Lecture 4

Scalable K-means

Haiping Lu

https://github.com/haipinglu/ScalableML

COM6012 Scalable Machine Learning

Spring 2019

Week 4 Contents / Objectives

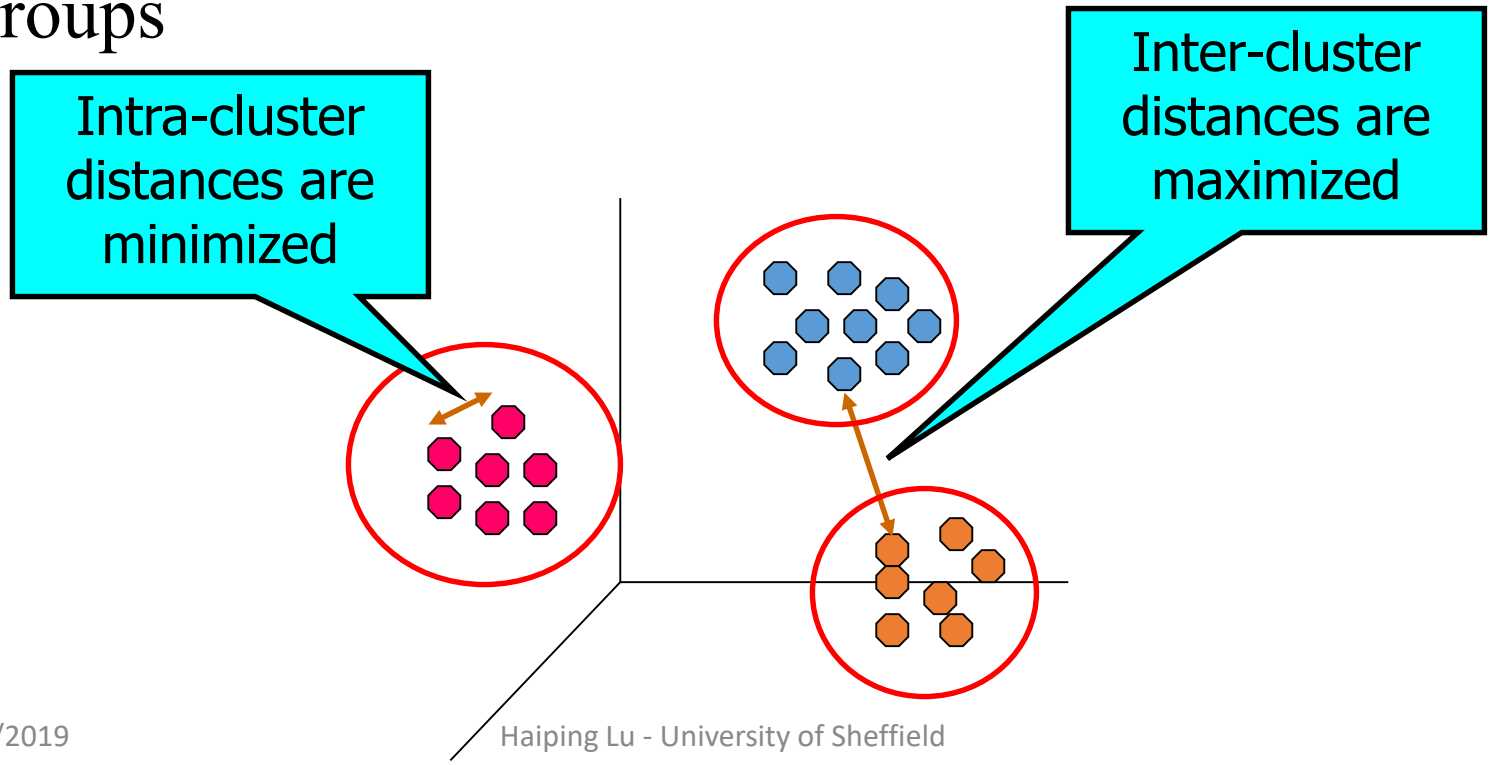
- Introduction to Cluster Analysis
- K-means Clustering
- Scalable K-means
- K-means in Spark & Limitations

Week 4 Contents / Objectives

- **Introduction to Cluster Analysis**
- K-means Clustering
- Scalable K-means
- K-means in Spark & Limitations

What is Cluster Analysis?

- Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups



Cluster Analysis

- Divide data into (clusters) that are meaningful, useful, or both
- The study of techniques for automatically finding classes
- Clusters can help capture the natural structure of the data
- A starting point to further analysis
- An important role in a wide variety of fields:
  psychology, biology, statistics, pattern recognition, information retrieval, machine learning and data mining, etc

Clustering for Understanding

- Classes, or conceptually meaningful groups of objects that share some similarities, play an important role in how people analyze the describe the world
- Human beings are skilled at dividing objects into groups (clustering) and assigning particular objects to these groups (classification). E.g. children can quickly label the objects in a photograph as buildings, vehicles, people, animals, etc

Applications of Clustering

- Biology
  - Cluster analysis help create taxonomy of all living things: kingdom, phylum, class, order, family, etc
  - Cluster analysis on gene / protein data help annotate the function of genes / proteins
- Information retrieval.
  - Clustering help group the search results into a small number of clusters, each of which captures a particular aspect of the query. E.g. a query of “movie” might return Web pages grouped into categories such as reviews, trailers, starts, and theaters
- Climate
  - Cluster analysis has been applied to find patterns in the atmospheric pressure of polar regions and areas of the ocean that have a significant impact on land climate

Applications of Clustering

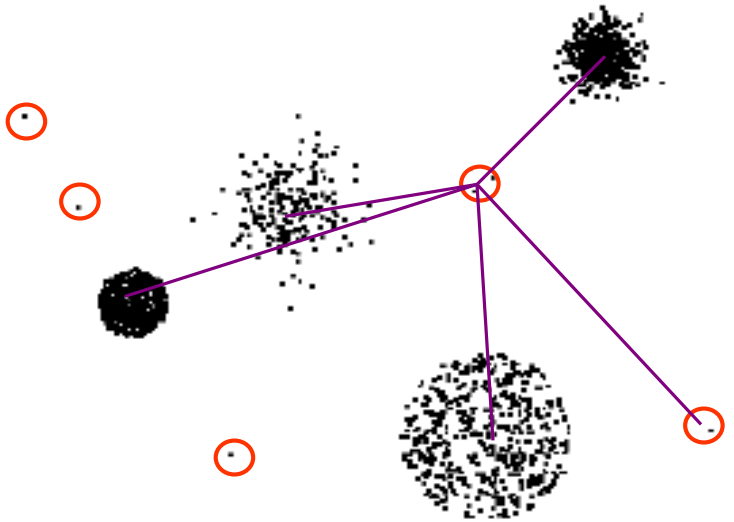
- Psychology and medicine.
  - Identify different types of diseases (e.g. depression)
  - Detect patterns in the spatial or temporal distribution of a disease
  - Help group patients with similar patterns
- Business
  - Clustering analysis can be used to segment customers into a small number of groups for additional analysis and marketing activities
- Anomaly/outlier detection

Anomaly/Outlier Detection

- What are anomalies/outliers?
  - The set of data points that are considerably different than the remainder of the data
- Applications:
  - Credit card fraud detection: purchasing behavior
  - Network intrusion detection: unusual behavior
  - Ecosystem disturbances: typhoon, fire
  - Public health: SARS, bird flu, HxNx
  - Medicine: unusual symptoms/test results

Clustering-Based Anomaly/Outlier Detection

- Cluster the data into groups of different density
- Choose points in small cluster as candidate outliers
- Compute the distance between candidate points and non-candidate clusters.
- If candidate points are far from all other non-candidate points, they are outliers

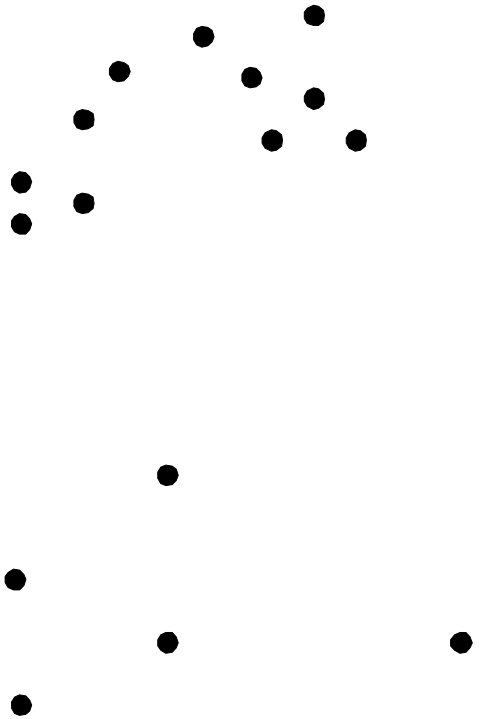


Hierarchical vs Partitional Clustering

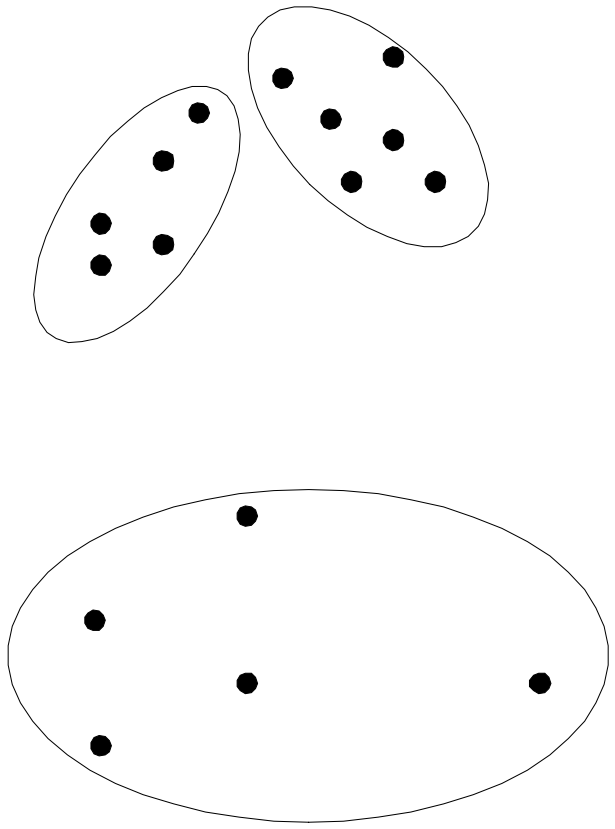
- Partitional Clustering
  - A simply a division of the set of data objects into **non-overlapping** subsets (clusters) such that each data object is in exactly one subset.

- Hierarchical clustering
  - A set of nested clusters organized as a hierarchical tree
  - Each node (cluster) in the tree (except for the leaf nodes) is the union of its children (subclusters), and the root of the tree is the cluster containing all the objects.

Partitional Clustering

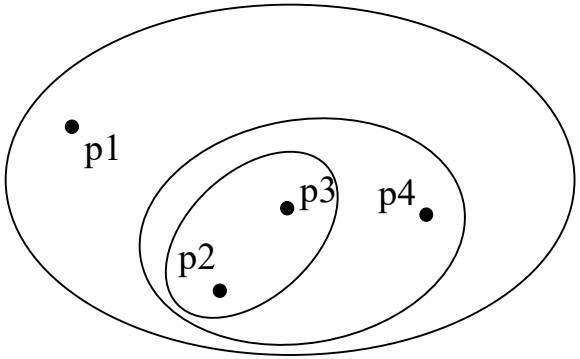


Original Points

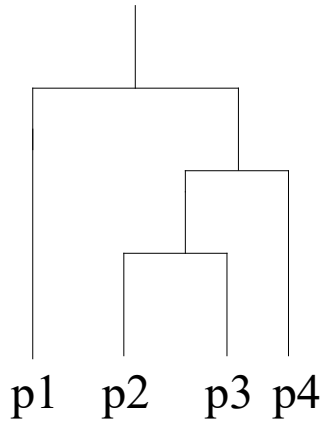


A Partitional Clustering

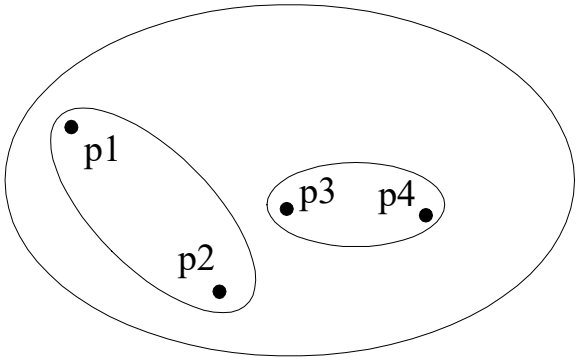
Hierarchical Clustering



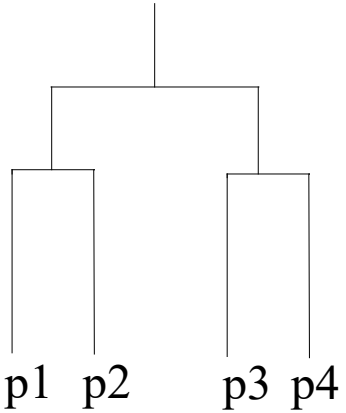
Traditional Hierarchical Clustering



Traditional Dendrogram

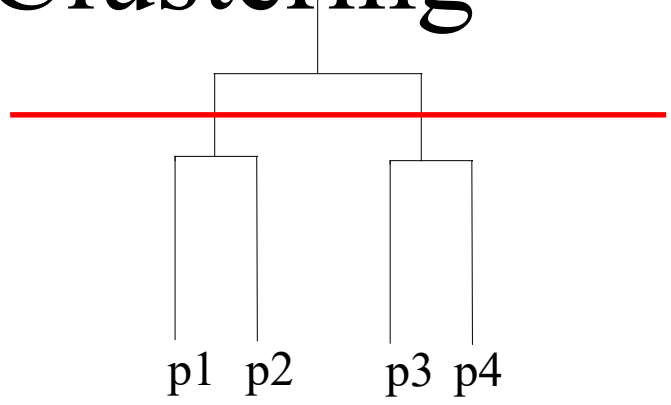


Non-traditional Hierarchical Clustering

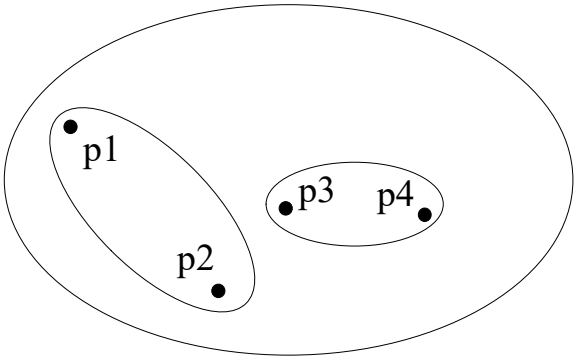


Non-traditional Dendrogram

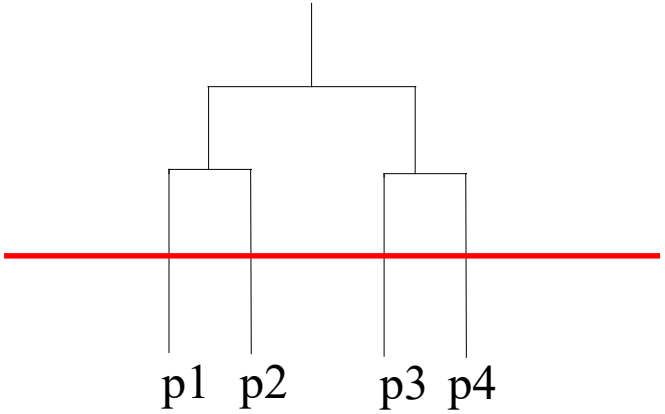
Hierarchical vs Partitional Clustering



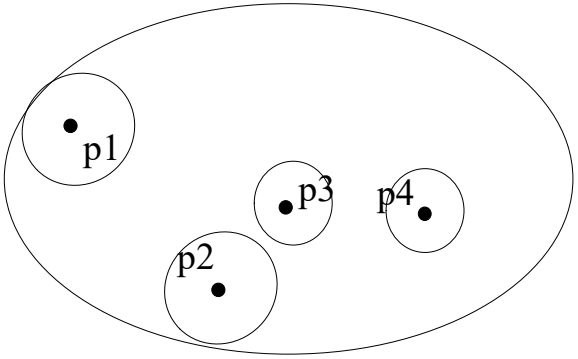
Hierarchical cluster



Partitional cluster



Hierarchical cluster



Partitional cluster

A hierarchical clustering can be viewed as a sequence of partitional clustering and a partitional clustering can be obtained by taking any member of the that sequences by cutting the hierarchical tree at a particular level

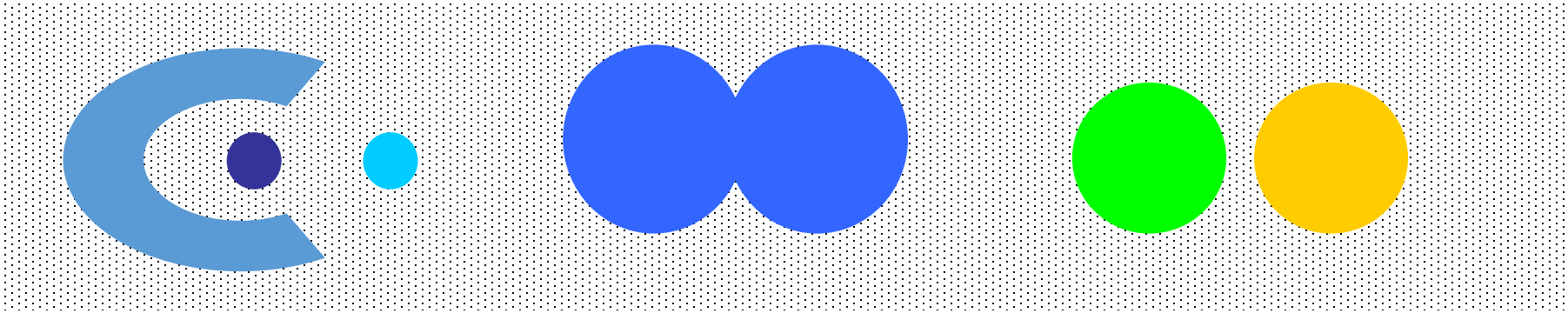
Types of Clusters: Center-Based

- Center-based (Prototype-based)
  - A cluster is a set of objects such that an object in a cluster is closer (more similar) to the "center" of a cluster, than to the center of any other cluster
  - The center of a cluster is often a centroid, the average of all the points in the cluster, or a medoid (when centroid not meaningful, e.g., categorical), the most "representative" point of a cluster

4 center-based clusters

Types of Clusters: Density-Based

- 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

Week 9 Contents

- Introduction to Cluster Analysis
- **K-means Clustering***
- Scalable K-means
- K-means in Spark & Limitations

***Slides credit**: **Bahman Bahmani, Stanford University**

K-means Clustering

- A **prototype**-based, **partitional** clustering approach
- Each cluster is associated with a centroid (centre point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, K, must be specified before clustering

K-means Clustering

- **Input:**
  - A set X={x_1, x_2, …, x_n} of n data points
  - Number of clusters k
- For a set $C=\{c_1, c_2, \ldots, c_k\}$ of cluster "centres" define the Sum of Squared Error (SSE) as:

$$\varphi_X(C) = \sum_{x \in X} d(x,C)^2$$

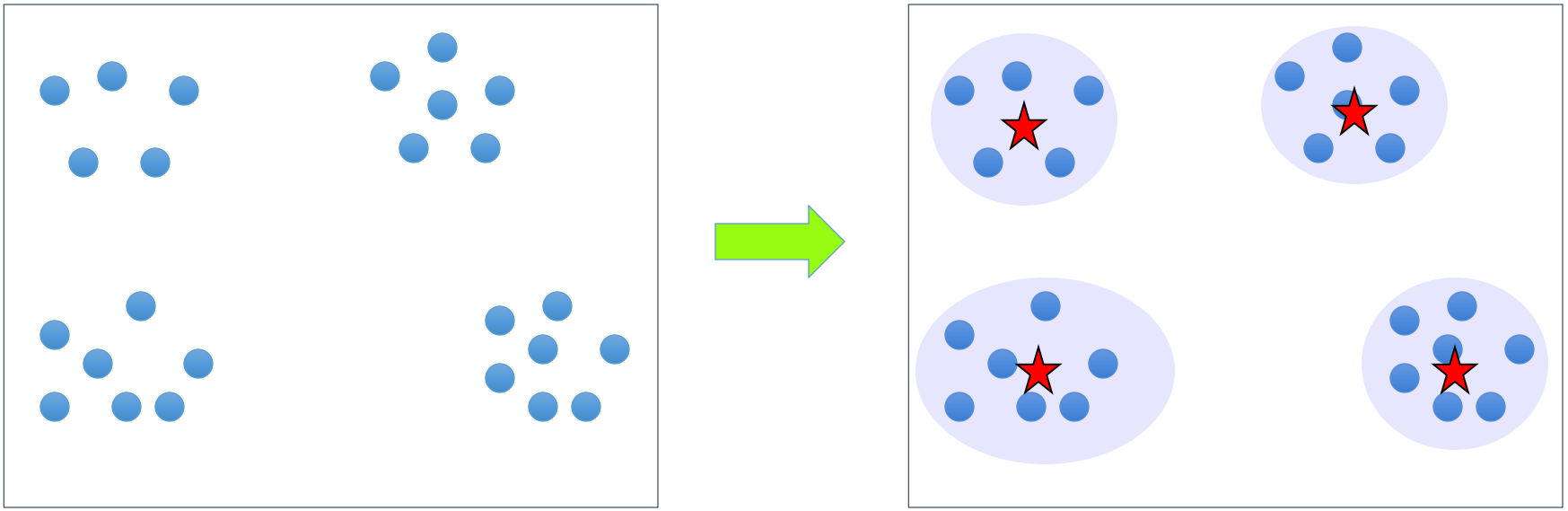
  d(x,C): distance from x to closest centre in C
- **Goal:** To find a set C of centres that minimizes the objective function $\varphi_X(C)$

Determine the number of clusters

There are different approaches of determining K

- K can be arbitrarily set as any number
- K can be determined according to the need of further analysis
- K can be determined according to field knowledge, or the knowledge obtained during data visualisation
- Different K's can be initially set, and find the best K using some criteria

K-means Clustering: Example



K = 4

Lloyd Algorithm

- Start with k arbitrary centres $\{c_1, c_2, \ldots, c_k\}$ (typically chosen uniformly at random from data points)
- Performs an EM-type local search till convergence
- Main advantages: Simplicity, scalability (iterations)

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

What's wrong with Lloyd Algorithm?

- Takes many iterations to converge
- Very sensitive to initialization
- Random initialization can easily get two centres in the same cluster
  - K-means gets stuck in a local optimum

Lloyd Algorithm: Initialization

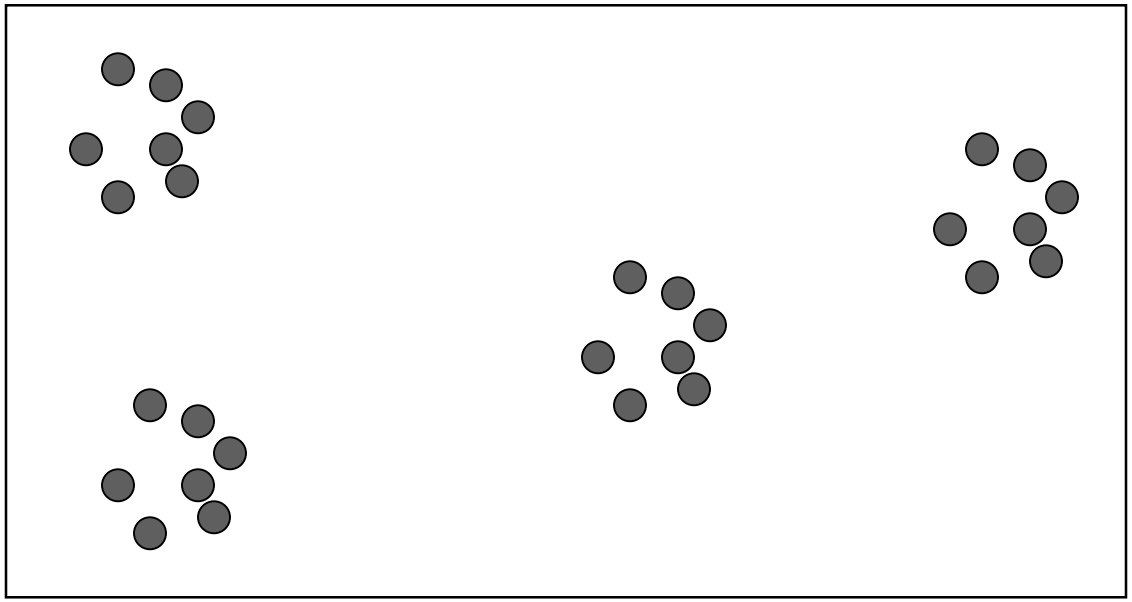


Figure credited to David Arthur

Lloyd Algorithm: Initialization

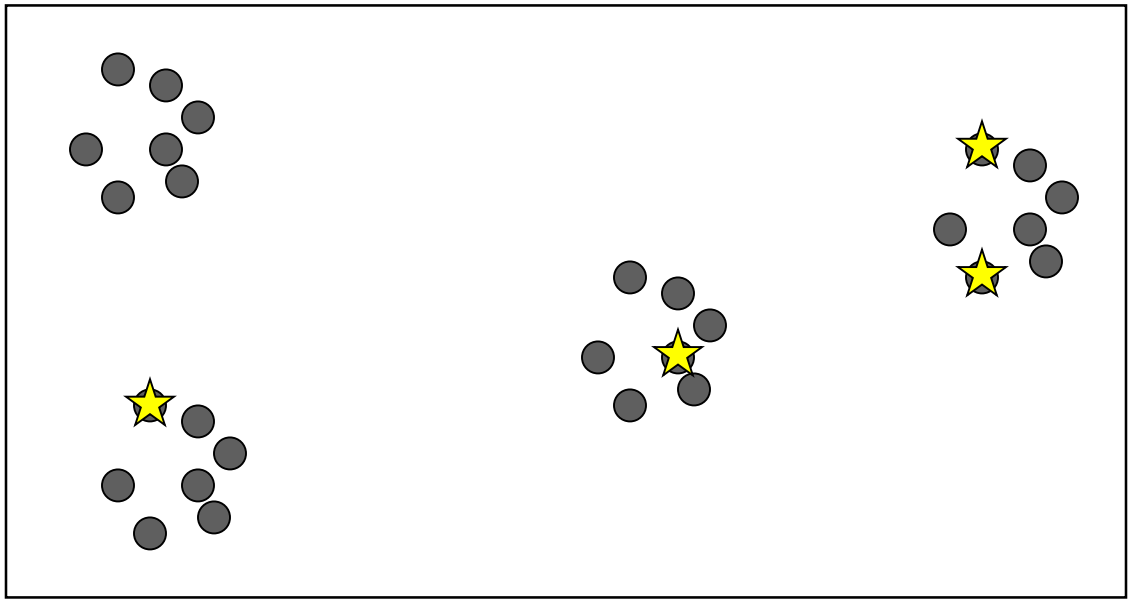


Figure credited to David Arthur

Lloyd Algorithm: Initialization

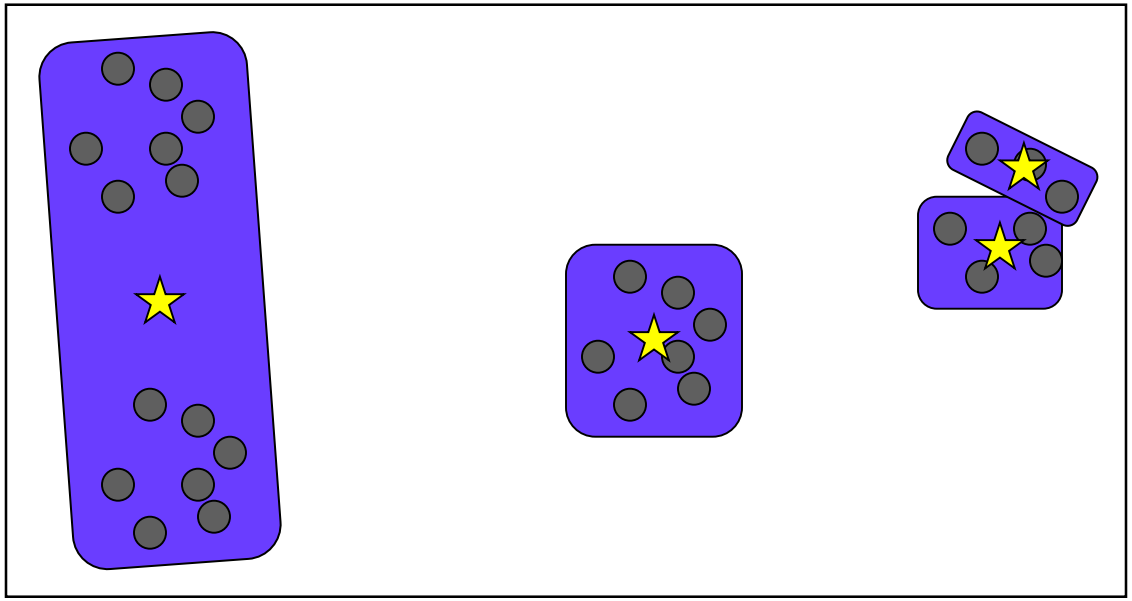


Figure credited to David Arthur

Lloyd Algorithm: Initialization

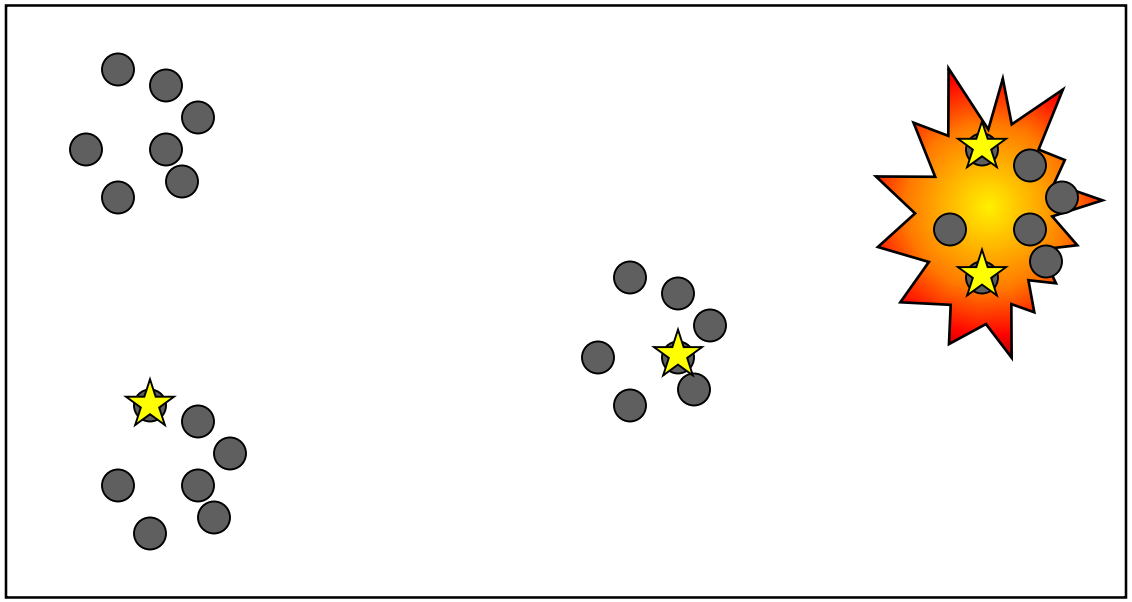


Figure credited to David Arthur

Week 9 Contents

- Introduction to Cluster Analysis
- K-means Clustering
- **Scalable K-means***
- K-means in Spark & Limitations

***Slides credit**: **Bahman Bahmani, Stanford University**

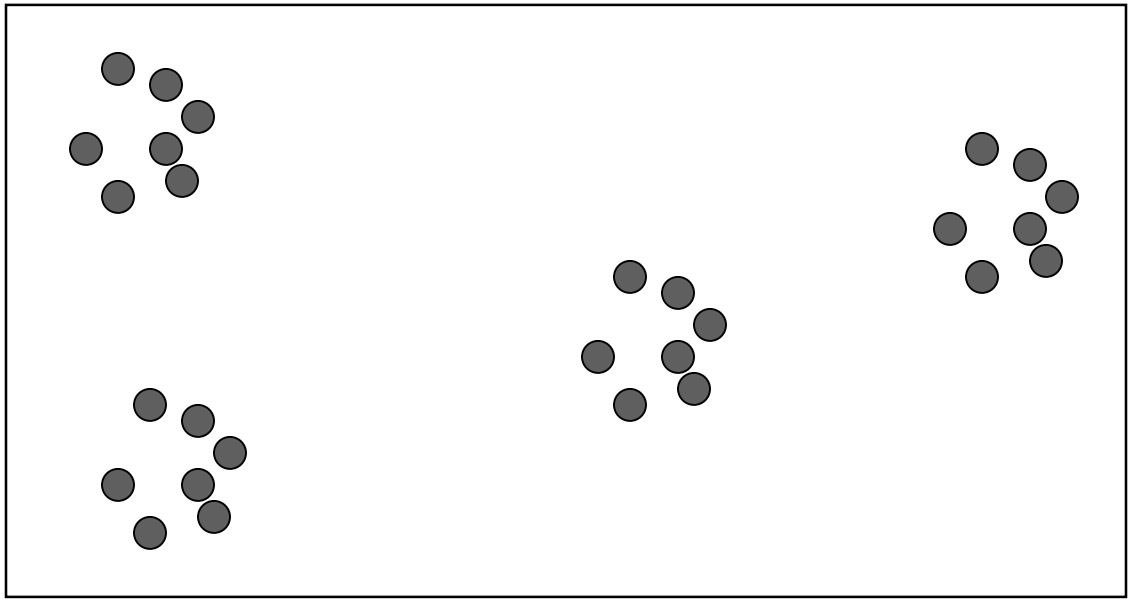
K-means++ [Arthur et al. '07]

- Spreads out the centres
- Choose first centre, c_1, uniformly at random from the data set
- Repeat for $2 \leq i \leq k$:
  - Choose c_i to be equal to a data point x_0 sampled from the distribution:

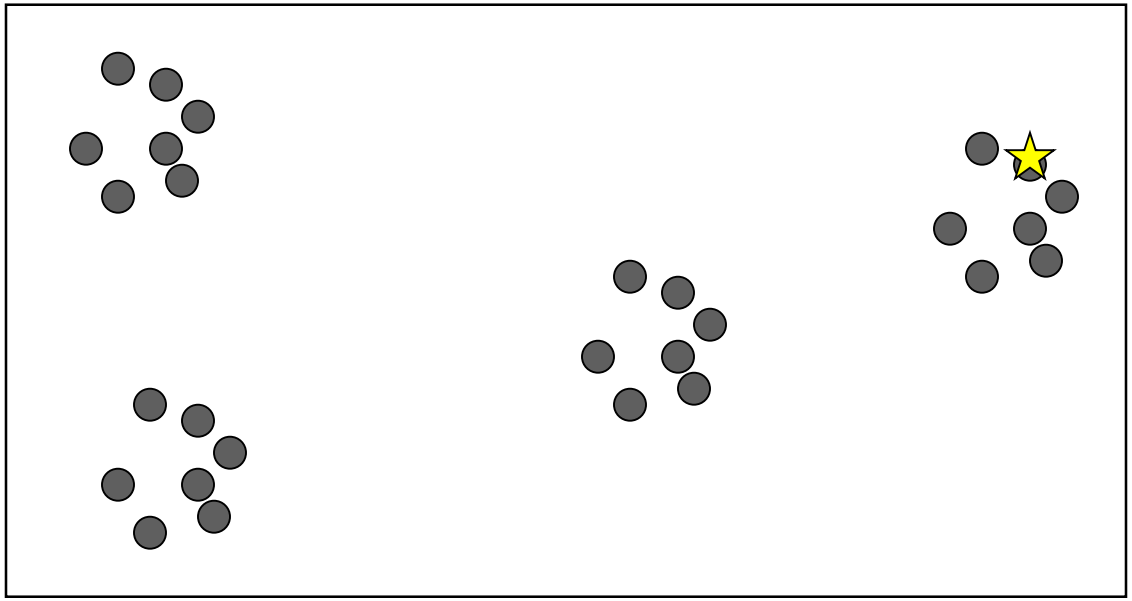
$$\frac{d(x_0, C)^2}{\varphi_X(C)} \propto d(x_0, C)^2$$

- **Theorem:** O(log k)-approximation to optimum, right after initialization

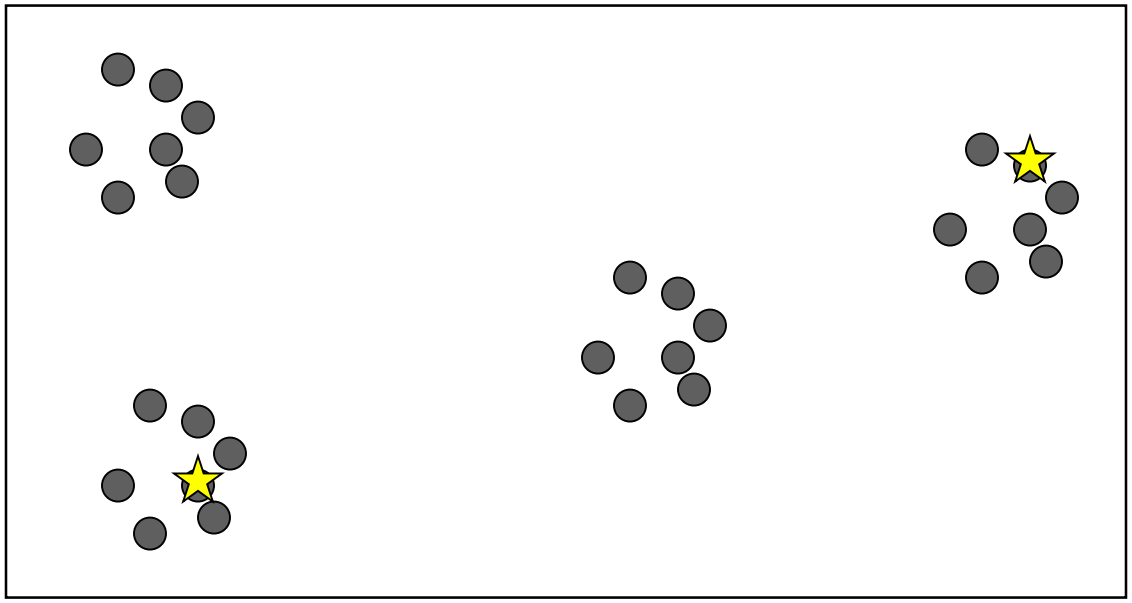
K-means++ Initialization



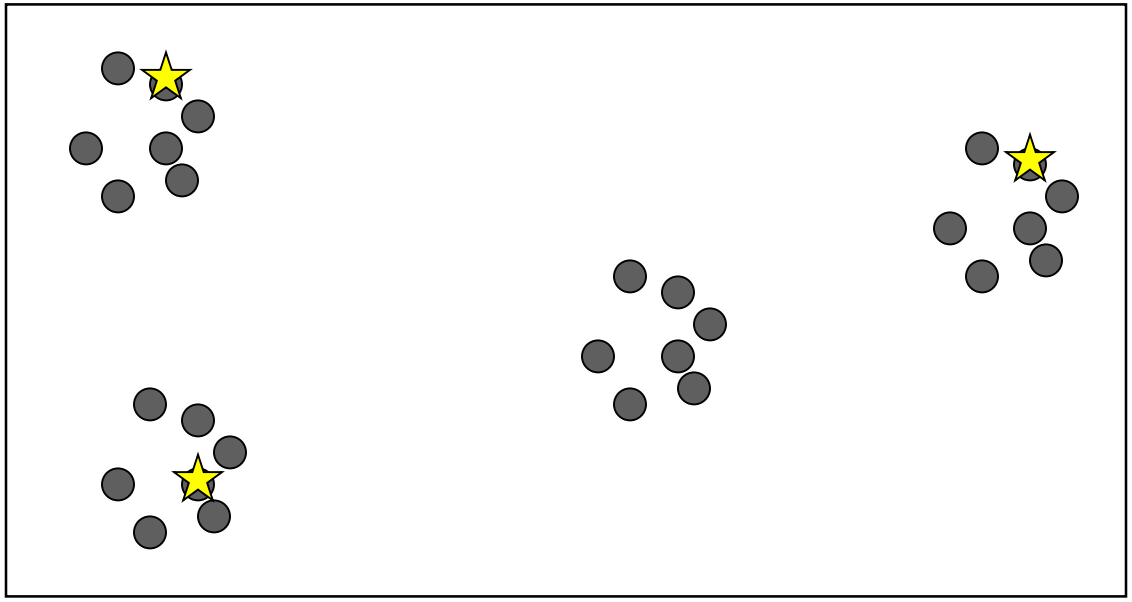
K-means++ Initialization



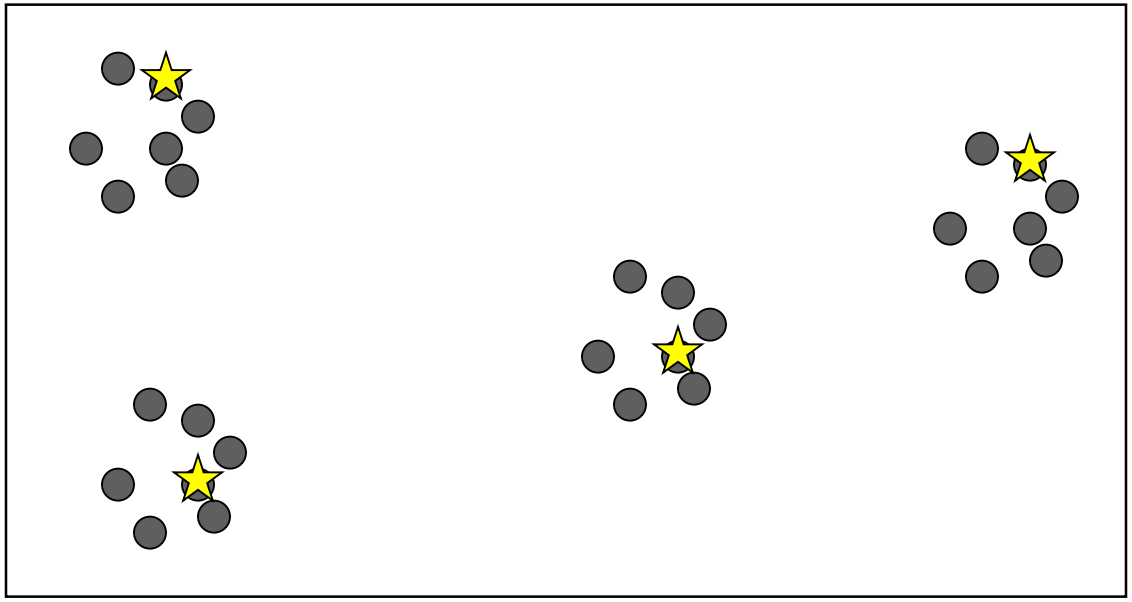
K-means++ Initialization



K-means++ Initialization



K-means++ Initialization



What's Wrong with K-means++?

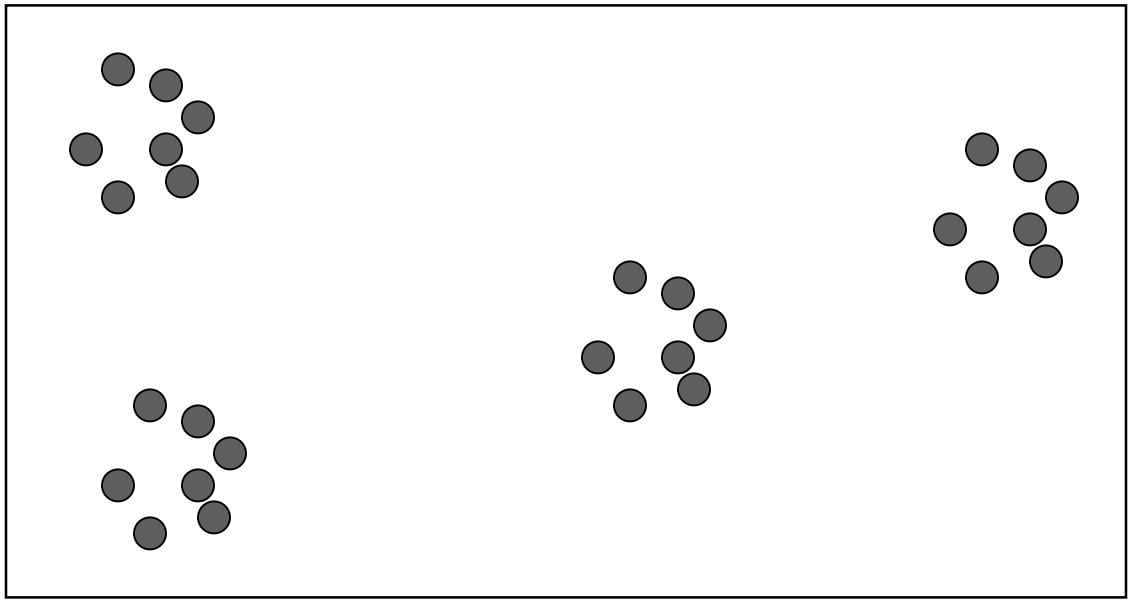
- Needs K passes over the data
- In large data applications, not only the data is massive, but also K is typically large (e.g., easily 1000).
- Does not scale!

Intuition for a Solution

- K-means++ samples one point per iteration and updates its distribution
- What if we **oversample** by sampling each point independently with a larger probability?
- Intuitively equivalent to updating the distribution much less frequently
  - Coarser sampling
- Turns out to be sufficient: K-means||

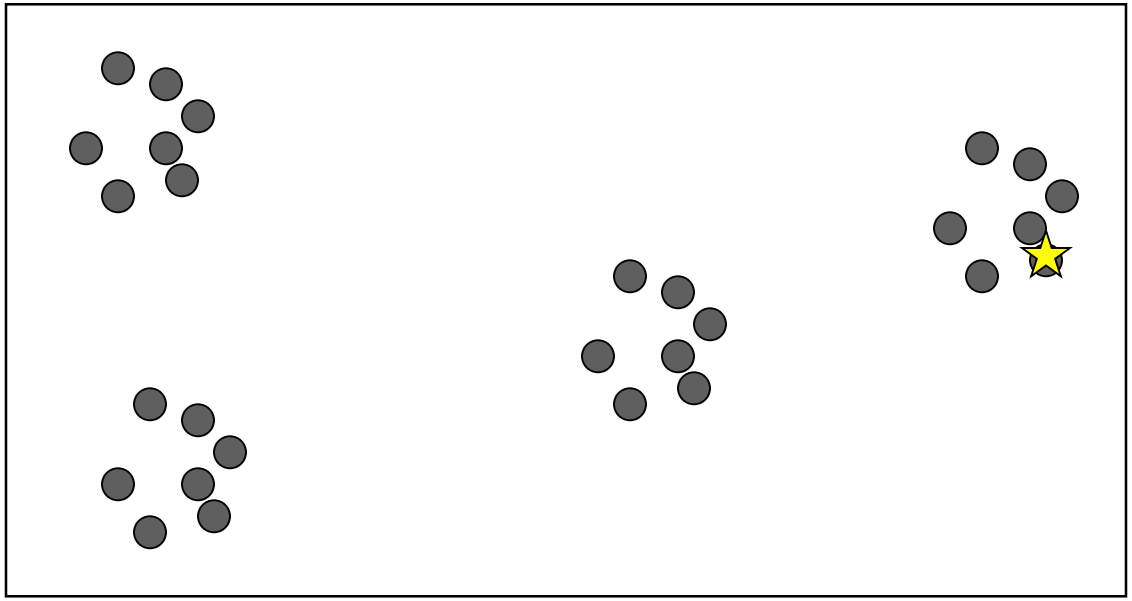
K-means|| Initialization [Bahmani et al. ’12]

K=4,
Oversampling factor L=3



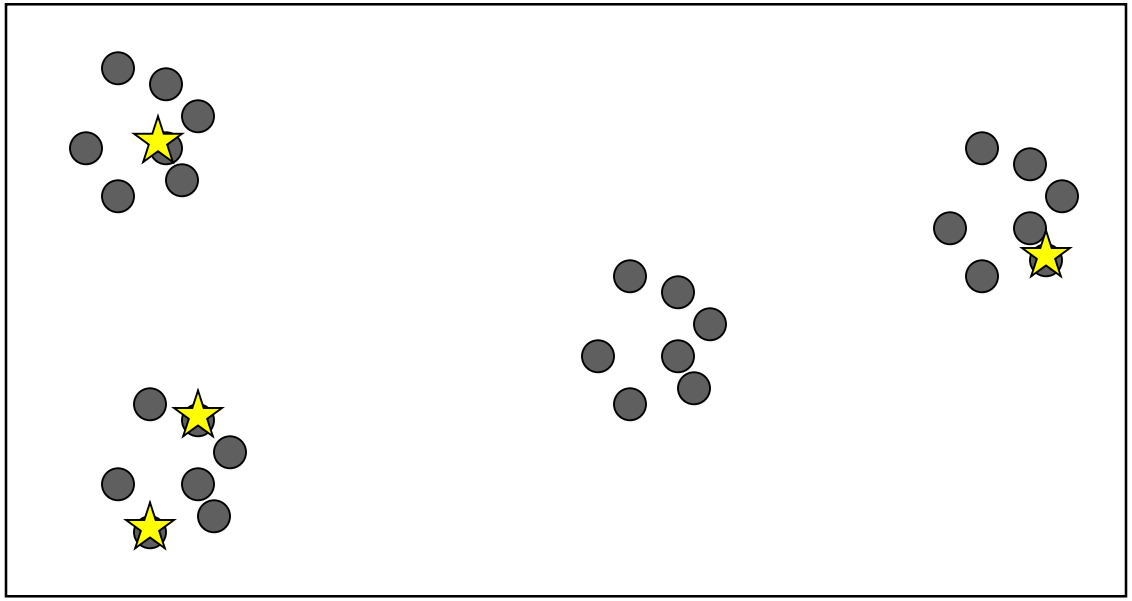
K-means|| Initialization

K=4,
Oversampling factor L=3



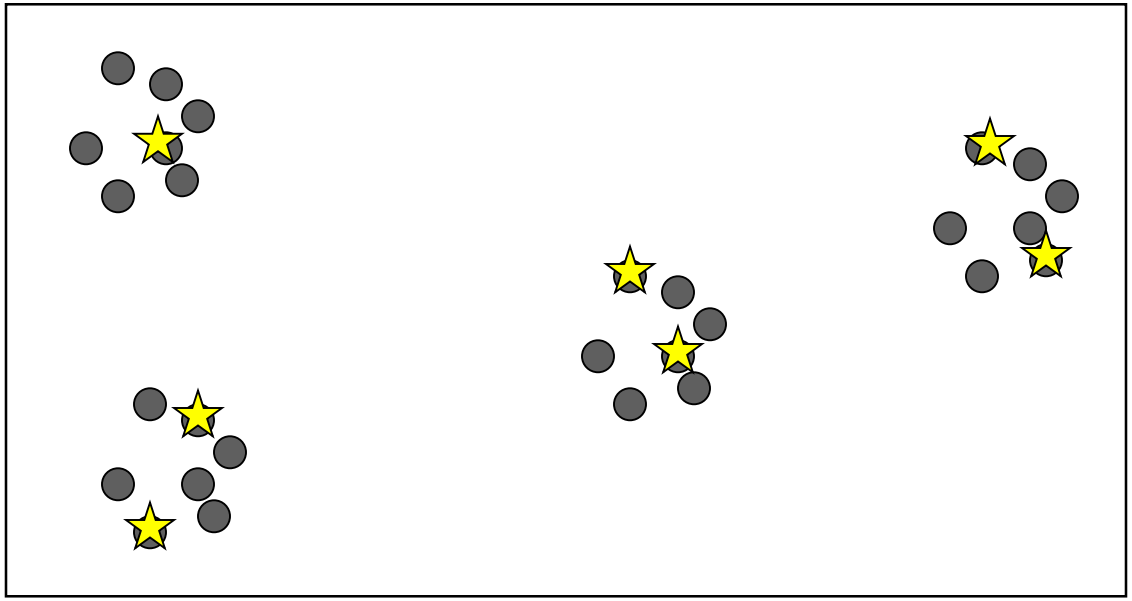
K-means|| Initialization

K=4,
Oversampling factor L=3



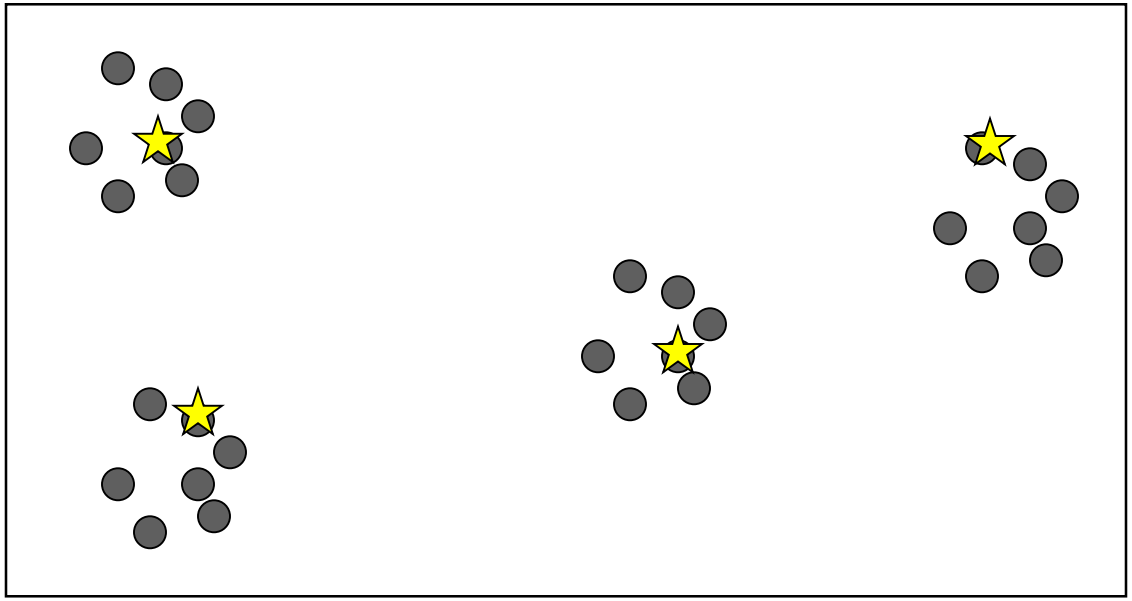
K-means|| Initialization

K=4,
Oversampling factor L=3



K-means|| Initialization

K=4,
Oversampling factor L=3



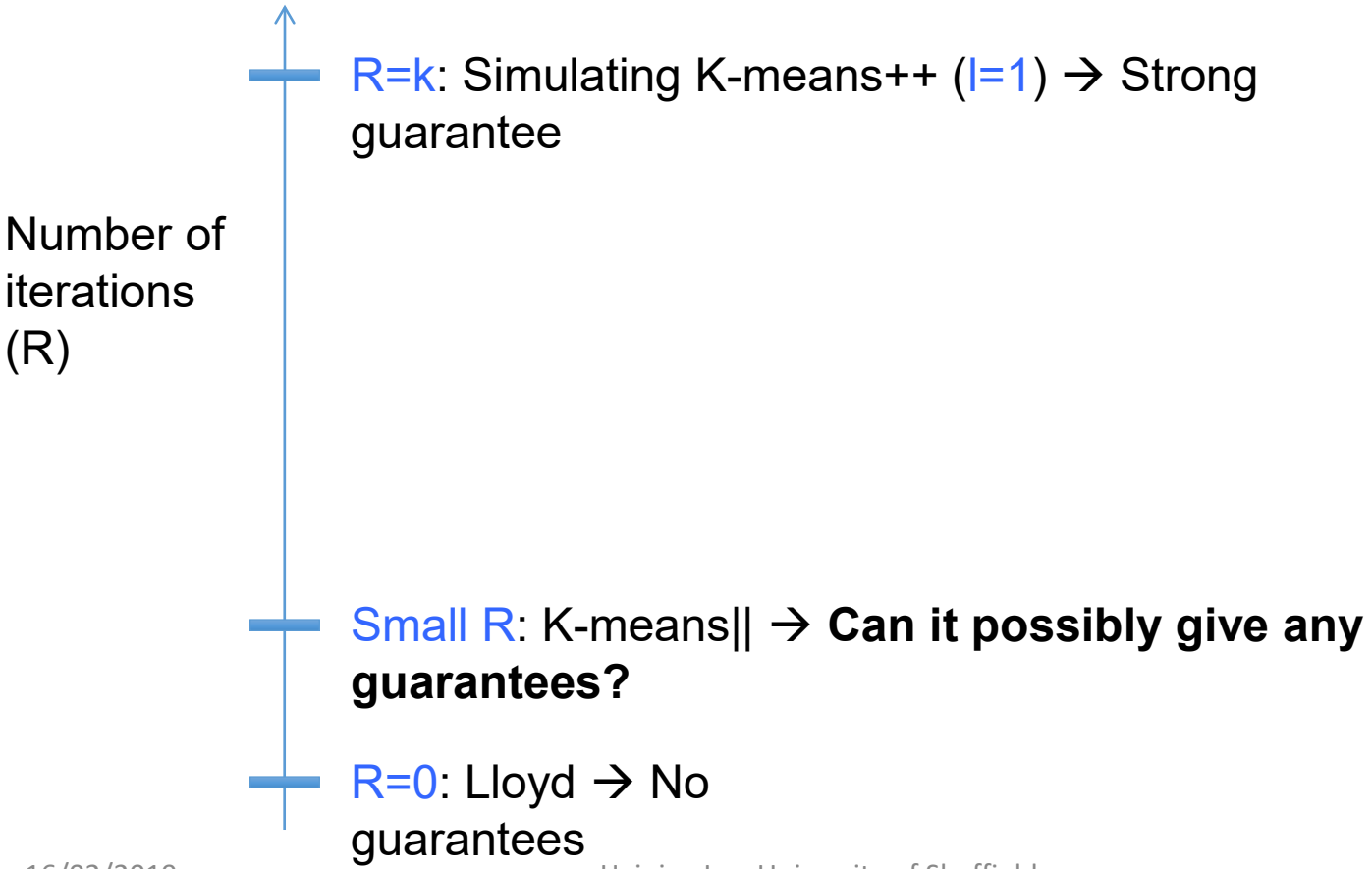
Cluster the intermediate centres

K-means|| [Bahmani et al. ’12]

- Choose L>1
- Initialize C to an arbitrary set of points
- For R iterations do:
  - Sample each point x in X independently with probability $p_x = Ld^2(x,C)/\varphi_X(C)$.
  - Add all the sampled points to C
- Cluster the (weighted) points in C to find the final k centres

K-means||: Intuition

- An interpolation between Lloyd and K-means++



K-means||: Benefits

- Using K-means++ for clustering the intermediate centres, the overall approximation factor = **O(log k)**
- K-means|| much harder than K-means++ to get confused with noisy outliers
- K-means|| reduces number of Lloyd iterations even more than K-means++

Week 9 Contents

- Introduction to Cluster Analysis
- K-means Clustering
- Scalable K-means
- **K-means in Spark & Limitations**

K-means in Spark ML

- Uses **MLlib Kmeans** (Kmeans | |)

- Code:

import org.apache.spark.mllib.clustering.{KMeans => MLlibKMeans, …

https://github.com/apache/spark/blob/v2.3.2/mllib/src/main/scala/org/apache/spark/ml/clustering/KMeans.scala

- MLlib:

https://github.com/apache/spark/blob/v2.3.2/mllib/src/main/scala/org/apache/spark/mllib/clustering/KMeans.scala

K-means in Spark

- *k:* the number of desired clusters.
- *maxIter:* the maximum number of iterations
- *initMode:* specifies either random initialization or initialization via k-means|| (compare)
- *initSteps*: determines the number of steps in the k-means|| algorithm (default=2, advanced)
- *tol:* determines the distance threshold within which we consider k-means to have converged.
- *seed*: setting the random seed (so that multiple runs have the same results)

Running Scalable K-means

- Data should be **cached** for high performance (check warning when you run your program)

```
val centers = initialModel match {
  case Some(kMeansCenters) =>
    kMeansCenters.clusterCenters.map(new VectorWithNorm(_))
  case None =>
    if (initializationMode == KMeans.RANDOM) {
      initRandom(data)
    } else {
      initKMeansParallel(data)
    }
}
```

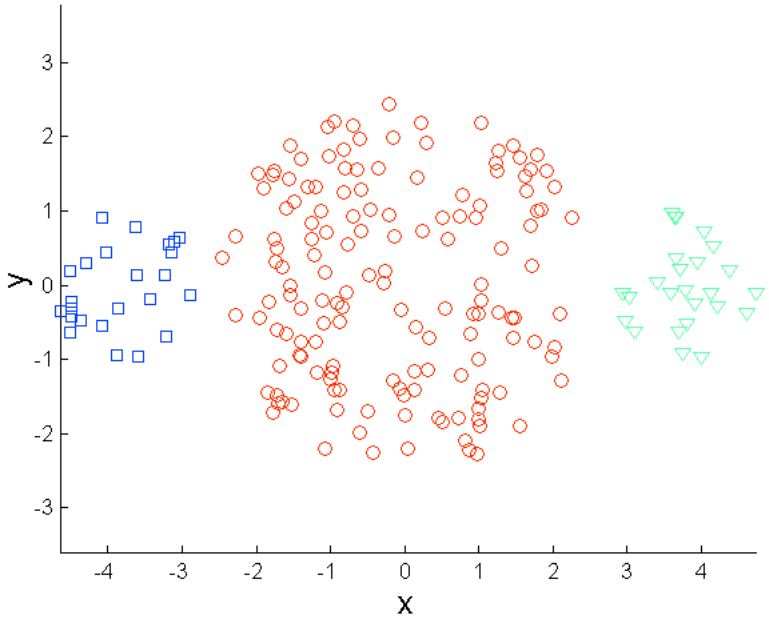
https://github.com/apache/spark/blob/v2.3.2/mllib/src/main/scala/org/apache/spark/mllib/clustering/KMeans.scala

Limitations of K-means

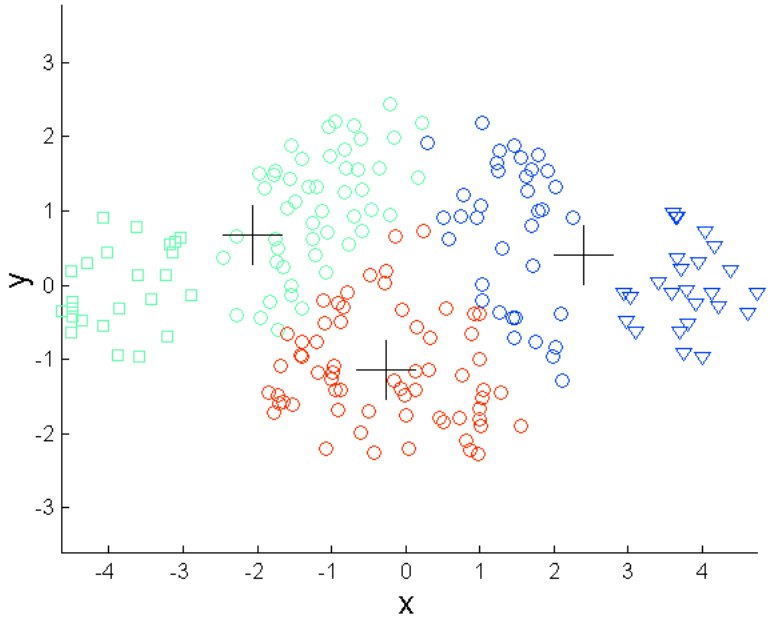
- K-means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-globular shapes

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

Limitations of K-means: Differing Sizes

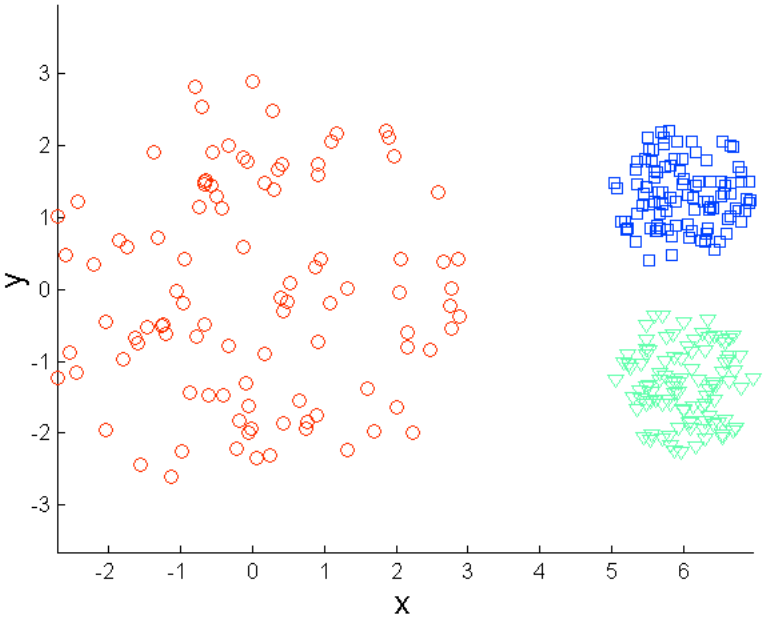


Original Points

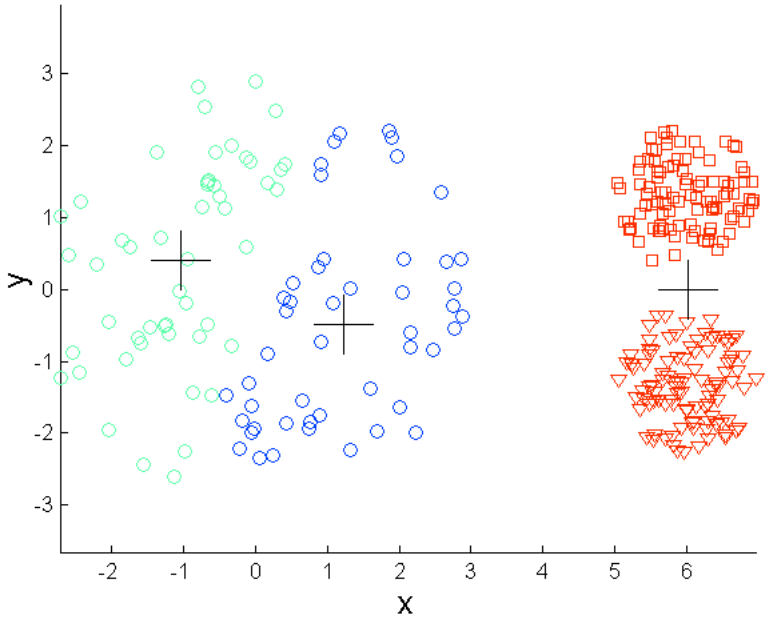


K-means (3 Clusters)

Limitations of K-means: Differing Density

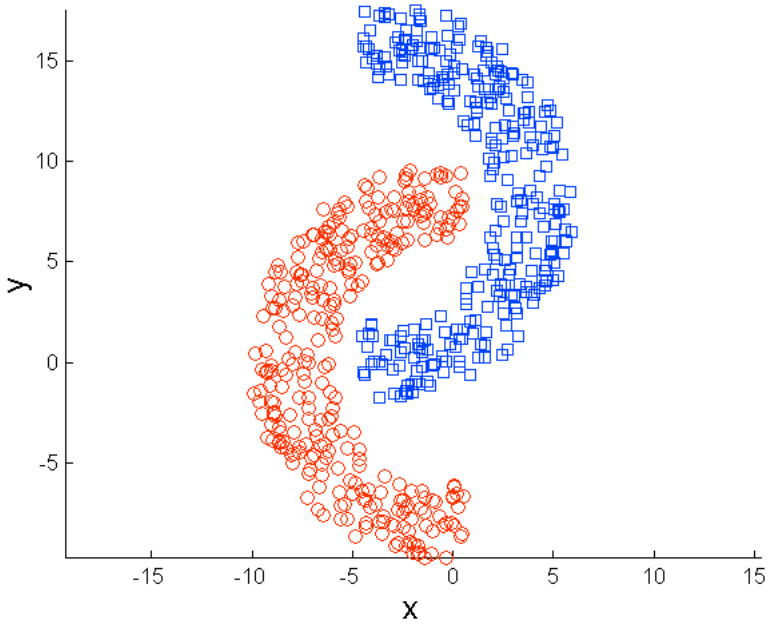


Original Points

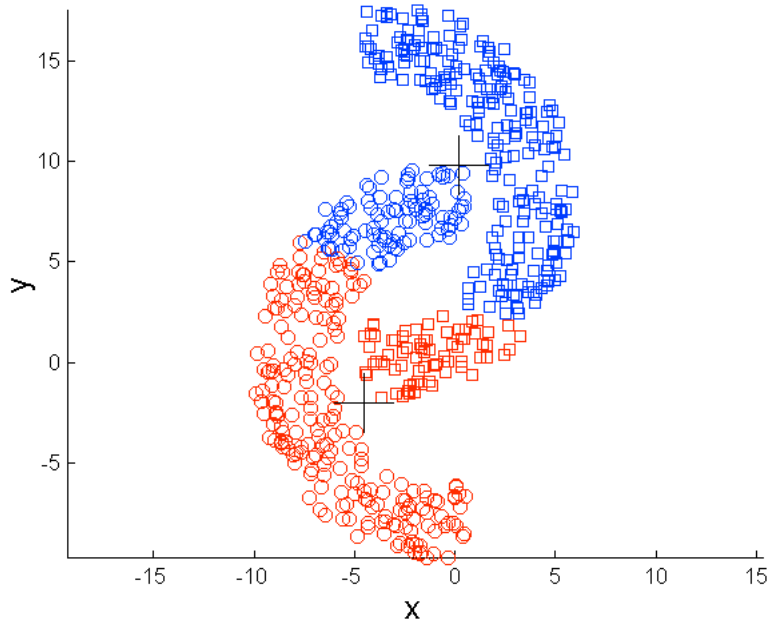


K-means (3 Clusters)

Limitations of K-means: Non-globular Shapes

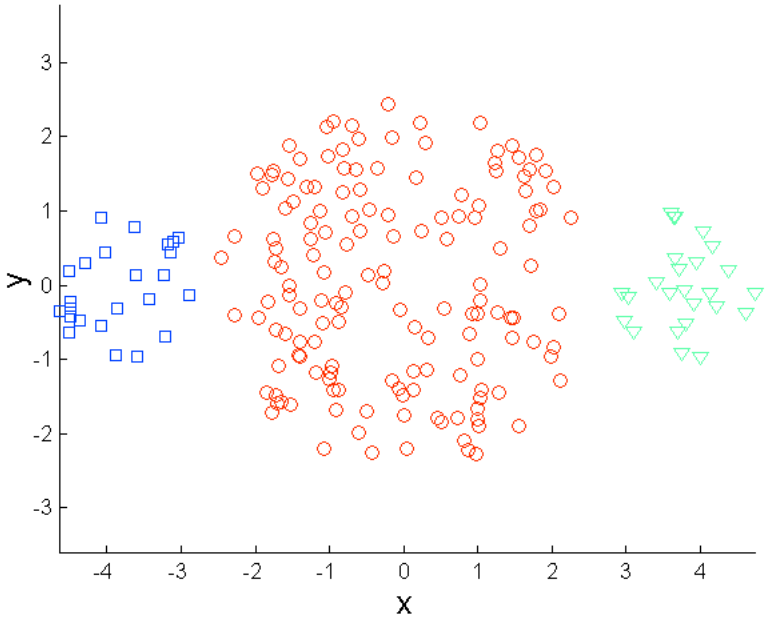


Original Points

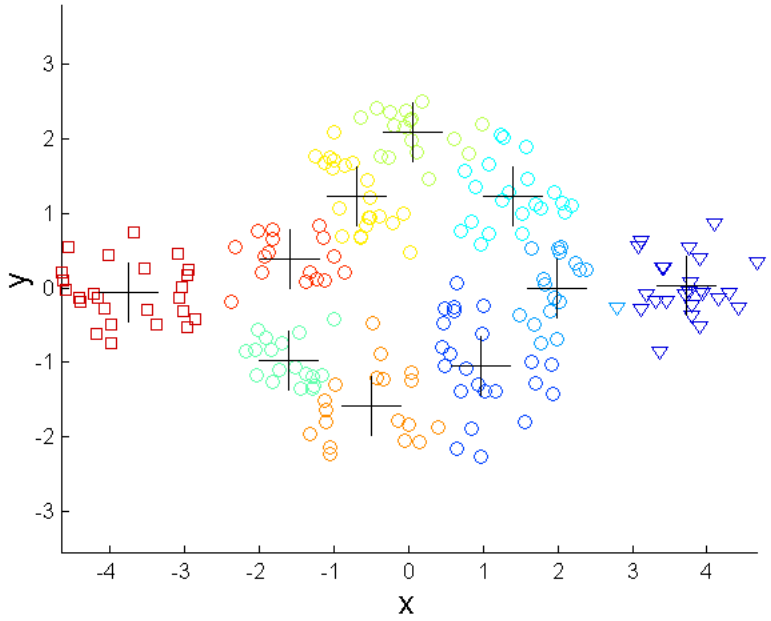


K-means (2 Clusters)

Overcoming K-means Limitations



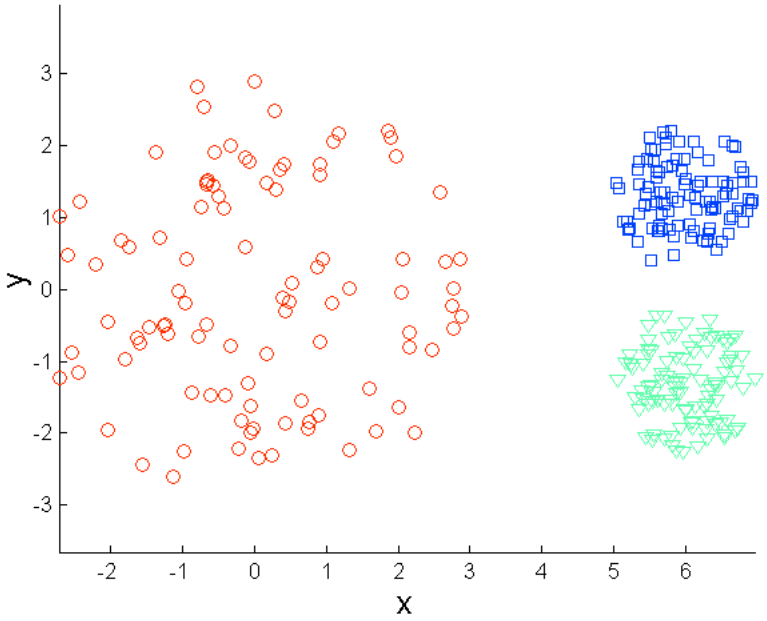
Original Points



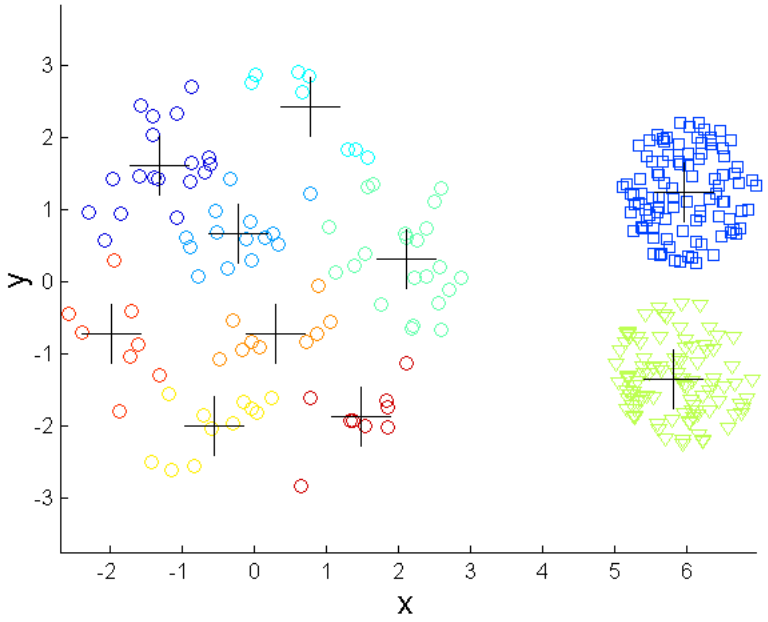
K-means Clusters

One solution is to use many clusters.
Find parts of clusters, but need to put together.

Overcoming K-means Limitations

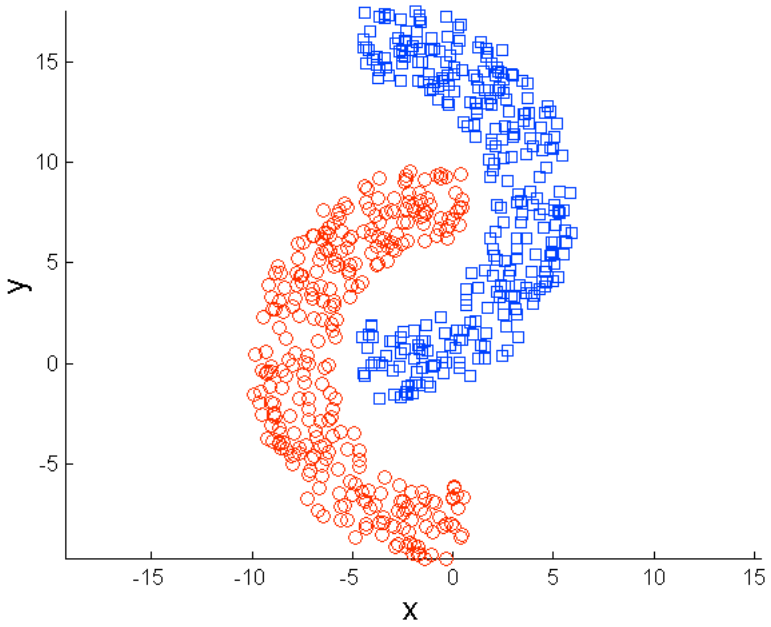


Original Points

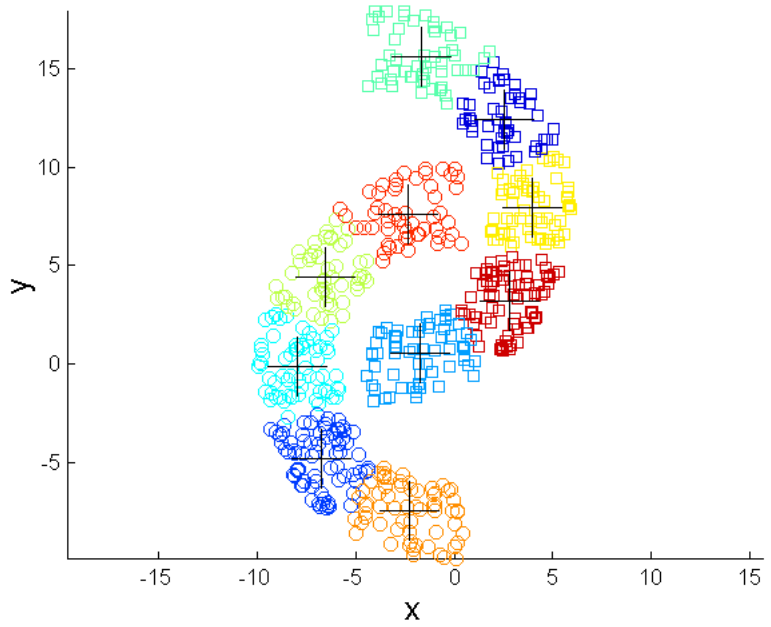


K-means Clusters

Overcoming K-means Limitations



Original Points



K-means Clusters

Pre-processing & 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

Acknowledgement & References

- Acknowledgement
  - Some slides are adapted from the K-means|| slides by Bahman Bahmani, Stanford University, 2012, and Tan, Steinbach, Kumar's slides for the book "Introduction to Data Mining"
- References
  - Chapter on clustering from a classic textbook (88 pages): https://www-users.cs.umn.edu/~kumar001/dmbook/ch7_clustering.pdf
  - K-means overview: https://en.wikipedia.org/wiki/K-means%2B%2B
  - K-means ++ paper: http://ilpubs.stanford.edu:8090/778/1/2006-13.pdf
  - K-means || paper: http://dl.acm.org/citation.cfm?doid=2180912.2180915
  - https://spark.apache.org/docs/2.3.2/api/scala/index.html#org.apache.**spark.ml**.clustering.KMeans
  - https://spark.apache.org/docs/2.3.2/api/scala/index.html#org.apache.**spark.mllib**.clustering.KMeans