Clustering and EM

STATS 303 Statistical Machine Learning

Spring 2022

Lecture 8

nonparametric methods (cont'd): smoothing kernels

recall: regression

- Given training set $\mathcal{X} = \{\boldsymbol{x}_n, y_n\}_{n=1}^{N}$ where $\boldsymbol{x}_n \in \mathbb{R}^d$,$y_n \in \mathbb{R}$

- Assume $y_n = g(\boldsymbol{x}_n) + \epsilon$
  - In parametric setting, we assumed a polynomial of certain order and compute its coefficients so that the sum of squared error is minimized

nonparametric regression

- Nonparametric setting:
  - We only assume that close x should have close $g(x)$
- Nonparametric approach:
  - Find the neighborhood of x, average the y values to calculate a local $\hat{g}(x)$
- Such an estimator is called a **smoother** and the estimate is called a **smooth**.

regressogram

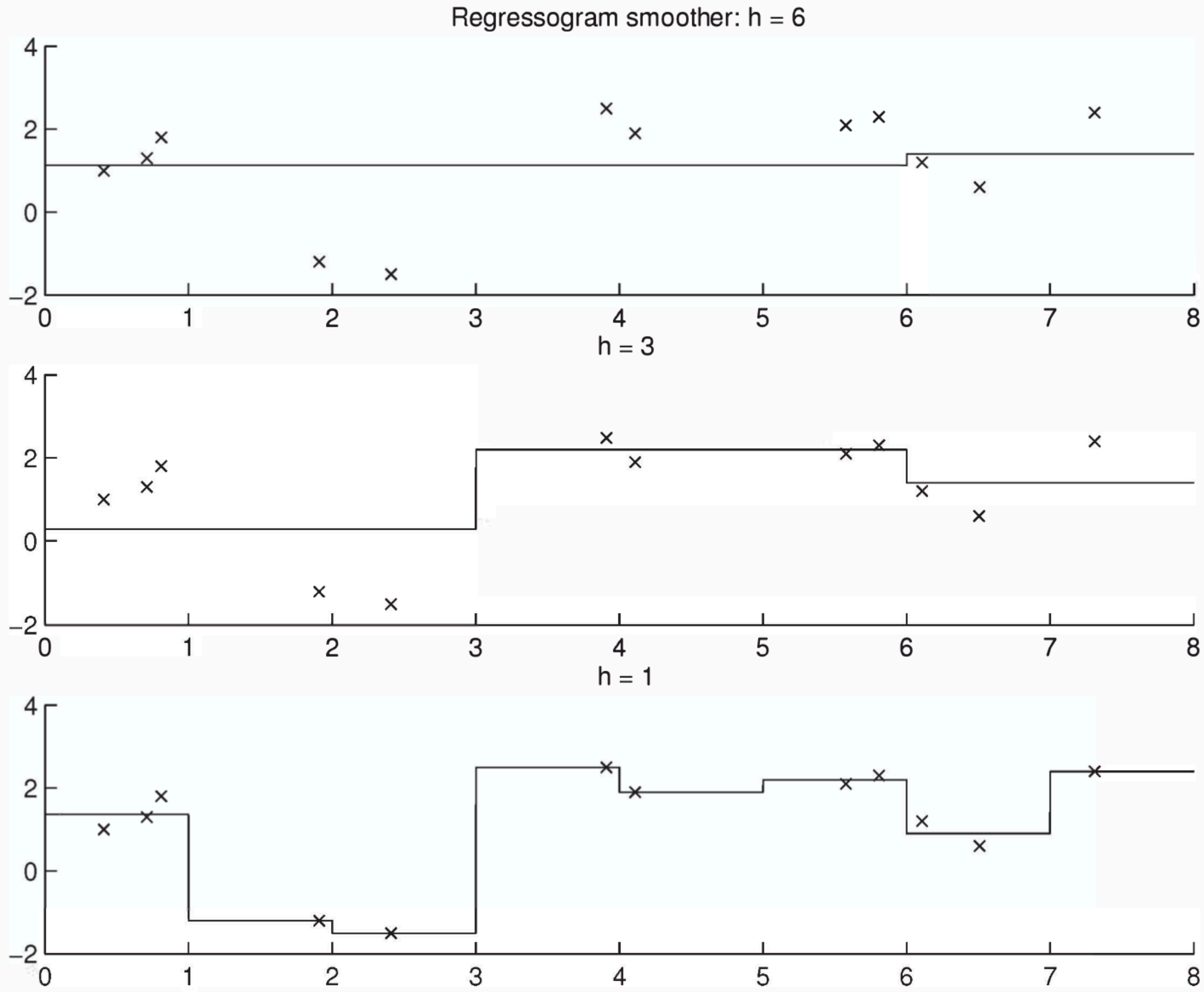


Image taken from [AI]

running mean smoother

- No fixed bins (similar to naïve estimators)

$$\hat{g}(x) = \frac{\sum_{n=1}^{N} w\left(\frac{x - x_n}{h}\right) y_n}{\sum_{t=1}^{N} w\left(\frac{x - x_n}{h}\right)}$$

where $w(u) = \begin{cases} 1\,, & \text{if } |u| < 1/2 \\ 0\,, & \text{otherwise} \end{cases}$

running mean smoother

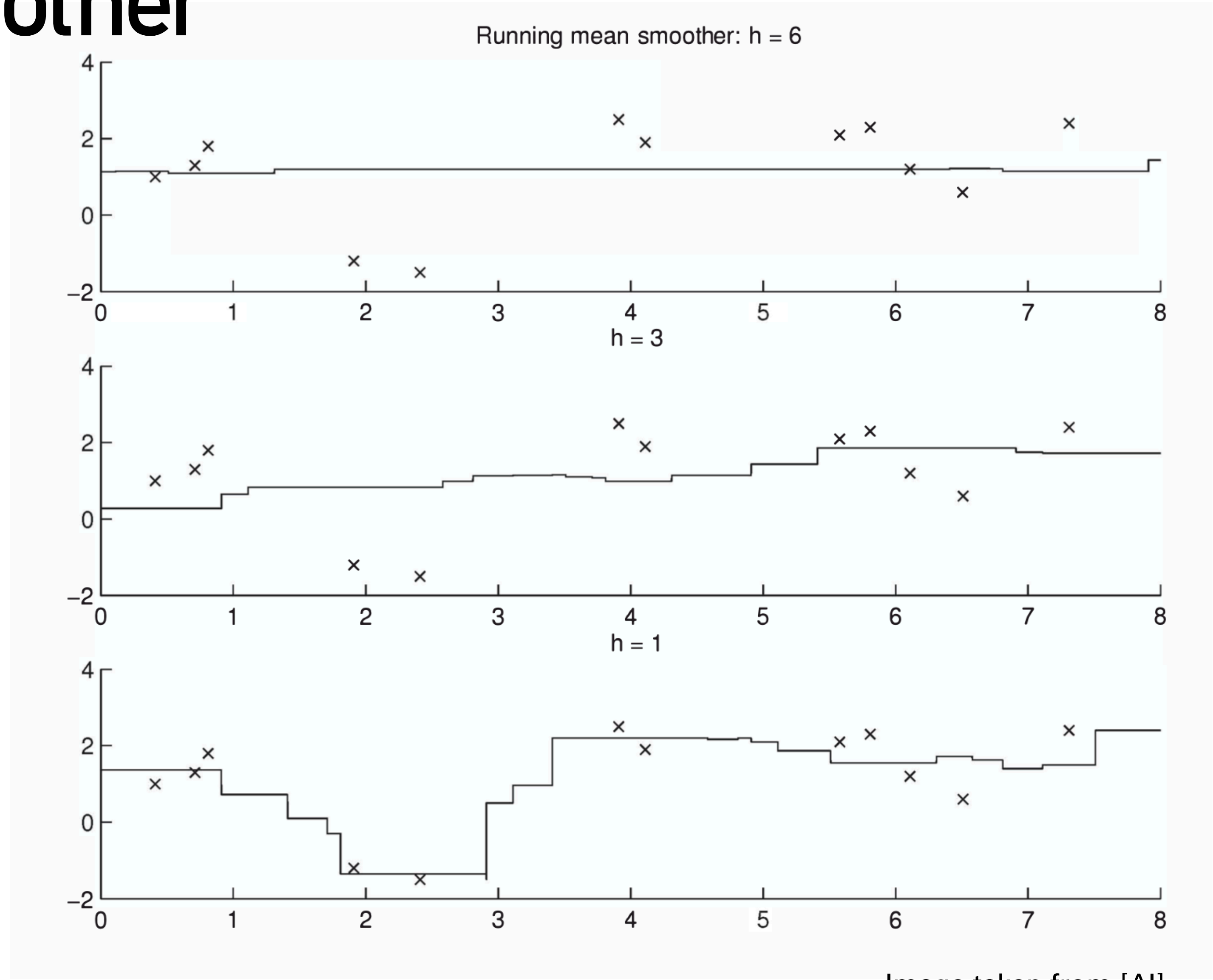


Image taken from [AI]

remark: median

1 2 3 4 50

mean : 12

median : 3

noise should be '5'.

- In either the regressogram or the running mean smoother, we tend to use the median of y_n instead of mean if there is noise in data.
- In general, the median is a more robust statistic than the mean.

kernel smoother

- We can also use a smooth kernel K

$$\hat{g}(x) = \frac{\sum_{t=1}^{N} K\left(\frac{x - x_n}{h}\right) y_n}{\sum_{t=1}^{N} K\left(\frac{x - x_n}{h}\right)}$$

- In nonparametric methods, a "kernel" is mainly used as a device for localization.
- Later in the course, we will talk about "kernel methods". In that context, kernels are used for nonlinear embedding.

kernel smoother

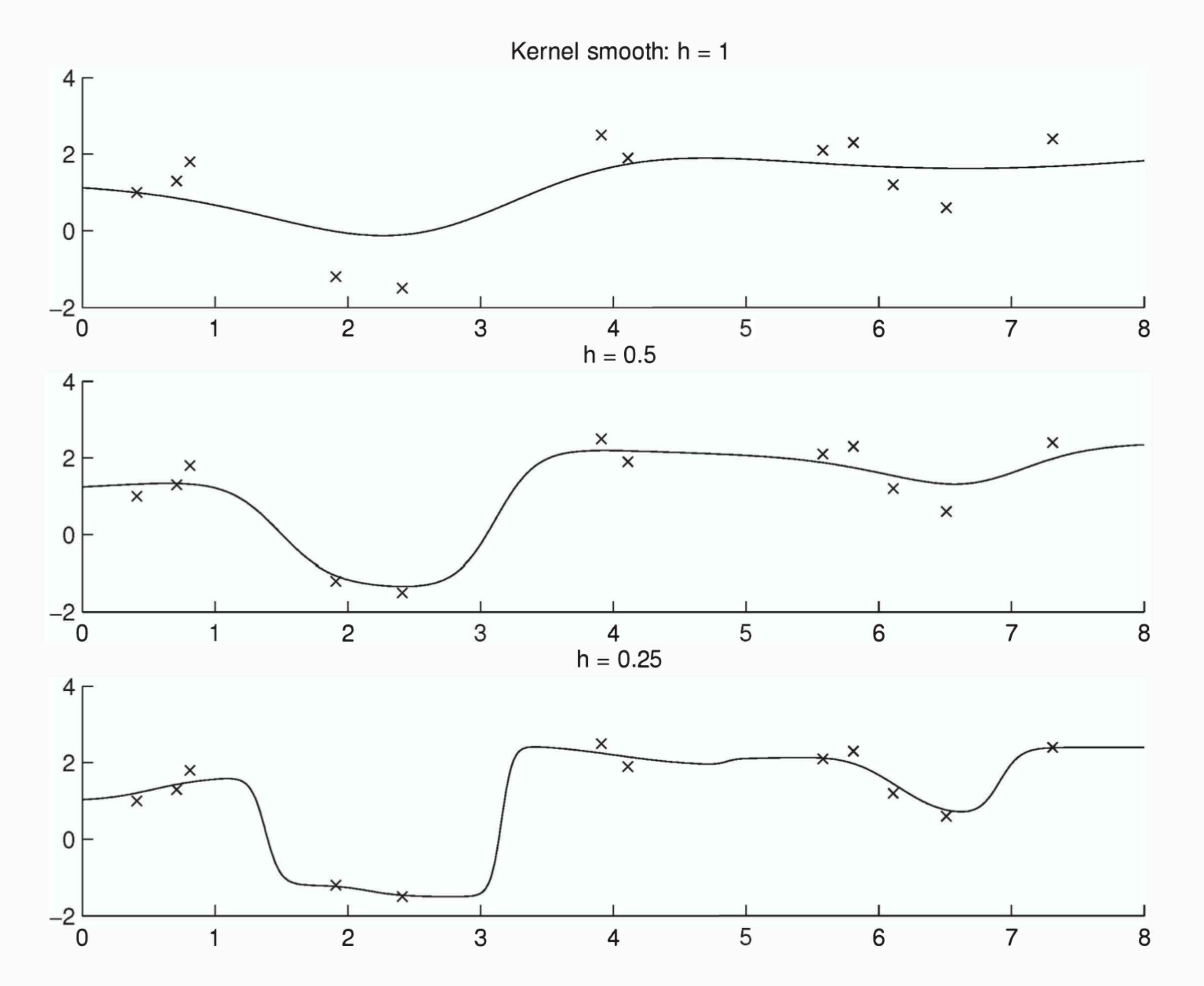


Image taken from [AI]

remark concerning matrix calculus

In machine learning. Suppose f is a function that takes value in $\mathbb{R}$.

$$\frac{\partial f}{\partial A} = \left(\frac{\partial f}{\partial A_i}\right)_{i \in I} \leftarrow \text{same shape as } A$$

vector, matrix or tensor

(Also denoted as $\nabla_A f$)

Specifically,

$\frac{\partial f}{\partial \vec{a}}$ is a vector whose i-th entry is $\frac{\partial f}{\partial a_i}$

$\frac{\partial f}{\partial \underline{A}}$ is a matrix whose (i,j)-th entry is $\frac{\partial f}{\partial A_{ij}}$

matrix

In some cases, there are nice formulae so that we can work directly with vectors / matrices.

e.g 1. $f(\vec{x}) = \vec{a}^T \vec{x} = \sum_i a_i x_i$

$$\frac{\partial f}{\partial x_i} = a_i$$

Therefore, $\frac{\partial f}{\partial \vec{x}}$ is a vector whose i-th entry is a_i

That is, $\frac{\partial f}{\partial \vec{x}} = \vec{a}$

e.g. 2 $\quad f(\vec{x}) = \frac{1}{2}\vec{x}^T A \vec{x}$ where $A = A^T$.

$$f(\vec{x}) = \frac{1}{2}\sum_i \sum_j x_i A_{ij} x_j = \frac{1}{2} A_{ii} x_i^2 + \frac{1}{2}\sum_i \sum_{i\neq j} x_i A_{ij} x_j + \frac{1}{2}\sum_{i\neq j}\sum_i x_j A_{ji} x_i$$

$$= \frac{1}{2} A_{ii} x_i^2 + \sum_i \sum_{i\neq j} A_{ij} x_j x_i$$

$$\frac{\partial f}{\partial x_i} = A_{ii} x_i + \sum_{j\neq i} A_{ij} x_j$$

$$= \sum_{j=i} A_{ij} x_j + \sum_{j\neq i} A_{ij} x_j$$

$$= \sum_j A_{ij} x_j = A_i \vec{x} = (A\vec{x})_i$$

A_i: i-th row of A.

Therefore.

$\frac{\partial f}{\partial \vec{x}}$ is a vector whose ith entry is given by $(A\vec{x})_i$

That is,

$$\frac{\partial f}{\partial \vec{x}} = A\vec{x}.$$

More examples are avaivable in Appendix C of [Bi].

e.g. 3. $\quad f(\underline{A}) = \mathrm{tr}(\underline{A}^{-1}\underline{B})$

$$\frac{\partial f}{\partial A_{ij}} = \frac{\partial}{\partial A_{ij}} \mathrm{tr}(A^{-1}B)$$

$$= \mathrm{tr}\left(\frac{\partial}{\partial A_{ij}} A^{-1} B \right)$$

$$= \mathrm{tr}\left(- A^{-1} \frac{\partial A}{\partial A_{ij}} A^{-1} B \right)$$

(see [Bi] Appendix C Eq (C.21))

matrix whose (i,j)-th entry is 1, and all the other entries are 0's.

$$= \mathrm{tr}\left(- A^{-1} \underline{e_i} e_j^T A^{-1} B \right)$$

vector whose i-th entry is 1.

$$= \mathrm{tr}\left(- e_j^T A^{-1} B A^{-1} e_i \right)$$

(since $\mathrm{tr}(XY) = \mathrm{tr}(YX)$)

$$= - e_j^T A^{-1} B A^{-1} e_i = (j,i)\text{-th entry of } A^{-1} B A^{-1}.$$

$$= (i,j)\text{-th entry of } (A^{-1} B A^{-1})^T.$$

Therefore, $\quad \dfrac{\partial f}{\partial \underline{A}} = (\underline{A}^{-1} \underline{B} \underline{A}^{-1})^T$

K-means and EM

review of K-means

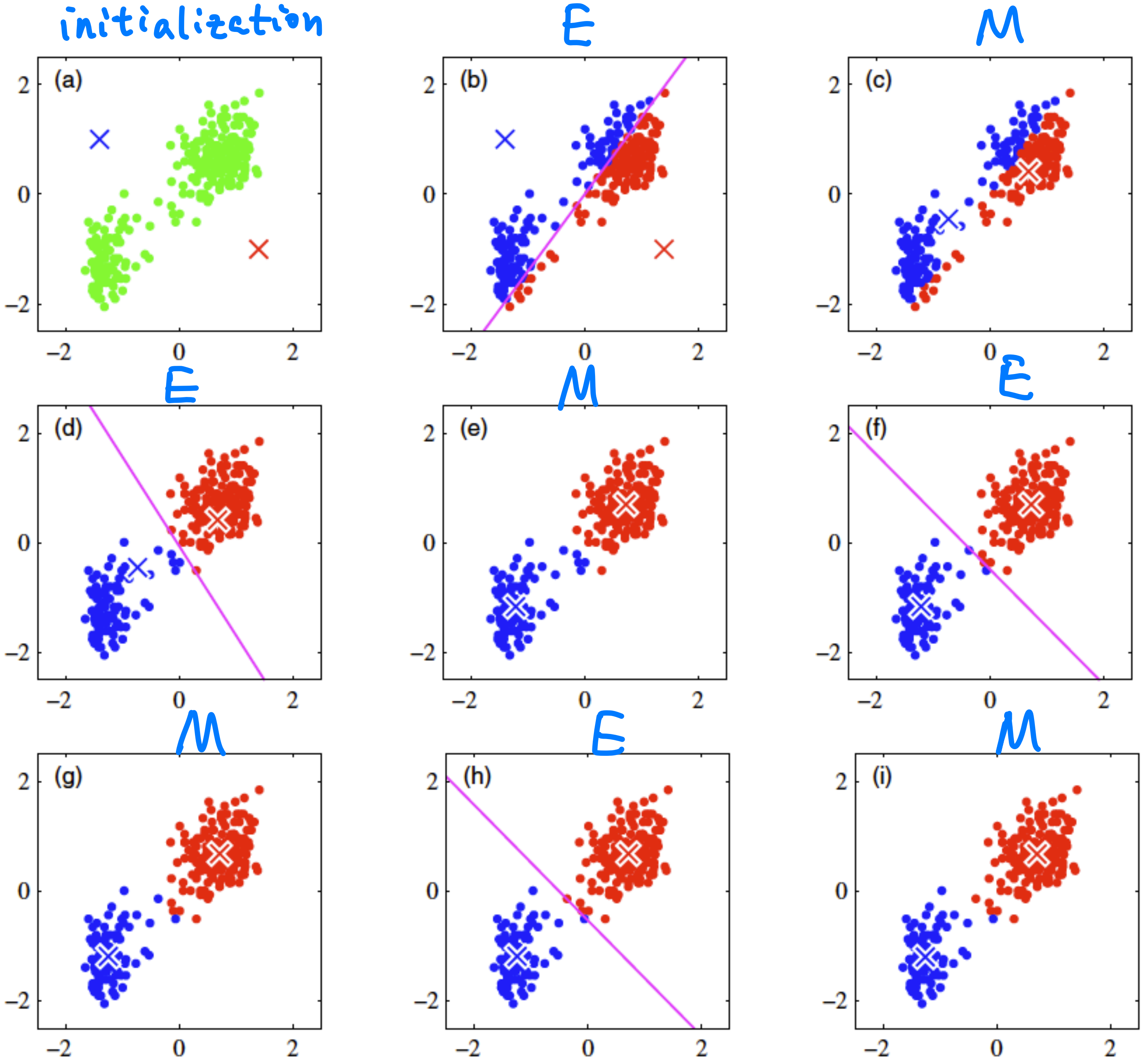


Image taken from [Bi]

review of K-means

- We can assume K "centers" of the clusters, denoted by $\mu_1, \cdots, \mu_K$
- We would like that the "total distance" between data points is small:

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

indicator:

- $r_{nk} = 1$ if $\mathbf{x}_n$ is assigned to the k-th class
- $r_{nk} = 0$ if $\mathbf{x}_n$ is not assigned to the k-th class

review of K-means

- To minimize J, we need to deal with both $\{r_{nk}\}$ and $\{\mu_k\}$, which is difficult if we want to find the global minimizer.
- Instead, we iteratively update $\{r_{nk}\}$ and $\{\mu_k\}$:
  1. (Initialization) randomly initialize $\mu_1, \cdots, \mu_K$
  2. iteratively do the following until convergence:

     (E-step): for fixed $\mu_1, \cdots, \mu_K$,find $\{r_{nk}\}$ that minimize J

     i.e., assign points to the closest center

     (M-step): for fixed $\{r_{nk}\}$,find $\mu_1, \cdots, \mu_K$ that minimize J

     i.e., calculate the sample means

review of K-means

- K-means is “hard”: assigning a point to a cluster deterministically.
- We may want to take a “softer” approach: need to consider a probabilistic view.

EM for Gaussian mixture models

overview

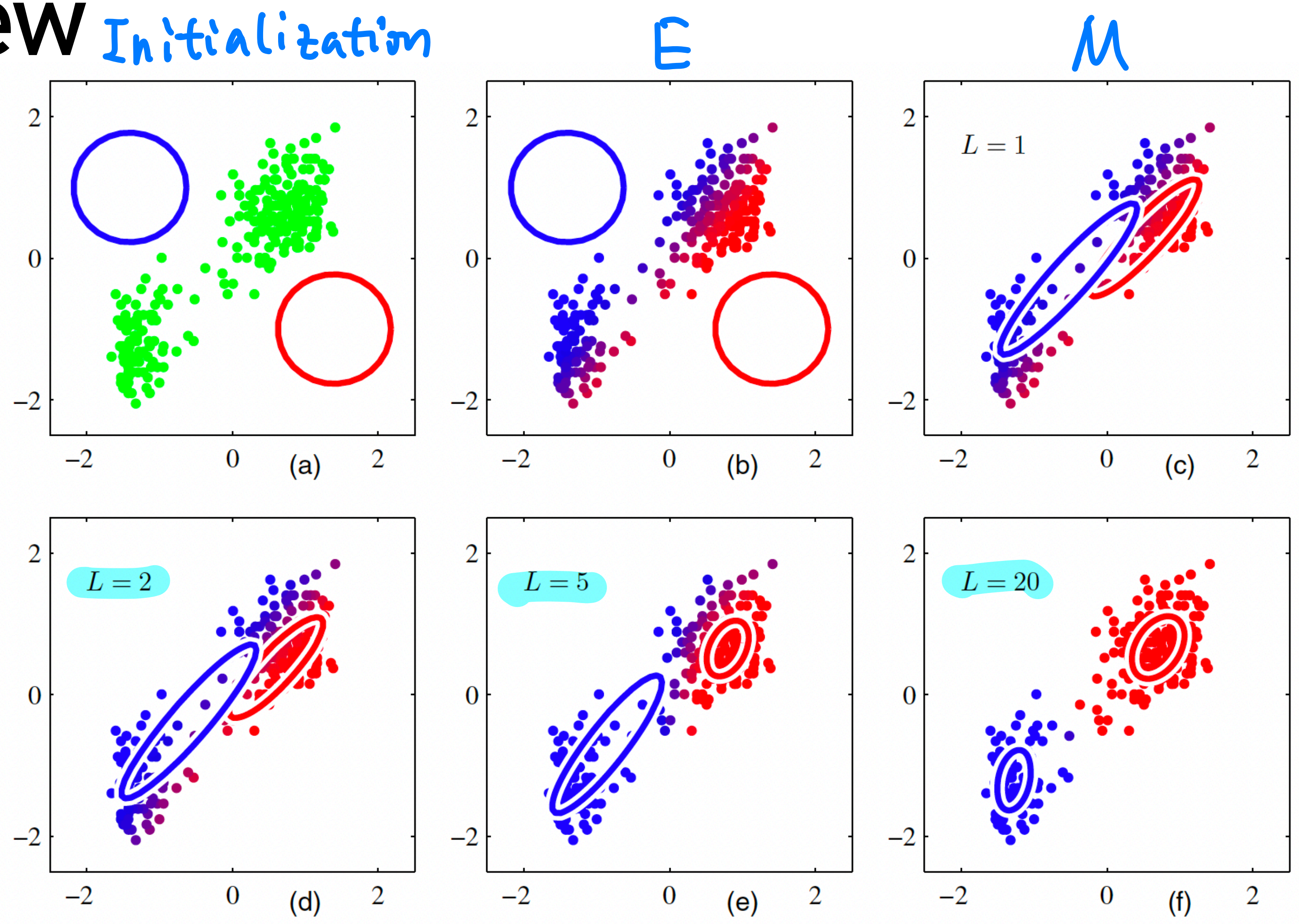


Figure 9.8 Illustration of the EM algorithm using the Old Faithful set as used for the illustration of the K-means algorithm in Figure 9.1. See the text for details.

Image taken from [Bi]

semiparametric approach

- In the parametric approach, we assumed that the sample comes from a known distribution.
- In cases when such an assumption is untenable and a nonparametric approach is not informative, we use a semiparametric approach that allows a mixture of distributions to be used for estimating the input sample.

Gaussian mixture model (GMM)

$z = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \leftarrow$ k-th entry

- Let $\mathbf{z}$ be a random variable that denotes the clustering.
  - $\mathbf{z}$ is one-hot and $z_k = 1$ implies choosing the k-th cluster.

assignment to k-th cluster.

- The marginal distribution over $\mathbf{z}$ is given by

$$p(z_k = 1) = \pi_k \Leftrightarrow P(z = e_k) = \pi_k$$

where the parameters satisfy

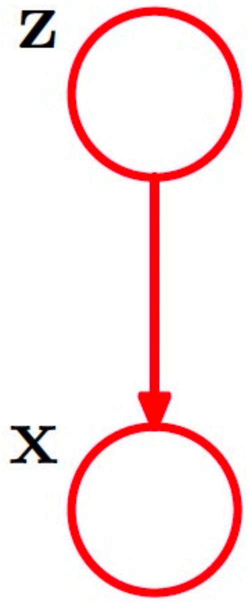
$$0 \leqslant \pi_k \leqslant 1$$

$$\sum_{k=1}^{K} \pi_k = 1$$

Gaussian mixture model (GMM)

- Similar to multi-class classification, we can write

$$p(\mathbf{z}) = \prod_{k=1}^{K} \pi_k^{z_k}$$



- In a Gaussian mixture model (GMM), the conditional distribution $p(\mathbf{x}|\mathbf{z})$ satisfies

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

$p(x|z = e_k)$

- That is, each cluster is a Gaussian. We can write

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

Gaussian mixture model (GMM)

$$\sum_{k=1}^{K} p(z=e_k)\, p(x|z=e_k)$$

- Therefore,

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

- By Bayes' Theorem,

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

"responsibility" that z_k takes in explaining $\mathbf{x}$

Questions?

Reference

- *Matrix Calculus*
  - *[Bi] Appendix C*
- *K-means:*
  - *[Al] Ch.7.3*
  - *[HaTF] Ch.13.2.1*
  - *[Bi] Ch.9.1*
- *EM:*
  - *[Al] Ch.7.2, 7.4*
  - *[HaTF] Ch.13.2.3*
  - *[Bi] Ch.9.2-9.4*
- *Spectral clustering:*
  - *[Al] Ch.6.12 7.7*
  - *[HaTF] Ch.14.5.3*