Data Mining
Cluster Analysis: Basic Concepts and Algorithms

Lecture Notes for Chapter 7

Introduction to Data Mining, 2nd Edition

by

Tan, Steinbach, Karpatne, Kumar

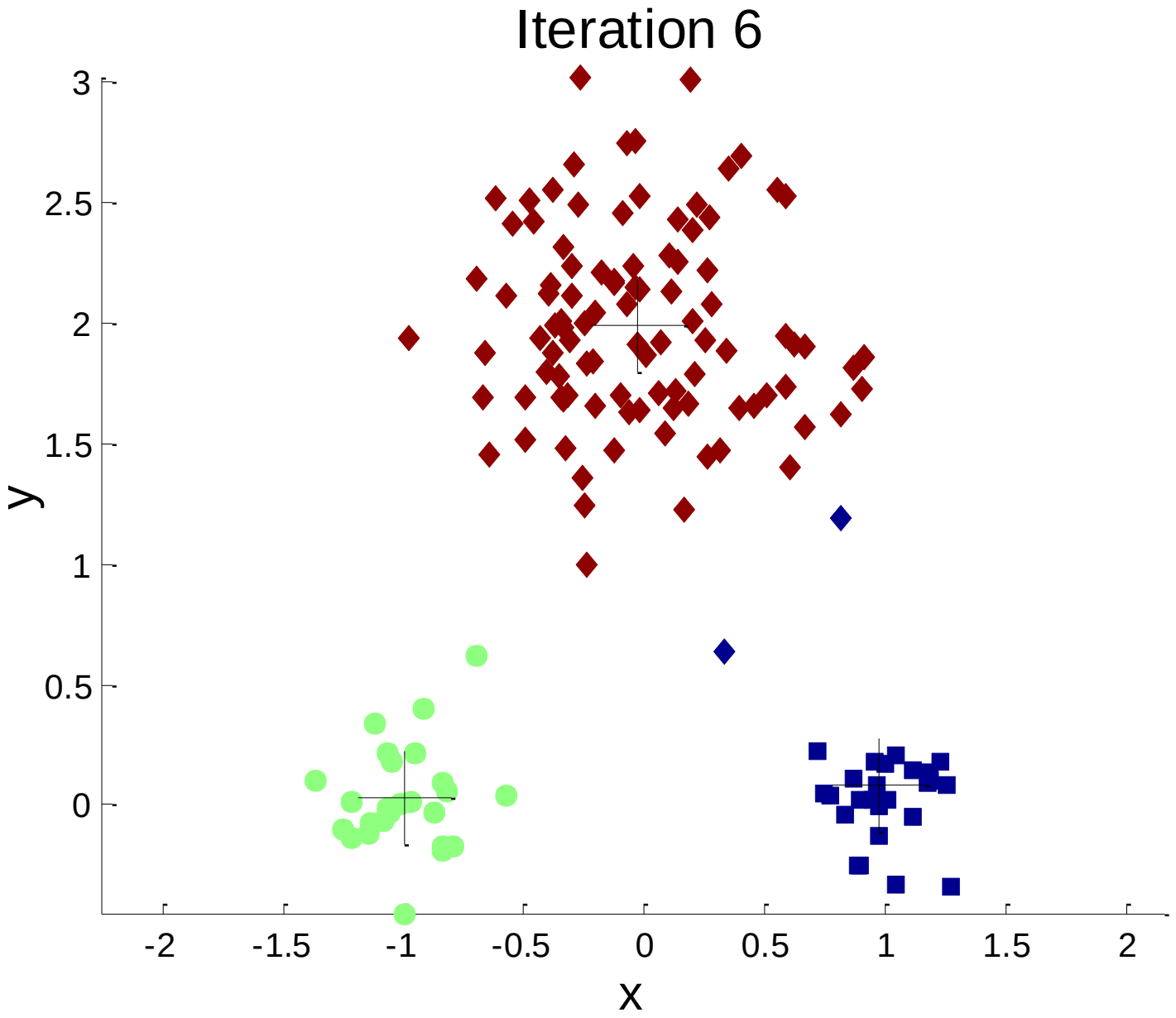
K-means

K-means Clustering

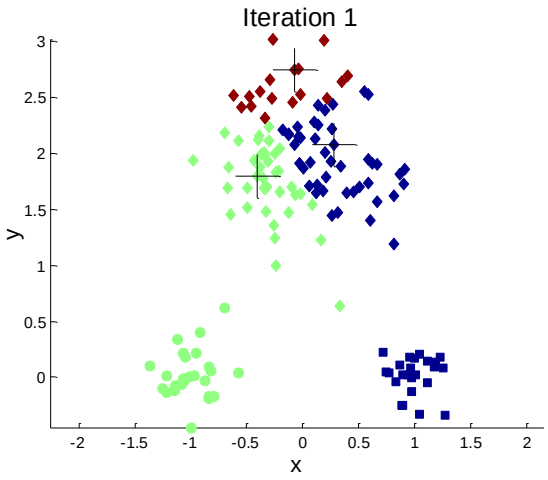
- Partitional clustering approach
- Number of clusters, K, must be specified
- Each cluster is associated with a **centroid** (center point)
- Each point is assigned to the cluster with the **closest centroid**
- The basic algorithm is very simple

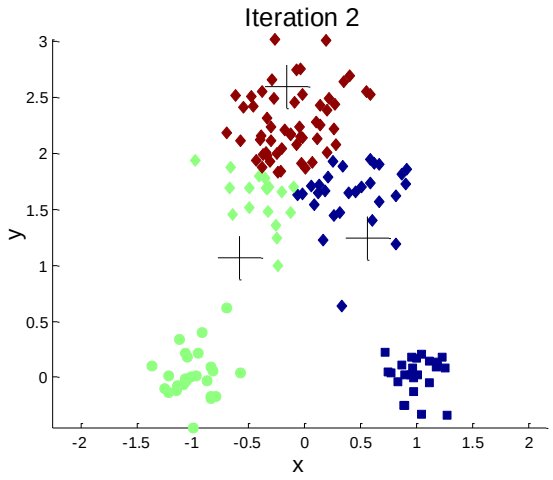
1: Select K points as the initial centroids.
2: **repeat**
3: Form K clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: **until** The centroids don't change

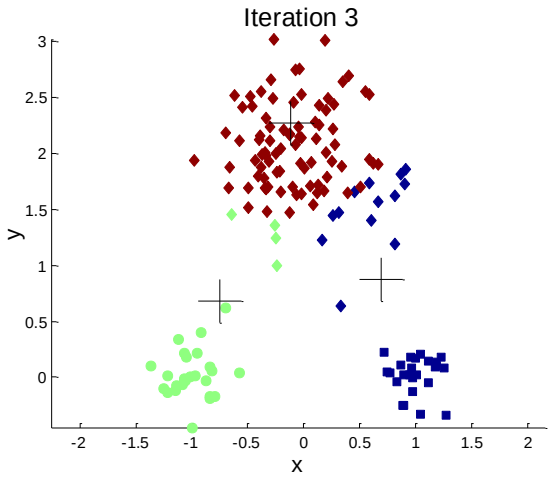
Example of K-means Clustering

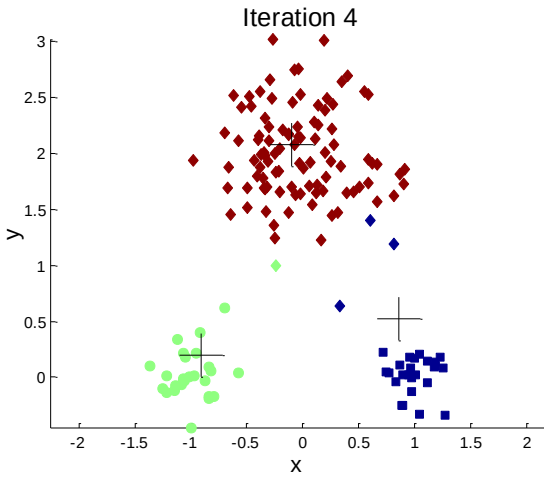


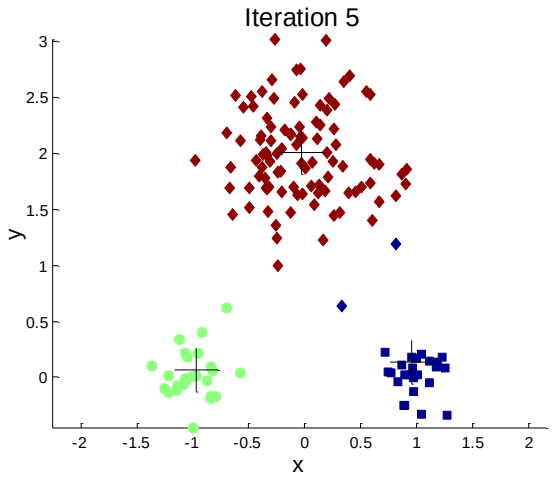
Example of K-means Clustering

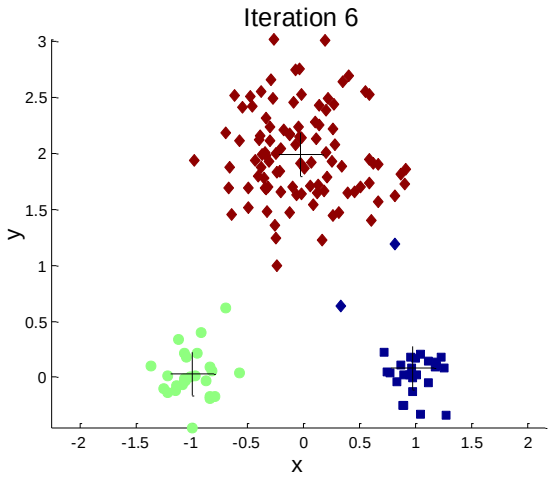












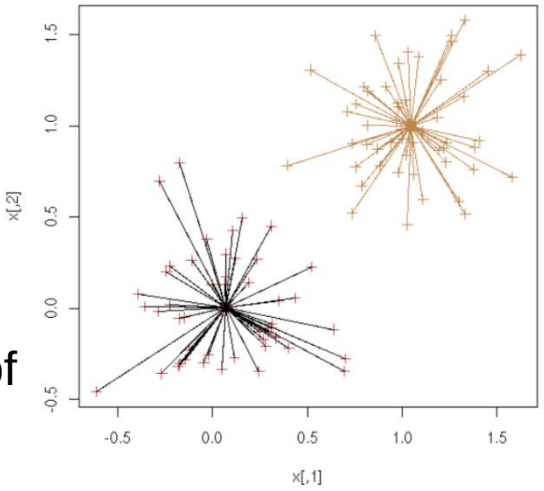
K-means Clustering – Details

- 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

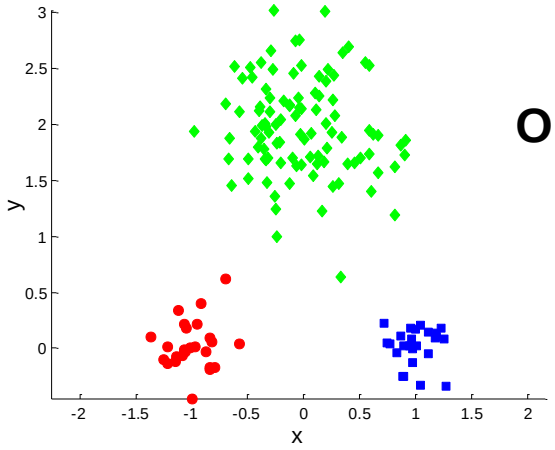
- Most common measure is Sum of Squared Error (SSE)
  - For each point, the error is the distance to the nearest cluster
  - To get SSE, we square these errors and sum them.

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

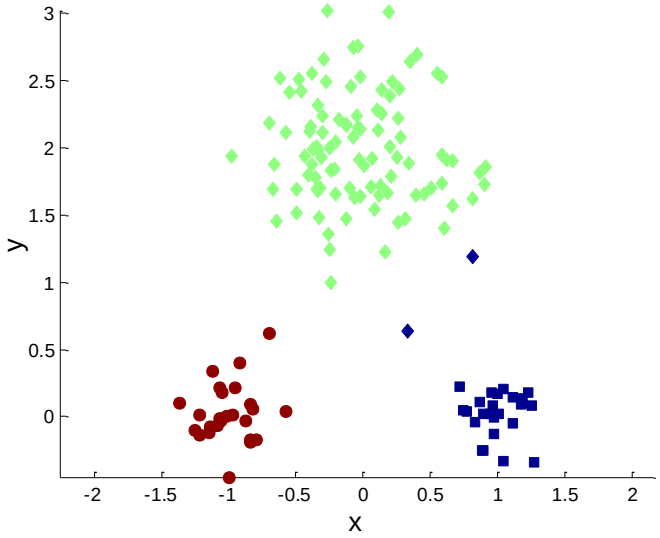


- x is a data point in cluster C_i and m_i is the representative point for cluster Ci
  - can show that mi corresponds to the center (mean) of the cluster
- Given two sets of clusters, we prefer the one with the smallest error
- One easy way to reduce SSE is to increase K, the number of clusters
- A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

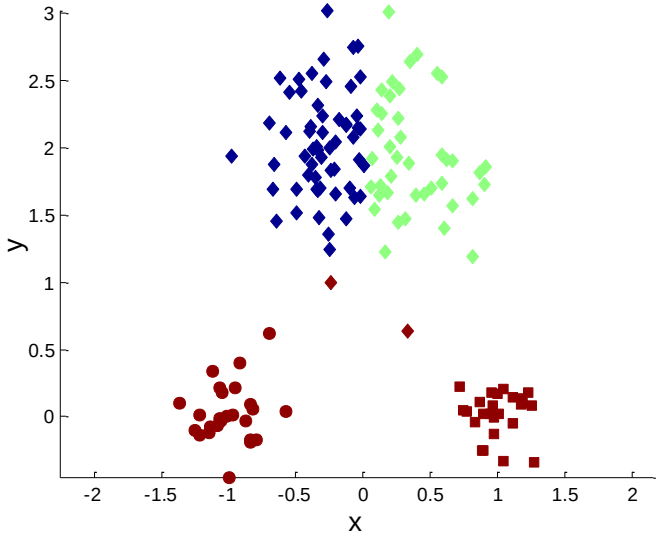
Two different K-means Clusterings



Original Points



Optimal Clustering



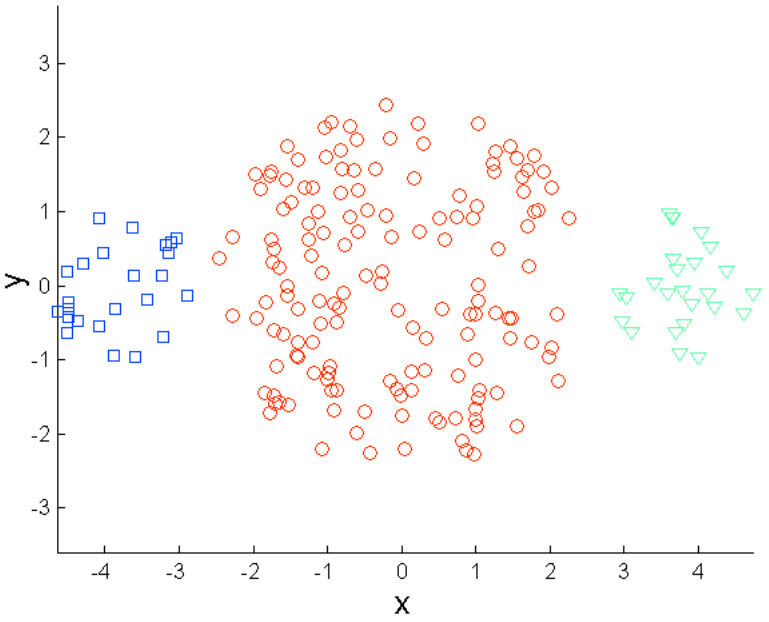
Sub-optimal Clustering

Limitations of K-means

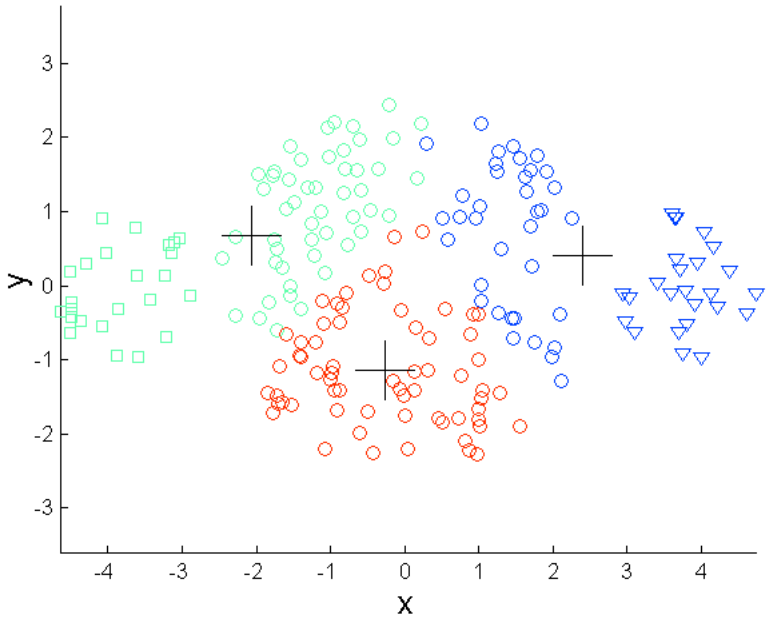
- K-means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-globular shapes

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

Limitations of K-means: Differing Sizes

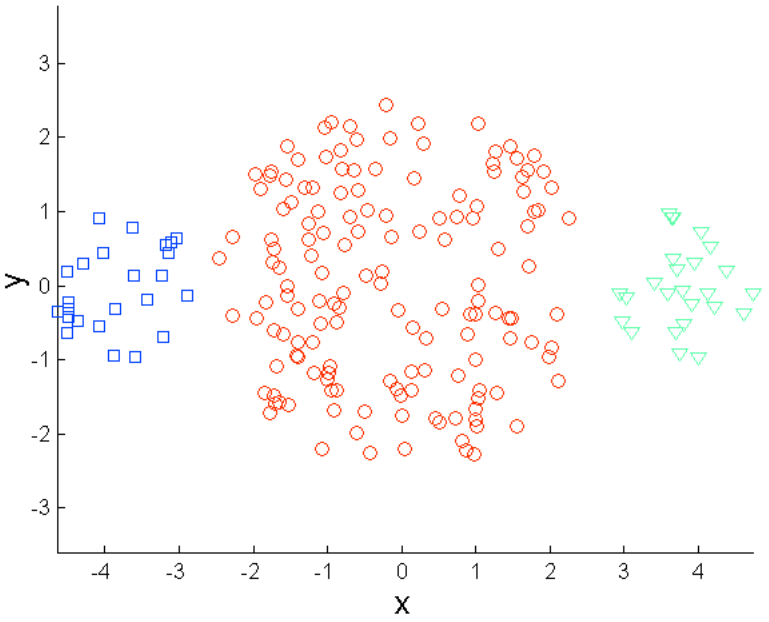


Original Points

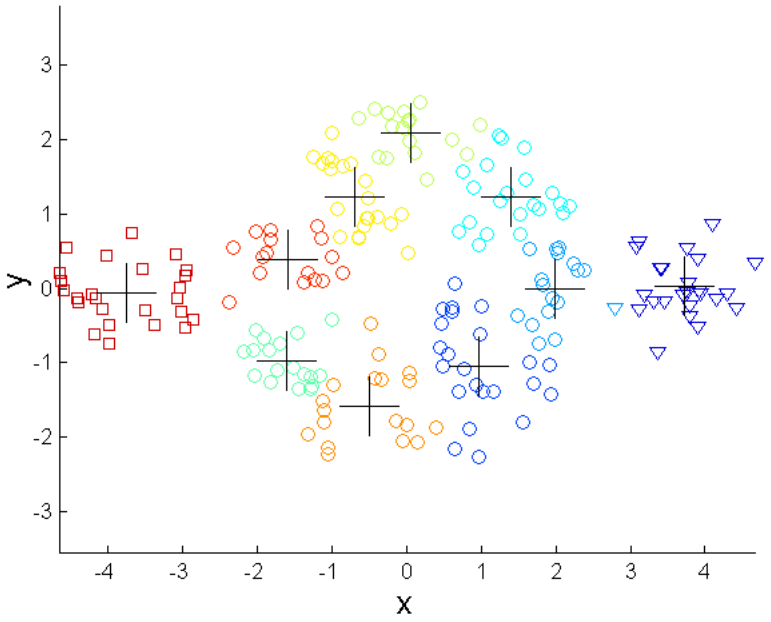


K-means (3 Clusters)

Overcoming K-means Limitations



Original Points

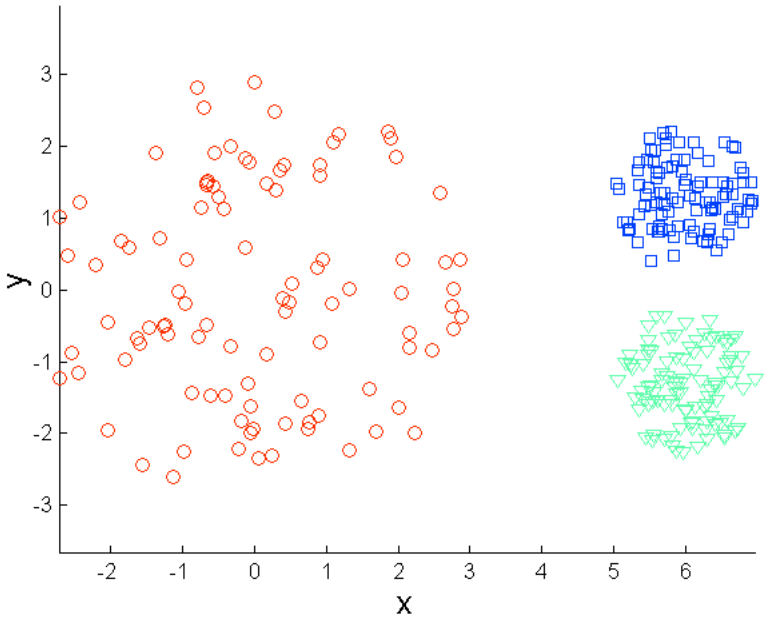


K-means Clusters

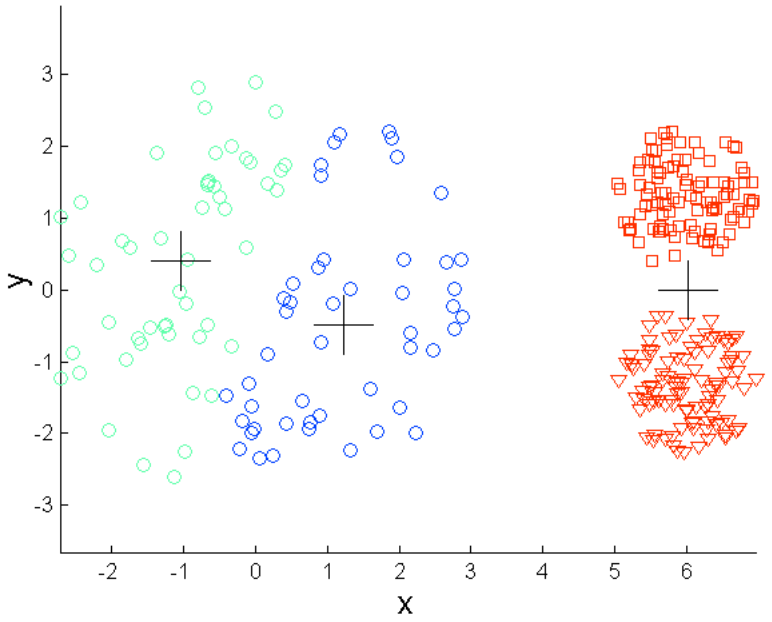
One solution is to use many clusters.

Find parts of clusters, but need to put together.

Limitations of K-means: Differing Density

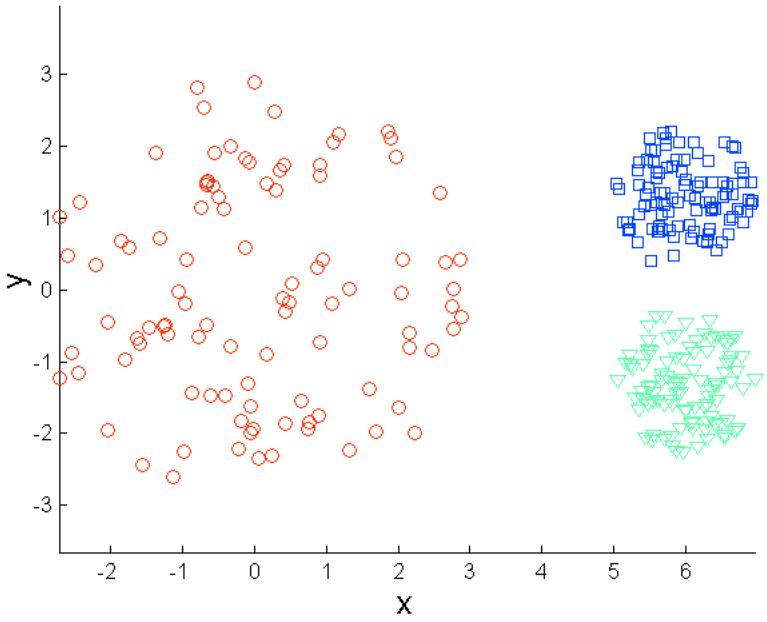


Original Points

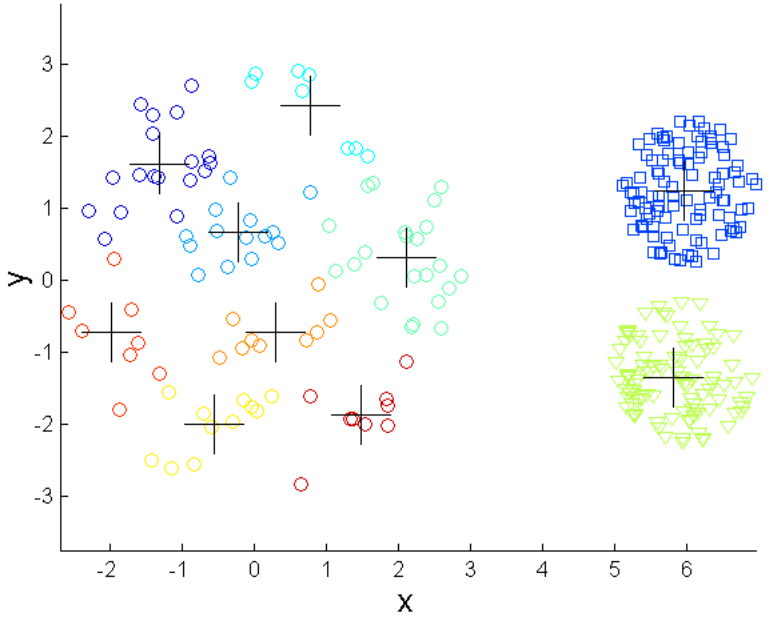


K-means (3 Clusters)

Overcoming K-means Limitations

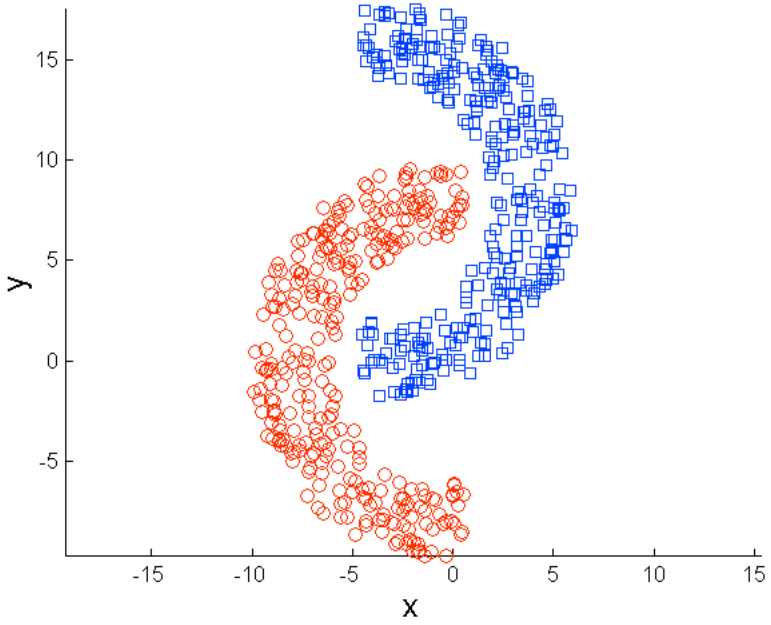


Original Points

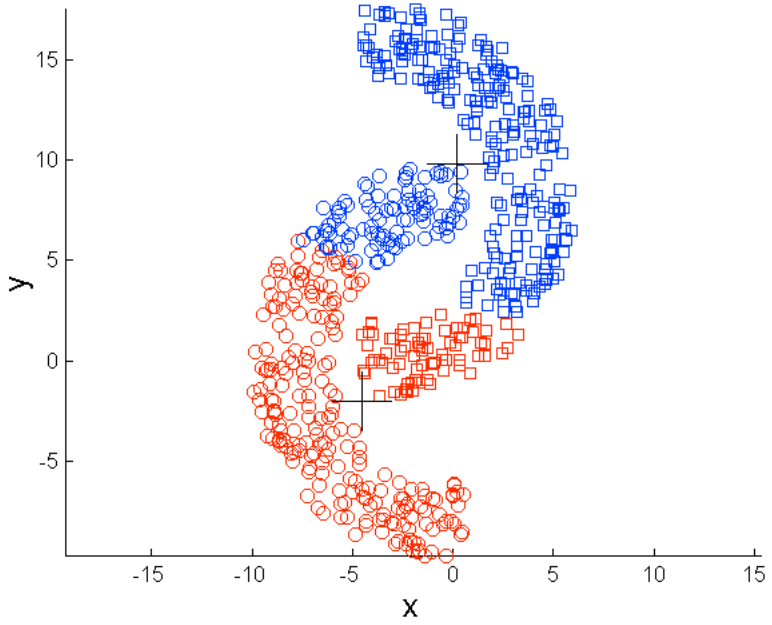


K-means Clusters

Limitations of K-means: Non-globular Shapes

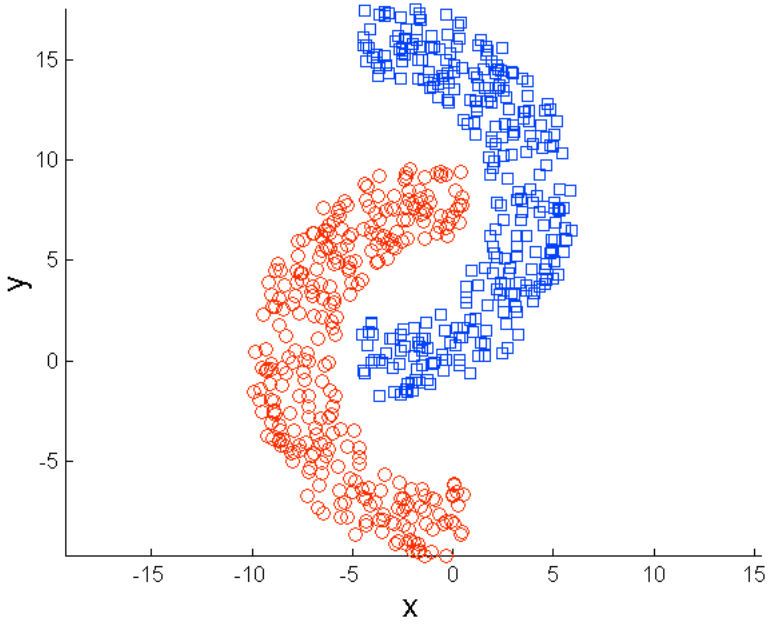


Original Points

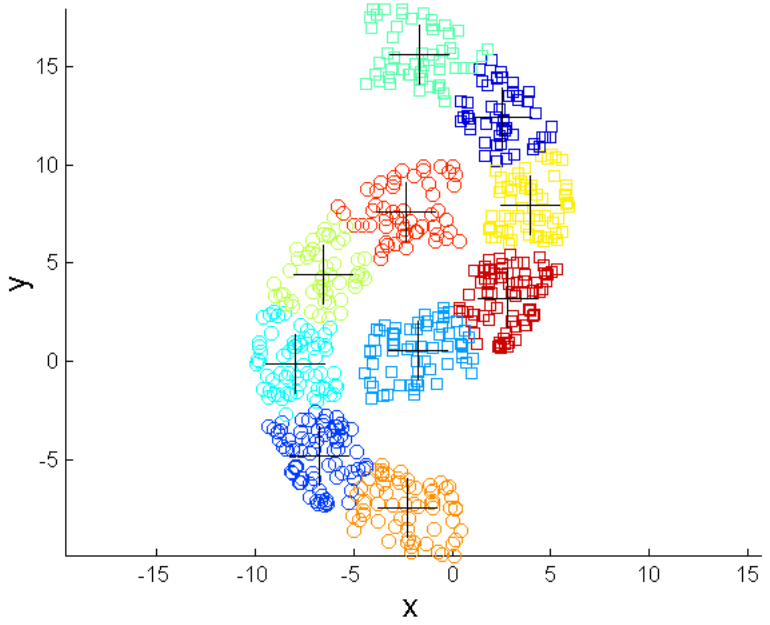


K-means (2 Clusters)

Overcoming K-means Limitations



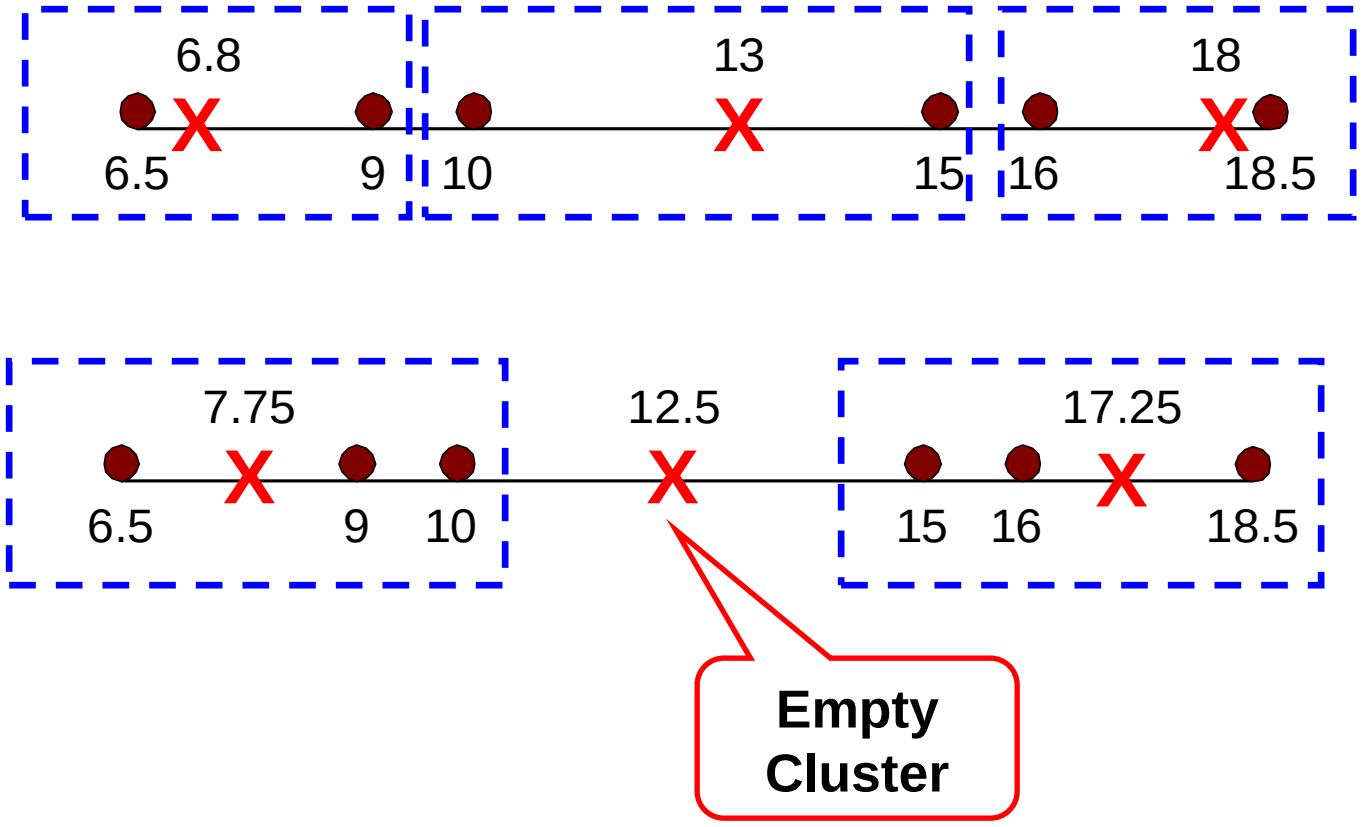
Original Points



K-means Clusters

Empty Clusters

- K-means can yield empty clusters



Handling Empty Clusters

- Basic K-means algorithm can yield empty clusters

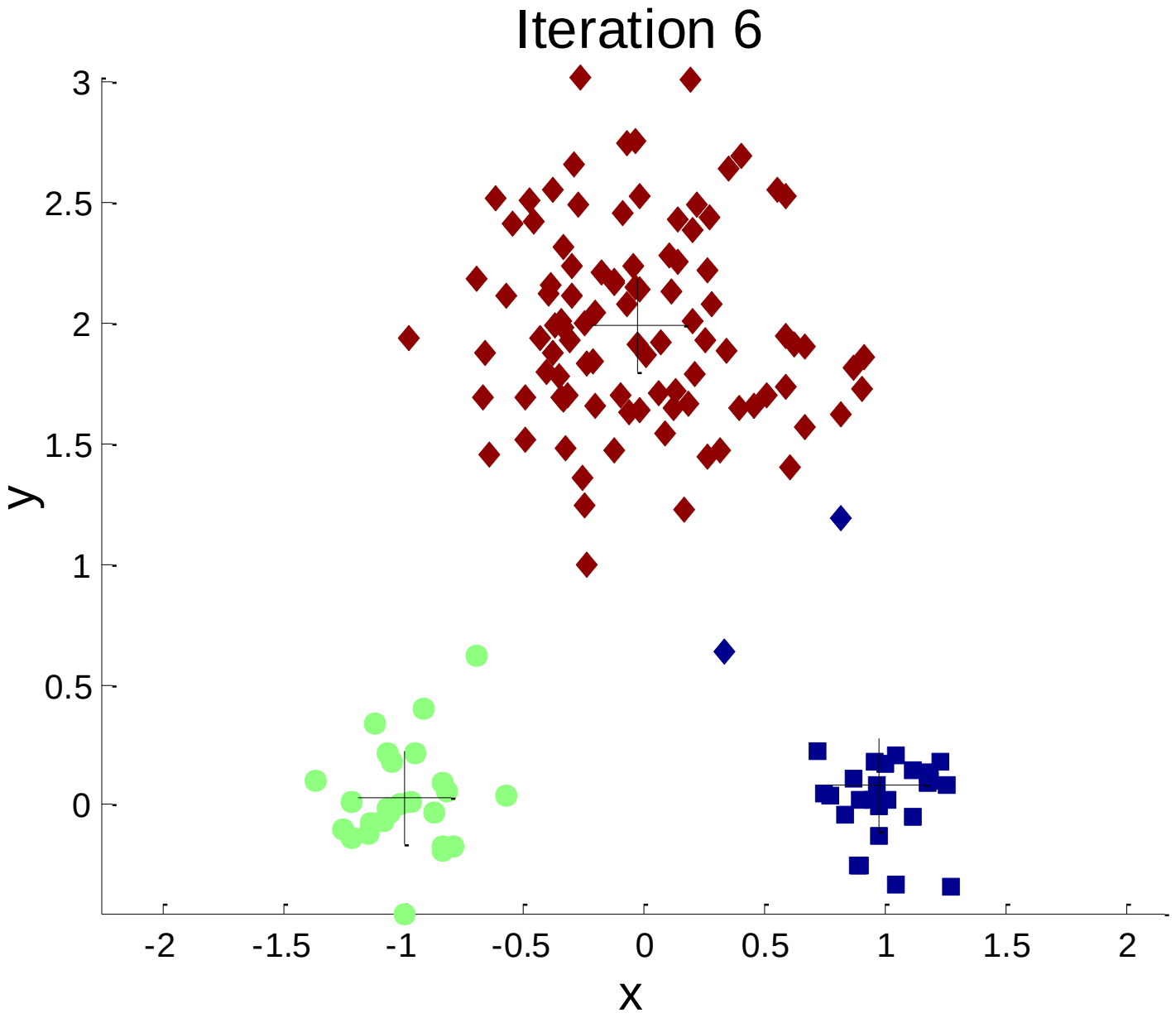
- Several strategies
  - Choose a point and assign it to the cluster
    - Choose the point that contributes most to SSE
    - Choose a point from the cluster with the highest SSE

- If there are several empty clusters, the above can be repeated several times.

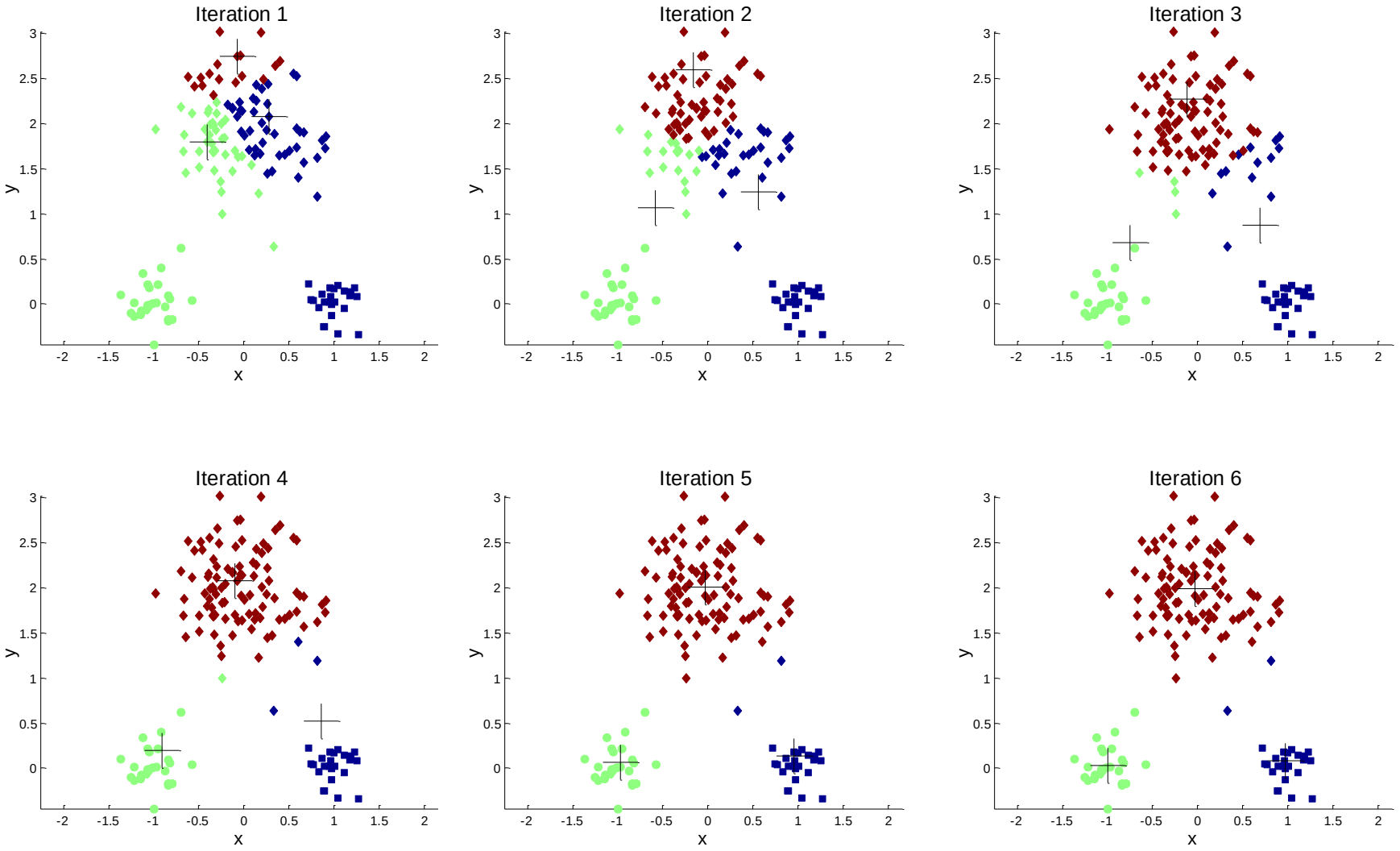
Pre-processing and Post-processing

- 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

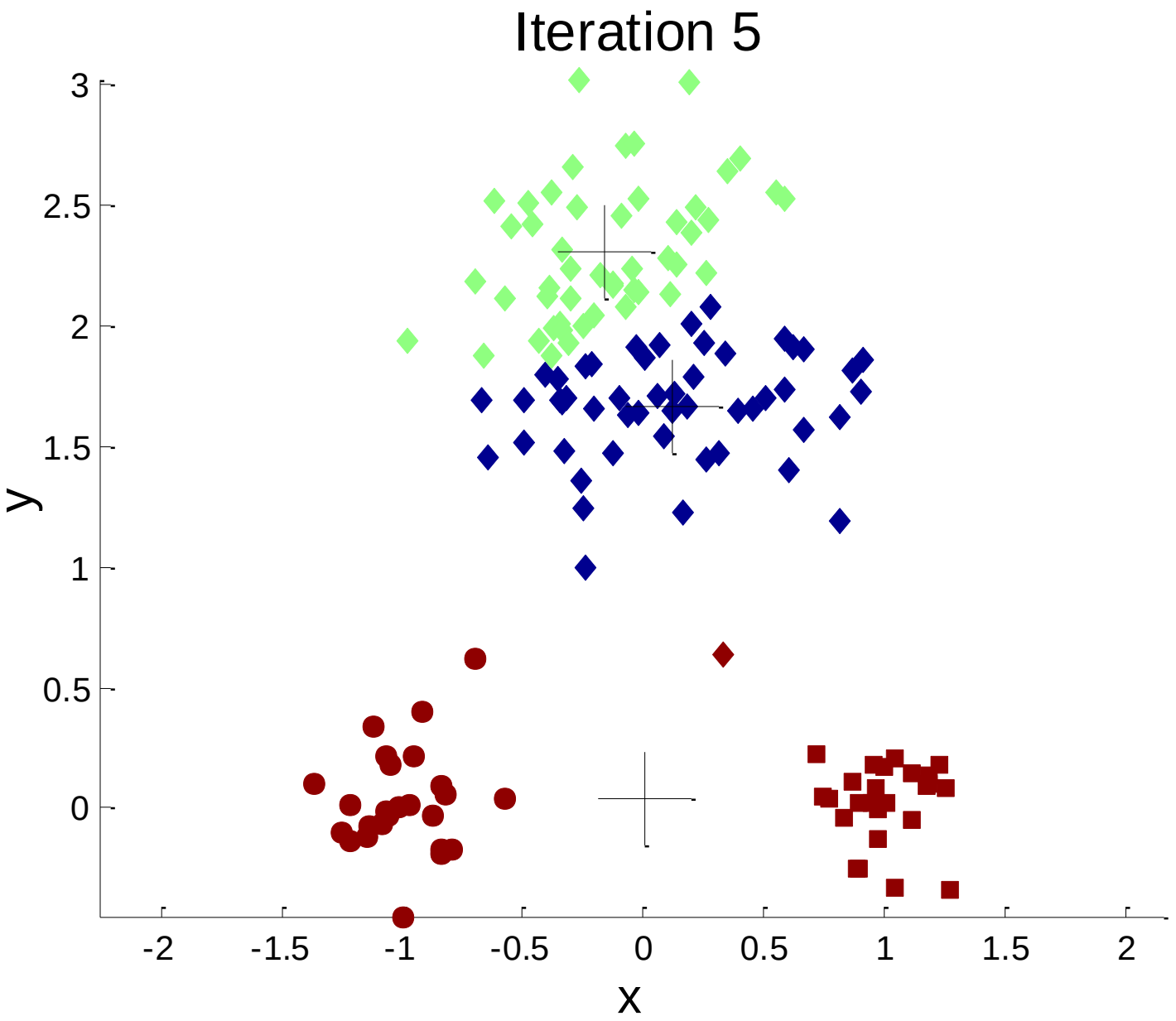
Importance of Choosing Initial Centroids



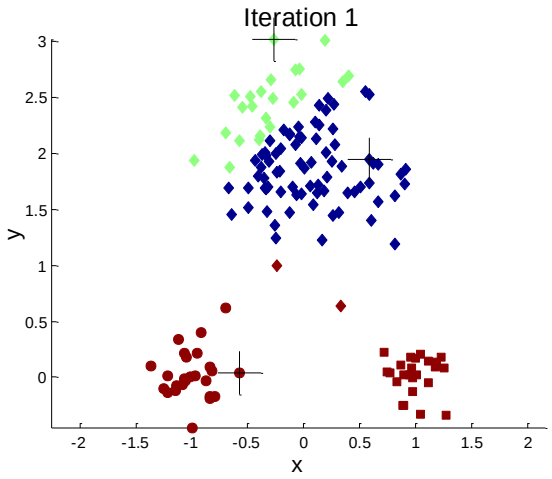
Importance of Choosing Initial Centroids

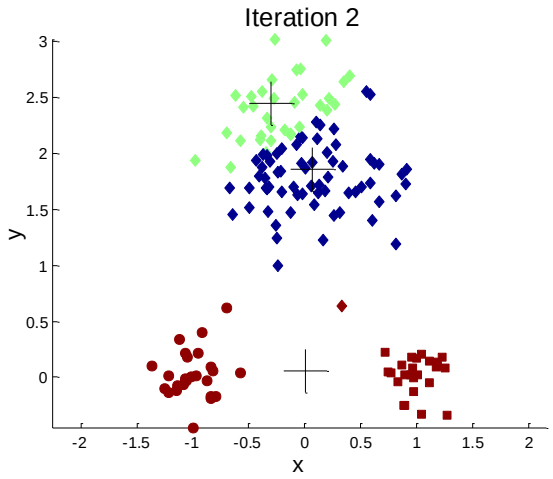


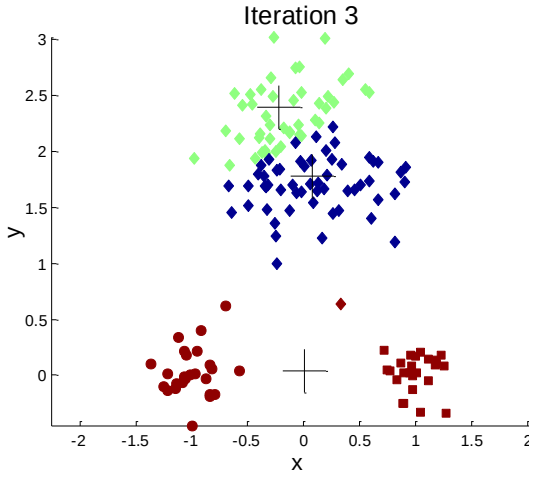
Importance of Choosing Initial Centroids ...

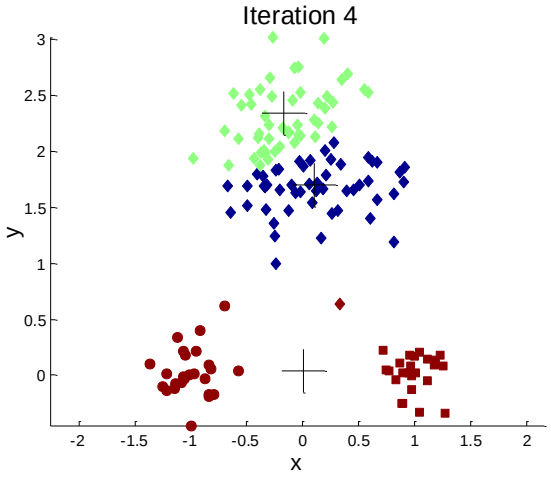


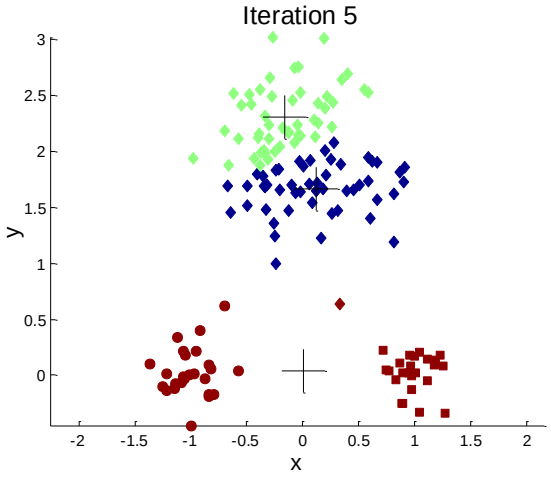
Importance of Choosing Initial Centroids …











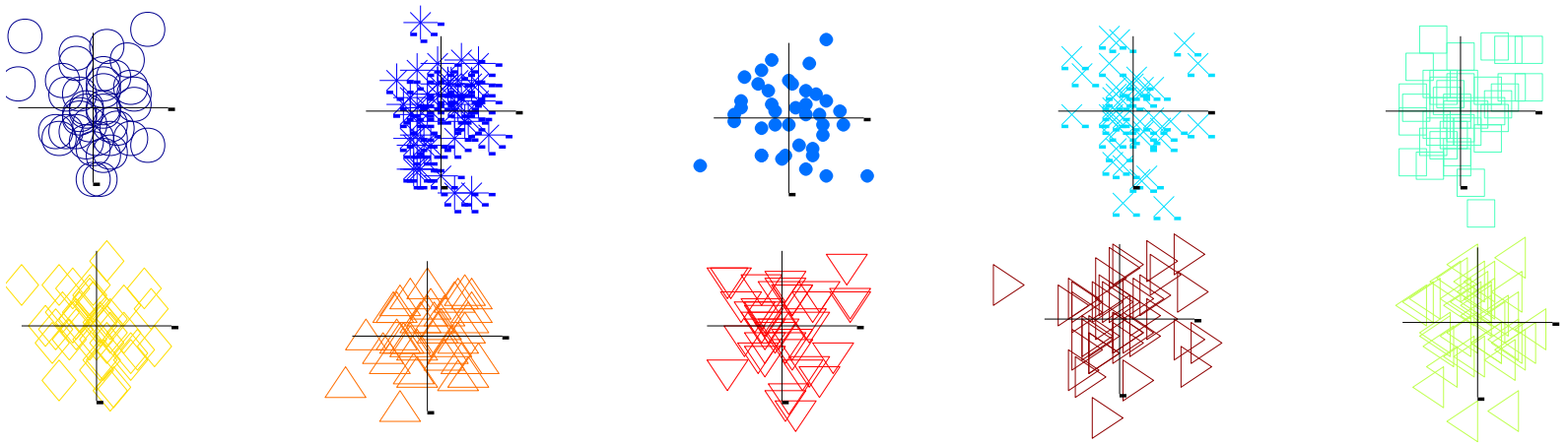
Problems with Selecting Initial Points

- If there are K 'real' clusters then the chance of selecting one centroid from each cluster is small.
  - Chance is relatively small when K is large
  - If clusters are the same size, n, then

$$P = \frac{\text{number of ways to select one centroid from each cluster}}{\text{number of ways to select } K \text{ centroids}} = \frac{K!n^K}{(Kn)^K} = \frac{K!}{K^K}$$

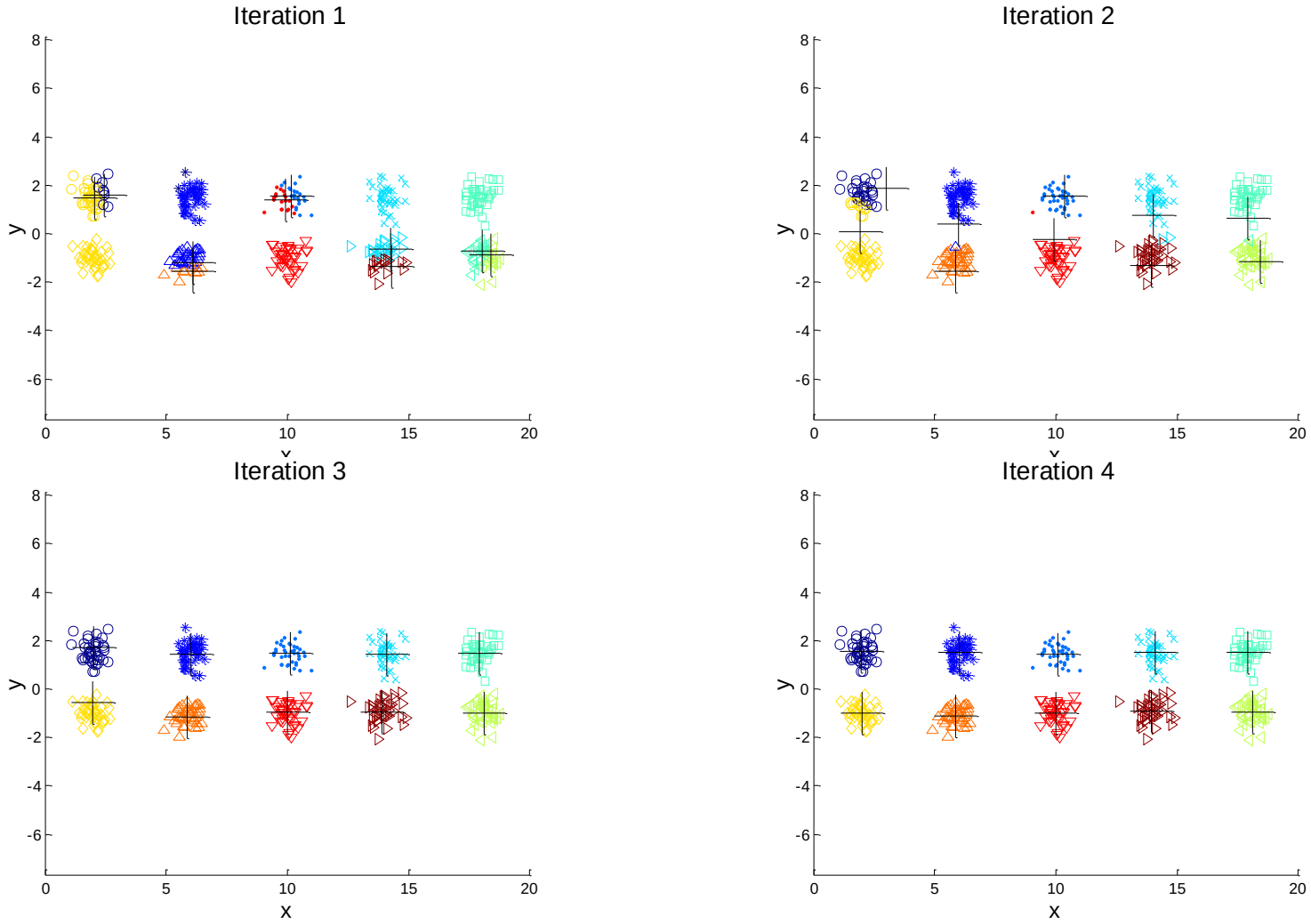
  - For example, if K = 10, then probability = $10!/10^{10} = 0.00036$
  - Sometimes the initial centroids will readjust themselves in 'right' way, and sometimes they don't
  - Consider an example of five pairs of clusters

10 Clusters Example



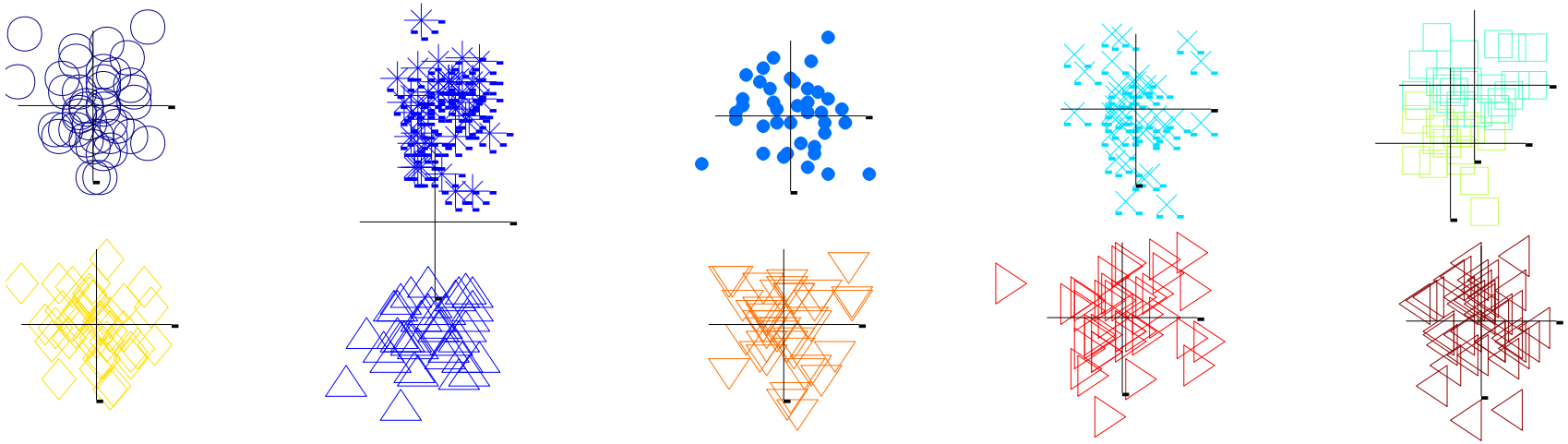
Starting with two initial centroids in one cluster of each pair of clusters

10 Clusters Example



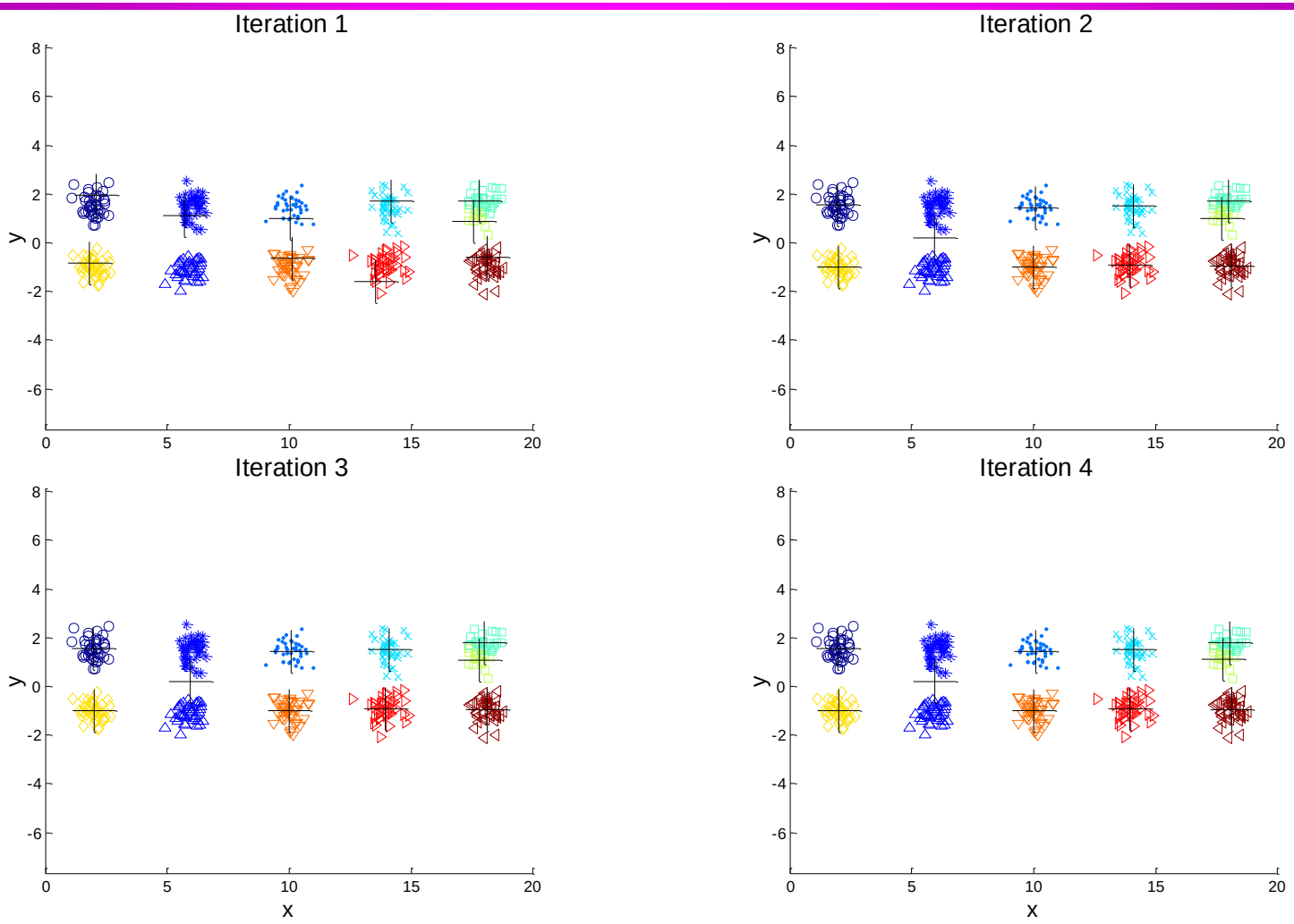
Starting with two initial centroids in one cluster of each pair of clusters

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
  - Helps, but probability is not on your side
- **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
- Generate a larger number of clusters and then perform a hierarchical clustering
- Bisecting K-means
  - Not as susceptible to initialization issues

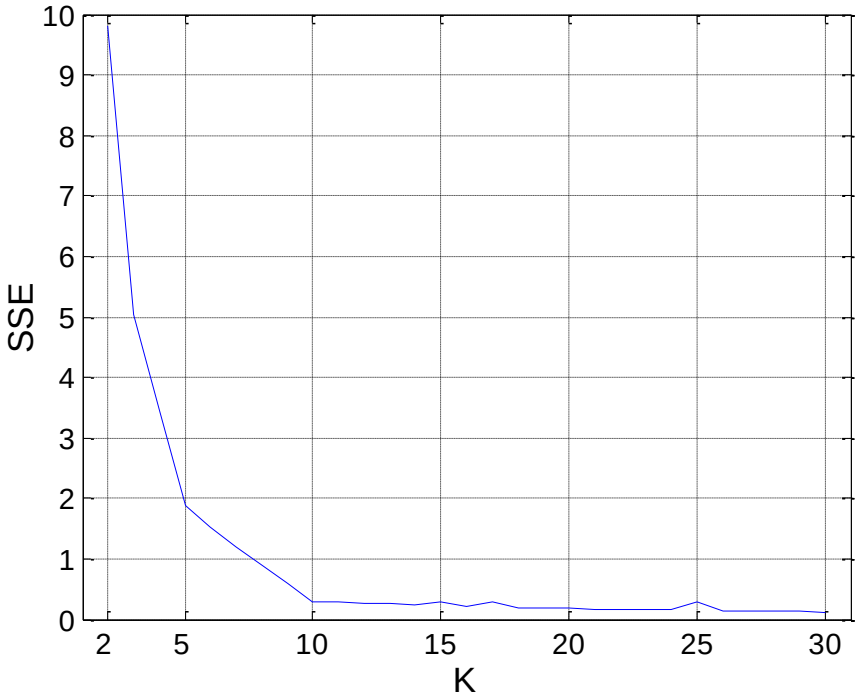
Updating Centers Incrementally

- In the basic K-means algorithm, centroids are updated after all points are assigned to a centroid

- An alternative is to update the centroids after each assignment (incremental approach)
  - Each assignment updates zero or two centroids
  - **More expensive**
  - Introduces an **order dependency**
  - Never get an empty cluster
  - Can use “weights” to change the impact

Finding the best number of clusters

- In k-means the number of clusters *K* is given
  - Partition *n* objects into predetermined number of clusters
  - **Finding the "right" number of clusters is part of the problem**



Convergence of *K*-Means

- Define goodness measure of cluster c as sum of squared distances from cluster centroid:
  - $SSE_c(c,s) = \Sigma_i (d_i - s_c)^2$ (sum over all d_i in cluster c)
  - $G(C,s) = \Sigma_c SSE_c(c,s)$
- Re-assignment monotonically decreases G
  - It is a coordinate descent algorithm (**opt one component at a time**)
- At any step we have some value for G(C,s)

  1) Fix s, optimize C → assign d to the closest centroid → G(C',s) <= G(C,s)

  2) Fix C', optimize s → take the new centroids → G(C',s') <= G(C',s) <= G(C,s)

The new cost is smaller than the original one → local minimum

Bisecting K-means

Bisecting K-means

Variant of K-means that can produce a hierarchical clustering

Algorithm 8.2 Bisecting K-means algorithm.

1: Initialize the list of clusters to contain the cluster consisting of all points.
2: **repeat**
3: Remove a cluster from the list of clusters.
4: {Perform several "trial" bisections of the chosen cluster.}
5: **for** $i = 1$ to *number of trials* **do**
6: Bisect the selected cluster using basic K-means.
7: **end for**
8: Select the two clusters from the bisection with the lowest total SSE.
9: Add these two clusters to the list of clusters.
10: **until** Until the list of clusters contains K clusters.

Bisecting K-Means

- The algorithm is exhaustive terminating at singleton clusters (unless K is known)

- Terminating at singleton clusters
  - Is time consuming
  - Singleton clusters are meaningless
  - Intermediate clusters are more likely to correspond to real classes

- No criterion for stopping bisections before singleton clusters are reached.

Combining Bisecting K-means and K-means

- The resulting clusters can be refined by using their centroids as **the initial centroids for the basic K-means.**

- Why is this necessary?
  - K-means algorithm is guaranteed to find a clustering that represents a local minimum wrt the SSE
  - Bisecting K-means uses the K- means algorithm **locally** to bisect individual clusters.
  - **The final set of clusters does not represent a clustering that is a local minimum wrt the total SSE**

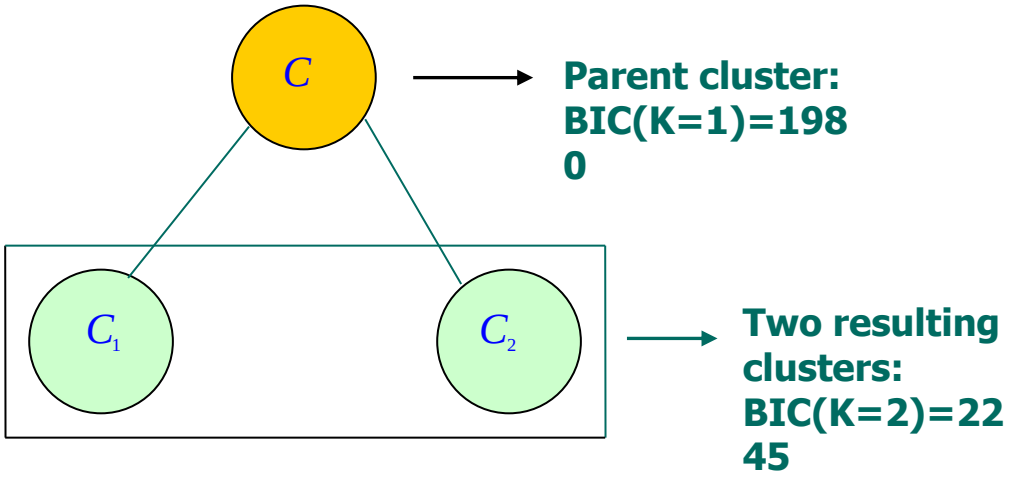
X-Means

- **X-Means** clustering algorithm is an **extended K-Means** which tries to automatically **determine the number of clusters** based on BIC scores.
- As Bisecting K-means starts with only one cluster
- The X-Means goes into action after each run of K-Means, **making local decisions** about which subset of the current centroids should split in order to better fit the data.
- The splitting decision is done by computing the **Bayesian Information Criterion** (BIC).

Bayesian Information Criterion (BIC)

- A strategy to stop the Bisecting algorithm when meaningful clusters are reached to avoid **over-splitting**
- Using BIC 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.
- Compute the BIC score of:
  - A cluster
  - Two children clusters
- BIC approximates the probability that the M_j is describing the real clusters in the data

BIC based split



The BIC score of the parent cluster is less than BIC score of the generated cluster structure => we accept the bisection.

X-Means

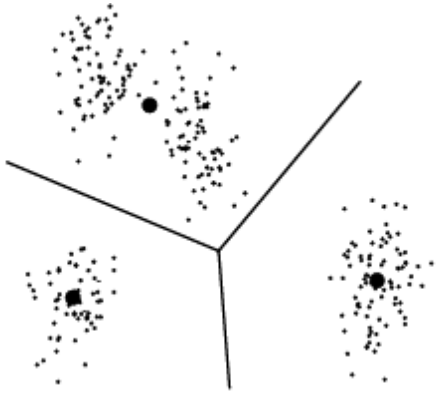
- Forward search for the appropriate value of k in a given range $[r_1, r_{max}]$:
  - Recursively split each cluster and use BIC score to decide if we should keep each split

1. Run K-means with $k=r_1$
2. Improve structure
3. If $k > r_{max}$ Stop and return the best-scoring model

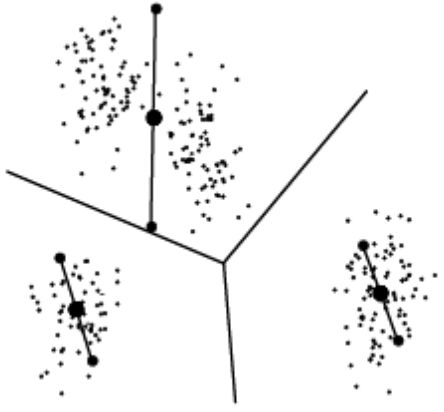
- Use local BIC score to decide on keeping a split
- Use global BIC score to decide which K to output at the end

X-Means

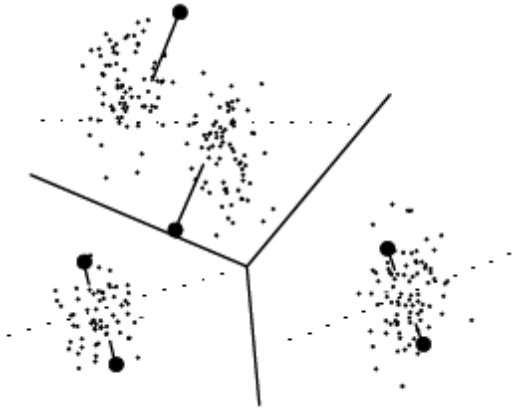
1. K-means with k=3



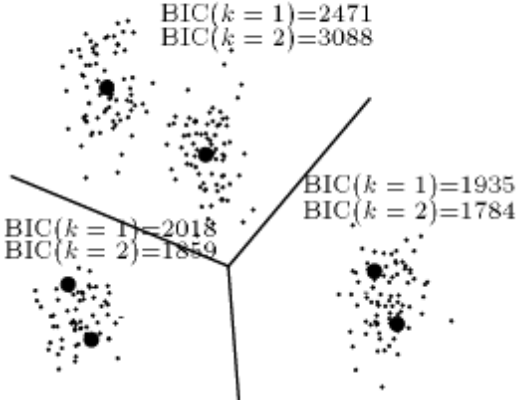
2. Split each centroid in 2 children moving a distance propotional to the region size in opposite direction (random)



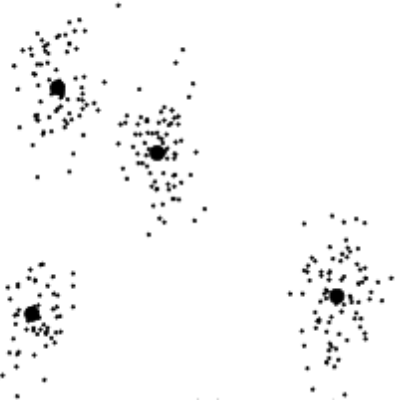
3. Run 2-means in each region locally



4. Compare BIC of parent and children



4. Only centroids with higher BIC survives



BIC Formula

- 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 data set 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

Approximate the probability that the M_j is describing the real clusters in the data

BIC (Bayesian Information Criterion)

- 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

$$P(x_i | x_i \in D_n) = \frac{1}{(2\pi\sigma^2)^{M/2}} \exp\left(-\frac{1}{2\sigma^2}\|x_i - \mu_n\|^2\right)$$

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

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

Mixture Models and the EM Algorithm

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.

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.)

EM (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 $P(G|x_i)$ and $P(C|x_i)$
  - M-Step: Compute the parameter values that (in expectation) maximize the data likelihood

E-Step: Assignment of points to clusters:
K-means: hard assignment,
EM: soft assignment

M-Step:
K-means: Computation of centroids
EM: Computation of the new model parameters

Gaussian Distribution

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

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

  - μ = mean, σ = standard deviation

Gaussian Model

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

Fitting the model

- We want to find the normal distribution 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 $X = (x_1, \ldots, x_n)$ of values
- And we want to fit a Gaussian $N(\mu, \sigma)$ model to the data
- Probability of observing 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 (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 maximize the probability $P(X|\theta)$

Maximum Likelihood Estimation (MLE)

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

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

- Maximum Likelihood Estimation
  - Find 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

MLE

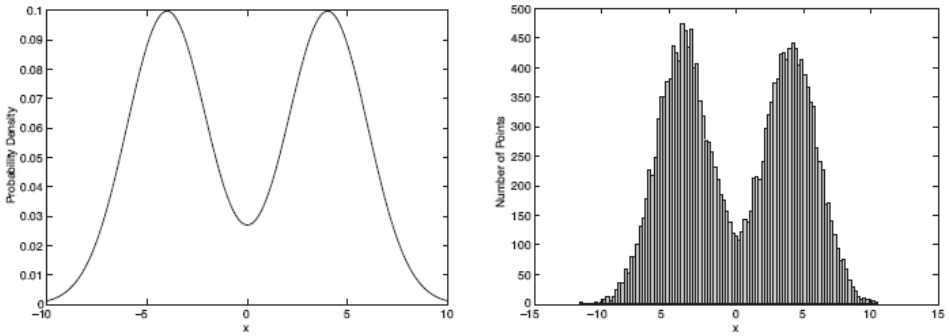
- Note: these are also the most likely parameters given the data

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

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

Mixture of Gaussians

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



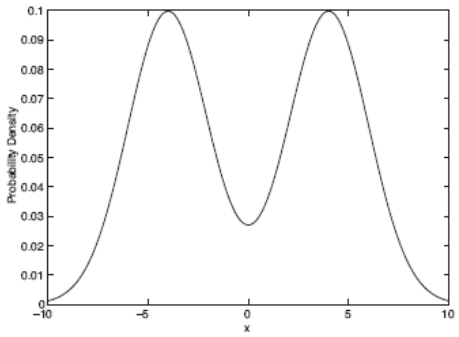
(a) Probability density function for the mixture model.

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

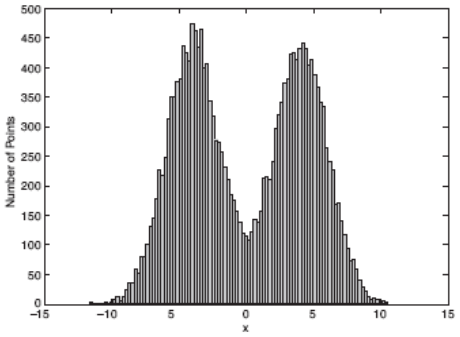
Figure 9.2. Mixture model consisting of two normal distributions with means of -4 and 4, respectively. Both distributions have a standard deviation of 2.

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 Models

- 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 $X = (x_1, \ldots, x_n)$

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

- We want to estimate the parameters that maximize the Likelihood of the data

Mixture Models

- Once we have the parameters $\Theta = (\pi_G, \pi_C, \mu_G, \mu_C, \sigma_G, \sigma_C)$ 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)

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

EM (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 $P(G|x_i)$ and $P(C|x_i)$
  - M-Step: Compute the parameter values 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

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

Advantages & Disadvantages

- Disadvantages of EM:
  - It can be slow thus it's not suitable fot large dimensionality
  - It does not work in case of few data points
  - It has difficulty in case of noise and outliers

- Advantages of EM:
  - More geneal wrt K-means because it can use different types f distributions
  - It can find cluster with different size and shape