香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

Introduction to Data Science

Lecture 22 Unsupervised learning: Clustering

Zicheng Wang

Recap

What is logistic regression model?

- Model the conditional probability of the label given the data

$P(it\ is\ a\ dog\ |$  $) = ?$

- Use all labeled samples to estimate the parameters of the conditional probability model.

What is logistic regression model?

- **Simplest** case (two classes): y ∈ $\{0\ ,1\}$

- **Logistic regression model:**

Logistic function

$$p(y=1|\boldsymbol{x},\boldsymbol{\theta},b)=\boxed{\frac{1}{1+\exp(-(\boldsymbol{\theta}^\top\boldsymbol{x}+b))}}$$

$$p(y=0|\boldsymbol{x},\boldsymbol{\theta},b)=\frac{\exp(-(\boldsymbol{\theta}^\top\boldsymbol{x}+b))}{1+\exp(-(\boldsymbol{\theta}^\top\boldsymbol{x}+b))}$$

Train the Model

- **How to find θ *and* b? MLE**

- Given m labeled samples $(\boldsymbol{x}^i, y^i)$, i = 1, ...m

- Find $\boldsymbol{\theta}$ and b such that the likelihood of observing the labeled samples is maximized

$$\max_{\boldsymbol{\theta},b} l(\boldsymbol{\theta}, b) := \log \prod_{i=1}^{m} P\left(y^i \middle| \boldsymbol{x}^i, \boldsymbol{\theta}, b\right) = \sum_{i=1}^{m} \log P\left(y^i \middle| \boldsymbol{x}^i, \boldsymbol{\theta}, b\right)$$

- Usually, we equivalently maximize the averaged likelihood

$$\max_{\boldsymbol{\theta},b} \frac{1}{m} l(\boldsymbol{\theta}, b) := \frac{1}{m} \sum_{i=1}^{m} \log P\left(y^i \middle| \boldsymbol{x}^i, \boldsymbol{\theta}, b\right)$$

Good news: $l(\boldsymbol{\theta}, b)$ is concave in $(\boldsymbol{\theta}, b)$

Bad news: no closed form solution to the problem

We need to use numerical methods to find $(\boldsymbol{\theta}^*, b^*)$ that maximizes $l(\boldsymbol{\theta}, b)$

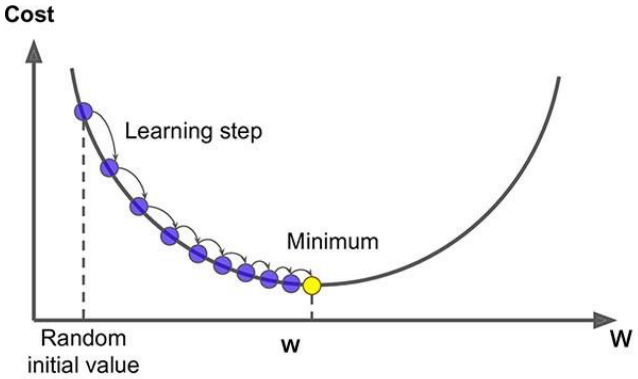
Gradient Descent Method

- Start with an initial point $x^{(0)}$

$\alpha^{(t)}$: the step size or learning rate

- Update our point by the following rule:

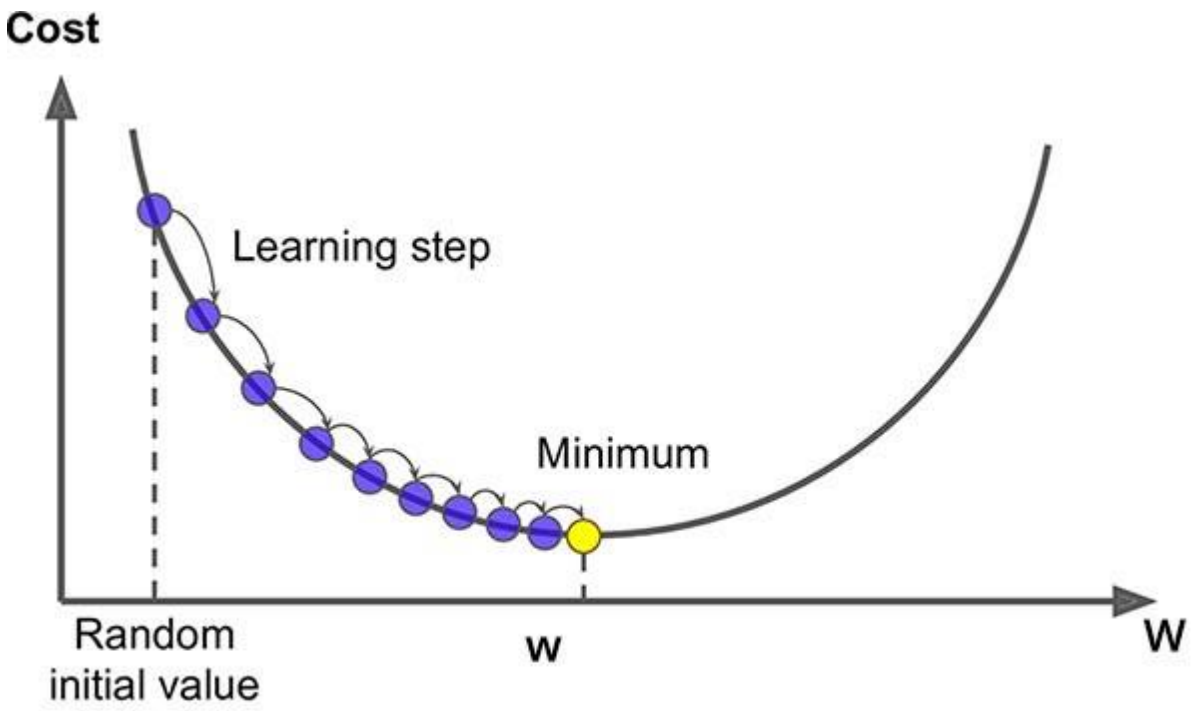
$$x^{(t+1)} = x^{(t)} - \boxed{\alpha^{(t)}} f'(x^{(t)})$$



- Stopping criteria:
  - $|x^{(t+1)} - x^{(t)}| \leq \varepsilon$
  - or $|f'(x^{(t)})| \leq \varepsilon$

How to select $\alpha^{(t)}$? The selection of $\alpha^{(t)}$ will affect the rate at which we find the local minimizer. A bad selection of $\alpha^{(t)}$ can result in the failure of the algorithm.

We may want the step size, $\alpha^{(t)}$, to be large during the initial steps and smaller as we approach the local minimizer

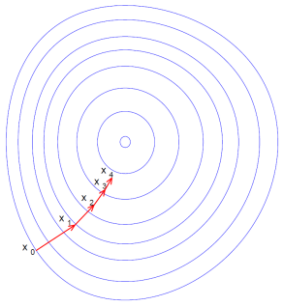


Gradient Descent algorithm for logistic regression

- Initialize parameter $(\boldsymbol{\theta}^0, b^0)$
- While $|\theta^{t+1} - \theta^t| > \epsilon$ or $|b^{t+1} - b^t| > \epsilon$, Do

$$\boldsymbol{\theta}^{t+1} \leftarrow \boldsymbol{\theta}^t + \alpha^{(t)} \frac{1}{m} \sum_i (y^i - 1)\, \boldsymbol{x}^i + \frac{\exp\left(-\boldsymbol{\theta}^{t^T} \boldsymbol{x}^i - b^t\right) \boldsymbol{x}^i}{1 + \exp\left(-\boldsymbol{\theta}^{t^T} \boldsymbol{x}^i - b^t\right)}$$

$$b^{t+1} \leftarrow b^t + \alpha^{(t)} \frac{1}{m} \sum_i (y^i - 1) + \frac{\exp\left(-\boldsymbol{\theta}^{t^T} \boldsymbol{x}^i - b^t\right)}{1 + \exp\left(-\boldsymbol{\theta}^{t^T} \boldsymbol{x}^i - b^t\right)}$$



A variant: Stochastic gradient descent

- At each iteration, we randomly choose a small batch of samples in the training data set, and update using the stochastic gradient

$$\boldsymbol{\theta}^{t+1} \leftarrow \boldsymbol{\theta}^{t} + \alpha^{(t)} \frac{1}{|B|} \sum_{i \in B} \left((y^i - 1)\, \boldsymbol{x}^i + \frac{\exp\left(-\boldsymbol{\theta}^{t^T} \boldsymbol{x}^i - b^t\right) \boldsymbol{x}^i}{1 + \exp\left(-\boldsymbol{\theta}^{t^T} \boldsymbol{x}^i - b^t\right)} \right.$$

$$b^{t+1} \leftarrow b^{t} + \alpha^{(t)} \frac{1}{|B|} \sum_{i \in B} \left((y^i - 1) + \frac{\exp\left(-\boldsymbol{\theta}^{t^T} \boldsymbol{x}^i - b^t\right)}{1 + \exp\left(-\boldsymbol{\theta}^{t^T} \boldsymbol{x}^i - b^t\right)} \right.$$

- B: the batch we use in each iteration

Unsupervised Learning

Unsupervised Learning

- Data lacks structured or objective answers, such as labels.

- In other words, for all samples (x^i, y^i), where $i = 1, \ldots N$, you can observe x^i but y^i remains unseen.

Training data

What Is Unsupervised Machine Learning?

- There is no predefined correct output for a given input.
- Instead, the algorithm must interpret the input and make the appropriate decision.
- The aim is to **examine the data and discern underlying patterns**.

Examples

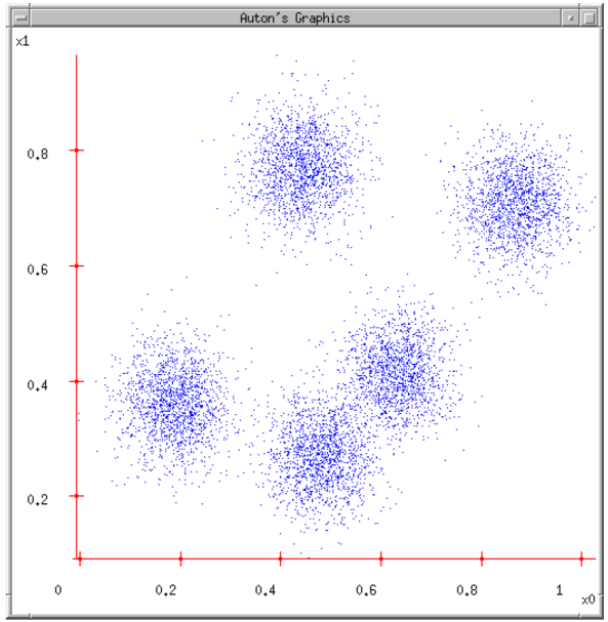
- The algorithm can identify customer segments who possess similar attributes. Customers within these segments can then be targeted by similar marketing campaigns.

- The algorithms are subsequently used to segment topics, identify outliers and recommend items.

Clustering

So, what is clustering in general?

- The algorithm figures out the grouping of objects based on the chosen **similarity/dissimilarity function**
  - **Points within a cluster are similar**
  - **Points across clusters are not so similar**



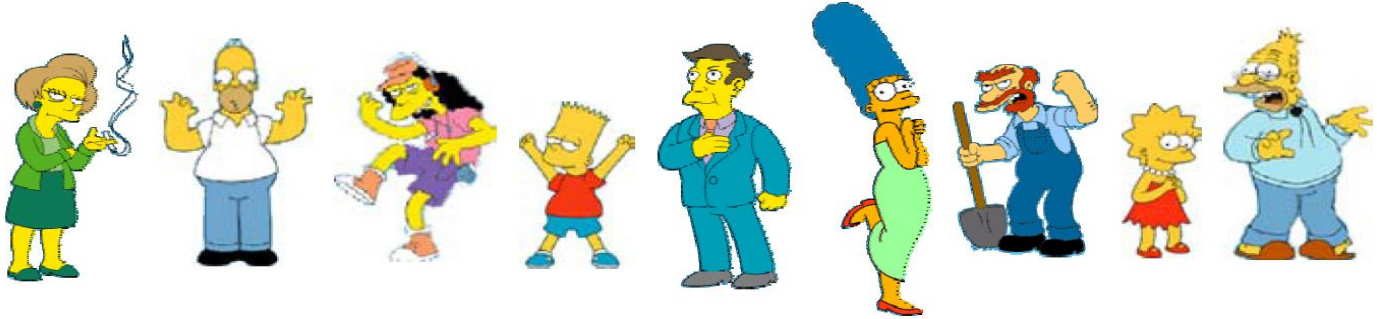
How to conduct?

Introduce a similarity function to measure whether two objects are similar.

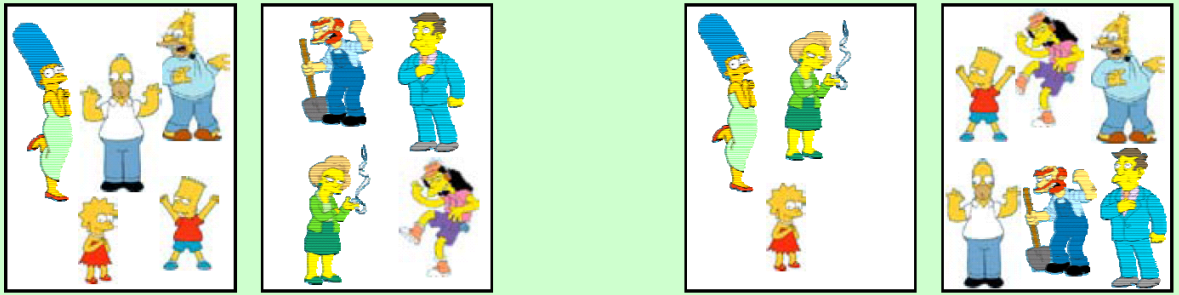
Divide objects into groups.

Do these two pictures exhibit similarity?

There is no universal standard for clustering



Clustering is subjective



Simpson's Family | School Employees | Females | Males

How to develop a dissimilarity/similarity function?

- Desired properties of dissimilarity functions
  - Symmetry: $d(x, y) = d(y, x)$
    - *Otherwise you could claim "Alex looks like Bob, but Bob looks nothing like Alex“*
  - Positive separability: $d(x, y) = 0$, if and only if $x = y$
    - *Otherwise there are objects that are different, but you cannot tell apart*
  - Triangular inequality: $d(x, y) \leq d(x, z) + d(z, y)$
    - *Otherwise you could claim "Alex is very like Bob, and Alex is very like Carl, but Bob is very unlike Carl"*

Distance functions for vectors

- Given two data points, both in R^n
  - $x = (x_1, x_2, \dots, x_n)^\top$
  - $y = (y_1, y_2, \dots, y_n)^\top$

- Euclidian distance: $d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$

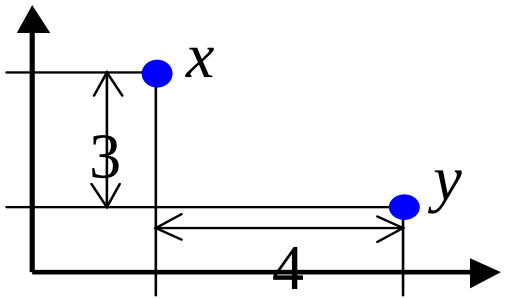
- Minkowski distance: $d(x, y) = \sqrt[p]{\sum_{i=1}^{n} (x_i - y_i)^p}$

  - Euclidian distance: $p = 2$

  - Manhattan distance: $p = 1, d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$

  - “inf”-distance: $p = \infty, d(x, y) = \max_{i=1}^{n} |x_i - y_i|$

Distance example



- Euclidian distance: $\sqrt{4^2 + 3^2} = 5$
- Manhattan distance: $4 + 3 = 7$
- “inf”-distance: $max\{4,3\} = 4$

How to conduct?

Introduce a similarity function to measure whether two objects are similar.

Divide objects into groups.

K-Means Clustering

Intuition

- The commonality within the same group is represented by the average value of data points.
  - Cluster centers

- The nearest cluster centers for any two points within the same group are the same

K-means

- Given m data points, $\{x^1, x^2, \dots x^m\}$
- Find k cluster centers, $\{c^1, c^2, \dots, c^k\}$
- And assign each data point i to one cluster, $\pi(i) \in \{1, \dots, k\}$
- Such that the sum of the distances from each data point to its respective cluster center is minimized

$$\min_{c,\pi} \sum_{i=1}^{m} d\left(x^i, c^{\pi(i)}\right)$$

L2-norm distance

- Given m data points, $\{x^1, x^2, \dots x^m\}$
- Find k cluster **centers**, $\{c^1, c^2, \dots, c^k\}$
- And assign each data point i to one cluster, $\pi(i) \in \{1, \dots, k\}$
- Such that the averaged distances from each data point to its respective cluster center is minimized

$$\min_{c,\pi} \frac{1}{m} \sum_{i=1}^{m} \left\| x^i - c^j \right\|^2$$

K-means algorithm

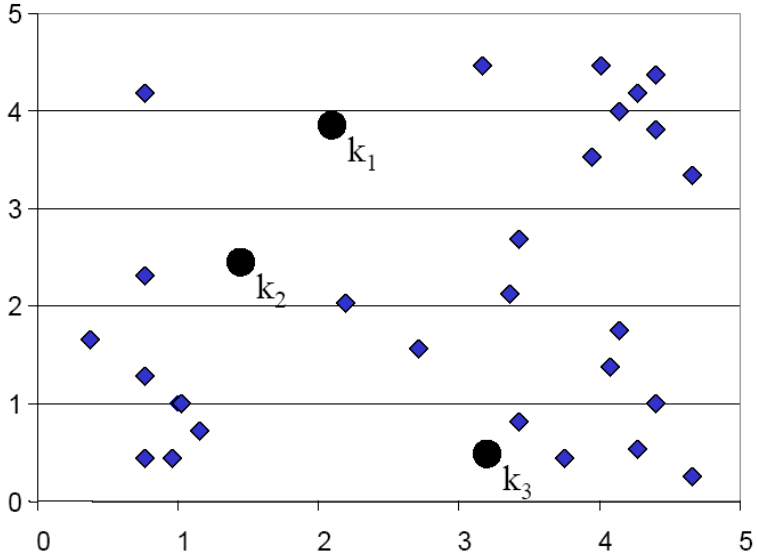
- Step 1: Initialize k cluster centers, $\{c^1, c^2, \dots, c^k\}$, randomly
- Step 2: Do
  - Decide the cluster memberships of each data point, x^i, by assigning it to the nearest cluster center (cluster assignment)

$$\pi(i) = argmin_{j=1,\dots,k} \left\| x^i - c^j \right\|^2$$

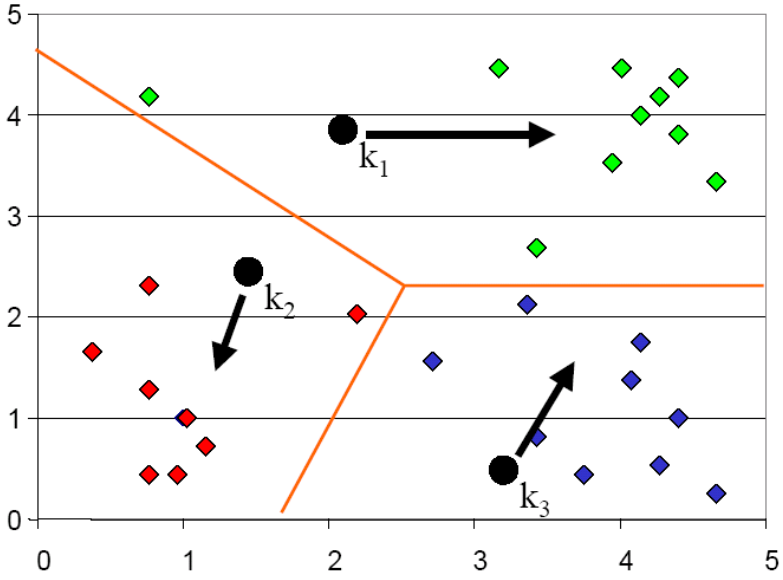
  - Adjust the cluster centers (center adjustment)

$$c^j = \frac{1}{|\{i: \pi(i) = j\}|} \sum_{i:\pi(i)=j} x^i$$

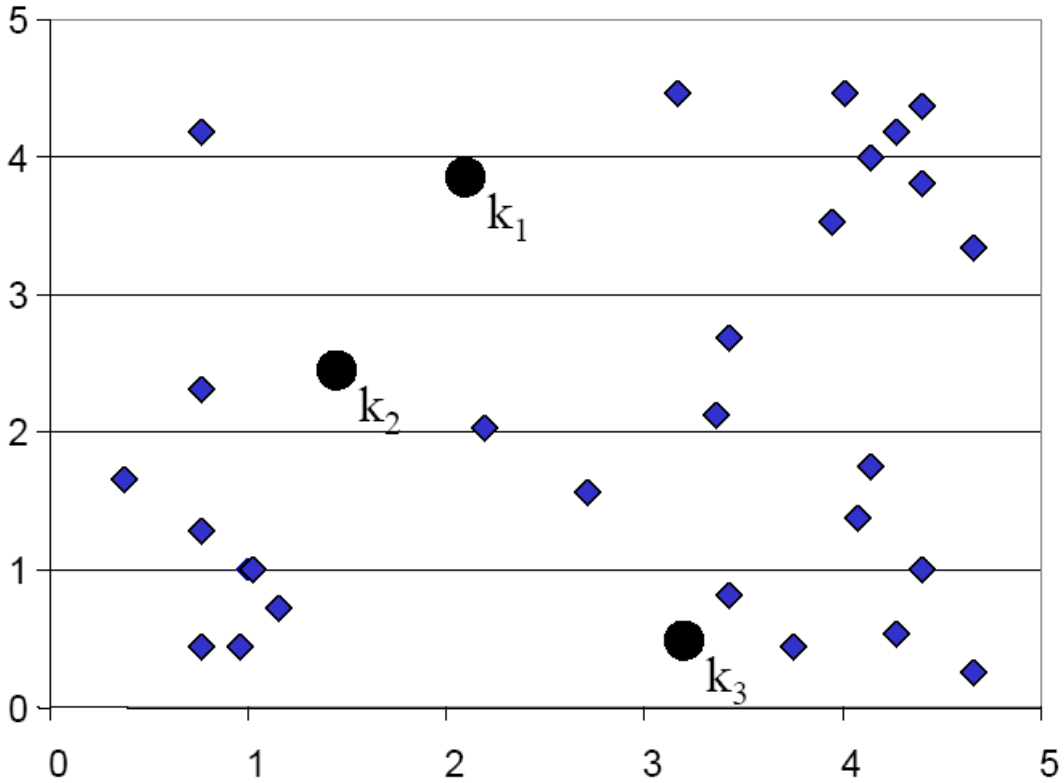
- While any cluster center undergoes changes, go to Step 2



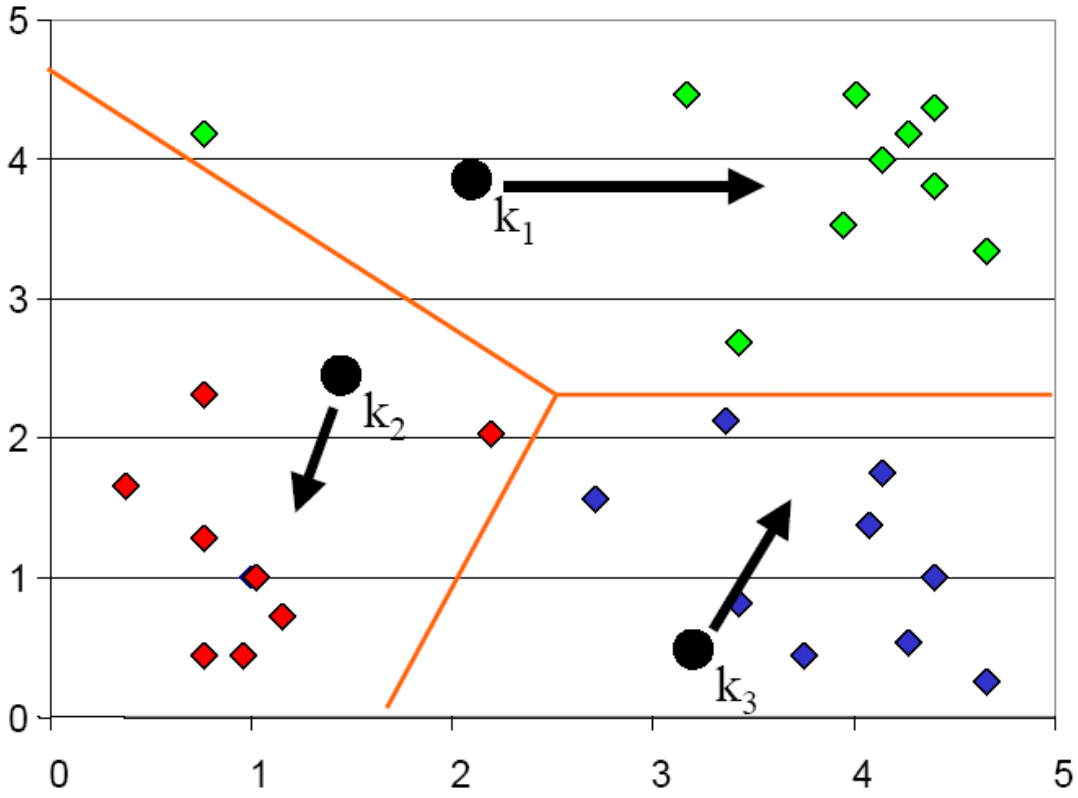
assign each data point to one cluster



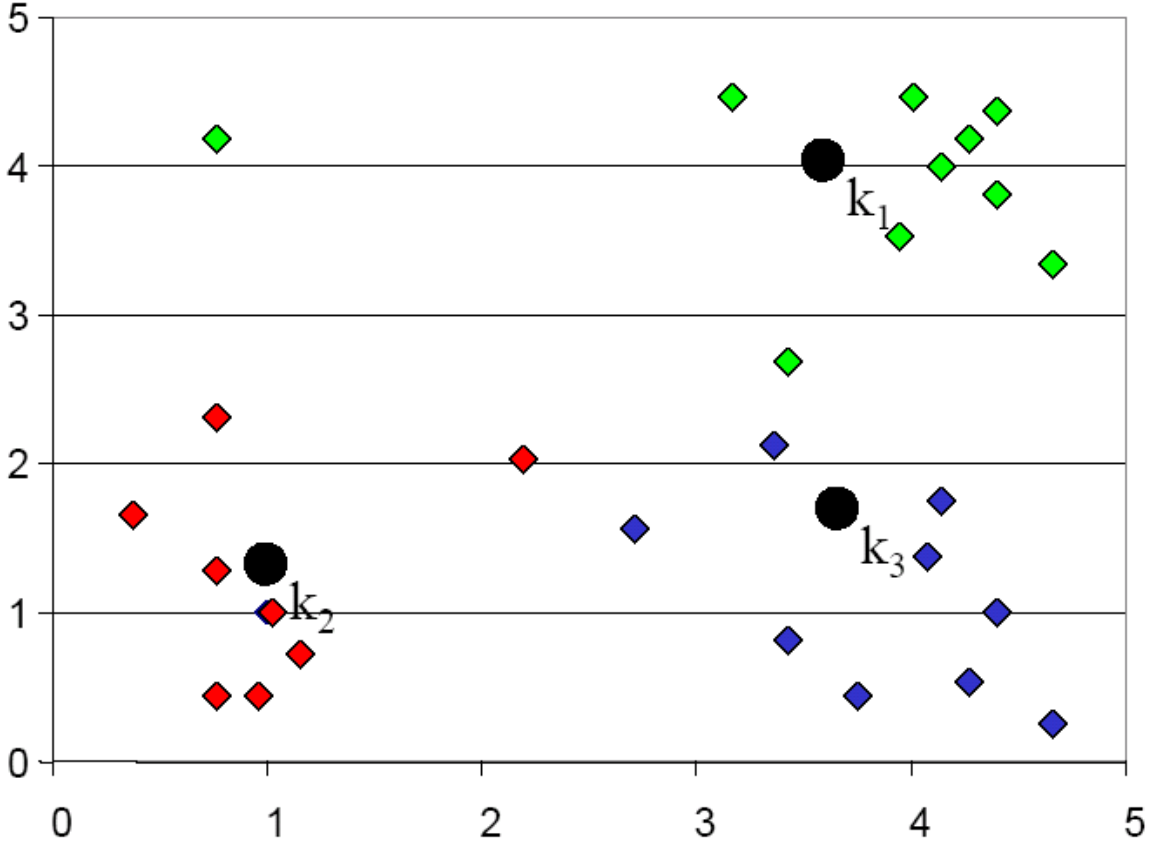
K-means: step 1



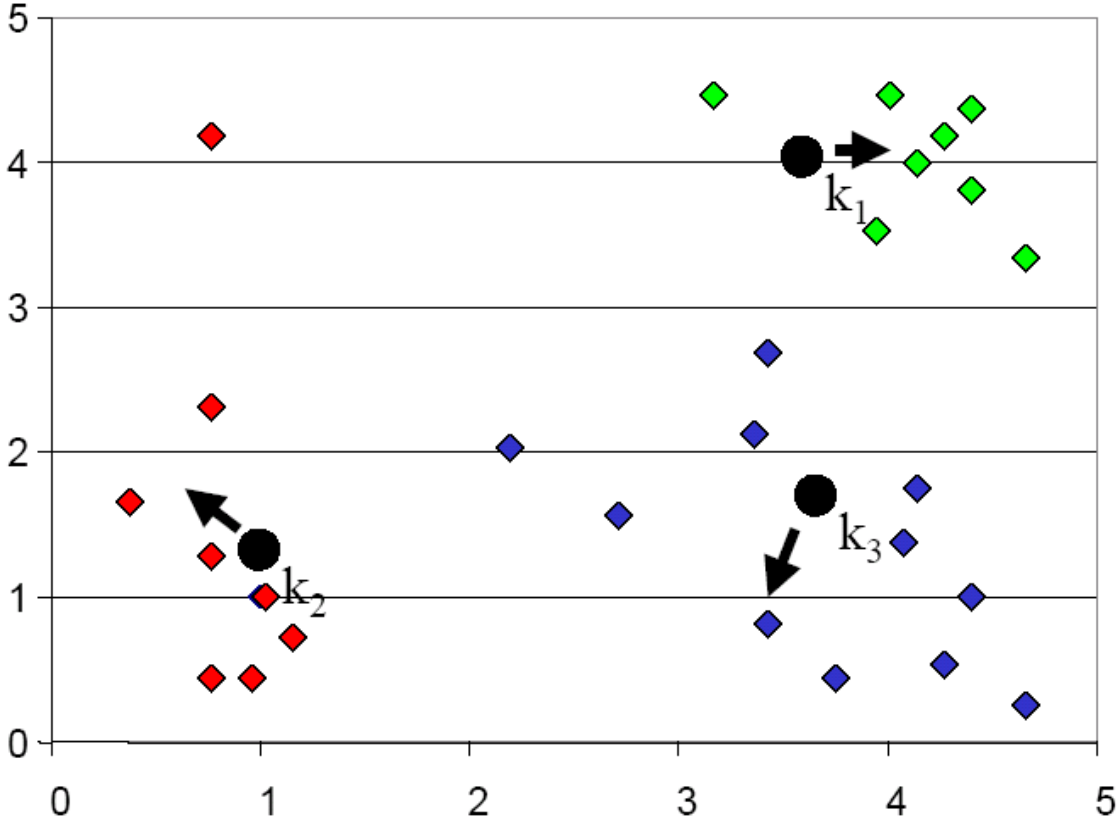
K-means: step 2



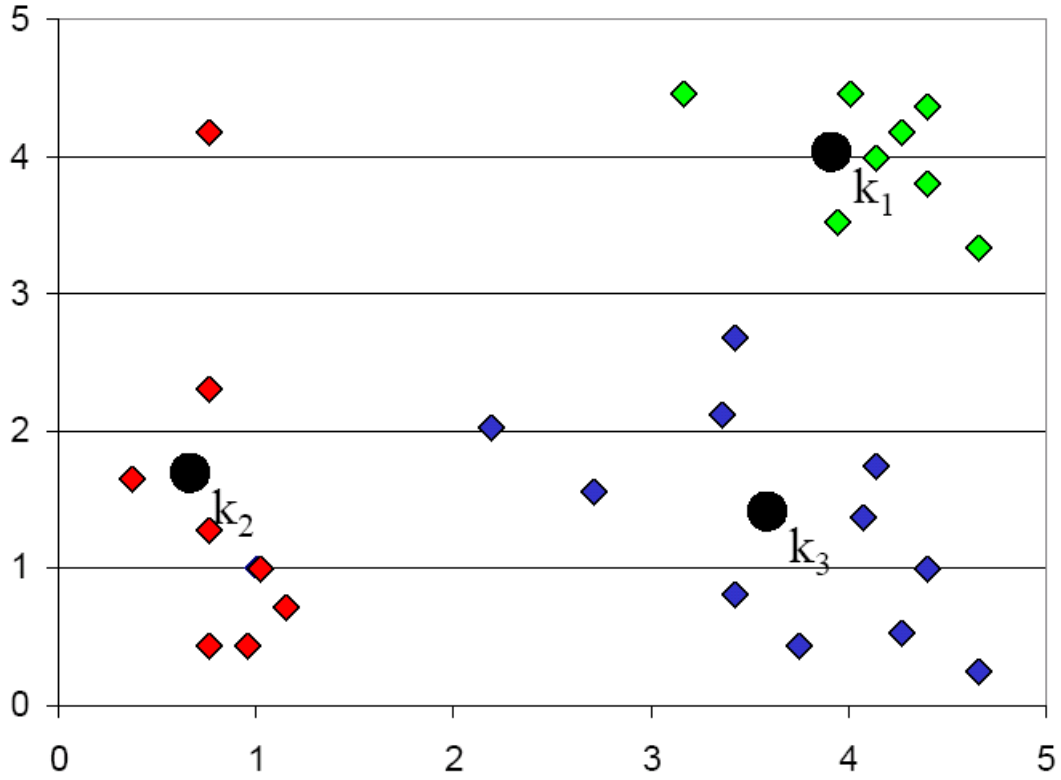
K-means: step 3



K-means: step 4



K-means: step 5



Questions

- Will different initialization lead to different results?
  - Yes
  - No

- Will the algorithm always stop after some iteration?
  - Yes
  - No

Gradient involving Matrices (useful but not required)

- $f(\boldsymbol{x}) = \boldsymbol{a}^T\boldsymbol{x}$, then $\nabla f(\boldsymbol{x}) = \boldsymbol{a}$
- $f(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}$, then $\nabla f(\boldsymbol{x}) = \left(\boldsymbol{A}^{\boldsymbol{T}} + \boldsymbol{A}\right)\boldsymbol{x}$
- $f(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{x}$, then $\nabla f(\boldsymbol{x}) = 2\boldsymbol{x}^T$