K-means

AW

Lecture Overview

1. **Recap**
2. **K-means Algorithm**
3. **K-means Optimization**
4. **Initial Centroids**
5. **K-medoids Algorithm**

Basic Clustering Model

Input: a set of elements, X, supplied with one of the following:

- A distance function $d: X \times X \rightarrow \mathbb{R}^{\geq 0}$, i.e. non-negative function that is symmetric, satisfies identity (i.e. $d(x, y) \ = \ 0 \Leftrightarrow x \equiv y$ for all $x \in X$) and satisfies the triangle inequality, i.e. $d(x, y) \leq d(x, z) + d(z, y)$.
- A similarity function $\mathrm{s}: X \times X \rightarrow [0,1]$ that is symmetric and satisfies $s(x, x) = 1$

Optional input: some clustering algorithms expect the number of required clusters).

Output: a partition of the domain set X into subsets, i.e. $C \ = (C_1, \ldots, C_k)$ such that $X = \bigcup_{i=1}^{k} C_i$ and for all $i \neq j$, $C_i \cap C_j = \emptyset$.

Model Modifications

Basic Model: Partitional Clustering:

- Output: A partitioning of data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset

Model Modification: Hierarchical clustering:

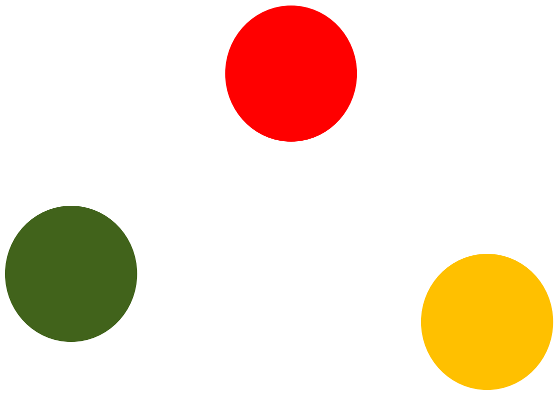
- Output: A dendrogram of clusters, i.e. a set of nested clusters organized as a hierarchical tree of domain subsets, having the singleton sets in its leaves, and the full domain as its root. Each ‘slice’ of a dendrogram is a partitional clustering

More Model Modifications

- Basic model: partitioning is based on distance similarity function:
    - Well-separated clusters
    - Center-based clusters (or medoid based)
    - Contiguous clusters
- Modified Models:
    - Density-based clusters
    - Property or Conceptual
    - Described by an Objective Function

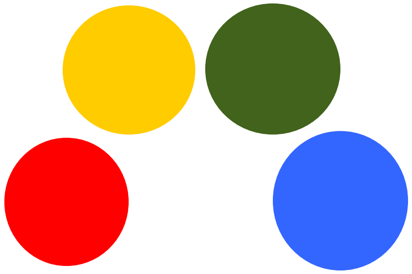
Well-separated vs Center-based Clusters

- Well-Separated Clusters:
  - a point belong to a cluster iff its distance (similarity) to ALL points in a cluster is smaller (bigger) than to any other data point



3 well-separated clusters

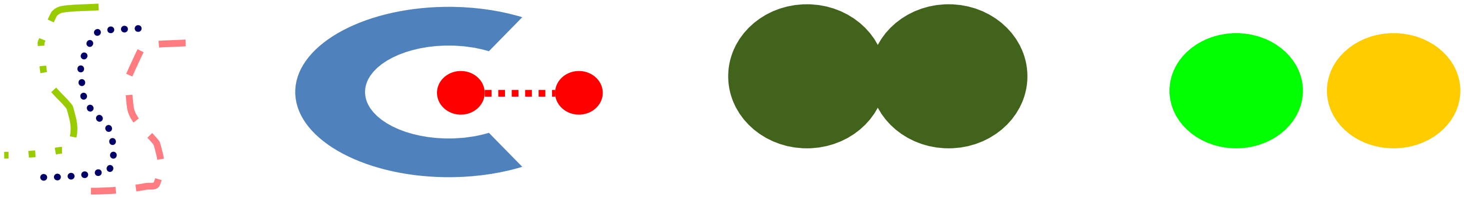
- Center-based
  - A cluster is a set of data points such that a data point in a cluster is closer (more similar) to the "leader" of a cluster, than to a leader of any other cluster
  - The leader of a cluster is often a centroid, the average of all the points in the cluster (may be a point of a domain space that is not a data point), or a medoid, the most "representative" data point of a cluster



4 center-based clusters

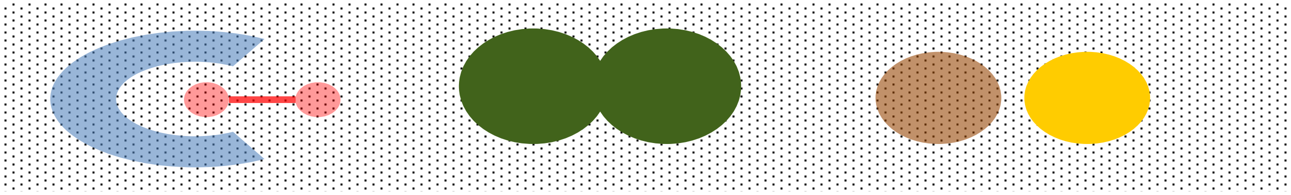
Contiguity-based vs Density-based Clusters

- Contiguous Cluster (Nearest neighbor or Transitive)
  - A cluster is a set of points such that a point in a cluster is closer (or more similar) to one or more other points in the cluster than to any point not in the cluster.



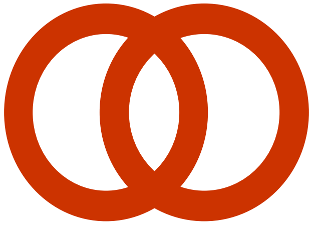
8 contiguous clusters

- Density-based
  - A cluster is a dense region of points, which is separated by low-density regions, from other regions of high density.
  - Used when the clusters are irregular or intertwined, and when noise and outliers are present.



6 density-based clusters

Pattern-based and Defined by a Goal



2 Overlapping Circles

- Shared Property or Conceptual Clusters
  - Finds clusters that share some common property or represent a particular concept.
- Clusters Defined by an Objective Function
  - Clusters defined as a solution to optimization problem (e.g. maximize total between cluster distance).
  - Always have brute force solution: enumerate all possible ways of dividing the points into clusters and evaluate the `goodness' of each potential set of clusters by using the given objective function. (NP Hard)
  - Can have global or local objectives.
    - Hierarchical clustering typically have local objective functions (i.e. global objective is satisfied when some local objective is)
    - Partition-inducing algorithms typically have global objectives (greedy algorithms can only be approximate

.

Clusters by Objective Function

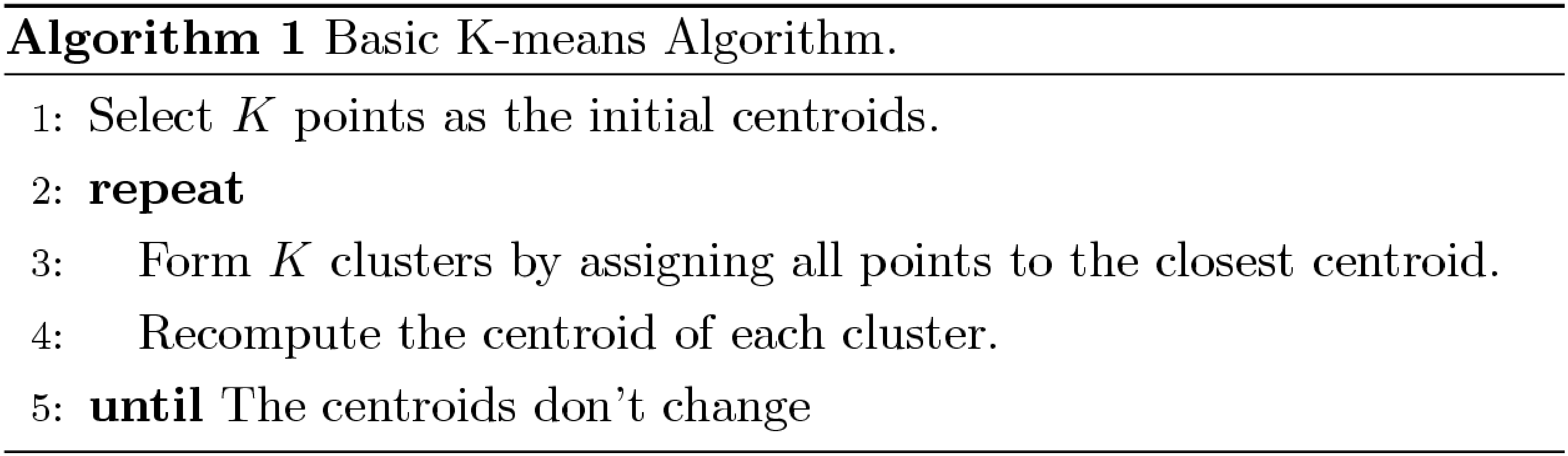
- Clusters Defined by an Objective Function
  - Clusters defined as a solution to optimization problem (e.g. maximize total between cluster distance).
  - Always have brute force solution: enumerate all possible ways of dividing the points into clusters and evaluate the `goodness' of each potential set of clusters by using the given objective function. (NP Hard)
  - Can have global or local objectives.
    - Hierarchical clustering typically have local objective functions (i.e. global objective is satisfied when some local objective is)
    - Partition-inducing algorithms typically have global objectives (greedy algorithms can only be approximate)

Lecture Overview

1. Recap
2. **K-means Algorithm**
3. K-means Optimization
4. Initial Centroids
5. K-medoids Algorithm

K-means Clustering

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



Algorithm 1 Basic K-means Algorithm.

```
Select $K$ points as the initial centroids.
repeat
    Form $K$ clusters by assigning all points to the closest centroid.
    Recompute the centroid of each cluster.
until The centroids don't change
```

K-means Clustering – Details

- Initial centroids are often chosen randomly.
  - Clusters produced vary from one run to another.
- The centroid may be the concept point of the domain that is computed from the points in the cluster or one of the points in the cluster that is chosen in a specified way e.g. for $\mathbb{R}^n$ centroid can be
  - the mean of the computed cluster
  - the point that is the center of the ball of minimum radius that contains all points of the cluster .
- Tightness of the cluster is measured w.r.t. to some measure, e.g.
  - Euclidean distance. Then tightness is sum of Euclidean distances of all points in the cluster to its center,
  - Cosine similarity. Then tightness can be average cosine similarity of points in a cluster to a centroid, etc.
- K-means converges for common distance/similarity measures

Example: 2-means, $X \subset \mathbb{R}$, centroid=mean

1. 1-dim dataset: {2, 3, 4, 10, 11, 12, 20, 25, 30}, suppose centers generated at random are c_1=2,c_2=4. What are the class groupings?

Example: 2-means, $X \subset \mathbb{R}$, centroid=mean

1. 1-dim dataset: {2, 3, 4, 10, 11, 12, 20, 25, 30}, suppose centers generated at random are c_1=2,c_2=4. What are the class groupings?

   C_1 = {2, 3} C_2 = {4, 10, 11, 12, 20, 25, 30}. What are new centers?

Example: 2-means, $X \subset \mathbb{R}$, centroid=mean

1. 1-dim dataset: {2, 3, 4, 10, 11, 12, 20, 25, 30}, suppose centers generated at random are c_1=2,c_2=4. What are the class groupings?

   C_1 = {2, 3} C_2 = {4, 10, 11, 12, 20, 25, 30}. What are new centers?
2. c_1=2.5, c_2=112/7=16. What are the class groupings?

Example: 2-means, $X \subset \mathbb{R}$, centroid=mean

1. 1-dim dataset: {2, 3, 4, 10, 11, 12, 20, 25, 30}, suppose centers generated at random are c_1=2,c_2=4. What are the class groupings?

   C_1 = {2, 3} C_2 = {4, 10, 11, 12, 20, 25, 30}. What are new centers?
2. c_1=2.5, c_2=112/7=16. What are the class groupings?

   C_1 = {2, 3, 4} C_2 = {10, 11, 12, 20, 25, 30}. What are new centers?

Example: 2-means, $X \subset \mathbb{R}$, centroid=mean

1. 1-dim dataset: {2, 3, 4, 10, 11, 12, 20, 25, 30}, suppose centers generated at random are c_1=2,c_2=4. What are the class groupings?

   C_1 = {2, 3} C_2 = {4, 10, 11, 12, 20, 25, 30}. What are new centers?
2. c_1=2.5, c_2=112/7=16. What are the class groupings?

   C_1 = {2, 3, 4} C_2 = {10, 11, 12, 20, 25, 30}. What are new centers?
3. c_1=3, c_2=108/6=18. What are the class groupings?

ExampExample: 2-means, $X \subset \mathbb{R}$, centroid=mean

1. 1-dim dataset: {2, 3, 4, 10, 11, 12, 20, 25, 30}, suppose centers generated at random are c_1=2,c_2=4. What are the class groupings?

   C_1 = {2, 3} C_2 = {4, 10, 11, 12, 20, 25, 30}. What are new centers?
2. c_1=2.5, c_2=112/7=16. What are the class groupings?

   C_1 = {2, 3, 4} C_2 = {10, 11, 12, 20, 25, 30}. What are new centers?
3. c_1=3, c_2=108/6=18. What are the class groupings?

   C_1 = {2, 3, 4,10} C_2 = {11, 12, 20, 25, 30}. What are new centers?
4. c_1=4.75 c_2=19.6. What are the class groupings?

   C_1 = {2, 3, 4,10,11,12} C_2 = {20, 25, 30}. What are new centers?
5. c_1=7 c_2=25. What are the class groupings?

   C_1 = {2, 3, 4,10,11,12} C_2 = {20, 25, 30} - converged

K-means Convergence in $\mathbb{R}^n$

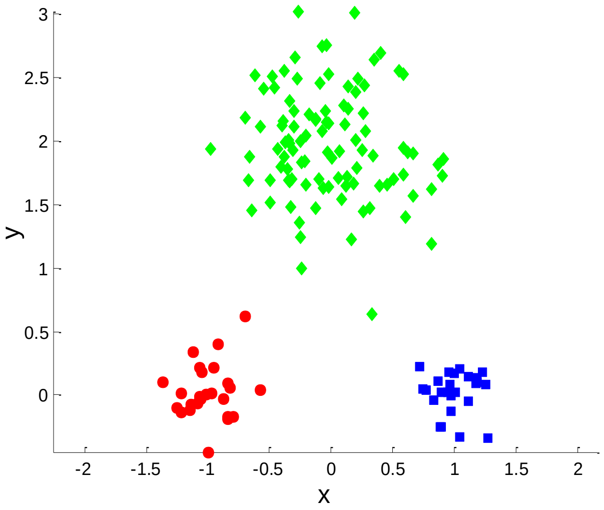
Theorem. *For any dataset X and any number of clusters k, the K-means algorithm with centroids=means converges in a finite number of iterations, where convergence is measured by $\mathcal{L}(k, X) = \sum_{i=1}^{k} \sum_{\bar{x} \in C_i} \left\| \bar{x} - \bar{c}^i \right\|^2$ ceasing the change.*

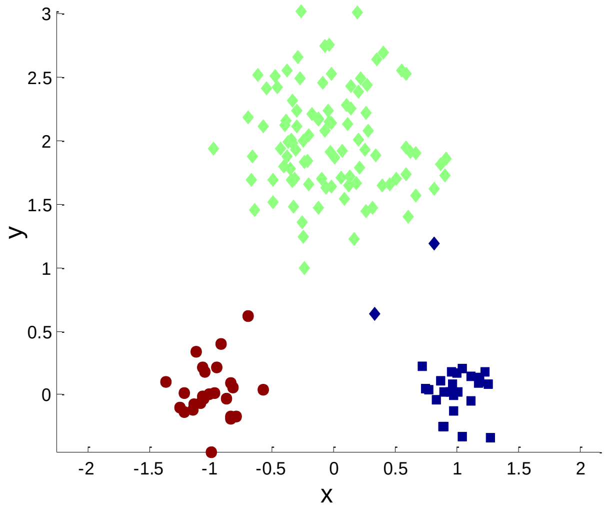
Proof.

- k target clusters are among $2^{|X|}$ subsets of X, so cluster centers are among means of $2^{|X|}$ subsets = finitely many possible label assignments to each $\bar{x}$ and as many candidate points centroids. Also $\mathcal{L}(k, X) \geq 0$ thus $\mathcal{L}$ ca only decrease finite number of times.
- Cluster assignment i for point $\overline{x}$ can change in line 3. Value of $\bar{c}^i$ can change in line 4. Both lines can only decrease $\mathcal{L}(k, X)$:
  - In 3 suppose assignment was $i \in 2^{|X|}$ and it became $j \in 2^x$ but then $\left\| \bar{x} - \mu_j \right\|^2 \leq \| \bar{x} - \mu_i \|^2$
  - Line 4 computes μ_i as the mean of the points $\overline{x}$ which are labeled i = k, which minimizes squared distances (will prove later)

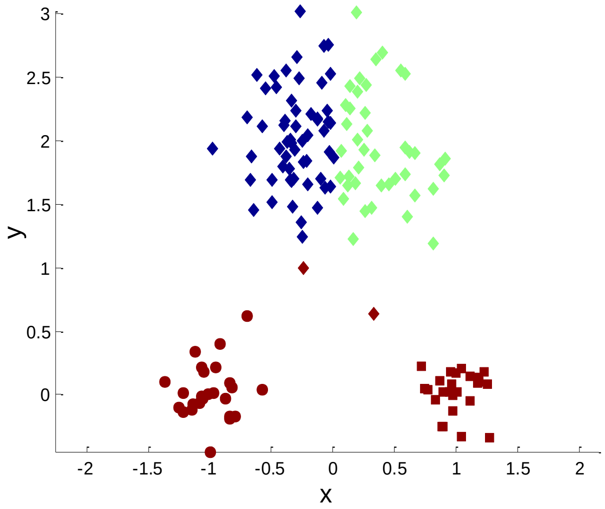
Two different K-means Clusterings

Original Points



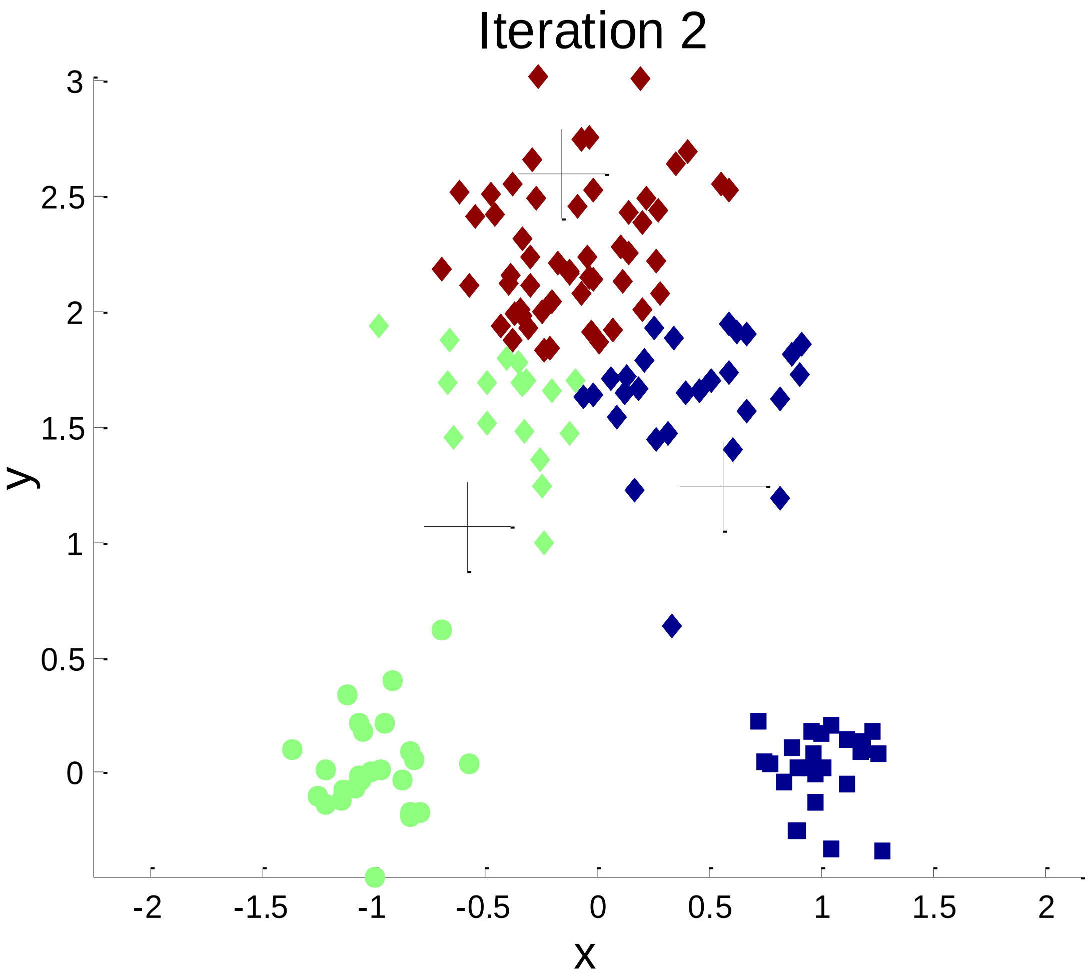


Optimal Clustering

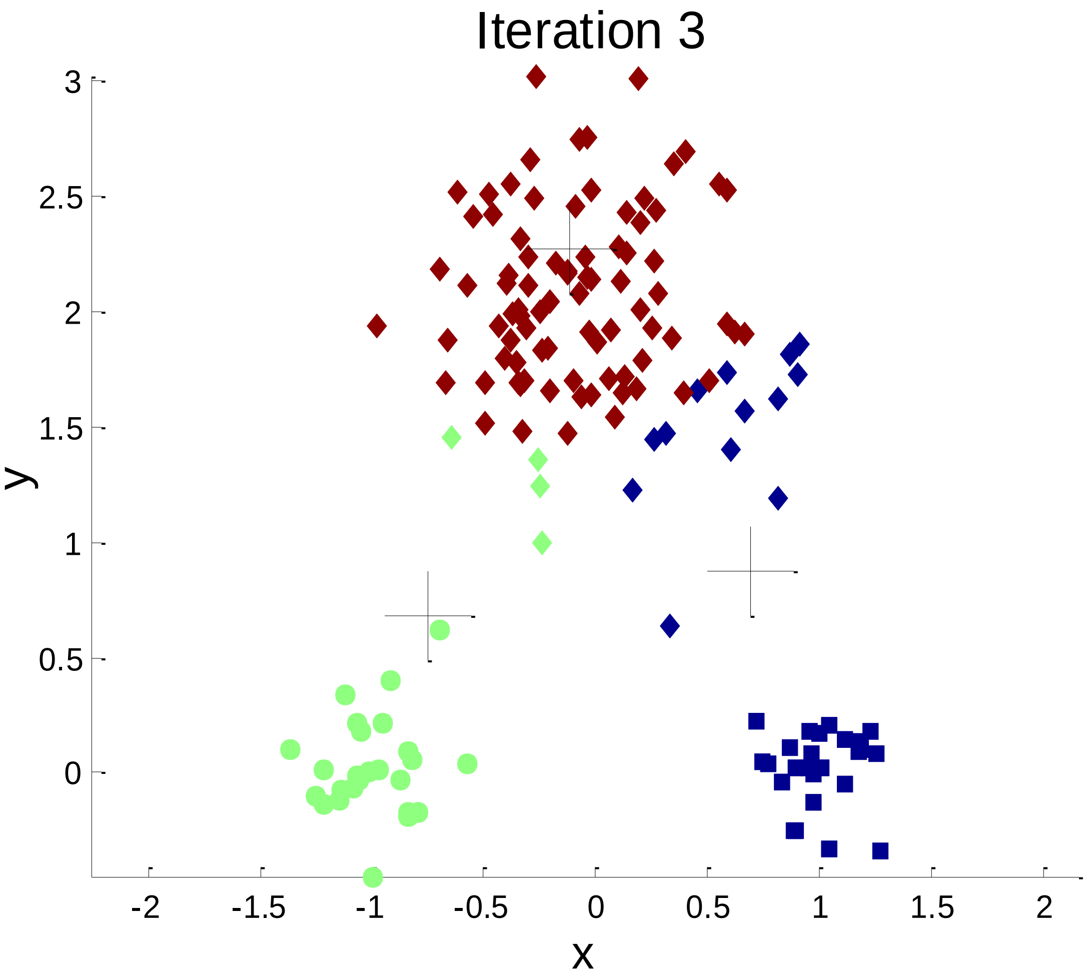


Sub-optimal Clustering

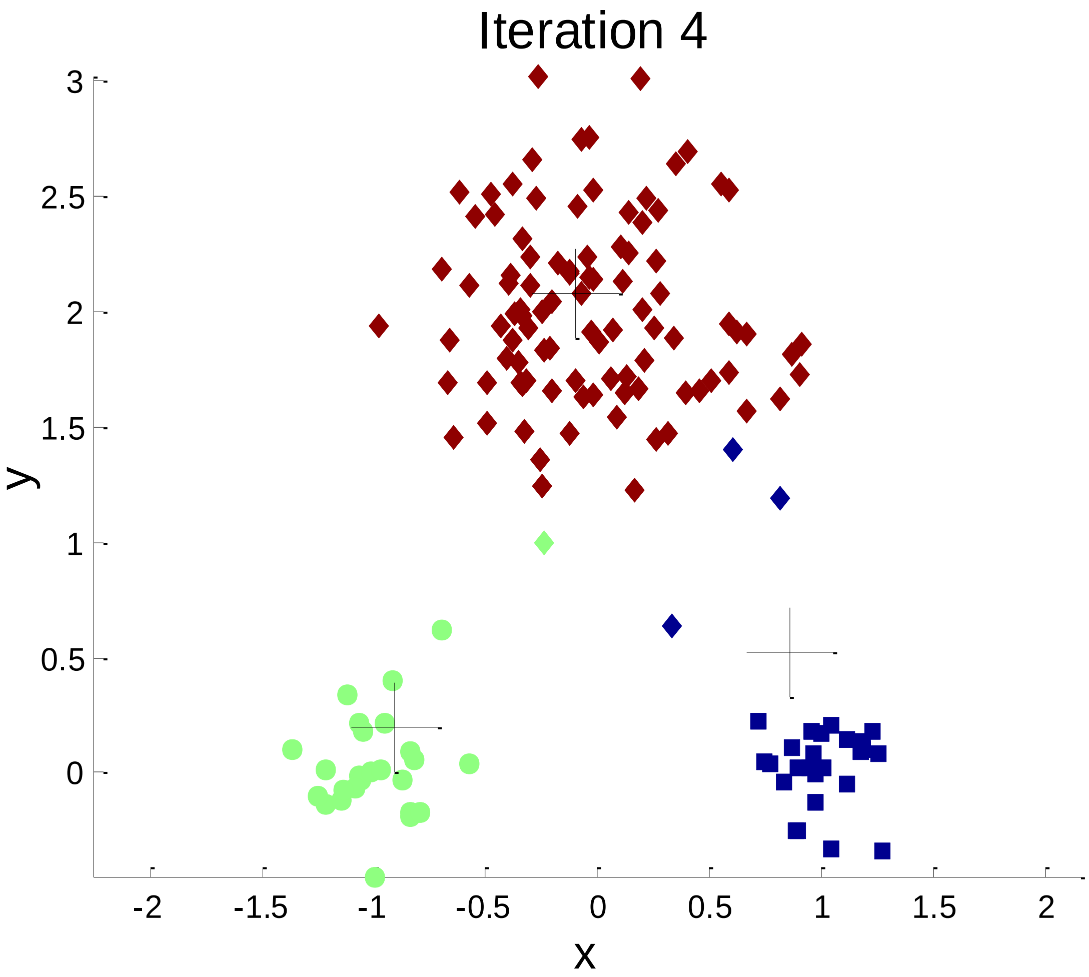
Importance of Choosing Initial Centroids



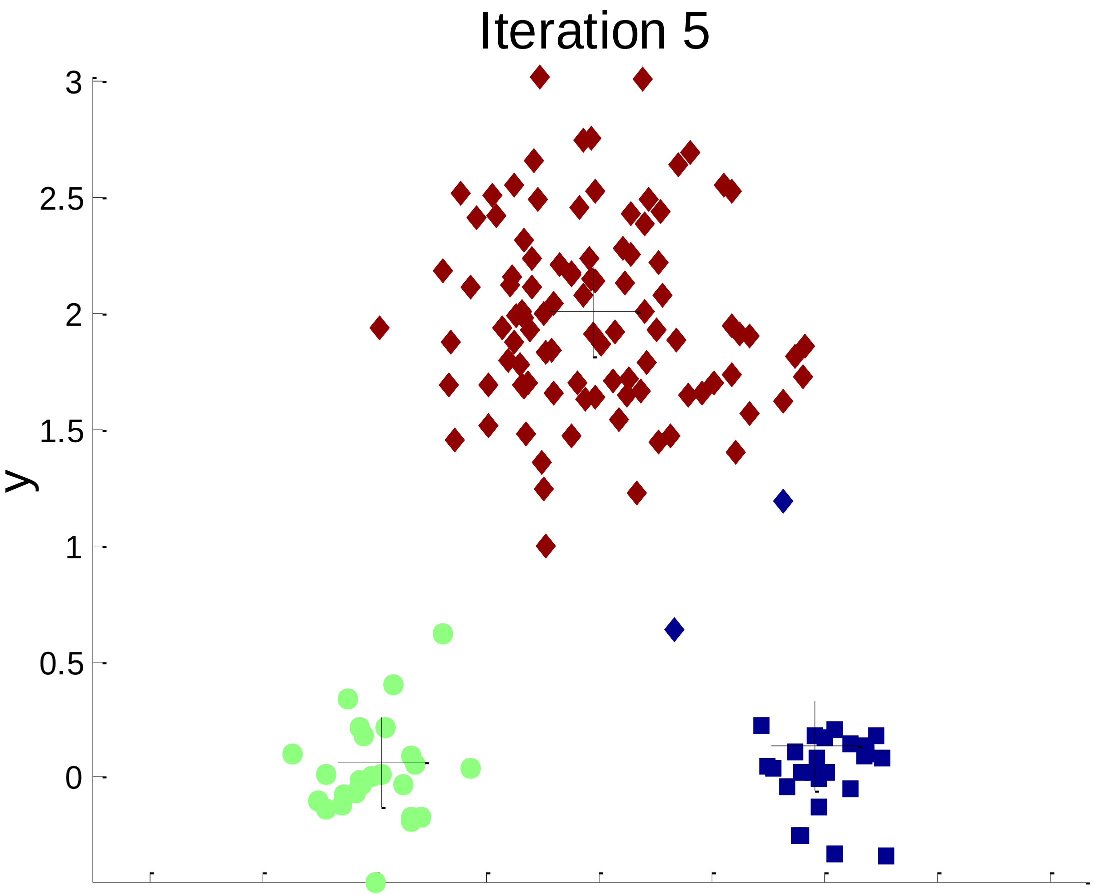
Importance of Choosing Initial Centroids



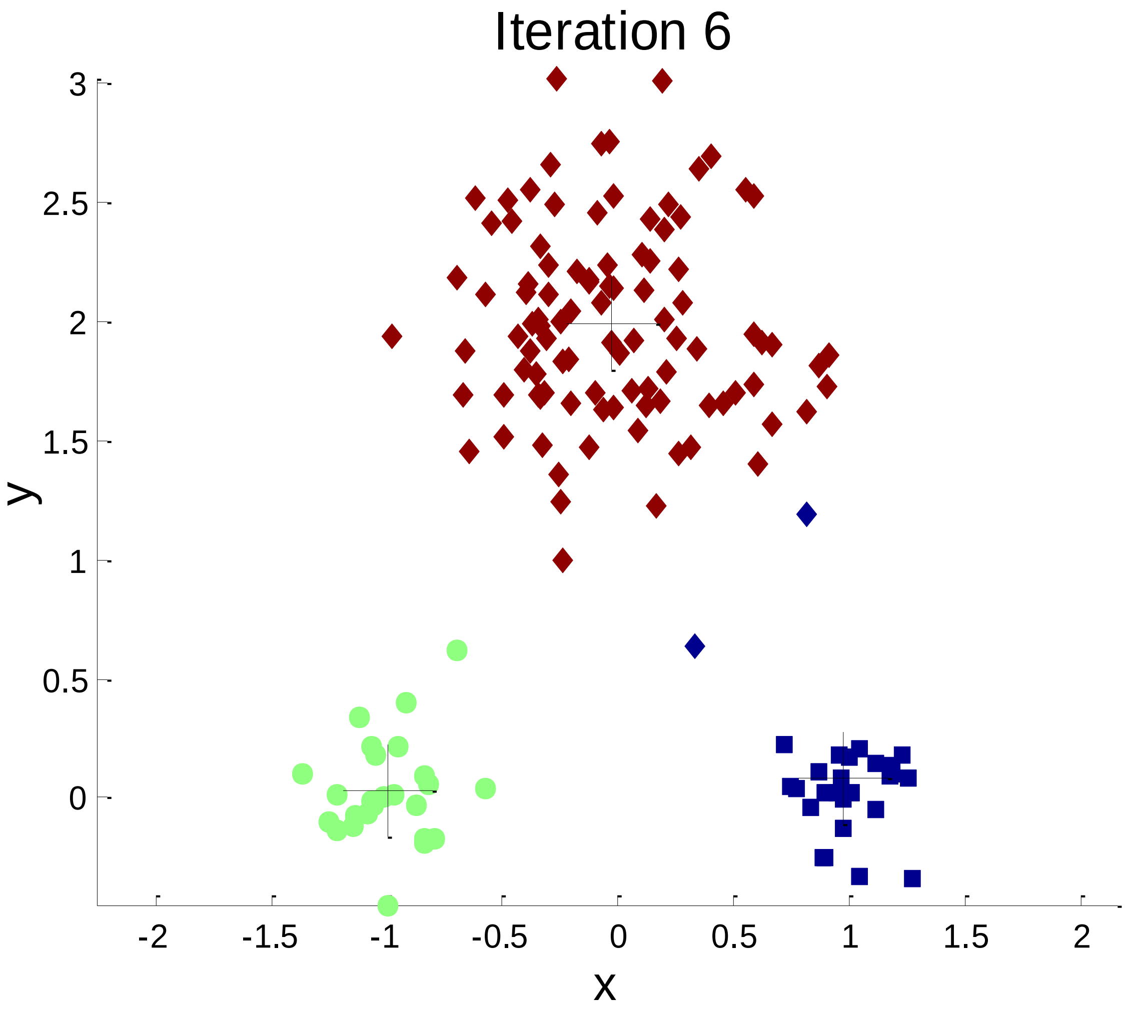
Importance of Choosing Initial Centroids



Importance of Choosing Initial Centroids



Importance of Choosing Initial Centroids



Lecture Overview

1. Recap

2. K-means Algorithm

3. **K-means Optimization**

4. Initial Centroids

5. K-medoids Algorithm

K-means as Iterative Optimization

- Standard measure in real n-dimensional space is Euclidean distance. Let’s measure the tightness (quality) of clustering by Sum of Squared Errors:

$$SSE = \sum_{i=1}^{k} \sum\nolimits_{\bar{x} \in C_i} \left\| \bar{x} - \bar{c}^i \right\|^2$$

- The optimization Problem: find $\{C_1, \dots, C_k\}$ minimizing SSE

K-means is iterative solution

- Initialization: pick cluster centers. Then
  - For $\bar{x}$ the error is the distance to nearest cluster center $\bar{c}^i$. In K-means given a point $\bar{x}$ and k clusters, we choose to associate $\bar{x}$ with a cluster center which minimizes the SSE measure (objective function SSE). Thus we choose the closest cluster center
- Iteration step:
  - Re-computation of centroids $\bar{c}^i$ is the task of fining solution minimum SSE problem
  - We show that in $\mathbb{R}^n$ center is the mean of the cluster

K-means Recurrent Objective

- Given objective function $f(x_1, \dots, x_m, k)$ optimize f;
- Examples for data in $\mathbb{R}^n$
  - $f = MSE = \min \sum_{i=1}^{k} \sum_{\bar{x} \in C_i} \left\| \bar{x} - \bar{c}^i \right\|^2$ where C_i are clusters and $\bar{c}^i$are respective cluster centers
  - $f = \min SAE = \min \sum_{i=1}^{k} \sum_{\bar{x} \in C_i} \sum_{j=1}^{n} \left| x_j - c_j^i \right|$
  - $f = \min SMD = \min \sum_{i=1}^{k} \sum_{\bar{x} \in C_i} \left(\bar{x} - \bar{c}^i \right) \Sigma^{-1} (\bar{x} - \bar{c}^i)^T$ where Σ^{-1} is the inverse of the covariance matrix of the data points.

    $\left(\bar{x} - \bar{c}^i \right) \Sigma^{-1} (\bar{x} - \bar{c}^i)^T$ is called Mahalnobis distance between $\bar{x}$ and $\bar{c}^i$. Sum of Mahallanobis distances is computationally expensive
- Examples for text: $Coh = \min \sum_{i=1}^{k} \sum_{\bar{x} \in C_i} csine\left(\bar{x}, \bar{c}^i \right)$ minimum total cohesion

Cluster Means minimize SSE!

Why do we choose means as centroids in Euclidean k-Means? No one said it minimizes SSE

Claim. For MSE objective function optimum centroids are means of clusters.

Proof. To minimize SSE we need $\frac{\partial(SSE)}{\partial \overline{c}^i} = 0$ for each centroid $\boldsymbol{c}_i$ so

$$\frac{\partial}{\partial \overline{c}^i}\left(\sum_{i=1}^{k}\sum_{\overline{x}\in C_i}\|\overline{x} - \overline{c}^i\|^2\right) = \sum_{i=1}^{k}\sum_{\overline{x}\in C_i}\frac{\partial}{\partial \overline{c}^i}\left(\|\overline{x} - \overline{c}^i\|^2\right)$$

$$= \sum_{\overline{x}\in C_j} 2\|\overline{x} - \overline{c}^i\| = 0 \Rightarrow \left\|\sum_{x\in C_j}(\overline{x} - \overline{c}^i)\right\| = 0 \Rightarrow$$

$$\left\|\left(\sum_{\overline{x}\in C_j}\overline{\boldsymbol{x}}\right) - |C_j|\overline{c}^j\right\| = 0 \Rightarrow \left(\sum_{\overline{x}\in C_j}\overline{x}\right) - |C_j|\overline{c}^j = \overline{0} \Rightarrow$$

$$\overline{c}^j = \frac{1}{|C_j|}\sum_{\overline{x}\in C_j}\overline{x}\text{–}\mathit{mean\ of\ the\ cluster}$$

K-means Optimization – Solutions

Objective function	Proximity measure	Centroid
min Sum of Absolute Errors (SAE)	Manhattan distance	median
min Sum of Squared Errors (SSE)	Euclidean distance	Mean
min Total Cohesion	Cosine similarity	Mean
Min Sum of Mahalanobis Distances (SMD)	Mahalanobis distance (centered)	Mean

K-means Optimization

- The algorithm pic the optimal solution at every step. Is it possible that it’ll end up returning non-optimal clusters?

K-means Optimization

- The algorithm pic the optimal solution at every step. Is it possible that it’ll end up returning non-optimal clusters?
- Of course! This is the local search algorithm, even though it optimally chooses centroids at every iterative step
- The result depends on the starting points
  - It may return global minimum
  - It may return local minimum rather than global

Lecture Overview

1. Recap

2. K-means Algorithm

3. K-means Optimization

4. **Initial Centroids**

5. K-medoids Algorithm

K-means: Choosing Initial Centroids …

Typical procedure have number of clusters k as input

1. It chooses centroids at random
2. If SSE is less than threshold T then it accepts clustering
3. Otherwise it repeats steps 1 and 2 a given number N of times
4. If no clustering is found number of clusters k incremented
5. It repeat steps 1-4 either until the clustering is found or until max number of times.
6. If no clustering with SSE below threshold is found then the procedure reports failure

Problems with Selecting Initial Points

- If there are K 'real' clusters each of size n then the chance of selecting one centroid from each cluster is small:

$$\text{P}=\frac{\text{number of ways to select one centroid from each cluster}}{\text{number of ways to select K centroids}}=?$$

Problems with Selecting Initial Points

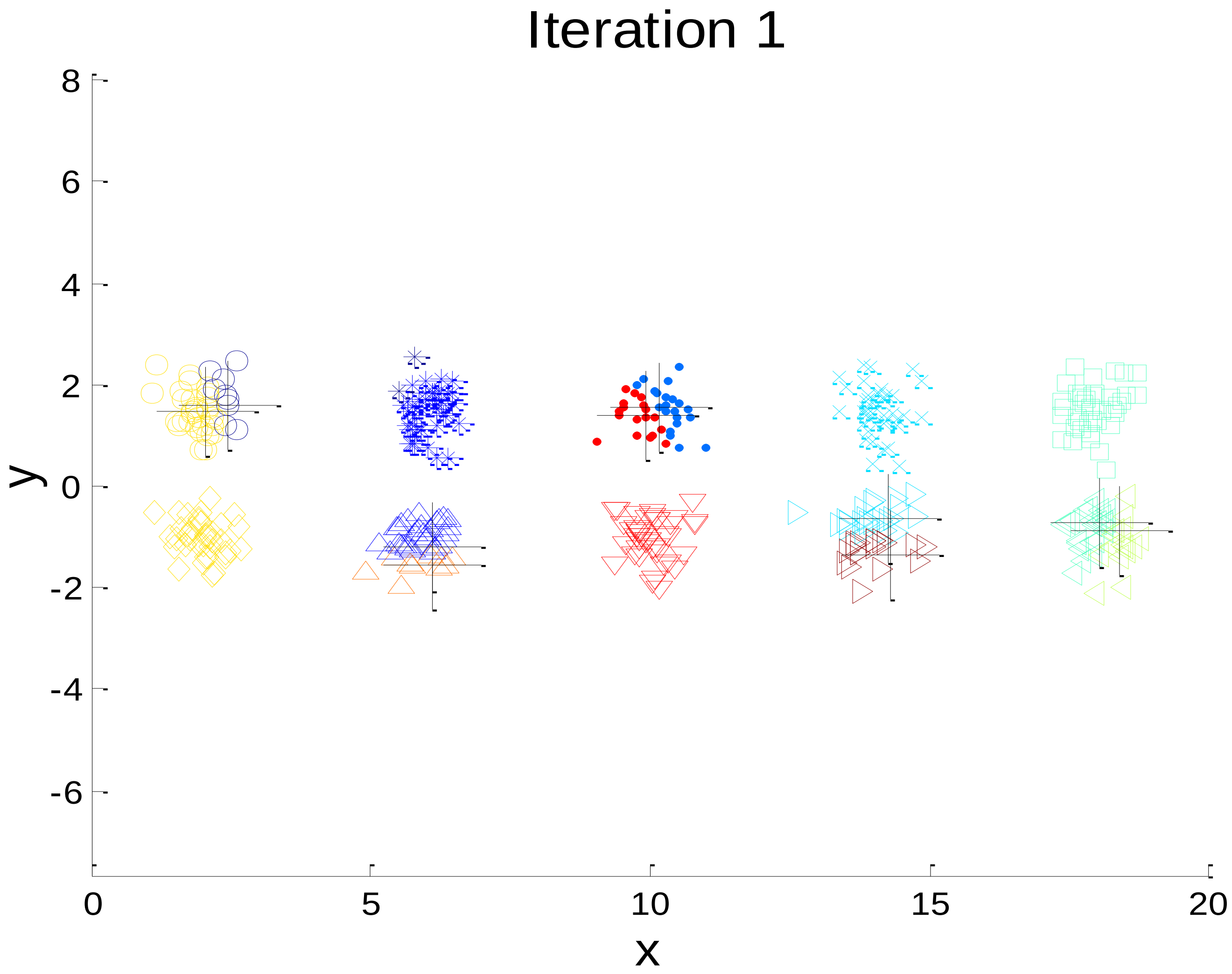
- If there are K ‘real’ clusters each of size n then the chance of selecting one centroid from each cluster is small:

$$P = \frac{\text{number of ways to select one centroid from each cluster}}{\text{number of ways to select K centroids}}$$

$$= \frac{(K-1)n}{Kn} \cdot \frac{(K-2)}{Kn} \cdot \ldots \cdot \frac{1}{Kn} = \frac{(K-1)!\, n^{K-1}}{K^{K-1} n^{K-1}} = \frac{K!}{K^k}$$

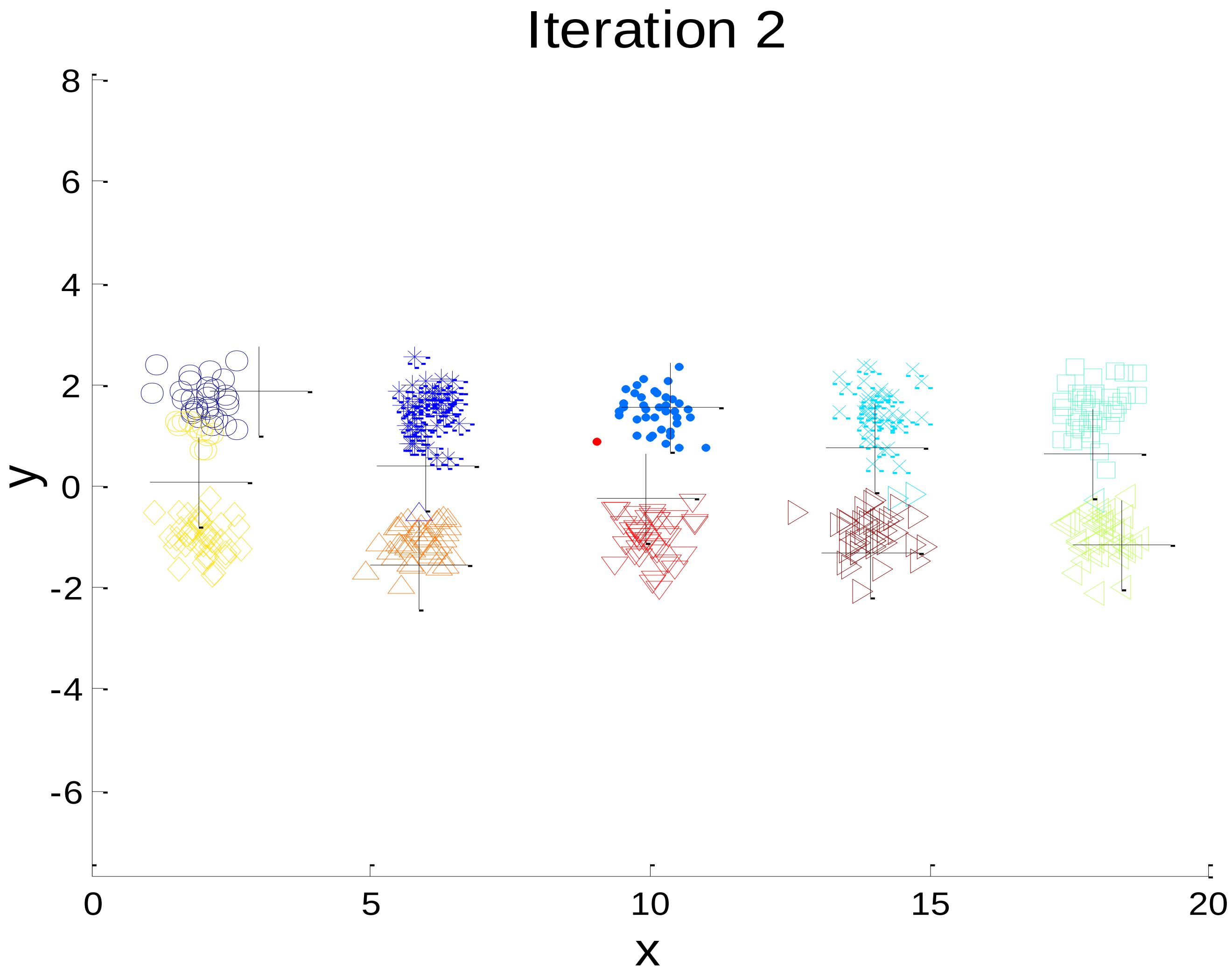
- Thus chance is relatively small when K is large and if clusters are about the same size
- Consider an example of five pairs of clusters: if K = 10, then probability = $10!/10^{10}$ = 0.00036
- Sometimes the initial centroids will readjust themselves in ‘right’ way, and sometimes they don’t

10 Clusters Example - Lucky Centroids



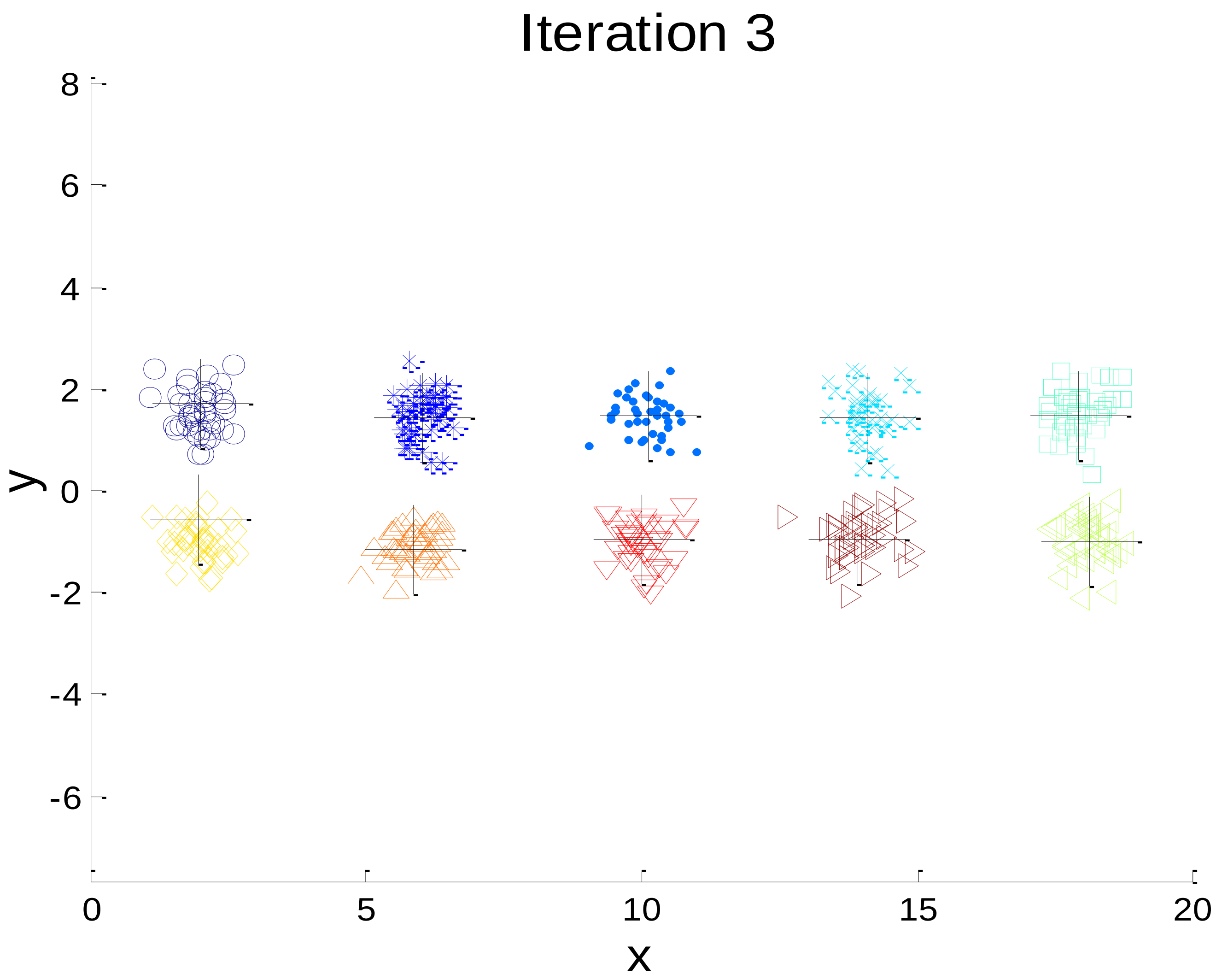
Starting with two initial centroids in one cluster of each pair of clusters

10 Clusters Example - Lucky Centroids



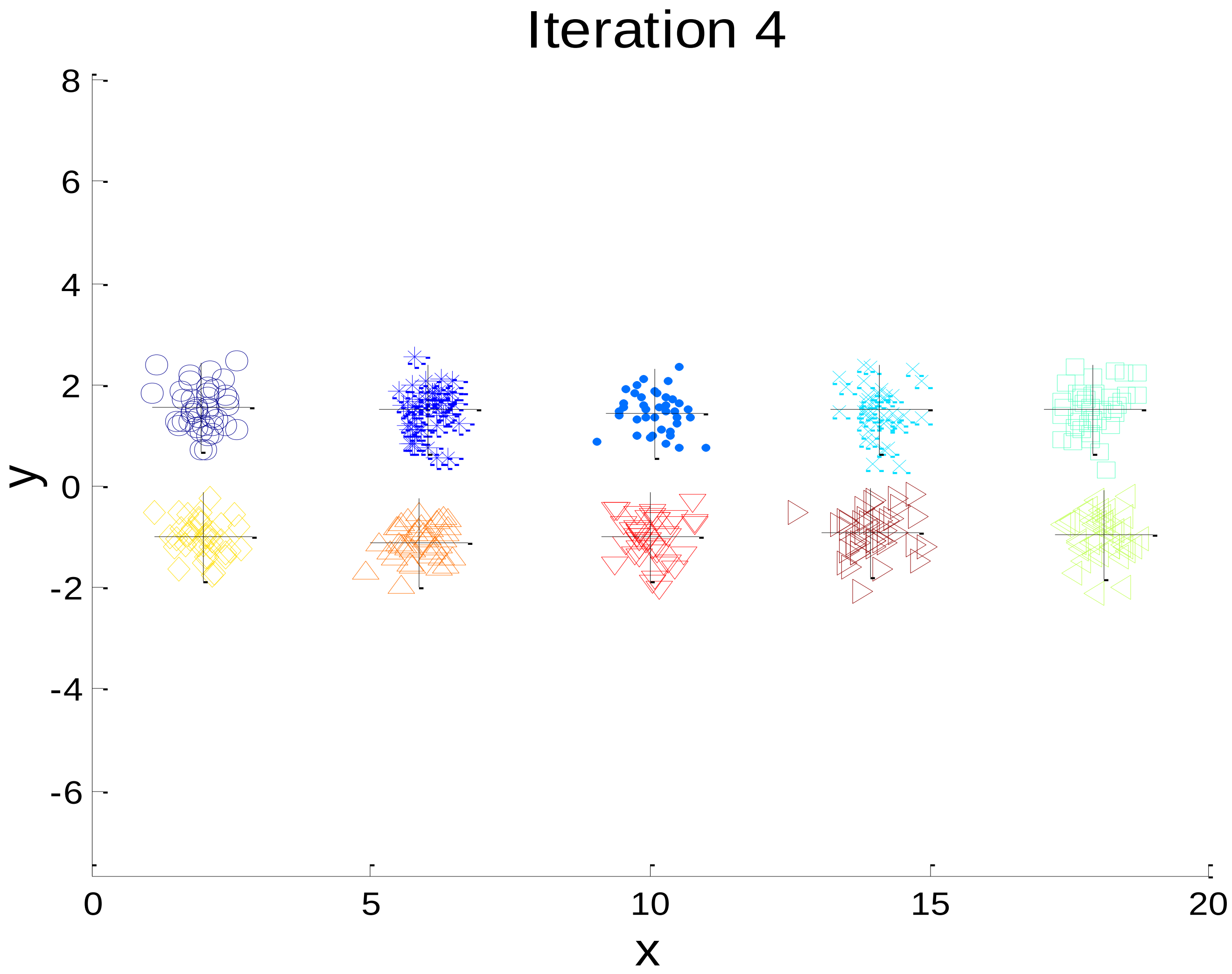
Starting with two initial centroids in one cluster of each pair of clusters

10 Clusters Example - Lucky Centroids



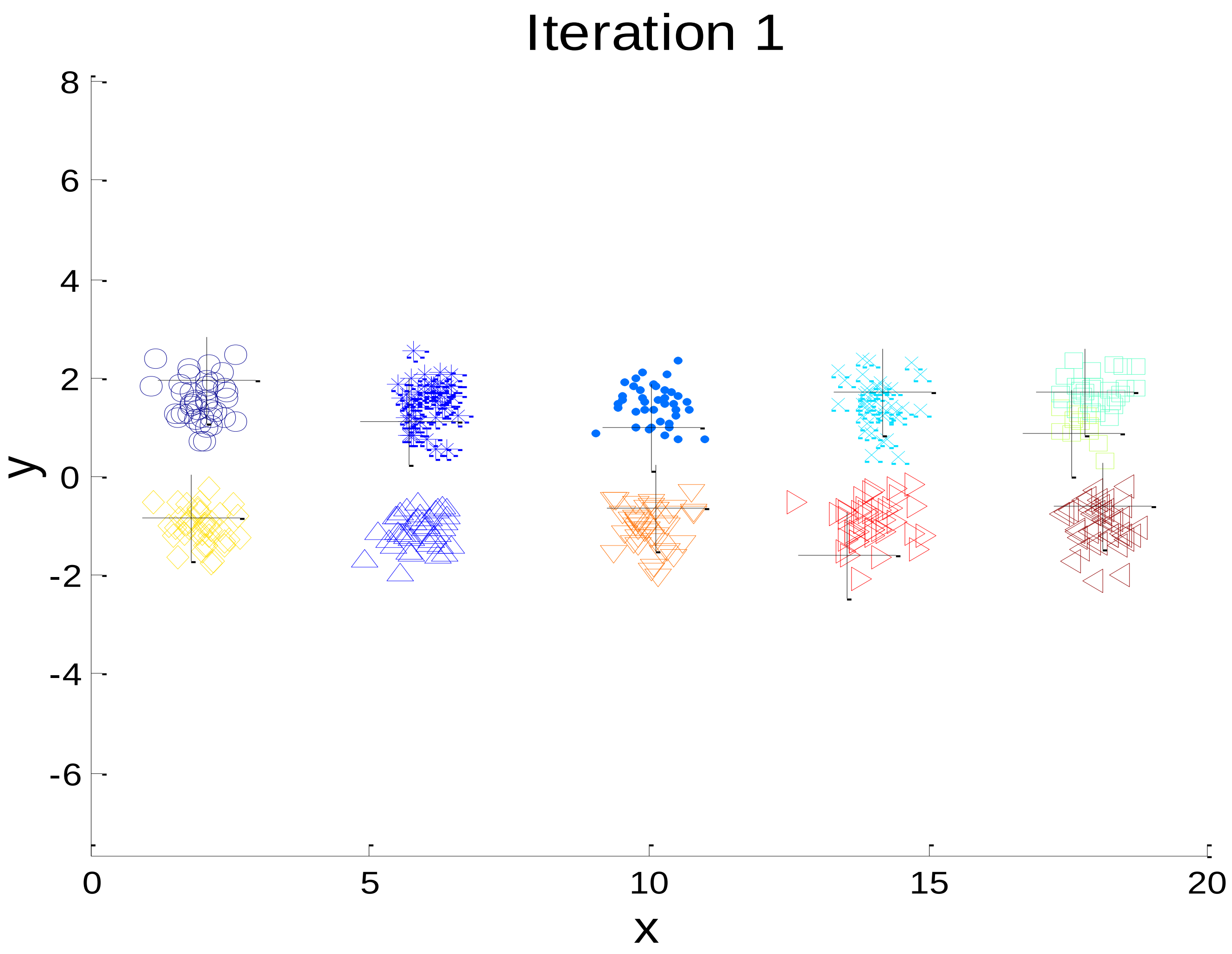
Starting with two initial centroids in one cluster of each pair of clusters

10 Clusters Example - Lucky Centroids



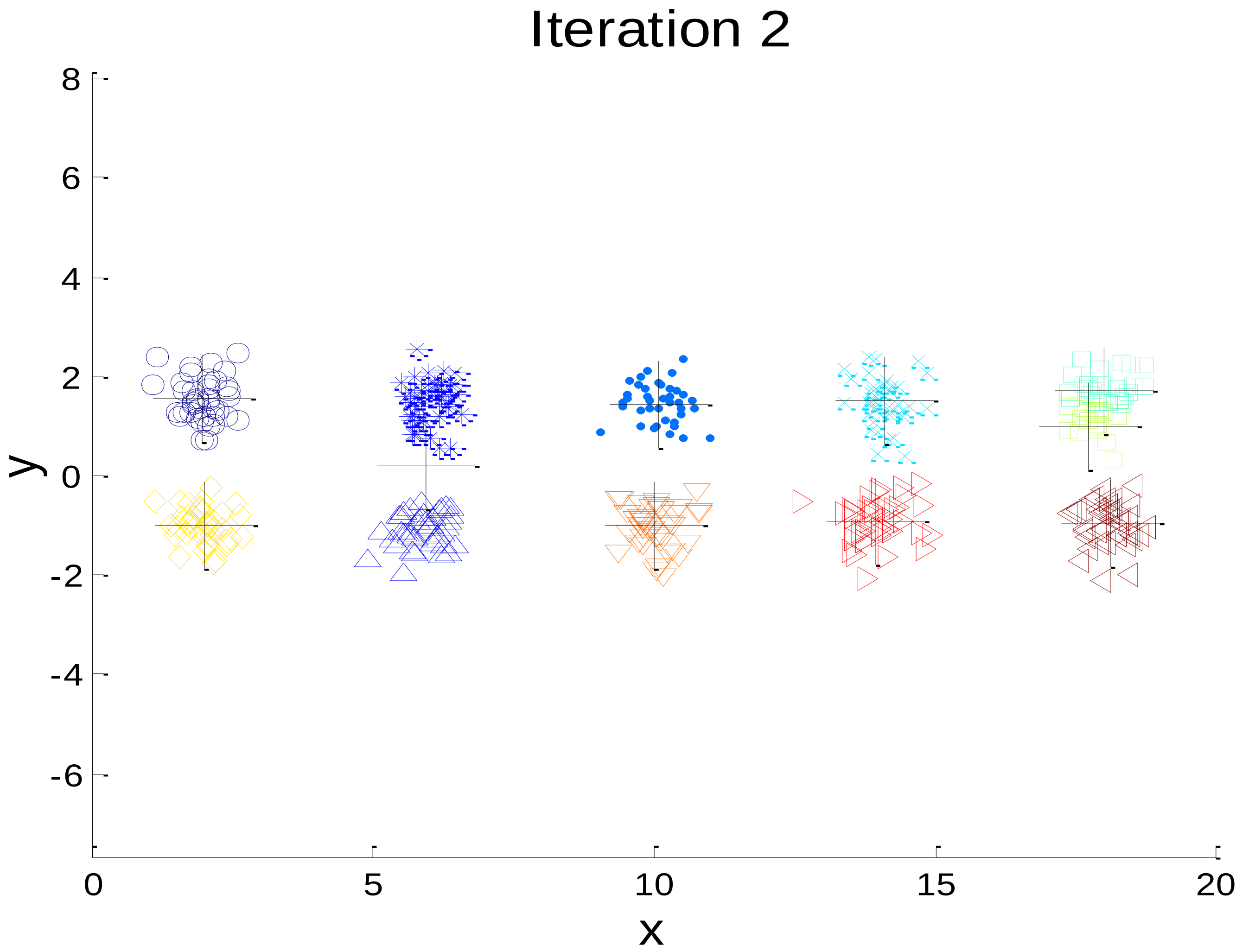
Starting with two initial centroids in one cluster of each pair of clusters

10 Clusters Example - Unlucky Centroids



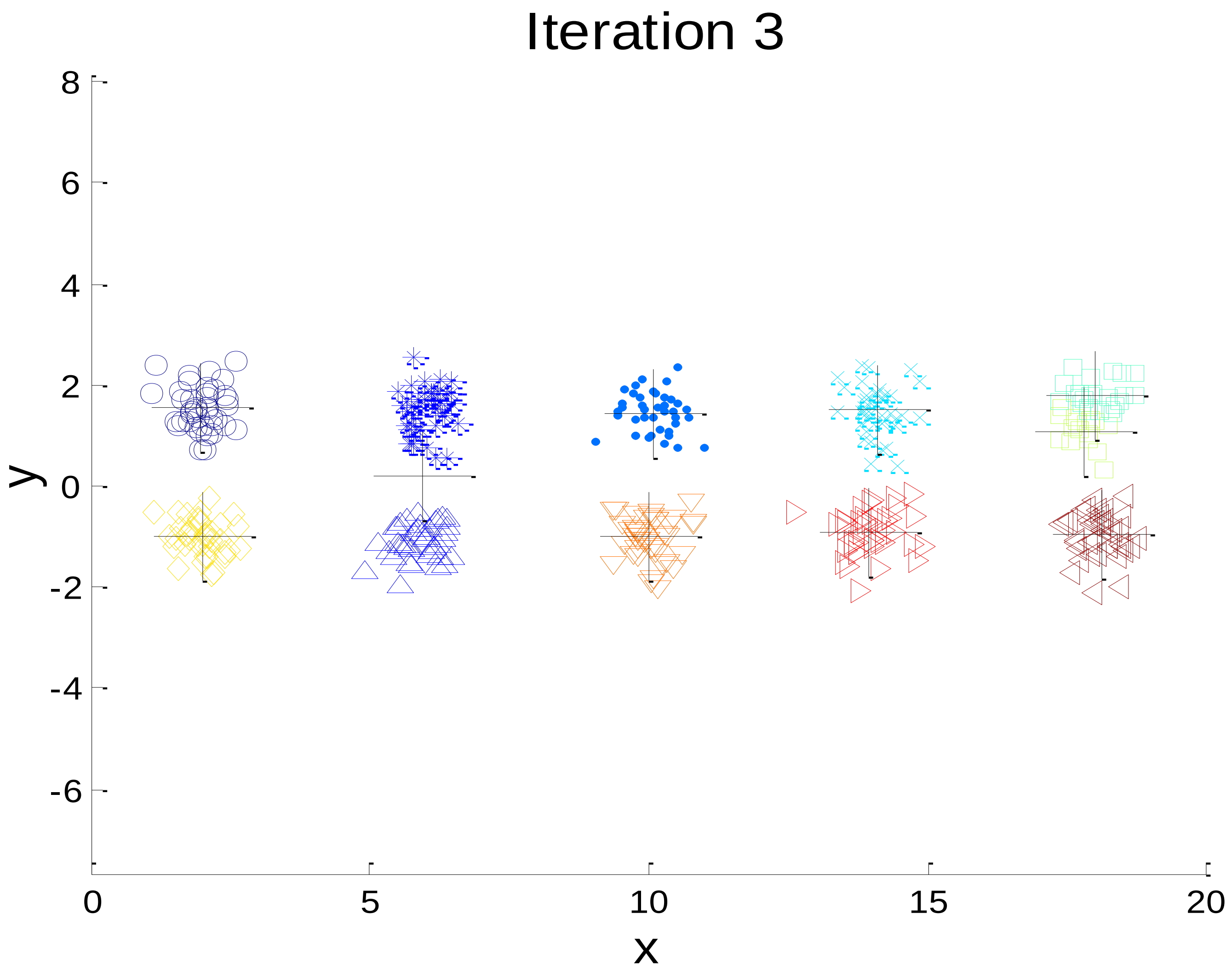
Starting with some pairs of clusters having three initial centroids, while other have only one.

10 Clusters Example - Unlucky Centroids



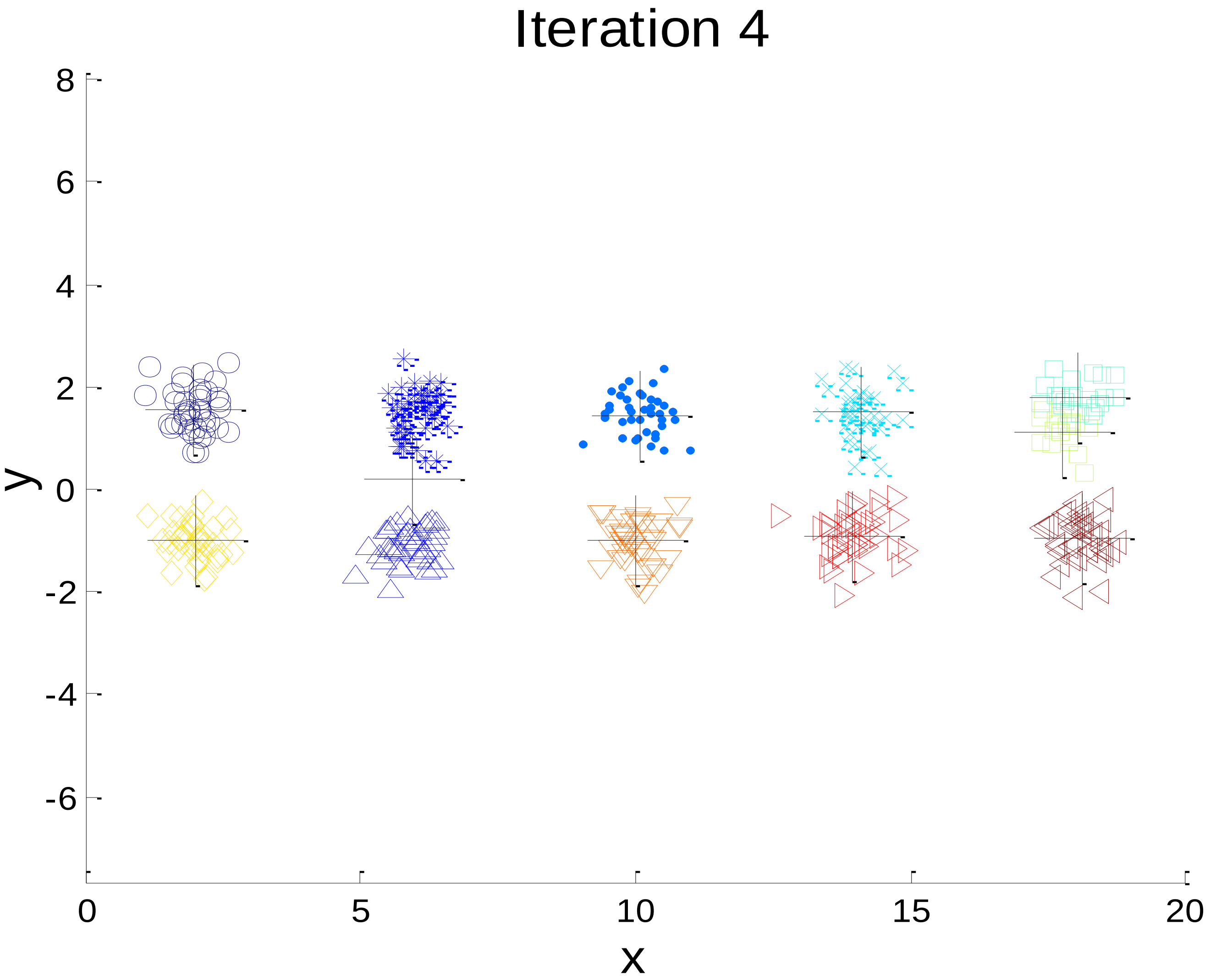
Starting with some pairs of clusters having three initial centroids, while other have only one.

10 Clusters Example - Unlucky Centroids



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

10 Clusters Example - Unlucky Centroids



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

Solutions to Initial Centroids Problem

- Multiple runs
  - Initial assignment of centroids is called configuration. Define number of configuratuions
  - Helps, but probability is not on our side
  - See implementation on next slide

- Select more than k initial centroids and then select among these initial centroids
  - Select most widely separated data points
  - Add post-processing – merge some clusters
    - See a little bit later
- Bisecting K-means (in the next lecture)
  - Not as susceptible to initialization issues
- Sample and use hierarchical clustering to determine initial centroids (we'll see later)

K-means in R

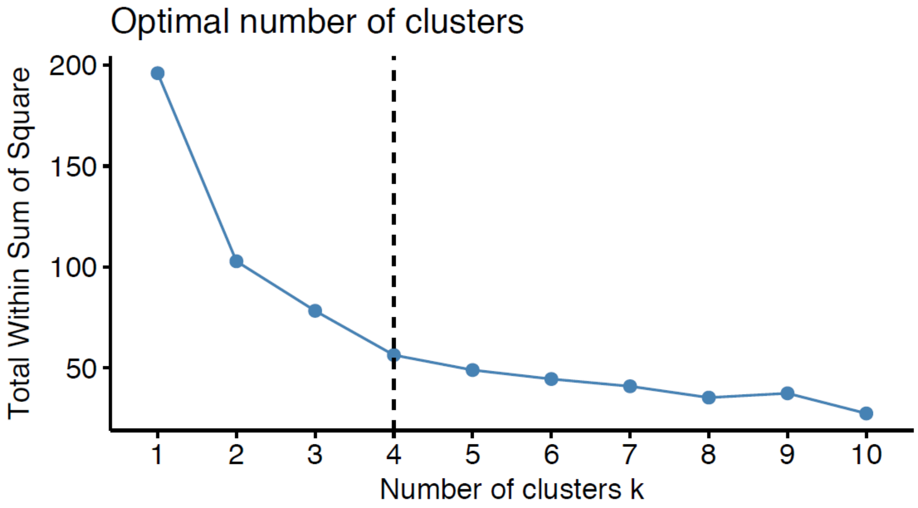
```
library(tidyverse)  # data manipulation
library(cluster)    # clustering algorithms
library(factoextra) #clustering algorithms & visualization
arr_data <- scale(na.omit(USArrests))
distance <- get_dist(arr_data)
fviz_dist(distance, gradient = list(low = "#00AFBB", mid = "white", high =
"#FC4E07"))
k3 <- kmeans(arr_data, centers = 3, nstart = 25)
str(k3)
p2 <- fviz_cluster(k3, geom = "point",  data = arr_data) + ggtitle("k = 3");p2
#fviz_cluster will perform principal component analysis (PCA) and plot the
data points according
#to the first two principal components that explain the majority of the
variance.
arr_data%>%
  as_tibble() %>% mutate(cluster = k3$cluster,state = row.names(arr_data))%>%
ggplot(aes(UrbanPop, Murder, color = factor(cluster), label = state)) +
geom_text()
# the function %>% is a pipe. It passes the left hand side of the operator
# to the first argument of the right hand side of the operator.
#as.tibble same as as.frame applied to a list or a table
#Mutate adds new variables and preserves existing
#ggpolt - graphics plot. Its args describe how variables in the data are
mapped to visual properties  (aesthetics) of geoms17
```

How Many Clusters?

What if we do not know the number of clusters?

Simple solution:

- compute k-means clustering using increasing number of clusters k
  - Track within clusters sum of squares (wss)
- Construct a graph of total wss as a function of number of clusters
  - It must be smooth decreasing function
- Find the inflection point of this function (elbow). This point is the number of clusters that should be used



- More than 4 clusters do not significantly decrease wss

Number of Clusters/Initial Points in R

```
library(tidyverse)  # data manipulation
library(cluster)    # clustering algorithms
library(factoextra) #clustering algorithms & visualization
arr_data <- scale(na.omit(USArrests))
distance <- get_dist(arr_data)
wss <- function(k) {
  kmeans(arr_data, k, nstart = 10 )$tot.withinss
}
# Compute and plot wss for k = 1 to k = 15
k.values <- 1:15
# extract wss for 2-15 clusters
wss_values <- map_dbl(k.values, wss)
#map functions transform their input by applying a
function to each element and returning a vector
#the same length as the input.
plot(k.values, wss_values,
     type="b", pch = 19, frame = FALSE,
     xlab="Number of clusters K",
     ylab="Total within-clusters sum of squares")
```

Lecture Overview

1. Recap
2. K-means Algorithm
3. K-means Optimization
4. Initial Centroids
5. K-medoids Algorithm

K-medoids

- Problem with k-means: outliers shift means
- Solution: instead of centroid use representative points that are among data points (medoids)
- Assuming we have computed distance/dissimilarity matrix

1. Select k objects to become the medoids, or in case these objects were provided use them as the medoids;
2. Calculate the dissimilarity matrix if it was not provided;
3. Assign every object to its closest medoid;
4. For each cluster search if any of the object of the cluster decreases the total within dissimilarity sum (twds); if it does, select the entity that decreases the total twds the most as the medoid for this cluster;
5. If at least one medoid has changed go to (3), else end the algorithm.

PAM in R

```
library(cluster)
library(factoextra)
data("USArrests") # Load the data set
df <- scale(USArrests) # Scale the data
head(df, n = 3) # View the firt 3 rows of the data
library(factoextra)
fviz_nbclust(df, pam, method = "silhouette")+
  theme_classic()
pam.res <- pam(df, 2)
print(pam.res)
pam.res$medoids
fviz_cluster(pam.res,
             palette = c("#00AFBB", "#FC4E07"), # color
palette
             ellipse.type = "t", # Concentration ellipse
             repel = TRUE, # Avoid label overplotting
(slow)
             ggtheme = theme_classic()
)
```

Reading

- TSK 7.1-7.2