Università di Verona

A.Y. 2021-22

Machine Learning & Artificial Intelligence

Unsupervised Learning & Clustering
K-means, Mixture of Gaussians, Mean-shift, Expectation-Maximization

Vittorio Murino

Introduction

- Previously, all our training samples were labeled, and the training procedure is called *supervised*
- We now investigate a number of *unsupervised* procedures which use *unlabeled* samples
- Collecting and labeling a large set of sample patterns can be costly
- We can train with large amounts of (less expensive) unlabeled data, and only then use supervision to label the groupings found, this is appropriate for “data mining” applications where the contents of a large database are not known beforehand

Introduction

- This is also appropriate in many applications when the characteristics of the patterns can change slowly over time
- We can use unsupervised methods to identify features that will then be useful for categorization
- We gain some insight into the nature (or structure) of the data
- Learning is “agnostic” since categories are not pre-defined

Clustering

Clustering is the task of dividing a set of data points into a number of groups (clusters) in such a way that data points in the same group are more similar to each other than to those in other groups.

Clustering

Given:

- a set of data points or "objects" $x_i \in X, X \in R^D$
- a definition of inter-object distance measure $Dist(x_i, x_j)$
- Optional: the number K of clusters

Goal:

- **find a partition** $P(x)$ such that $P(x_i) = P(x_j)$ iff x_i and x_j are in the same cluster

Trivial solutions:

- $P(x) = 1 \quad \forall x$
- $P(x) = x \quad \forall x$

Distance measures

Euclidean distance (numerical attributes): $Dist(x, x') = \sqrt{\sum_d |x_d - x'_d|^2}$

- symmetric, spherical, all dimensions are equally treated
- sensitive to large differences in single attributes

Hamming distance (categorical attributes): $Dist(x, x') = \sum_d \mathbf{1}_{x_d \neq x'_d}$

- counts the number of attributes where $x \neq x'$

Properties of clustering

- **Richness**: for any assignment of data points to clusters, there exist some distance D such that P_D returns that clustering.
- **Scale-invariance**: scaling distances by a positive value does not change the clustering.
- **Consistency**: shrinking intra-cluster distances and expanding inter-cluster distances does not change the clustering.

Clustering approaches

- **Partitioning methods**: partition the objects into *k* clusters by minimizing a cost function (e.g. *k-means*)
- **Density-based methods**: locates regions of high-density that are separated one another by regions of low-density (e.g. *mean-shift*)
- **Hierarchical methods**: form a tree structure (dendrogram) of clusters where a new cluster is formed by merging (or splitting) previously defined clusters (e.g. Linkage)

Hierarchical Clustering

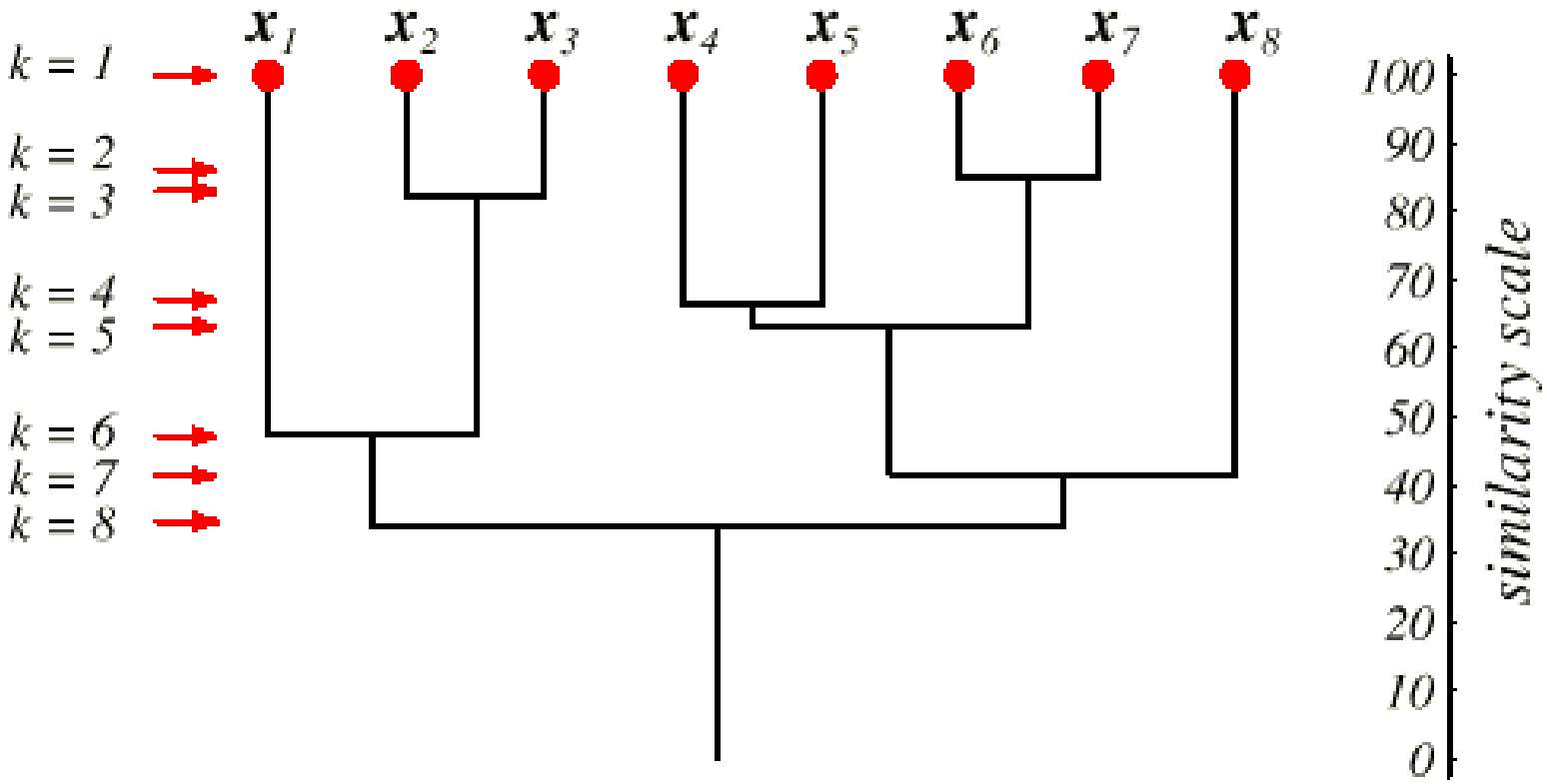
- **Agglomerative**: start with each point in a different cluster and iteratively merge them to bigger groups (*bottom-up*)

- **Divisive**: start with all data points belonging to one single cluster and split them iteratively in smaller groups (*top-down*)

Hierarchical Clustering

- Many times, clusters are not disjoint, but a cluster may have subclusters, in turn having sub-subclusters, etc.
- Consider a sequence of partitions of the *n* samples into *c* clusters
  - The first is a partition into *n* cluster, each one containing exactly one sample
  - The second is a partition into *n-1* clusters, the third into *n-2*, and so on, until the *n-th* in which there is only one cluster containing all of the samples
  - At the level *k* in the sequence, $c = n-k+1$.

- Given any two samples $\boldsymbol{x}$ and $\boldsymbol{x}$', they will be grouped together *at some level*, and if they are grouped at level k, they remain grouped for all higher levels
- Hierarchical clustering $\Rightarrow$ tree representation called *dendrogram*



- The similarity values may help to determine if the groupings are natural or forced, but if they are evenly distributed no information can be gained
- Another representation is based on set theory, e.g., on the Venn diagrams

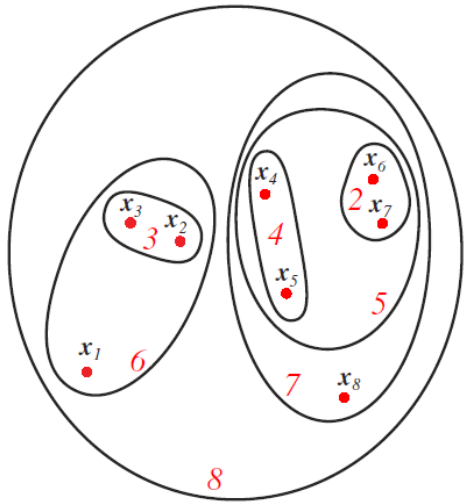


Figure 10.11: A set or Venn diagram representation of two-dimensional data (which was used in the dendrogram of Fig. 10.10) reveals the hierarchical structure but not the quantitative distances between clusters. The levels are numbered in red.

Agglomerative Hierarchical Clustering algorithm

Algorithm 4. (Agglomerative Hierarchical Clustering)

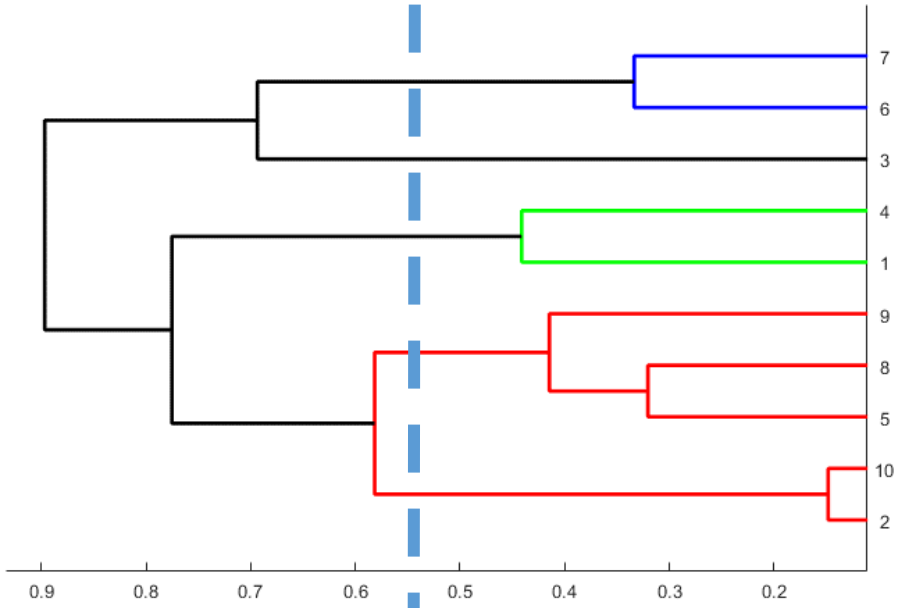
```
1 begin initialize c, ĉ ← n, D_i ← {x_i}, i = 1, ..., n
2         do ĉ ← ĉ − 1
3             find nearest clusters, say, D_i and D_j
4             merge D_i and D_j
5         until c = ĉ
6   return c clusters
7 end
```

- The procedure terminates when the specified number of cluster has been obtained, and returns the cluster as sets of points, rather than the mean or a representative vector for each cluster

Single Linkage Clustering (SLC)

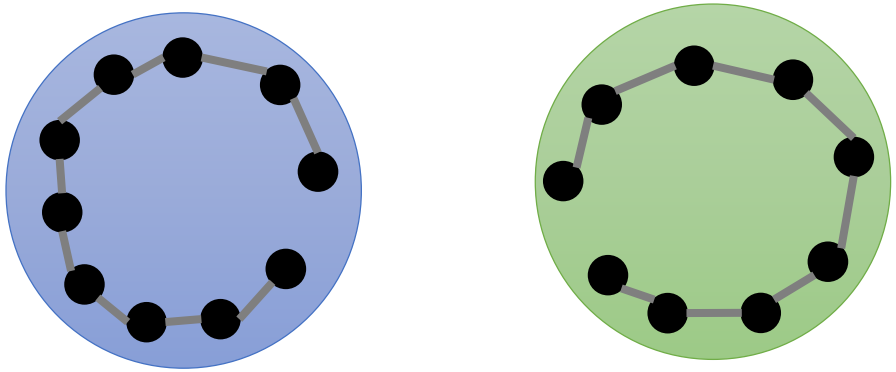
- Initialization:
  - Each data point is a cluster $P_D(x) = x \quad \forall x$
- Recursion:
  - Compute cluster-to-cluster distance $D(c_1, c_2) = \min_{x \in c_1, y \in c_2} Dist(x, y)$
  - Merge the cluster with the minimum distance $D(c_1, c_2)$
- Termination:
  - all the data points belong to the same cluster and decide at which level you want to "cut" the dendrogram, i.e., decide *K*

Agglomerative Hierarchical Clustering



K = 5

Single Linkage Clustering (SLC)

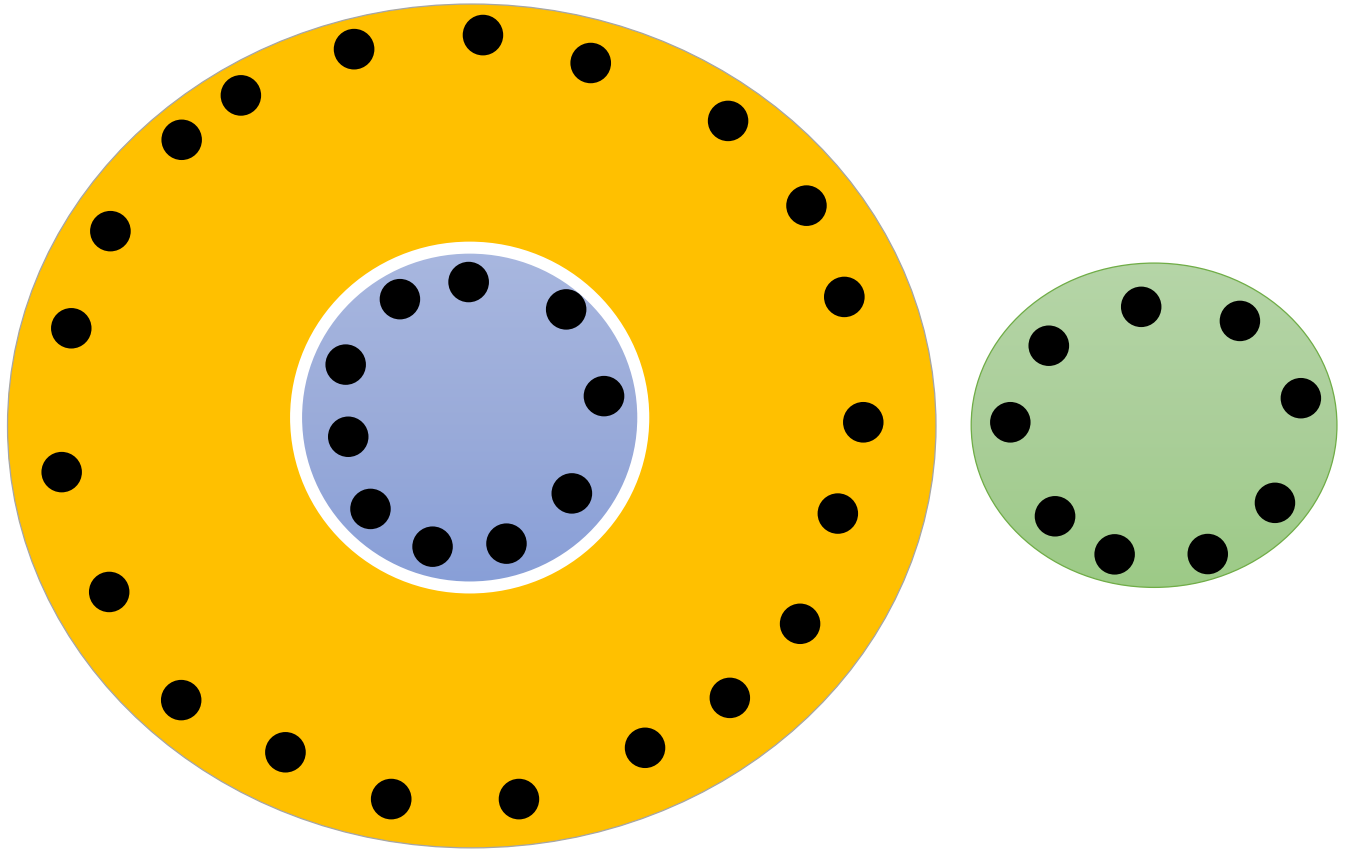


$$K = 2$$

$$Dist(c_1, c_2) = \min_{x \in c_1, y \in c_2} D(x, y)$$

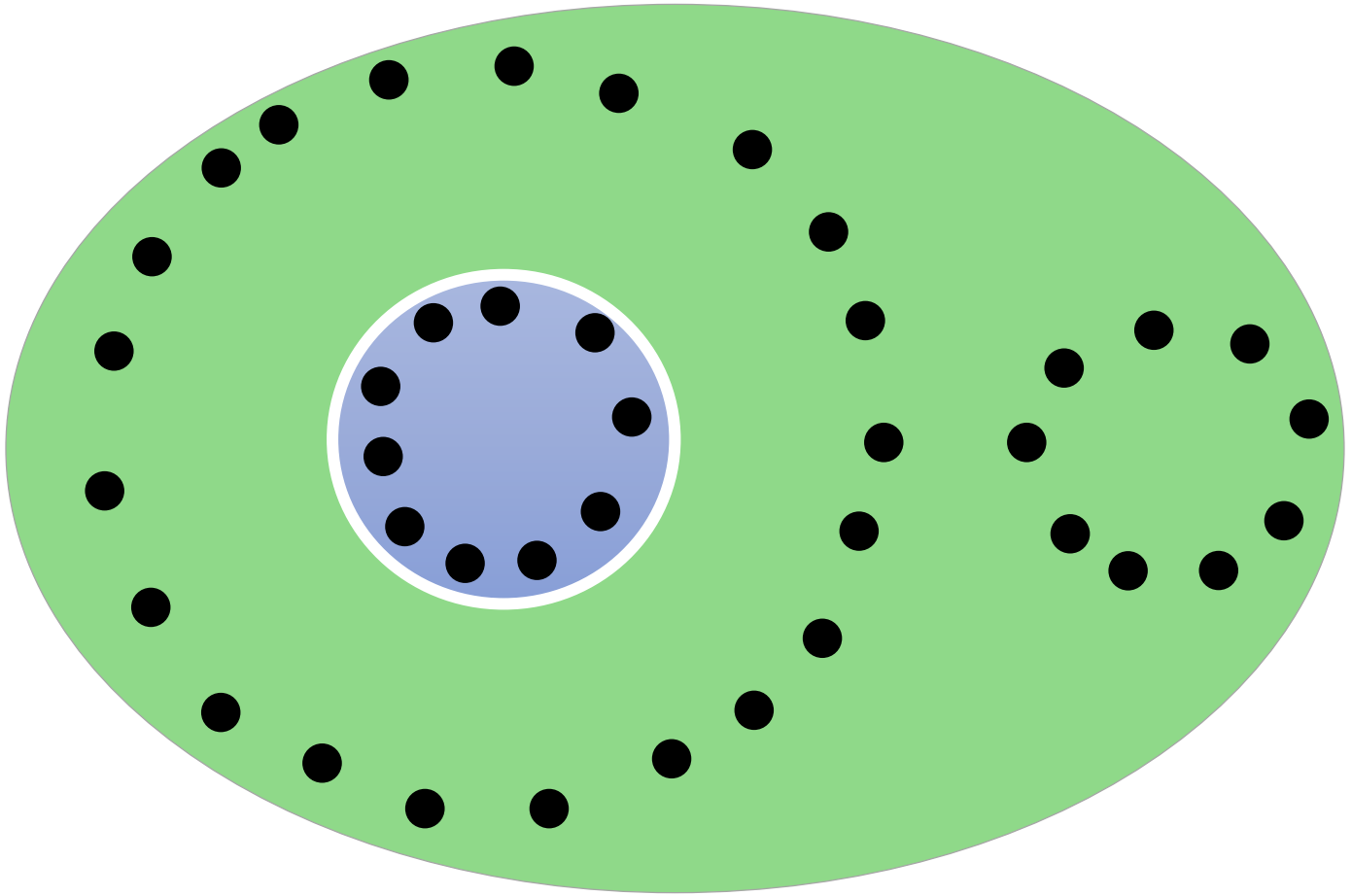
Issues with SLC

$K = 3$



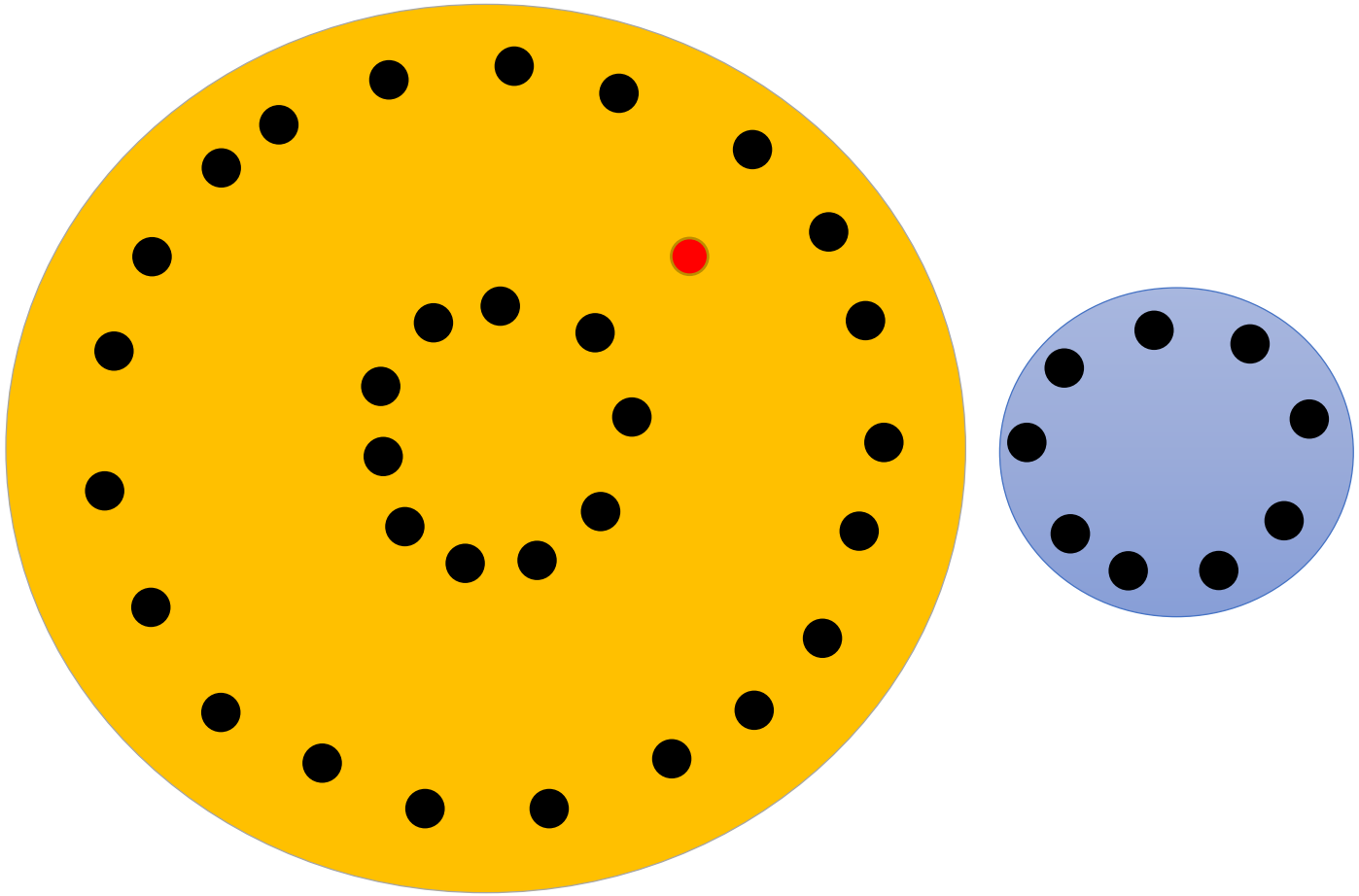
Issues with SLC

$$K = 2$$



Issues with SLC

$K = 2$



Summarizing SLC

PROS

- Deterministic
- Hierarchical
- Works for any K

CONS

- Sensitive to outliers
- Computationally expensive $O(n^3)$

K-Means clustering

- Data set: $\{\mathbf{x}_1, ...,\mathbf{x}_N\}$, N observations of a random D-dimensional variable $\mathbf{x}$
- Goal: partition data set into K clusters, K fixed a priori
- A cluster is a group of data points whose inter-point distances are small as compared with the distances to points outside of the cluster
- Introducing $\boldsymbol{\mu}_k$, D-dimensional vector, $k=1,..,K$, representing the center of the clusters

K-Means clustering

- Notation:
  - for each data point $\mathbf{x}_n$ there is a set of binary indicator variables $r_{nk} \in \{0,1\}$, $k=1,..,K$ describing which of the K clusters the data point $\mathbf{x}_n$ is assigned to (1-of-K coding scheme)
- Define an objective function (distortion measure)

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \left\| \mathbf{x}_n - \boldsymbol{\mu}_k \right\|^2$$

  sum of the squares of the distances of each data point to its assigned vectors $\boldsymbol{\mu}_k$
- The goal is now to find values of r_{nk} and $\boldsymbol{\mu}_k$ so as to minimize J

K-Means clustering

- Algorithm: iterative, 2 steps, alternate optimization wrt r_{nk} and $\boldsymbol{\mu}_k$
  - choose initial value for the $\boldsymbol{\mu}_k$, $k=1,..,K$
  - Step 1: minimize J wrt r_{nk} keeping $\boldsymbol{\mu}_k$ fixed
  - Step 2: minimize J wrt $\boldsymbol{\mu}_k$ keeping r_{nk} fixed
  - repeat until convergence or a limited number of iterations
- Determination of r_{nk}
  - J is a linear function of r_{nk}, so closed form solution
  - optimise each $\mathbf{x}_n$ separately, choose r_{nk} to be 1 for whichever value of k gives the minimum value of

$$\left\|\mathbf{x}_n - \boldsymbol{\mu}_k\right\|^2 \qquad r_{nk} = \begin{cases} 1 & \text{if } k = \operatorname{argmin}_j \left\|\mathbf{x}_n - \boldsymbol{\mu}_j\right\|^2 \\ 0 & \text{otherwise} \end{cases}$$

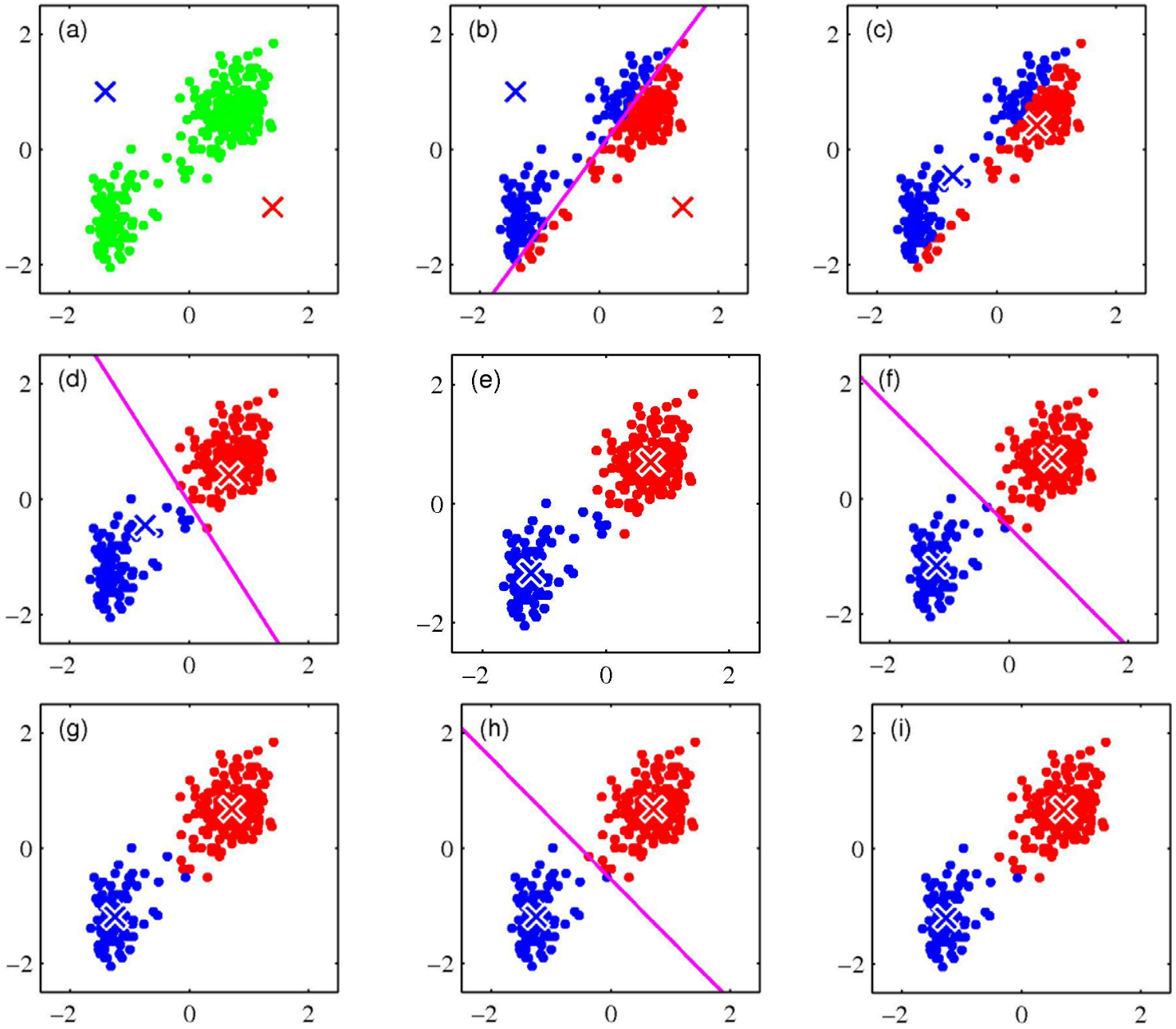
K-Means clustering

- Determination of $\boldsymbol{\mu}_k$
  - J is a quadratic function of $\boldsymbol{\mu}_k$, so solution is given by setting its derivative (wrt $\boldsymbol{\mu}_k$) to zero

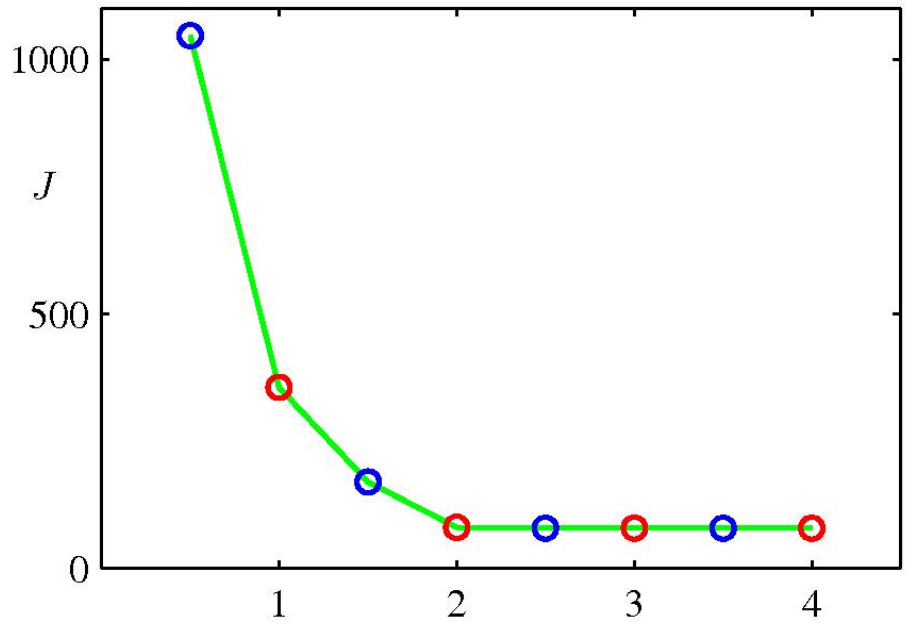
$$2\sum_{n=1}^{N} r_{nk}\left(\mathbf{x}_n - \boldsymbol{\mu}_k\right) = 0 \Rightarrow \boldsymbol{\mu}_k = \frac{\sum_n r_{nk}\mathbf{x}_n}{\sum_n r_{nk}}$$

    where the denominator is equal to the number of points assigned to cluster k
    → K-means

- Stop when there is no further change or after some maximum number of iterations
- May converge to a *local* (than global) minimum of J
- Issues: fast implementation, initialization of the cluster centers


(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)

- E-steps: r_{nk} updating
- M-steps: μ_k updating

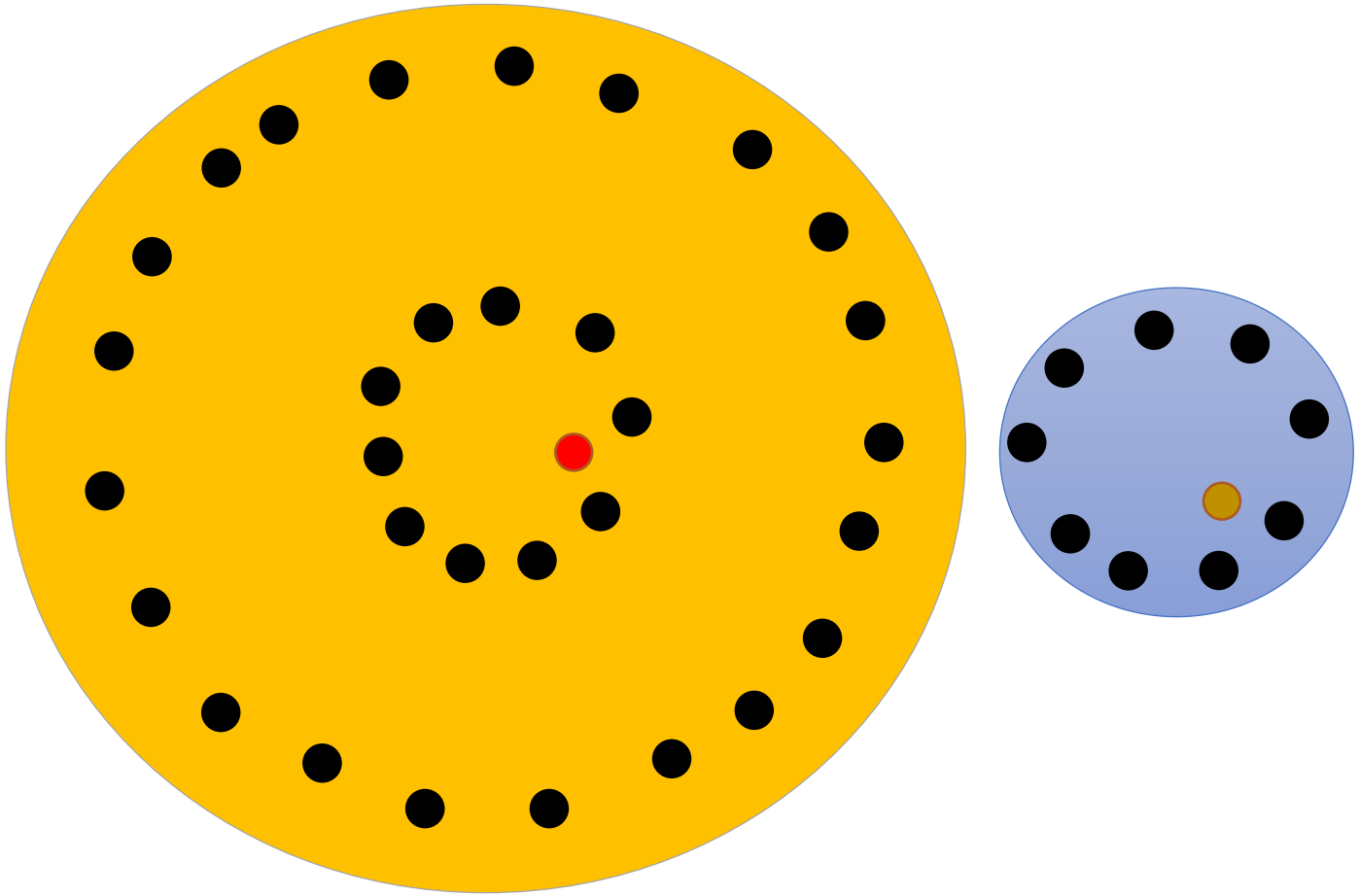


K-Means clustering alternative algorithm

- Initialization:
  - “randomly” select K centroids $\boldsymbol{\mu}_k$, $k=1,..,K$
- Recursion:
  - For each data point $x_i \in X$
    - find the nearest centroid $\boldsymbol{\mu}_j$
    - Assign point x_i to cluster C_j
  - For each cluster $j = 1, \ldots, k$
    - Compute the centroid $\boldsymbol{\mu}_j = \frac{1}{|C_j|} \sum_{x_k \in C_j} x_k$
- Termination:
  - Convergence (i.e. all clusters unchanged in one iteration)

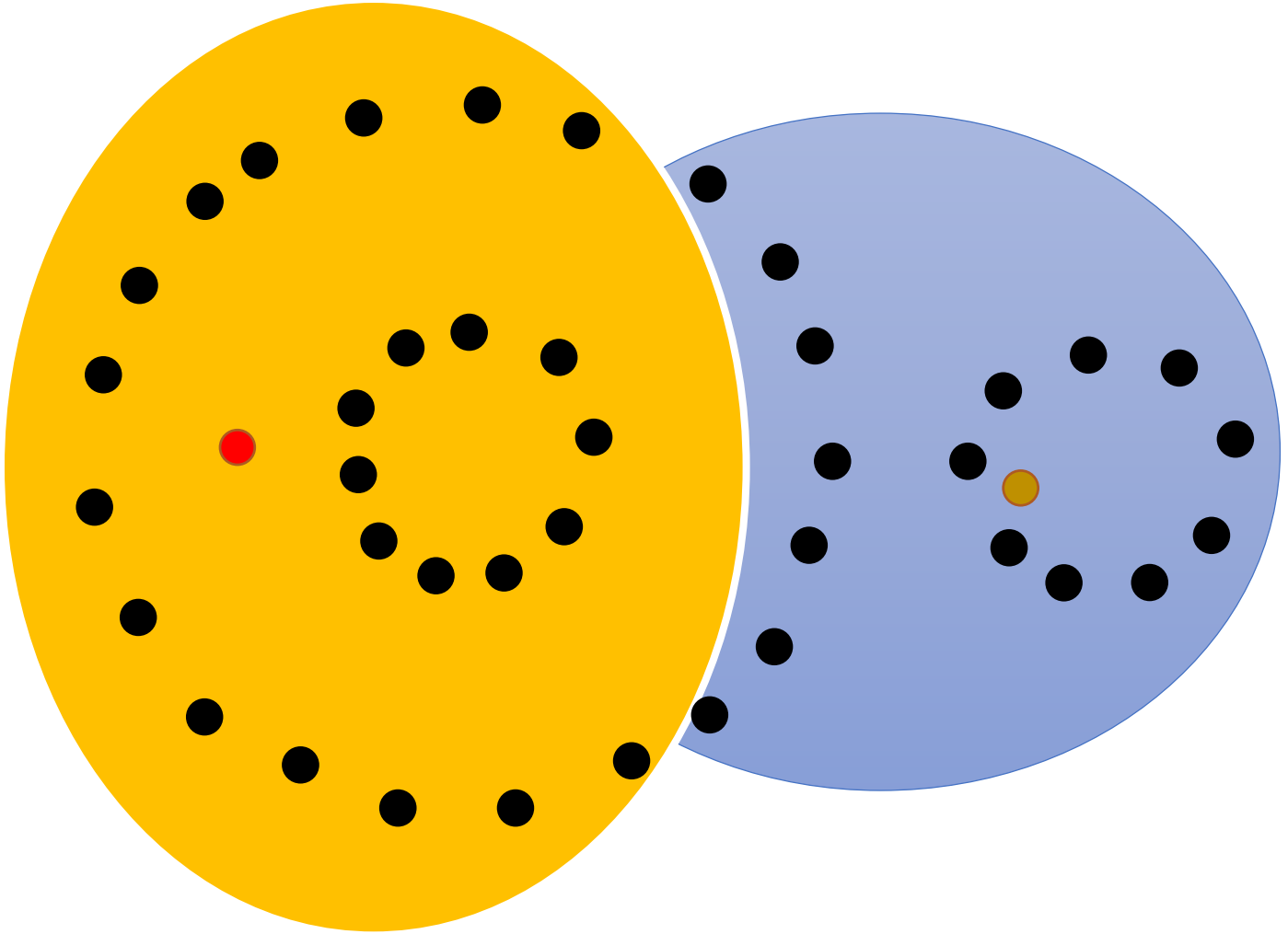
Issues with K-means

$K = 2$



Issues with K-means

$K = 2$



Summarizing K-means

PROS

- Simple
- Suitable for large datasets
- Efficient

CONS

- Non-deterministic
- Non-optimal
- Sensitive to scale, initialization and outliers
- Only finds compact, “spherical” clusters

Visualizing K-means

https://www.naftaliharris.com/blog/visualizing-k-means-clustering/

Mixtures of Gaussians

- Gaussian mixtures of D-dimensional variables introducing *latent* variables

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{\mu}_k, \Sigma_k)$$

- The latent variable $\mathbf{z}$ is a K-dimensional binary random variable having $1\text{-of-}\mathrm{K}$ representation: [0 ... 0 1 0 ... 0]

$$z_k \in \{0, 1\}, \Sigma_k z_k = 1$$

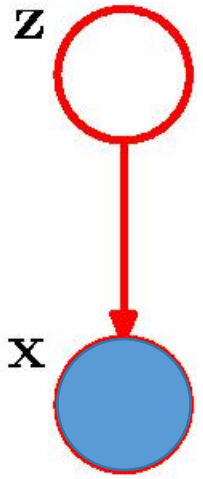
- Let's define the joint distribution $p(\mathbf{x},\mathbf{z})$ in terms of the marginal distribution $p(\mathbf{z})$ and the conditional distribution $p(\mathbf{x}|\mathbf{z})$.
  The marginal is:

$$p(z_k = 1) = \pi_k, \quad 0 \le \pi_k \le 1, \quad \Sigma_k \pi_k = 1 \quad \rightarrow \quad p(\mathbf{z}) = \prod_{k=1}^{K} \pi_k^{z_k}$$

- The conditional distribution

$$p(\mathbf{x}|z_k = 1) = \mathcal{N}(\boldsymbol{\mu}_k, \Sigma_k)$$

$$p(\mathbf{x}|\mathbf{z}) = \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{\mu}_k, \Sigma_k)^{z_k}$$



- The joint distribution

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}) = \sum_{\mathbf{z}} p(\mathbf{z}) p(\mathbf{x}|\mathbf{z}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{\mu}_k, \Sigma_k)$$

- If we have N observations $\mathbf{x}_1, \ldots \mathbf{x}_\mathrm{N}$, for each datapoint $\mathbf{x}_\mathrm{n}$ there is a latent variable $\mathbf{z}_\mathrm{n}$
- Hence, we have found an equivalent formulation of the mixture of Gaussians involving an explicit latent variable

- If we consider the conditional probability of $\mathbf{z}$ given $\mathbf{x}$

$$\gamma(z_k) \equiv p(z_k = 1 | \mathbf{x}) =$$

$$= \frac{p(z_k = 1)\, p(\mathbf{x} | z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)\, p(\mathbf{x} | z_j = 1)} = \frac{\pi_k \mathcal{N}(\boldsymbol{\mu}_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{\mu}_j, \Sigma_j)}$$

- π_k can be viewed as the prior probability of $z_k = 1$
- $\gamma(z_k)$ is the correspondent posterior probability as we have observed $\mathbf{x}$
  - It can be interpreted as the *responsibility* that component k takes for 'explaining' the observation $\mathbf{x}$

- To generate random samples from such a mixture model, we can generate a value for $\mathbf{z}$ – $\mathbf{z}^*$ – from the marginal $p(\mathbf{z})$, and then, generate a value for $\mathbf{x}$ from $p(\mathbf{x}|\mathbf{z})$ → plot (a) in the figure
- Samples from marginal $p(\mathbf{x})$ can be taken in the same way ignoring the values of $\mathbf{z}$ → plot (b) in the figure

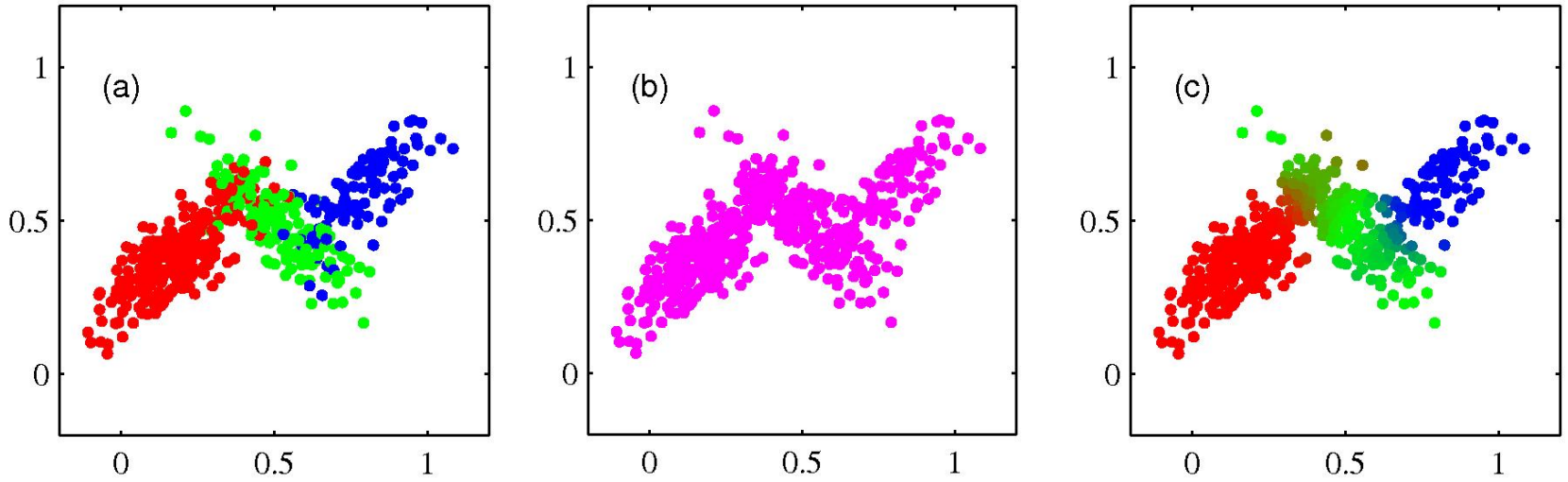
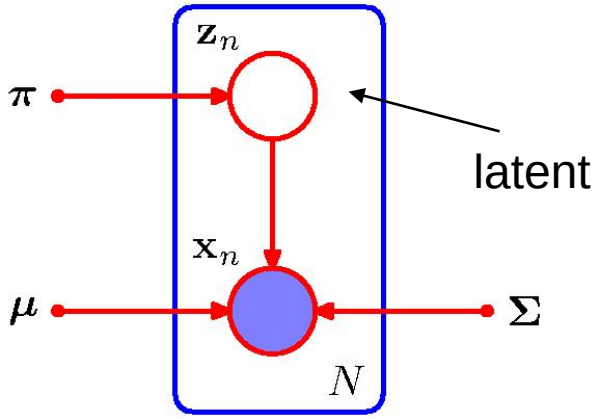


Figure 9.5 Example of 500 points drawn from the mixture of 3 Gaussians shown in Figure 2.23. (a) Samples from the joint distribution $p(\mathbf{z})p(\mathbf{x}|\mathbf{z})$ in which the three states of $\mathbf{z}$, corresponding to the three components of the mixture, are depicted in red, green, and blue, and (b) the corresponding samples from the marginal distribution $p(\mathbf{x})$, which is obtained by simply ignoring the values of $\mathbf{z}$ and just plotting the $\mathbf{x}$ values. The data set in (a) is said to be *complete*, whereas that in (b) is *incomplete*. (c) The same samples in which the colours represent the value of the responsibilities $\gamma(z_{nk})$ associated with data point $\mathbf{x}_n$, obtained by plotting the corresponding point using proportions of red, blue, and green ink given by $\gamma(z_{nk})$ for $k = 1, 2, 3$, respectively

$\gamma(\mathbf{z}_k)$ → R,G,B

Maximum likelihood



- Consider:
  - N observations $\{\mathbf{x}_1, \ldots \mathbf{x}_N\}$
  - Modeling a mixture → NxD matrix $\mathbf{X}$ with rows $\mathbf{x}_n^T$
  - Latent variables denoted by NxK matrix $\mathbf{Z}$ with rows $\mathbf{z}_n^T$

- The Gaussian mixture model of this i.i.d. data is

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \Sigma_k) \right\}$$

- ML for mixtures is more complex than for a single Gaussian (sum over k inside the ln)

  → no closed form solution

Expectation-Maximization for Gaussian mixtures

- Maximum of the log-likelihood function: derivative wrt $\boldsymbol{\mu}_k$ set to zero

$$0 = +\sum_{n=1}^{N} \frac{\pi_k \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \Sigma_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_j, \Sigma_j)} \Sigma_k^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_k)$$

from which

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n \quad \text{where} \quad N_k = \sum_{n=1}^{N} \gamma(z_{nk})$$

- N_k can be interpreted as the effective number of points assigned to cluster k.

- Accordingly if we derive wrt Σ_k

$$\Sigma_k = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})(\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T$$

  assuming the same interpretation as for $\boldsymbol{\mu}_k$

- Finally, we maximize $ln\ p(\mathbf{X}|\ \boldsymbol{\pi}_k, \boldsymbol{\mu}_k, \Sigma_k)$ wrt $\boldsymbol{\pi}_k$; since $\boldsymbol{\pi}_k$'s should sum to 1, we have to use Lagrange multipliers

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \lambda\left(\sum_{k=1}^{K}\pi_k - 1\right)$$

$$0 = \sum_{n=1}^{N}\frac{\mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \Sigma_k)}{\sum_j \pi_j N(\mathbf{x}_n \mid \boldsymbol{\mu}_j, \Sigma_j)} + \lambda \quad \Rightarrow \pi_k = \frac{N_k}{N}$$

- Please, note that
  - this is not a closed form solution because parameters depend on $\gamma(z_k)$ in a complex way
  - suggest an iterative procedure for the solution:
    - choose initial values for means, covariances and mixing coefficients, and alternate between E ad M steps
    - *E step*
      - use current values of parameters to evaluate the responsibilities γ
    - *M step*
      - use responsibilities to re-estimate means, covariances and mixing coefficients
    - similarity with *K-means*: EM is slower, K-Means can be used to initialize EM

EM for Gaussian Mixtures

Given a Gaussian mixture model, the goal is to maximize the likelihood function with respect to the parameters (comprising the means and covariances of the components and the mixing coefficients).

1. Initialize the means $\boldsymbol{\mu}_k$, covariances $\boldsymbol{\Sigma}_k$ and mixing coefficients π_k, and evaluate the initial value of the log likelihood.
2. **E step**. Evaluate the responsibilities using the current parameter values

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}. \tag{9.23}$$

3. **M step**. Re-estimate the parameters using the current responsibilities

$$\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})\mathbf{x}_n \tag{9.24}$$

$$\boldsymbol{\Sigma}_k^{\text{new}} = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})\left(\mathbf{x}_n-\boldsymbol{\mu}_k^{\text{new}}\right)\left(\mathbf{x}_n-\boldsymbol{\mu}_k^{\text{new}}\right)^{\mathrm{T}} \tag{9.25}$$

$$\pi_k^{\text{new}} = \frac{N_k}{N} \tag{9.26}$$

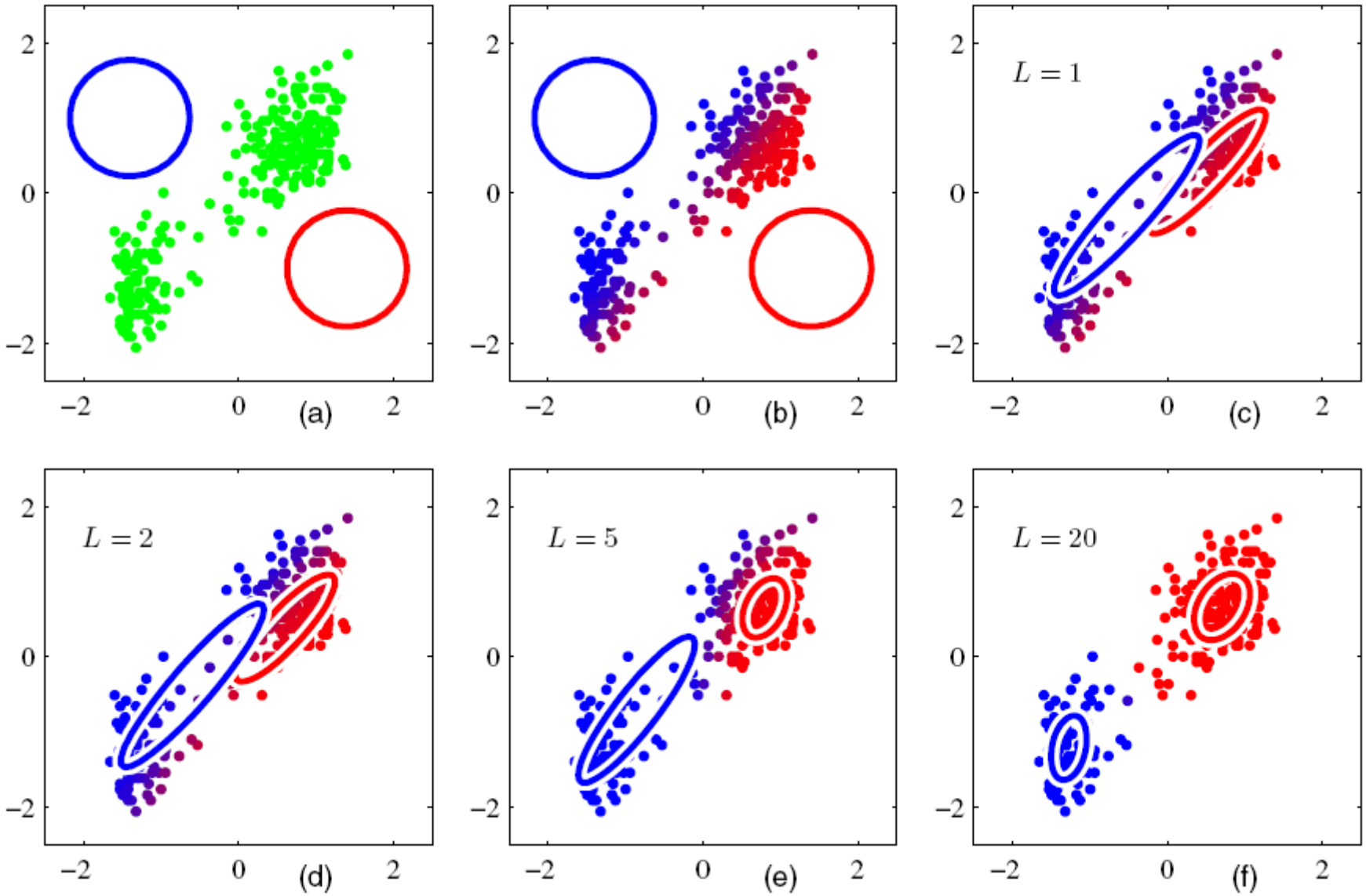
where

$$N_k = \sum_{n=1}^{N}\gamma(z_{nk}). \tag{9.27}$$

4. Evaluate the log likelihood

$$\ln p(\mathbf{X}|\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\pi}) = \sum_{n=1}^{N}\ln\left\{\sum_{k=1}^{K}\pi_k\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)\right\} \tag{9.28}$$

and check for convergence of either the parameters or the log likelihood. If the convergence criterion is not satisfied return to step 2.


2
0
−2
−2
0
2
(a)
(b)
$L = 1$
(c)
$L = 2$
(d)
$L = 5$
(e)
$L = 20$
(f)

The general EM algorithm

- Goal of EM: find maximum likelihood solution for models having latent variables
- Set of *observed* data $\mathbf{X}$, set of latent variables $\mathbf{Z}$, set of parameters $\boldsymbol{\theta}$

$$L(\boldsymbol{\theta}) = \ln p(\mathbf{X}|\boldsymbol{\theta}) = \ln\left\{\sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})\right\}$$

- Please, note that summation over latent variables appears inside the logarithm, even if joint probability distribution belongs to exponential family, the marginal does not because of summation

- Complete data set $\{\mathbf{X},\mathbf{Z}\}$
- Incomplete data set $\mathbf{X}$

- We have only the incomplete set $\mathbf{X}$, our knowledge about the values of latent variables in $\mathbf{Z}$ are given only by the posterior $p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta})$

- We cannot use the complete data log-likelihood and consider its expected value under the posterior (E step), and then maximize this expectation (M step)

- In the E step, we use current parameters $\boldsymbol{\theta}^{\text{old}}$ to find the posterior of the latent variable given by $p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{\text{old}})$.
- The, we use this posterior to find the expectation of the complete data log-likelihood Q, wrt this posterior, evaluated at an arbitrary $\boldsymbol{\theta}$, that is:

$$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}}) \ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$$

- In the M step, we determine the revised parameter estimate $\boldsymbol{\theta}^{\text{new}}$ by maximizing

$$\boldsymbol{\theta}^{\text{new}} = \arg\max_{\boldsymbol{\theta}} \mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}})$$

- Each cycle of EM will increase the incomplete data log-likelihood

The General EM Algorithm

Given a joint distribution $p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$ over observed variables $\mathbf{X}$ and latent variables $\mathbf{Z}$, governed by parameters $\boldsymbol{\theta}$, the goal is to maximize the likelihood function $p(\mathbf{X}|\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$.

1. Choose an initial setting for the parameters $\boldsymbol{\theta}^{\text{old}}$.

2. **E step** Evaluate $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}})$.

3. **M step** Evaluate $\boldsymbol{\theta}^{\text{new}}$ given by

$$\boldsymbol{\theta}^{\text{new}} = \arg\max_{\boldsymbol{\theta}} \mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) \tag{9.32}$$

   where

$$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}}) \ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}). \tag{9.33}$$

4. Check for convergence of either the log likelihood or the parameter values. If the convergence criterion is not satisfied, then let

$$\boldsymbol{\theta}^{\text{old}} \leftarrow \boldsymbol{\theta}^{\text{new}} \tag{9.34}$$

   and return to step 2.

Expectation Maximization

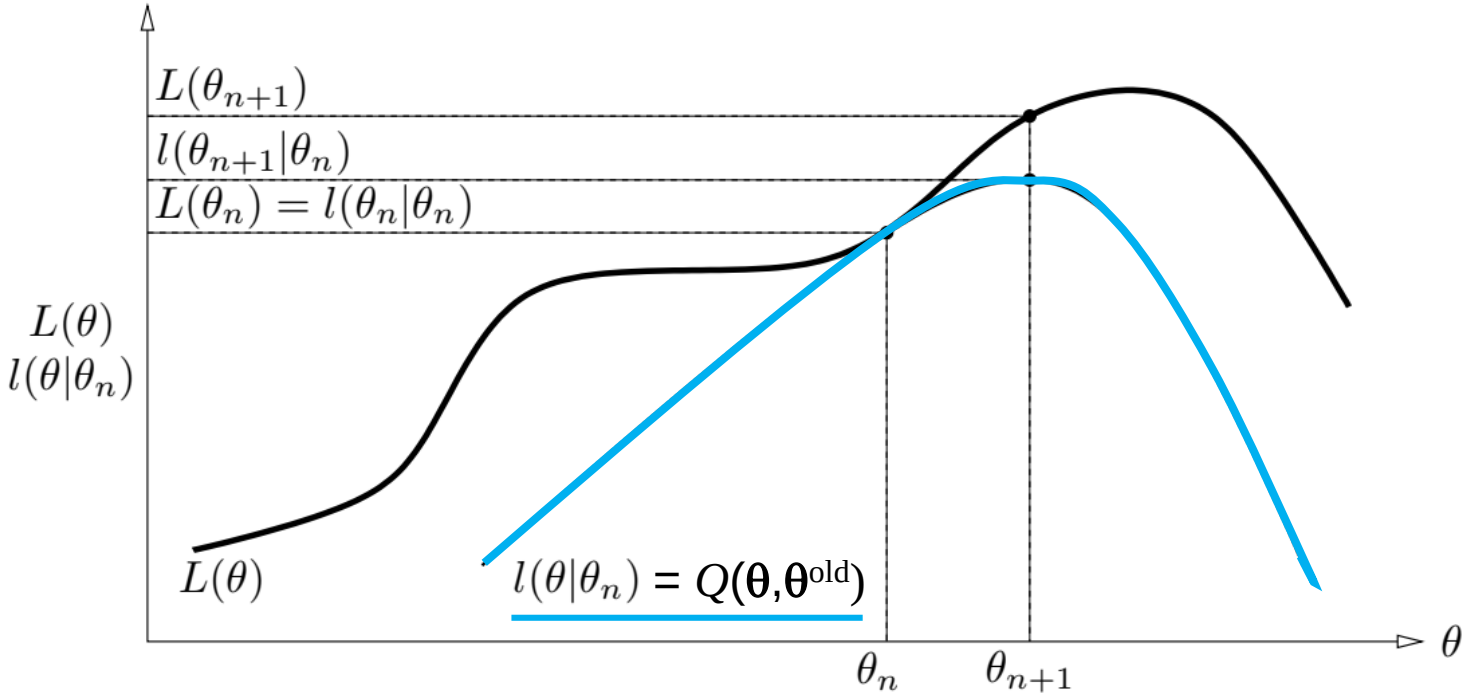
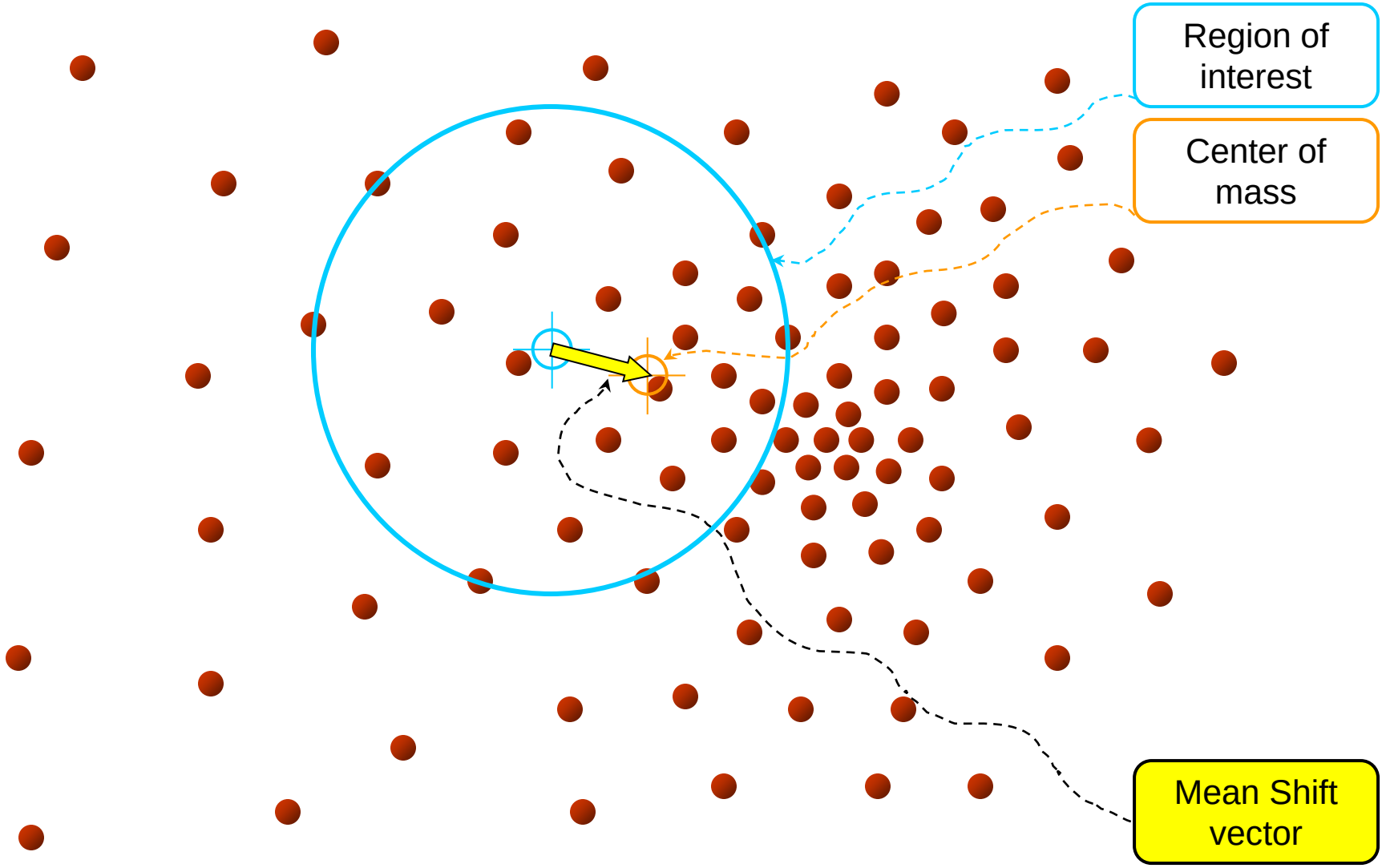


Figure 2: Graphical interpretation of a single iteration of the EM algorithm: The function $L(\theta|\theta_n)$ is upper-bounded by the likelihood function $L(\theta)$. The functions are equal at $\theta = \theta_n$. The EM algorithm chooses θ_{n+1} as the value of θ for which $l(\theta|\theta_n)$ is a maximum. Since $L(\theta) \geq l(\theta|\theta_n)$ increasing $l(\theta|\theta_n)$ ensures that the value of the likelihood function $L(\theta)$ is increased at each step.

Density-based clustering: Mean-shift

- Assume all the points are sampled from a finite number of distributions.
- We look for the **modes** (i.e. local density maxima) within windows of given dimensions.

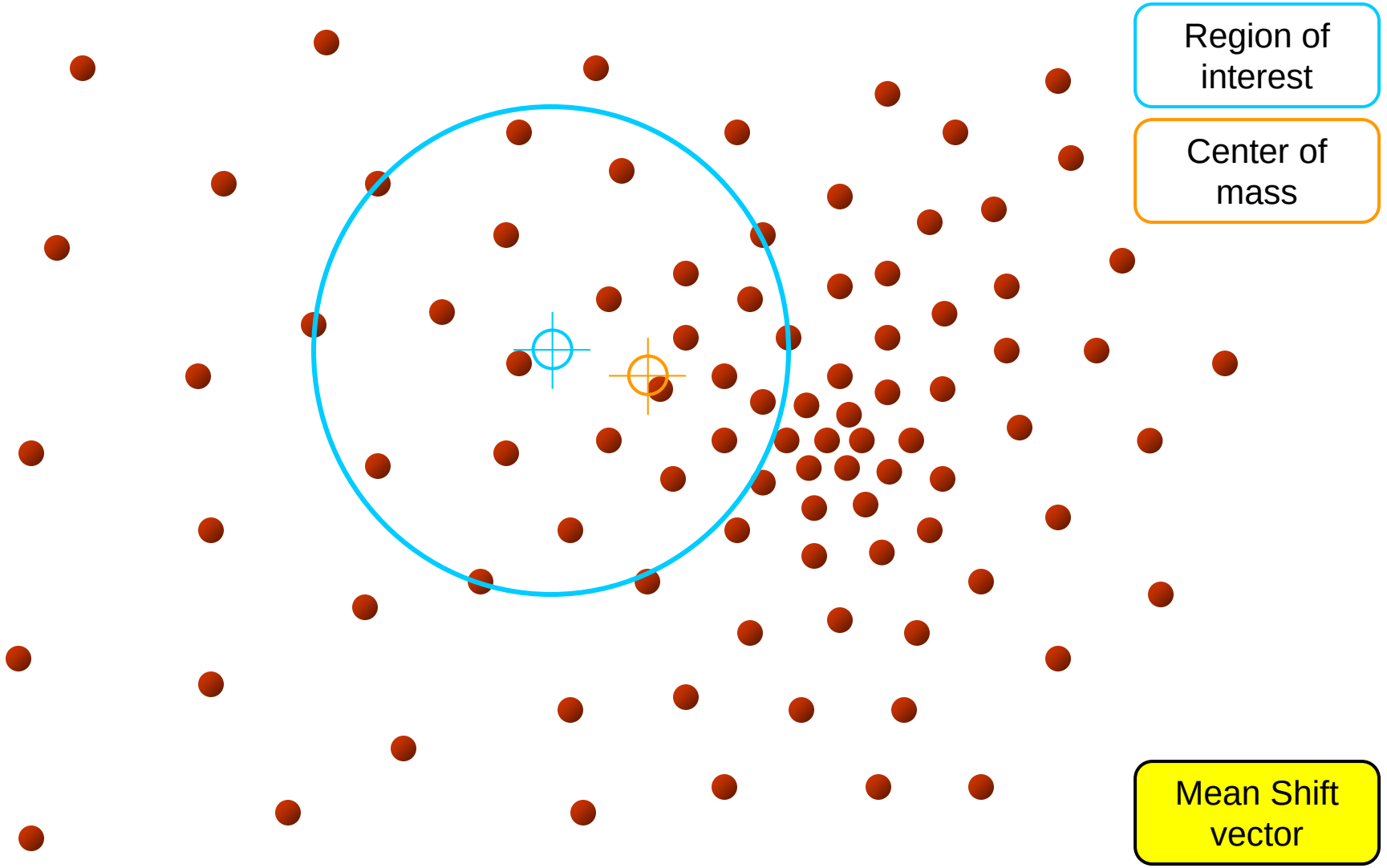
Intuitive Description



Objective : Find the densest region

Distribution of identical billiard balls

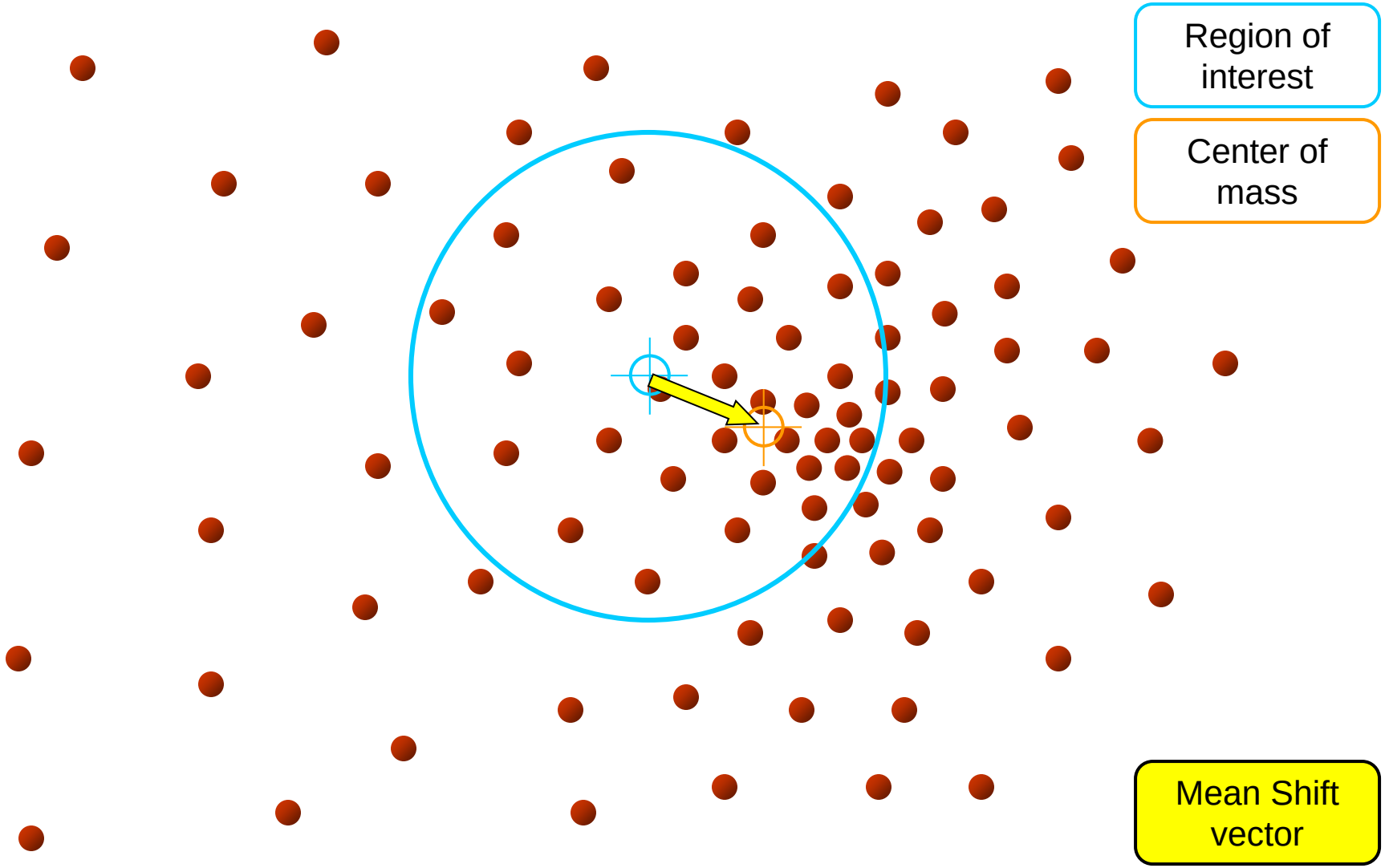
Intuitive Description



Objective : Find the densest region

Distribution of identical billiard balls

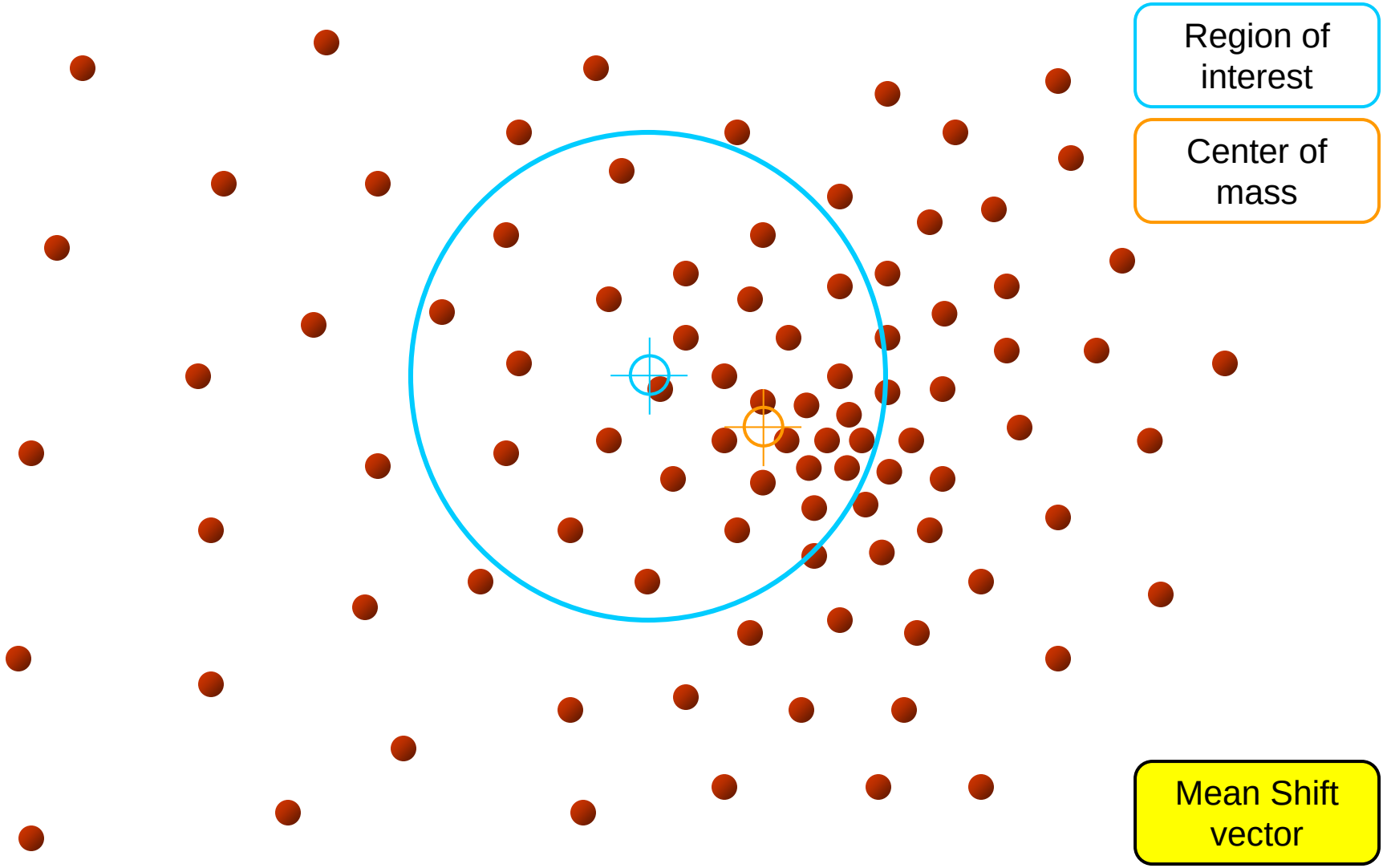
Intuitive Description



Objective : Find the densest region

Distribution of identical billiard balls

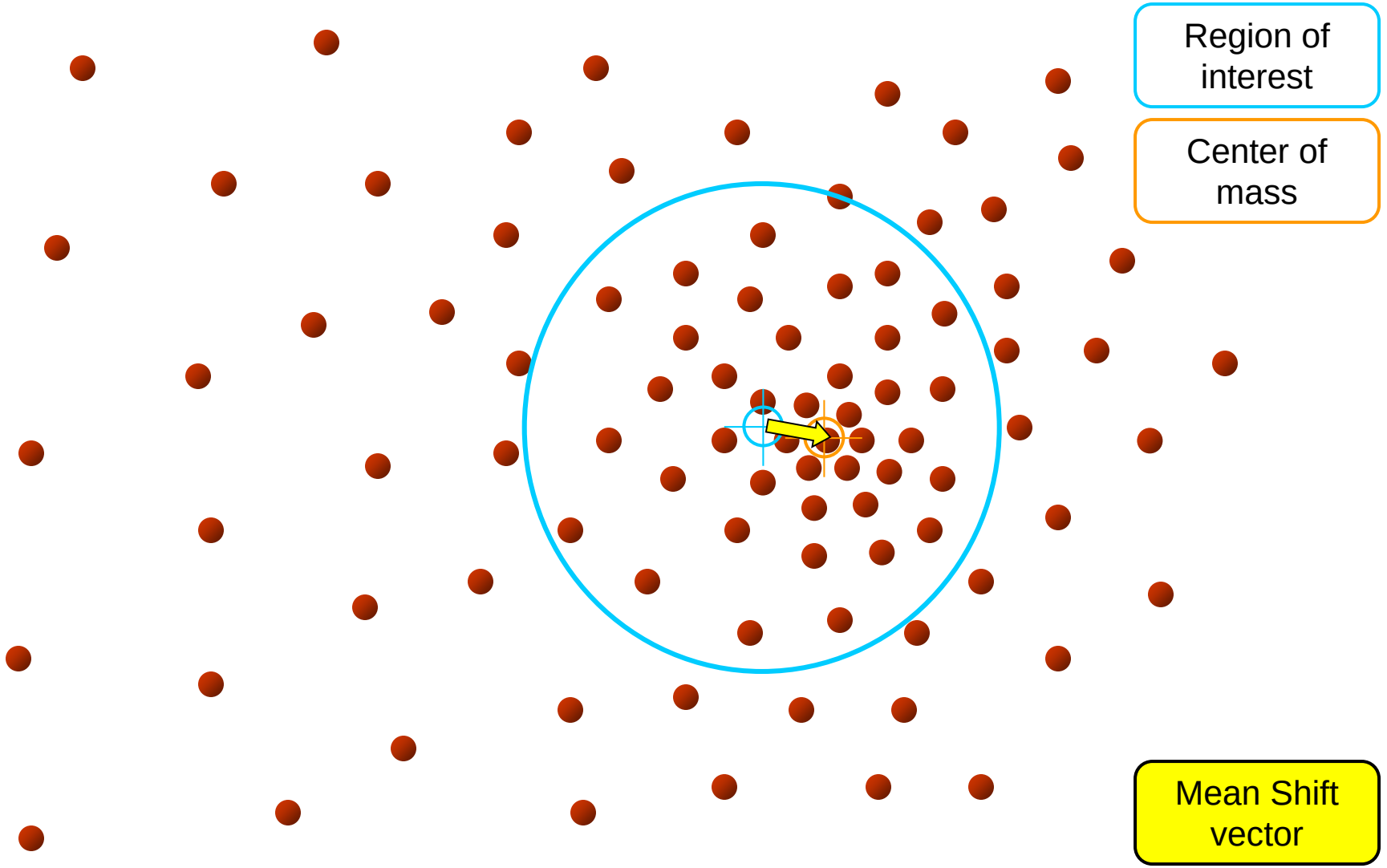
Intuitive Description



Objective : Find the densest region

Distribution of identical billiard balls

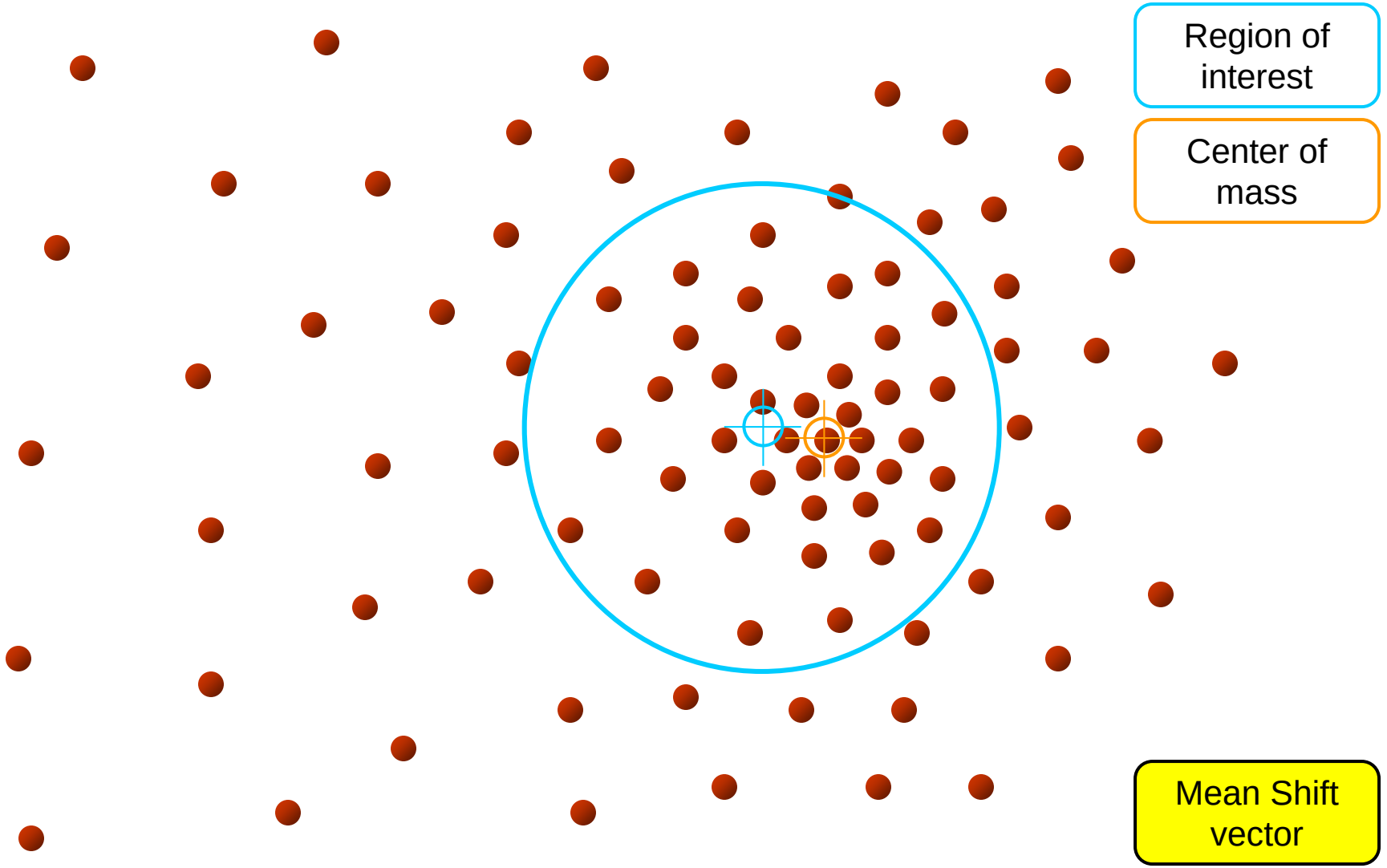
Intuitive Description



Objective : Find the densest region

Distribution of identical billiard balls

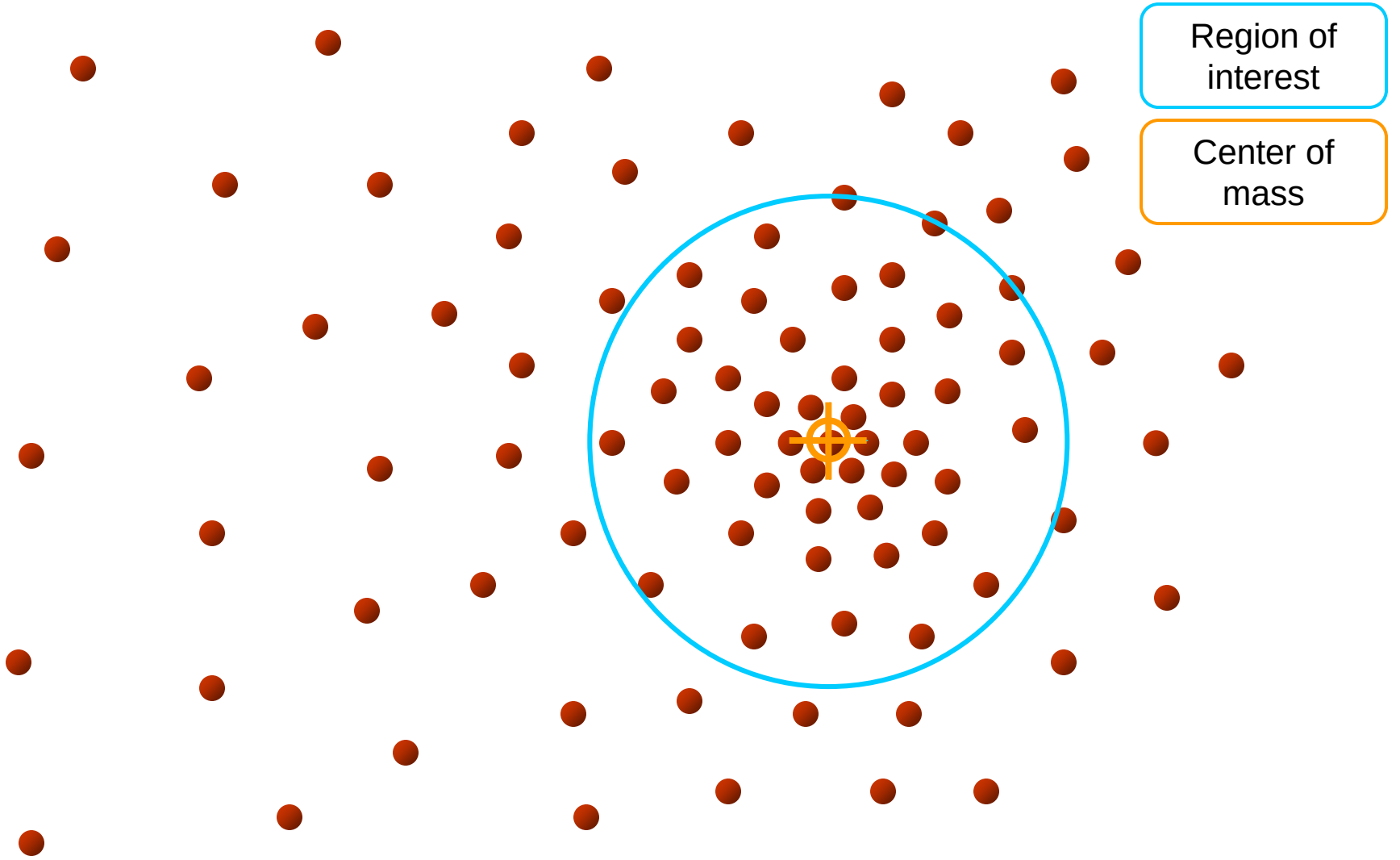
Intuitive Description



Objective : Find the densest region

Distribution of identical billiard balls

Intuitive Description



Objective : Find the densest region

Distribution of identical billiard balls

Mean-shift

- Assume:
  - $K(x)$ is a given kernel function (e.g. $K(x) = e^{-\frac{x^2}{2\sigma^2}}$)
  - $N(x)$ is the neighbourhood region of x as the set of points with $K(x) \neq 0$

- **Mean shift** is $m(x) - x$, where $m(x) = \frac{\sum_{x_i \in N(x)} K(x_i - x)\, x_i}{\sum_{x_i \in N(x)} K(x_i - x)}$
- Set $x = m(x)$ and iterate until convergence
- Repeat for each point $x \in X$

Summarizing Mean shift

PROS

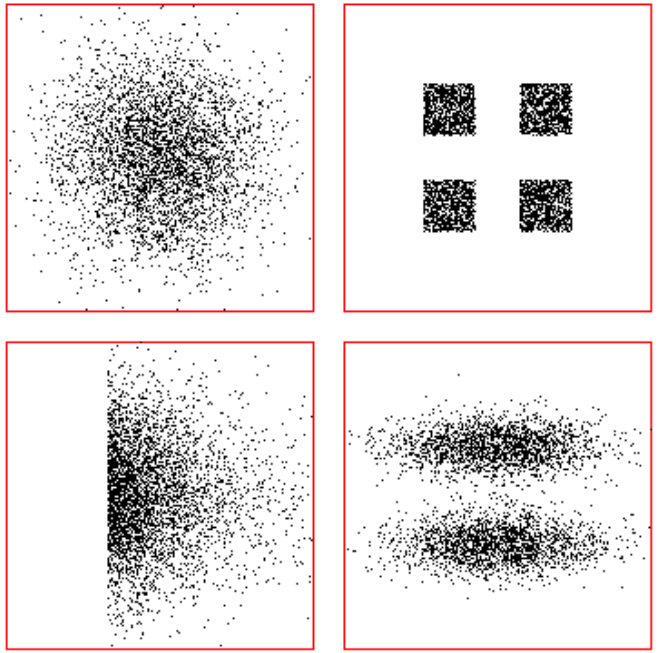
- No assumptions on data
- Arbitrary feature spaces
- Only one parameter (width of N)

CONS

- Need to select N
- Sensitive to window size (i.e. width of N)

Data Clustering

- Structures of multidimensional patterns are important for clustering
- If we know that data come from a specific distribution, such data can be *represented* by a compact set of parameters (*sufficient statistics*)
- If samples are considered coming from a specific distribution, but actually they are not, this statistics is a misleading representation of the data



data distributions having all identical statistics up to the 2nd order

- Mixture of normal distributions can approximate a large variety of situations (i.e., any density functions).
- In these cases, one can use *parametric* methods to estimate the parameters of the mixture density.
- If little prior knowledge can be assumed, the assumption of a parametric form is meaningless: we are actually imposing structure on data, not finding structure on it!
- In these cases, one can use *non parametric* methods to estimate the unknown mixture density.
- If the goal is to find subclasses, one can use a *clustering procedure* to identify groups of data points having strong internal similarities

Similarity measures

- The question is how to evaluate that the samples in one cluster are more similar among them than samples in other clusters.
- Two isses:
  - How to measure the similarity between samples?
  - How to evaluate a partitioning of a set into clusters?
- The most obvious measure of similarity (or dissimilarity) between 2 samples is the distance between them, i.e., define a *metric*.
- Once defined this measure, one would expect the distance between samples of the same cluster to be significantly less than the distance between samples in different classes.

- Euclidean distance is a possible metric: a possible criterion is to assume samples belonging to same cluster if their distance is less than a threshold d_0

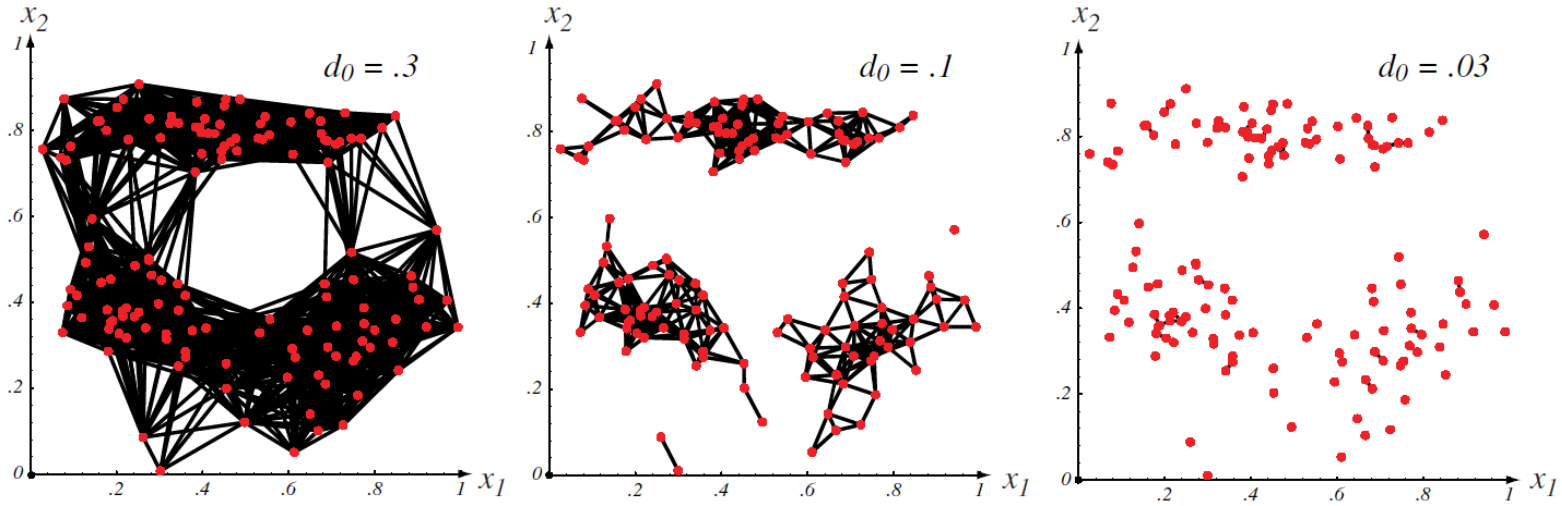


Figure 10.6: The distance threshold affects the number and size of clusters. Lines are drawn between points closer than a distance d_0 apart for three different values of d_0 — the smaller the value of d_0, the smaller and more numerous the clusters.

- Clusters defined by Euclidean distance are invariant to translations and rotation of the feature space, but not invariant to general transformations that distort the distance relationship

- To achieve invariance, one can normalize the data, e.g., such that they all have zero means and unit variance, or use principal components for invariance to rotation
- A broad class of metrics is the Minkowsky metric

$$d(\mathbf{x},\mathbf{x}') = \left(\sum_{k=1}^{d} \left| x_k - x_k' \right|^q \right)^{1/q}$$

where $q \geq 1$ is a selectable parameter:

$q = 1 \Rightarrow$ Manhattan or city block metric

$q = 2 \Rightarrow$ Euclidean metric

- One can also used a *nonmetric* similarity function $s(\mathbf{x},\mathbf{x}')$ to compare 2 vectors.

- It is typically a symmetric function whose value is large when $\mathbf{x}$ and $\mathbf{x}'$ are similar.
- For example, the inner product

$$s(\mathbf{x}, \mathbf{x}') = \frac{\mathbf{x}^t \mathbf{x}'}{\|\mathbf{x}\| \|\mathbf{x}'\|}$$

- In case of binary-valued features, we have, e.g.:

$$s(\mathbf{x}, \mathbf{x}') = \frac{\mathbf{x}^t \mathbf{x}'}{d}$$

$$s(\mathbf{x}, \mathbf{x}') = \frac{\mathbf{x}^t \mathbf{x}'}{\mathbf{x}^t \mathbf{x} + \mathbf{x}'^t \mathbf{x}' + \mathbf{x}^t \mathbf{x}'}$$

The problem of the number of clusters

- Typically, the number of clusters is known.
- When it's not, there are several ways of proceed.
- When clustering is done by extremizing a criterion function, a common approach is to repeat the clustering with c=2, c=3, c=4, etc.
- Another approach is to state a penalty for the creation of a new cluster; this is adapt to online cases but depends on the order of presentation of data.
- These approaches are similar to *model selection* procedures, typically used to determine the topology and number of states (e.g., clusters, parameters) of a model, given a specific application.