Lecture 7: Unsupervised Learning

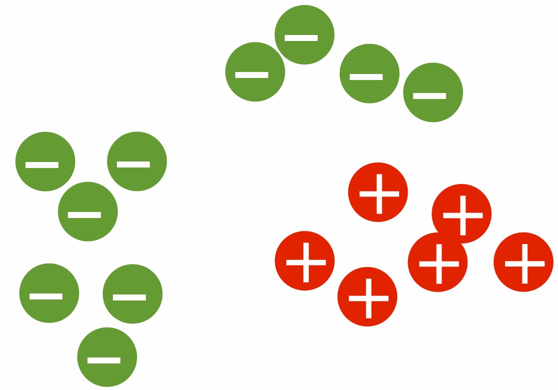
density estimation and clustering

http://cs.nju.edu.cn/yuy/course_dm13ms.ashx

Unsupervised learning

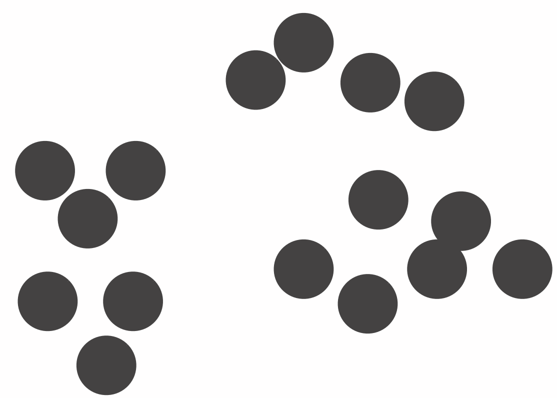
data for supervised learning

target: find a mapping $h: \mathcal{X} \to \mathcal{Y}$



data for unsupervised learning:

target: find structures of the data



what structures ?

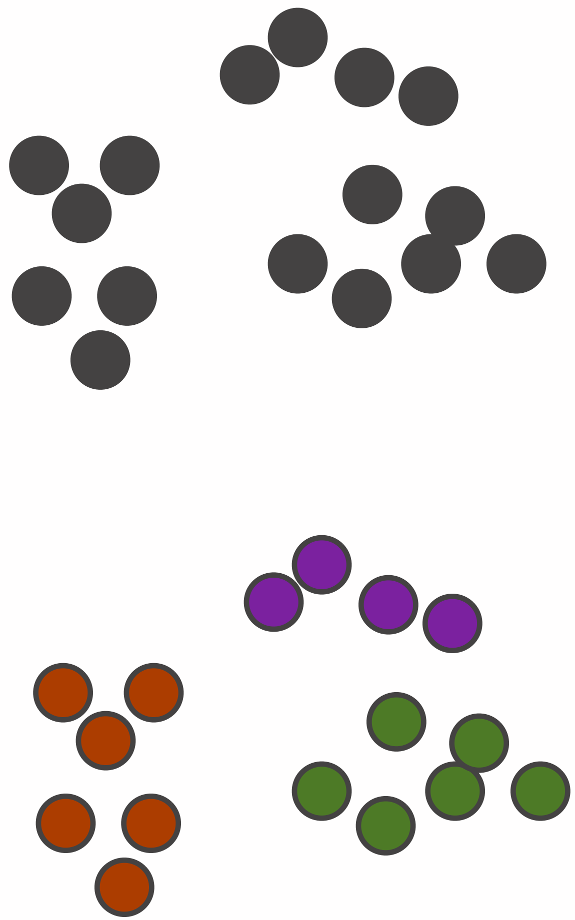
Unsupervised learning

why unsupervised learning?

natural need of discovery
of structures in data

act as a preprocessing step
to help supervised learning

...

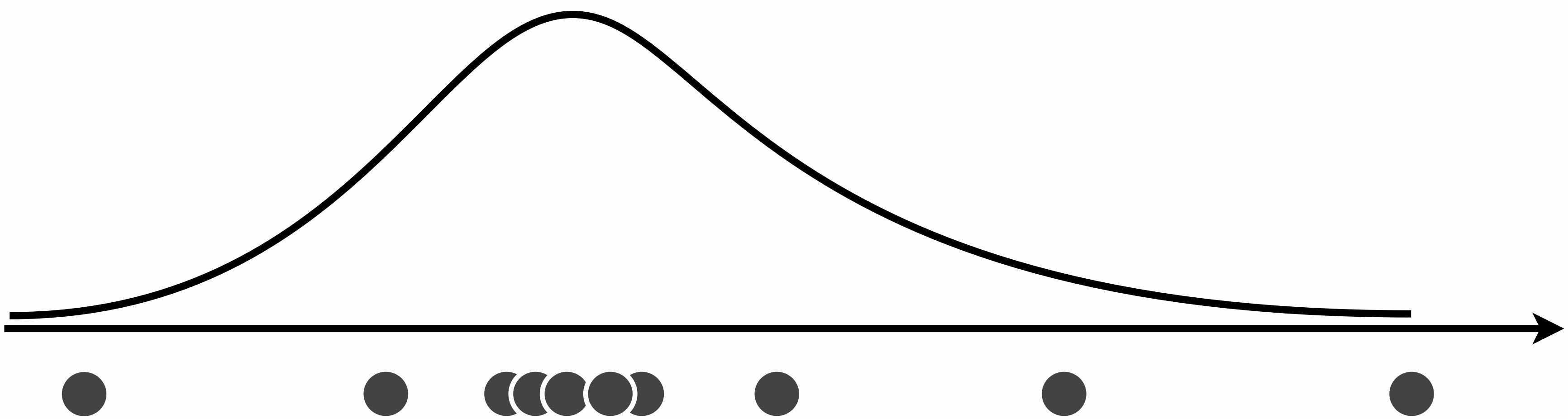


Density estimation

There exists a probability *density* function p
a data set D sampled i.i.d. from p

how large is the density at $\boldsymbol{x}$*, i.e.,* $p(\boldsymbol{x})$*?*

reconstruct p from D
estimate the density of any instance



Parametric methods

Assume the family of the density function, estimate the parameters

Normal distribution/Gaussian model:

$$p(\boldsymbol{x}) = \frac{1}{(2\pi)^{k/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}$$

Estimation:

$\boldsymbol{\mu}$ is data mean

Σ is data covariance matrix

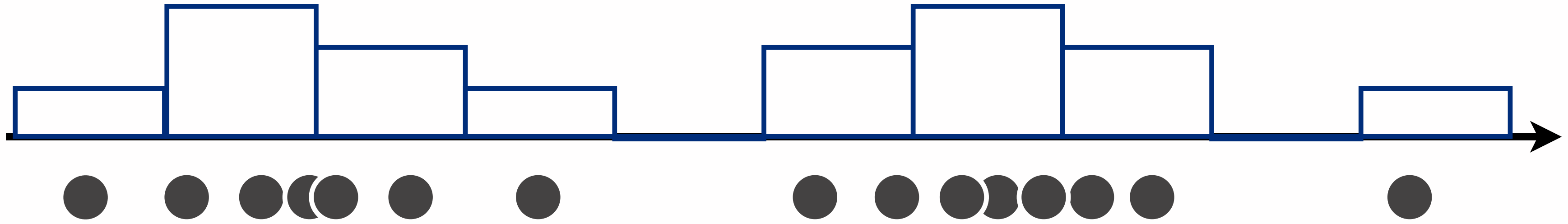
$$\Sigma = \frac{1}{n-1}\sum_{i=1}^{n}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^\top$$

Nonparametric methods

Histogram estimator

divide the input space into bins
count the frequency of instances in each bin

$$p(x) = \frac{\#\ \text{instances in bin}(x)}{m \times \text{bin-width}}$$

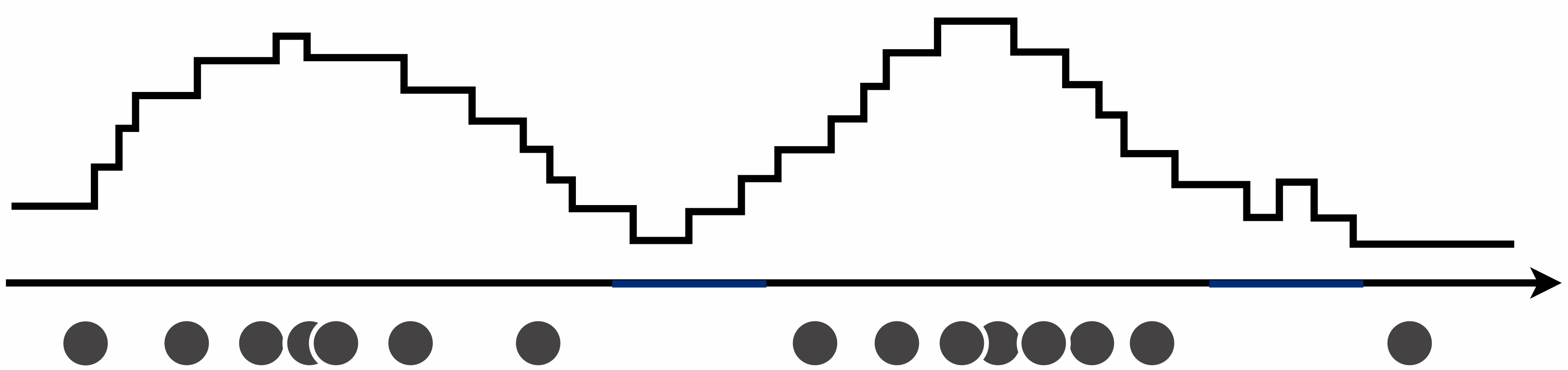


Nonparametric methods

Naive estimator

for each position, count instances in the neighbor range

$$p(x) = \frac{\#\text{ instances in } [x-h, x+h]}{m \times 2h}$$



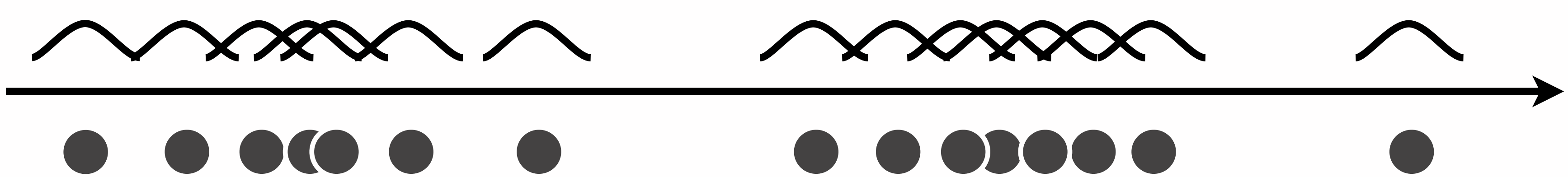
Nonparametric methods

Kernel estimator/Parzen window

for each position, the influence of an instance decreases with the distance

$$p(x) = \frac{1}{mh} \sum_{i=1}^{m} K(\frac{x - x_i}{h})$$

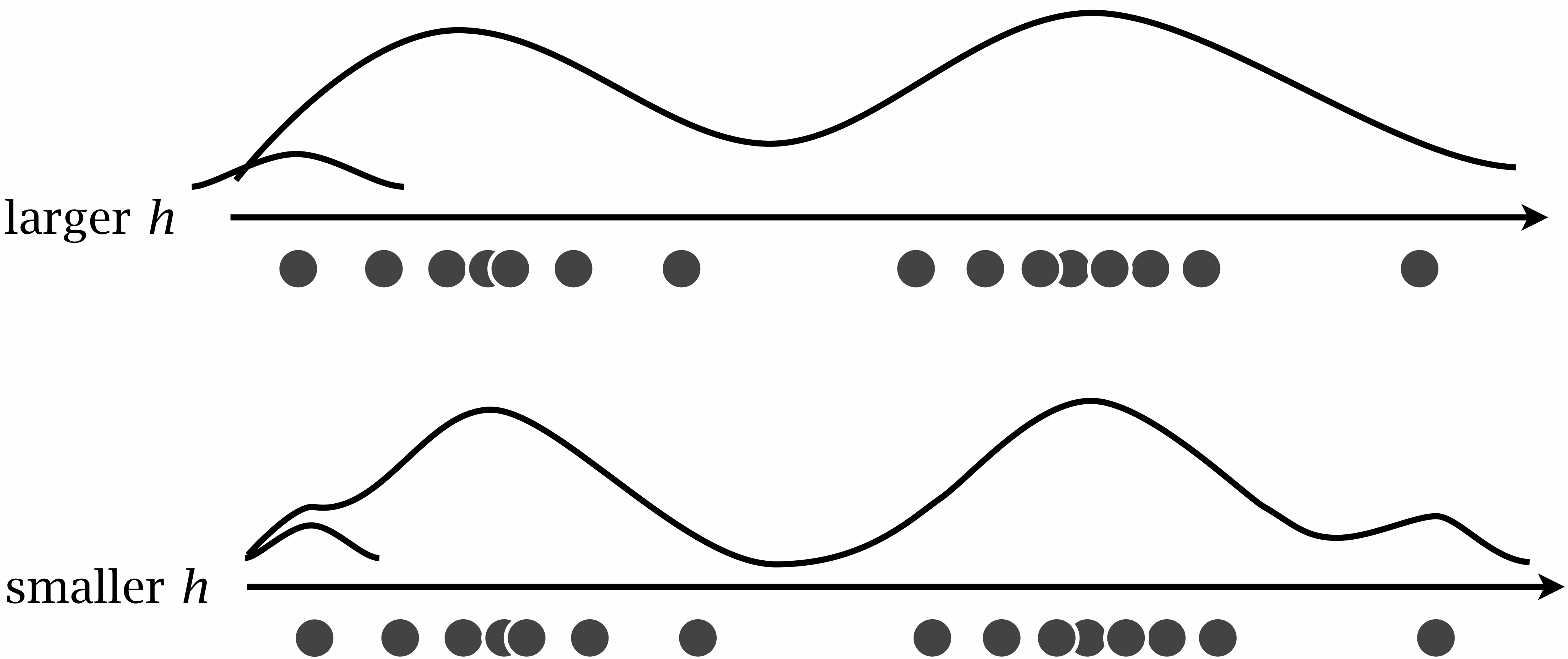
Gaussian kernel: $K(\Delta) = \frac{1}{\sqrt{2\pi}} e^{-\Delta^2/2}$



Nonparametric methods

$$p(x) = \frac{1}{mh} \sum_{i=1}^{m} K(\frac{x - x_i}{h})$$

Gaussian kernel: $K(\Delta) = \frac{1}{\sqrt{2\pi}} e^{-\Delta^2/2}$



Nonparametric methods

random partition based method (non-metric)

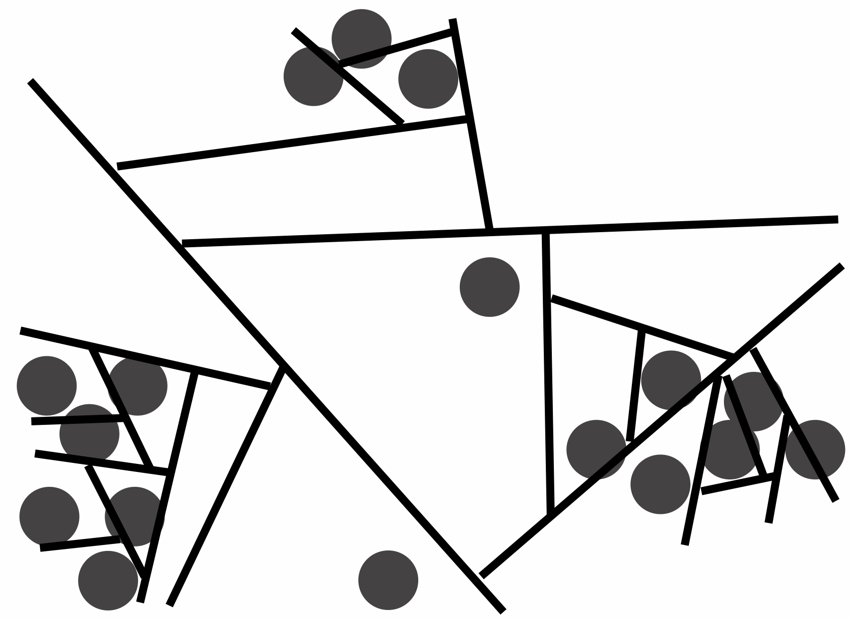
instance in low density region is easily separated

1. grow a full complete random oblique decision tree

2. the leave depth implies the density

3. build and average many trees to smooth

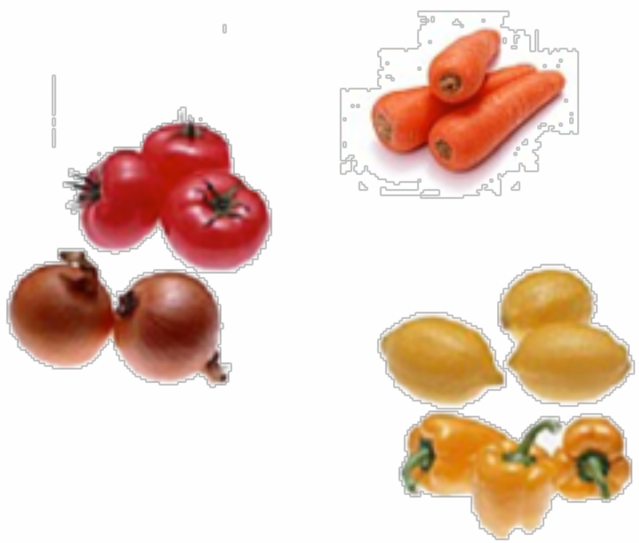
(normalization is needed)

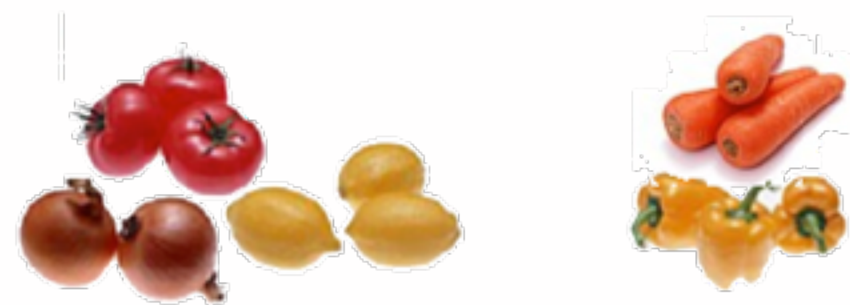


Clustering

Clustering is to find clusters in the data

Unfortunately, there is no clear definition of what should be in a cluster





the subjectivity of clustering

Clustering

hierarchical methods

density-based methods

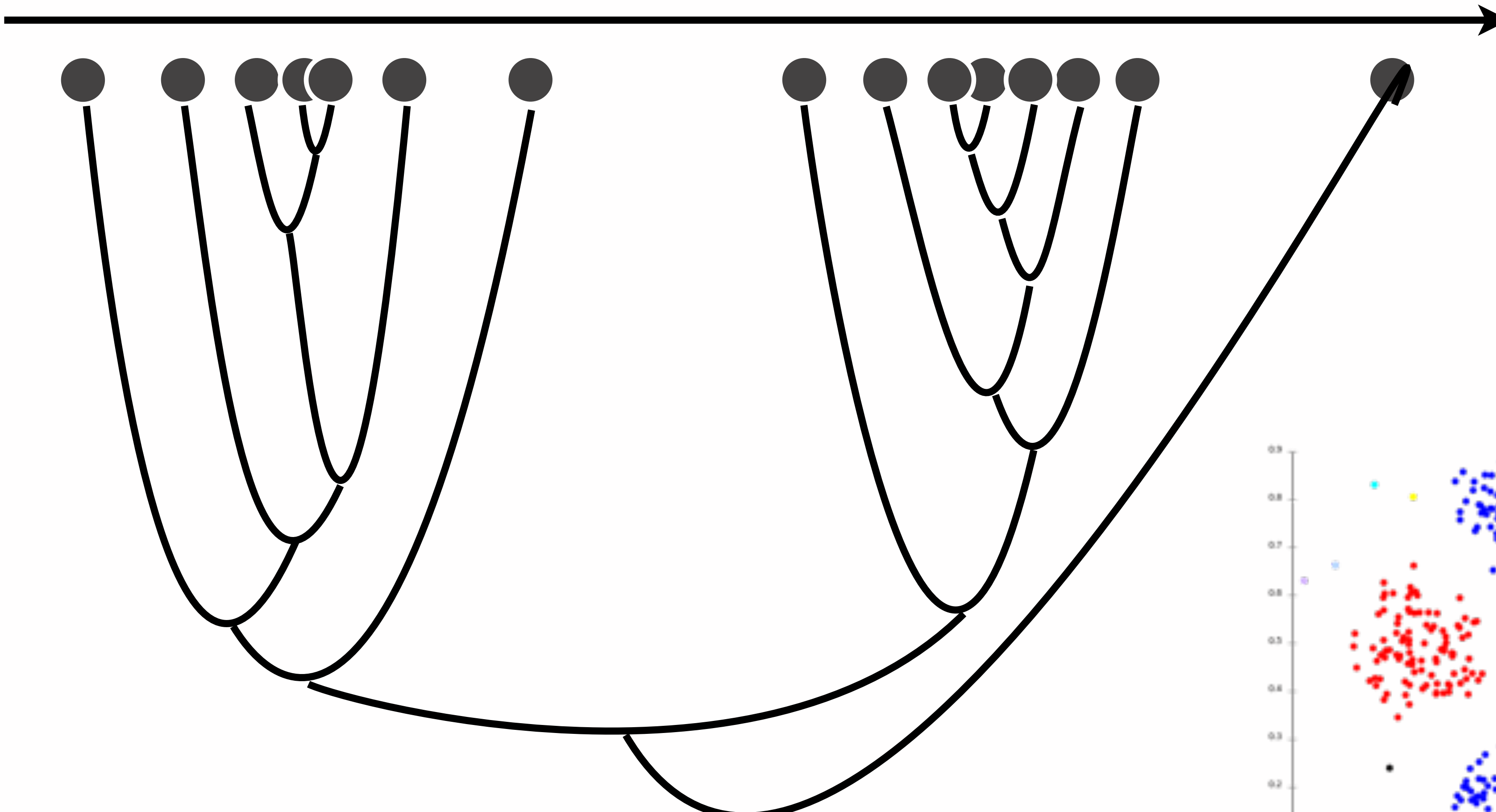
centroid-based methods

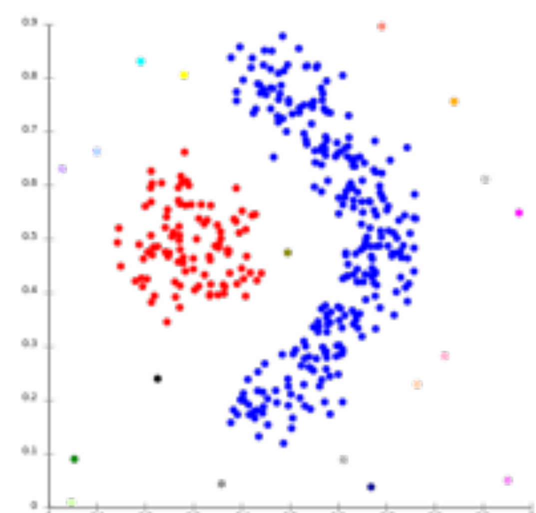
model-based methods

...

Hierarchical methods

bottom-up: single-link clustering

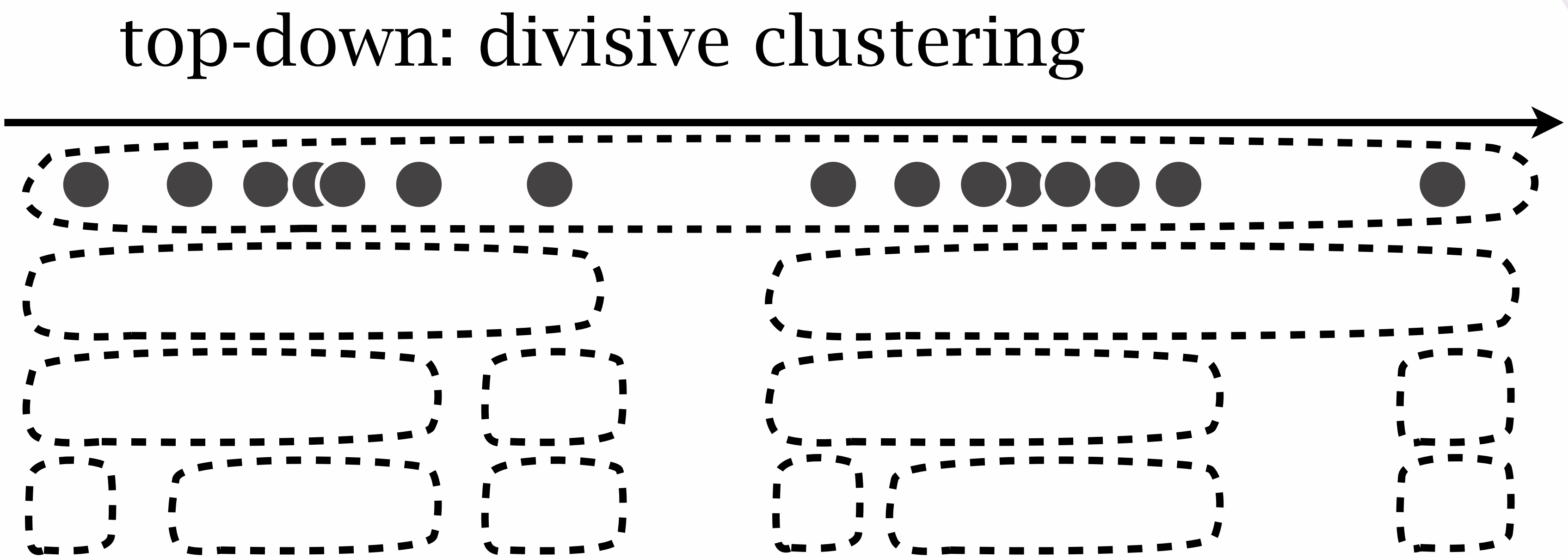




[from wikipedia]

stop at a minimum distance threshold
e.g. average distance

Hierarchical methods



separate data into two groups by maximizing the inter-group distance

expensive in each level

Density-based methods

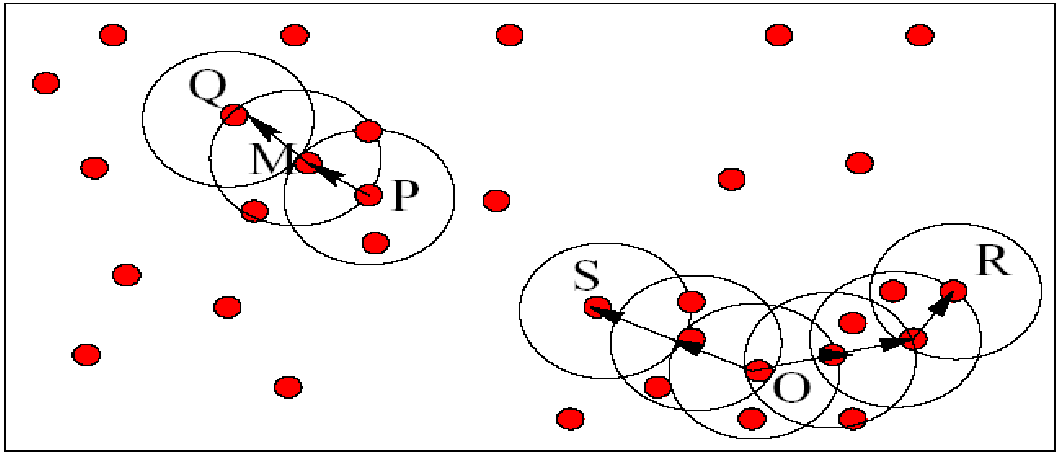
DBSCAN

focus on dense instances, clustering by connectivity

key concepts:

- an object P whose ε-neighborhood containing no less than *MinPts* number of objects is a core object with respect to ε and *MinPts*
- an object M is directly density-reachable from object P with respect to ε and *MinPts* if M is within the ε-neighborhood of P which contains at least a minimum number of points, *MinPts*
- an object Q is density-reachable from object P with respect to ε and *MinPts* if there is a chain of objects $p_1, \ldots, p_n$, $p_1 = P$ and $p_n = Q$, p_{i+1} is directly density-reachable from p_i with respect to ε and *MinPts*
- an object S is density-connected to object R with respect to ε and *MinPts* if there is an object O such that both S and R are density-reachable from O with respect to ε and *MinPts*

strictly not a clustering algorithm, leaving instances unclustered

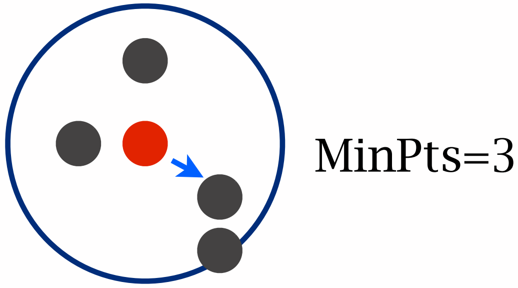


Density-based methods

OPTICS

order instances to identify the cluster structure

for each core object, calculate core-distance to be the distance to the *MinPts*-th nearest instance



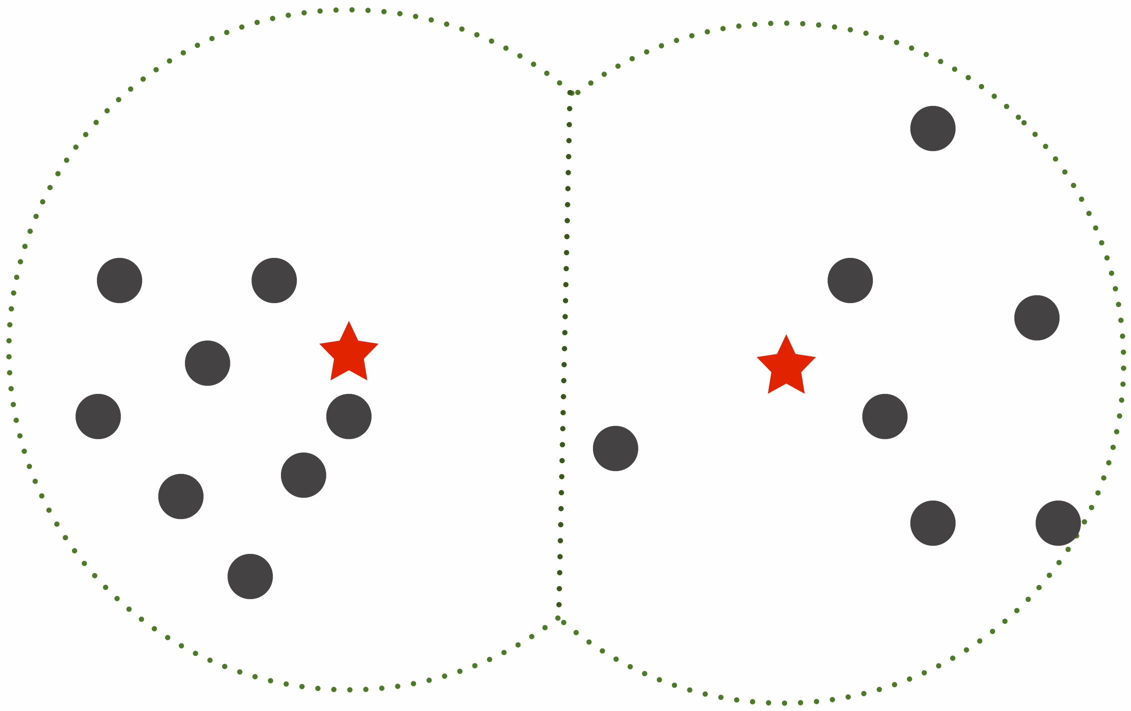
for instance *p*, calcuate reachability-distance to a core object to be

max{core-distance(o), distance(o,p)}

similar to DBSCAN, but adjust the scanning order so that closer instances are ordered closer

[Ankerst et al., SIGMOD99]

Centroid-based methods



Centroid-based methods

k-means

Step1: randomly generate *k* centers

Step2: for each instance, assign it to the cluster whose center is the nearest to the instance

fix centers, update clusters

Step3: compute the means of the cluster and regard them as the centers

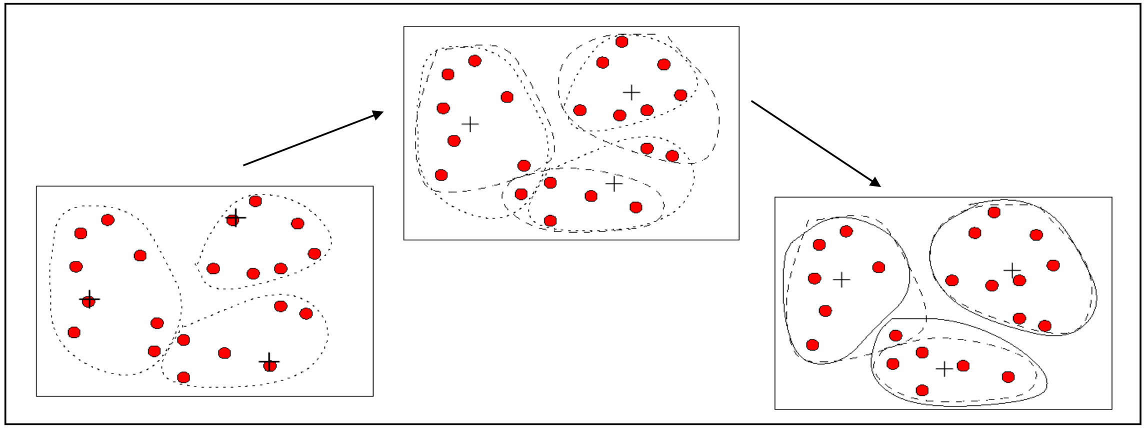
fix clusters, update centers

Step4: if there is no change, exit. otherwise go to Step2

objective:

$$\arg\min_{\boldsymbol{S}} \sum_{i=1}^{k} \sum_{\boldsymbol{x} \in S_i} \|\boldsymbol{x} - \mu_i\|^2$$

converge to local optimal



Centroid-based methods

k-medoids

Step1: randomly select k objects as the centers of the clusters

Step2: for each remaining object, assign it to the cluster whose center is the nearest to the object

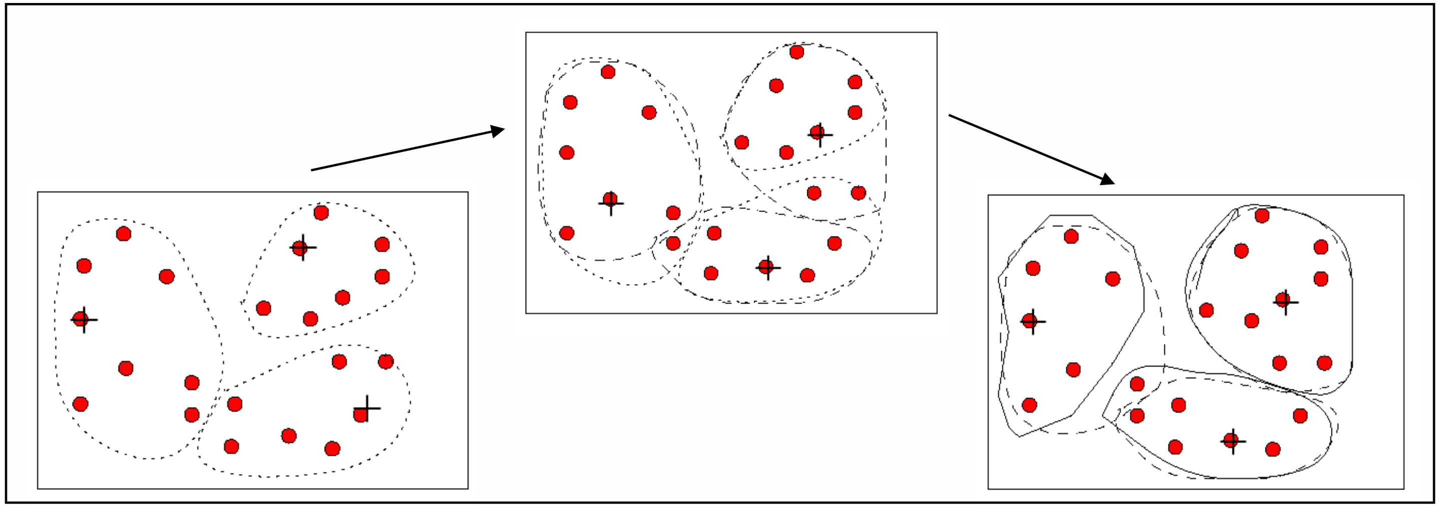
fix centers, update clusters

Step3: compute the means of the cluster, and assign the instance nearest to the mean as the centers

fix clusters, update centers

Step4: if there is no improvement, exit. otherwise go to Step2

$$\arg\min_{\boldsymbol{S}} \sum_{i=1}^{k} \sum_{\boldsymbol{x} \in S_i} \|\boldsymbol{x} - \mu_i\|^2$$



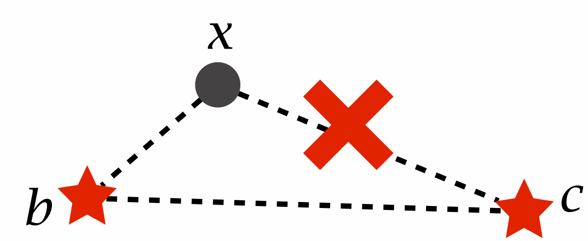
Centroid-based methods

accelerate *k*-means [Elkan, ICML03]

in the original *k*-means algorithm the later iterations do not utilize earlier information

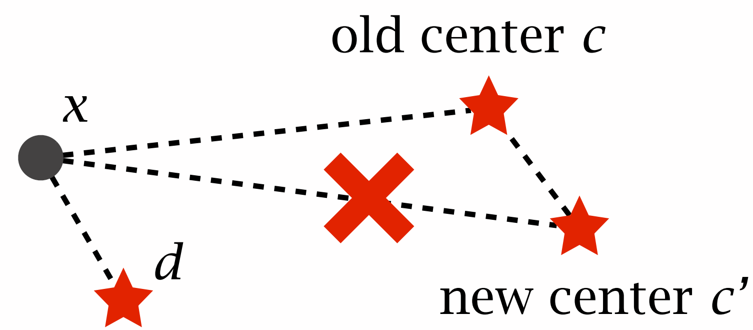
Lemma 1: Let x be a point and let b and c be centers. If $d(b,c) \geq 2d(x,b)$ then $d(x,c) \geq d(x,b)$.

when *d(x,b)* is calculated, we don't need to calculate *d(x,c)* in order to know *x* is closer to *b* than *c.*



Lemma 2: Let x be a point and let b and c be centers. Then $d(x,c) \geq \max\{0, d(x,b) - d(b,c)\}$.

when we know *d(x,c)* and that the new center moves a distance Δ, we know *d(x,c')* is at least *d(x,c)* - Δ (or 0) without calculate the exact distance.

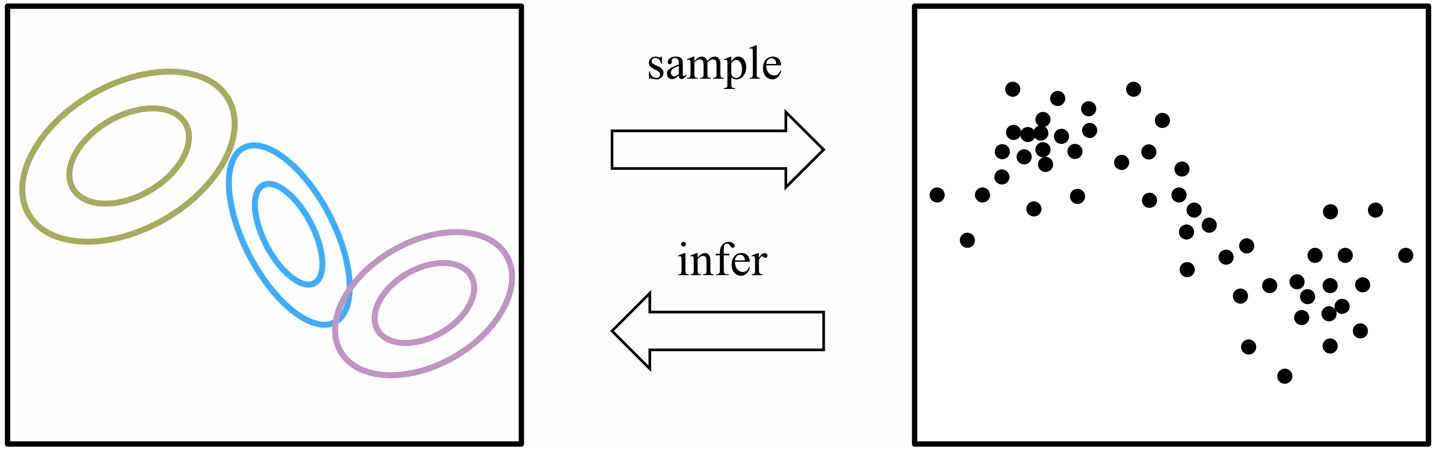


Model-based methods

Gaussian-mixture model

A perspective of dealing with unlabeled data is to imagine how the data is *generated*

assume that the data were generated from multiple Gaussian components



Clustering: To infer the Gaussian components from data

Model-based methods

Gaussian models:

Gaussian model has two parameters: $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

Density function:

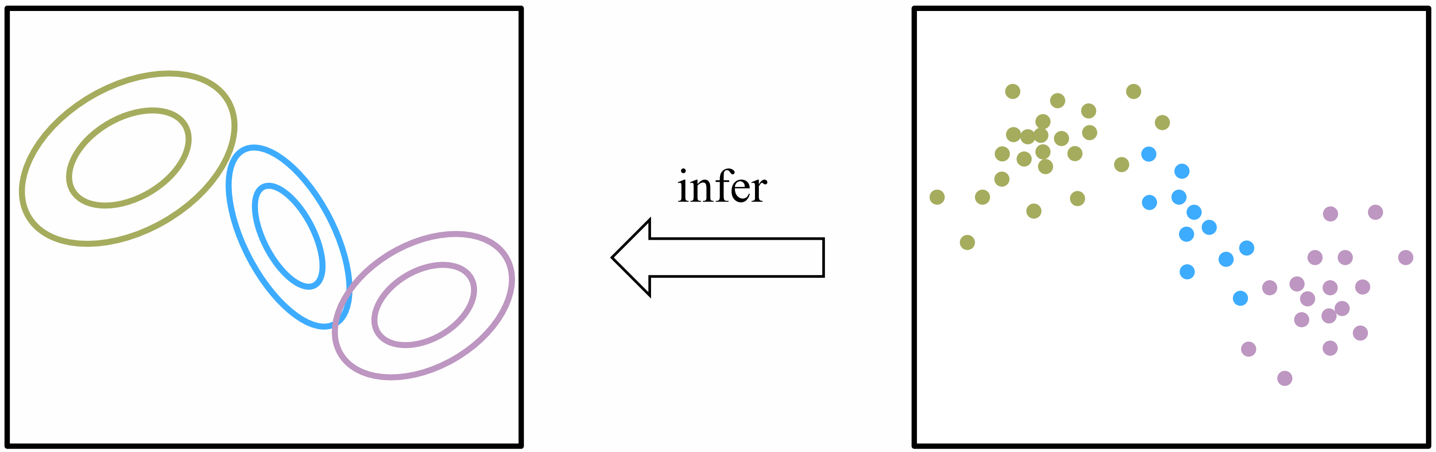
$$p(\boldsymbol{x}) = \frac{1}{(2\pi)^{k/2}|\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}$$

Log-likelihood function:

$$\ln p(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\frac{1}{2}\Big(k \ln(2\pi) + \ln|\boldsymbol{\Sigma}| + (\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\Big)$$

Model-based methods

When data clusters are known:



We know that there are three Gaussians models

for each model, calculate its parameters by maximizing the log-likelihood function: $\sum_{\boldsymbol{x}} \ln p(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$

$$\begin{cases} \partial \sum_{\boldsymbol{x}} \ln p(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) / \partial \boldsymbol{\mu} = 0 \\ \partial \sum_{\boldsymbol{x}} \ln p(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) / \partial \boldsymbol{\Sigma} = 0 \end{cases}$$

$$\boldsymbol{\mu} = \frac{1}{N} \sum_{\boldsymbol{x}} p(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \cdot \boldsymbol{x}$$

(data mean)

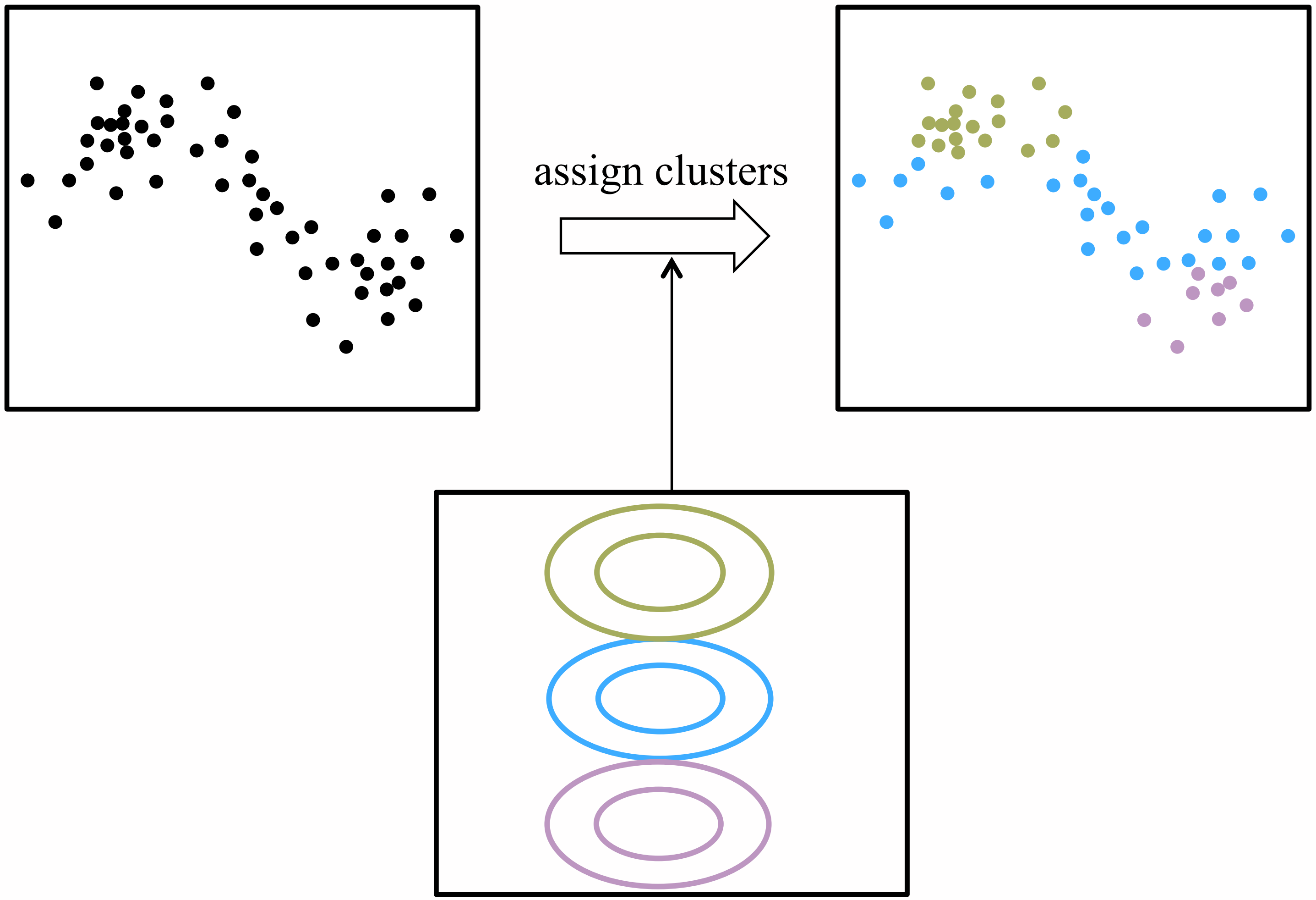
$$N = \sum_{\boldsymbol{x}} p(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

$$\boldsymbol{\Sigma} = \frac{1}{N} \sum_{\boldsymbol{x}} p(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) (\boldsymbol{x} - \boldsymbol{\mu}) (\boldsymbol{x} - \boldsymbol{\mu})^{\top}$$

(data covariance)

Model-based methods

When data clusters are unknown:



Guess the model at first!

Model-based methods

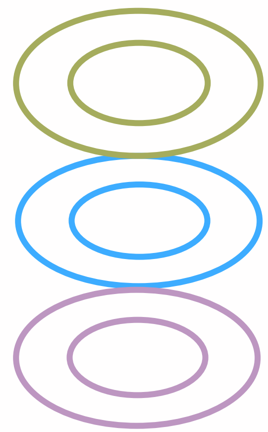
How to assign clusters to data:

Assume the models and their prior probabilities

model 1 $z = 1$

model 2 $z = 2$

model 3 $z = 3$



$$w_1 = p(z = 1) = \frac{1}{3}$$

$$w_2 = p(z = 2) = \frac{1}{3}$$

$$w_3 = p(z = 3) = \frac{1}{3}$$

density function   prior probability

Bayes rule:
$$p(z \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid z)p(z)}{p(\boldsymbol{x})}$$

Assign the cluster of the largest posterior probability

$$c(\boldsymbol{x}) = \arg\max_{i=1,2,3} p(\boldsymbol{x} \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \cdot w_i$$

Model-based methods

EM algorithm:

The original EM approach [Dempster et al, J Royal Statistical Society'77]

1. Initial guess of models (with equal prior probabilities)

$$(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, w_1 = \tfrac{1}{k}), \ldots, (\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, w_k = \tfrac{1}{k})$$

2. Assign clusters to data

$$c(\boldsymbol{x}) = \arg\max_{i=1,\ldots,k} p(\boldsymbol{x} \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \cdot w_i$$

Expectation
complete the data

3. Re-estimate model parameters from data

$$\boldsymbol{\mu}_i = \frac{1}{N_i} \sum\nolimits_{\boldsymbol{x}} p(\boldsymbol{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\boldsymbol{x}$$

$$\boldsymbol{\Sigma}_i = \frac{1}{N_i} \sum\nolimits_{\boldsymbol{x}} p(\boldsymbol{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)(\boldsymbol{x} - \boldsymbol{\mu}_i)(\boldsymbol{x} - \boldsymbol{\mu}_i)^\top$$

$$w_i = N_i / N$$

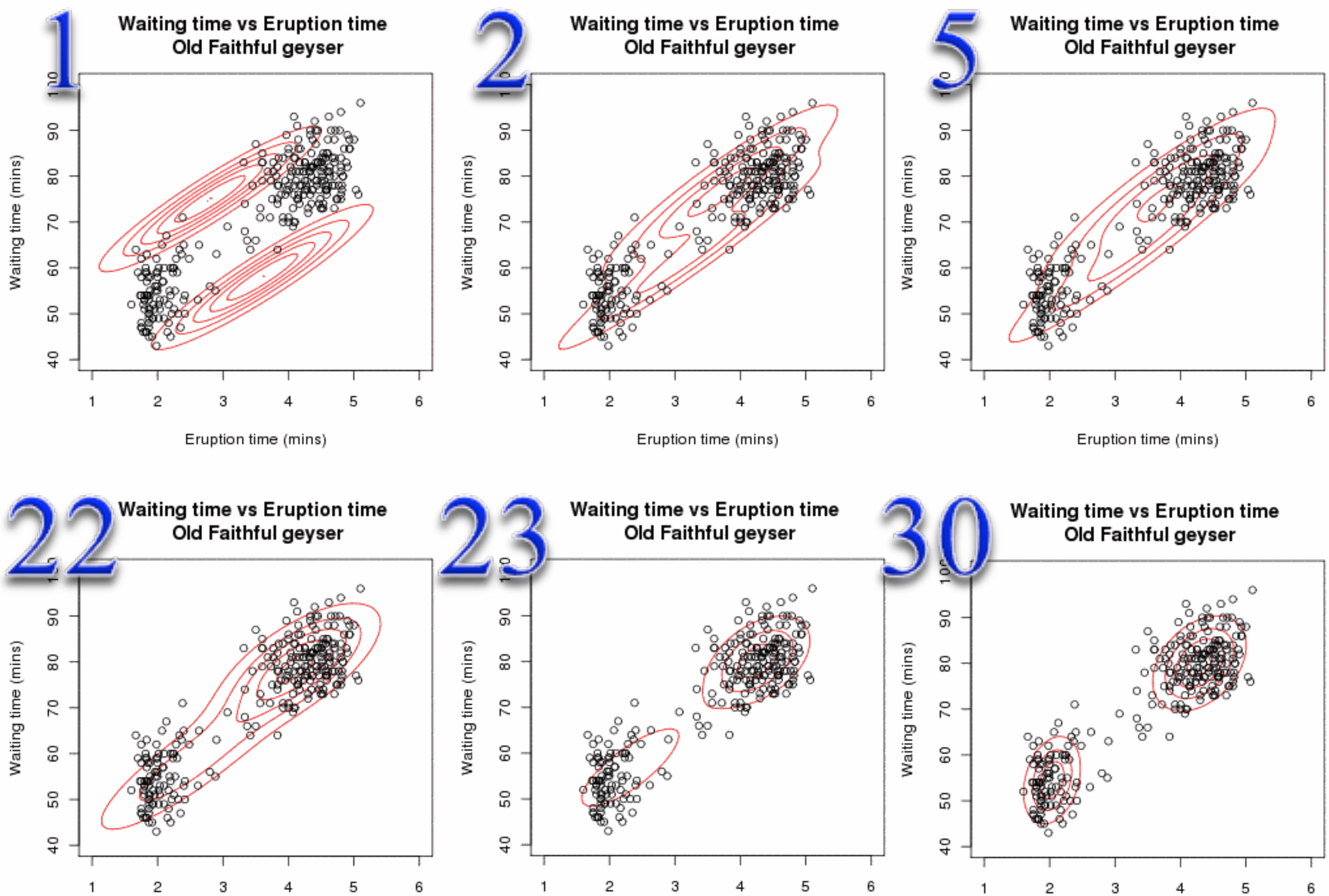
Maximization
complete the model

$N_i = \sum_{\boldsymbol{x}} p(\boldsymbol{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$

4. Go to 2 if not *converged*

Model-based methods

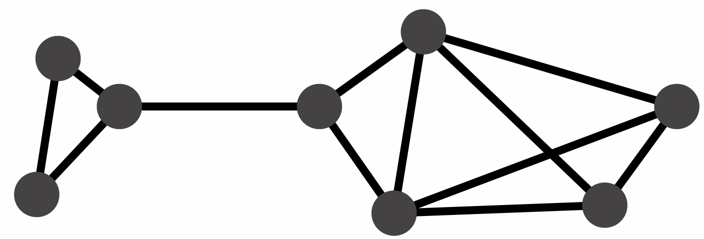
GMM example:



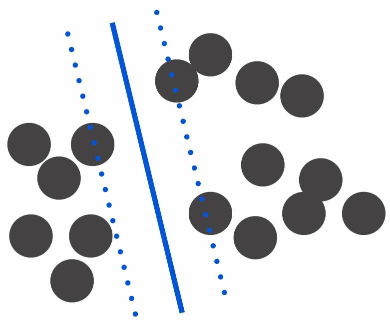
(from wikipedia)

Some other methods

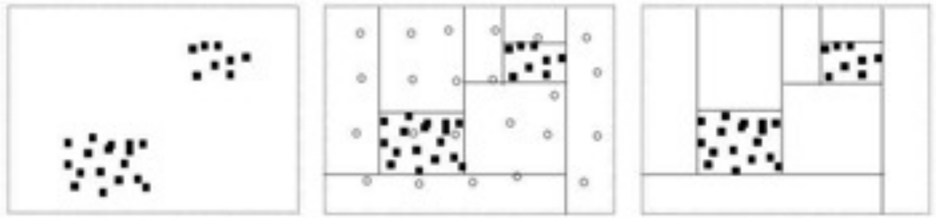
spectral clustering [Shi and Malik, PAMI00]:



maximum margin clustering [Xu *et al.*, NIPS05]:



decision tree-based clustering [Liu *et al.*, FADM05]:



Determine the number of clusters

Rule of thumb

$$k = \sqrt{n/2}$$

Cross-validation

leave a subset of data as *test data*
try different number of clusters to maximize the performance on the test data

Using density based method

Use OPTICS to find the number of clusters, then run *k*-means

...

习题

使用核密度估计(kernel estimator)方法是否会受到距离函数的影响?

k-means 聚类算法的停止条件是什么?

k-means 聚类算法的优化目标是什么?

阐述k-means聚类算法的执行过程和关键步骤。