CSC4007 Advanced Machine Learning

Lesson 06: Unsupervised Learning

by Vien Ngo
EEECS / ECIT / DSSC

Some slides are from the lectures of Prof. Andrew W. Moore (CMU) and Geoffrey Hinton (UToronto)

Outline

- Unsupervised Learning
- Clustering Problems
- K-means Clustering Algorithm

Outline

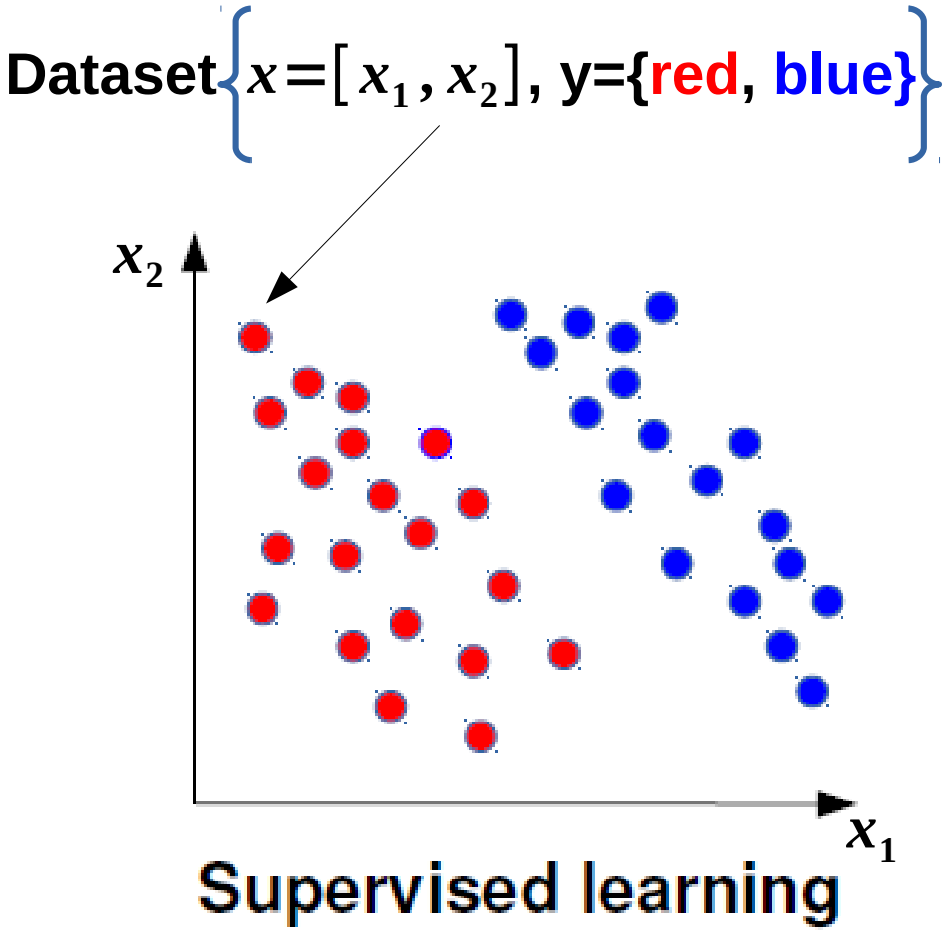
- **Unsupervised Learning**
- Clustering Problems
- K-means Clustering Algorithm

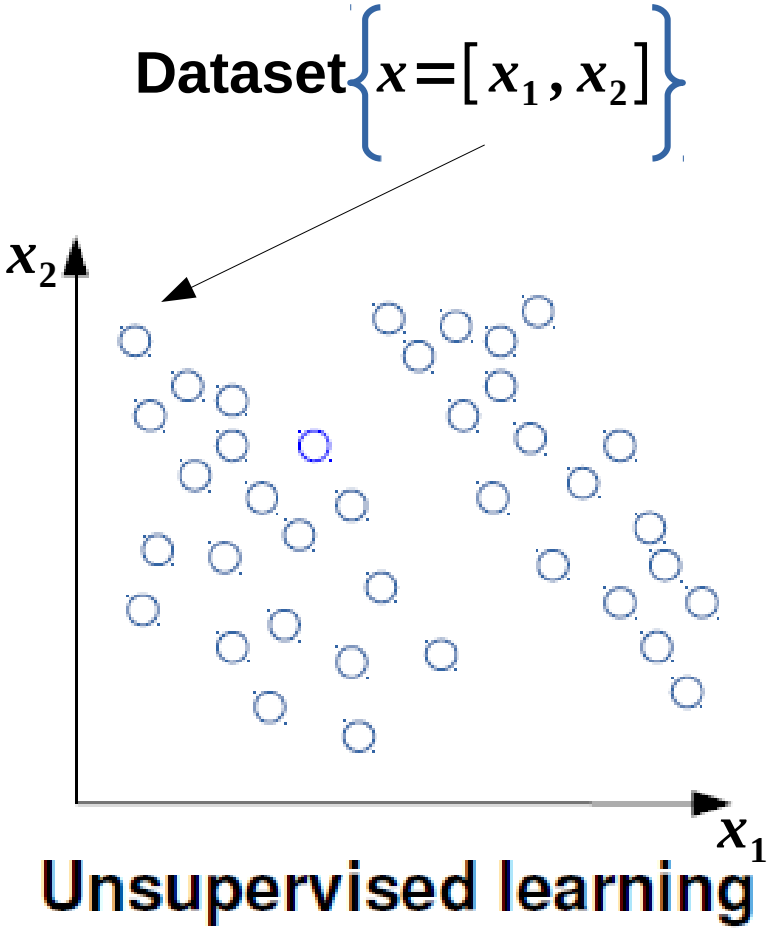
Supervised learning vs. unsupervised learning

- Supervised learning: discover patterns in the data that relate input variables with a target (class) variable.
  - These patterns are then utilized to predict the values of the target variable in future data instances.
- Unsupervised learning: The data have no target variable.
  - We want to explore the data to find some intrinsic structures in them.

Unsupervised Learning

Supervised learning vs. Unsupervised learning





Unsupervised Learning

- Three major problems:
  - Learning generative models
  - Dimensionality reduction
  - Clustering

Learning a Generative Model

Supervised Learning: Example

Dataset

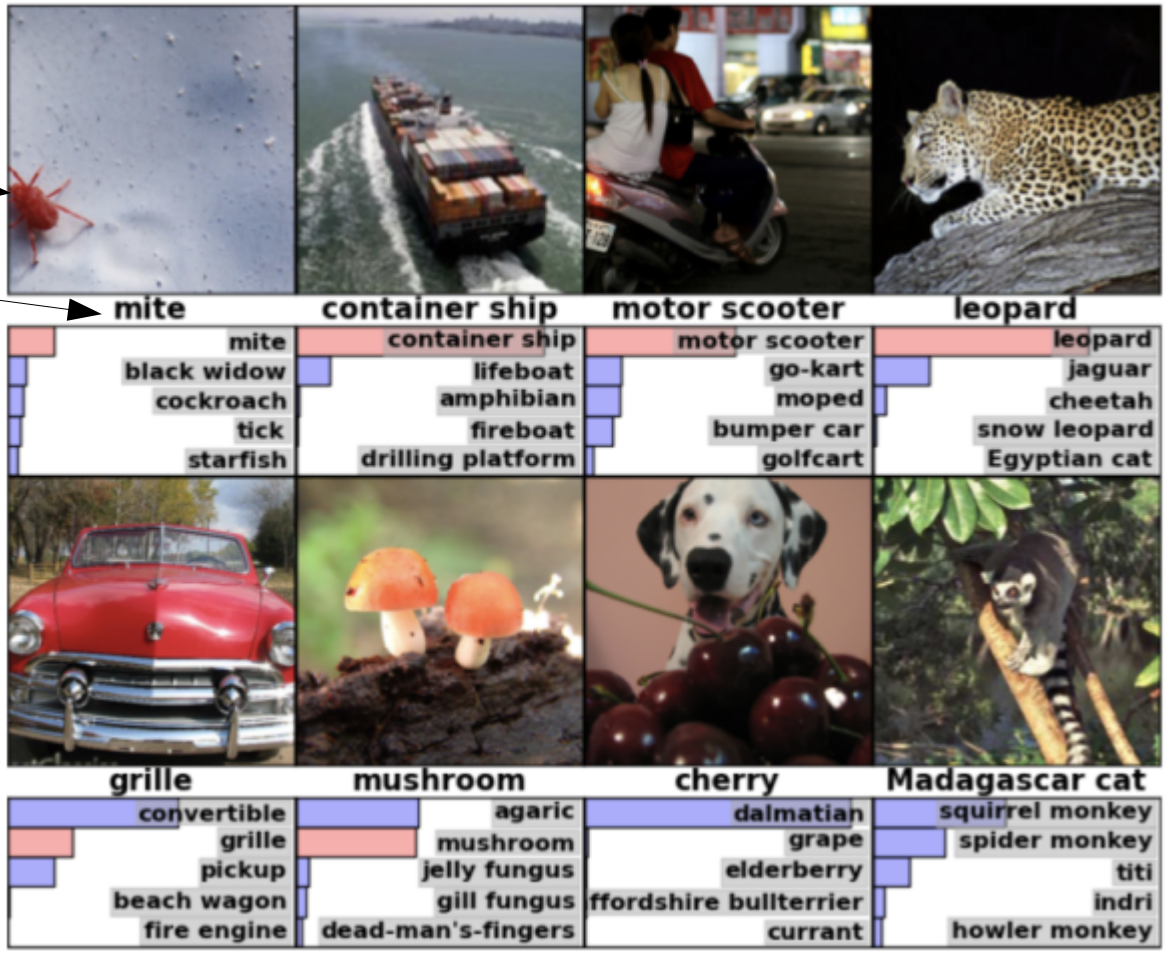
Image inputs **x**

Image labels **y**

Supervised learning

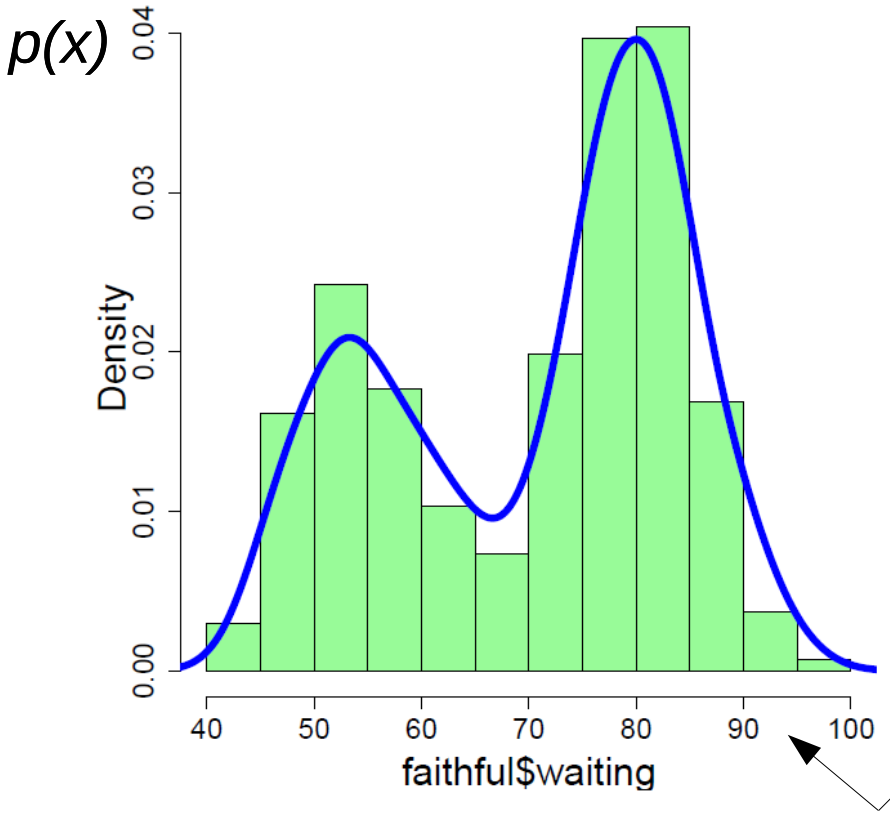
Prediction: y=F(new_x)

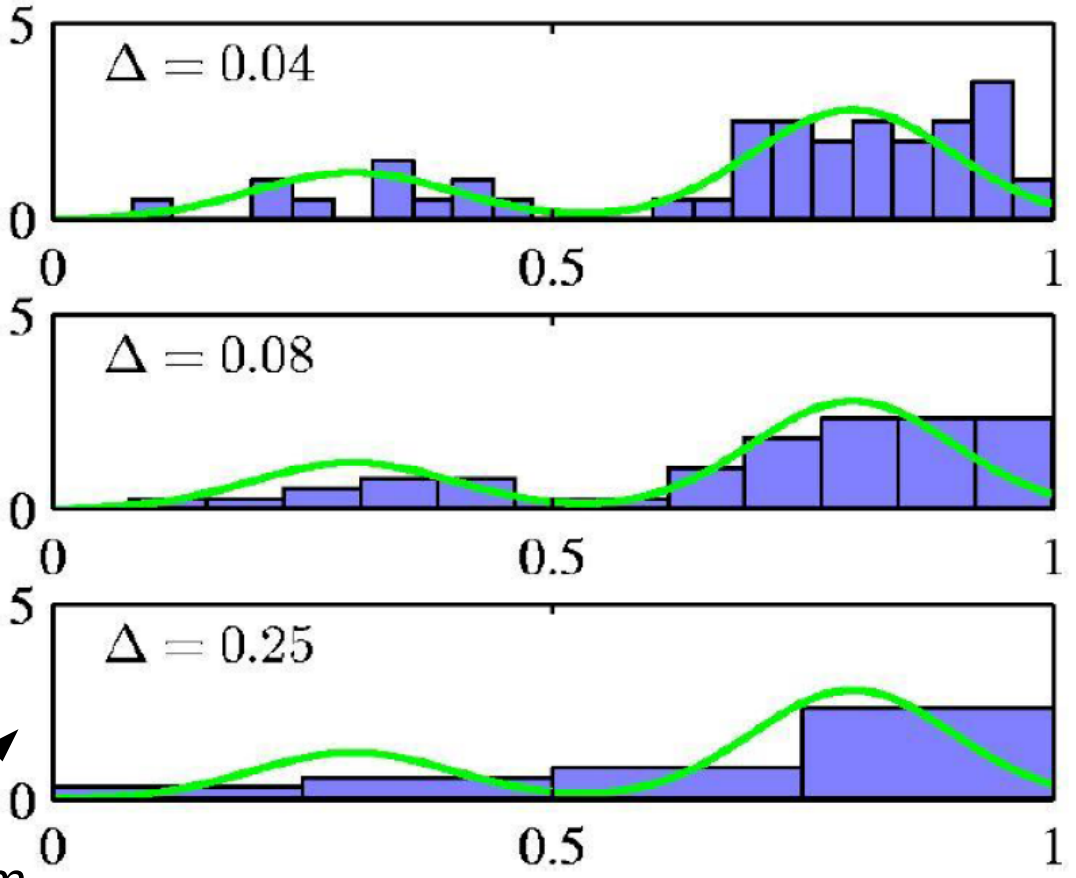
Classification function



Unsupervised Learning: Learning a Generative Model

- We are given input data (unlabeled): $x_1, x_2, \ldots, x_n$
- We want to recover the underlying probability density function generating our dataset : *p(x)*



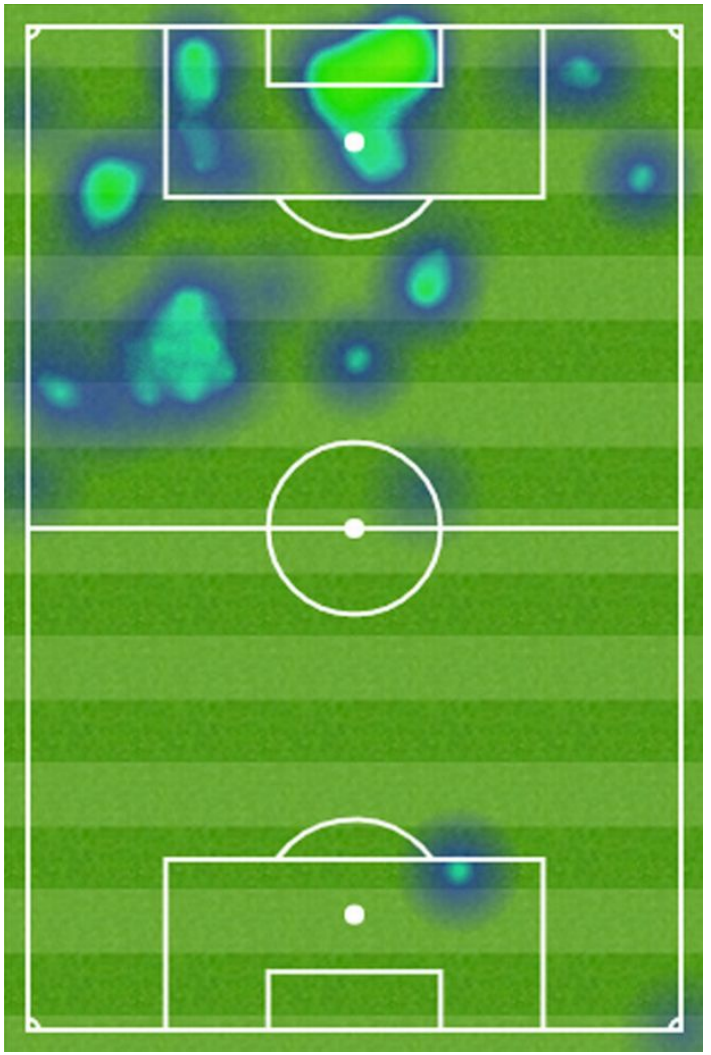


histogram

Learning generative models: Example

- Learning a density function:
  - Given observations: e.g. positions of players on the field

Observations (inputs x) ⟶ P(x)



Heatmap of Ronaldo in a winning game

Unsupervised Learning: Learning a Generative Model

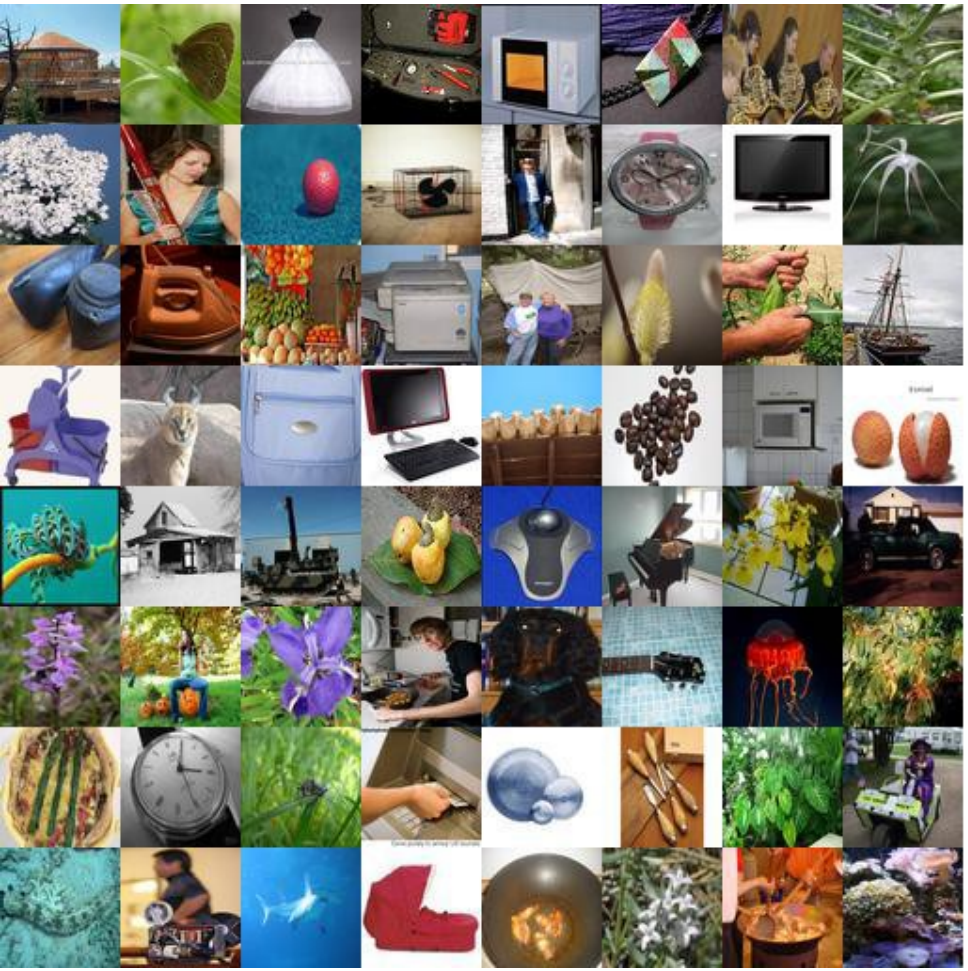
Generative model learning

Dataset Image inputs **x**

P(x)

generative model of images

sampled Images



Learning generative models: Example

- AI faces …

Nvidia's AI-generated faces

Learning generative models: Example

- A source image of a real person (the top row) has the facial characteristics of another person (right-hand column) imposed onto it.
- Traits like skin and hair color are blended together, creating what looks like to be an entirely new person in the process.

https://www.theverge.com

Learning generative models

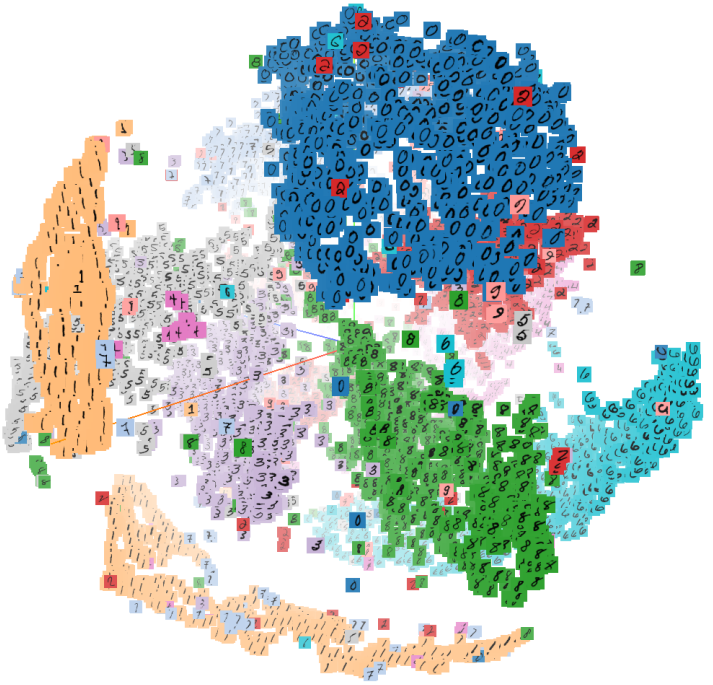
- Techniques for learning generative models:
  - Kernel density estimation
  - Principle component analysis (PCA)
  - (Variational) autoencoder (AE or VAE)
  - Generative adversarial networks (GAN)
  - etc.

https://www.theverge.com

Dimensionality Reduction

Why dimensionality reduction?

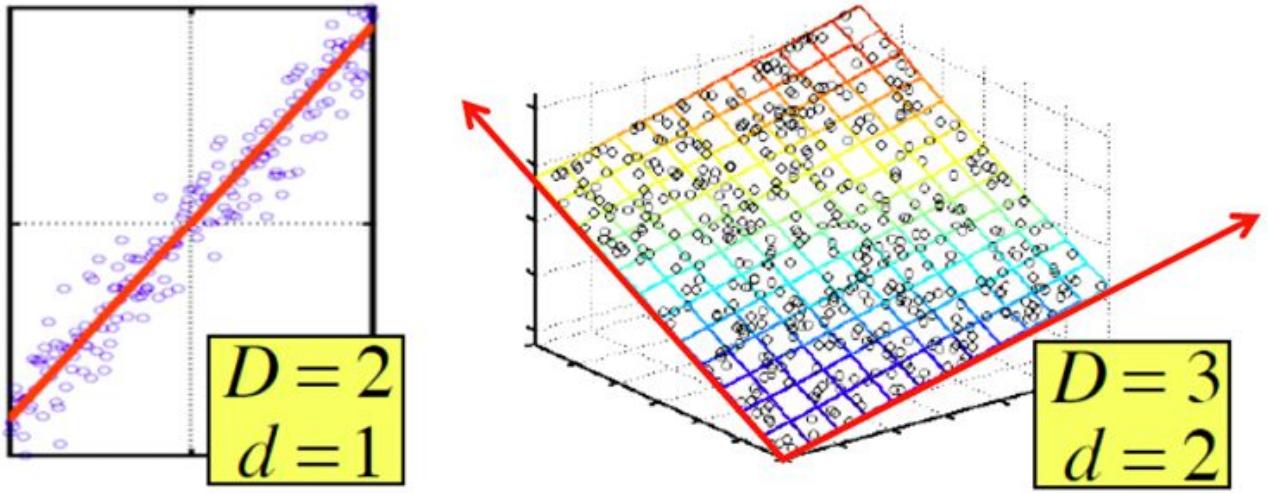
- Some features may be irrelevant
  - E.g. Text:
    - remove stop-words (and, a, the, …)
    - Stemming (going → go, Tom's → Tom, …)
- We want to visualize high dimensional data



- "Intrinsic" dimensionality may be smaller than the number of features

Unsupervised learning: Dimensionality reduction

High-dimensional input data (**D-dim**) might lie in a **lower dimensional space** (**d-dim**)



Eigenfaces

Input: Pictures of human faces Output: A subset of generic faces

Unsupervised Learning: Dimensionality Reduction

- Techniques for dimensionality reduction:
  - Feature Selection
  - Feature Extraction: principle componenent analysis (PCA), LDA, etc.
  - Manifold Learning: t-SNE, ISOMAP, Spectral Embedding, etc.

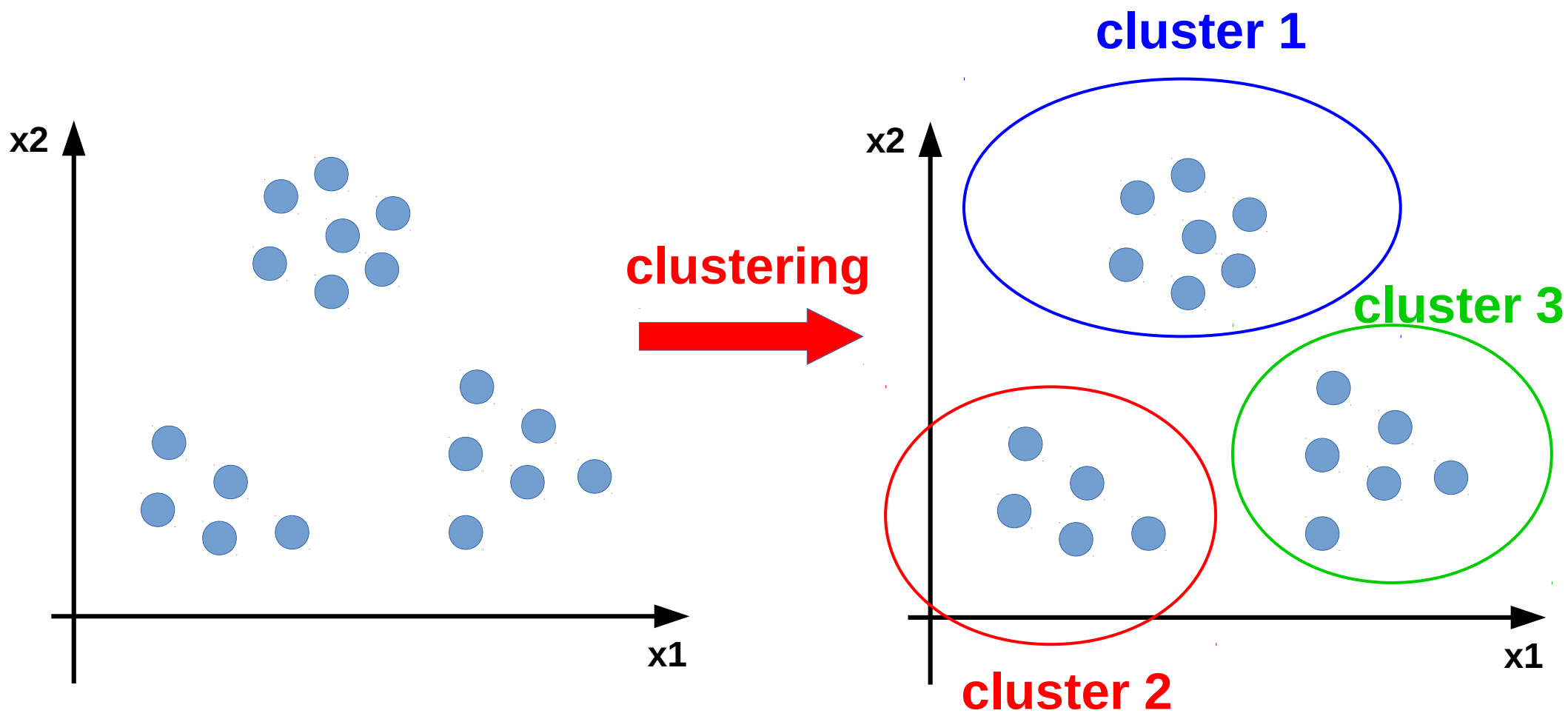
Outline

- Unsupervised Learning
- **Clustering Problems**
- K-means Clustering Algorithm

Clustering

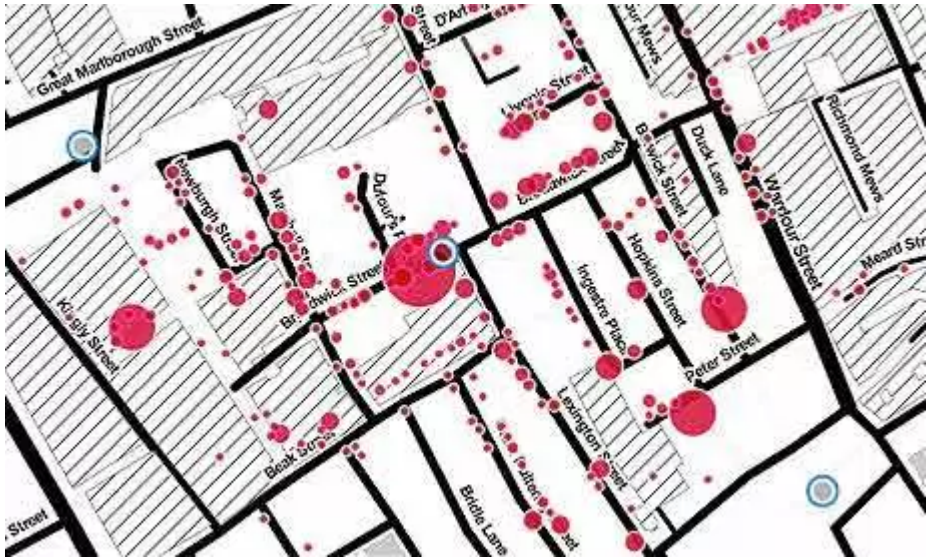
- Clustering is a technique for finding similarity groups in data, called **clusters**. i.e.,
  - it groups data instances that are similar to (near) each other in one cluster and data instances that are very different (far away) from each other into different clusters.
- Clustering is often called an **unsupervised learning** task as no class values denoting an *a priori* grouping of the data instances are given, which is the case in supervised learning.
- Due to historical reasons, clustering is often considered synonymous with unsupervised learning.
  - In fact, association rule mining is also unsupervised
- **This lecture focuses on clustering.**

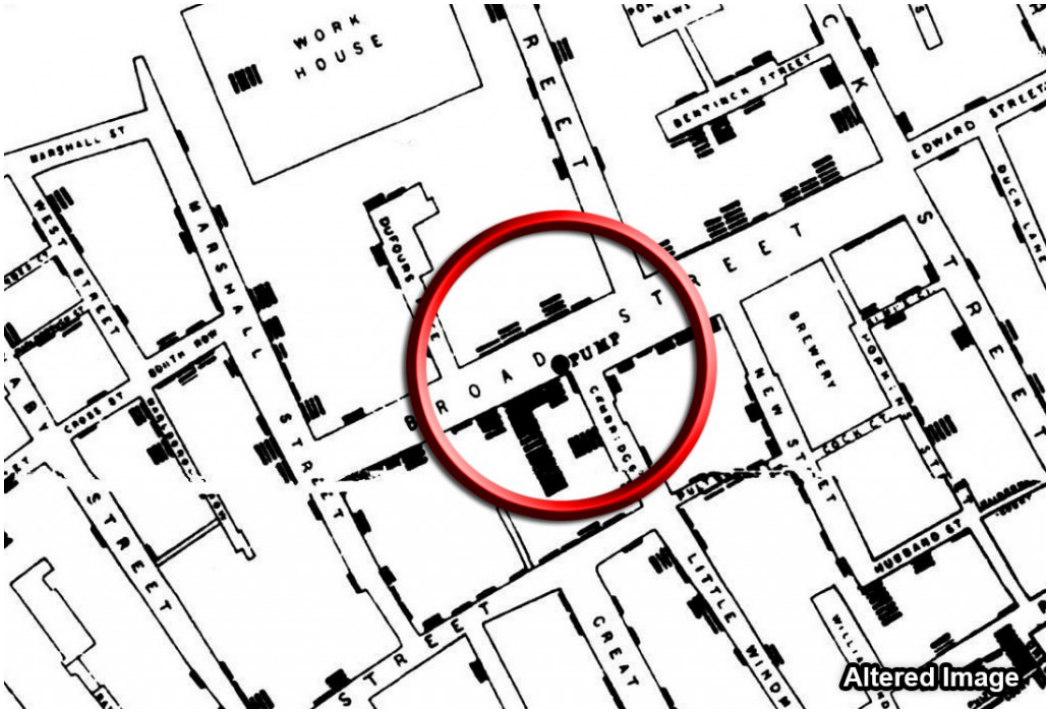
Unsupervised Learning: Clustering



Historic application of clustering

- In 1854, a Cholera ourbreak swept through the Soho neighborhood of London
- Dr. John Snow plotted cholera deaths on a map, and in the corner of a particularly hard-hit quadrangle of buildings was a water pump.
- The locations indicated that cases were clustered around certain intersections where there were polluted wells



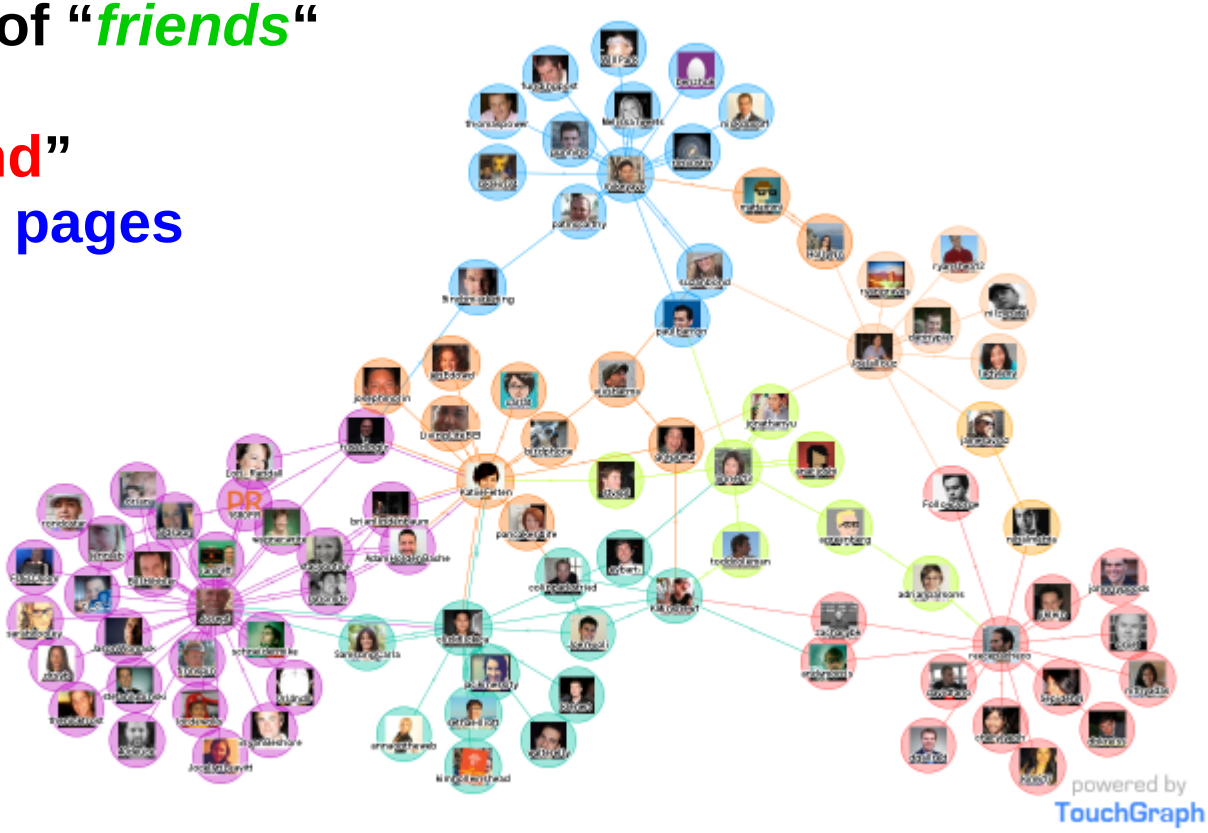


Clustering: Examples

- **Social network analysis**

Input: {person, his interest, his likes, his groups, his friends links, etc.}

Clustering: into groups of "*friends*"
Facebook uses this to**:**
– recommend "**Add Friend**"
– recommend to **like new pages**
– put **new ads**
– etc.

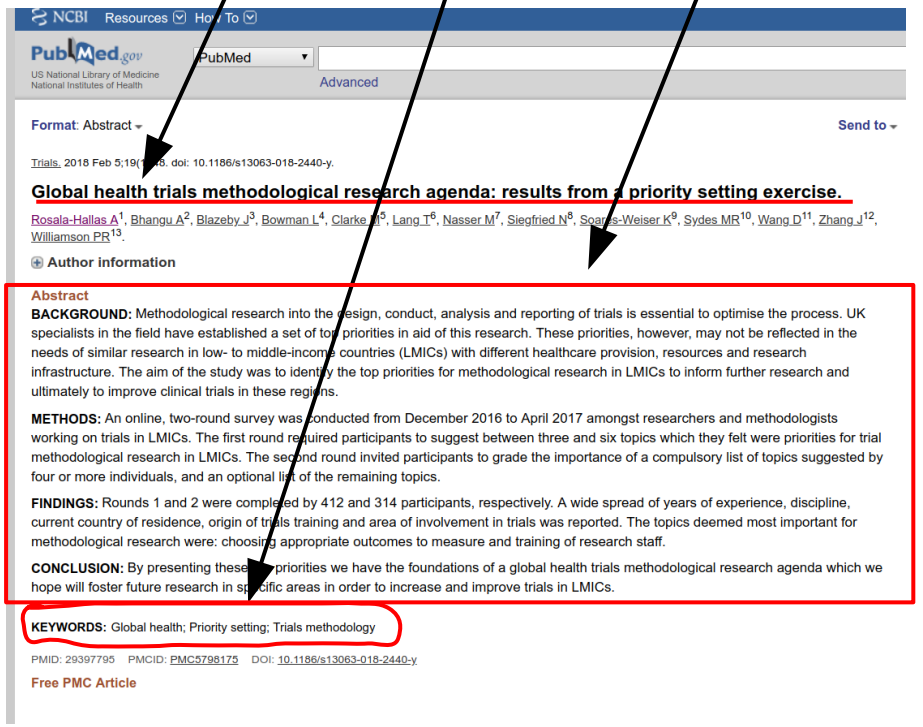


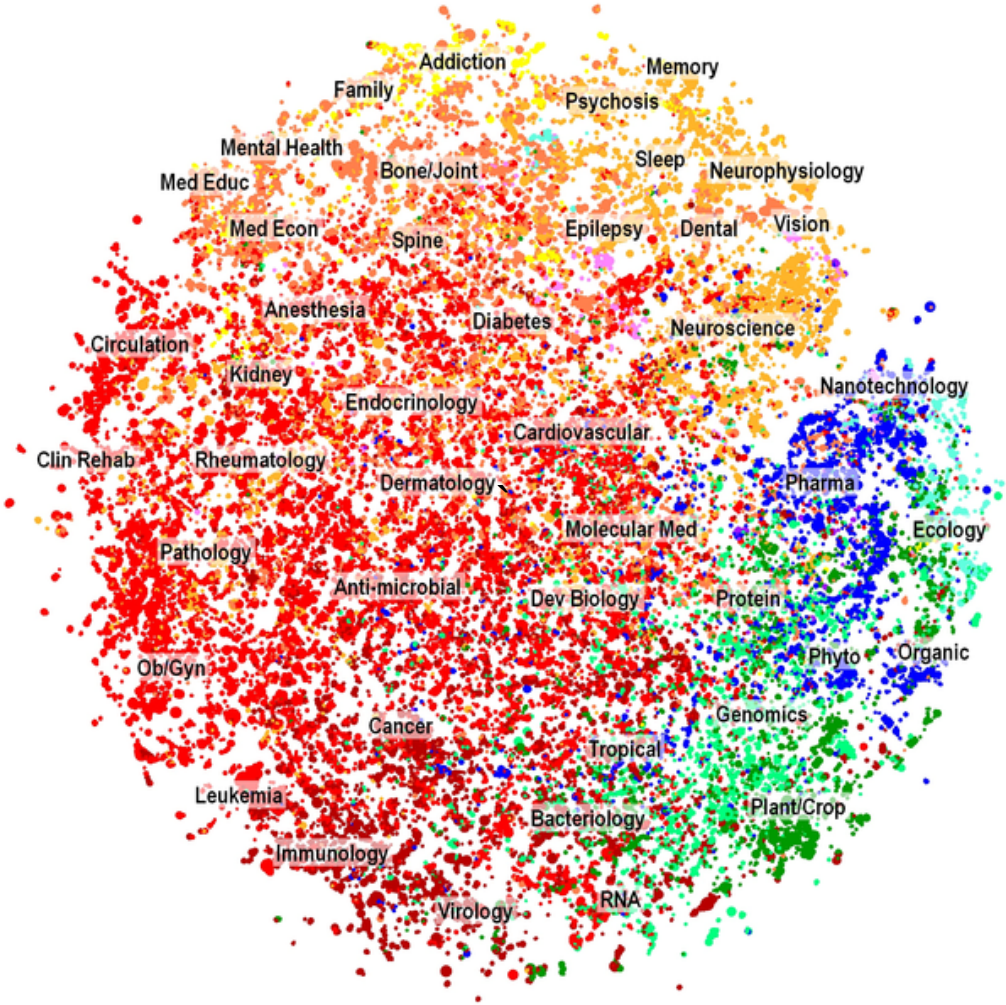
by TouchGraph

Clustering: Example

Given a collection of text documents, we want to organize them according to their content similarities, topic hierarchy, etc.

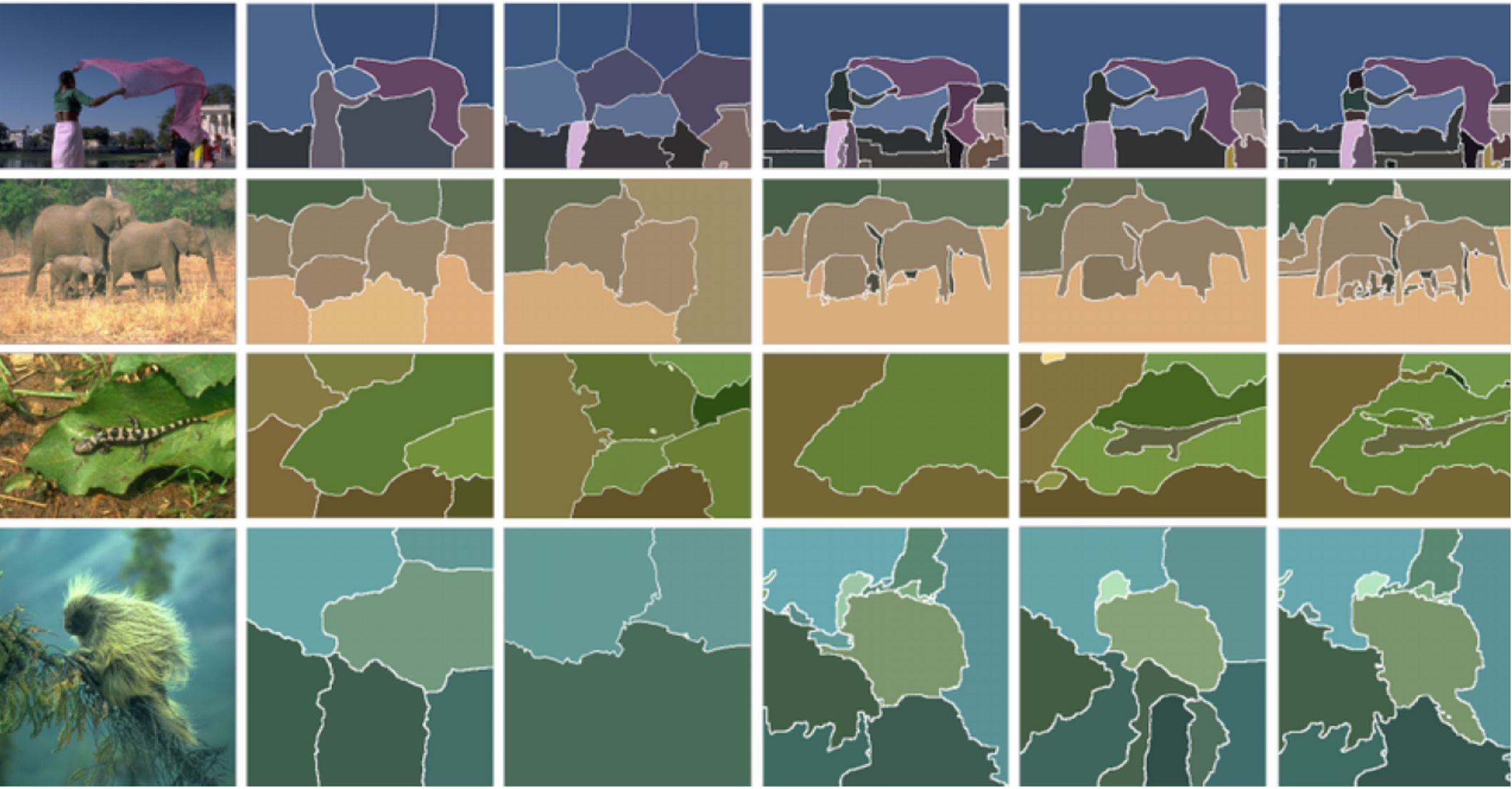
- Clustering **biomedical documents** into topics
  - Dataset: 2 millions documents on PubMed
  - Inputs: { title, keywords, abstracts}





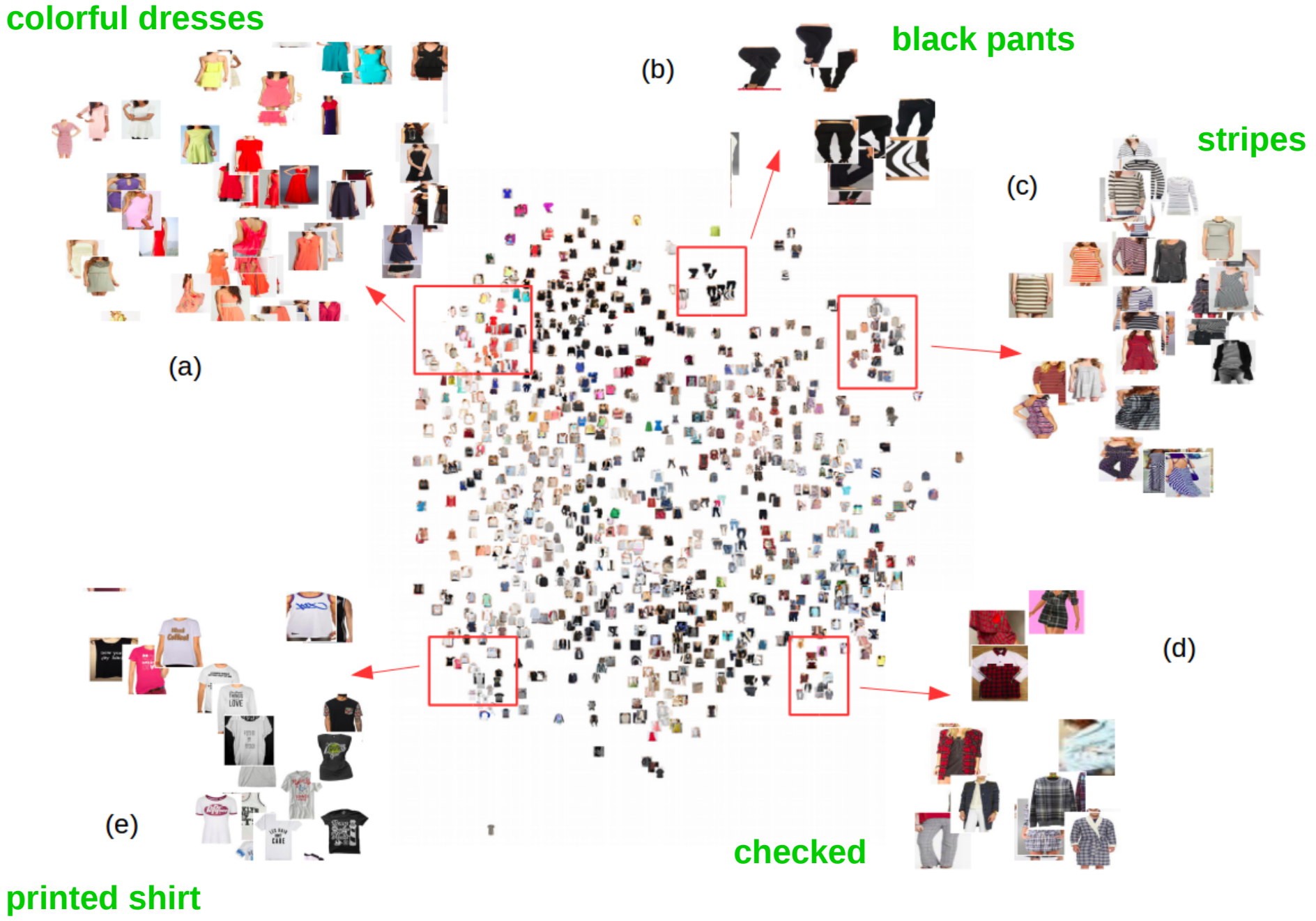
Clustering: Examples

- Computer vision: Image segmentation



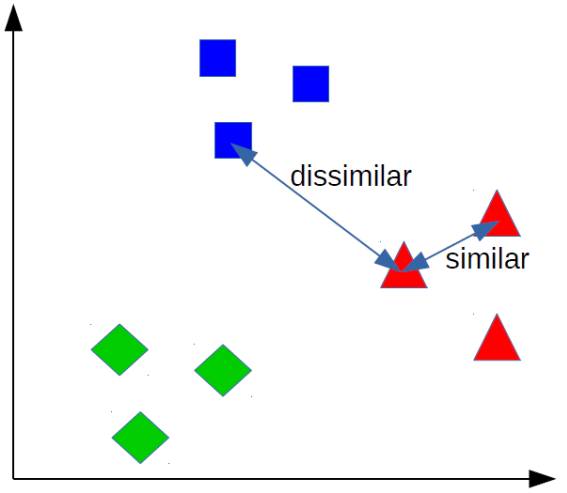
Yizhou Yu et. al., 2015

Clustering: Data analytics

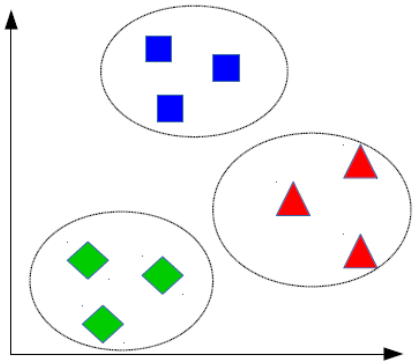


What is needed for clustering?

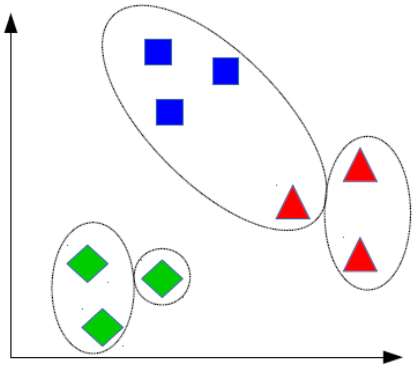
- Proximity measure **d(x, x')**:
  - distance between two points



- Criteria to evaluate a clustering:
  - e.g. good vs. bad (what objective to minimize?)



good clustering



bad clustering

- Algorithm to compute clustering
  - how to minimize the objective?

Outline

- Unsupervised Learning
- Clustering Problems
- **K-means Clustering Algorithm**

Some slides are from Bing Liu (UIC)

K-means clustering

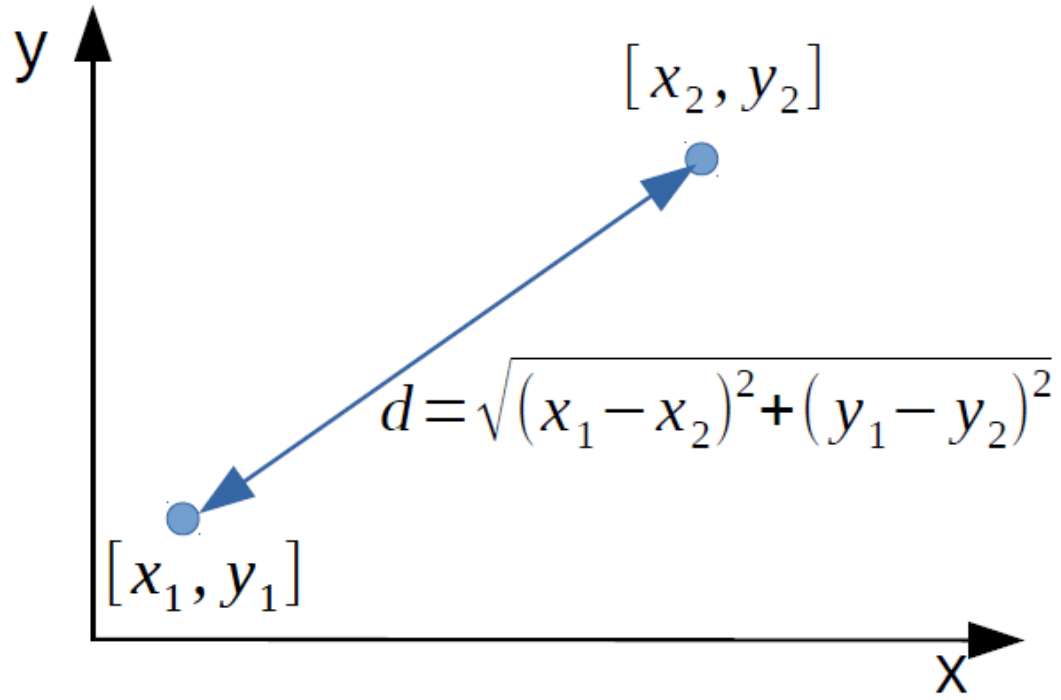
- K-means is a partitional clustering algorithm
- Let the set of data points (or instances) D be

  $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$,

  where $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{id})$ is a vector in a real-valued space $X \subseteq R^d$, and d is the number of variables (dimensions) in the data.
- The *k*-means algorithm partitions the given data into *k* clusters.
  - Each cluster has a cluster **center**, called **centroid**.
  - *k* is specified by the user (a hyper-parameter)

K-means Clustering

- a simple choice is **Euclidean distance**
  – Euclidean distance in 2D



– Euclidean distance in $\mathbb{R}^d$, for example $x \in \mathbb{R}^d$ and $x' \in \mathbb{R}^d$ where $x = [x_1, x_2, \ldots, x_d]^\top$, $x' = [x'_1, x'_2, \ldots, x'_d]^\top$

$$d(x, x') = \sqrt{(x_1 - x'_1)^2 + (x_2 - x'_2)^2 + \ldots + (x_d - x'_d)^2}$$

Non-Euclidean spaces

Here are some examples where a distance measure without a Euclidean space makes sense.

- Web pages: Roughly 10^8-dimensional space where each dimension corresponds to one word. Rather use vectors to deal with only the words actually present in documents a and b.
- Character strings, such as DNA sequences: Rather use a metric based on the LCS---Lowest Common Subsequence.
- Objects represented as sets of symbolic, rather than numeric, features: Rather base similarity on the proportion of features that they have in common.

Non-Euclidean spaces (cont'd)

object1 = {small, red, rubber, ball}

object2 = {small, blue, rubber, ball}

object3 = {large, black, wooden, ball}

similarity(object1, object2) = 3 / 4

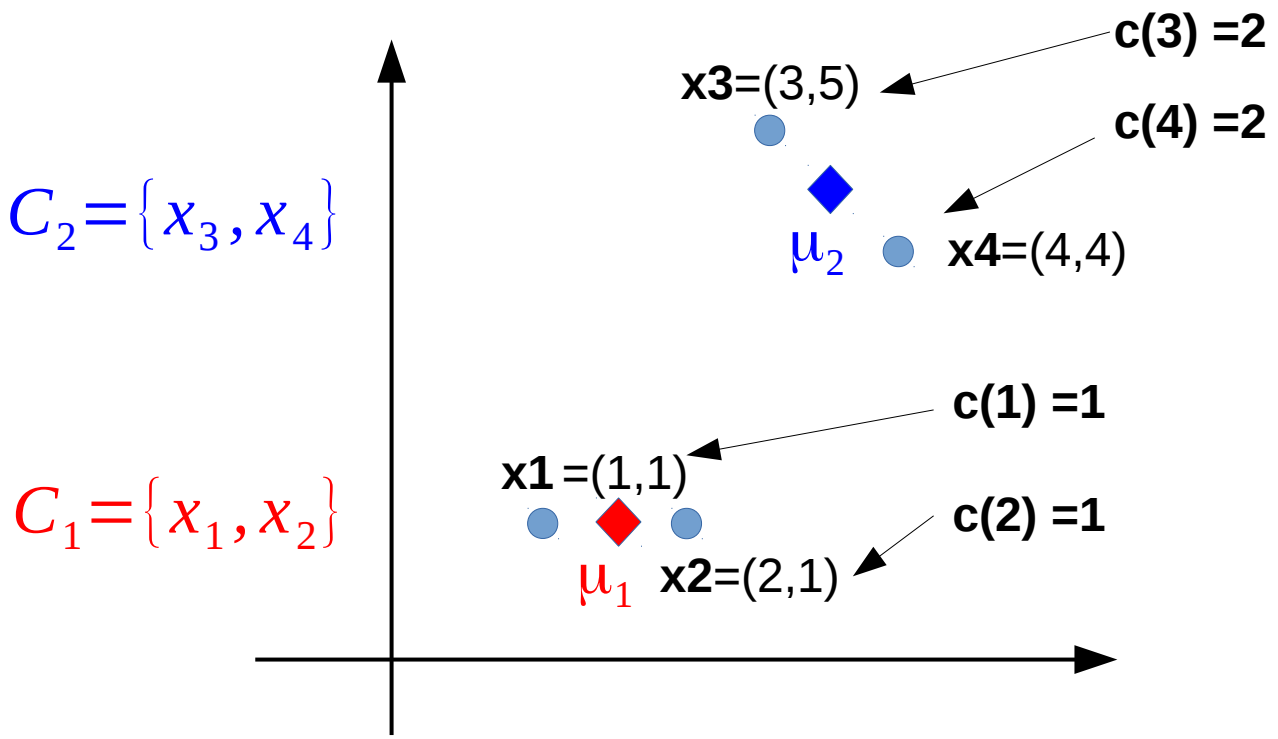
similarity(object1, object3) =
similarity(object2, object3) = 1/4

Note that it is possible to assign different weights to features.

K-means Clustering: Criteria

- given data $D = \{x_i\}_{i=1}^{n}$, where $x_i \in \mathbb{R}^d$
- objective: find K clusters with **centroids** $\mu_k \in \mathbb{R}^d$ $(k = 1, 2, \ldots, K)$
- denote a data assignment $c : i \mapsto k$ (assign data point i to cluster k)
- denote C_k: a set of all data points have assignment to cluster k

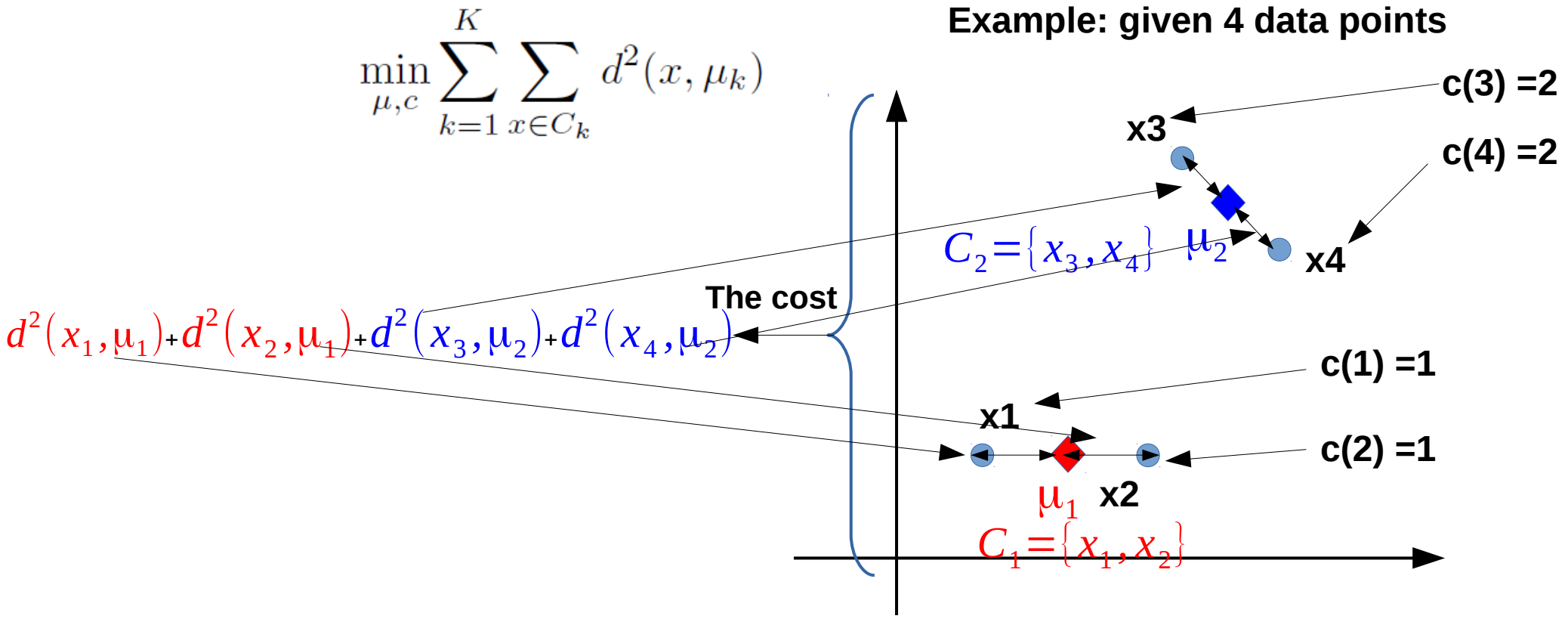
Example: given 4 data points



K-means Clustering: Criteria

- given data $D = \{x_i\}_{i=1}^{n}$, where $x_i \in \mathbb{R}^d$
- objective: find K clusters with **centroids** $\mu_k \in \mathbb{R}^d$ $(k = 1, 2, \ldots, K)$
- denote a data assignment $c : i \mapsto k$ (assign data point i to cluster k)
- denote C_k: a set of all data points have assignment to cluster k
- Criteria: minimize the within-cluster **sum of square errors (SSE)**

$$\min_{\mu,c} \sum_{k=1}^{K} \sum_{x \in C_k} d^2(x, \mu_k)$$



K-means Clustering: Algorithm

- Initialize: pick K data points **randomly** to initialize the centroids μ_k
- Iterate:
  – for each data point, **assign to a closest centroid**

$$\text{for each } i: \; c(i) = \operatorname*{argmin}_k d(x_i, \mu_k)$$

  – **change the centroid to the average** of its assigned points

$$\text{for each } k: \; \mu_k = \frac{1}{n_k} \sum_{x \in C_k} x$$

  where n_k is the number of elements in cluster k (members in C_k).

  – stop when **no new assignments**

K-means Clustering: Example

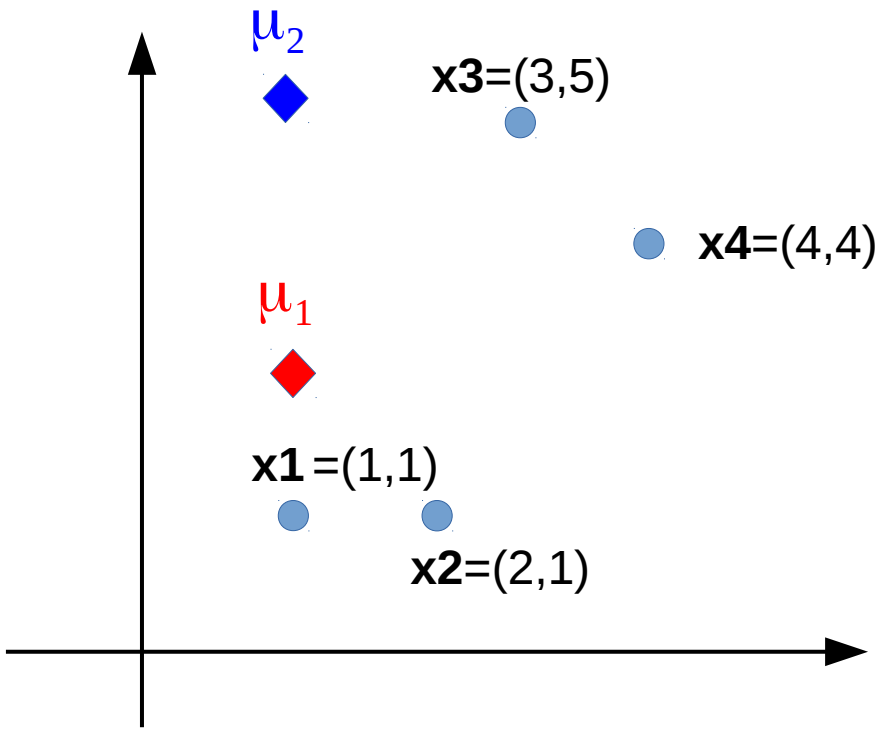
Initialize: (randomly)

$$\mu_1 = (1,2)$$
$$\mu_2 = (1,5)$$

Iteration 1:

– for each data point, **assign to a closest centroid**

Example: given 4 data points



Data point x1

$$d(x_1,\mu_1) = \sqrt{(x_{11}-\mu_{11})^2+(x_{12}-\mu_{12})^2} = \sqrt{(1-1)^2+(1-2)^2} = 1$$
$$d(x_1,\mu_2) = \sqrt{(x_{11}-\mu_{21})^2+(x_{12}-\mu_{22})^2} = \sqrt{(1-1)^2+(1-5)^2} = 4$$
$$c(1) = arg\,min\{d(x_1,\mu_1), d(x_1,\mu_2)\} = arg\,min\{1,4\} = 1$$

Data point x2

$$d(x_2,\mu_1) = \sqrt{(x_{21}-\mu_{11})^2+(x_{22}-\mu_{12})^2} = \sqrt{(2-1)^2+(1-2)^2} = \sqrt{2}$$
$$d(x_2,\mu_2) = \sqrt{(x_{21}-\mu_{21})^2+(x_{22}-\mu_{22})^2} = \sqrt{(2-1)^2+(1-5)^2} = \sqrt{17}$$
$$c(2) = arg\,min\{d(x_2,\mu_1), d(x_2,\mu_2)\} = arg\,min\{\sqrt{2},\sqrt{17}\} = 1$$

Data point x3

$$d(x_3,\mu_1) = ???$$
$$d(x_3,\mu_2) = ???$$
$$c(3) = ???$$

Data point x4

$$d(x_4,\mu_1) = ???$$
$$d(x_4,\mu_2) = ???$$
$$c(4) = ???$$

K-means Clustering: Criteria

Initialize: (randomly)

$$\mu_1 = (1,2)$$
$$\mu_2 = (1,5)$$

Iteration 1:

– for each data point, **assign to a closest centroid**

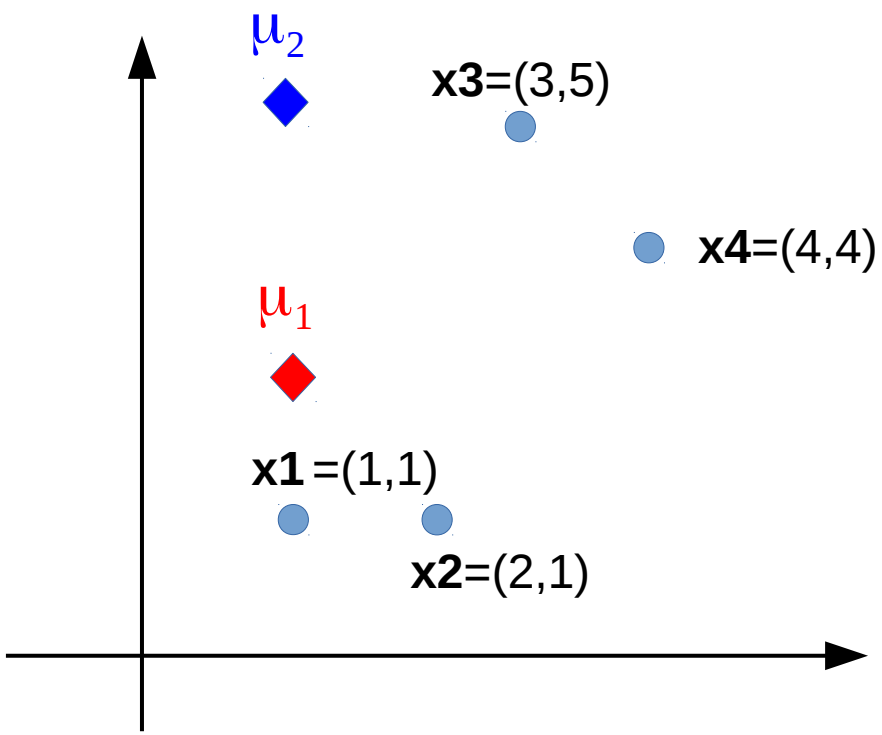
Data point x1

$$d(x_1,\mu_1) = \sqrt{(x_{11}-\mu_{11})^2+(x_{12}-\mu_{12})^2} = \sqrt{(1-1)^2+(1-2)^2} = 1$$
$$d(x_1,\mu_2) = \sqrt{(x_{11}-\mu_{21})^2+(x_{12}-\mu_{22})^2} = \sqrt{(1-1)^2+(1-5)^2} = 4$$
$$c(1) = arg\,max\{d(x_1,\mu_1), d(x_1,\mu_2)\} = arg\,max\{1,4\} = 1$$

Data point x2

$$d(x_2,\mu_1) = \sqrt{(x_{21}-\mu_{11})^2+(x_{22}-\mu_{12})^2} = \sqrt{(2-1)^2+(1-2)^2} = \sqrt{2}$$
$$d(x_2,\mu_2) = \sqrt{(x_{21}-\mu_{21})^2+(x_{22}-\mu_{22})^2} = \sqrt{(2-1)^2+(1-5)^2} = \sqrt{17}$$
$$c(2) = arg\,max\{d(x_1,\mu_1), d(x_1,\mu_2)\} = arg\,max\{\sqrt{2},\sqrt{17}\} = 1$$

Example: given 4 data points



Data point x3

$$d(x_3,\mu_1) = \sqrt{13}$$
$$d(x_3,\mu_2) = 2$$
$$c(3) = 2$$

Data point x4

$$d(x_4,\mu_1) = \sqrt{13}$$
$$d(x_4,\mu_2) = \sqrt{10}$$
$$c(4) = 2$$

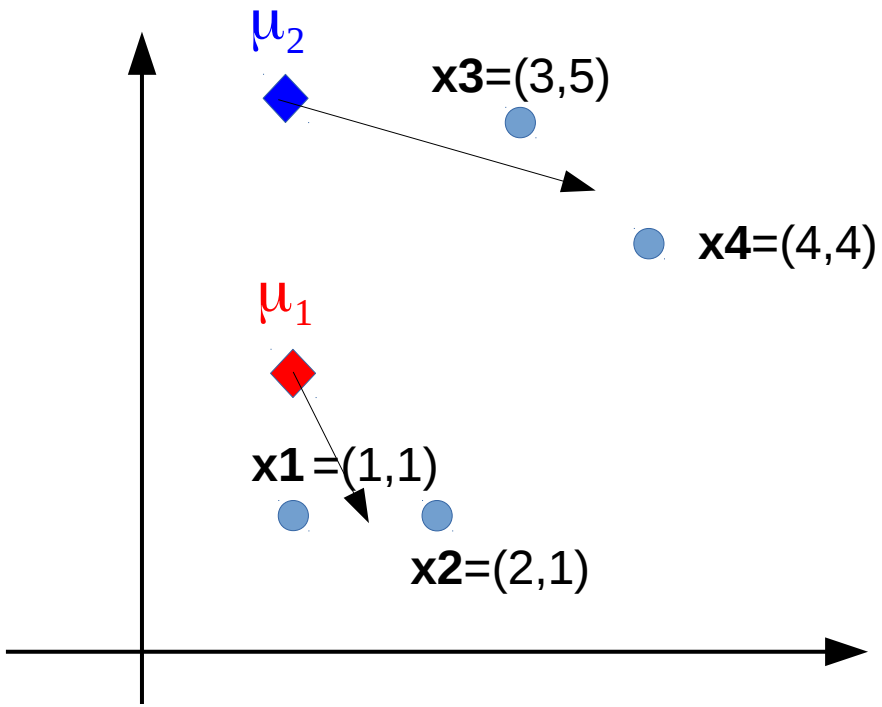
K-means Clustering: Criteria

Initialize: (randomly)

$$\mu_1 = (1,2)$$

$$\mu_2 = (1,5)$$

Example: given 4 data points



Iteration 1:

– for each data point, **assign to a closest centroid**

Data point x1 $c(1)=1$

Data point x2 $c(2)=1$

Data point x3 $c(3)=2$

Data point x4 $c(4)=2$

– **change the centroid to the average** of its assigned points

$$\text{for each } k: \ \mu_k = \frac{1}{n_k} \sum_{x \in C_k} x$$

$$\mu_1 = \frac{1}{2}\{x_1 + x_2\} = \frac{1}{2}(3,2) = (1.5,1)$$

$$\mu_2 = \frac{1}{2}\{x_3 + x_4\} = \frac{1}{2}(7,9) = (3.5,4.5)$$

where n_k is the number of elements in cluster k (members in C_k).

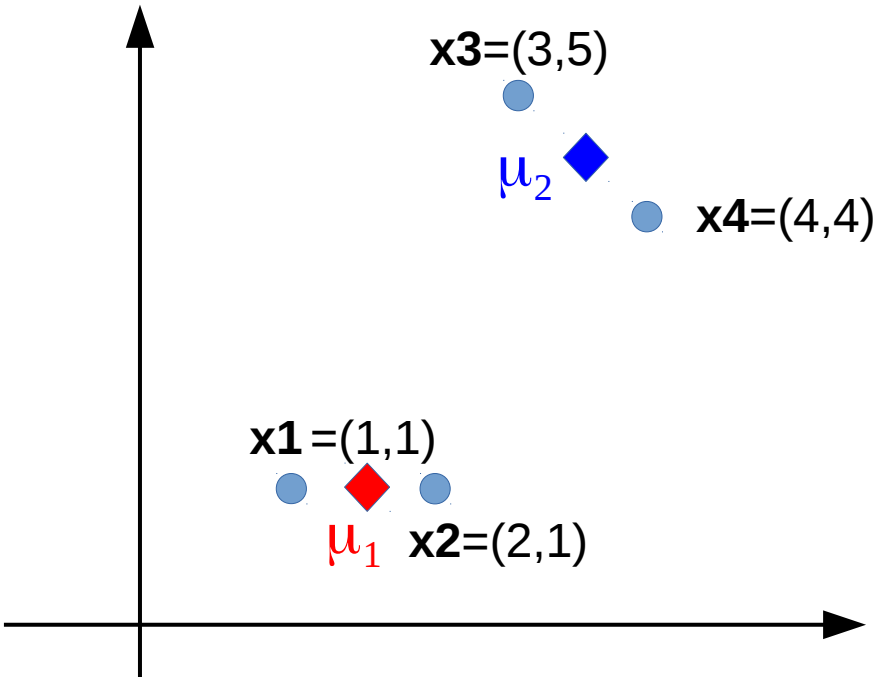
K-means Clustering: Criteria

Initialize: (randomly)

$$\mu_1 = (1,2)$$

$$\mu_2 = (1,5)$$

Example: given 4 data points



Iteration 1:

– for each data point, **assign to a closest centroid**

Data point x1 $c(1)=1$

Data point x2 $c(2)=1$

Data point x3 $c(3)=2$

Data point x4 $c(4)=2$

– **change the centroid to the average** of its assigned points

$$\text{for each } k: \ \mu_k = \frac{1}{n_k} \sum_{x \in C_k} x$$

$$\mu_1 = \frac{1}{2}\{x_1 + x_2\} = \frac{1}{2}(3,2) = (1.5, 1)$$

$$\mu_2 = \frac{1}{2}\{x_3 + x_4\} = \frac{1}{2}(7,9) = (3.5, 4.5)$$

where n_k is the number of elements in cluster k (members in C_k).

Strengths of k-means

- Strengths:
  - Simple: easy to understand and to implement
  - Efficient: Time complexity: $O(tkn)$,

    where n is the number of data points,

    k is the number of clusters, and

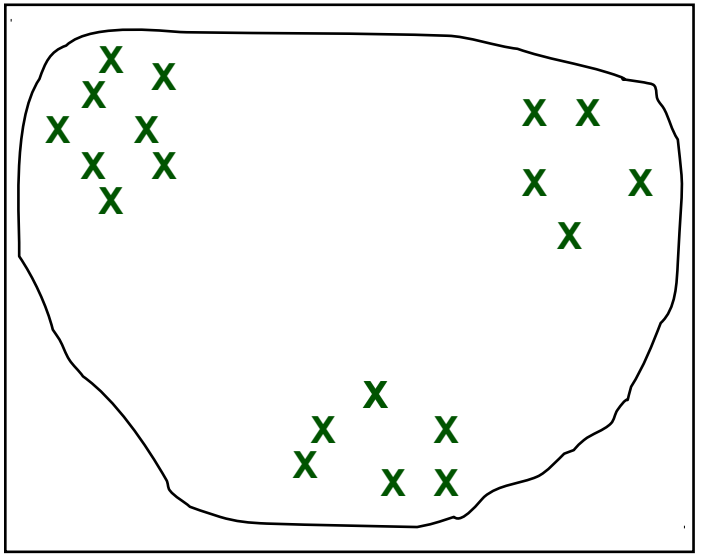
    t is the number of iterations.
  - Since both k and t are small. k-means is considered a linear algorithm.
- K-means is the most popular clustering algorithm.

Note that: it terminates at a local optimum if the sum of squared error (SSE) is used. The global optimum is hard to find due to complexity.

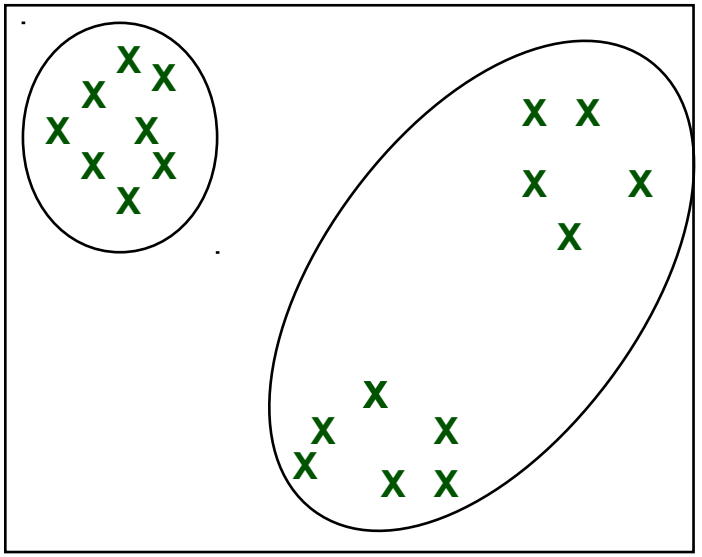
Weaknesses of k-means

- The algorithm is only applicable if the mean is defined.
  - For categorical data, *k*-mode - the centroid is represented by most frequent values.
- The user needs to specify *k*.
- The algorithm is sensitive to **outliers**
  - Outliers are data points that are very far away from other data points.
  - Outliers could be errors in the data recording or some special data points with very different values.

Specifying k

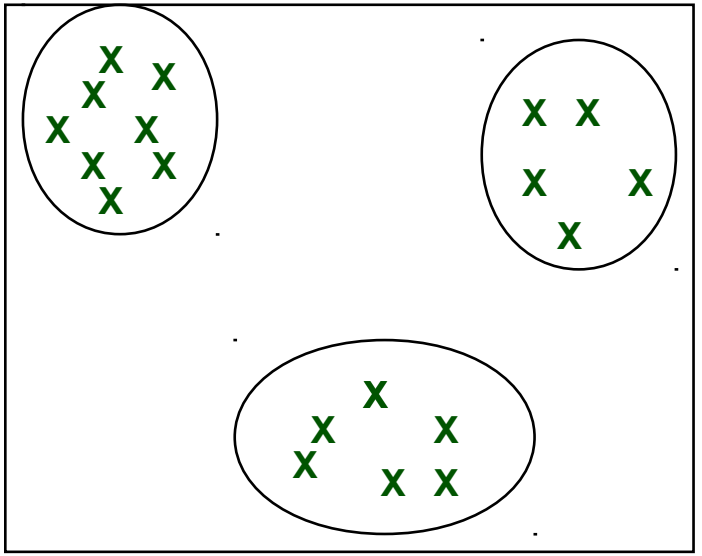


When *k* = 1, all the points are in one cluster, and the average distance to the centroid will be high.

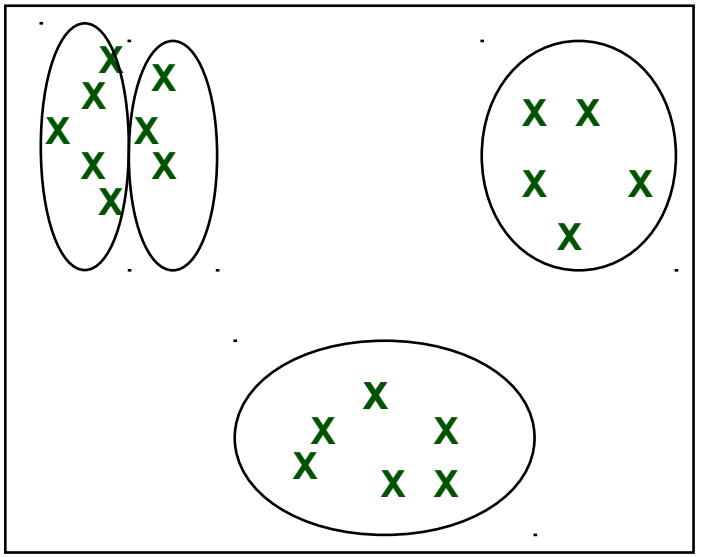


When *k* = 2, one of the clusters will be by itself and the other two will be forced into one cluster. The average distance of points to the centroid will shrink considerably.

Specifying k (cont'd)

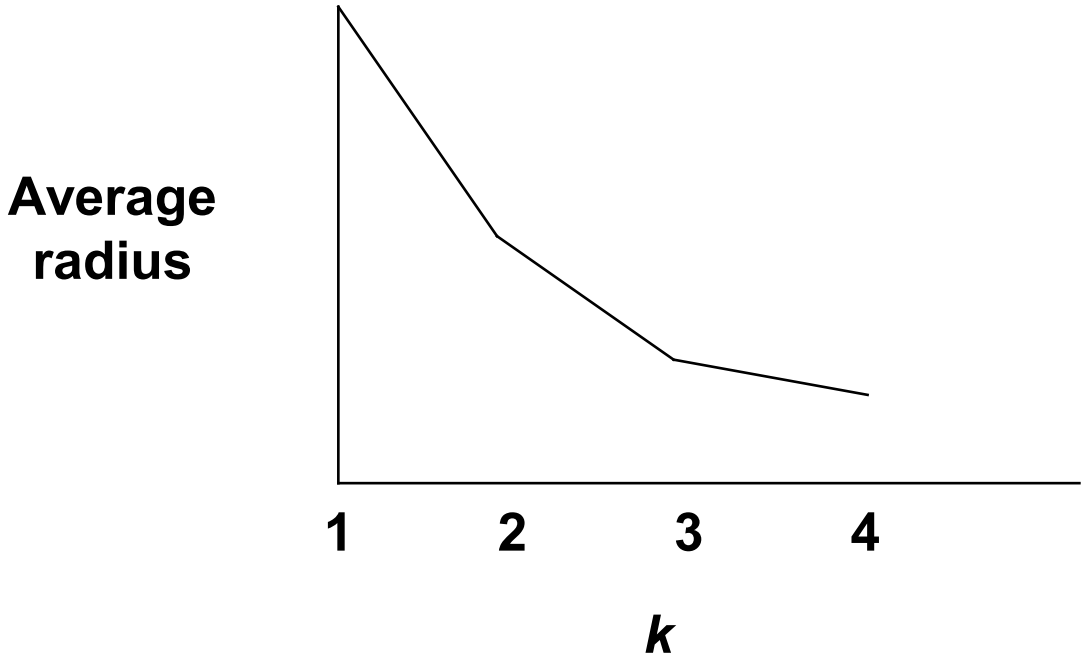


When *k* = 3, each of the apparent clusters should be a cluster by itself, and the average distance from the points to their centroids shrinks again.



When *k* = 4, then one of the true clusters will be artificially partitioned into two nearby clusters. The average distance to the centroids will drop a bit, but not much.

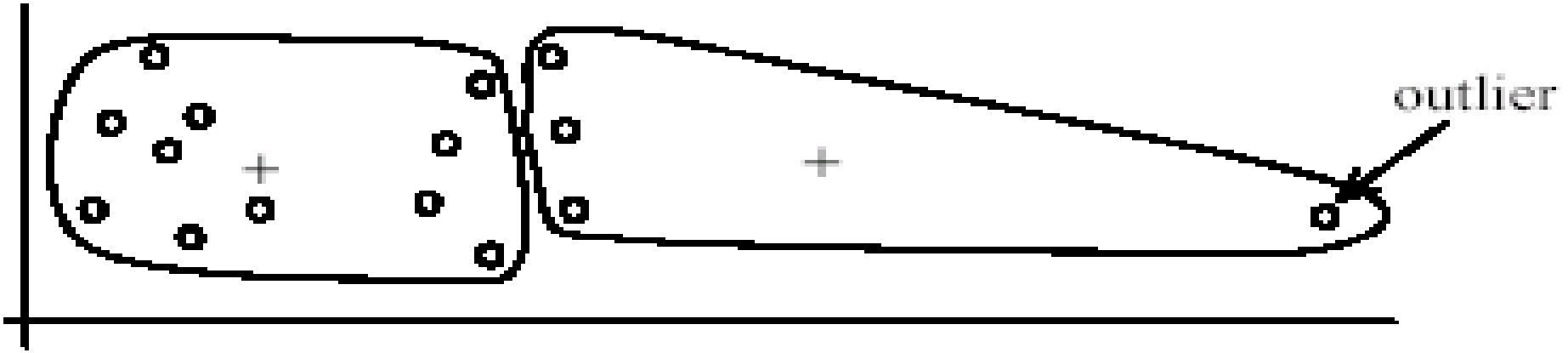
Specifying k (cont'd)



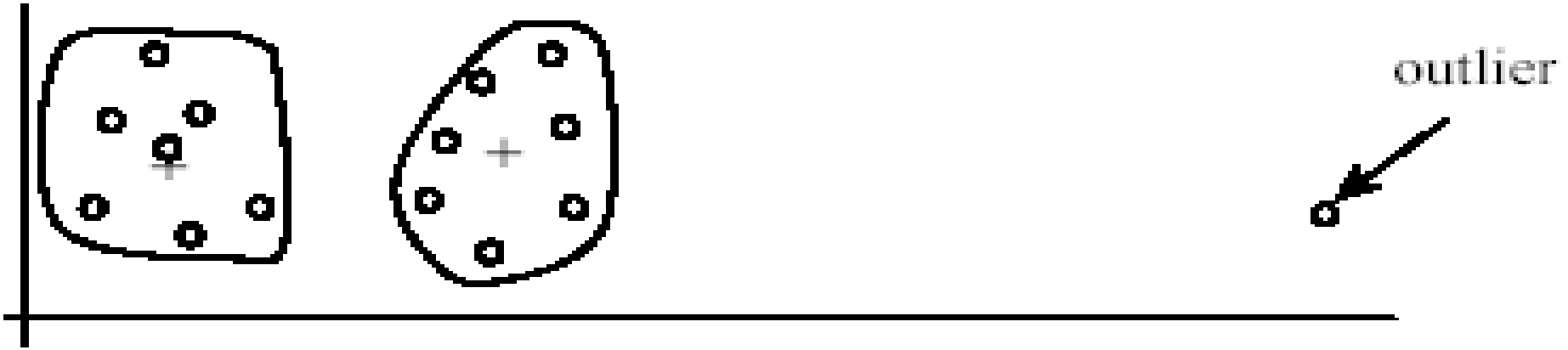
This failure to drop further suggests that k = 3 is right. This conclusion can be made even if the data is in so many dimensions that we cannot visualize the clusters.

Practical Labs will use cross-validation to determine k

Weaknesses of k-means: Problems with outliers



(A): Undesirable clusters



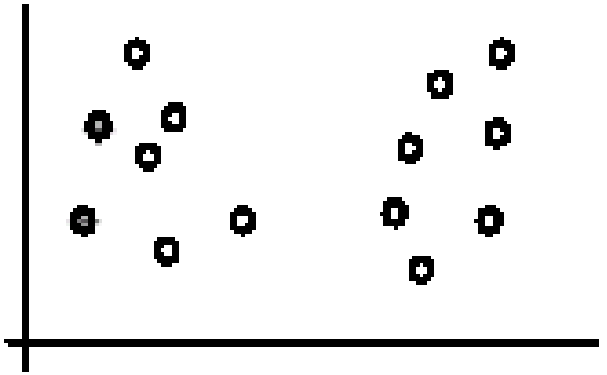
(B): Ideal clusters

Weaknesses of k-means: To deal with outliers

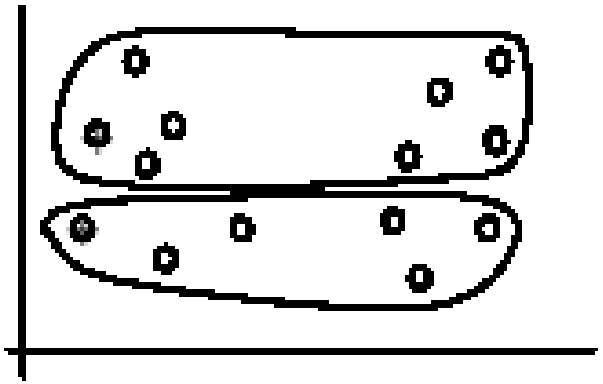
- One method is to remove some data points in the clustering process that are much further away from the centroids than other data points.
  - To be safe, we may want to monitor these possible outliers over a few iterations and then decide to remove them.
- Another method is to perform random sampling. Since in sampling we only choose a small subset of the data points, the chance of selecting an outlier is very small.
  - Assign the rest of the data points to the clusters by distance or similarity comparison, or classification

Weaknesses of k-means (cont …)

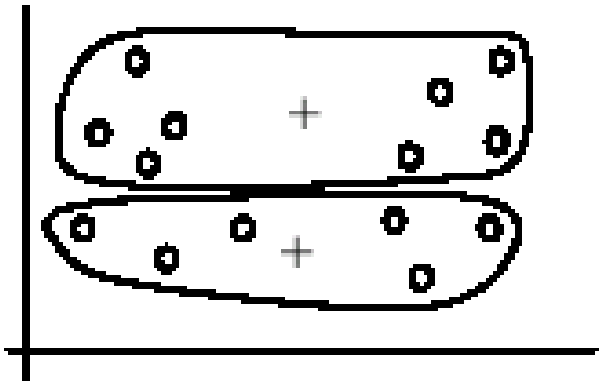
- The algorithm is sensitive to initial seeds.



(A). Random selection of seeds (centroids)



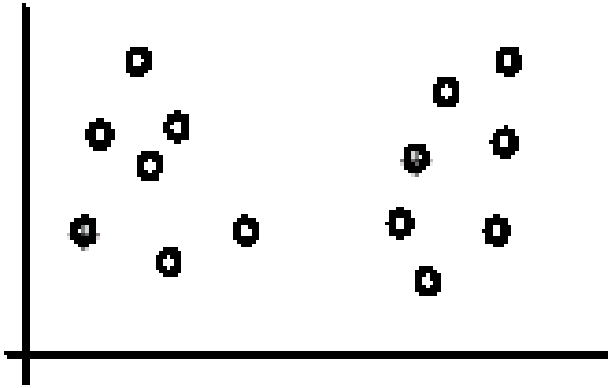
(B). Iteration 1



(C). Iteration 2

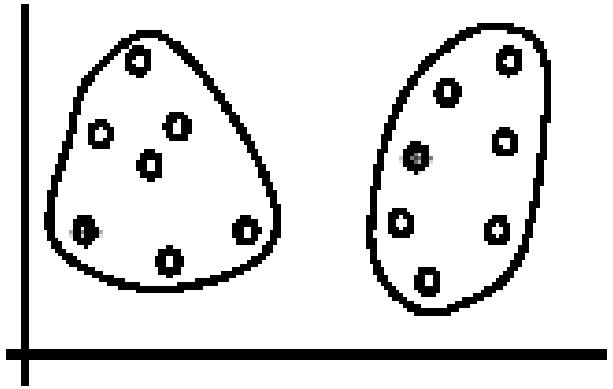
Weaknesses of k-means (cont …)

- If we use different seeds: good results

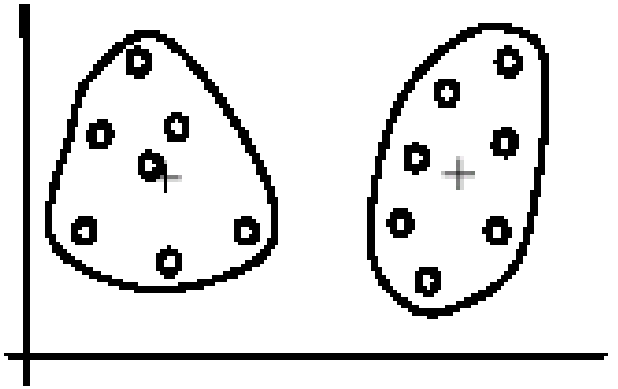


(A). Random selection of k seeds (centroids)

- There are some methods to help choose good seeds



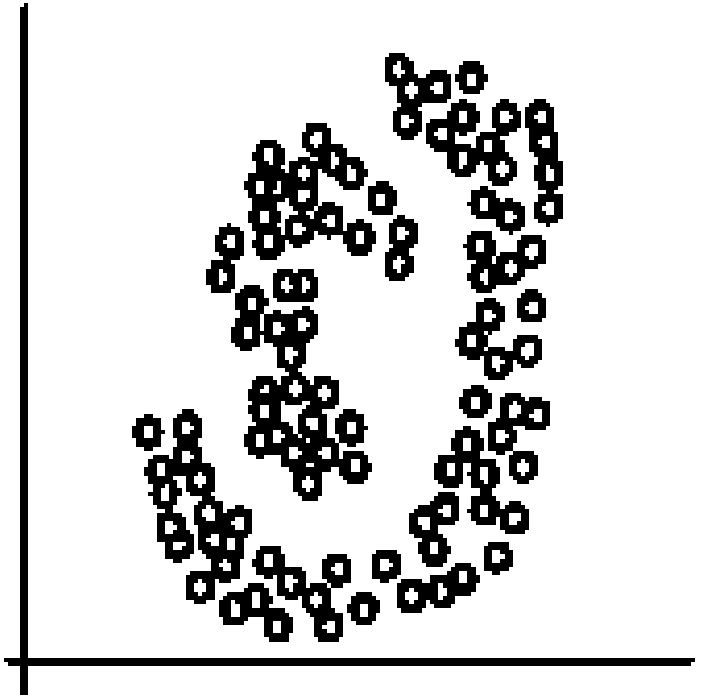
(B). Iteration 1



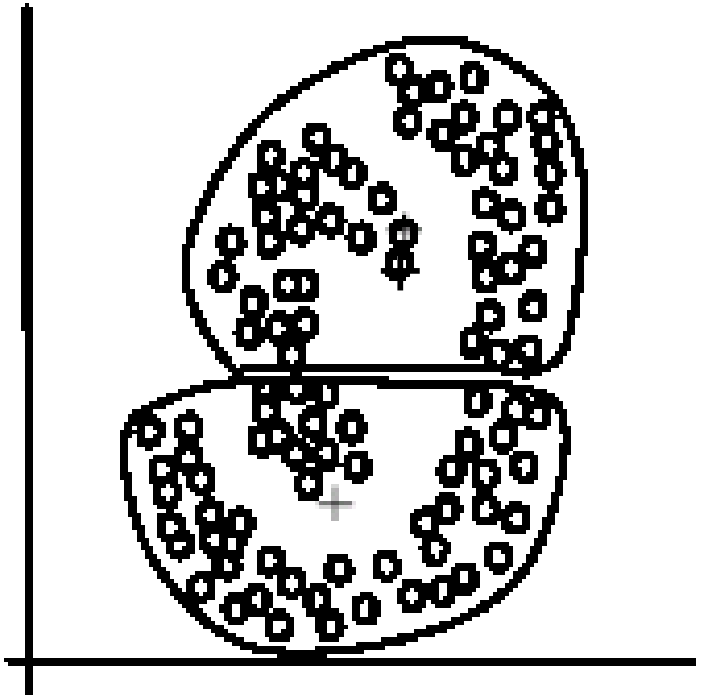
(C). Iteration 2

Weaknesses of k-means (cont …)

- The *k*-means algorithm is not suitable for discovering clusters that are not hyper-ellipsoids (or hyper-spheres).



(A): Two natural clusters



(B): *k*-means clusters

Must learn a problem-dependent metric (to measure data similarity)
Topic in ML: Manifold Learning

K-means summary

- Despite weaknesses, *k*-means is still the most popular algorithm due to its simplicity, efficiency and
  - other clustering algorithms have their own lists of weaknesses.
- No clear evidence that any other clustering algorithm performs better in general
  - although they may be more suitable for some specific types of data or applications.
- Comparing different clustering algorithms is a difficult task. No one knows the correct clusters!

Summary

- Unsupervised learning is a branch of machine learning that learns from data that has not been labeled, classified or categorized.
- k-means methods: intuitive, easy to understand and implement
- k-means method: still the most popular clustering algorithm
- k-means: Converges to local minimum (sensitive to initialization of centroids) → many restarts
- Sensitive to the choice of k? – choose a tradeoff between model complexity (large k) and data fit (small loss) → cross-validation
- Clustering is an active topic in ML, and highly dependent on applications