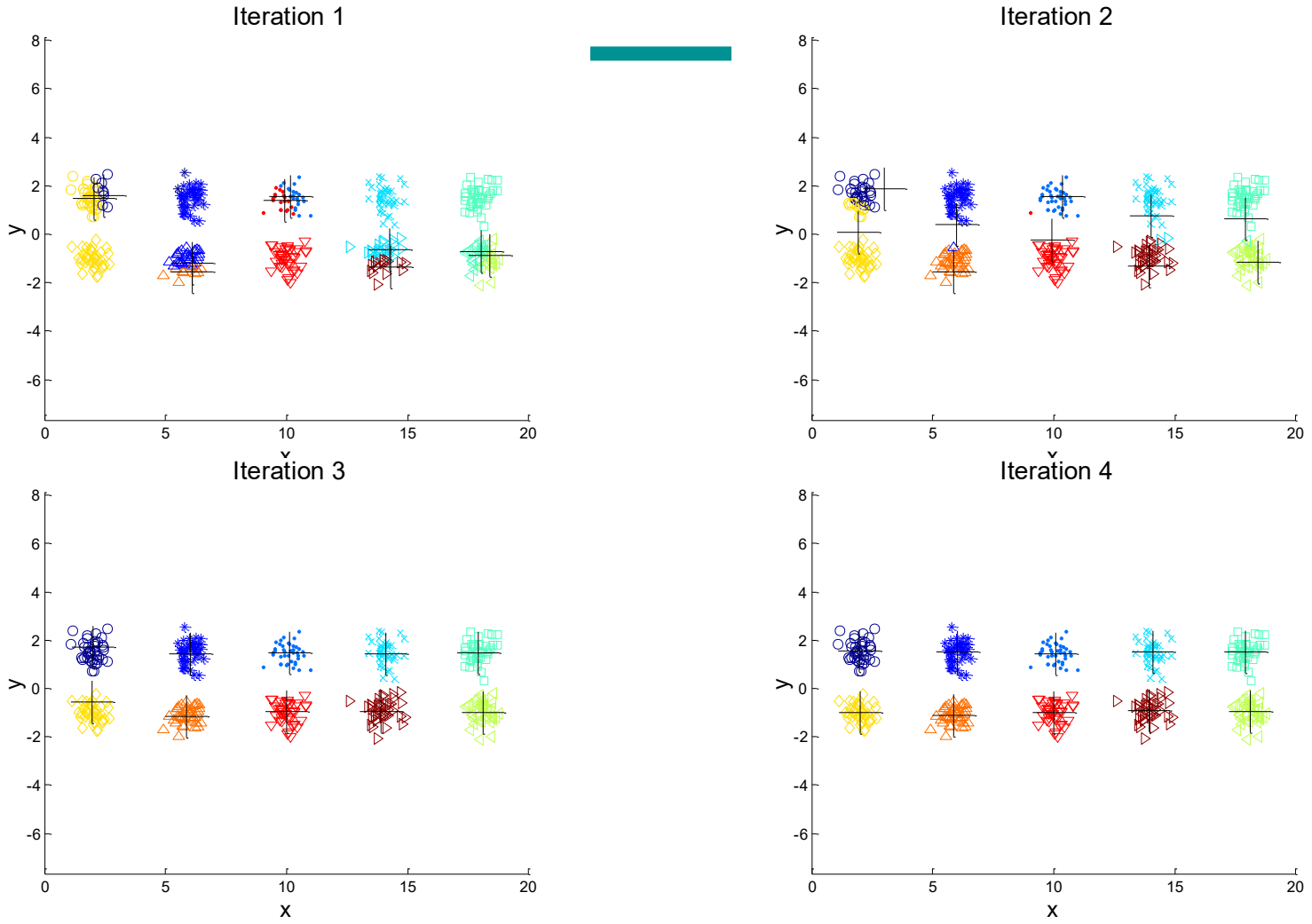
Sub-optimal Clustering

10 Clusters Example



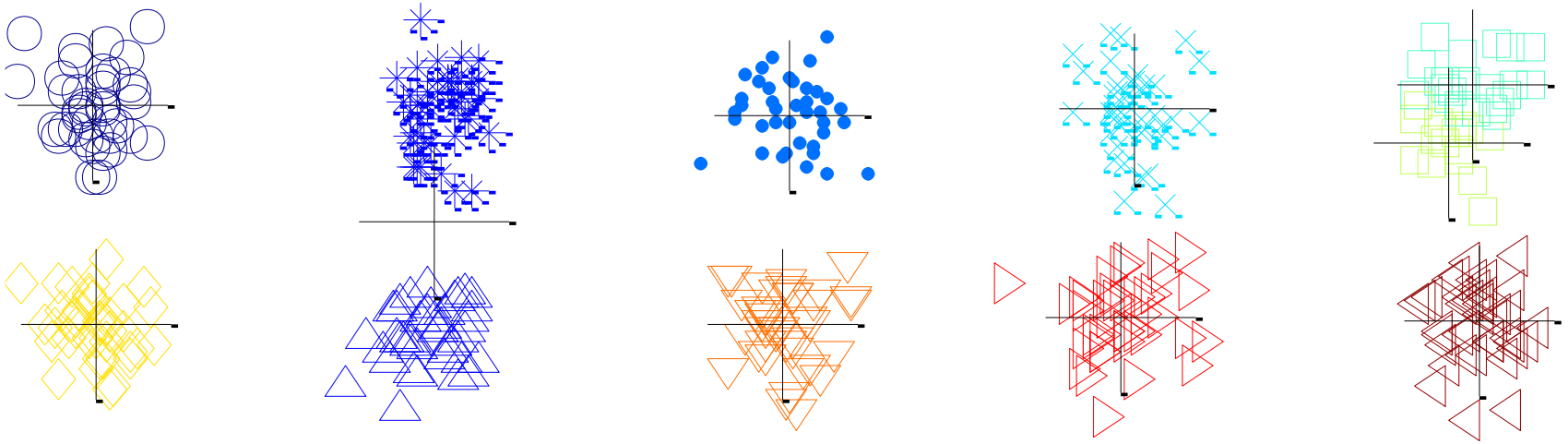
Starting with two initial centroids in one cluster of each pair of clusters

10 Clusters Example



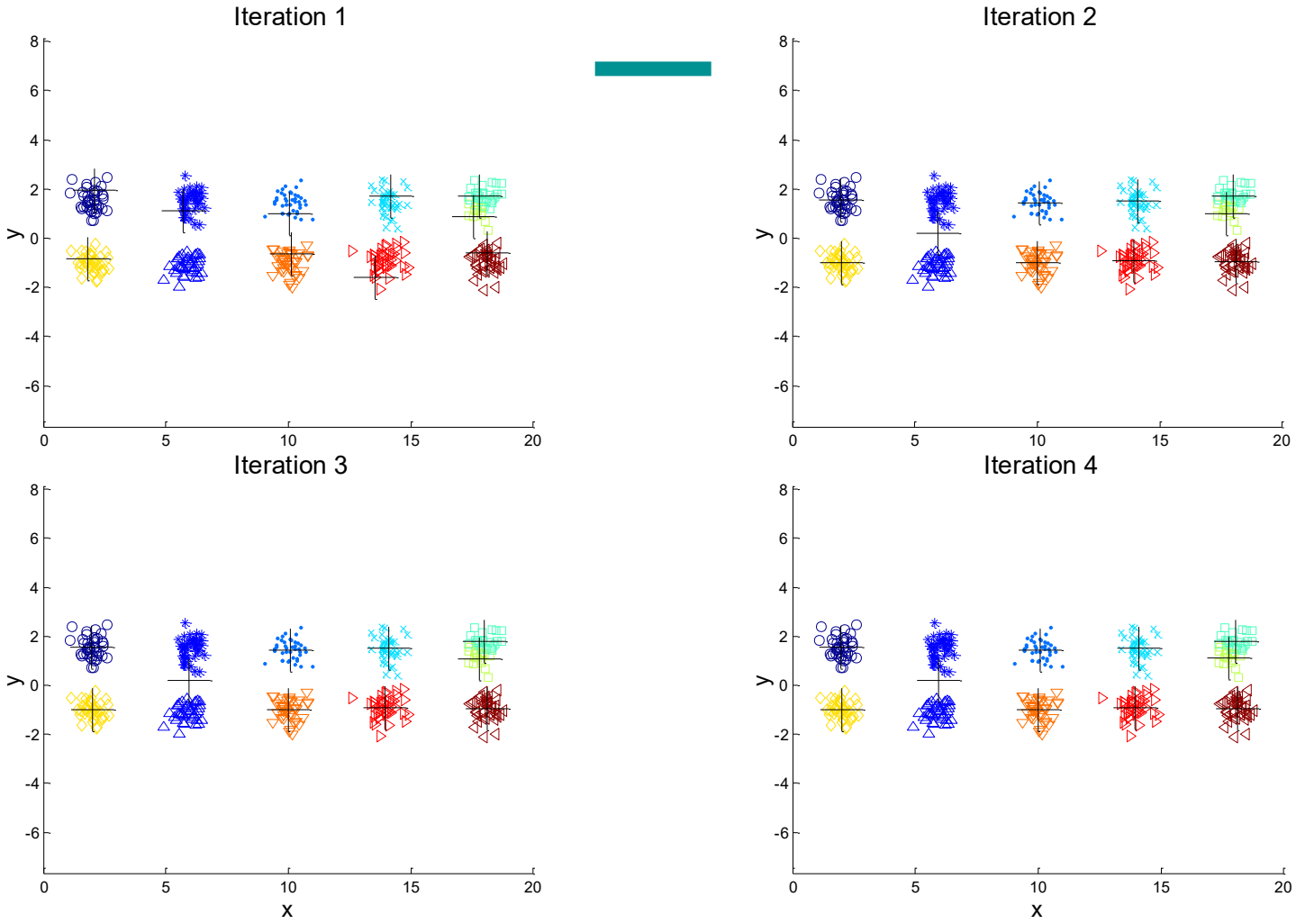
Starting with two initial centroids in one cluster of each pair of clusters

10 Clusters Example



Starting with some pairs of clusters having three initial centroids, while other have only one.

10 Clusters Example



Starting with some pairs of clusters having three initial centroids, while other have only one.

Solutions to Initial Centroids Problem

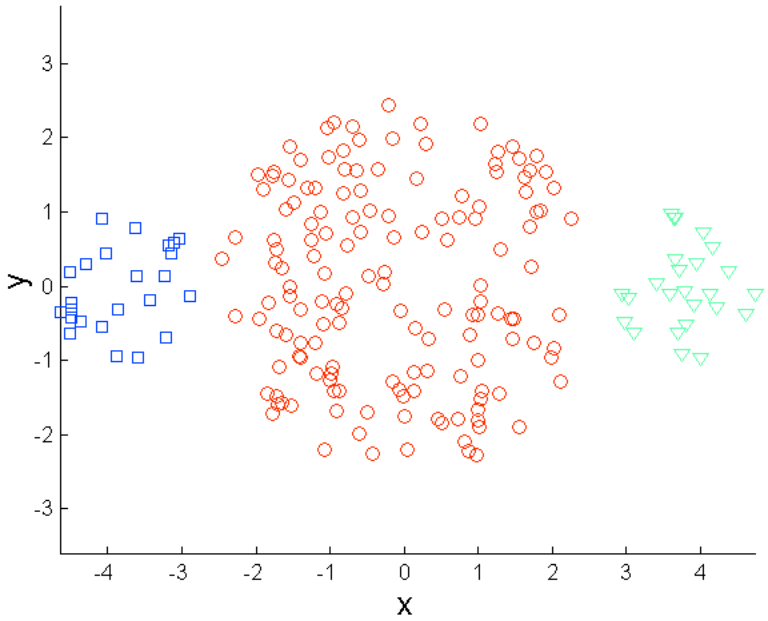
- Multiple runs
- Sample and use hierarchical clustering to determine initial centroids
- Select more than k initial centroids and then select among these initial centroids
  - Select most widely separated
- Postprocessing
- Bisecting K-means
  - Not as susceptible to initialization issues

Limitations of K-Means

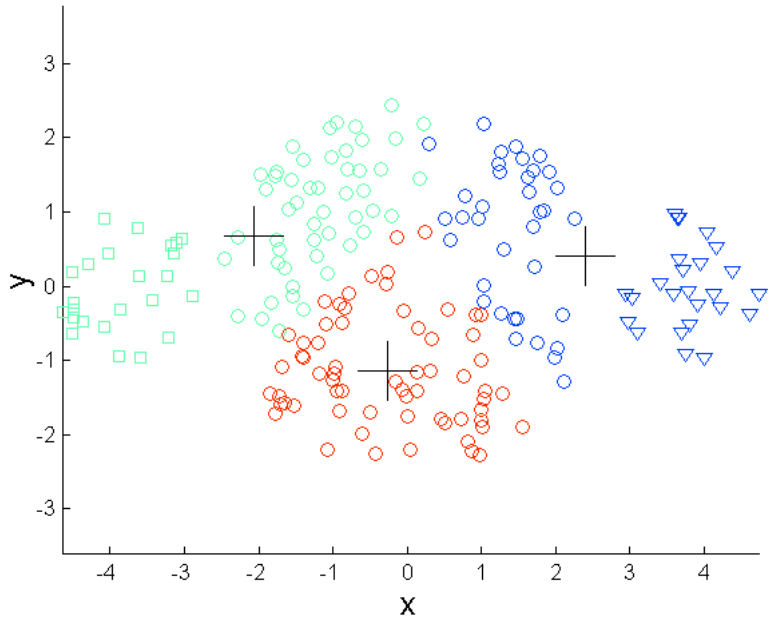
- K-Means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-globular shapes

- K-Means has problems when the data contains outliers.

Limitations of K-Means: Differing Sizes

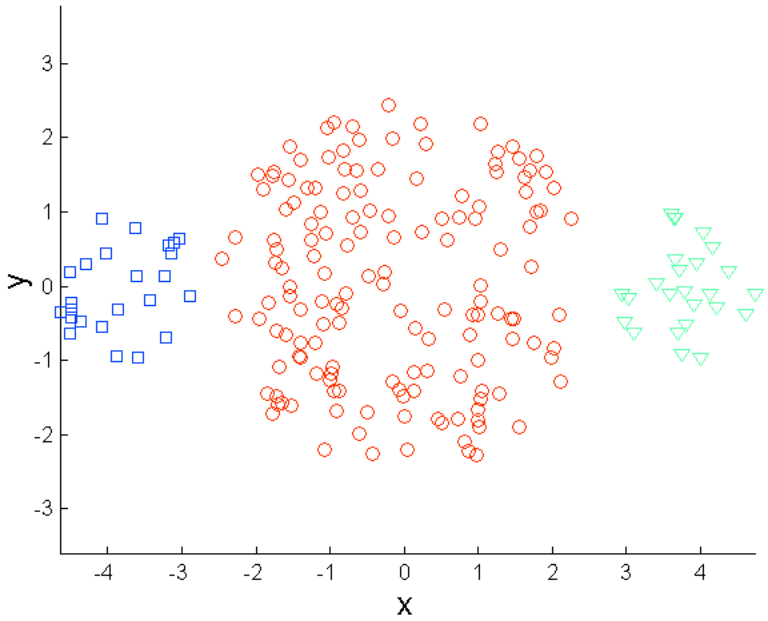


Original Points

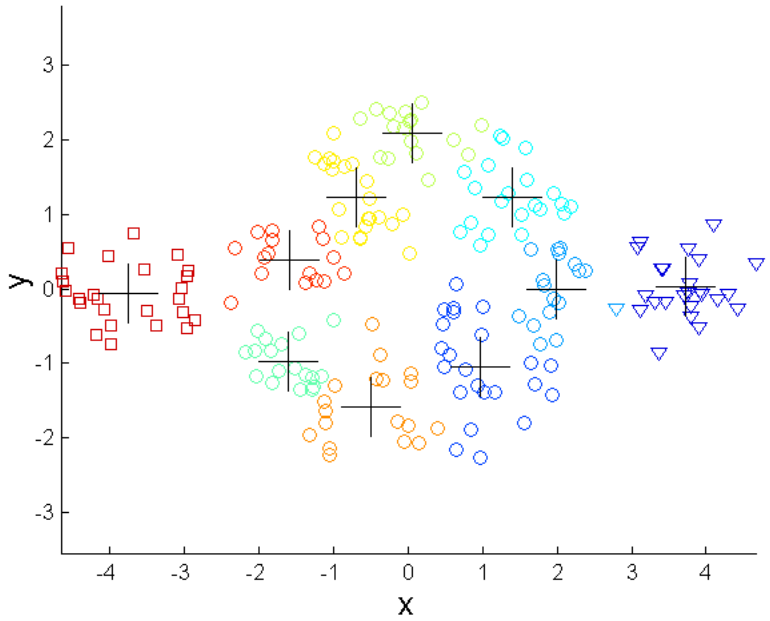


K-means (3 Clusters)

Overcoming K-Means Limitations



Original Points

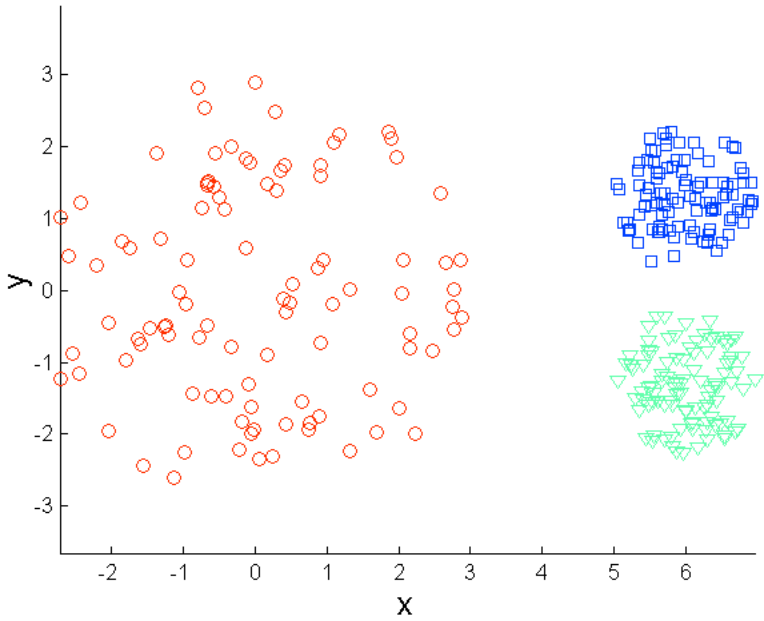


K-means Clusters

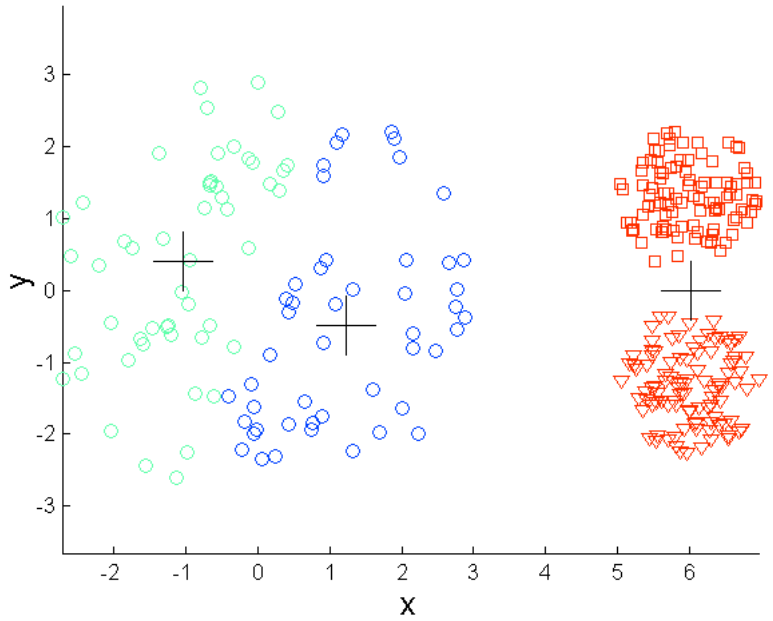
One solution is to use many clusters.

Find parts of clusters, but need to put together.

Limitations of K-Means: Differing Density

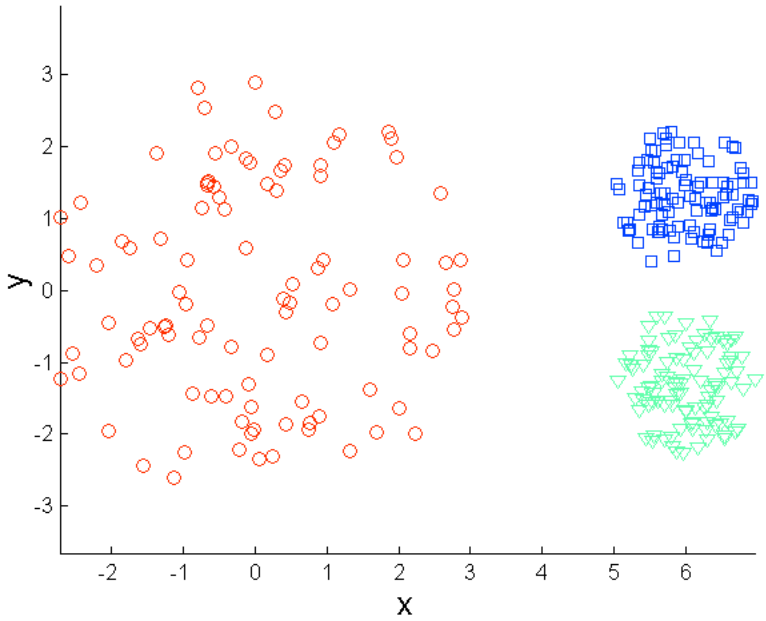


Original Points

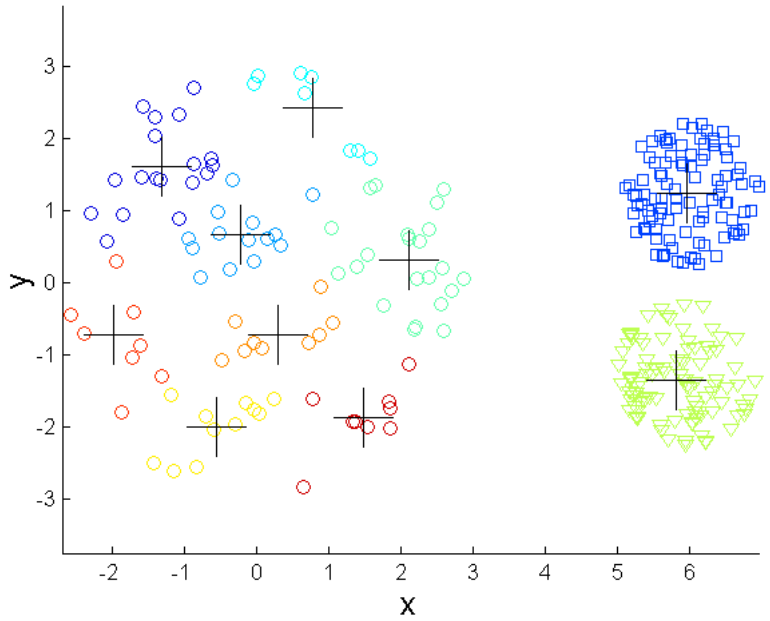


K-means (3 Clusters)

Overcoming K-Means Limitations

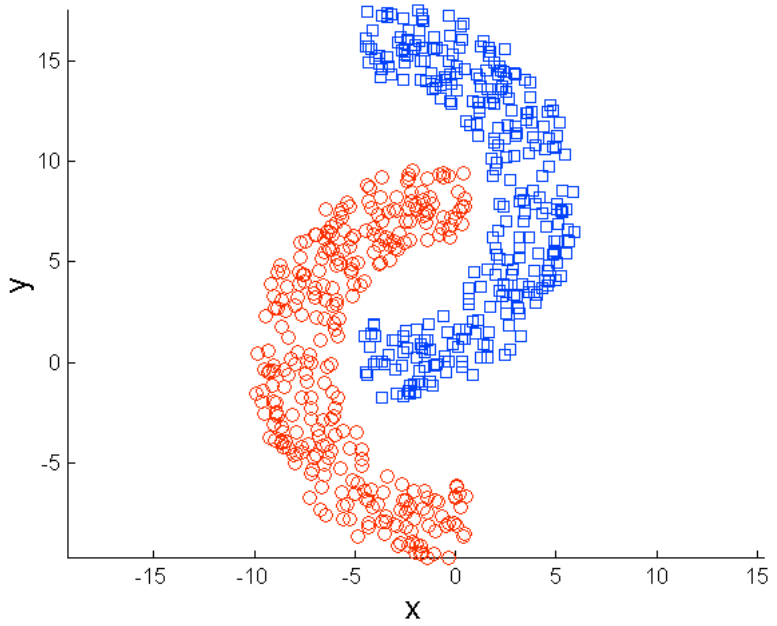


Original Points

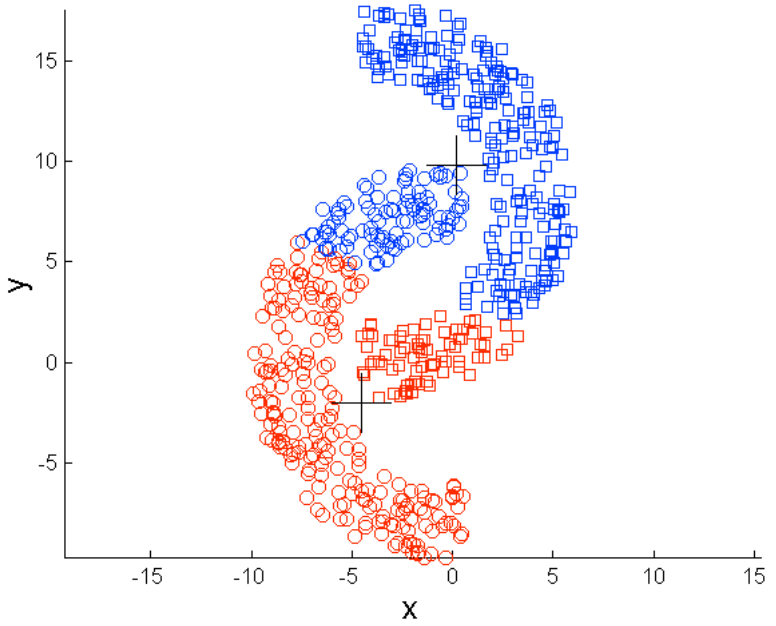


K-means Clusters

Limitations of K-Means: Non-globular Shapes

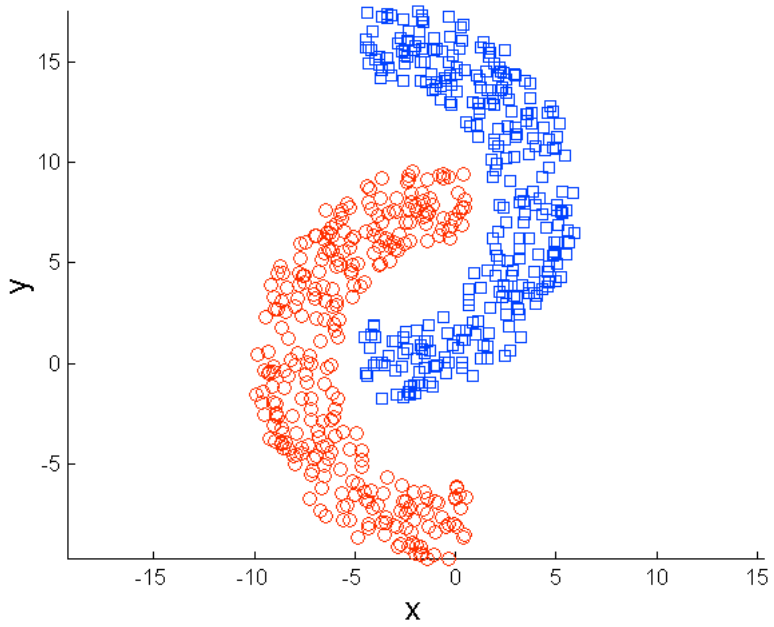


Original Points

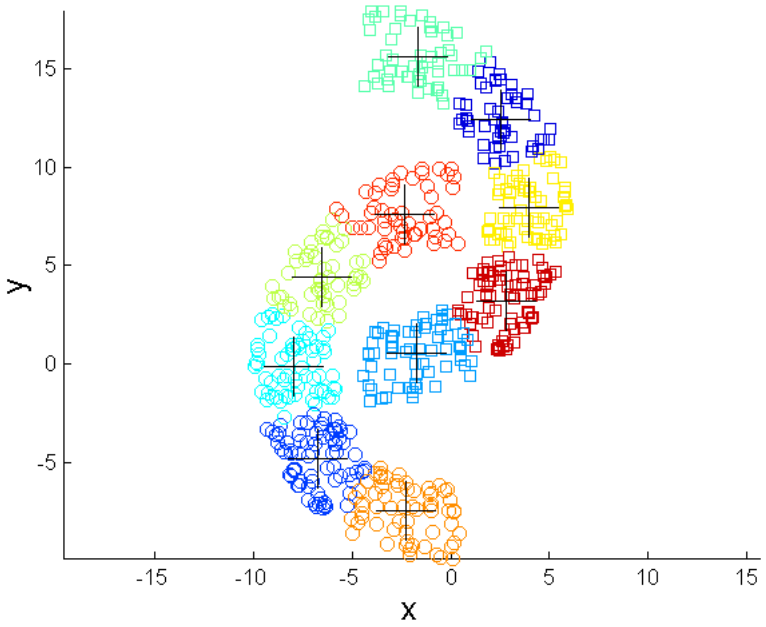


K-means (2 Clusters)

Overcoming K-Means Limitations



Original Points



K-means Clusters

Pre-processing and Post-processing

- Pre-processing
  - Normalize the data
  - Eliminate outliers
- Post-processing
  - Eliminate small clusters that may represent outliers
  - Split ‘loose’ clusters, i.e., clusters with relatively high SSE
  - Merge clusters that are ‘close’ and that have relatively low SSE
  - Can use these steps during the clustering process
    - ISODATA

K-Means Extensions

Bisecting K-Means

Bisecting K-means

- Variant of K-Means that can produce a hierarchical clustering
- The number of clusters K must be specified.
- Start with a unique cluster containing all the points.

1: Initialize the list of clusters to contain the cluster containing all points.
2: **repeat**
3: Select the cluster with the highest SSE to the list of clusters
4: **for** $i = 1$ to *number_of_iterations* **do**
5: Bisect the selected cluster using basic 2-Means
6: **end for**
7: Add the two clusters from the bisection to the list of clusters.
8: **until** Until the list of clusters contains K clusters

Bisecting K-means Limitations

- The algorithm can be also exhaustive and terminating at a singleton clusters if K is not specified.
- Terminating at singleton clusters
  - Is time consuming
  - Singleton clusters are meaningless (i.e., over-splitting)
  - Intermediate clusters are more likely to correspond to real classes

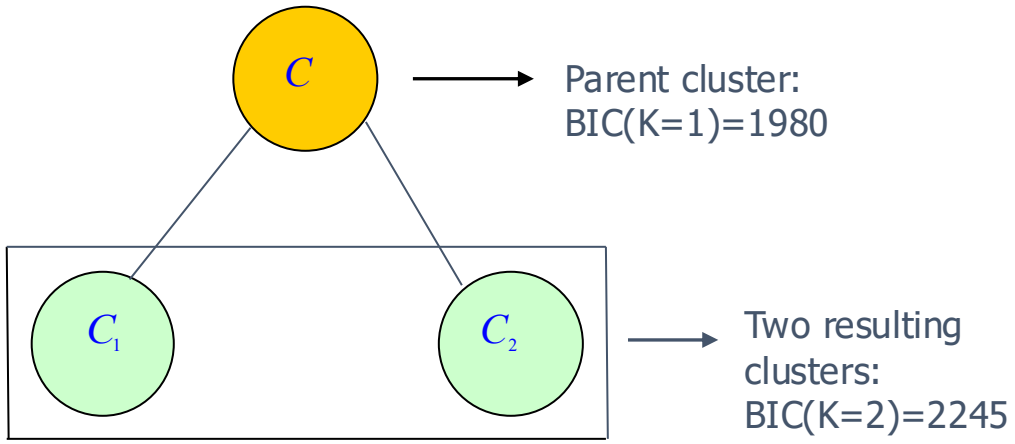
- Bisecting K-Means do not use any criterion for stopping bisections before singleton clusters are reached.

K-Means Extensions

X-Means

Bayesian Information Criterion (BIC)

- A strategy to stop the Bisecting algorithm when meaningful clusters are reached to avoid over-splitting.
- The **BIC** can be adopted as **splitting criterion** of a cluster in order to decide whether a cluster should split or no.
- BIC **measures the improvement** of the cluster structure between a cluster and its two children clusters.
- If the BIC of the parent is less than BIC of the children than we accept the bisection.



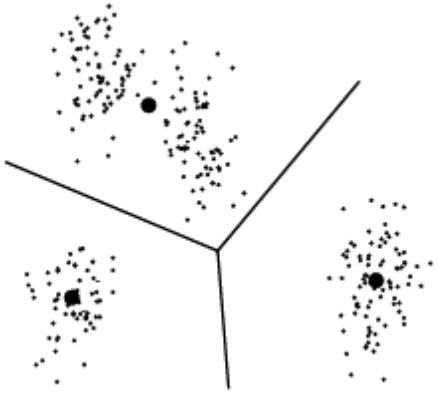
X-Means

For k in a given range $[r_1, r_{max}]$:

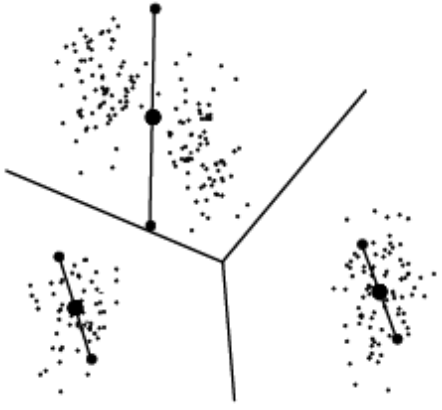
1. Improve Params: run K-Means with with the current k.
2. Improve Structure: recursively split each cluster in two (Bisecting 2-Means) and use *local BIC* to decide to keep the split. Stop if the current structure does not respect *local BIC* or the number of clusters is higher than r_{max}.
3. Store the actual configuration with a global BIC calculated on the whole configuration
4. If $k > r_{max}$ stop and return the best model w.r.t. the *global BIC*.

X-Means

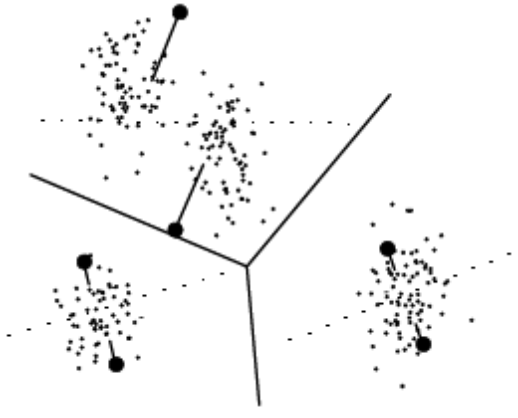
1. K-means with k=3



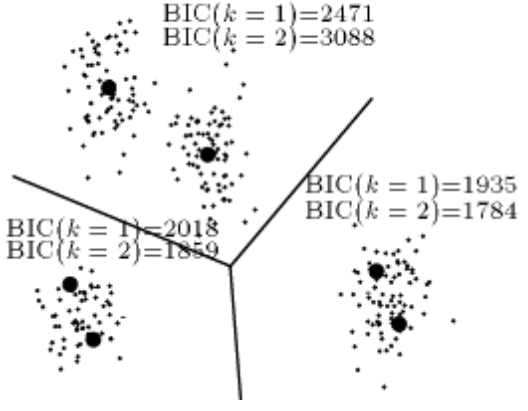
2. Split each centroid in 2 children moved a distance proportional to the region size in opposite direction (random)



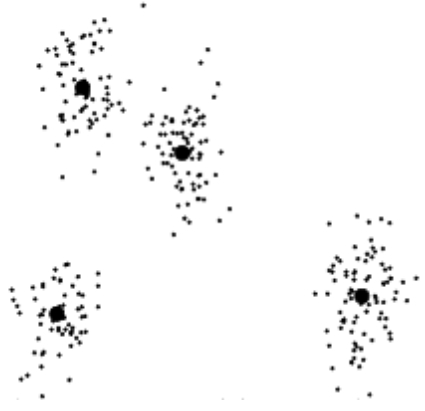
3. Run 2-means in each region locally



4. Compare BIC of parent and children



4. Only centroids with higher BIC survives



BIC Formula in X-Means

- The BIC score of a data collection is defined as (Kass and Wasserman, 1995):

$$BIC(M_j) = \hat{l}_j(D) - \frac{p_j}{2}\log R$$

- $\hat{l}_j(D)$ is the log-likelihood of the dataset D
- p_j is a function of the number of independent parameters: centroids coordinates, variance estimation.
- R is the number of points of a cluster, M is the number of dimensions
- Approximate the probability that the clustering in M_j is describing the real clusters in the data

BIC Formula in X-Means

- Adjusted Log-likelihood of the model.
- **The likelihood that the data is "explained by" the clusters** according to the spherical-Gaussian assumption of K-Means

$$BIC(M_j) = \hat{l}_j(D) - \frac{p_j}{2}\log R$$

- Focusing on the set D_n of points which belong to centroid *n*

$$\begin{aligned}\hat{l}(D_n) \quad = \quad & -\frac{R_n}{2}\log(2\pi) - \frac{R_n \cdot M}{2}\log(\hat{\sigma}^2) - \frac{R_n - K}{2} \\ & + R_n \log R_n - R_n \log R\end{aligned}$$

- It estimates how closely to the centroid are the points of the cluster.

K-Means Origins

Expectation Maximization

Model-based Clustering (probabilistic)

- In order to understand our data, we will assume that there is a generative process (a model) that creates/describes the data, and we will try to find the model that **best fits** the data.
  - Models of different complexity can be defined, but we will assume that our model is a distribution from which data points are sampled
  - **Example**: the data is the height of all people in **Greece**
- In most cases, a single distribution is not good enough to describe all data points**: different parts of the data follow a different distribution**
  - **Example**: the data is the height of all people in Greece and China
  - We need a mixture model
  - Different distributions correspond to different clusters in the data.

Expectation Maximization Algorithm

- Initialize the values of the parameters in Θ to some random values
- Repeat until convergence
  - **E-Step**: Given the parameters Θ **estimate** the membership probabilities $\mathrm{P}(G_j|x_i)$
  - **M-Step:** Given the probabilities $\mathrm{P}(G_j|x_i)$, calculate the parameter values Θ that (in expectation) **maximize** the data likelihood

- **Examples**
  - **E-Step**: Assignment of points to clusters
    - K-Means: **hard** assignment, EM: soft assignment
  - **M-Step**: Parameters estimation
    - K-Means: Computation of centroids, EM: Computation of the new model parameters

EM in K-Means

centroids

- Initialize the values of the parameters in Θ to some random values (randomly select the centroids)
- Repeat until convergence
  - **E-Step**: Given the parameters Θ (given the centroids) **estimate** the membership probabilities $\mathrm{P}(G_j|x_i)$ (assign points to clusters based on distances with the centroids)
  - **M-Step:** Given the probabilities $\mathrm{P}(G_j|x_i)$ (given the membership of points to clusters, i.e., 100% probability of belonging to a cluster) calculate the parameter values Θ that (in expectation) **maximize** the data likelihood (calculate the new centroids as mean values, i.e., those that minimize the distances with the other points in the cluster)

Expectation Maximization Algorithm

Algorithm 9.2 EM algorithm.

1: Select an initial set of model parameters.
(As with K-means, this can be done randomly or in a variety of ways.)
2: **repeat**
3: **Expectation Step** For each object, calculate the probability that each object belongs to each distribution, i.e., calculate $prob(distribution\ j|\mathbf{x}_i, \Theta)$.
4: **Maximization Step** Given the probabilities from the expectation step, find the new estimates of the parameters that maximize the expected likelihood.
5: **until** The parameters do not change.
(Alternatively, stop if the change in the parameters is below a specified threshold.)

K-Means Brother

K-Modes

K-Modes

$$\text{Minimise} \quad P(W, \boldsymbol{Q}) = \sum_{l=1}^{k} \sum_{i=1}^{n} w_{i,l}\, d(X_i, Q_l)$$

$$\text{subject to} \quad \sum_{l=1}^{k} w_{i,l} = 1, \quad 1 \le i \le n$$

$$w_{i,l} \in \{0, 1\}, \quad 1 \le i \le n,\ 1 \le l \le k$$

- $X = \{ X_1, \ldots, X_n \}$ is the dataset of objects.
- $X_i = [x_1, \ldots, x_m]$ is an object i.e., a vector of m categorical attributes
- W is a matrix $n \times k$, with $w_{i,l}$ equal to 1 if X_i belongs to Cluster l, 0 otherwise.
- $Q = \{ Q_1, \ldots, Q_k \}$ is the set of representative objects (mode) for the k clusters.
- $d(X_i , Q_l)$ is a distance function for objects in the data

K-Modes: Distance

- K-Means as distance uses Euclidean distance

$$d(X,Y) = \square_{i=1}^{m} (x_i - y_i)^2$$

- K-Modes as distance uses the number of mismatches between the attributes of two objects.

$$d_1(X, Y) = \sum_{j=1}^{m} \delta(x_j, y_j)$$

$$\delta(x_j, y_j) = \begin{cases} 0 & (x_j = y_j) \\ 1 & (x_j \neq y_j) \end{cases}$$

K-Modes: Mode

- K-Modes uses the mode as representative object of a cluster
- Given the set of objects in the cluster C_l the mode is get computing the max frequency for each attribute

$$f_r(A_j = c_{l,j} \mid X_l) = \frac{n_{c_{l,k}}}{n}$$

K-Modes: Algorithm

1. Randomly select the initial objects as modes
2. Scan of the data to assign each object to the closer cluster identified by the mode
3. Re-compute the mode of each cluster
4. Repeat the steps 2 and 3 until no object changes the assigned cluster

K-Means Brother

Mixture Gaussian Model

Gaussian Distribution

- Example: the data is the height of all people in Greece
- Experience has shown that this data follows a Gaussian (Normal) distribution

$$P(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- μ = mean, σ = standard deviation

Mixture Gaussian Model

- What is a model?
  - A Gaussian distribution is defined by the mean μ and the standard deviation σ
  - We define our model as the pair of parameters $\theta = (\mu, \sigma)$
- More generally, a model is defined as a vector of parameters θ

- We want to find the normal distribution $N(\mu, \sigma)$ that best fits our data
  - Find the best values for μ and σ
  - But what does "best fit" mean?

Maximum Likelihood Estimation (MLE)

- Suppose that we have a vector $X = \{x_1, \dots, x_n\}$ of values
- We want to fit a Gaussian model $N(\mu, \sigma)$ to the data
- Probability of observing a point x_i

$$P(x_i) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

- Probability of observing all points (we assume independence)

$$P(X) = \prod_{i=1}^{n} P(x_i) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

- We want to find the parameters $\theta = (\mu, \sigma)$ that maximizes the probability $P(X|\theta)$

Maximum Likelihood Estimation (MLE)

- The probability $P(X|\theta)$ as a function of θ is the **Likelihood** function

$$L(\theta) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

- It is usually easier to work with the **Log-Likelihood** function

$$LL(\theta) = -\sum_{i=1}^{n} \frac{(x_i-\mu)^2}{2\sigma^2} - \frac{1}{2} n \log 2\pi - n \log \sigma$$

- Thus, the Maximum Likelihood Estimation for the Gaussian Model consists in finding the parameters μ, σ that maximize $LL(\theta)$

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i = \mu_X$$

Sample Mean

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n} (x_i-\mu)^2 = \sigma_X^2$$

Sample Variance

Maximum Likelihood Estimation (MLE)

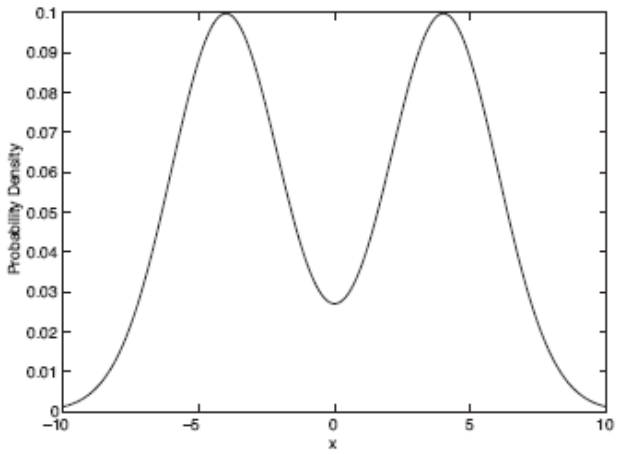
- Note: these are also the most likely parameters given the data.

$$P(\theta|X) = \frac{P(X|\theta)P(\theta)}{P(X)}$$

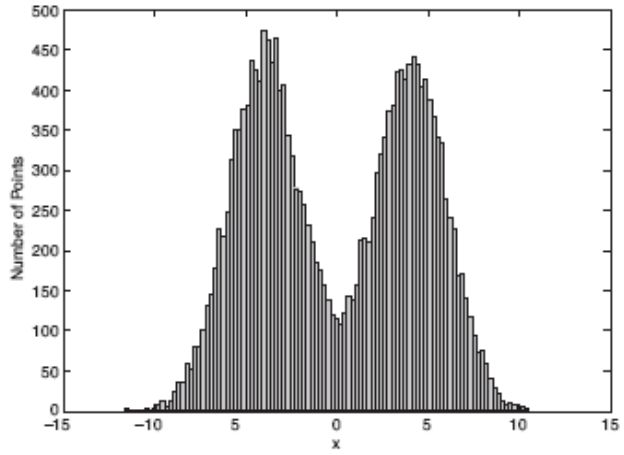
- If we have no prior information about θ, or X, then maximizing $P(\theta|X)$ is the same as maximizing $P(X|\theta)$.

Mixture of Gaussians

- Suppose that you have the heights of people from Greece and China and the distribution looks like the figure below (dramatization)



(a) Probability density function for the mixture model.

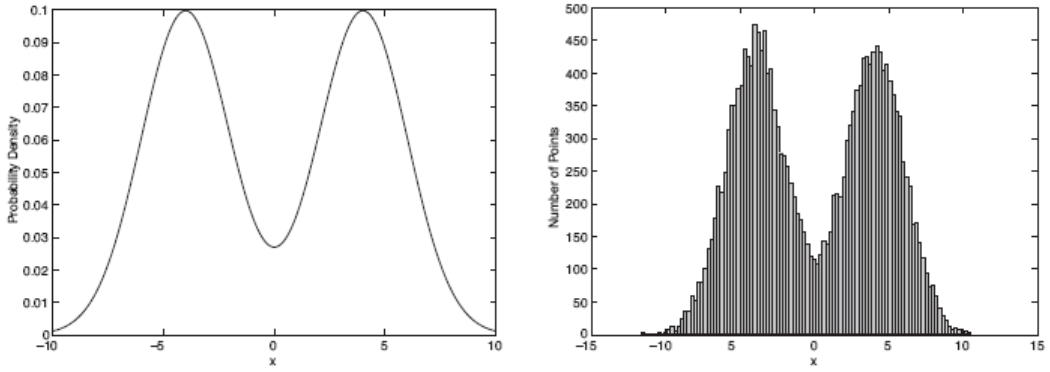


(b) 20,000 points generated from the mixture model.

Figure 9.2. Mixture model consisting of two normal distributions with means of -4 and 4, respectively. Both distributions have a standard deviation of 2.

Mixture of Gaussians

- In this case the data is the result of the **mixture** of two Gaussians
  - One for Greek people, and one for Chinese people
  - Identifying for each value which Gaussian is most likely to have generated it will give us a clustering.



(a) Probability density function for the mixture model.

(b) 20,000 points generated from the mixture model.

Figure 9.2. Mixture model consisting of two normal distributions with means of -4 and 4, respectively. Both distributions have a standard deviation of 2.

Mixture Model

- A value x_i is generated according to the following process:
  - First select the nationality
    - With probability π_G select Greek, with probability π_C select China ($\pi_G + \pi_C = 1$)
  - Given the nationality, generate the point from the corresponding Gaussian
    - $P(x_i|\theta_G) \sim N(\mu_G, \sigma_G)$ if Greece
    - $P(x_i|\theta_C) \sim N(\mu_C, \sigma_C)$ if China

Mixture Model

Assign a point to a cluster. In K-Means they are the membership: hard assignment.

Describe a cluster. In K-Means they are the centroids.

- Our model has the following parameters

$$\Theta = (\pi_G, \pi_C, \mu_G, \mu_C, \sigma_G, \sigma_C)$$

Mixture probabilities | Distribution Parameters

- For value x_i, we have:

$$P(x_i|\Theta) = \pi_G P(x_i|\theta_G) + \pi_C P(x_i|\theta_C)$$

- For all values $X = \{x_1, \dots, x_n\}$

$$P(X|\Theta) = \prod_{i=1}^{n} P(x_i|\Theta)$$

- We want to estimate the parameters that **maximize** the Likelihood

Mixture Model

- Once we have the parameters $\theta = (\pi_G, \pi_C, \mu_G, \sigma_G, \mu_C, \sigma_C)$, we can **estimate** the **membership probabilities** $P(G|x_i)$ and $P(C|x_i)$ for each point x_i:

- This is the probability that point x_i belongs to the Greek or the Chinese population (cluster)

$$P(G|x_i) = \frac{P(x_i|G)P(G)}{P(x_i|G)P(G) + P(x_i|C)P(C)}$$
$$= \frac{P(x_i|G)\pi_G}{P(x_i|G)\pi_G + P(x_i|C)\pi_C}$$

Mixture of Gaussians as EM

- Initialize the values of the parameters in θ to some random values
- Repeat until convergence
  - **E-Step**: Given the parameters Θ **estimate** the membership probabilities $\mathrm{P}(G|x_i)$ and $\mathrm{P}(C|x_i)$.
  - **M-Step:** Calculate the parameter values Θ that (in expectation) **maximize** the data likelihood.

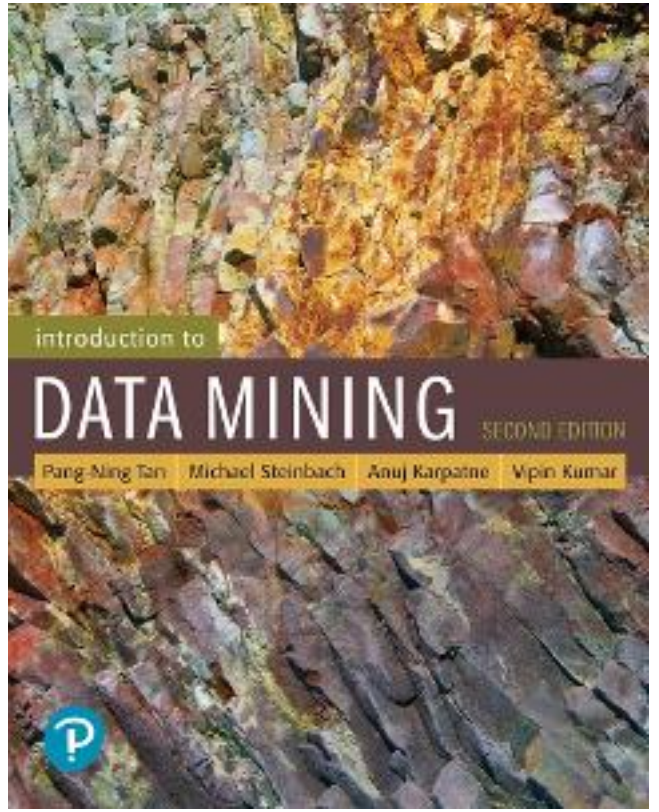
$$\pi_G = \frac{1}{n}\sum_{i=1}^{n} P(G|x_i) \qquad \pi_C = \frac{1}{n}\sum_{i=1}^{n} P(C|x_i)$$

Fraction of population in G,C

$$\mu_C = \sum_{i=1}^{n} \frac{P(C|x_i)}{n * \pi_C} x_i \qquad \mu_G = \sum_{i=1}^{n} \frac{P(G|x_i)}{n * \pi_G} x_i$$

MLE Estimates if π's were fixed

$$\sigma_C^2 = \sum_{i=1}^{n} \frac{P(C|x_i)}{n * \pi_C} (x_i - \mu_C)^2 \qquad \sigma_G^2 = \sum_{i=1}^{n} \frac{P(G|x_i)}{n * \pi_G} (x_i - \mu_G)^2$$

References

- Clustering. Chapter 7. Introduction to Data Mining.
- Pelleg, Dan, and Andrew W. Moore. "X-means: Extending k-means with efficient estimation of the number of clusters." 2000.