Data Clustering and Dimensionality Reduction

CS677

Instructor: Thirimachos Bourlai

Date: February 2020

Part 1 - Clustering

- Focus on
  - K-Means Clustering
- Other
  - Mixture Models
  - Hierarchical Clustering

Supervised vs. Unsupervised Learning

Definitions

- In pattern recognition, data analysis is concerned with predictive modeling: given some training data, we want to predict the behavior of the unseen test data. This task is also referred to as learning.

- Often, a clear distinction is made between learning problems that are:

  (i) **Supervised** (classification) or

  (ii) **Unsupervised** (clustering), the first involving only labeled data (training patterns with known category labels) while the latter involving only unlabeled data (Duda et al., 2001).

Semi - Supervised Learning

There is a growing interest in a hybrid setting, called **semi-supervised learning** (Chapelle et al., 2006):

- In semi-supervised classification, the **labels** of only a **small portion** of the training data set are available.
- The **unlabeled data**, instead of being discarded, are also used in the learning process.

In **semi-supervised clustering**, instead of specifying the class labels, pair-wise constraints are specified, which is a weaker way of encoding the prior knowledge.

- A pair-wise **must-link constraint** corresponds to the requirement that two objects should be assigned the same cluster label
- The cluster labels of two objects participating in a **cannot-link constraint** should be different.

Constraints can be **particularly beneficial in data clustering** (Lange et al., 2005; Basu et al., 2008), where precise definitions of underlying clusters are absent.

Unsupervised Learning / Clustering

- **Motivation**
  - We are not looking of a precise result **but**
    - A better understanding or a better looking at the data at hand
- **Goal:** The goal of data clustering (cluster analysis), is to discover the natural grouping(s) of a set of patterns, points, or objects
- **Big Data Problem**
  - Unlimited data to work with!
- **Solution to the Problem**: reduce data
- **How?**

> **Clustering**............................: reducing the # of examples

> **Dimensionality Reduction**: reducing the # of dimensions

Unsupervised Learning / Clustering

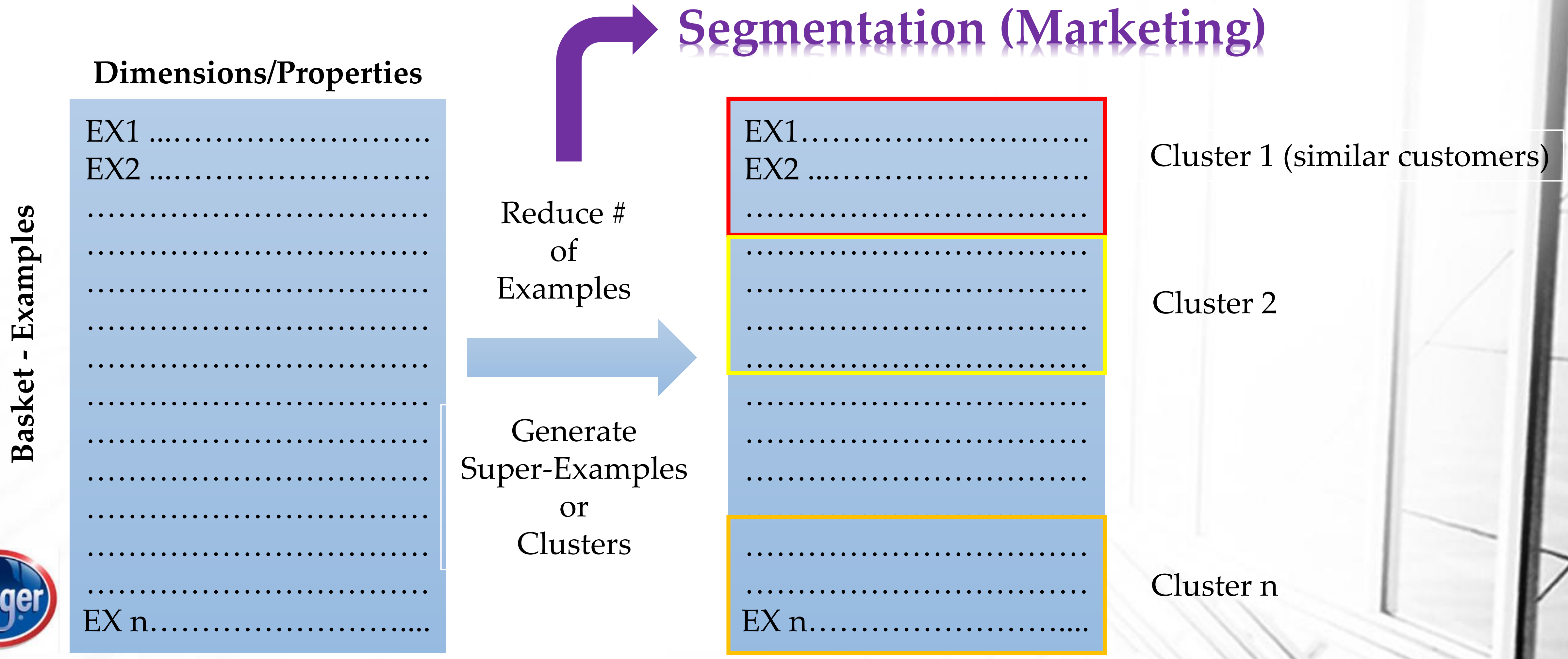
- An <u>operational definition of clustering</u> can be stated as follows:

Given a representation of n objects, find K groups based on a measure of similarity <u>such that</u>

- The *similarities between objects in the* ***same group*** *are high*
- *The similarities between objects in* ***different groups*** are low.

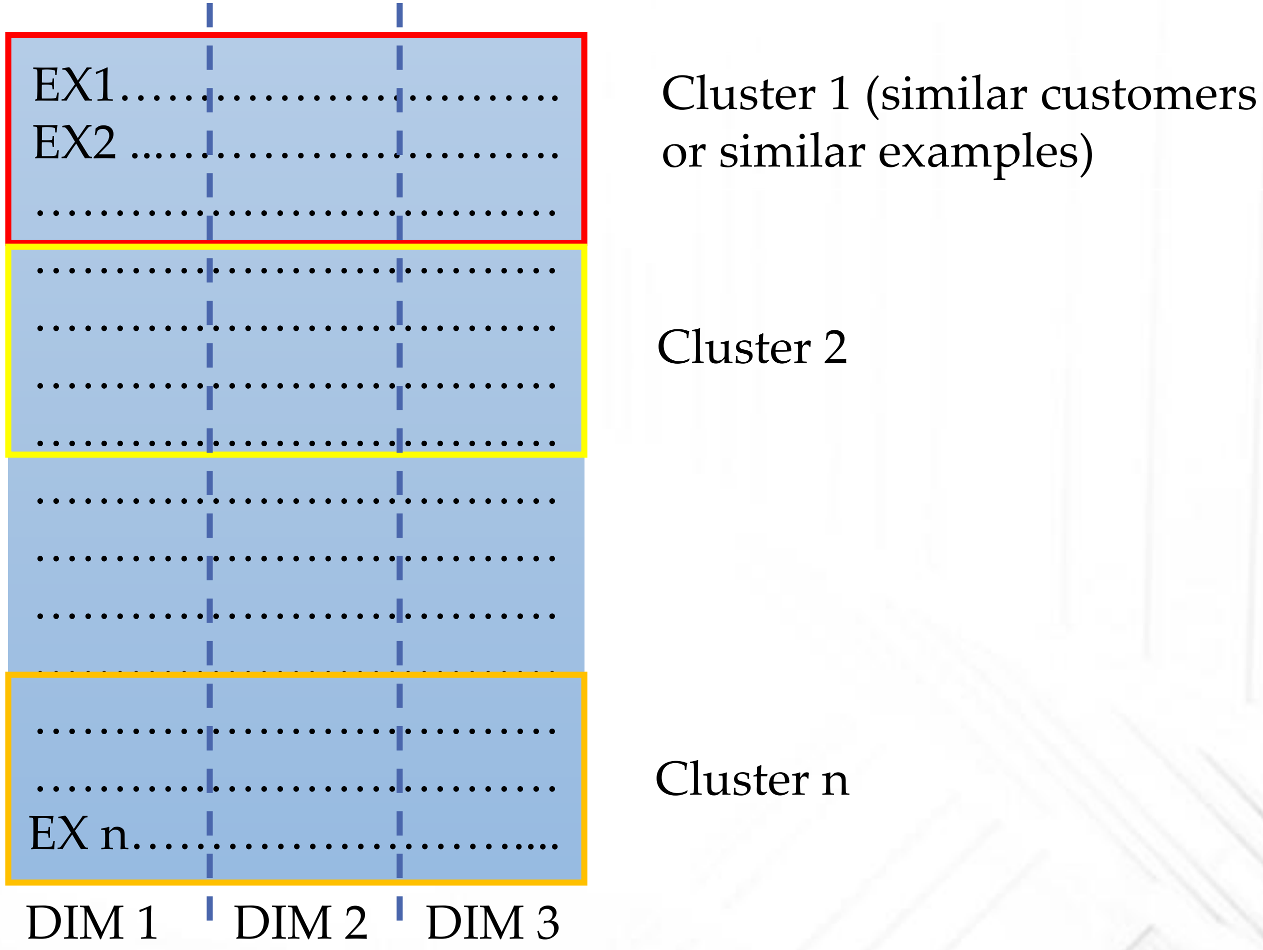
But, what is the notion of similarity?

Example


Segmentation (Marketing)
Dimensions/Properties
Basket - Examples
EX1 ……………………………
EX2 ……………………………
EX n……………………………
Reduce # of Examples
Generate Super-Examples or Clusters
EX1……………………………
EX2 ……………………………
EX n……………………………
Cluster 1 (similar customers)
Cluster 2
Cluster n

Example – Cluster within Dimensions

Segmentation (Marketing)



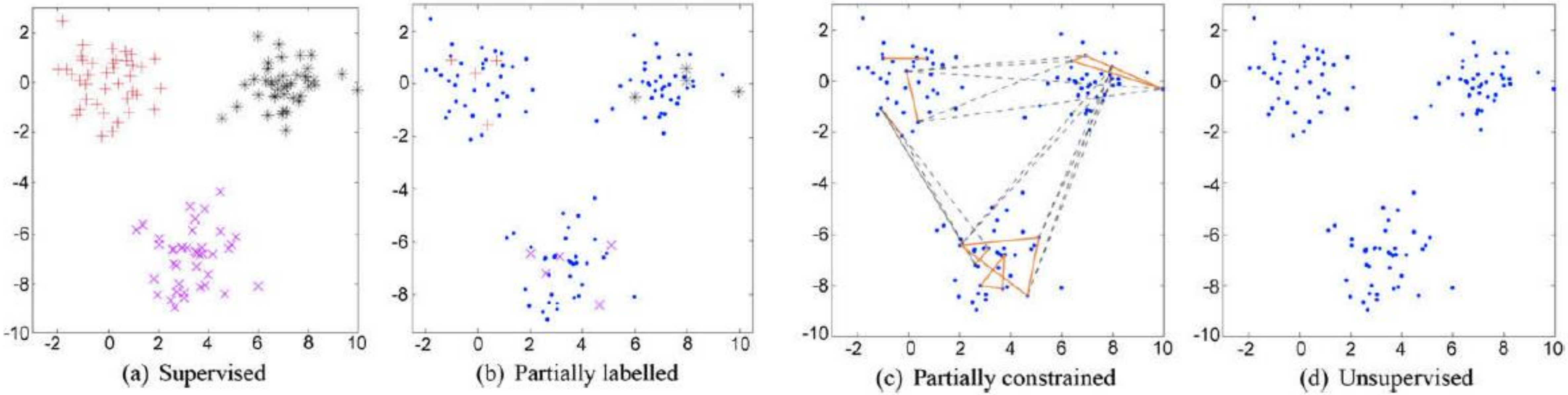
Clustering

Step 1: Have a set of samples A

Step 2: Divide the set of samples A into sub

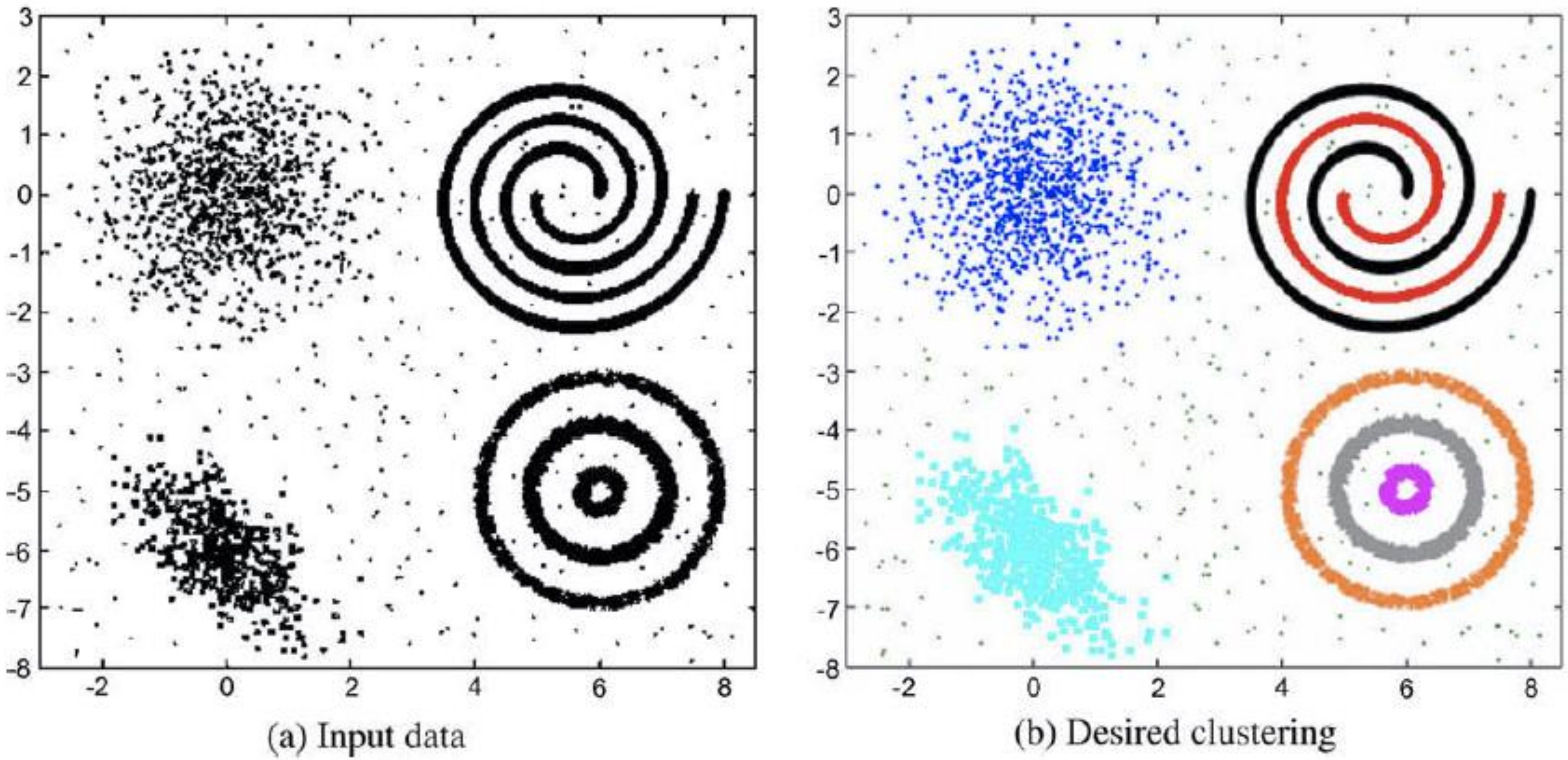
Challenges

- How do we measure similarity?
  - Euclidean Distance or something else? > Problem dependent
- Performance evaluations: how do we evaluate the quality of the results?
  - **Supervised** > clear decision, e.g. make predictions and see whether we are correct
  - **Unsupervised** > fuzzy decision, e.g. # clusters from the marketing perspective vs. the engineering perspective > affects code selection, K selection etc.
    - **Key – understand the phenomenon – problem domain**



- Learning problems: dots correspond to points without any labels
- Points with labels are denoted by plus signs, asterisks, and crosses.
- In (c), the must-link and cannot-link constraints are denoted by solid and dashed lines, respectively (figure taken from Lange et al. (2005).

Diversity of Clusters



Diversity of clusters. clusters in (a) (denoted by different colors in 1(b)) differ in shape, size, and density. Although these clusters are apparent to a data analyst, none of the available clustering algorithms can detect <u>all these clusters</u>.

Why Data Clustering is used?

- Data clustering has been used for the following three main purposes:

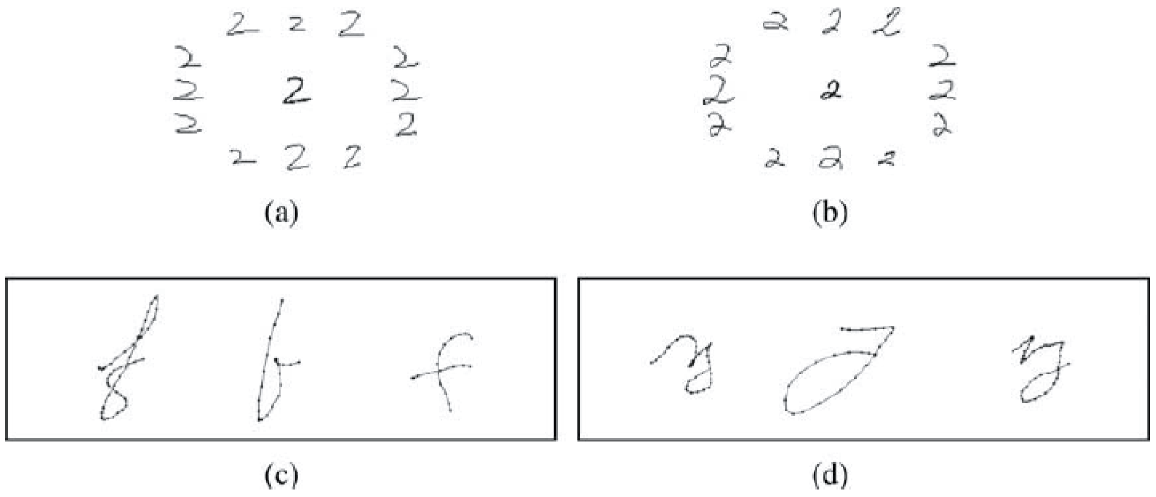
1) **Underlying structure**: to gain insight into data, generate hypotheses, detect anomalies, and identify salient features

2) **Natural classification**: to identify the degree of similarity among forms or organisms (phylogenetic relationship)

3) **Compression**: as a method for organizing the data and summarizing it through cluster prototypes.

Example from Class Discovery

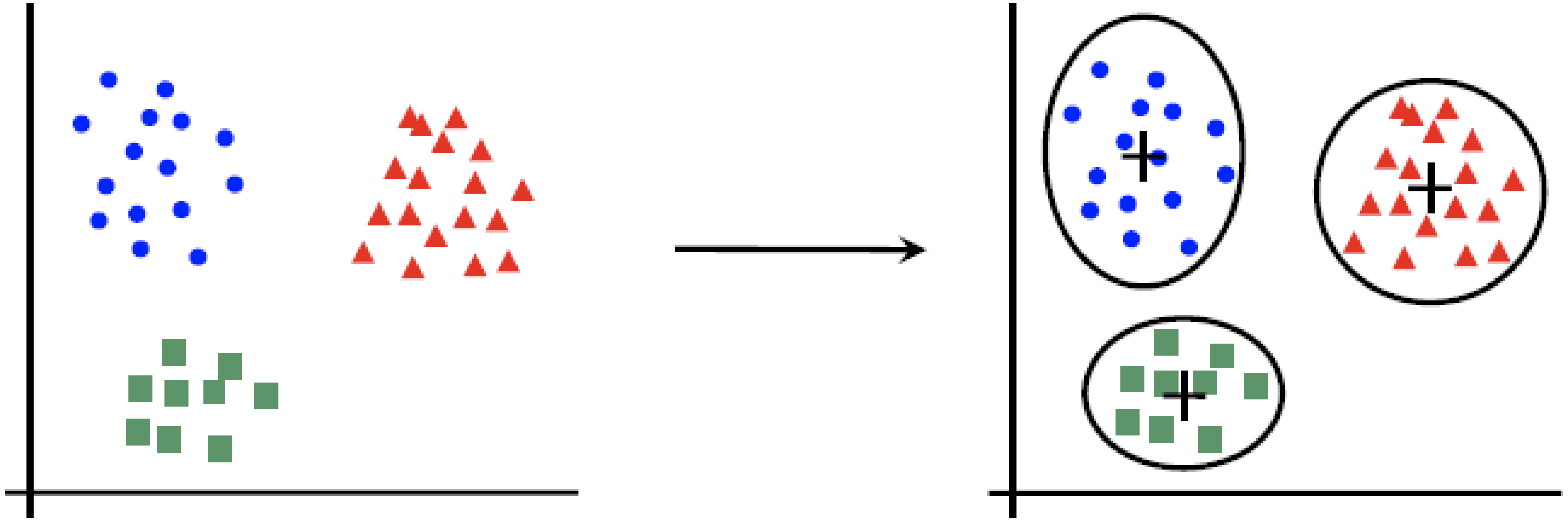
- An example of class discovery is shown in Fig. below.
- Here, clustering was used to discover subclasses in an online handwritten character recognition application (Connell and Jain, 2002)
- Different users write the same digits in different ways, thereby increasing the within-class variance.
- Clustering the training patterns from a class can discover new subclasses, called the lexemes in handwritten characters. Instead of using a single model for each character, multiple models based on the number of subclasses are used to improve the recognition accuracy



(a) (b) (c) (d)

K-Means

- clustering



K-means algorithm

Partition data into K sets

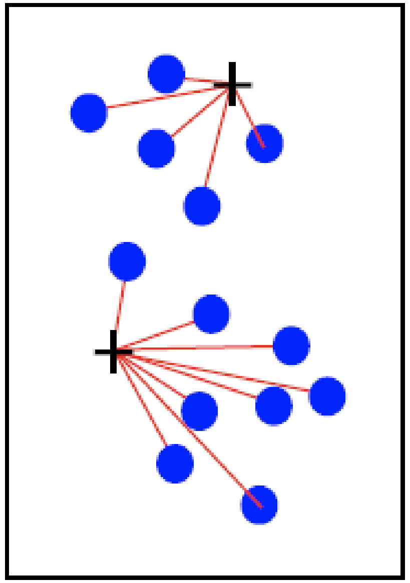
- Initialize: choose K centres (at random)

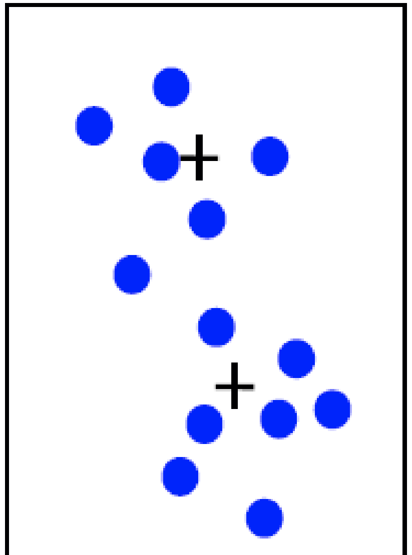
- Repeat:

  1. Assign points to the nearest centre

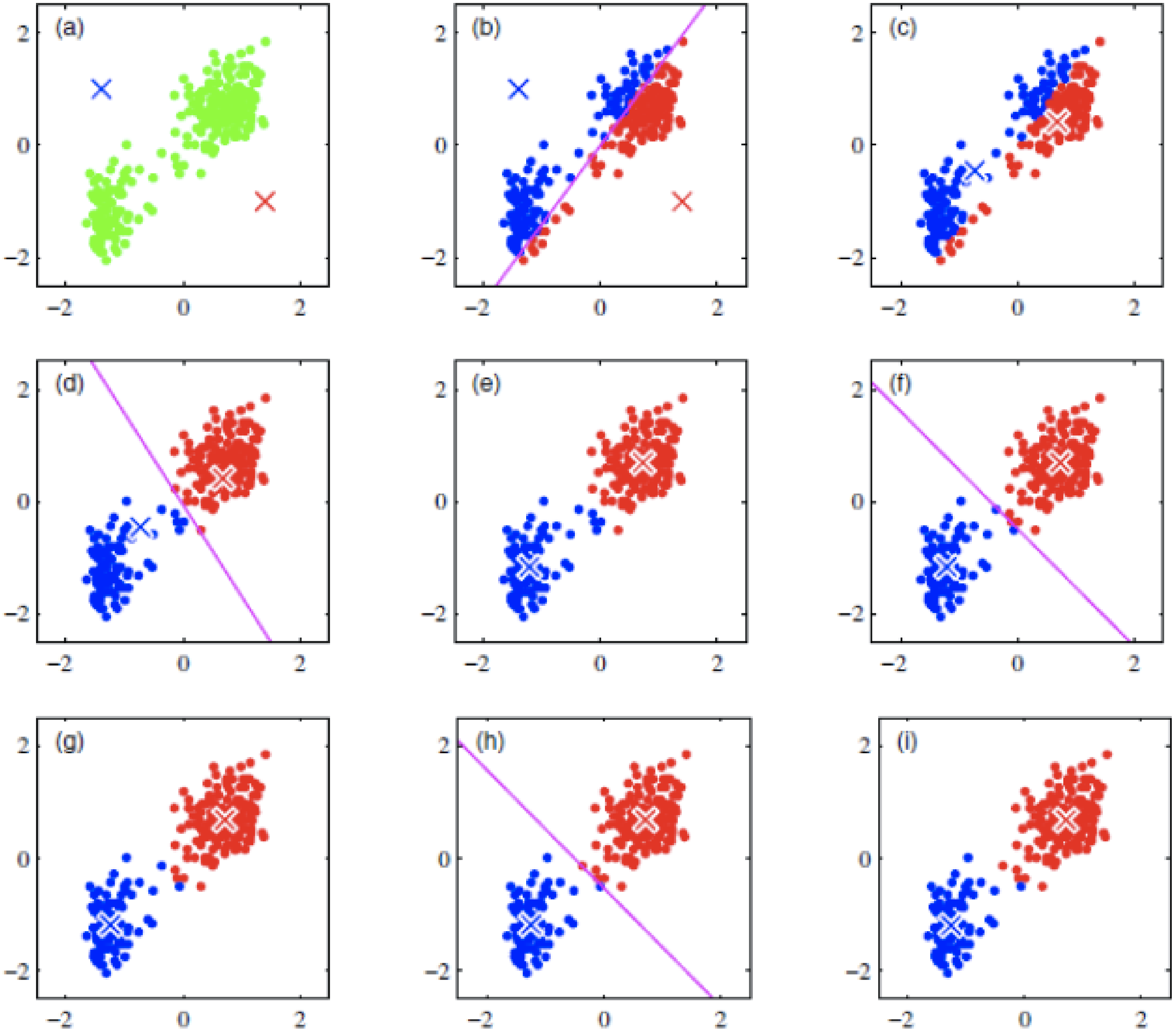
  2. New centre = mean of points assigned to it

- Until no change





Example



Cost function

K-means minimizes a measure of distortion for a set of vectors $\{\mathbf{x}_i\}, i = 1, \dots, N$

$$\mathrm{D} = \sum_{i=1}^{N} \|\mathbf{x}_i^k - \mathbf{c}_k\|^2$$

where $\mathbf{x}_i^k$ is the subset assigned to the cluster k. The objective is to find the set of centres $\{\mathbf{c}_k\}, k = 1, \dots, K$ that minimize the distortion:

$$\min_{\mathbf{c}_k} \sum_{i=1}^{N} \|\mathbf{x}_i^k - \mathbf{c}_k\|^2$$

Introducing binary assignment variables r_{ik}, the distortion can be written as

$$\mathrm{D} = \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \|\mathbf{x}_i - \mathbf{c}_k\|^2$$

where if $\mathbf{x}_i$ is assigned to cluster k then

$$r_{ij} = \begin{cases} 1 & j = k \\ 0 & j \neq k \end{cases}$$

Minimizing the Cost function

We want to determine

$$\min_{\mathbf{c}_k, r_{ik}} D = \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \|\mathbf{x}_i - \mathbf{c}_k\|^2$$

Step 1: minimize over assignments r_{ik}

Each term in $\mathbf{x}_i$ can be minimized independently by assigning $\mathbf{x}_i$ to the closest centre $\mathbf{c}_k$

Step 2: minimize over centres $\mathbf{c}_k$

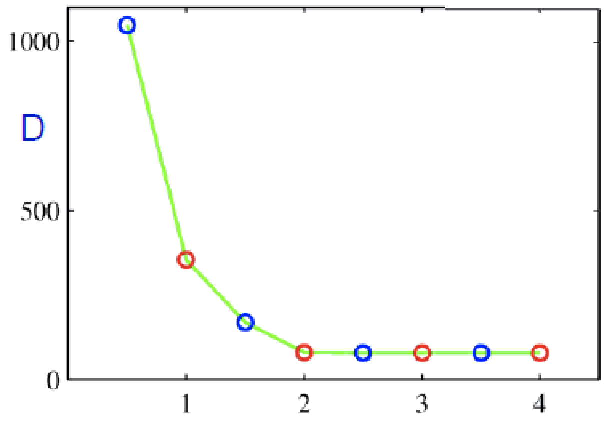
$$\frac{d}{d\mathbf{c}_k} \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \|\mathbf{x}_i - \mathbf{c}_k\|^2 = 2 \sum_{i=1}^{N} r_{ik}(\mathbf{x}_i - \mathbf{c}_k) = \mathbf{0}$$

Hence

$$\mathbf{c}_k = \frac{\sum_{i=1}^{N} r_{ik}\mathbf{x}_i}{\sum_{i=1}^{N} r_{ik}}$$

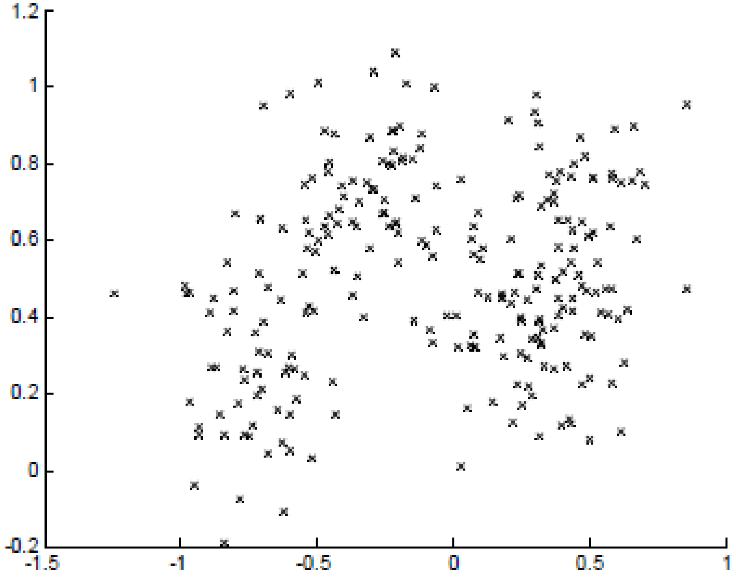
i.e. $\mathbf{c}_k$ is the mean (centroid) of the vectors assigned to it.

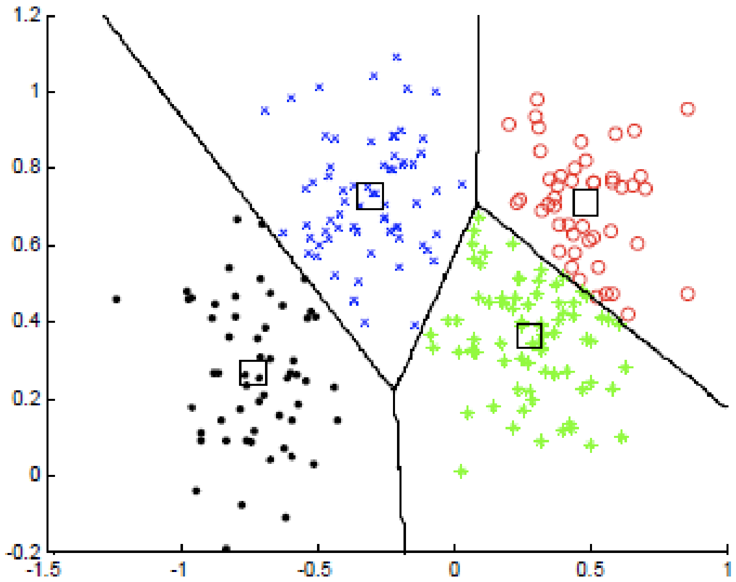
Decrease in distortion cost with iterations

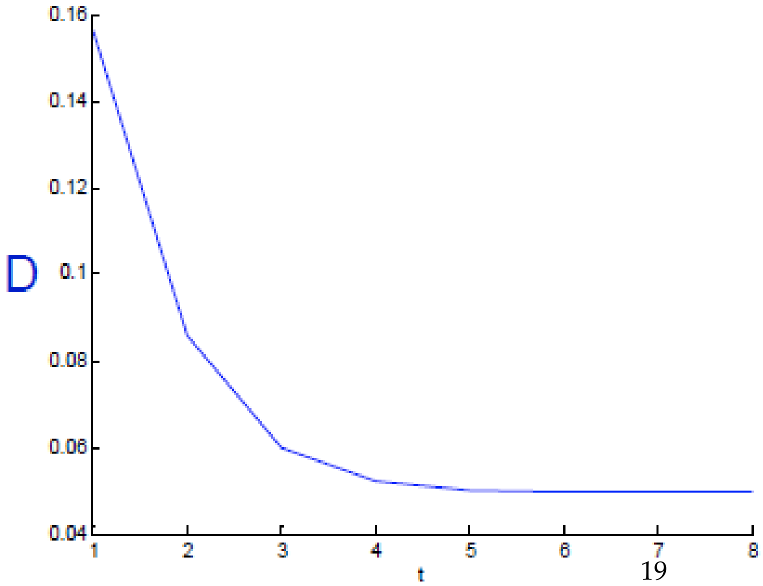


Note, since both steps decrease the cost D, the algorithm will converge – but it can converge to a local rather than global minimum.

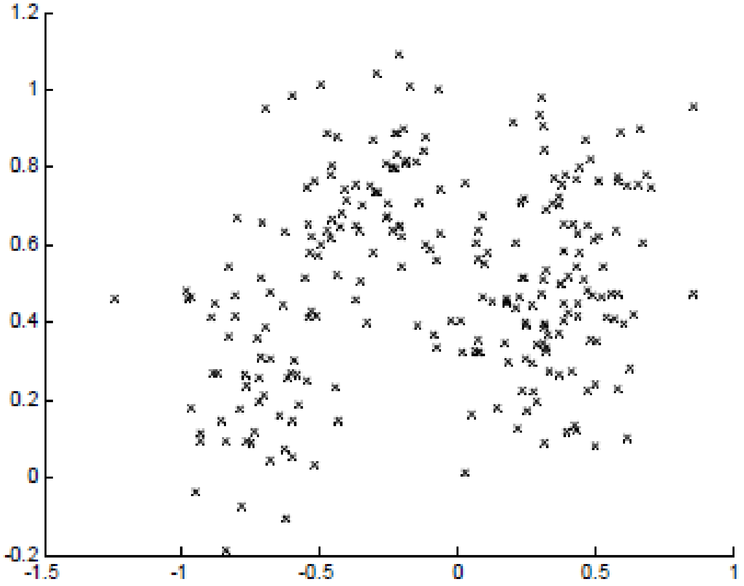
Sensitive to initialization

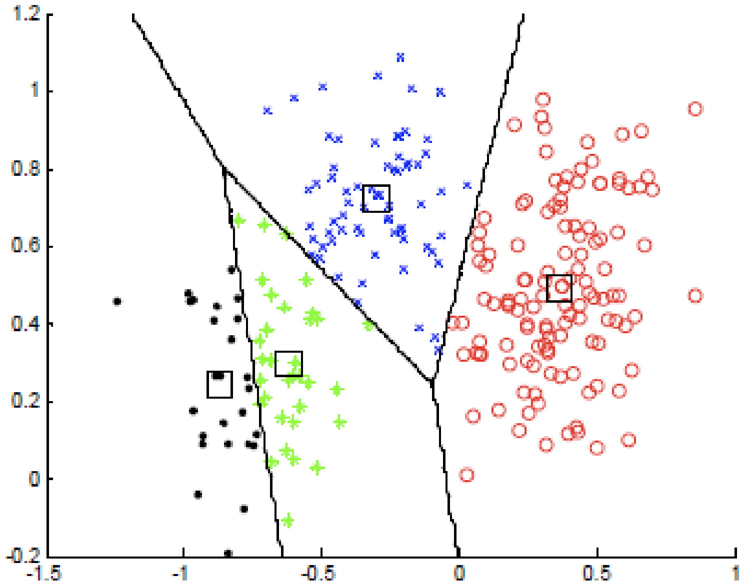


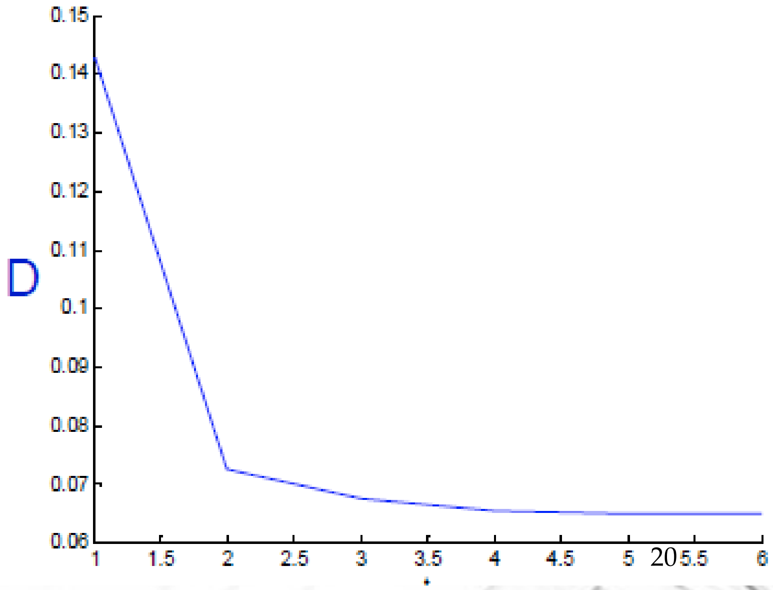




Sensitive to initialization







Practicalities

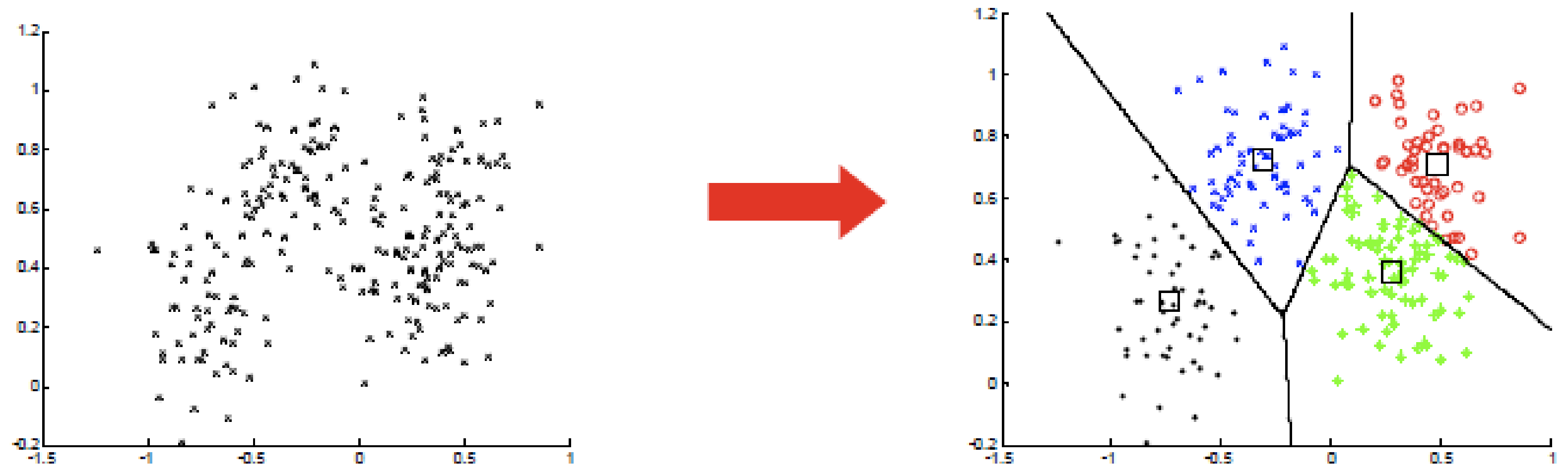
• always run algorithm several times with different initializations and keep the run with lowest cost

• choice of K

• suppose we have data for which a distance is defined, but it is non-vectorial (so can't be added). Which step needs to change?

• many other clustering methods: hierarchical K-means, K-medoids, agglomerative clustering …

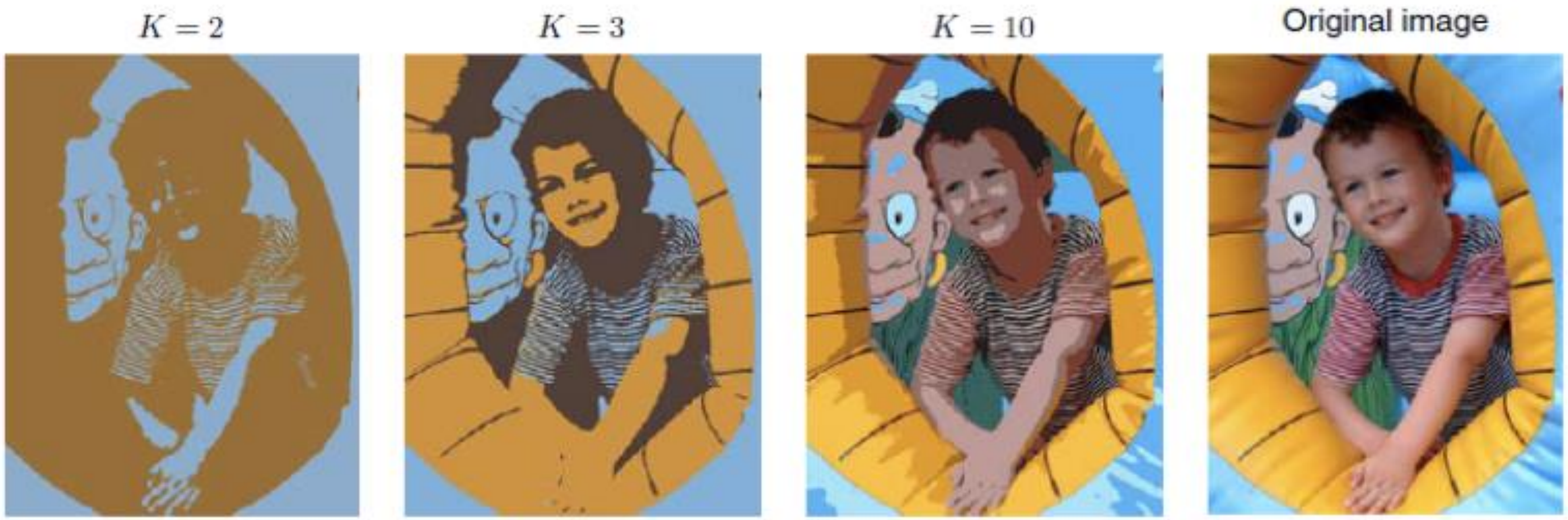
Example application 1: vector quantization



- all vectors in a cluster are considered equivalent
- they can be represented by a single vector – the cluster centre
- applications in compression, segmentation, noise reduction

Example: image segmentation

- K-means cluster all pixels using their colour vectors (3D)
- assign pixels to their clusters
- colour pixels by their cluster assignment



Example application 2: face clustering

- Determine the principal cast of a feature film
- Approach: view this as a clustering problem on faces

Algorithm outline

1. Detect faces for every fifth frame in the movie
2. Describe the face by a vector of intensities
3. Cluster using a K-means algorithm

Example – "Ground Hog Day" 2000 frames

Subset of detected faces in temporal order

Clusters for K = 4

EXAMPLES ….

K-Means

- The algorithm can group your data into k number of categories.
- The principle is to minimize the sum of squares of distances between data and the corresponding cluster centroids.

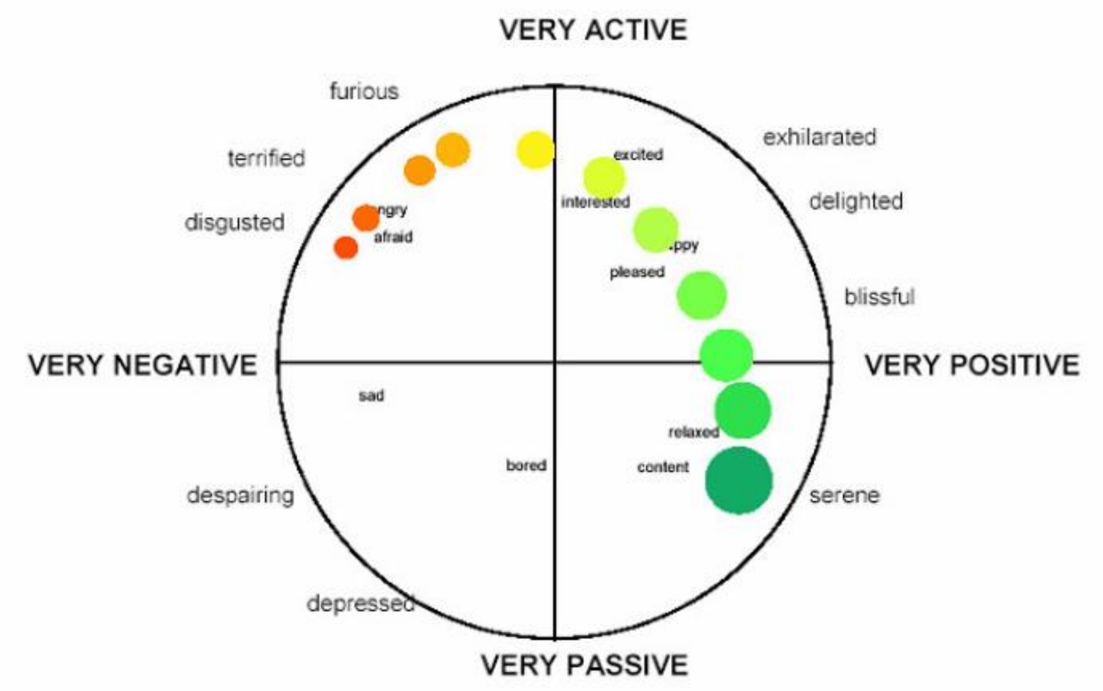
Example

arousal(-5 to 5)	valance
3	3
-1	-4
2	3
0	-5

We know that those data belong to two clusters. The question is how to determine which data points belong to cluster 1 and which belong to the other one.

EXAMPLES ….

Example



K-Mean Algorithm

Repeat the following three steps until convergence (stable):

Step 1: determine the centroid coordinates

Step 2: determine the distances of each data point to the centroids.

Step 3: group the data points based on minimum distance.

Example

Initialize the first two centroids

		c1=(3,3)	c2=(2,3)
3	3		
-1	-4		
2	3		
0	-5		

Measuring distances

- Calculate the distance between two data items.
- Euclidean distance

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$$

Iteration 1: calculate distances

		c1=(3,3)	c2=(2,3)
3	3	0	1
-1	-4	~~8.1~~	~~7.6~~
2	3	1	0
0	-5	~~8.5~~	~~8.2~~

Iteration 1: assign clusters

		c1=(3,3)	c2=(2,3)
3	3	0	1
-1	-4	8.1	7.6
2	3	1	0
0	-5	8.5	8.2

Iteration 1: compute new centroids

		c1=(3,3)	c2=(0.3,-2)
3	3	0	1
-1	-4	8.1	7.6
2	3	1	0
0	-5	8.5	8.2

Iteration 2: calculate distances

		c1=(3,3)	c2=(0.3,-2)
3	3	0	5.7
-1	-4	8.1	2.4
2	3	1	5.3
0	-5	8.5	3.0

1

Iteration 2: assign clusters

		c1=(3,3)	c2=(0.3,-2)
3	3	0	5.7
-1	-4	8.1	2.4
2	3	1	5.3
0	-5	8.5	3.0

2

Iteration 2: compute new centroids

		c1=(2.5,3)	c2=(-0.5,-4.5)
3	3	0	5.7
-1	-4	8.1	2.4
2	3	1	5.3
0	-5	8.5	3.0

3

Iteration 3

		c1=(2.5,3)	c2=(-0.5,-4.5)
3	3	0.5	8.2
-1	-4	7.8	0.7
2	3	0.5	7.9
0	-5	8.3	0.7

The new centroids will be the same!

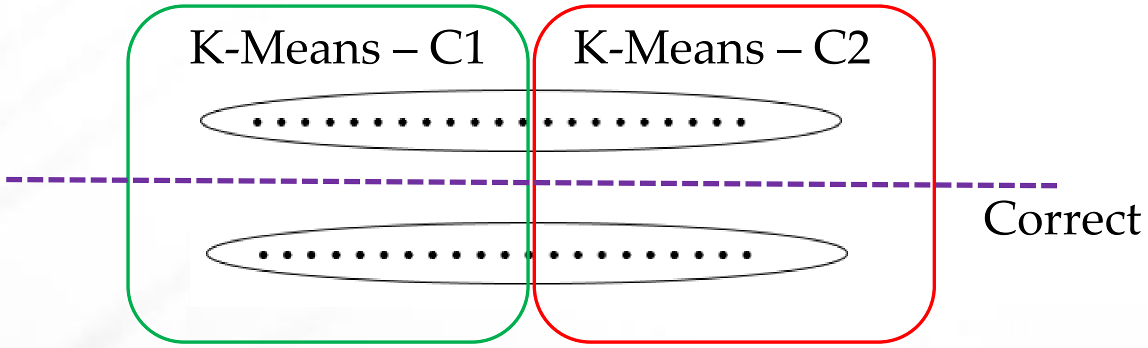
4

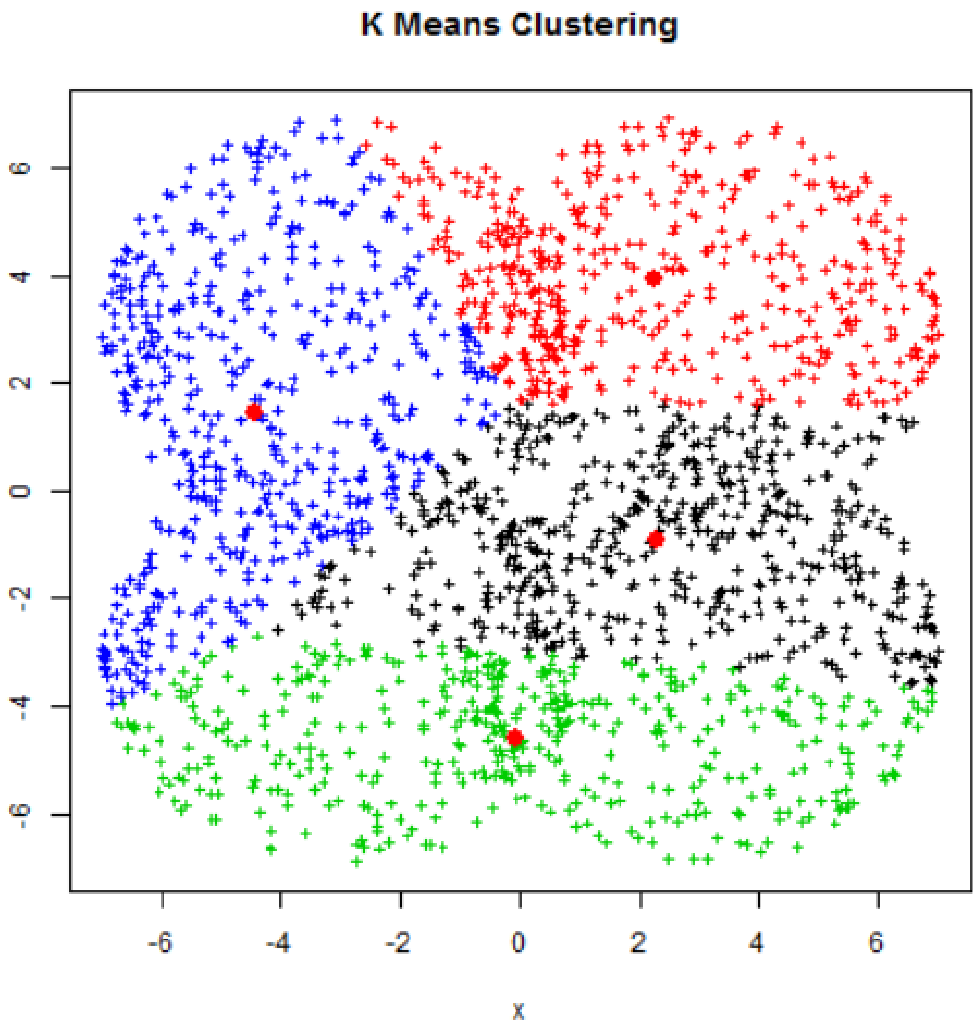
Matlab Example

```
function c = kmean(k,data)

[ndat ndim] = size(data);

% initilize the centra point
r = randperm(ndat);
c(1:k,:) = data(r(1:k),:);

ctemp = zeros(size(c));
cluster = zeros(size(data,1));
it = 0;
while (c ~= ctemp)
   it = it + 1;
   fprintf(1,'iteration %d: (%6.3f,%6.3f),(%6.3f,%6.3f),(%6.3f,%6.3f)\n', it,
    c(1,1),c(1,2),c(2,1),c(2,2),c(3,1),c(3,2));
   plot(data(:,1),data(:,2),'.r',c(:,1),c(:,2),'*b')
   pause;

   ctemp = c;
   dist = distance(data',c');
   [non cluster] = min(dist,[],2);
   for i = 1 : k
      c(i,:) = mean(data(find(cluster == i),:));
   end;
end;
```

When K-Means does work …

- Clusters are spherical
- Clusters are well separated
- Clusters are of similar volume
- Clusters have similar number





Questions?