04.02.2020

Statistical Methods in AI (CSE/ECE 471)

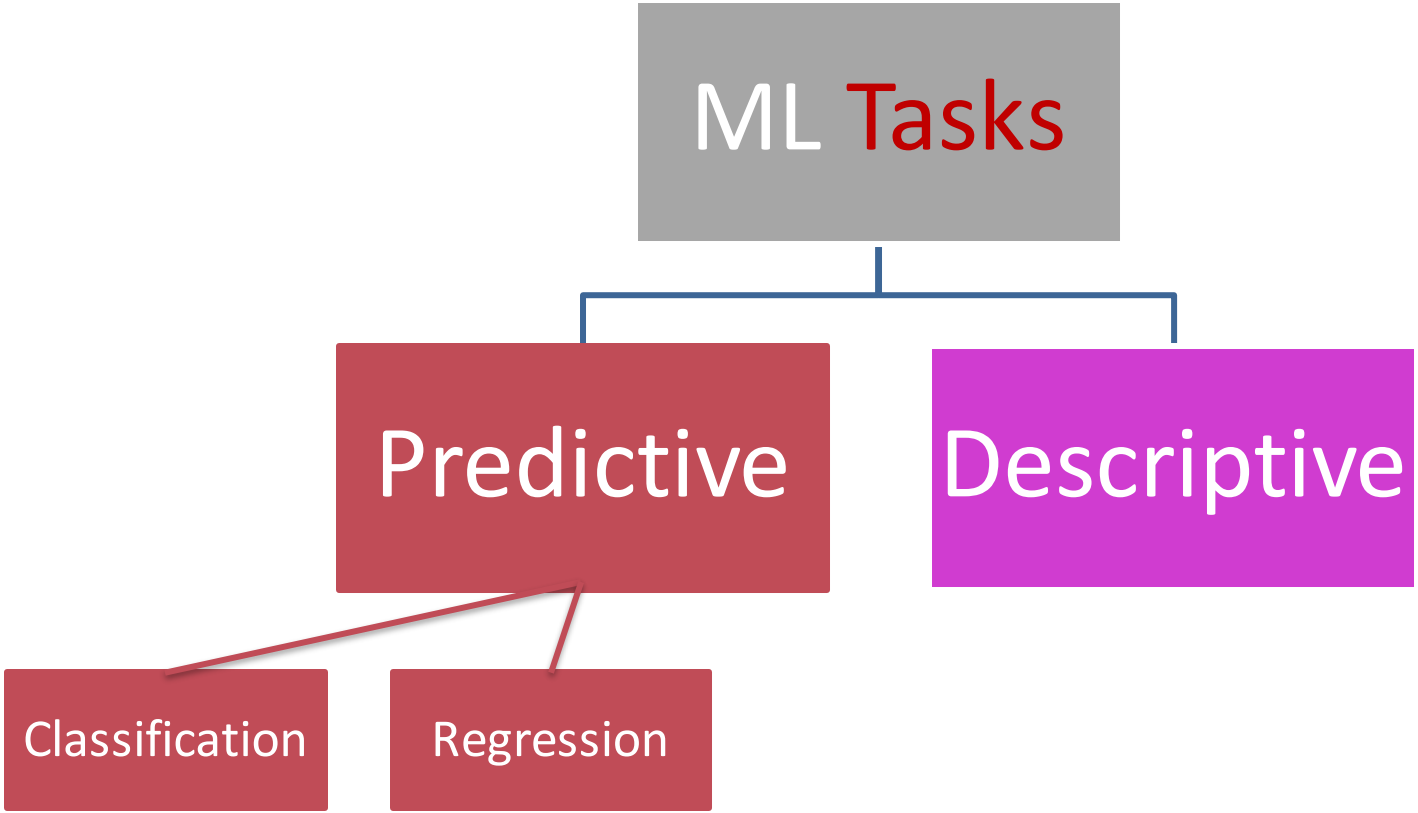
Lecture-9: Unsupervised Learning (k-means, GMM)

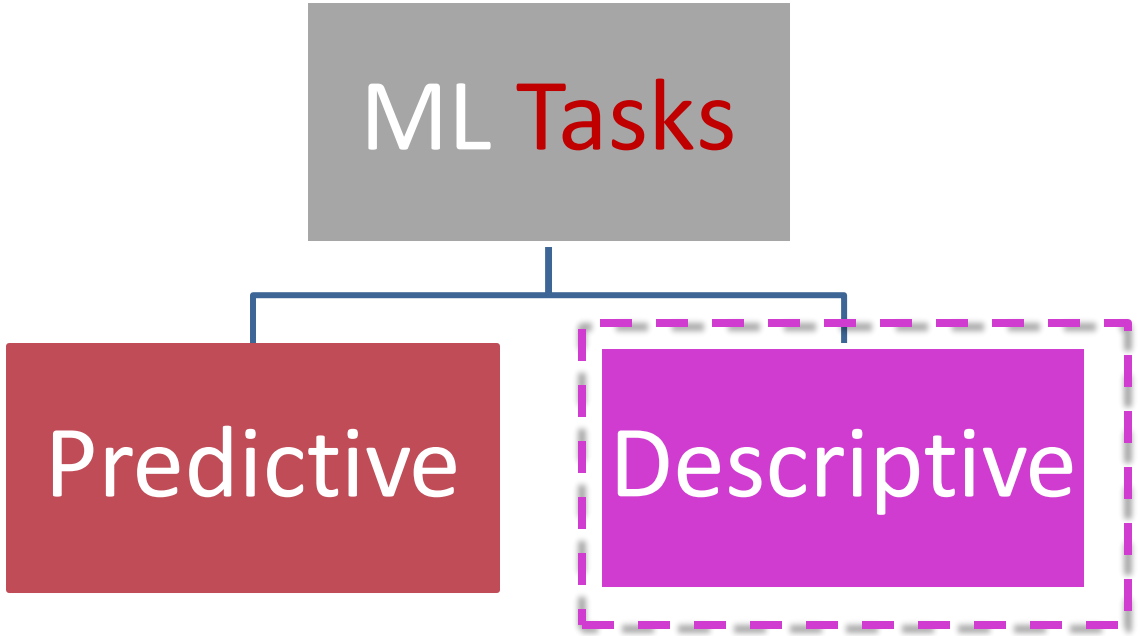
Ravi Kiran (ravi.kiran@iiit.ac.in)

Vineet Gandhi (v.gandhi@iiit.ac.in)

Center for Visual Information Technology (CVIT)

IIIT Hyderabad


ML Tasks
Predictive
Descriptive
Classification
Regression


ML Tasks
Predictive
Descriptive

ML::Tasks → Descriptive

- Study/Exploit the ‘structure’ of data
  - Clustering
  - Dimensionality Reduction
  - Density Estimation
- Also studied as ‘Unsupervised Learning’
  - ‘Input’ data without paired ‘Output’

Unsupervised Learning

Task: Given $X \in \mathcal{X}$, learn $f(X)$.

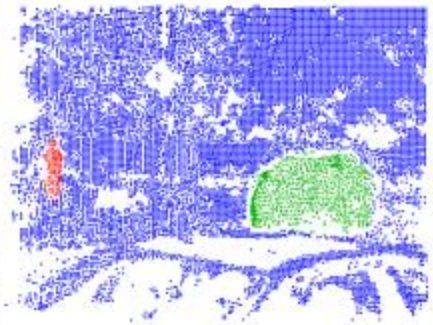
Unsupervised Learning → Clustering

Group similar things e.g. images

[Goldberger et al.]

- Determine groups of people in image above
  - based on clothing styles
  - gender, age, etc

features

- Determine moving objects in videos

Topic Modelling

Topics

gene	0.04
dna	0.02
genetic	0.01
. . .	

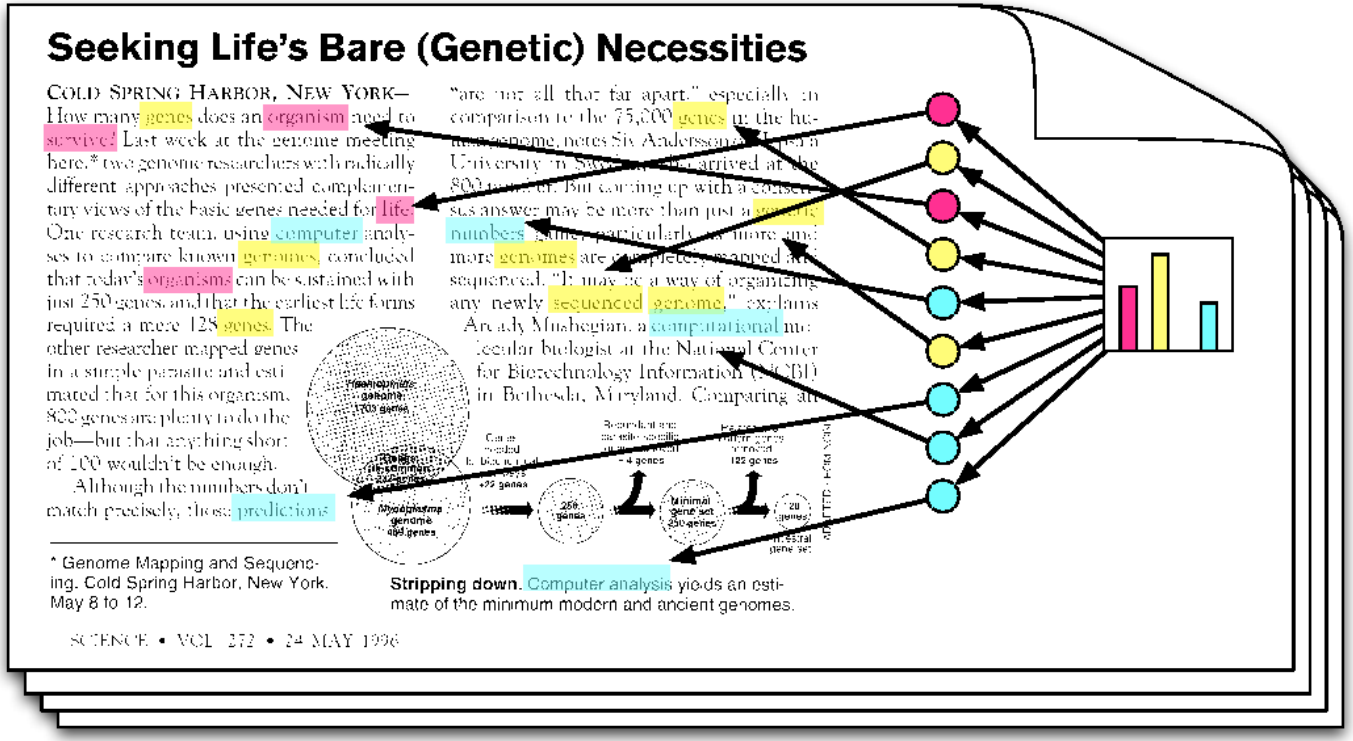
life	0.02
evolve	0.01
organism	0.01
. . .	

brain	0.04
neuron	0.02
nerve	0.01
. . .	

data	0.02
number	0.02
computer	0.01
. . .	

Documents

Topic proportions and assignments



Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 200 wouldn't be enough.

Although the numbers don't match precisely, those predictions "are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of the University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing all

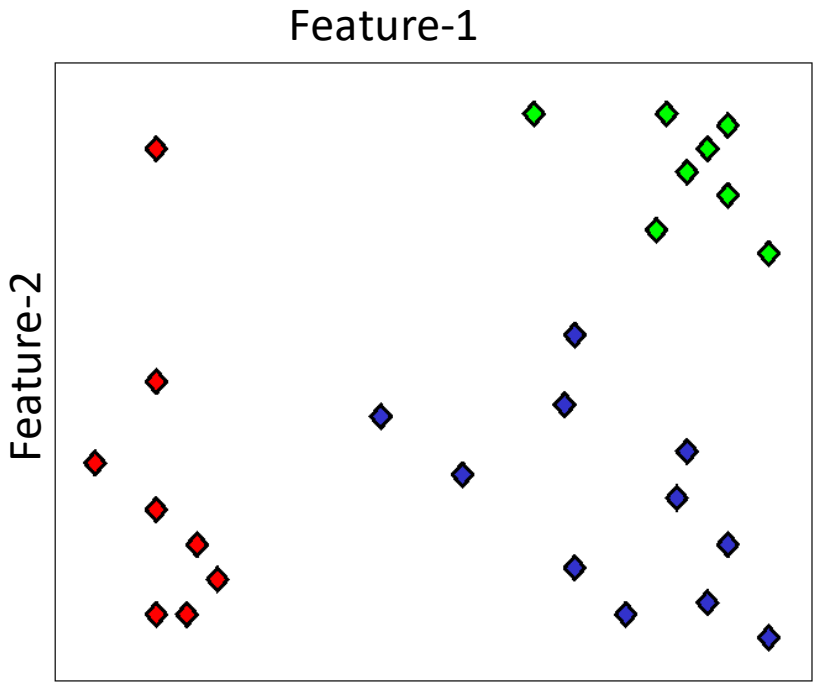
* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

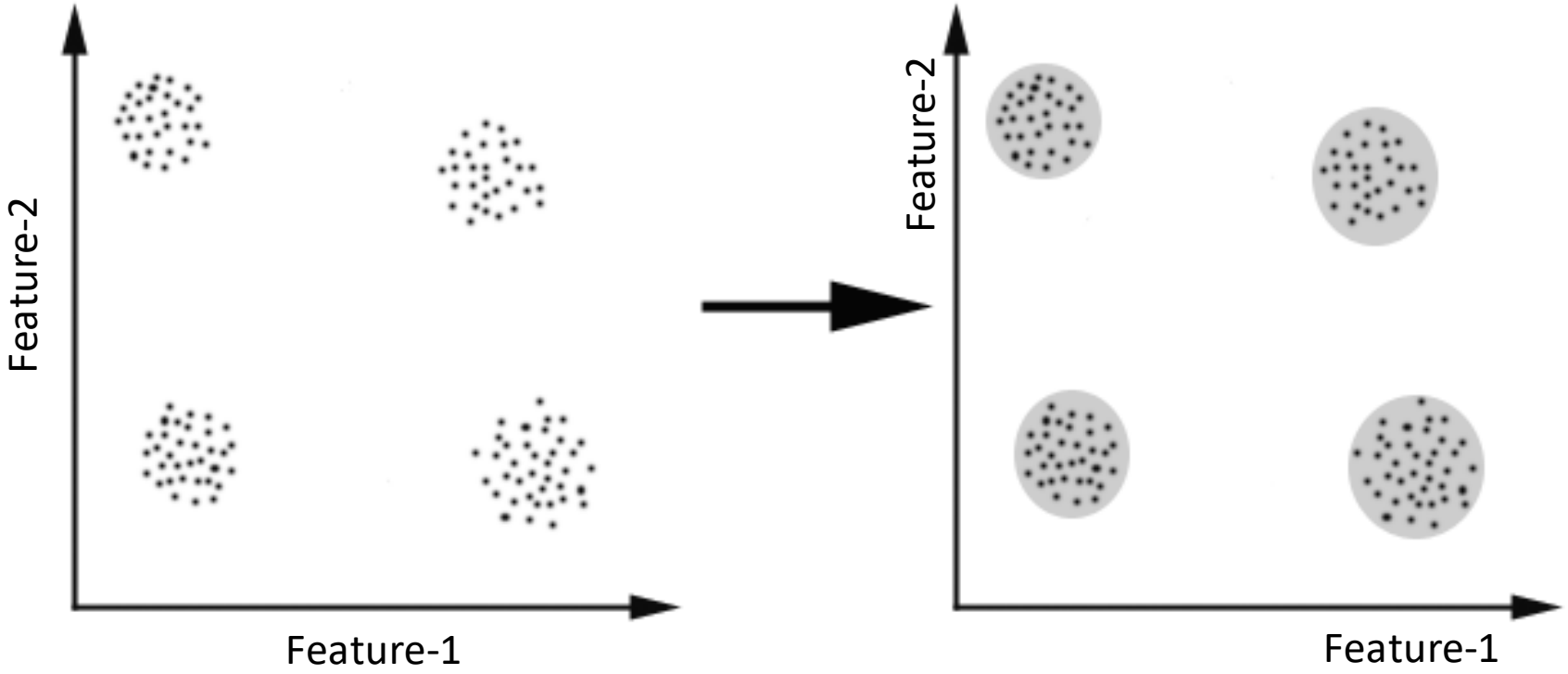
Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

SCIENCE • VOL. 272 • 24 MAY 1996

What is Clustering ?

- Organizing data into groups s.t.
  - High intra-group similarity (within members of a cluster)
  - Low inter-group similarity (across clusters)



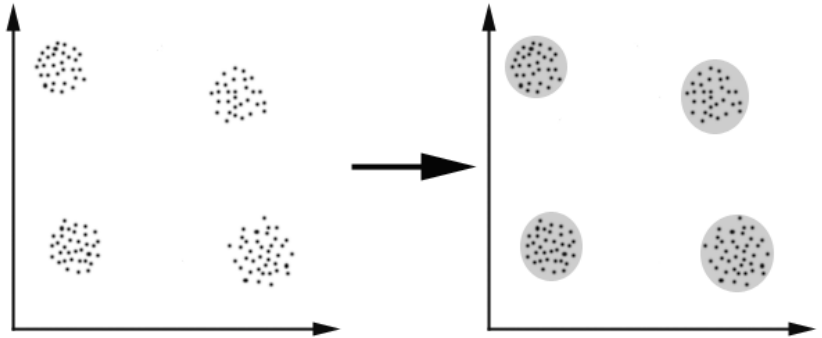

Feature-2
Feature-1
Feature-2
Feature-1

Approach-1: Assume that we (somehow) know the number of clusters

- # of clusters = K (often a ‘guess’)

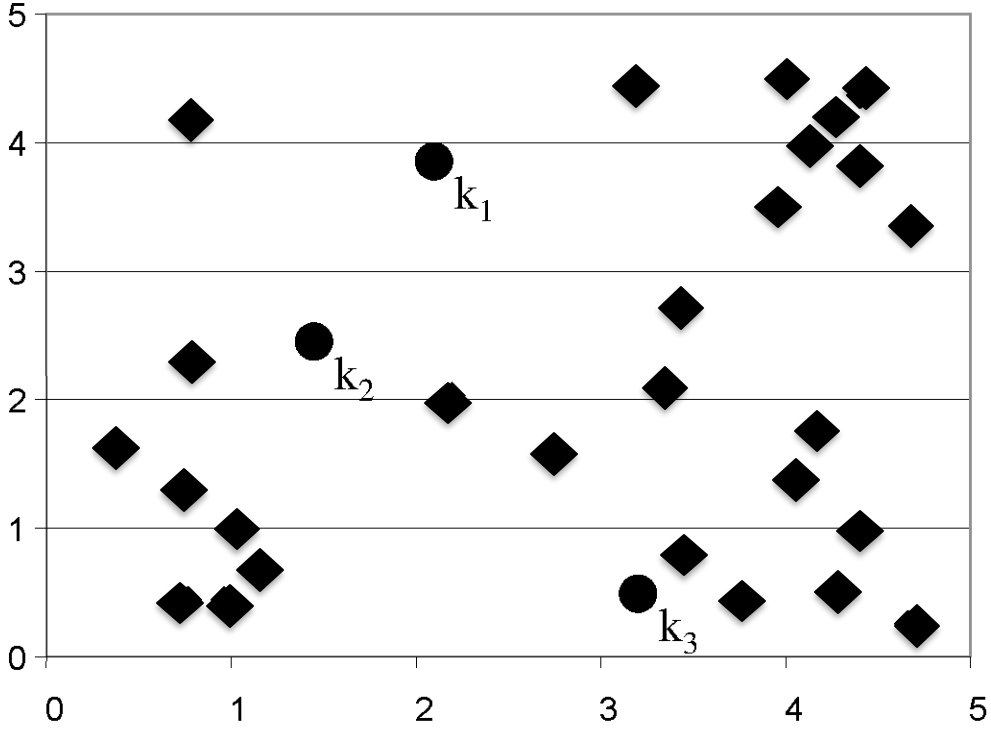
- How to represent a cluster ?
  - Option-1: Cluster members
  - Option-2: A suitable statistic of the cluster members

- Suitable statistic = Mean



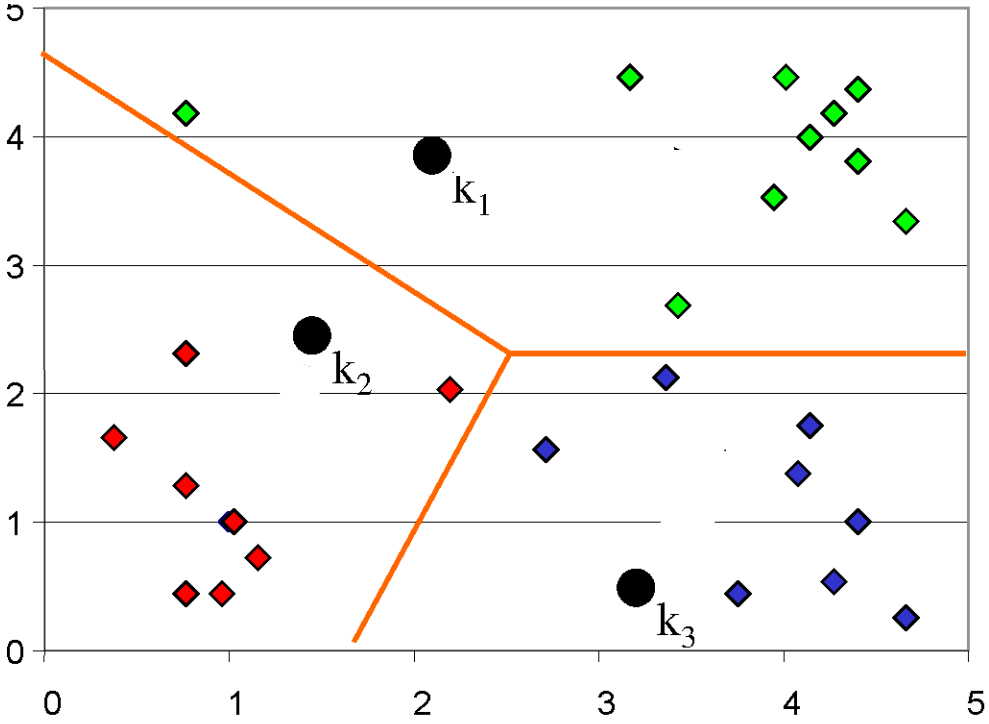
K-means Clustering: Initialization

Decide K, and initialize K centers (randomly)



K-means Clustering: Iteration 1

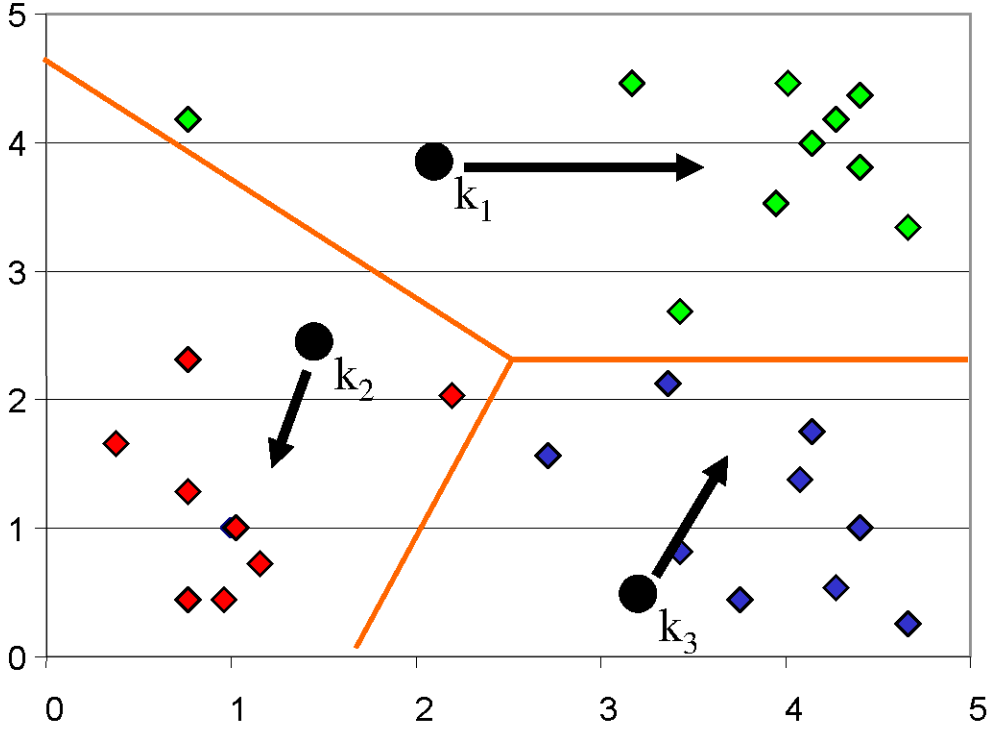
Assign all objects to the nearest center.



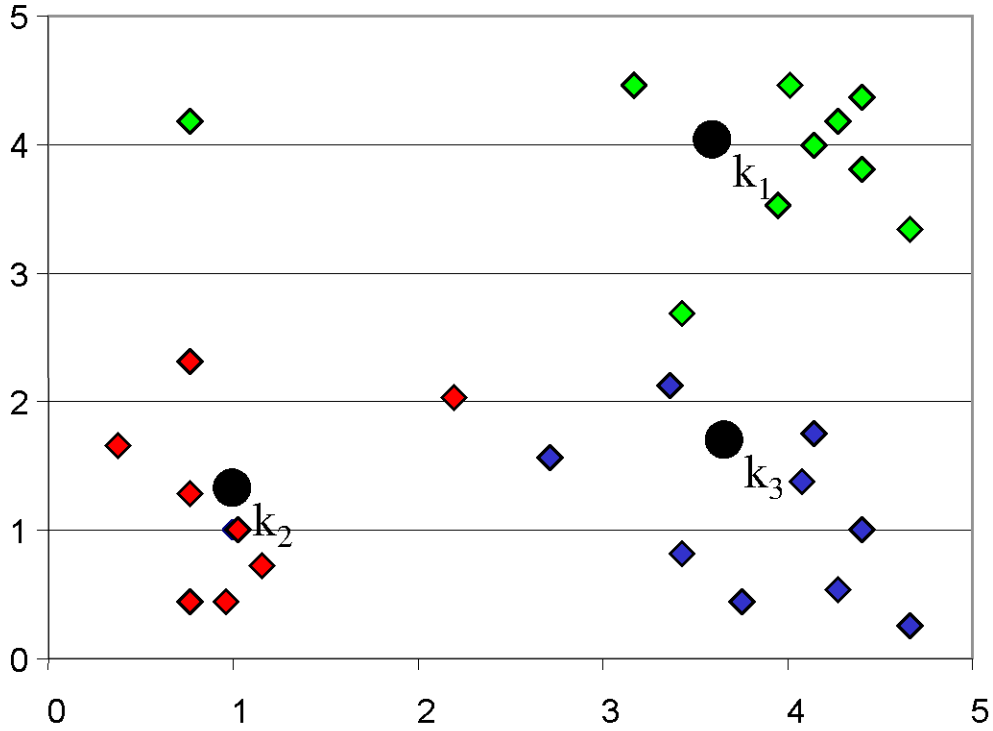
K-means Clustering: Iteration 1

Assign all objects to the nearest center.

Move a center to the mean of its members.

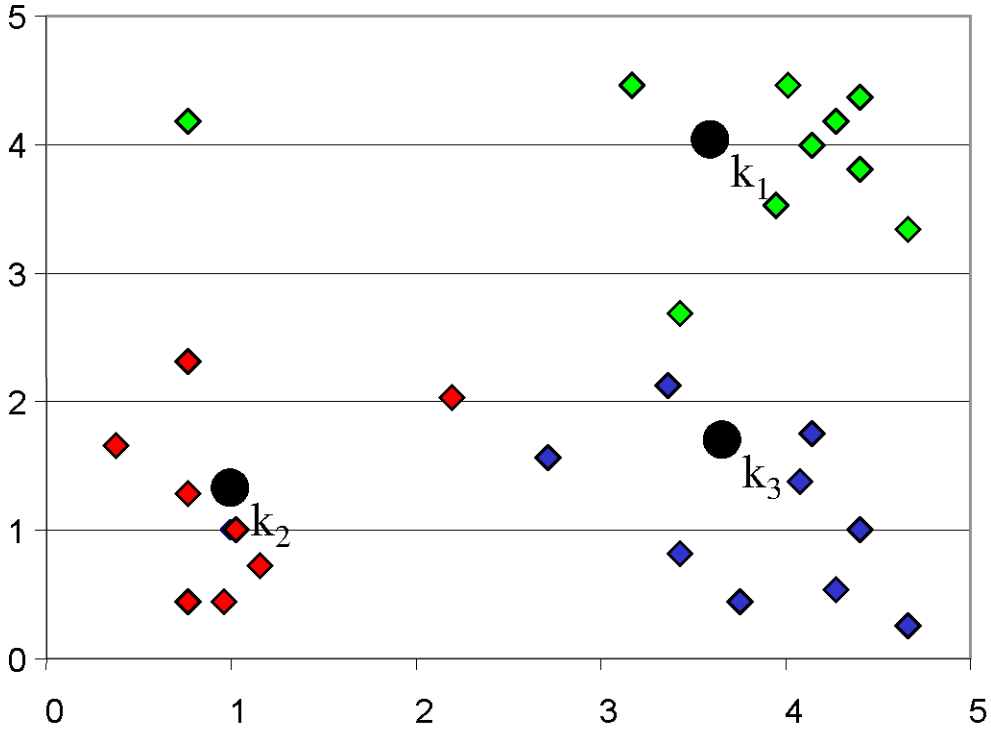


K-means Clustering: Iteration 2



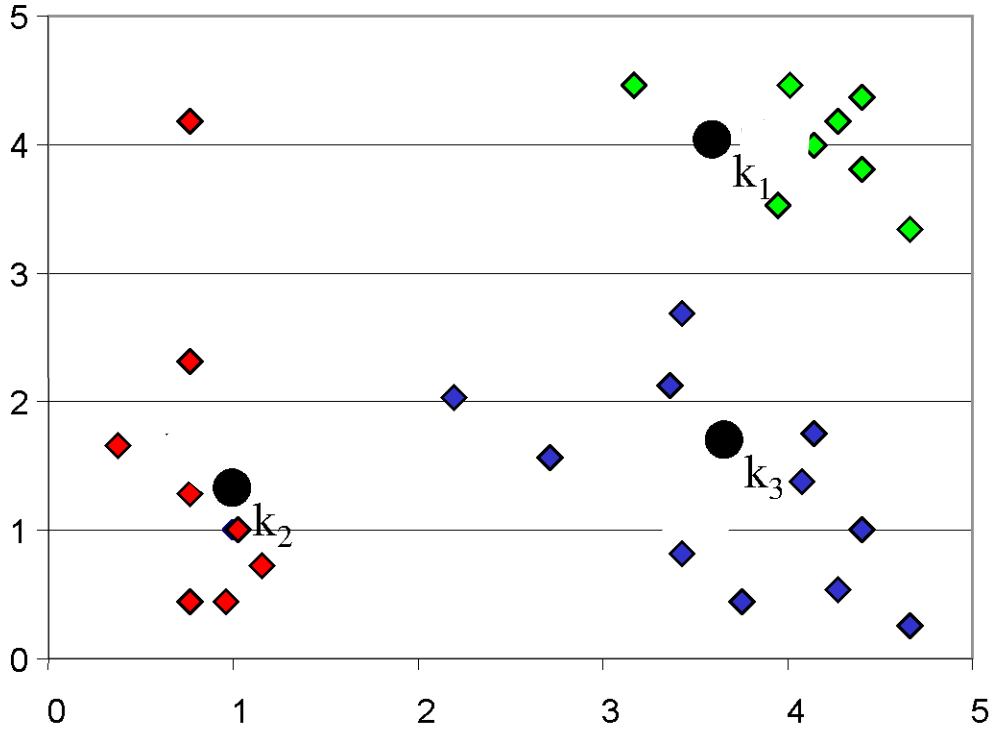
K-means Clustering: Iteration 2

After moving centers, re-assign the objects…



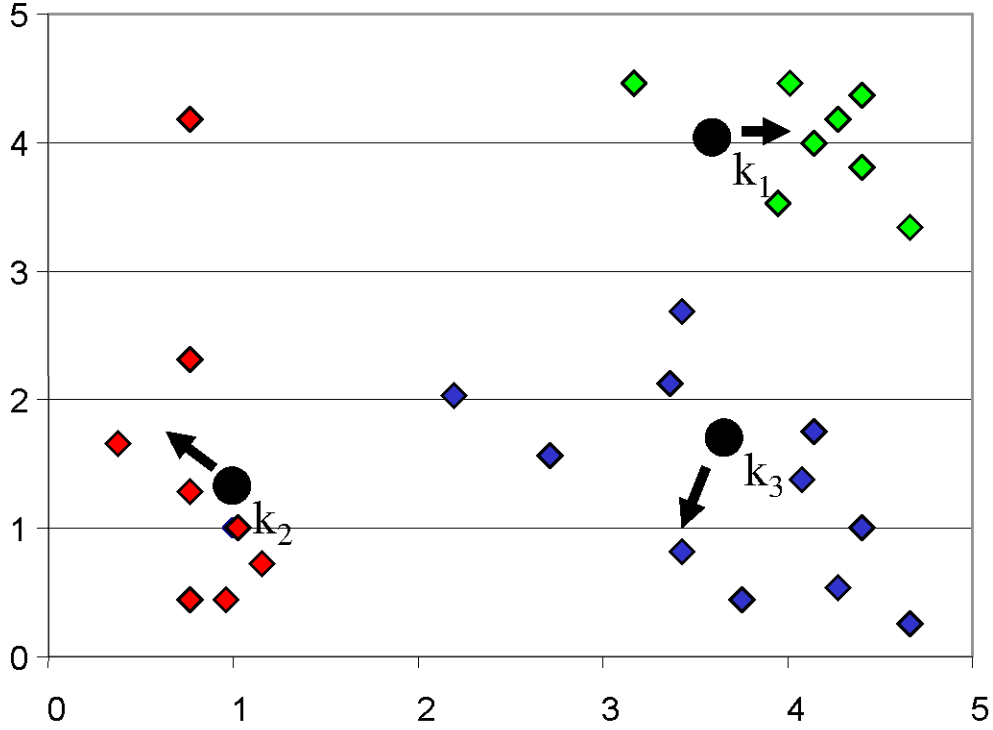
K-means Clustering: Iteration 2

After moving centers, re-assign the objects to nearest centers.



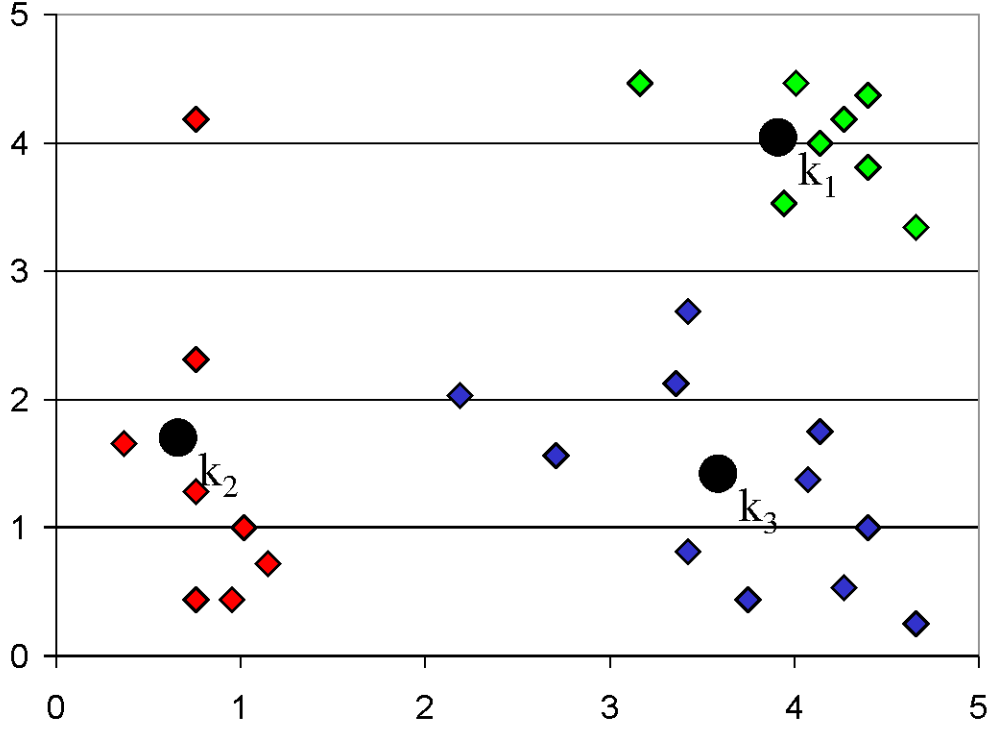
K-means Clustering: Iteration 2

After moving centers, re-assign the objects to nearest centers.
Move a center to the mean of its new members.

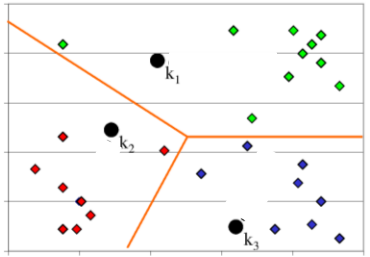


K-means Clustering: Finished!

Re-assign and move centers, until …
no objects changed membership.



$\{x^{(1)}, \ldots, x^{(m)}\}$ $\quad x^{(i)} \in \mathbb{R}^n$



The k-means clustering algorithm is as follows:

1. Initialize **cluster centroids** $\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}^n$ randomly.
2. Repeat until convergence: {

   For every i, set

$$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2.$$

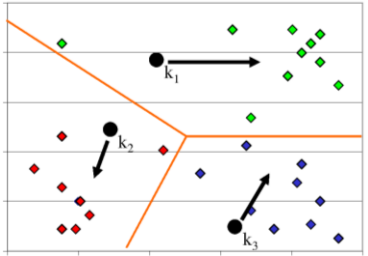
   For each j, set

$$\mu_j := \frac{\sum_{i=1}^m 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^m 1\{c^{(i)} = j\}}.$$

   }

Assignment step: Assign each data point to the closest cluster

Refitting step: Move each cluster center to the center of the data assigned to it



$\{x^{(1)}, \ldots, x^{(m)}\}$ $\quad x^{(i)} \in \mathbb{R}^n$

The k-means clustering algorithm is as follows:

1. Initialize **cluster centroids** $\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}^n$ randomly.
2. Repeat until convergence: {

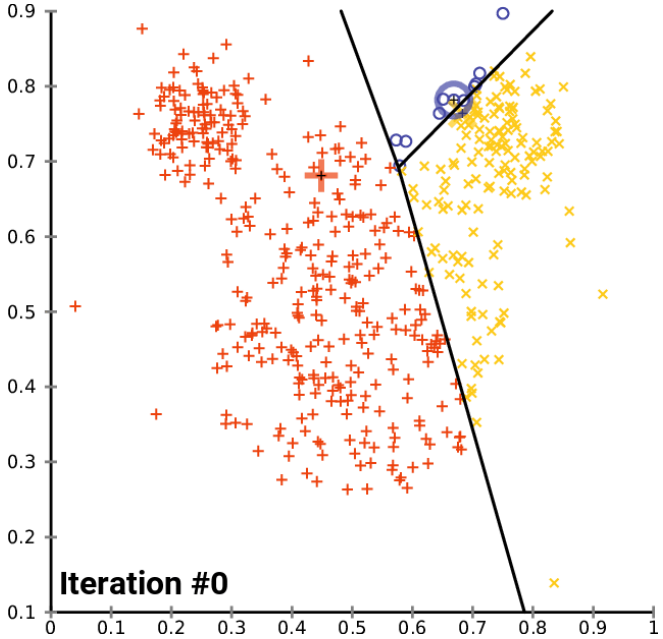
   For every i, set

   $$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2.$$

   For each j, set

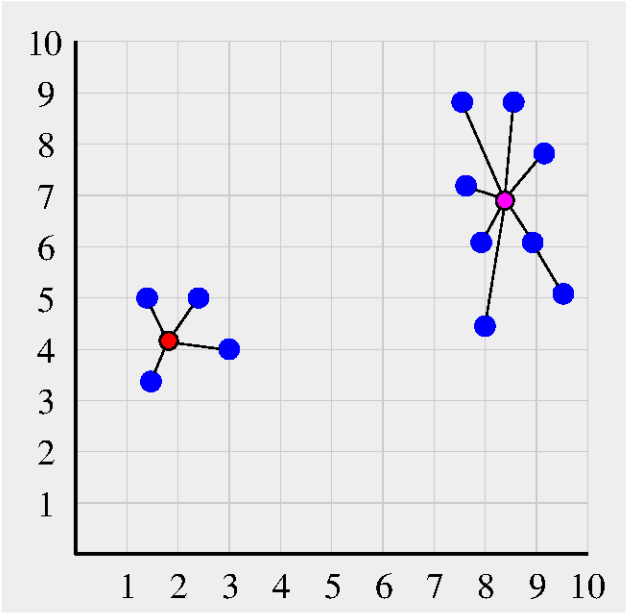
   $$\mu_j := \frac{\sum_{i=1}^m 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^m 1\{c^{(i)} = j\}}.$$

   }



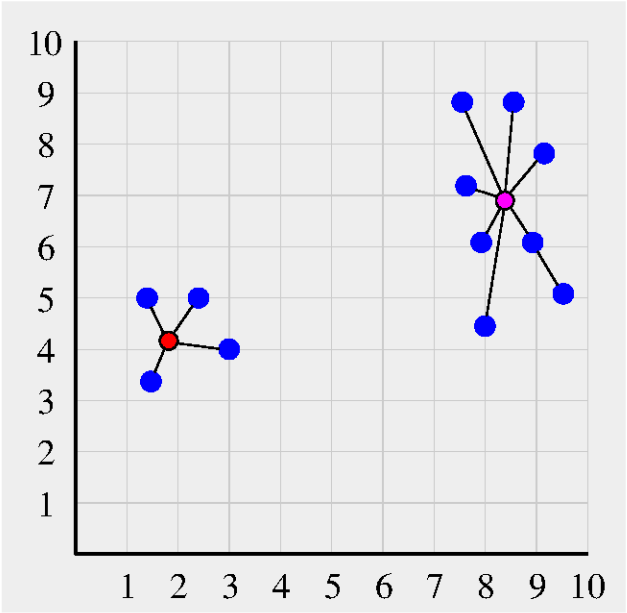
Why K-means Works

- What is a good partition?
- High intra-cluster similarity



Why K-means Works

- What is a good partition?
- High intra-cluster similarity
- K-means optimizes



$$J = \sum_{k=1}^{K} \sum_{i=1}^{n_k} \left\| x_{ki} - \mu_k \right\|^2$$

Why does K-means work ?

Repeat until convergence: {

For every i, set

$$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2.$$

For each j, set

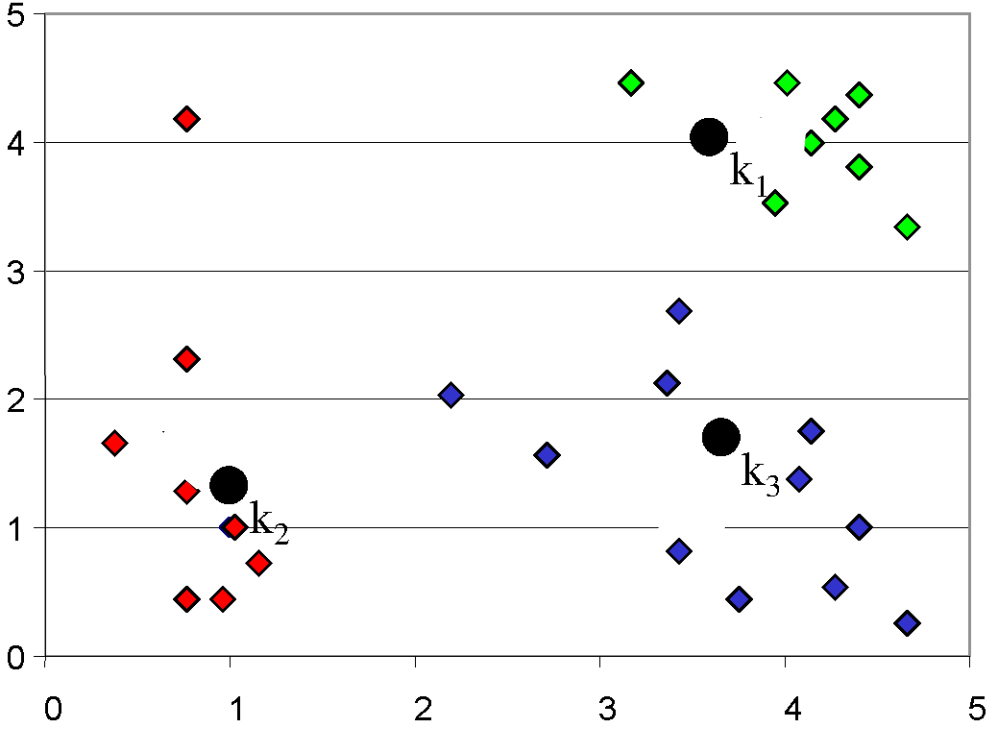
$$\mu_j := \frac{\sum_{i=1}^{m} 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{m} 1\{c^{(i)} = j\}}.$$

}

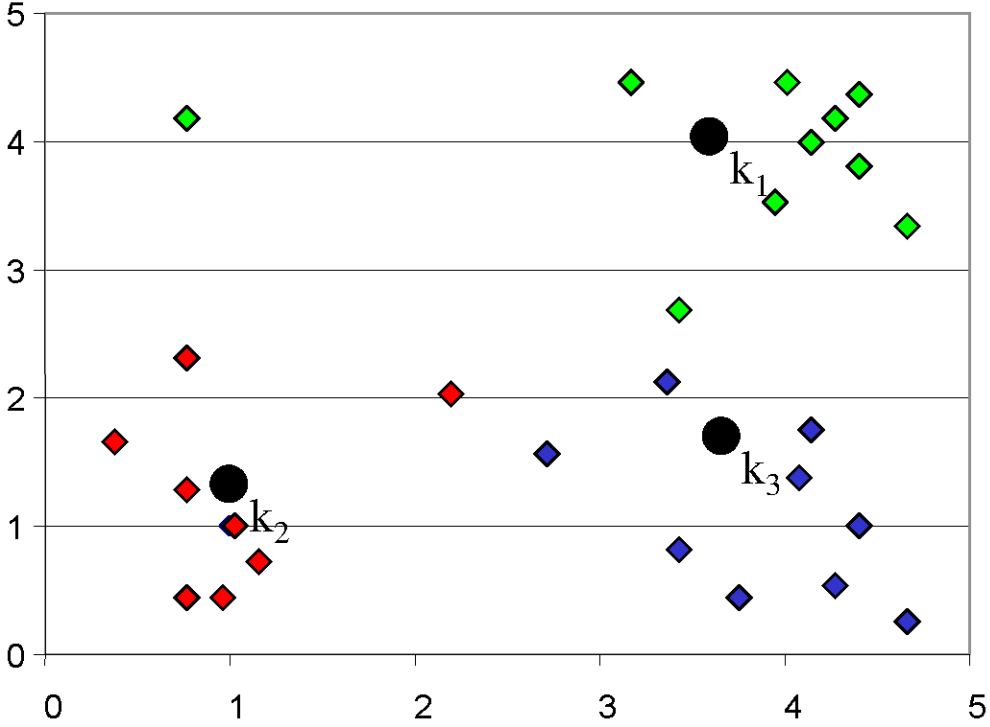
$$J = \sum_{k=1}^{K} \sum_{i=1}^{n_k} \left\| x_{ki} - \mu_k \right\|^2$$

- Whenever an assignment is changed, the sum squared distances J of data points from their assigned cluster centers is reduced.

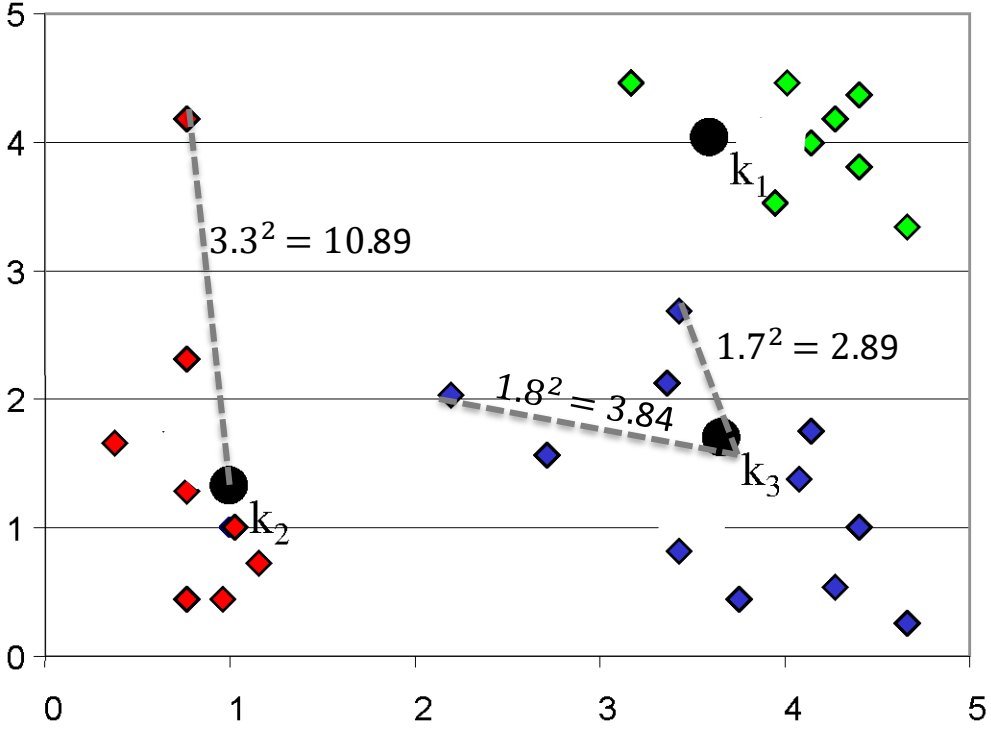
$$J = \sum_{k=1}^{K} \sum_{i=1}^{n_k} \left\| x_{ki} - \mu_k \right\|^2$$



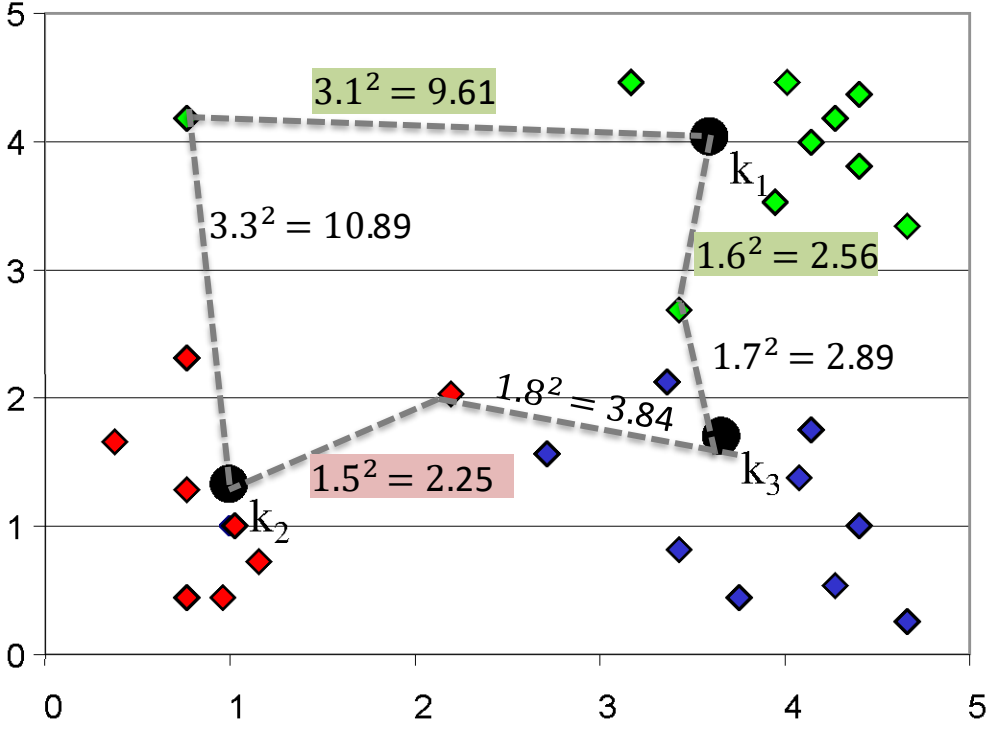
$$J = \sum_{k=1}^{K} \sum_{i=1}^{n_k} \left\| x_{ki} - \mu_k \right\|^2$$



$$J = \sum_{k=1}^{K} \sum_{i=1}^{n_k} \left\| x_{ki} - \mu_k \right\|^2$$



$$J = \sum_{k=1}^{K} \sum_{i=1}^{n_k} \left\| x_{ki} - \mu_k \right\|^2$$



Why does K-means work ?

Repeat until convergence: {

For every i, set

$$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2.$$

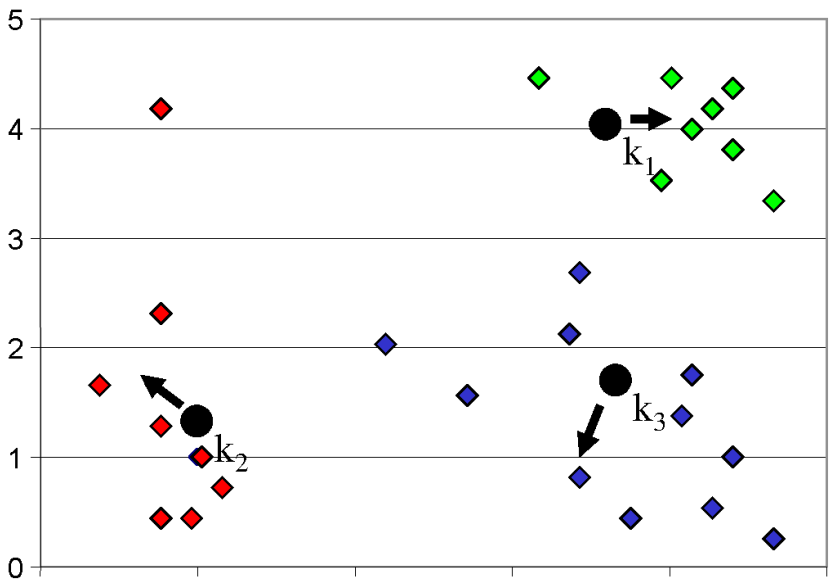
For each j, set

$$\mu_j := \frac{\sum_{i=1}^{m} 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{m} 1\{c^{(i)} = j\}}.$$

}

$$J = \sum_{k=1}^{K} \sum_{i=1}^{n_k} \left\| x_{ki} - \mu_k \right\|^2$$

- Whenever an assignment is changed, the sum squared distances J of data points from their assigned cluster centers is reduced.
- Whenever a cluster center is moved, J is reduced.



Repeat until convergence: {

For every i, set

$$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2.$$

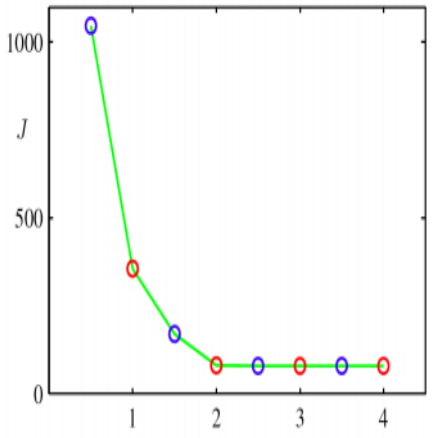
For each j, set

$$\mu_j := \frac{\sum_{i=1}^{m} 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{m} 1\{c^{(i)} = j\}}.$$

}

$$J = \sum_{k=1}^{K} \sum_{i=1}^{n_k} \left\| x_{ki} - \mu_k \right\|^2$$

- Whenever an assignment is changed, the sum squared distances J of data points from their assigned cluster centers is reduced.
- Whenever a cluster center is moved, J is reduced.
- Test for convergence: If the assignments do not change in the assignment step, we have converged (to at least a local minimum).



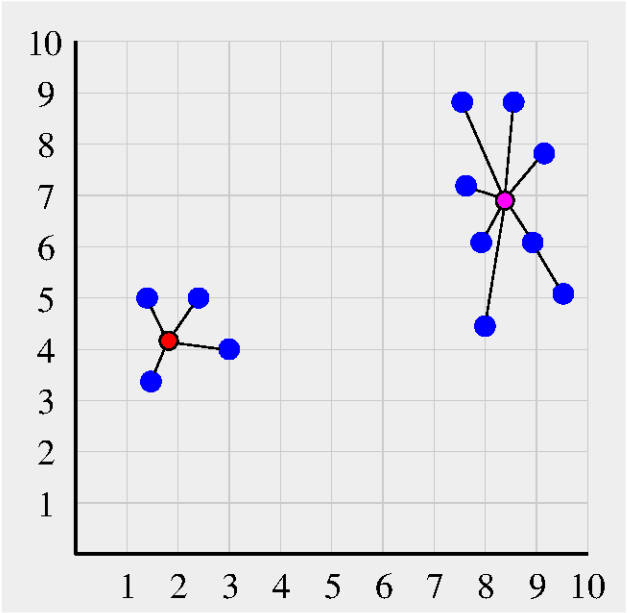
Why K-means Works

- What is a good partition?
- High intra-cluster similarity
- K-means optimizes
  - the average distance to members of the same cluster

$$\sum_{k=1}^{K} \frac{1}{n_k} \sum_{i=1}^{n_k} \sum_{j=1}^{n_k} \left\| x_{ki} - x_{kj} \right\|^2$$

  - which is twice the total distance to centers,

$$J = \sum_{k=1}^{K} \sum_{i=1}^{n_k} \left\| x_{ki} - \mu_k \right\|^2$$



Algorithm *k-means*

1. Decide on a value for *K*, th

2. Initialize the *K* cluster cent
necessary).

3. Decide the class memberships of the *N* objects by assigning them to the nearest cluster center.

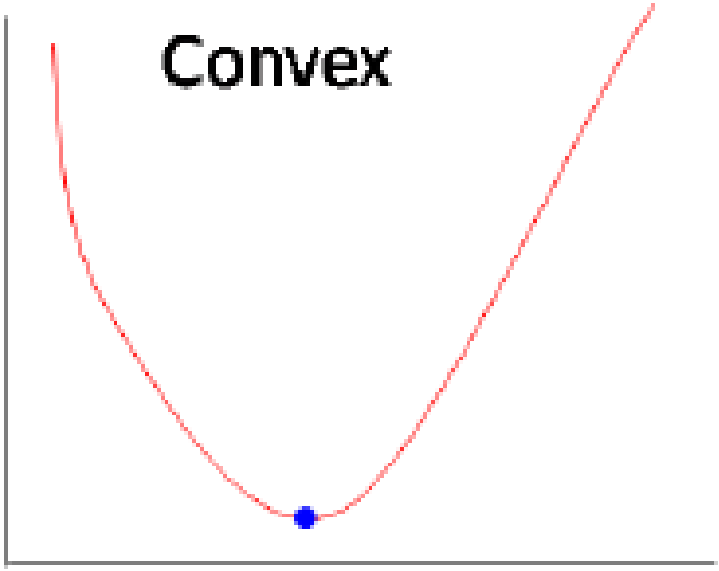
4. Re-estimate the *K* cluster centers, by assuming the memberships found above are correct.

5. Repeat 3 and 4 until none of the *N* objects changed membership in the last iteration

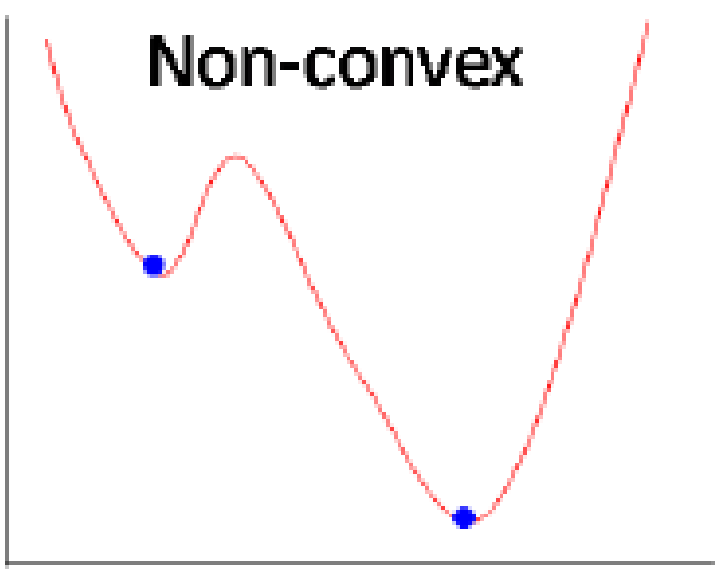
Use one of the distance / similarity functions we discussed earlier

Average / median of class members

Convex and Nonconvex functions

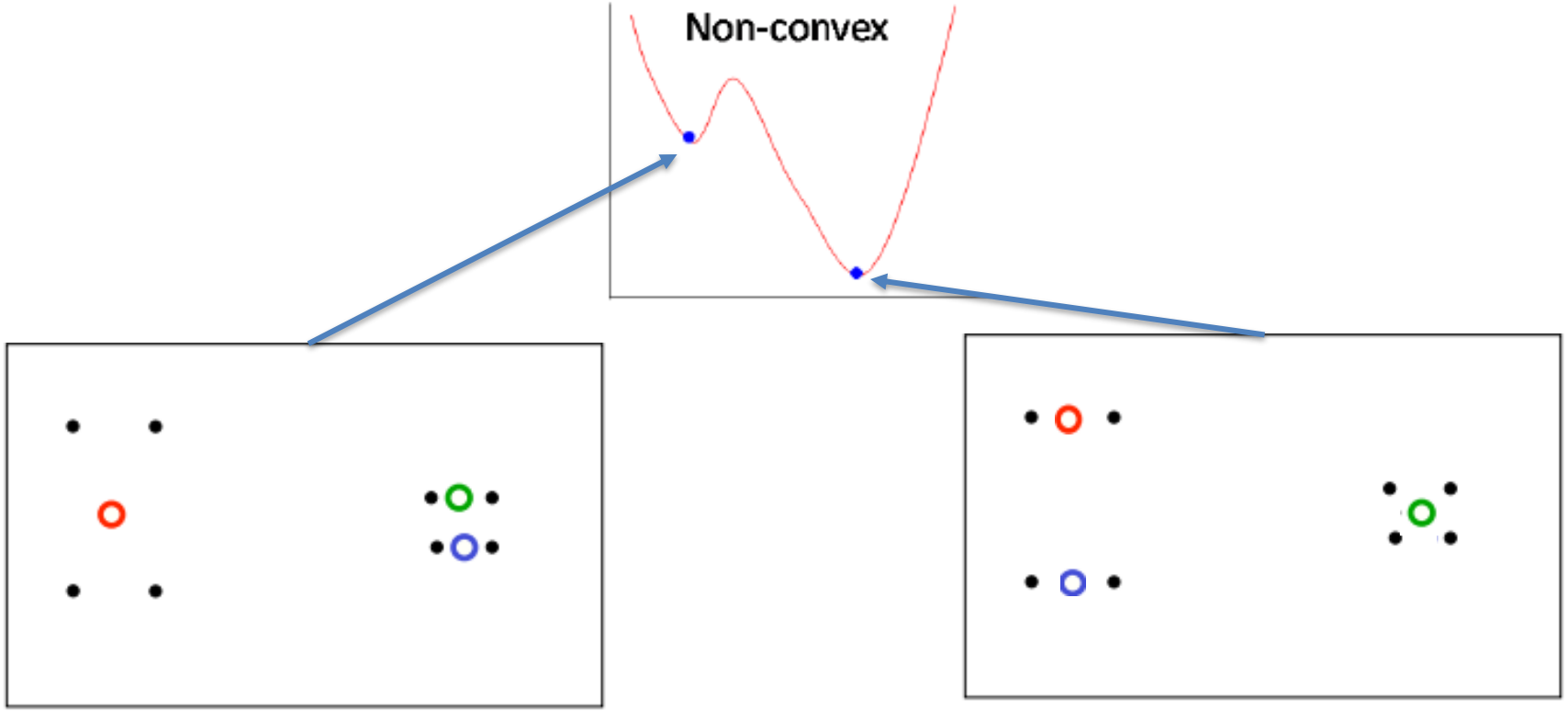


Unique minimum



Multiple minima

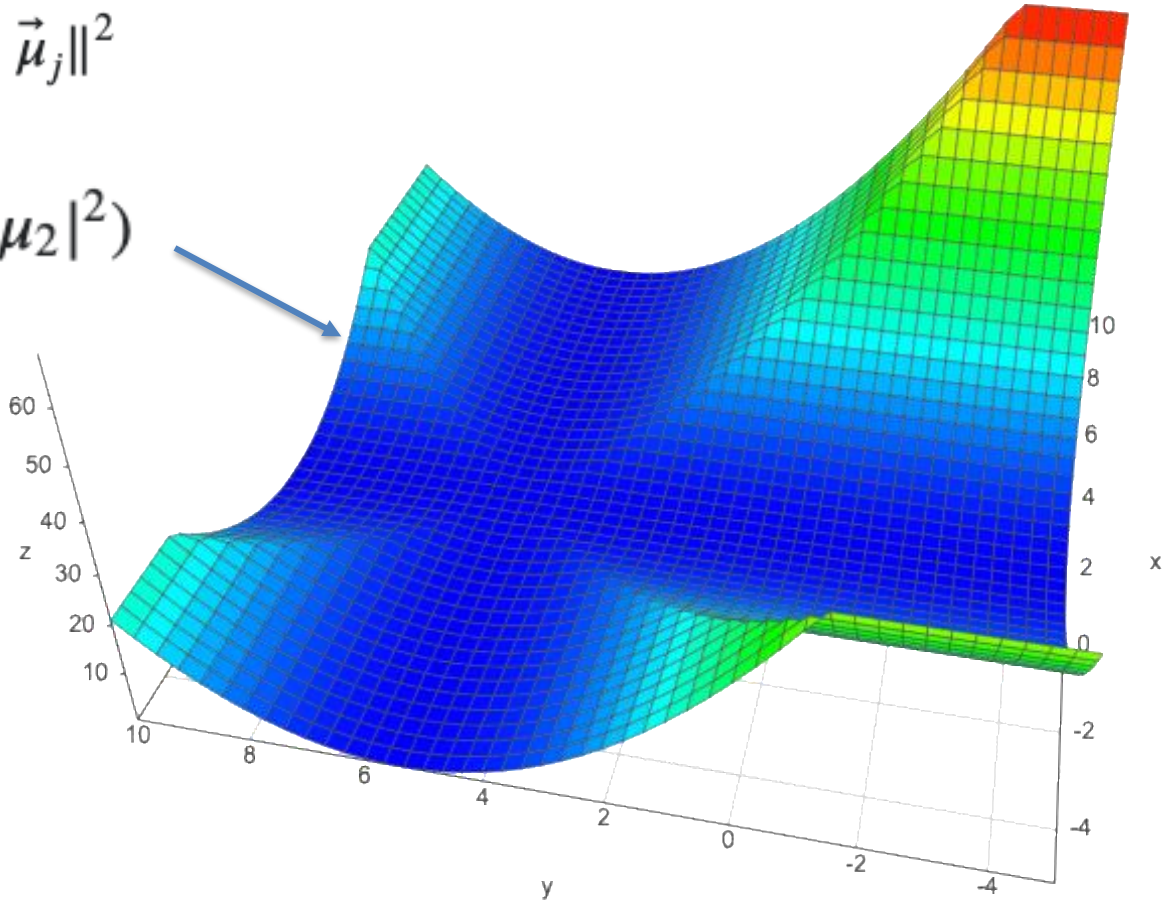
Objective function for k-means is non-convex



Let $\vec{x}_i, i = 1, 2, \ldots, n$ be the data points and $\vec{\mu}_j, j = 1, 2, \ldots, k$ be the k mean values.

$$\text{minimize} \sum_{i=1}^{n} \min_{j=1..k} \|\vec{x}_i - \vec{\mu}_j\|^2$$

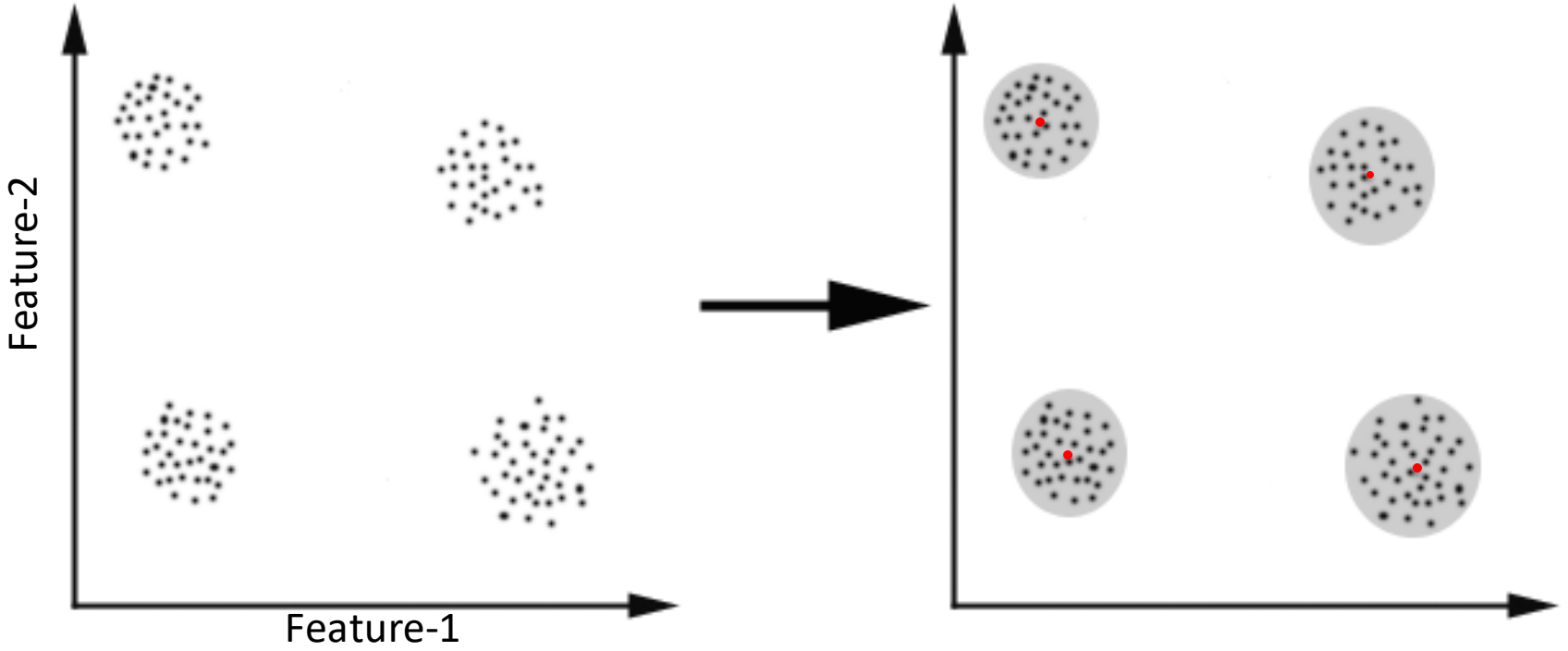
$$\min(|x_i - \mu_1|^2, |x_i - \mu_2|^2)$$



K-means++: Improving K-means initialization

- Common way to improve k-means - smart initialization!
- General idea - try to get good coverage of the data.
- k-means++ algorithm:
  1. Pick the first center randomly
  2. For all points $\mathbf{x}^{(n)}$ set $d^{(n)}$ to be the distance to closest center.
  3. Pick the new center to be at $\mathbf{x}^{(n)}$ with probability proportional to $d^{(n)2}$
  4. Repeat steps 2+3 until you have k centers

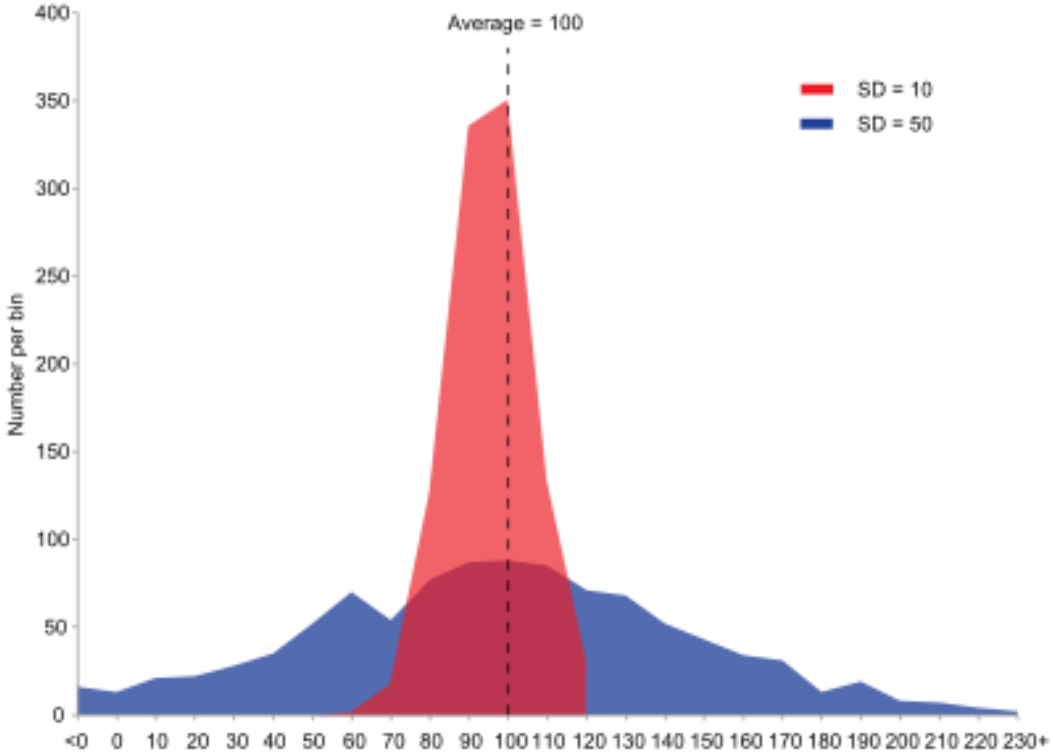
Perspective: Clustering as a ‘summary’ of input data



Output of k-means = ‘centers’

… but only these are not sufficient to summarize

Mean, Standard Deviation and Variance (1-D)



Encodes spread wrt mean

Sample Variance

$$s^2 = \frac{\sum(x - \bar{x})^2}{n - 1}$$

Sample Standard Deviation

$$s = \sqrt{\frac{\sum(x - \bar{x})^2}{n - 1}}$$

Covariance

Vectors 1 and 3

Cell (3, 1) or (1, 3)

$$A = \begin{bmatrix} 1 & 3 & 5 \\ 5 & 4 & 1 \\ 3 & 8 & 6 \end{bmatrix}$$

Covariance

$$\begin{bmatrix} 2.67 & 0.67 & -2.67 \\ 0.67 & 4.67 & 2.33 \\ -2.67 & 2.33 & 4.67 \end{bmatrix}$$

Covariance Matrix

Variance:

$$s^2 = \frac{\sum(\overline{X} - X_i)^2}{N}$$

$$cov(\boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

$$\begin{bmatrix} 0.39701 & 0.51117 \\ 0.55582 & 0.93003 \\ 0.59403 & 0.96645 \\ 0.51544 & 0.29759 \\ 0.85313 & 0.18118 \\ 0.88564 & 0.69114 \end{bmatrix}$$

$$\begin{pmatrix} & M1 & M2 & M3 & \dots & Mn \\ S1 & q_{1,1} & q_{1,2} & q_{1,3} & \dots & q_{1,n} \\ S2 & q_{2,1} & q_{2,2} & q_{2,3} & \dots & q_{2,n} \\ S3 & q_{3,1} & q_{3,2} & q_{3,3} & \dots & q_{3,n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ Sm & q_{m,1} & q_{m,2} & q_{m,3} & & q_{m,n} \end{pmatrix}$$

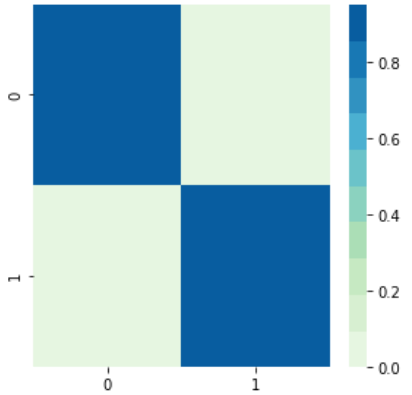
$$Cov(M_a, M_b) = \frac{1}{m}\sum_{i=1}^{m}(q_{i,a} - \overline{q_a})(q_{i,b} - \overline{q_b})$$

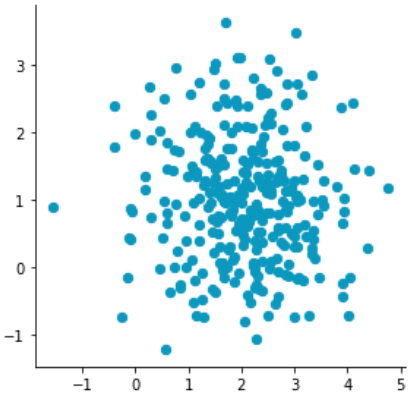
$$C = \begin{pmatrix} \mathrm{cov}(M_1, M_1) & \mathrm{cov}(M_1, M_2) & \cdots & \mathrm{cov}(M_1, M_n) \\ \mathrm{cov}(M_2, M_1) & \mathrm{cov}(M_2, M_2) & \cdots & \mathrm{cov}(M_2, M_n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{cov}(M_n, M_1) & \mathrm{cov}(M_n, M_2) & \cdots & \mathrm{cov}(M_n, M_n) \end{pmatrix}_{n \times n}$$

n-dimensional Covariance Matrix

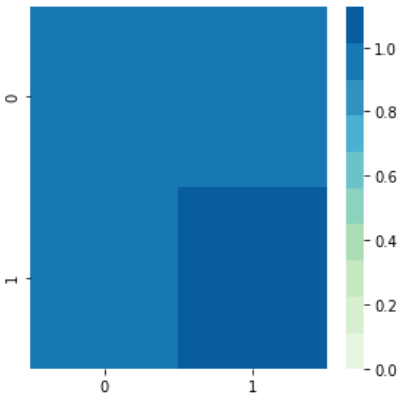
Covariance

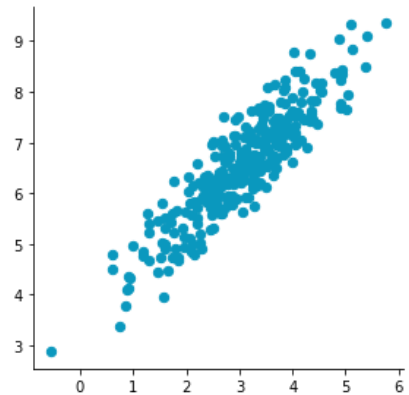
$$cov(\boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$





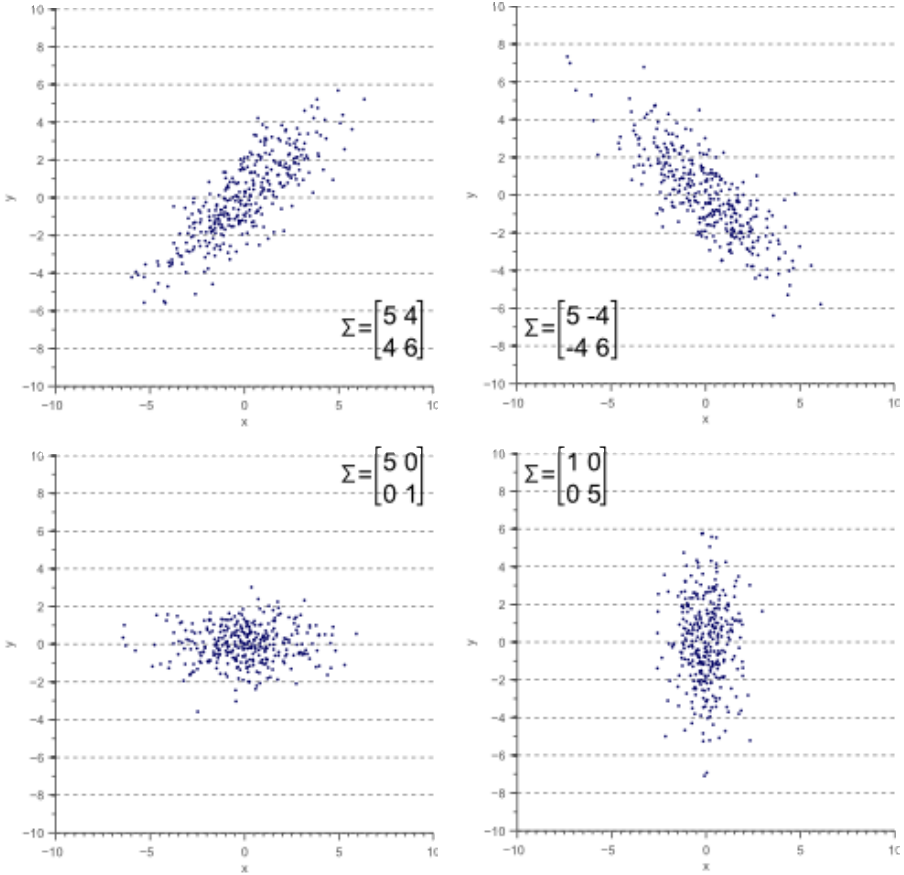
$$C = \begin{bmatrix} +0.95 & -0.04 \\ -0.04 & +0.87 \end{bmatrix}$$



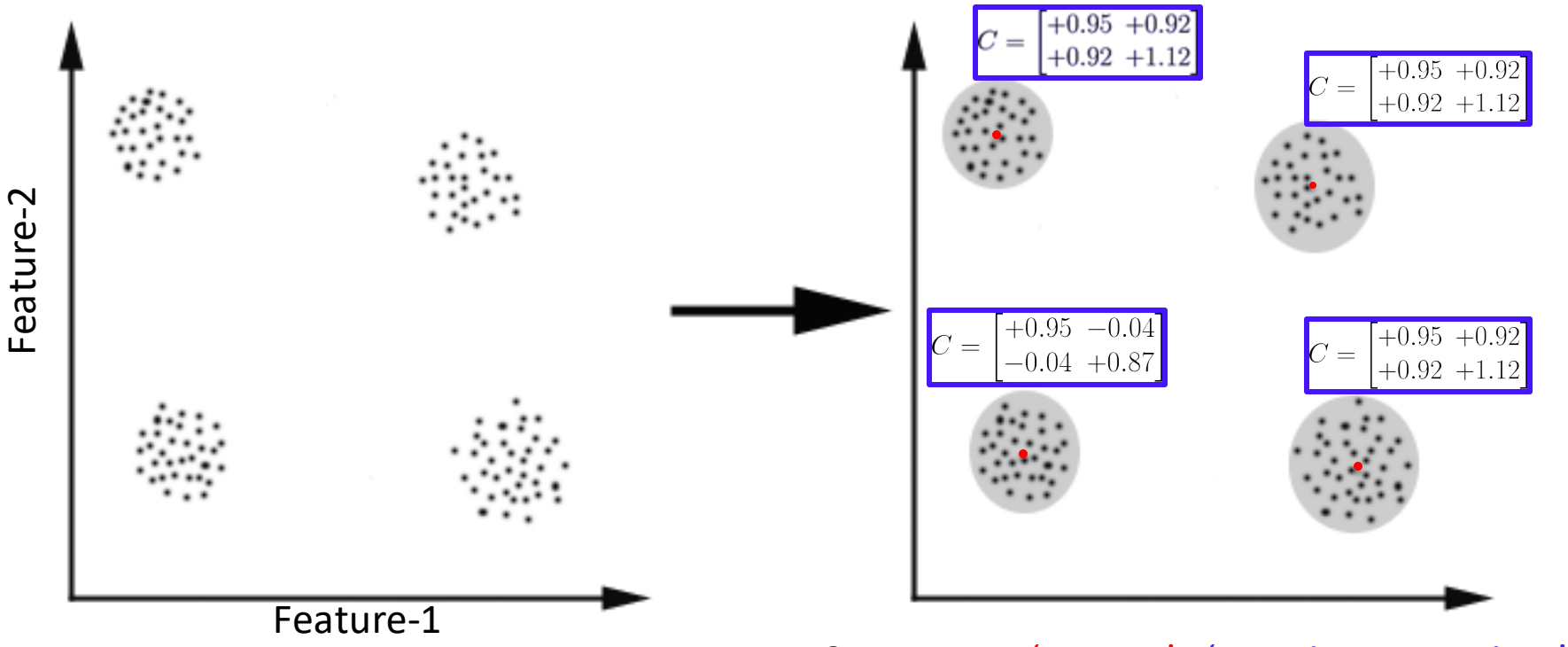


$$C = \begin{bmatrix} +0.95 & +0.92 \\ +0.92 & +1.12 \end{bmatrix}$$

Covariance characterizes spread of data (wrt mean)



Perspective: Clustering as a ‘summary’ of input data version 2



Summary = ‘centers’ , ‘covariance matrices’

K-means: Additional issues

- ‘Hard’ assignments
- Euclidean ➔ Favors ‘Spherical’ clusters
  - Cluster-distribution-adaptive ?