Ex3: Spark

Objectives

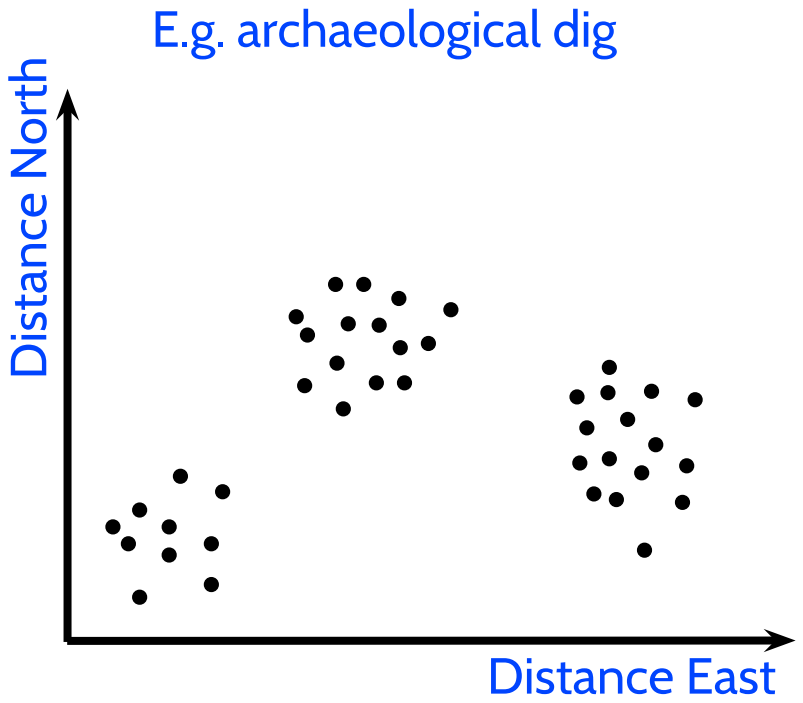
Learn the following RDD manipulation functions

- groupByKey
- map
- reduce
- collect
- takeSample

by implementing a k-means clustering algorithm

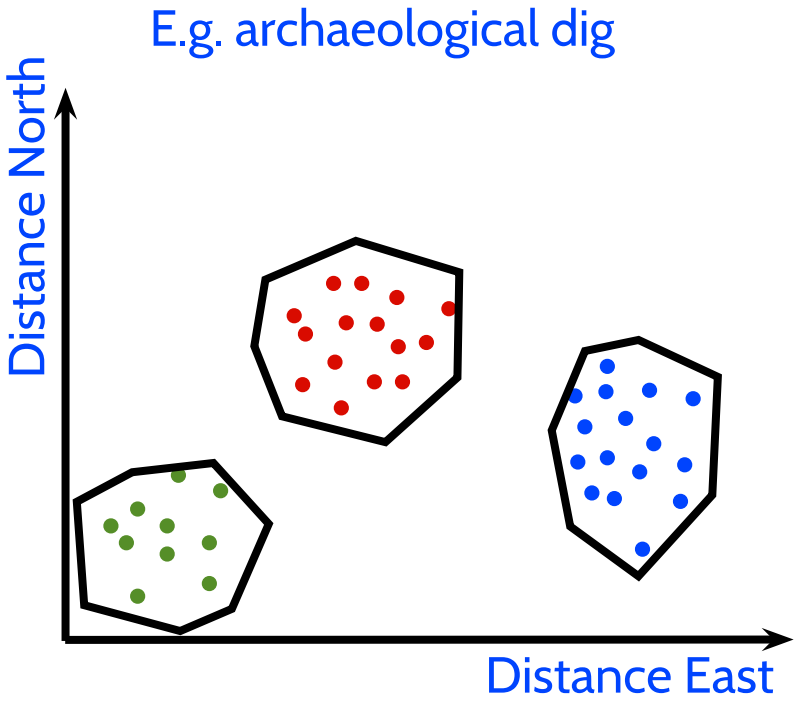
Clustering

Grouping data according to similarity



Clustering

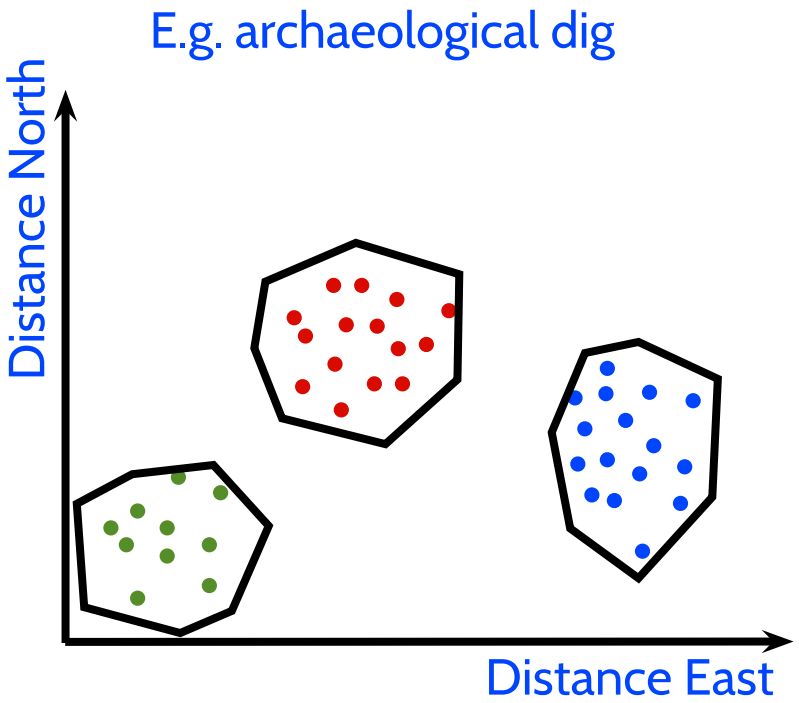
Grouping data according to similarity



K-Means Algorithm

Benefits

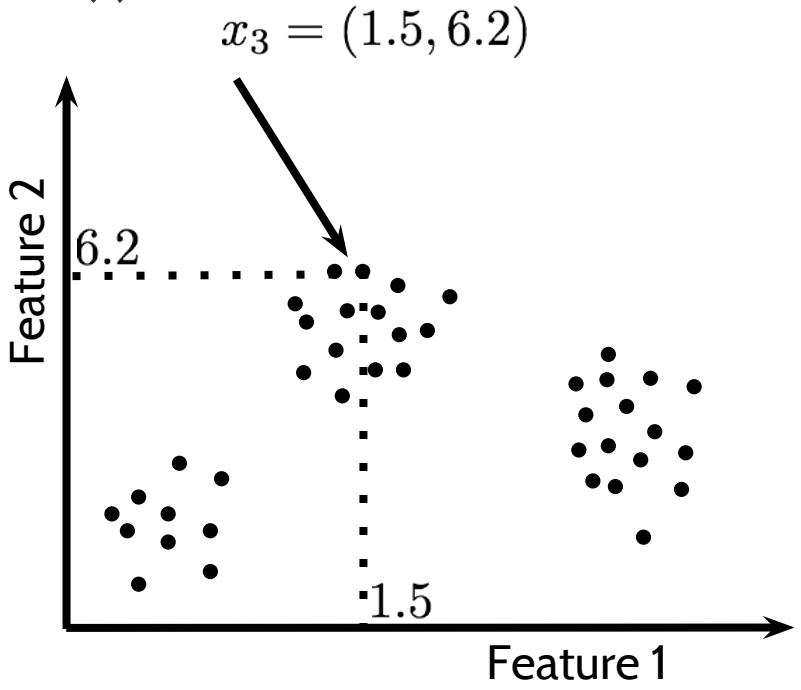
- Popular
- Fast
- Conceptually straightforward



K-Means: preliminaries

Data: Collection of values

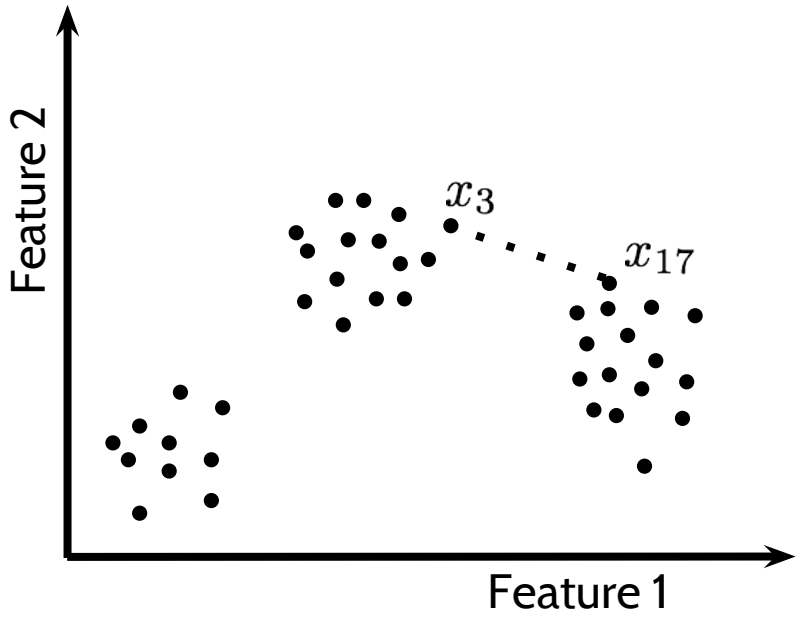
```
var data = lines.map(l => Vector.empty
  ++ l.split('\t').map(_.toDouble))
```



K-Means: preliminaries

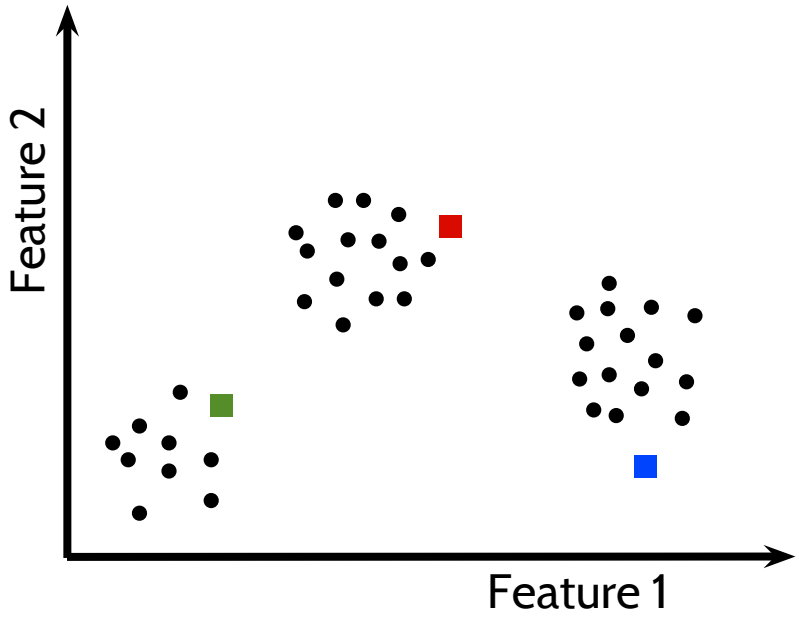
Dissimilarity:

```
math.sqrt(p.zip(q).map(pair =>
    math.pow((pair._1 - pair._2),2)).reduce(_+_));
```



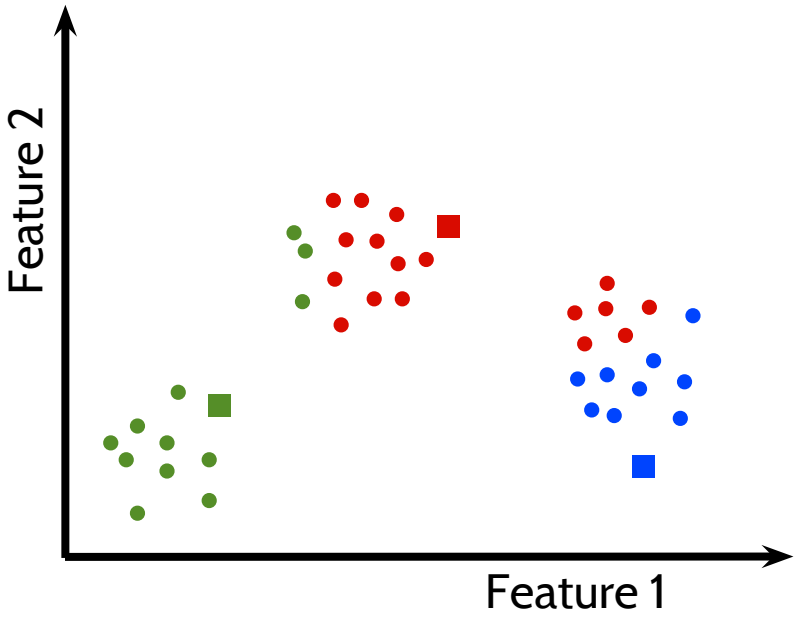
K-Means Algorithm

- **Initialize K cluster centers**
- Repeat until convergence:
  - Assign each data point to the cluster with the closest center.
  - Assign each cluster center to be the mean of its cluster's data points.



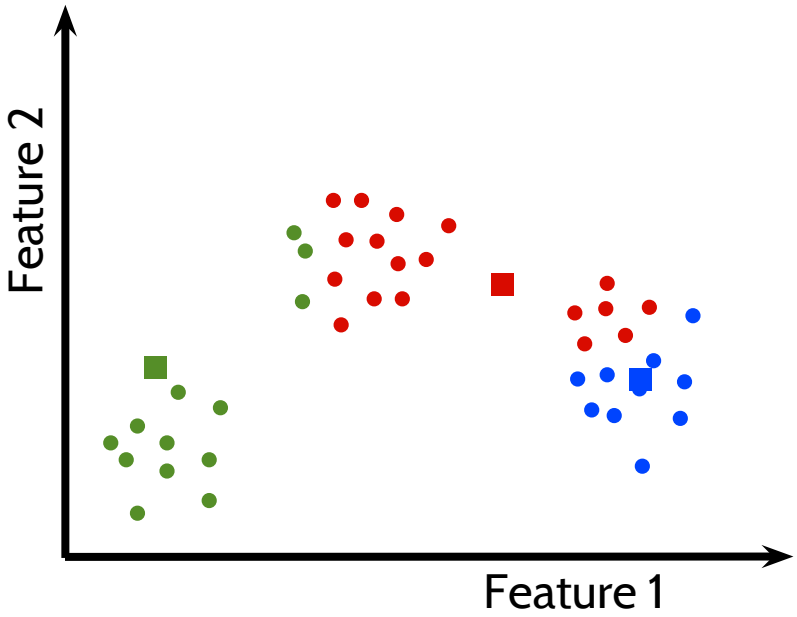
K-Means Algorithm

- Initialize K cluster centers
- Repeat until convergence:
  Assign each data point to the cluster with the closest center.
  Assign each cluster center to be the mean of its cluster's data points.



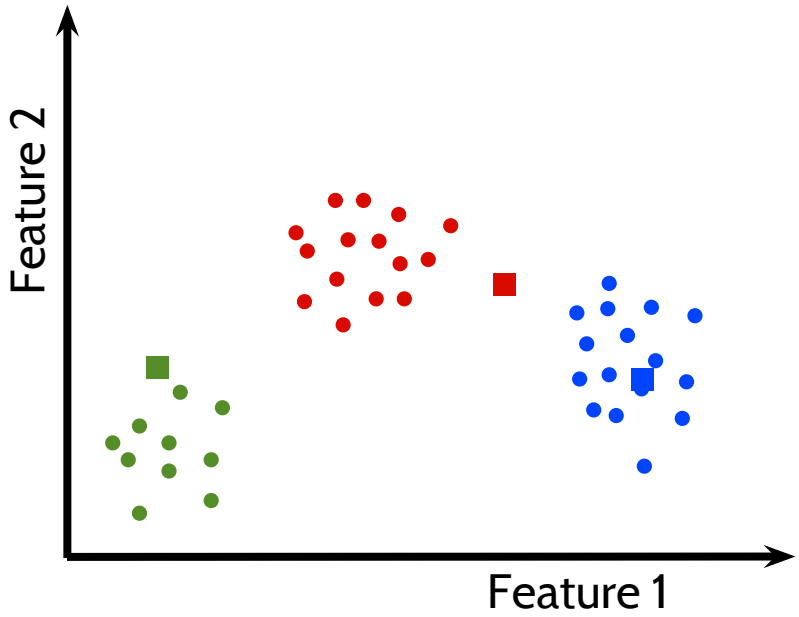
K-Means Algorithm

- Initialize K cluster centers
- Repeat until convergence:
  Assign each data point to the cluster with the closest center.
  Assign each cluster center to be the mean of its cluster's data points.



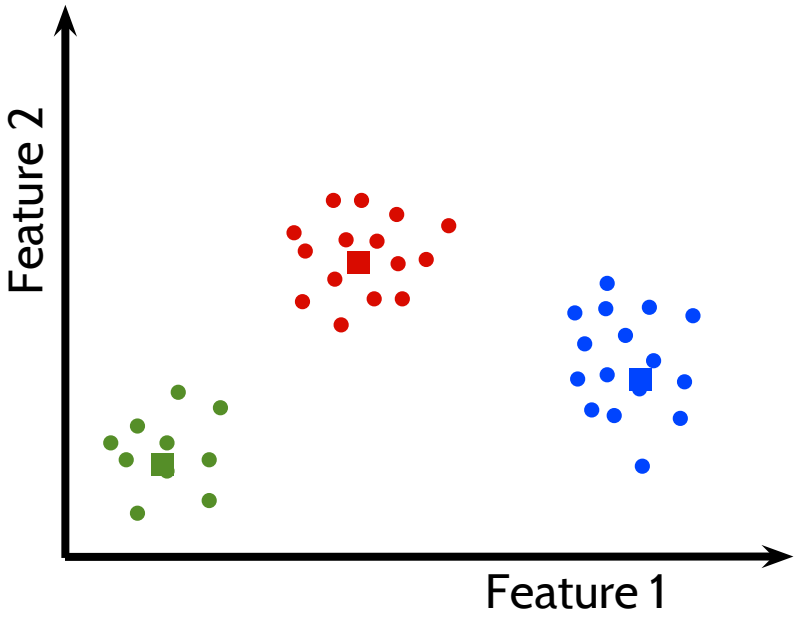
K-Means Algorithm

- Initialize K cluster centers
- Repeat until convergence:
  Assign each data point to the cluster with the closest center.
  Assign each cluster center to be the mean of its cluster's data points.



K-Means Source

- Initialize K cluster centers
- Repeat until convergence:
  Assign each data point to the cluster with the closest center.
  Assign each cluster center to be the mean of its cluster's data points.



Your Tasks

- On your VM
- Download a dataset from Dropbox
  - wget https://dl.dropboxusercontent.com/u/27408780/clustering_dataset.txt
- Get also a template from
  - wget https://dl.dropboxusercontent.com/u/27408780/kmeans.scala

Your Tasks

- [E] (60 points) Implement the following utility functions

```
def distance(p:Vector[Double], q:Vector[Double]) : Double = {
}

def clostestpoint(q: Vector[Double], candidates: Array[Vector[Double]]): Vector[Double] = {
}

def add_vec(v1: Vector[Double], v2: Vector[Double]): Vector[Double] = {
}

def average(cluster: Iterable[Vector[Double]]): Vector[Double] = {
}
```

Your Tasks

- [E] (40 Points) Implement the discussed k-means clustering algorithm on Spark based using those functions that you just implemented