Machine Learning

Volker Roth

Department of Mathematics & Computer Science
University of Basel

Section 9

Mixture Models

Structure and mixtures

- Assume that input examples come in different potentially **unobserved types (groups, clusters, etc.).**
- Assume that
  1. there are m underlying types $z = 1, \ldots, m$;
  2. each type z occurs with probability $P(z)$;
  3. examples of type z distributed according to $p(\boldsymbol{x}|z)$.
- According to this model, each observed $\boldsymbol{x}$ comes from a **mixture distribution:**

$$p(\boldsymbol{x}) = \sum_{j=1}^{m} \underbrace{P(z=j)}_{\pi_j} p(\boldsymbol{x}|z=j, \boldsymbol{\theta}_j)$$

- In many practical data analysis problems (such as probabilistic clustering), we want to **estimate** such parametric models from **samples** $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$. In paticular, we are often interested in finding the types that have generated the examples.

Mixture of Gaussians

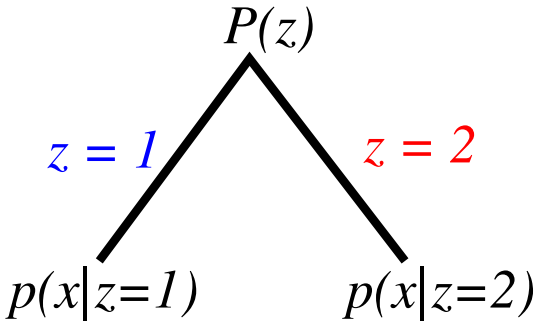
A mixture of Gaussians model has the form

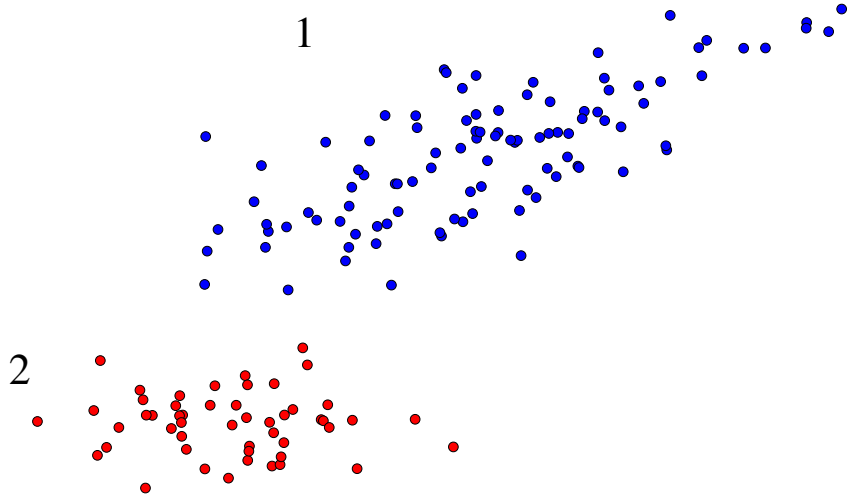
$$p(\boldsymbol{x}|\boldsymbol{\theta}) = \sum_{j=1}^{m} \pi_j \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_j, \Sigma_j),$$

where $\boldsymbol{\theta} = \pi_1, \ldots, \pi_m, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_m, \Sigma_1, \ldots, \Sigma_m$ contains all the parameters. $\{\pi_j\}$ are the **mixing proportions**.

Mixture densities

- Data generation process:



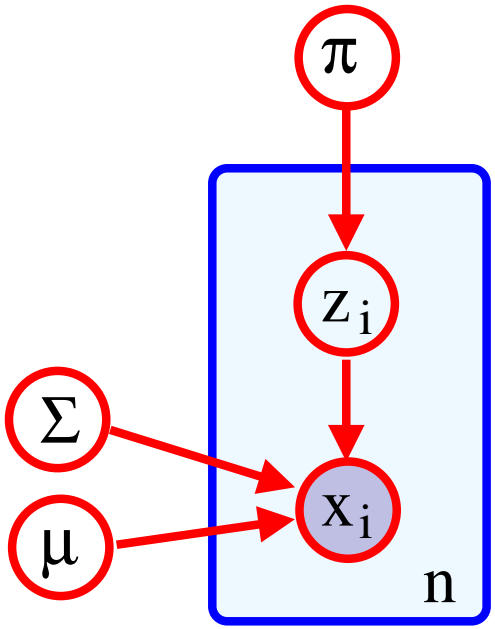


$$p(\boldsymbol{x}|\boldsymbol{\theta}) = \sum_{j=1}^{m} \pi_j p(\boldsymbol{x}|\boldsymbol{\mu}_j, \Sigma_j)$$

- Any data point $\boldsymbol{x}$ could have been **generated in two ways**.
  $\rightsquigarrow$ the responsible component needs to be **inferred**.

Mixtures as Latent Variable Models

- In the model $p(\boldsymbol{x}|z = j, \boldsymbol{\theta})$ the class indicator variable z is **latent**. This is an example of a large class of **latent variable models** (LVM).
- Bayesian network (DAG) = graphical representation of the joint distribution of RVs (nodes) as $P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|parents(x_i))$



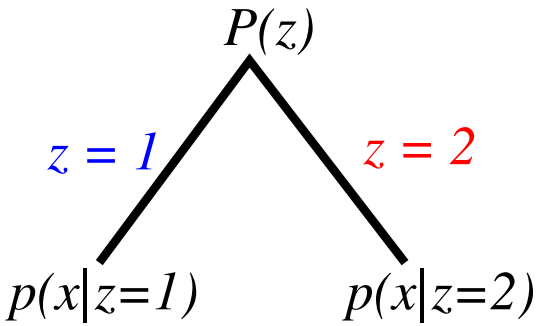
$$p(x_i|\boldsymbol{\theta}) = \sum_{z_i} p(\boldsymbol{x}_i, z_i|\boldsymbol{\theta})$$
$$= \sum_{z_i} p(\boldsymbol{x}_i|\boldsymbol{\mu}, \Sigma, z_i) p(z_i|\boldsymbol{\pi}).$$

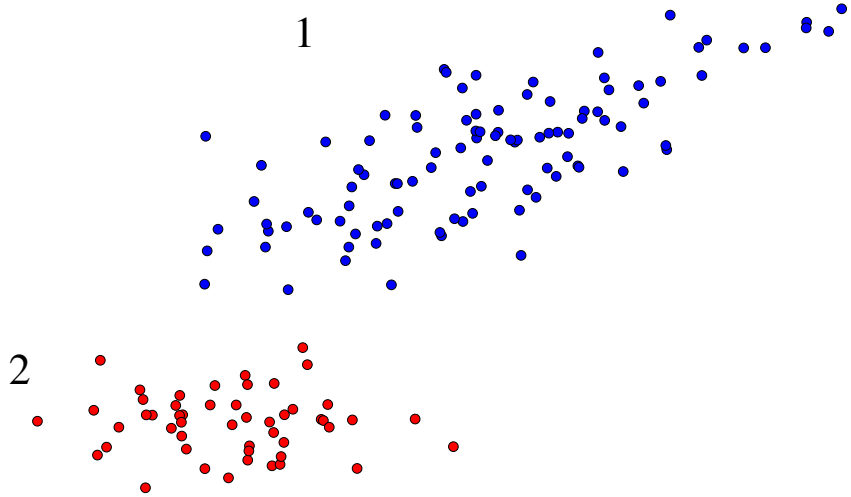
Mixture densities

- Consider a two component mixture of Gaussians model.

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = \pi_1 p(\boldsymbol{x}|\boldsymbol{\mu}_1, \Sigma_1) + \pi_2 p(\boldsymbol{x}|\boldsymbol{\mu}_2, \Sigma_2)$$

- If we knew the generating component $z_i = \{1, 2\}$ for each example $\boldsymbol{x}_i$, then the estimation would be easy.





- In particular, we can estimate each Gaussian independently.

Mixture density estimation

- Let $\delta(j|i)$ be an indicator function of whether example i is labeled j. Then for each $j = 1, 2$

$$\begin{aligned}
\hat{\pi}_j &\leftarrow \frac{\hat{n}_j}{n}, \quad \text{where } \hat{n}_j = \sum_{i=1}^{n} \delta(j|i) \\
\hat{\boldsymbol{\mu}}_j &\leftarrow \frac{1}{\hat{n}_j} \sum_{i=1}^{n} \delta(j|i) \boldsymbol{x}_i \\
\hat{\Sigma}_j &\leftarrow \frac{1}{\hat{n}_j} \sum_{i=1}^{n} \delta(j|i) (\boldsymbol{x}_i - \boldsymbol{\mu}_j)(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^t
\end{aligned}$$

Mixture density estimation

- We don't have such labels... but we can guess what the labels might be based on our current distribution.
- One possible choice: evaluate posterior probability that an observed $\boldsymbol{x}$ was generated from first component

$$\begin{aligned} P(z=1|\boldsymbol{x},\boldsymbol{\theta}) &= \frac{P(z=1)\cdot p(\boldsymbol{x}|z=1)}{\sum_{j=1,2} P(z=j)\cdot p(\boldsymbol{x}|z=j)} \\ &= \frac{\pi_1 p(\boldsymbol{x}|\boldsymbol{\mu}_1,\Sigma_1)}{\sum_{j=1,2}\pi_j p(\boldsymbol{x}|\boldsymbol{\mu}_j,\Sigma_j)} \end{aligned}$$

  $\rightsquigarrow$ Information about the component responsible for generating $\boldsymbol{x}$.
- **Soft labels** or **posterior probabilities**

  $$\hat{p}(j|i) \leftarrow P(z_i = j|\boldsymbol{x}_i,\boldsymbol{\theta}),$$

  where $\sum_{j=1,2}\hat{p}(j|i) = 1,\ \forall\ i = 1,\ldots,n.$

The EM algorithm: iteration k

- **E-step:** softly assign examples to mixture components

  $$\hat{p}(j|i) \leftarrow P(z_i = j|\boldsymbol{x}_i, \boldsymbol{\theta}^t), \forall j = 1, 2 \text{ and } i = 1, \ldots, n.$$

  Note: superscript is time index.

- **M-step:** estimate new mixture parameters $\boldsymbol{\theta}^{t+1}$ based on the soft assignments (can be done separately for the two Gaussians)

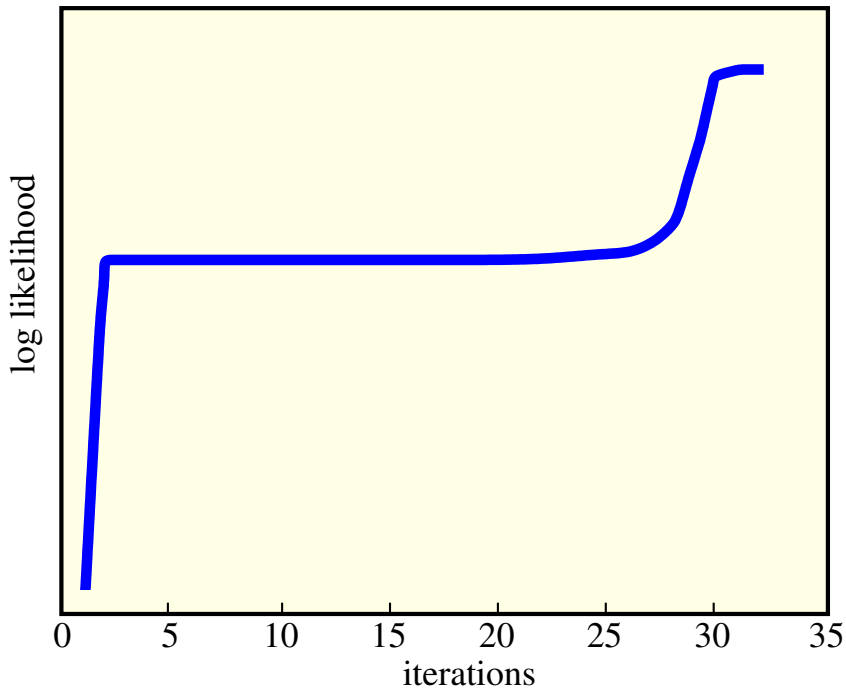
$$\begin{aligned}
\hat{\pi}_j &\leftarrow \frac{\hat{n}_j}{n}, \quad \text{where } \hat{n}_j = \sum_{i=1}^{n} \hat{p}(j|i) \\
\hat{\boldsymbol{\mu}}_j &\leftarrow \frac{1}{\hat{n}_j} \sum_{i=1}^{n} \hat{p}(j|i) \boldsymbol{x}_i \\
\hat{\Sigma}_j &\leftarrow \frac{1}{\hat{n}_j} \sum_{i=1}^{n} \hat{p}(j|i) (\boldsymbol{x}_i - \boldsymbol{\mu}_j)(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^t
\end{aligned}$$

The EM-algorithm: Convergence

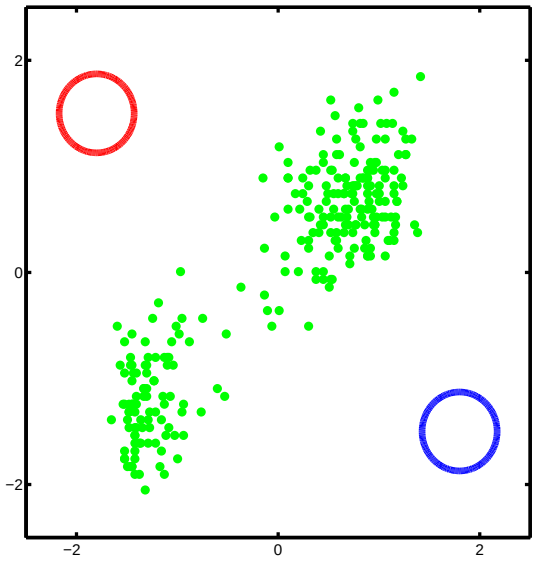
The EM-algorithm **monotonically increases the log-likelihood** of the training data (we will show this later). In other words,

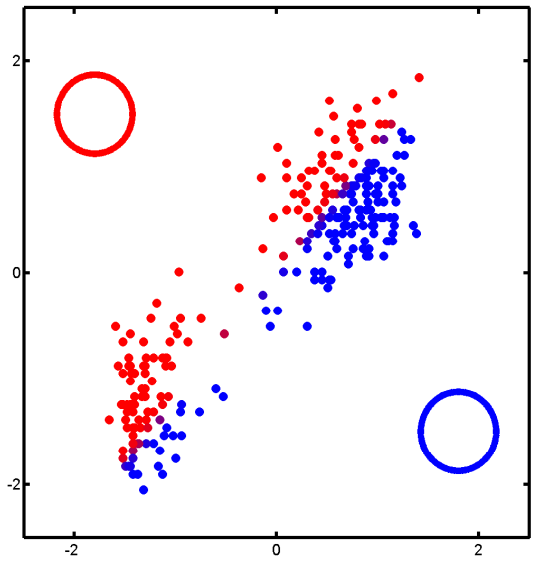
$$l(\boldsymbol{\theta}^0) < l(\boldsymbol{\theta}^1) < l(\boldsymbol{\theta}^2) < \ldots \text{ until convergence}$$

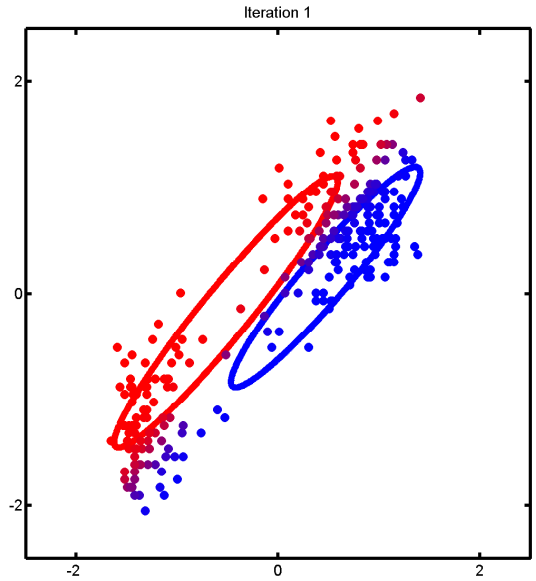
$$l(\boldsymbol{\theta}^t) = \sum_{i=1}^{n} \log p(\boldsymbol{x}_i|\boldsymbol{\theta}^t).$$

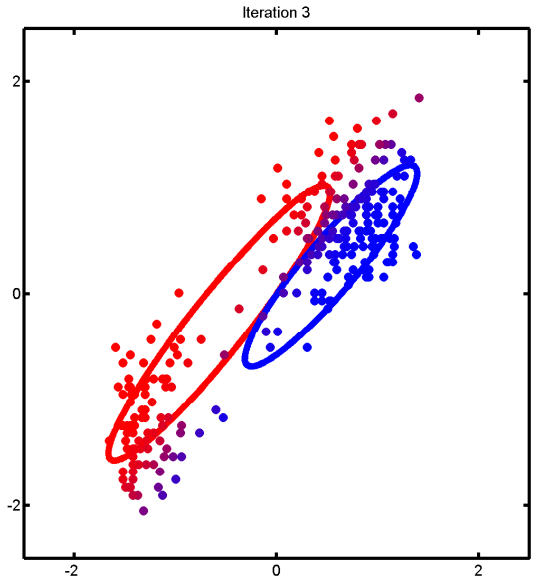


Mixture density estimation: example









Mixture density estimation: example

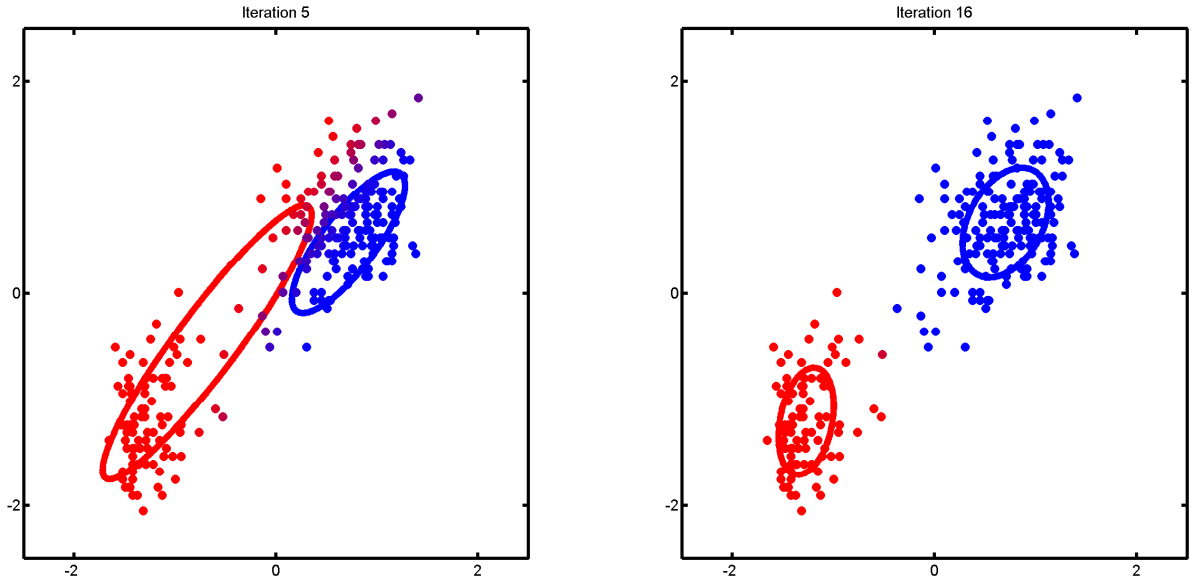
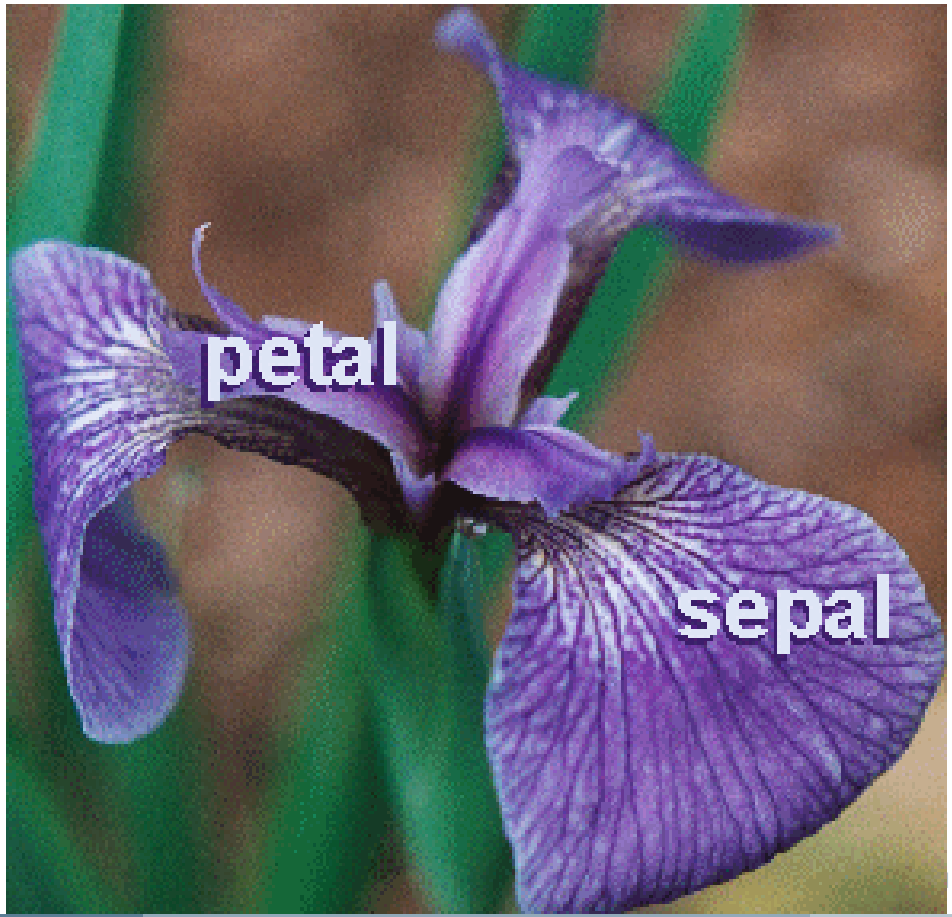
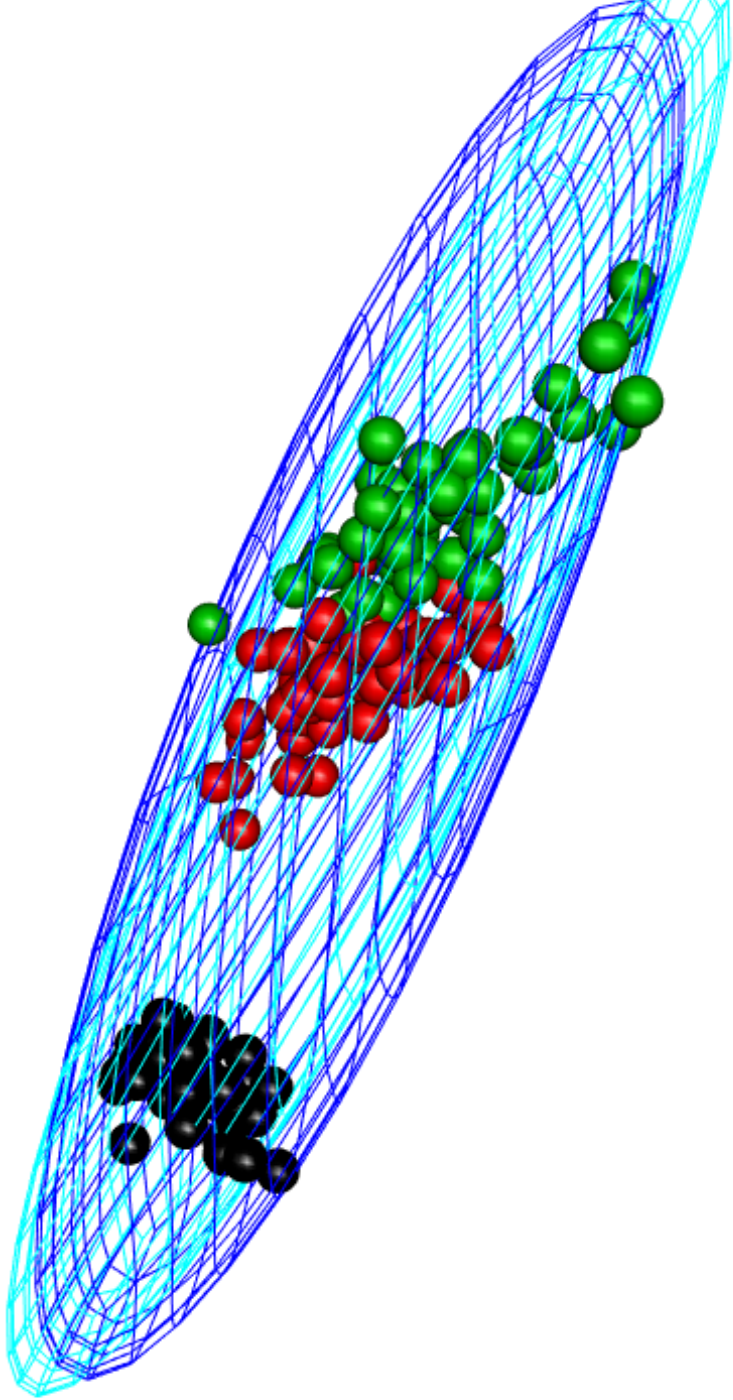


Fig. 11.11 in K. Murphy

EM example: Iris data

- The famous (Fisher's or Anderson's) iris data set gives the measurements in centimeters of the variables **sepal length and width** and **petal length and width**, respectively, for 50 flowers from each of **3 species of iris**.
- The species are **Iris setosa, versicolor, and virginica**.





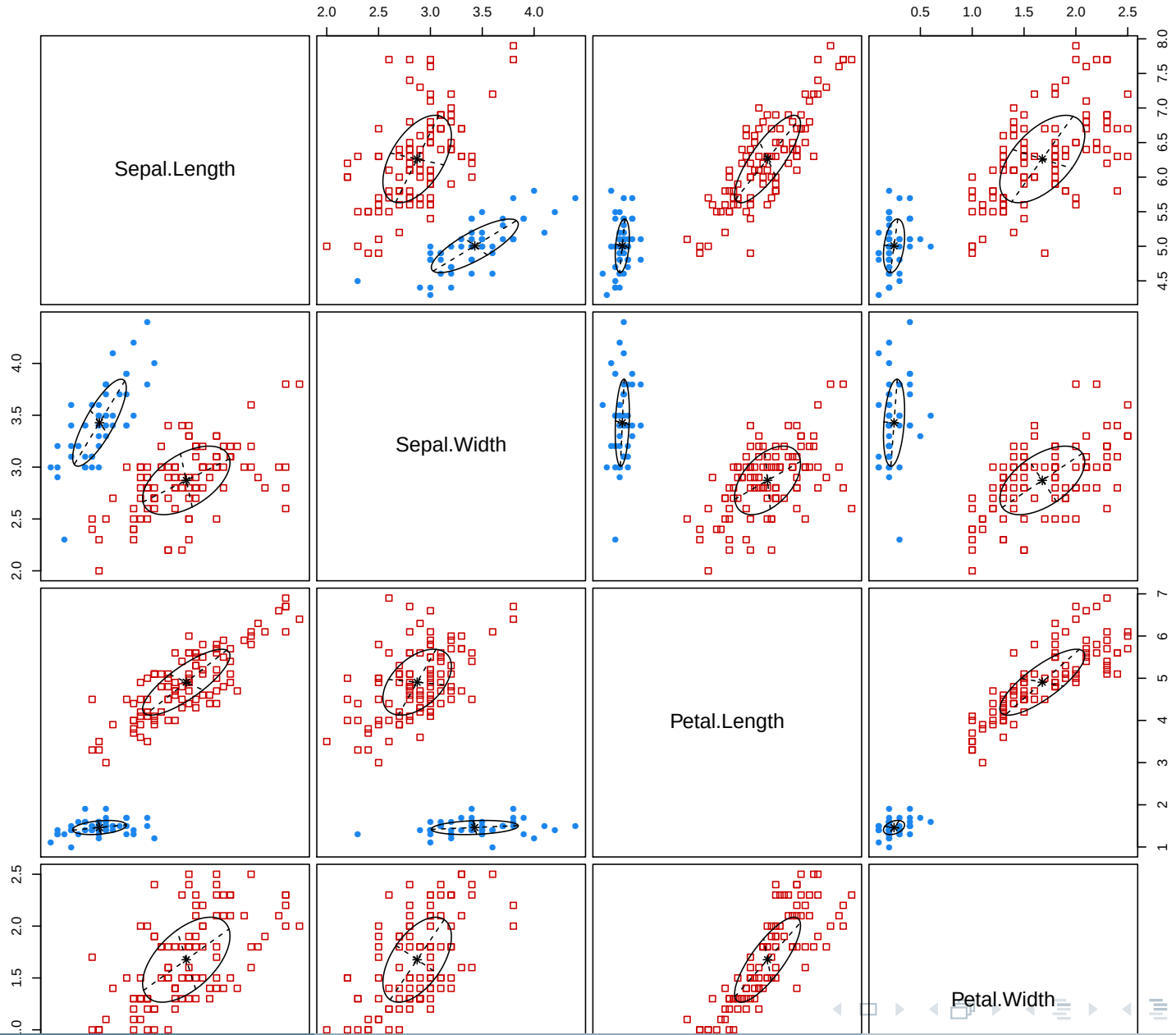
Bayesian model selection for mixture models

As a simple strategy for selecting the appropriate number of mixture components, we can find m that minimizes the overall description length (cf. BIC):

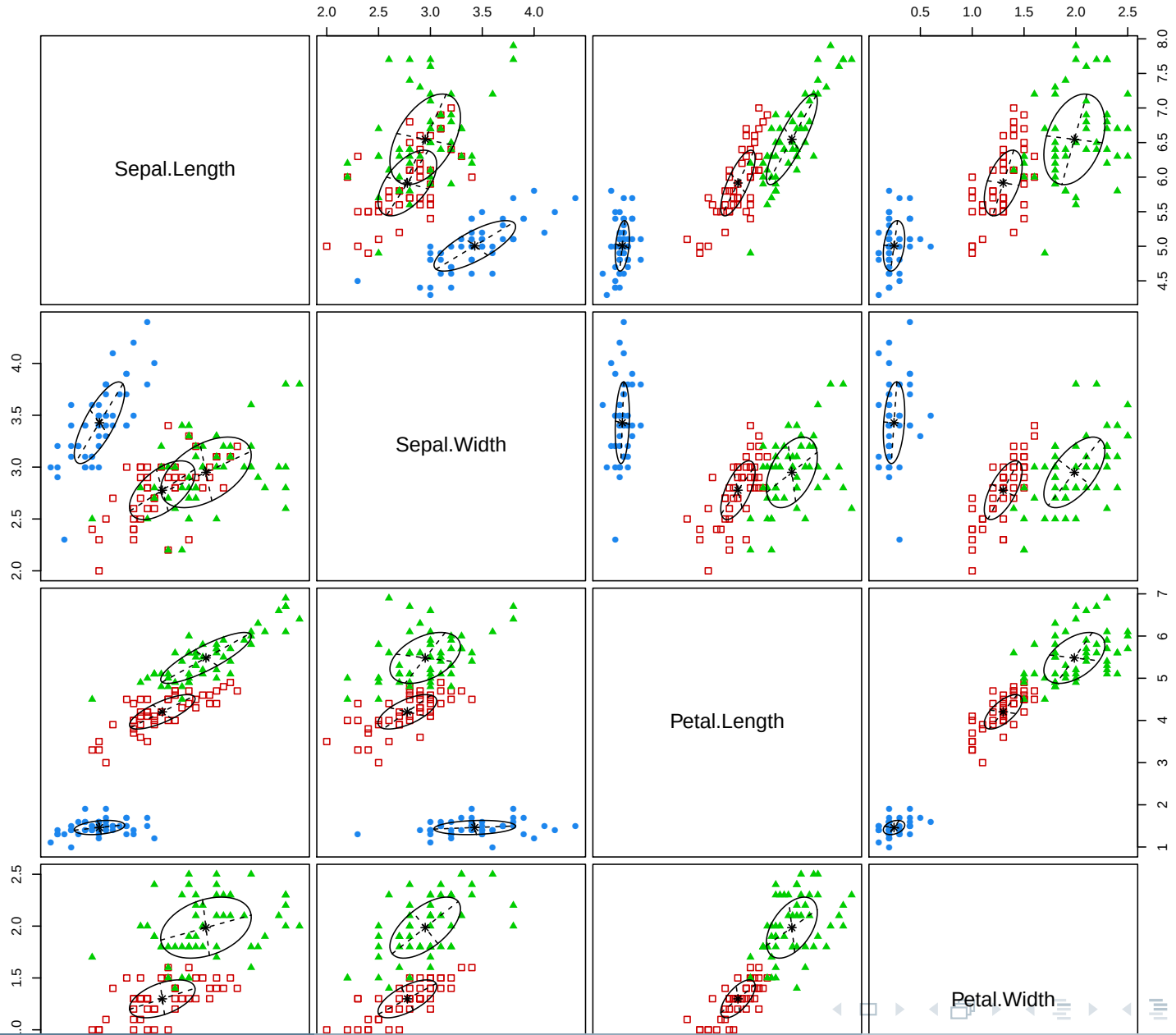
$$DL \approx -\log p(\text{data}|\hat{\boldsymbol{\theta}}_m) + \frac{d_m}{n}\log(n)$$

- n is the number of training points,
- $\hat{\boldsymbol{\theta}}_m$ are the maximum likelihood parameters for the m-component mixture, and
- d_m is the number of parameters in the m-component mixture.

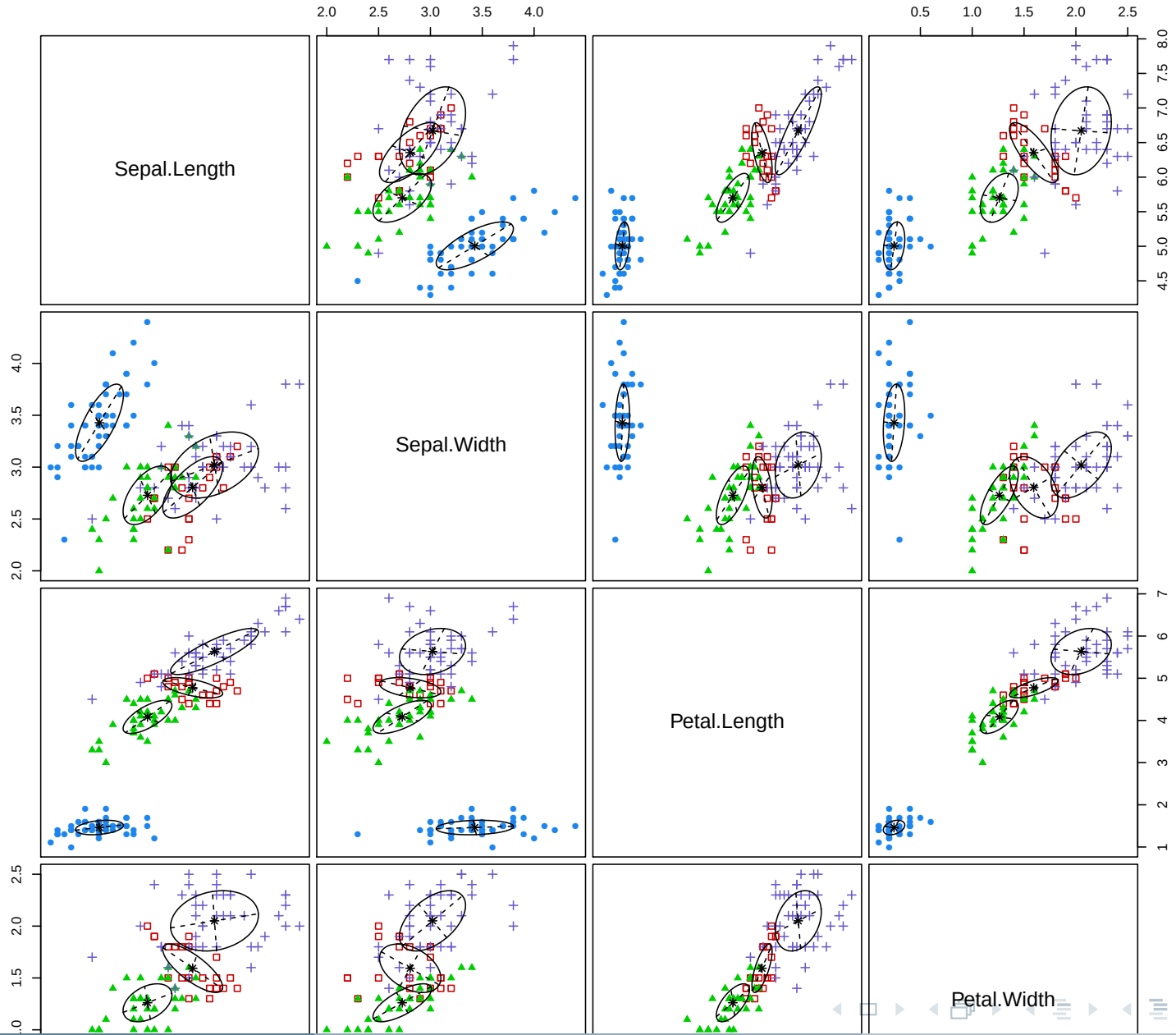
Model selection example: Iris data, m = 2



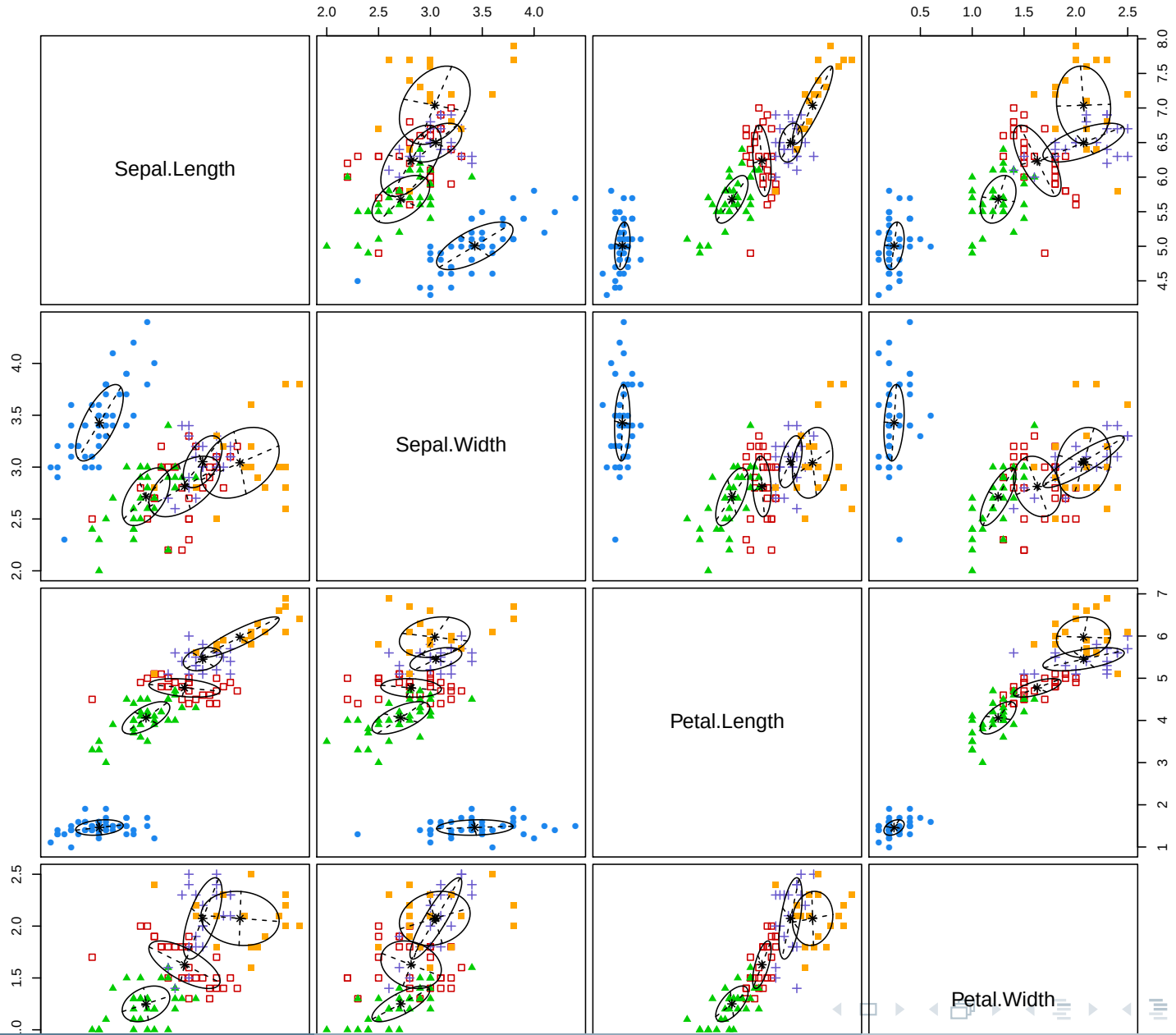
Model selection example: Iris data, m = 3



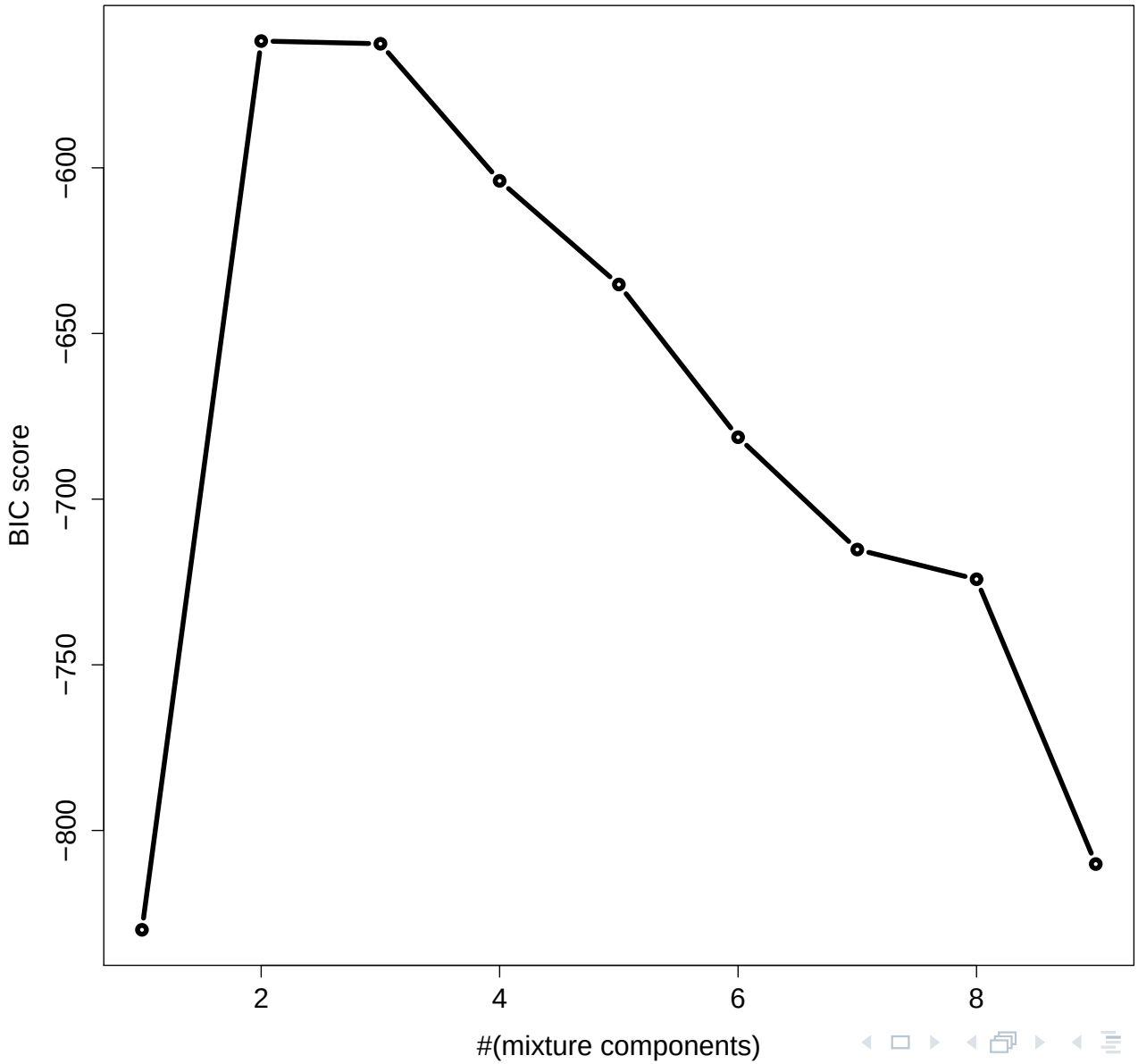
Model selection example: Iris data, m = 4



Model selection example: Iris data, m = 5



Model selection example: Iris data, BIC



The EM-algorithm: Convergence

Step 0: specify the initial setting of the parameters $\boldsymbol{\theta} = \boldsymbol{\theta}^0$.
E-step: complete the incomplete data (missing z) with the posterior probabilities ("soft labels")

$$P(z = j|\boldsymbol{x}_i, \boldsymbol{\theta}^t),\ j = 1, \ldots, m,\ i = 1, \ldots, n.$$

M-step: find the new setting of the parameters $\boldsymbol{\theta}^{t+1}$ by maximizing the log-likelihood of the inferred (or "expected complete") data

$$\boldsymbol{\theta}^{t+1} = \arg\max_{\boldsymbol{\theta}} \underbrace{\sum_{i=1}^{n} \sum_{j=1}^{m} P(z = j|\boldsymbol{x}_i, \boldsymbol{\theta}^t) \log[p_j p(\boldsymbol{x}_i|\boldsymbol{\theta}_j)]}_{\text{inferred (= expected complete) log-likelihood } Q(\boldsymbol{\theta}, \boldsymbol{\theta}^t)}.$$

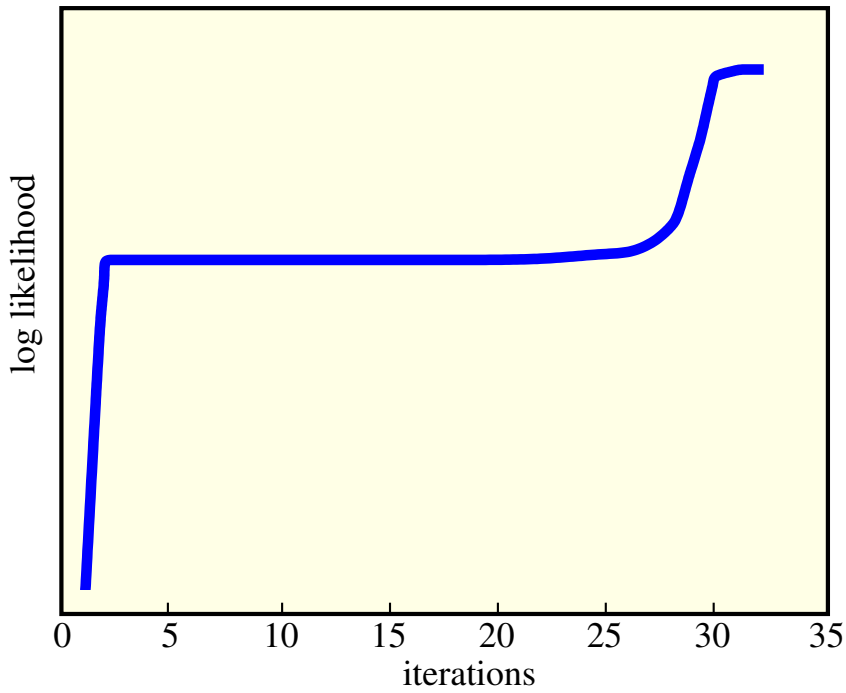
The expected complete log-likelihood $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^t)$ is called **auxiliary objective.**

The EM-algorithm: Convergence

The EM-algorithm **monotonically increases the log-likelihood** of the training data. In other words,

$$l(\boldsymbol{\theta}^0) < l(\boldsymbol{\theta}^1) < l(\boldsymbol{\theta}^2) < \ldots \text{ until convergence}$$

$$l(\boldsymbol{\theta}^t) = \sum_{i=1}^{n} \log p(\boldsymbol{x}_i|\boldsymbol{\theta}^t).$$



Jensen's inequality

- Convex function: secant line above graph of the function
  $\rightsquigarrow$ Jensen's inequality for two points.
- Secant line consists of weighted means of the convex function.
  For $a \in [0,1]$:

  $$af(x_1) + (1-a)f(x_2).$$

  Graph: convex function of the weighted means:

  $$f\left(ax_1 + (1-a)x_2\right).$$

- Thus, Jensen's inequality is

  $$f\left(ax_1 + (1-a)x_2\right) \leq af(x_1) + (1-a)f(x_2).$$

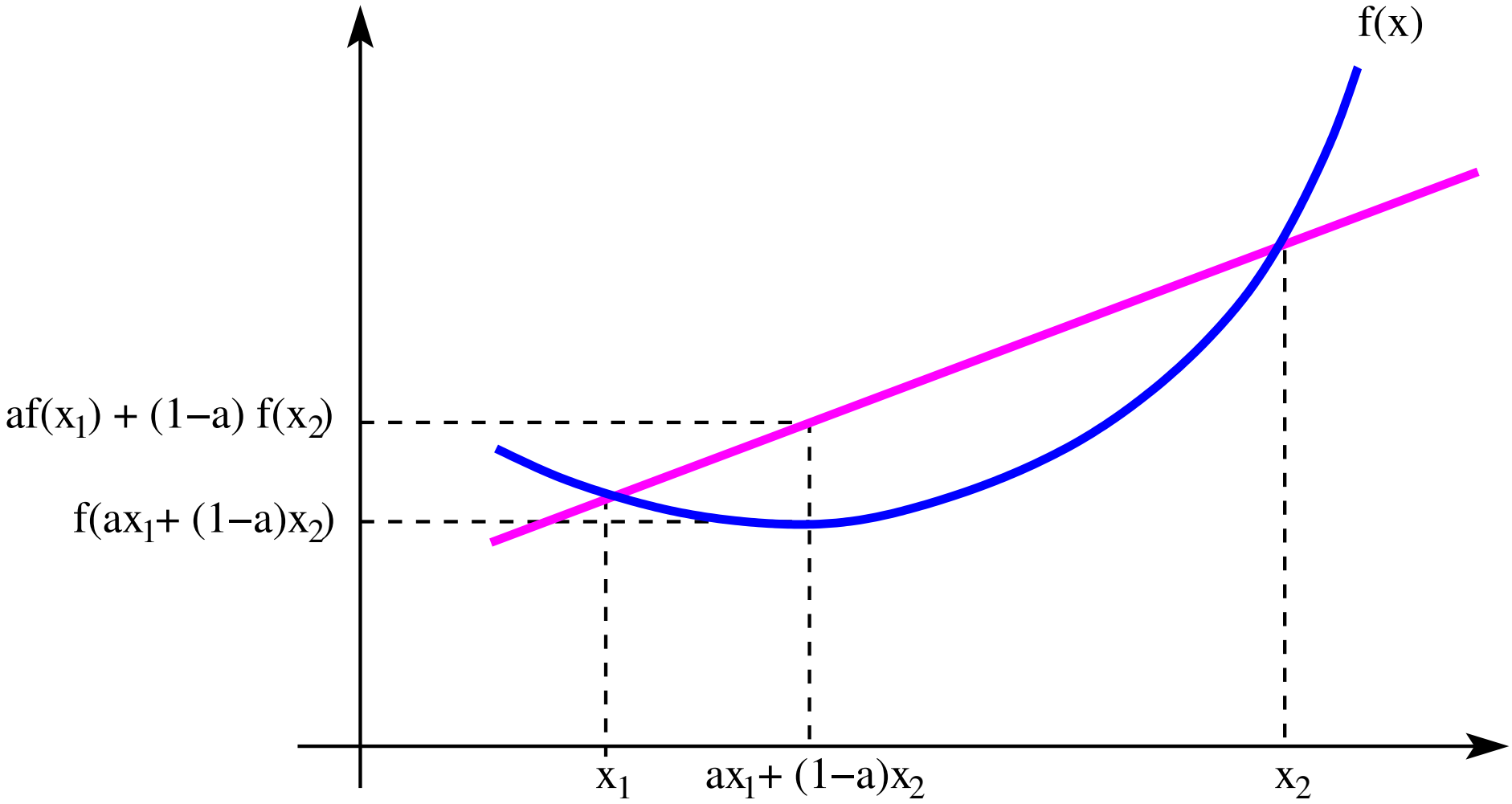
- Probability theory: if X is a RV and φ a convex function, then

  $$\varphi\left(\mathrm{E}[X]\right) \leq \mathrm{E}\left[\varphi(X)\right].$$

- φ convex $\rightsquigarrow$ $\psi := -\varphi$ concave:

  $$\psi\left(\mathrm{E}[X]\right) \geq \mathrm{E}\left[\psi(X)\right]. \quad \text{Example:} \quad \log\left(\mathrm{E}[X]\right) \geq \mathrm{E}\left[\log(X)\right].$$

Jensen's inequality



Non-negativity of KL divergence

$$
\begin{aligned}
-\mathbb{KL}(p(x)\|q(x)) &= \int p(x) \log\left(\frac{q(x)}{p(x)}\right)\, dx \\
\text{(Jensen's inequality)} \quad &\leq \log\left(\int p(x)\frac{q(x)}{p(x)}\, dx\right) \\
&= \log\left(\int q(x)\, dx\right) \\
&= \log(1) = 0
\end{aligned}
$$

This is also called **Gibbs' inequality.**

The EM-algorithm: Theoretical basis

Consider distribution $q(z_i)$ over latent assignment variables.
Log-likelihood:

$$
\begin{aligned}
l(\boldsymbol{\theta}) &= \sum_{i=1}^{n} \log p(\boldsymbol{x}_i|\boldsymbol{\theta}) \\
&= \sum_{i=1}^{n} \log \sum_{z_i} p(\boldsymbol{x}_i, z_i|\boldsymbol{\theta}) \\
&= \sum_{i=1}^{n} \log \sum_{z_i} q(z_i) \frac{p(\boldsymbol{x}_i, z_i|\boldsymbol{\theta})}{q(z_i)} \\
&= \sum_{i=1}^{n} \log \mathbb{E}_{q_i} \frac{p(\boldsymbol{x}_i, z_i|\boldsymbol{\theta})}{q(z_i)} \\
\text{(Jensen's inequality)} \quad &\geq \sum_{i=1}^{n} \sum_{z_i} q(z_i) \log \frac{p(\boldsymbol{x}_i, z_i|\boldsymbol{\theta})}{q(z_i)} \\
&=: Q(\boldsymbol{\theta}, q).
\end{aligned}
$$

- Lower bound

$$l(\boldsymbol{\theta}) \geq Q(\boldsymbol{\theta}, q) := \sum_{i=1}^{n} \sum_{z_i} q(z_i) \log \frac{p(\boldsymbol{x}_i, z_i|\boldsymbol{\theta})}{q(z_i)}$$

valid for any positive distribution q. Which one should we choose?

- Intuition: pick the q that yields the tightest lower bound.
This will be the **E-step.**
- At time t, assume we have chosen q^t based on current parameters $\boldsymbol{\theta}^t$.
In the next **M-step** we maximize the
expected complete log-likelihood:

$$\boldsymbol{\theta}^{t+1} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^t) = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \mathbb{E}_{q_i^t} \log p(\boldsymbol{x}_i, z_i|\boldsymbol{\theta})$$

- Last equation follows from

$$Q(\boldsymbol{\theta}, q) = \underbrace{\sum_{i=1}^{n} \mathbb{E}_{q_i} \log p(\boldsymbol{x}_i, z_i|\boldsymbol{\theta})}_{\text{Expected complete log-l}} + \sum_{i=1}^{n} [\underbrace{- \sum_{z_i} q(z_i) \log q(z_i)}_{h(q_i),\ \text{independent of } \boldsymbol{\theta}}].$$

The E-step

Re-write lower bound as

$$Q(\boldsymbol{\theta}, q) = \sum_i L(\boldsymbol{\theta}, q_i),$$

with

$$\begin{aligned} L(\boldsymbol{\theta}, q_i) &= \sum_{z_i} q(z_i) \log \frac{p(\boldsymbol{x}_i, z_i|\boldsymbol{\theta})}{q(z_i)} \\ &= \sum_{z_i} q(z_i) \log \frac{p(z_i|\boldsymbol{x}_i, \boldsymbol{\theta}) p(\boldsymbol{x}_i|\boldsymbol{\theta})}{q(z_i)} \\ &= \sum_{z_i} q(z_i) \log \frac{p(z_i|\boldsymbol{x}_i, \boldsymbol{\theta})}{q(z_i)} + \sum_{z_i} q(z_i) \log p(\boldsymbol{x}_i|\boldsymbol{\theta}) \\ &= -\underbrace{\mathbb{KL}(q(z_i) \| p(z_i|\boldsymbol{x}_i, \boldsymbol{\theta}))}_{\text{always } \geq 0, \text{ and } = 0, \text{ if } q = p} + \underbrace{\log p(\boldsymbol{x}_i|\boldsymbol{\theta})}_{\text{independent of } q_i} . \end{aligned}$$

The E step

For $q_i^t(z_i) = p(z_i|\boldsymbol{x}_i, \boldsymbol{\theta}^t)$, the KL divergence is zero, and $L(\boldsymbol{\theta}^t, q_i)$ is maximized over all possible distributions q_i:

$$q_i^t(z_i) = p(z_i|\boldsymbol{x}_i, \boldsymbol{\theta}^t) = \arg\max_{q_i} L(\boldsymbol{\theta}^t, q_i) \quad (\rightsquigarrow \text{E-step})$$

$$L(\boldsymbol{\theta}^t, q_i^t) = \log p(\boldsymbol{x}_i|\boldsymbol{\theta}^t)$$

$$Q(\boldsymbol{\theta}^t, \boldsymbol{\theta}^t) = \sum_i \log p(\boldsymbol{x}_i|\boldsymbol{\theta}^t) = l(\boldsymbol{\theta}^t)$$

$\rightsquigarrow$ lower bound "touches" the log-likelihood

$\rightsquigarrow$ after the E-step, the auxiliary objective equals the log-likelihood

$\rightsquigarrow$ lower bound is tight after the E-step.

The E step

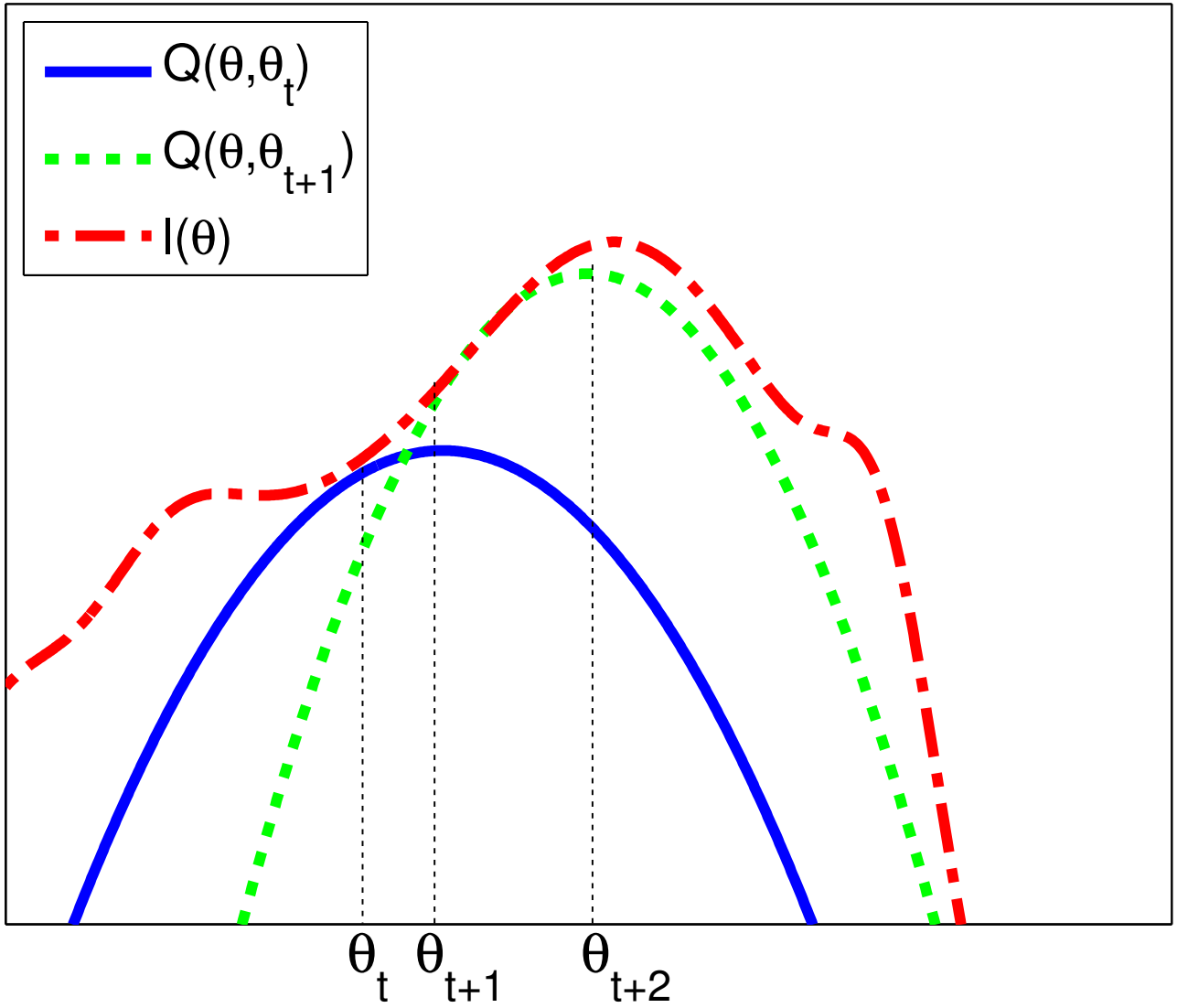


Fig 11.16 in K. Murphy

EM-algorithm: max-max and monotonicity

We can now rewrite the EM-algorithm in terms of two maximization steps involving the auxiliary objective:
E-step: $\boldsymbol{q}^t = \arg\max_{\boldsymbol{q}} Q(\boldsymbol{\theta}^t, \boldsymbol{q})$
M-step: $\boldsymbol{\theta}^{t+1} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^t)$.
The monotonic increase of the log-likelihood now follows from

$$\underbrace{l(\boldsymbol{\theta}^{t+1}) \geq Q(\boldsymbol{\theta}^{t+1}, \boldsymbol{\theta}^t)}_{Q(\boldsymbol{\theta}^{t+1},\bullet) \text{ is lower bound on } l(\boldsymbol{\theta}^{t+1})} \geq Q(\boldsymbol{\theta}^t, \boldsymbol{\theta}^t) = l(\boldsymbol{\theta}^t).$$

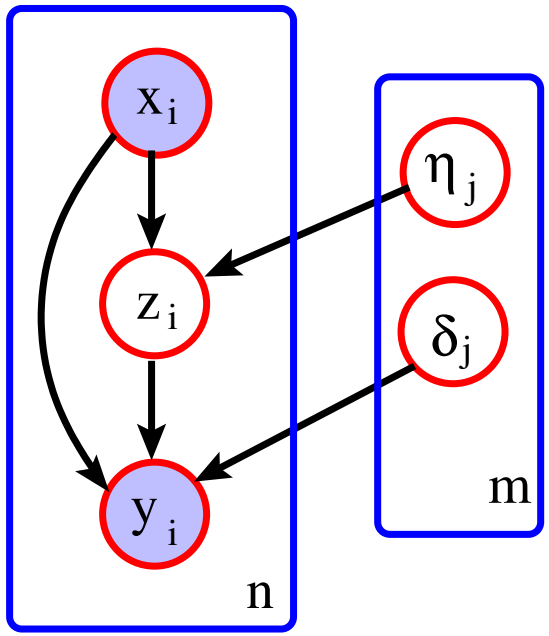
Second inequality: $Q(\boldsymbol{\theta}^{t+1}, \boldsymbol{\theta}^t) = \max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^t) \geq Q(\boldsymbol{\theta}^t, \boldsymbol{\theta}^t)$.

Conditional mixtures

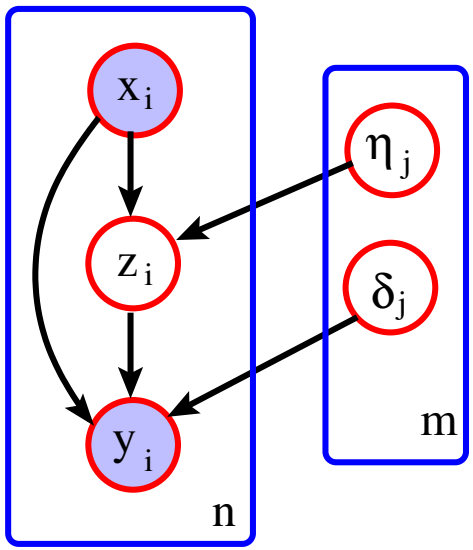
- Some regression or classification problems can be decomposed into easier sub-problems.
- Examples:
  - style in handwritten character recognition
  - dialect/accent in speech recognition, etc.
- Each sub-problem could be solved by a specific "expert".
- The selection of which expert to rely on now depends on the position $\boldsymbol{x}$ in the input space. **Mixtures of experts models.**

Experts (regression)

- Suppose we have several regression "experts" generating conditional Gaussian outputs
$$p(y|\boldsymbol{x}, z = j, \boldsymbol{\theta}) = \mathcal{N}(y|\beta_j^t \boldsymbol{x}, \sigma_j^2)$$
- $\boldsymbol{\delta}_j = \{\beta_j, \sigma_j^2\}$: Parameters of j-th expert.
- Need to find a way of allocating tasks to these experts.
- Parameter vector $\boldsymbol{\theta}$ contains the means and variances of the m experts and the additional parameters $\boldsymbol{\eta}$ of this allocation mechanism: $\boldsymbol{\theta} = \{\boldsymbol{\delta}_j, \boldsymbol{\eta}_j\}_{j=1}^m$.



Joint distribution



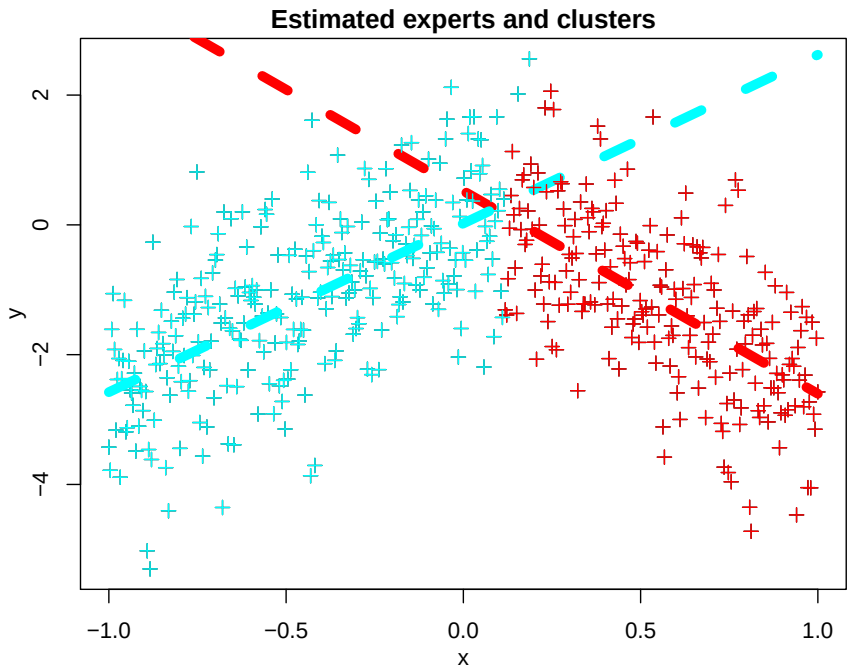
From the DAG we conclude:

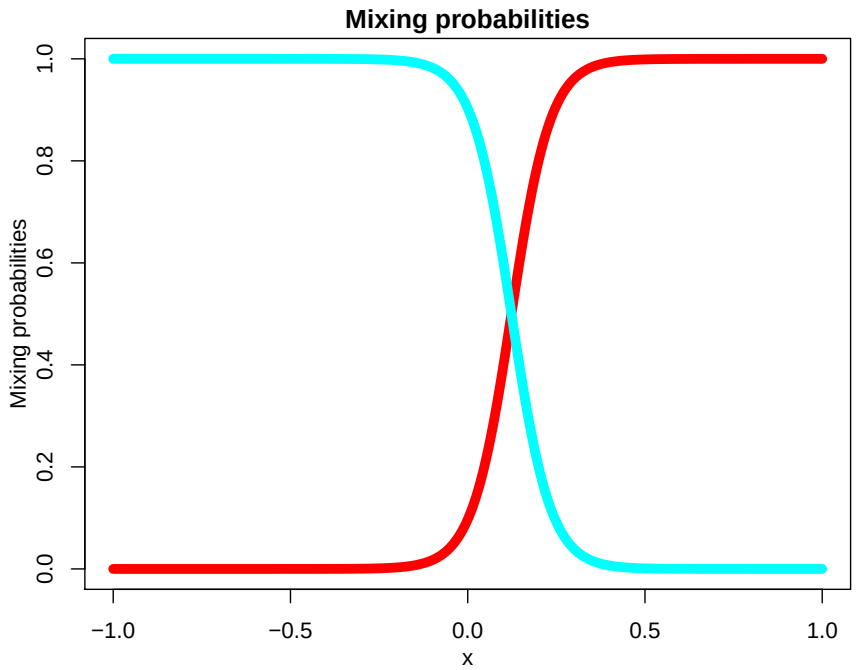
$$\begin{aligned} p(y, z = j|\boldsymbol{x}) &= p(y|\boldsymbol{\delta}, z = j, \boldsymbol{x}) P(z = j|\boldsymbol{\eta}, \boldsymbol{x}) \\ &= p(y|\boldsymbol{\delta}_j, \boldsymbol{x}) P(z = j|\boldsymbol{\eta}, \boldsymbol{x}) \\ &= \mathcal{N}(y|\boldsymbol{\beta}_j^t \boldsymbol{x}, \sigma_j^2) P(z = j|\boldsymbol{\eta}, \boldsymbol{x}) \end{aligned}$$

Thus, the overall prediction is

$$\begin{aligned} p(y|\boldsymbol{x}, \boldsymbol{\theta}) &= \sum_j p(y, z = j|\boldsymbol{x}, \boldsymbol{\eta}, \boldsymbol{\delta}) \\ &= \sum_j P(z = j|\boldsymbol{x}, \boldsymbol{\eta}) p(y|\boldsymbol{x}, \boldsymbol{\delta}_j) \\ &= \sum_j P(z = j|\boldsymbol{x}, \boldsymbol{\eta}) p(y|\boldsymbol{x}, \boldsymbol{\beta}_j, \sigma_j^2). \end{aligned}$$

Mixtures of experts





Here we need to switch from one linear regression model to another: $p(y|\boldsymbol{x}, z = j, \boldsymbol{\theta}) = \mathcal{N}(y|\beta_j^t \boldsymbol{x}, \sigma_j^2)$. The switch can be probabilistic $\rightsquigarrow$ probabilistic gating function $P(z|\boldsymbol{x}, \boldsymbol{\eta})$ (right).

Gating network

- A gating network specifies a distribution over m experts, conditionally on the input $\boldsymbol{x}$.
- Example: when there are just two experts the gating network can be a logistic regression model

$$P(z = 1|\boldsymbol{x}, \boldsymbol{\eta}) = \sigma(\boldsymbol{\eta}^t \boldsymbol{x}),$$

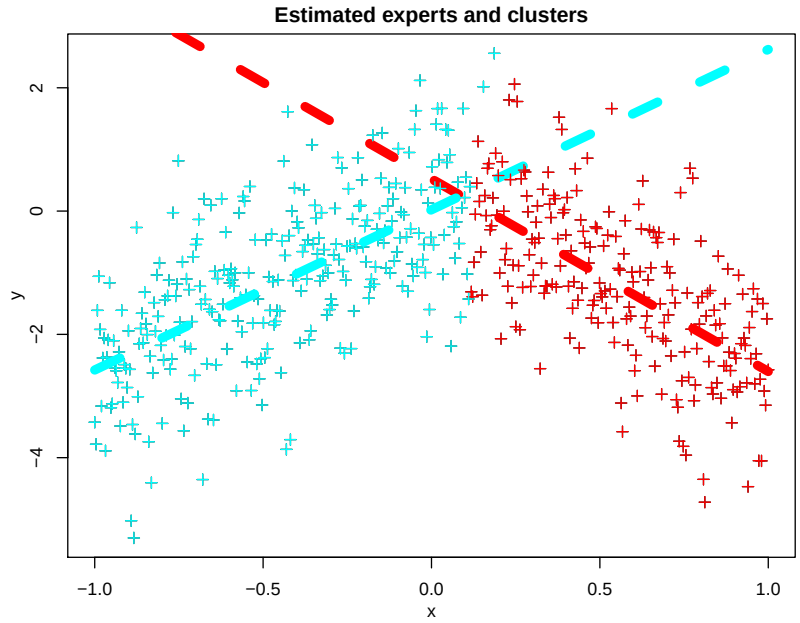
where $\sigma(z) = (1 + e^{-z})^{-1}$ is the logistic function.

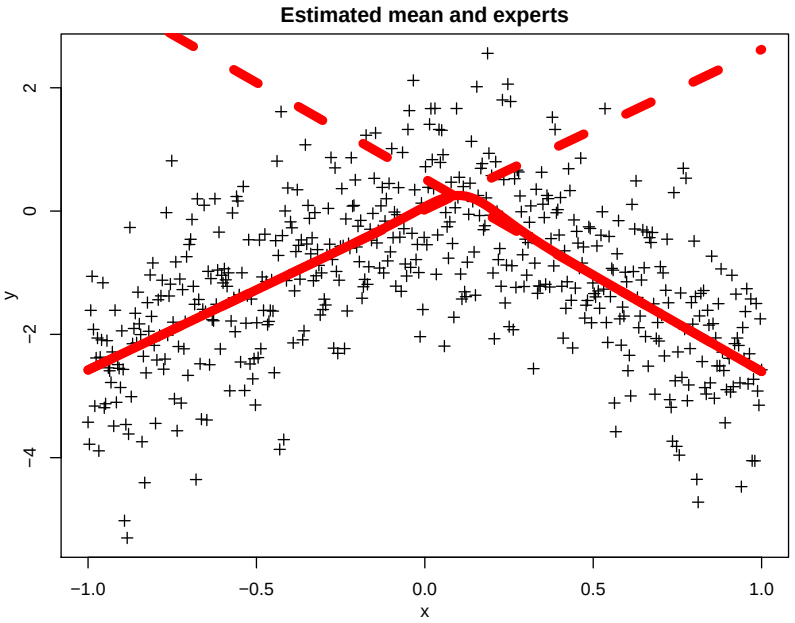
- For $m > 2$, the gating network can be a softmax model

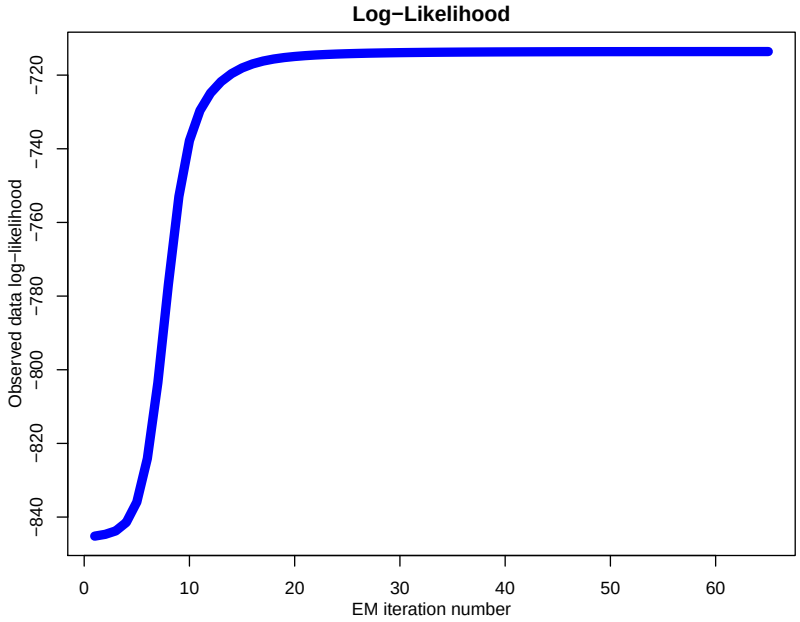
$$P(z = j|\boldsymbol{x}, \boldsymbol{\eta}) = \frac{\exp(\boldsymbol{\eta}_j^t \boldsymbol{x})}{\sum_{j'=1}^{m} \exp(\boldsymbol{\eta}_{j'}^t \boldsymbol{x})},$$

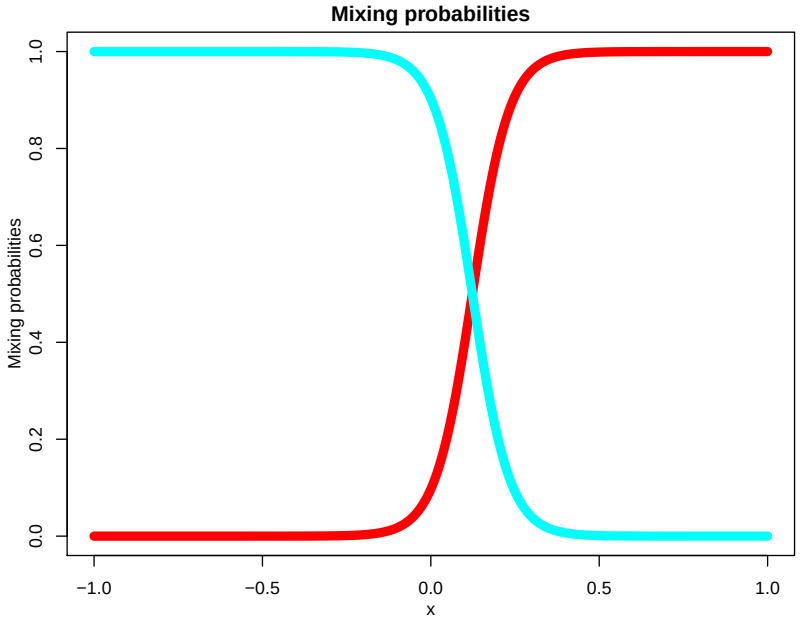
where $\boldsymbol{\eta} = \{\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_m\}$ are the parameters of the gating network.

- Overall prediction
$p(y|\boldsymbol{x}, \boldsymbol{\theta}) = \sum_j p(y, z = j|\boldsymbol{x}, \boldsymbol{\eta}, \boldsymbol{\delta}) = \sum_j P(z = j|\boldsymbol{x}, \boldsymbol{\eta}) p(y|\boldsymbol{x}, \boldsymbol{\delta}_j)$.


Estimated experts and clusters
y
x
−1.0
−0.5
0.0
0.5
1.0
2
0
−2
−4


Estimated mean and experts
y
x
−1.0
−0.5
0.0
0.5
1.0
2
0
−2
−4

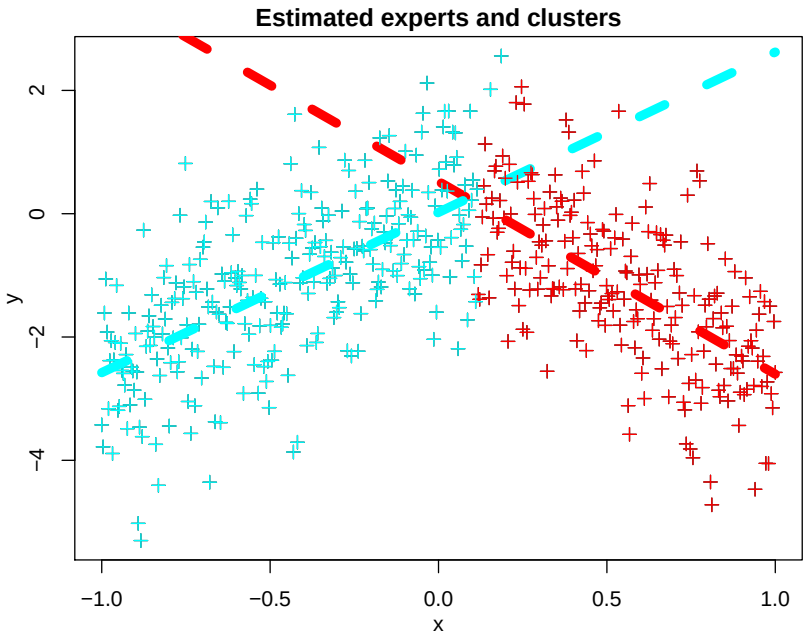

Log−Likelihood
Observed data log−likelihood
EM iteration number
0
10
20
30
40
50
60
−720
−740
−760
−780
−800
−820
−840

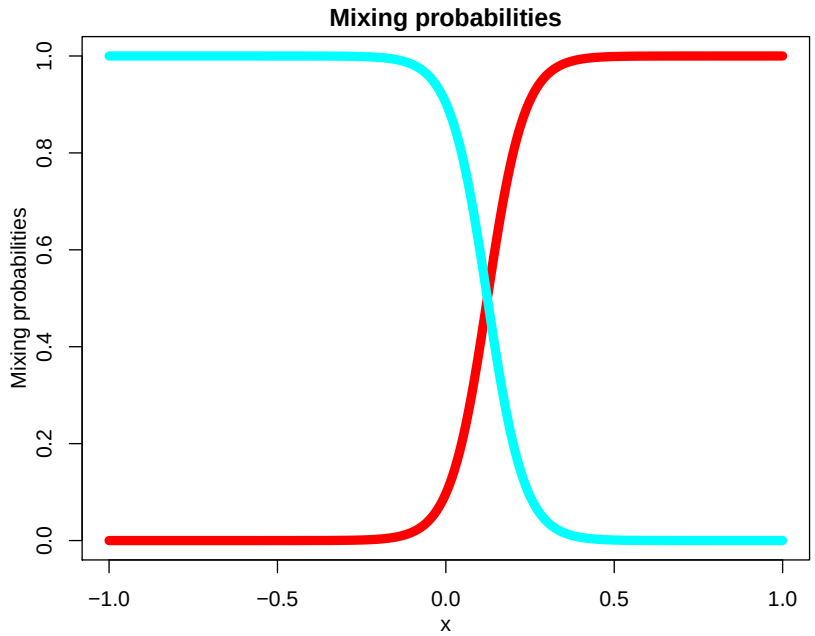

Mixing probabilities
Mixing probabilities
x
−1.0
−0.5
0.0
0.5
1.0
0.0
0.2
0.4
0.6
0.8
1.0

A mixture of experts model: estimation

"Soft labels": Conditional probability that $(\boldsymbol{x}_i, y_i)$ came from expert j:

$$\begin{aligned}\hat{P}(j|i) &= P(z = j|\boldsymbol{x}_i, y_i, \boldsymbol{\theta}) \\ &= \frac{P(z = j|\boldsymbol{x}_i, \boldsymbol{\eta}^t) p(y_i|\boldsymbol{x}_i, (\boldsymbol{\beta}_j, \sigma_j^2))}{\sum_{j'=1}^{m} P(z = j'|\boldsymbol{x}_i, \boldsymbol{\eta}^t) p(y_i|\boldsymbol{x}_i, (\boldsymbol{\beta}_{j'}, \sigma_{j'}^2))}\end{aligned}$$





EM for mixtures of experts

E-step: compute soft labels $\hat{P}(j|i)$
M-step: separately re-estimate the experts and the gating network based on these soft assignments:

1. For each expert j: find $(\hat{\beta}_j, \hat{\sigma}_j^2)$ that maximize

$$\sum_{i=1}^{n} \hat{P}(j|i) \log p(y_i|\boldsymbol{x}_i, (\beta_j, \sigma_j^2))$$

   $\rightsquigarrow$ linear regression with weighted training set.

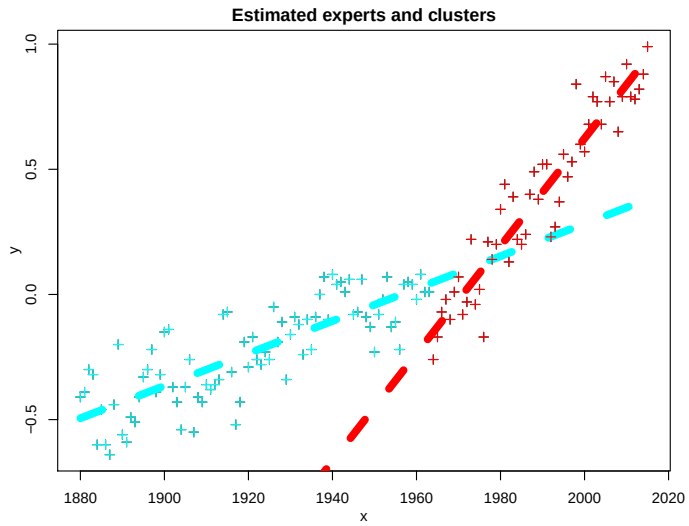
2. For the gating network: find $\hat{\boldsymbol{\eta}}$ that maximize

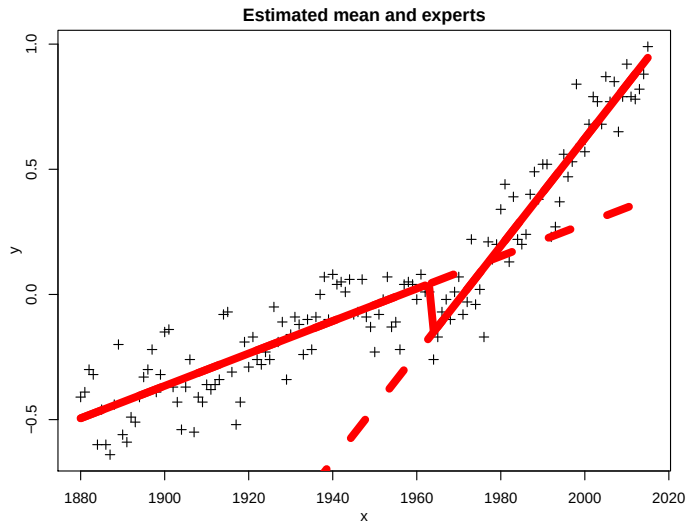
$$\sum_{i=1}^{n} \sum_{j=1}^{m} \hat{P}(j|i) \log P(j|\boldsymbol{x}_i, \boldsymbol{\eta}_j)$$

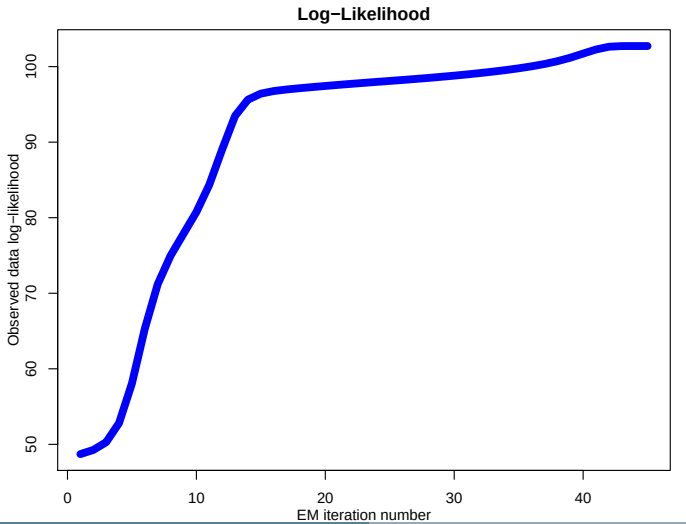
   $\rightsquigarrow$ logistic regression with weighted training set.

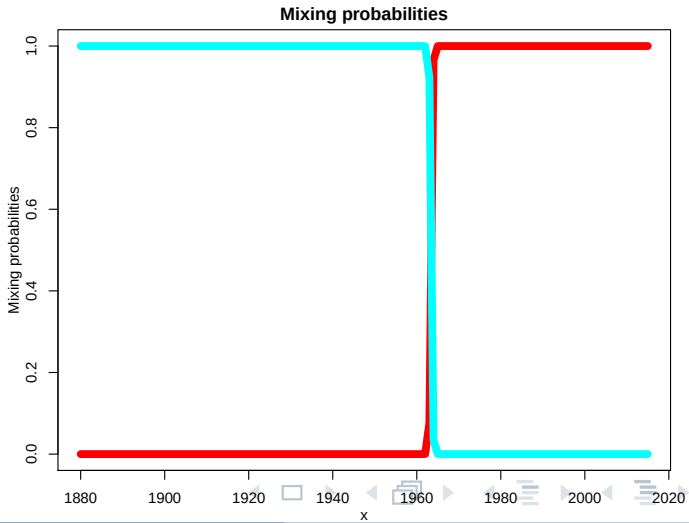
Real World Example

Global annual temperature anomalies (degrees C) computed using data from land meteorological stations, 1880-2015. Anomalies are relative to the 1951-1980 base period means.





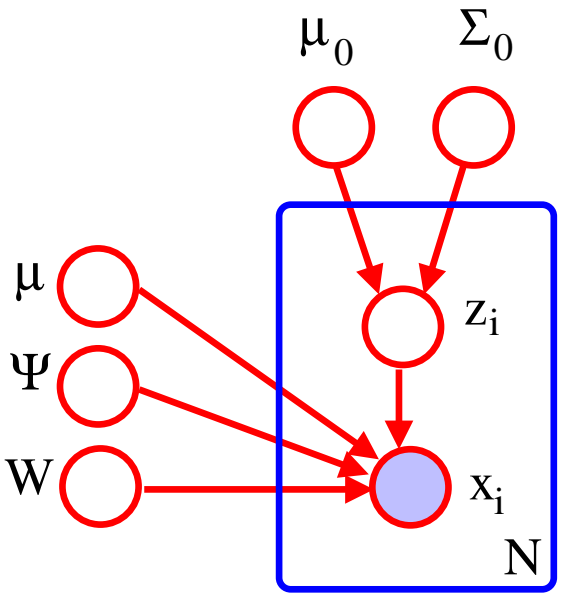




Section 10

Linear latent variable models

Factor analysis



- One problem with mixture models: **only a single latent variable**. Each observation can only come from one of K prototypes.
- Alternative: $\boldsymbol{z}_i \in \mathbb{R}^k$. Gaussian prior:

$$p(\boldsymbol{z}_i) = \mathcal{N}(\boldsymbol{z}_i | \boldsymbol{\mu}_0, \Sigma_0)$$

- For observations $\boldsymbol{x}_i \in \mathbb{R}^p$, we may use a **Gaussian likelihood.**
- As in linear regression, we assume the mean is a **linear** function:

$$p(\boldsymbol{x}_i | \boldsymbol{z}_i, \boldsymbol{\theta}) = \mathcal{N}(W\boldsymbol{z}_i + \boldsymbol{\mu}, \Psi),$$

  W: **factor loading matrix**, and Ψ: **covariance matrix**.
- We take Ψ to be **diagonal**, since the whole point of the model is to "force" z_i to **explain the correlation.**

Factor analysis: generative process

Generative process ($k = 1$, $p = 2$, diagonal Ψ):

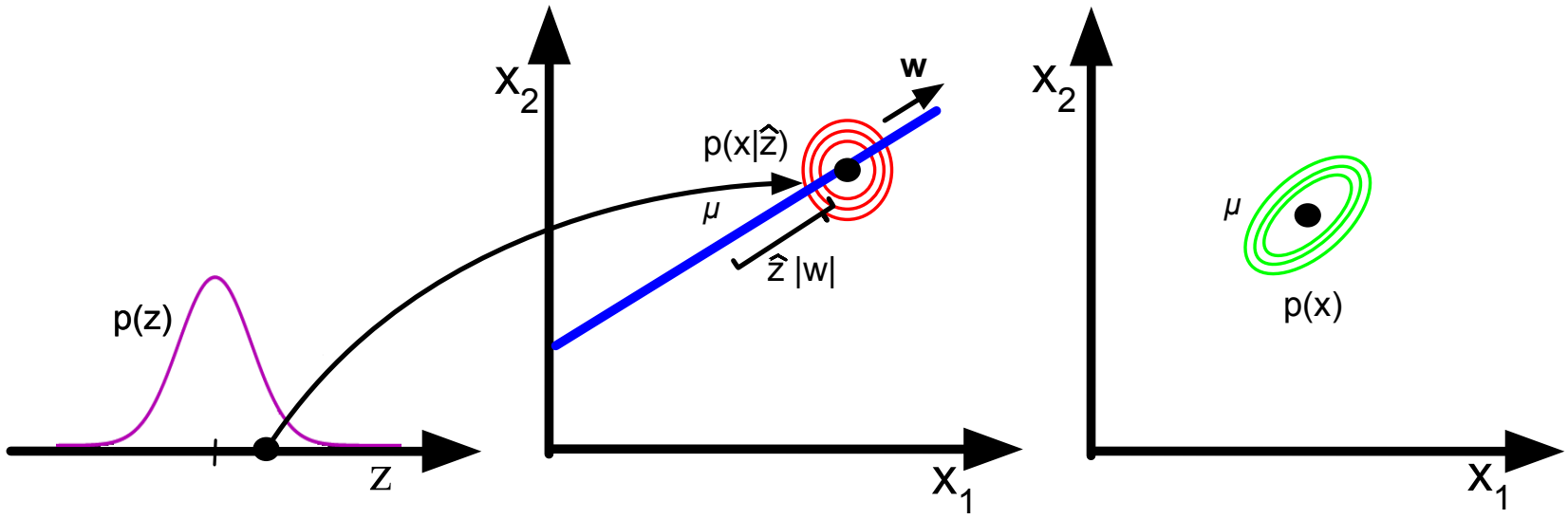


Figure 12.1 in K. Murphy

We take an isotropic Gaussian "spray can" and slide it along the 1d line defined by $wz_i + \mu$. This induces a correlated Gaussian in 2d.

Inference of the latent factors

- We hope that the latent factors z will reveal something interesting about the data $\rightsquigarrow$ compute posterior over the latent variables:

$$\begin{aligned} p(\boldsymbol{z}_i|\boldsymbol{x}_i, \boldsymbol{\theta}) &= \mathcal{N}(\boldsymbol{z}_i|\boldsymbol{m}_i, \Sigma) \\ \Sigma &= (\Sigma_0^{-1} + W^t \Psi^{-1} W)^{-1} \\ \boldsymbol{m}_i &= \Sigma_i (W^t \Psi^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}) + \Sigma_0^{-1} \boldsymbol{\mu}_0) \end{aligned}$$

- The posterior means $\boldsymbol{m}_i$ are called the **latent scores, or latent factors.**

Example

- Example from (Shalizi 2009). $p = 11$ variables and $n = 387$ cases describing aspects of cars: engine size, #(cylinders), miles per gallon (MPG), price, etc.
- Fit a $p = 2$ dim model. Plot m_i scores as points in $\mathbb{R}^2$.
- To get a better understanding of the "meaning" of the latent factors, project unit vectors $e_1 = (1, 0, \ldots, 0), e_2 = (0, 1, 0, \ldots, 0)$, etc. into the low dimensional space (blue lines)
- Horizontal axis represents price, corresponding to the features labeled "dealer" and "retail", with expensive cars on the right. Vertical axis represents fuel efficiency (measured in terms of MPG) versus size: heavy vehicles are less efficient and are higher up, whereas light vehicles are more efficient and are lower down.
- Verify by finding the closest exemplars in the training set.

Example

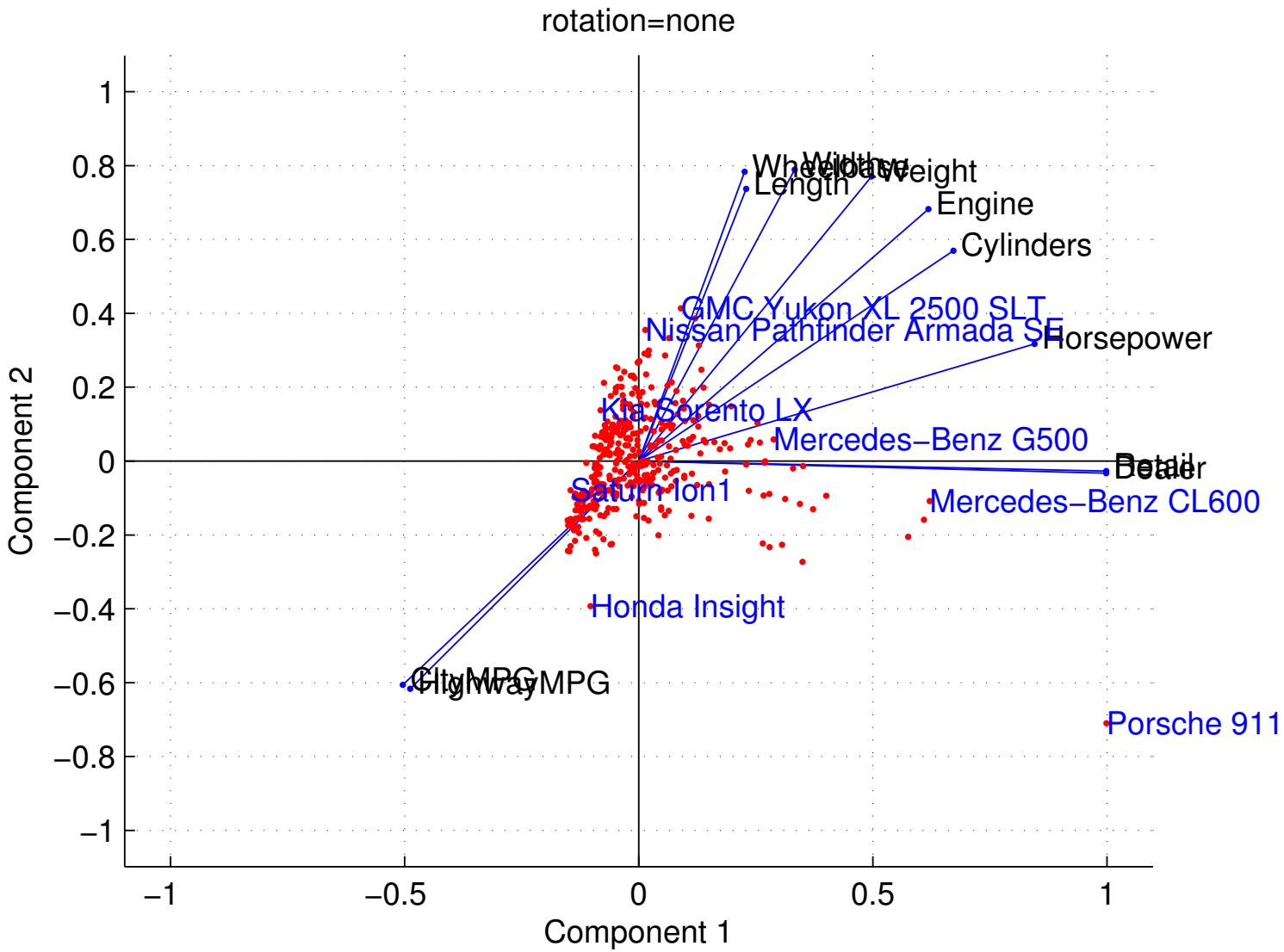
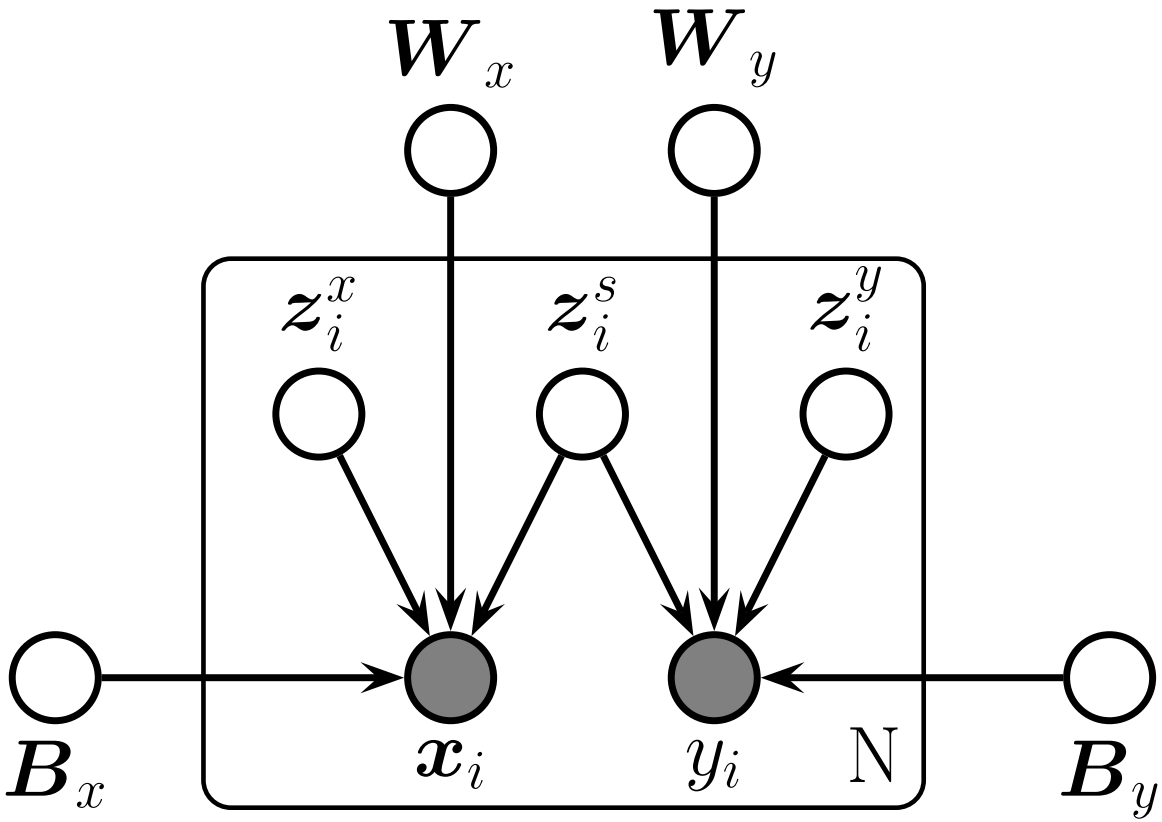


Figure 12.2 in K. Murphy

Special Cases: PCA and CCA

- Covariance matrix $\Psi = \sigma^2 I \rightsquigarrow$ (probabilistic) **PCA**.
- Two-view version involving $\boldsymbol{x}$ and $\boldsymbol{y} \rightsquigarrow$ **CCA**.



From figure 12.19 in K. Murphy

PCA and dimensionality reduction

Given n data points in p dimensions:

$$X = \begin{bmatrix} - & \boldsymbol{x}_1 & - \\ - & \boldsymbol{x}_2 & - \\ & \vdots & \\ - & \boldsymbol{x}_n & - \end{bmatrix} \in \mathbb{R}^{n \times p}$$

Want to reduce dimensionality from p to k. Choose k directions $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k$, arrange them as columns in matrix W:

$$W = \begin{bmatrix} \boldsymbol{w}_1 & \boldsymbol{w}_2 & \ldots & \boldsymbol{w}_k \end{bmatrix} \in \mathbb{R}^{p \times k}$$

For each $\boldsymbol{w}_j$, compute **similarity** $z_j = \boldsymbol{w}_j^t \boldsymbol{x}$, $j = 1 \ldots k$.
Project $\boldsymbol{x}$ down to $\boldsymbol{z} = (z_1, \ldots, z_k)^t = W^t \boldsymbol{x}$. How to choose W?

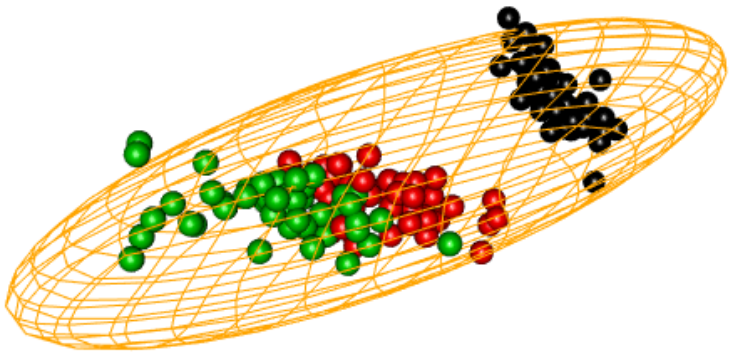
Encoding–decoding model

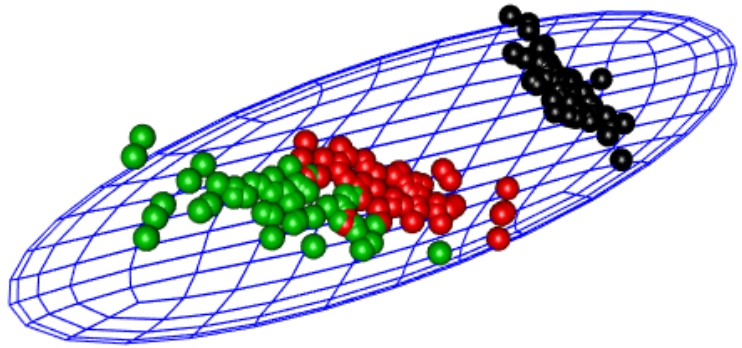
The projection matrix W serves two functions:

- **Encode:** $\boldsymbol{z} = W^t \boldsymbol{x}, \quad \boldsymbol{z} \in \mathbb{R}^k, \quad z_j = \boldsymbol{w}_j^t \boldsymbol{x}.$
  - The vectors $\boldsymbol{w}_j$ form a basis of the projected space.
  - We will require that this basis is orthonormal, i.e. $W^t W = I$.
- **Decode:** $\tilde{\boldsymbol{x}} = W\boldsymbol{z} = \sum_{j=1}^{k} z_j \boldsymbol{w}_j, \quad \tilde{\boldsymbol{x}} \in \mathbb{R}^p.$
  - If $k = p$, the above orthonormality condition implies $W^t = W^{-1}$, and encoding can be undone without loss of information.
  - If $k < p$, we use the pseudo-inverse
    $\rightsquigarrow$ the reconstruction error will be nonzero.
- Above we assumed that the origin of the coordinate system is in the sample mean, i.e. $\sum_i \boldsymbol{x}_i = 0$.

Principal Component Analysis (PCA)

In the general case, we want the reconstruction error $\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|$ to be small.
Objective: minimize $\min_{W \in \mathbb{R}^{p \times k}:\, W^t W = I} \sum_{i=1}^{n} \|\boldsymbol{x}_i - WW^t \boldsymbol{x}_i\|^2$





Finding the principal components

Projection vectors are orthogonal $\rightsquigarrow$ can treat them separately:

$$\min_{\boldsymbol{w}:\,\|\boldsymbol{w}\|=1} \sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{w}\boldsymbol{w}^t\boldsymbol{x}_i\|^2$$

$$\begin{aligned}
\sum_i \|\boldsymbol{x}_i - \boldsymbol{w}\boldsymbol{w}^t\boldsymbol{x}_i\|^2 &= \sum_{i=1}^{n} [\boldsymbol{x}_i^t\boldsymbol{x}_i - 2\boldsymbol{x}_i^t\boldsymbol{w}\boldsymbol{w}^t\boldsymbol{x}_i + \boldsymbol{x}_i^t\boldsymbol{w}\underbrace{\boldsymbol{w}^t\boldsymbol{w}}_{=1}\boldsymbol{w}^t\boldsymbol{x}_i] \\
&= \sum_i [\boldsymbol{x}_i^t\boldsymbol{x}_i - \boldsymbol{x}_i^t\boldsymbol{w}\boldsymbol{w}^t\boldsymbol{x}_i] \\
&= \sum_i \boldsymbol{x}_i^t\boldsymbol{x}_i - \boldsymbol{w}^t \sum_{i=1}^{n} \boldsymbol{x}_i\boldsymbol{x}_i^t\boldsymbol{w} \\
&= \underbrace{\sum_i \boldsymbol{x}_i^t\boldsymbol{x}_i}_{\text{const.}} - \boldsymbol{w}^t X^t X \boldsymbol{w}.
\end{aligned}$$

Finding the principal components

- Want to maximize $\boldsymbol{w}^t X^t X \boldsymbol{w}$ under the constraint $\|\boldsymbol{w}\| = 1$
- Can also maximize the ratio $J(\boldsymbol{w}) = \frac{\boldsymbol{w}^t X^t X \boldsymbol{w}}{\boldsymbol{w}^t \boldsymbol{w}}$.
- Optimal projection u is the eigenvector of $X^t X$ with largest eigenvalue (compare handout on spectral matrix norm).
- Note that we assumed that $\sum_i \boldsymbol{x}_i = 0$. Thus, the columns of X are assumed to sum to zero.
  $\rightsquigarrow$ compute SVD of "centered" matrix X
  $\rightsquigarrow$ column vectors in W are eigenvectors of $X^t X$
  $\rightsquigarrow$ they are the principal components.

Eigen-faces [Turk and Pentland, 1991]

- p = number of pixels
- Each $\boldsymbol{x}_i \in \mathbb{R}^p$ is a face image
- x_{ji} = intensity of the j-th pixel in image i

$$(X^t)_{p\times n} \approx W_{p\times k} (Z^t)_{k\times n}$$

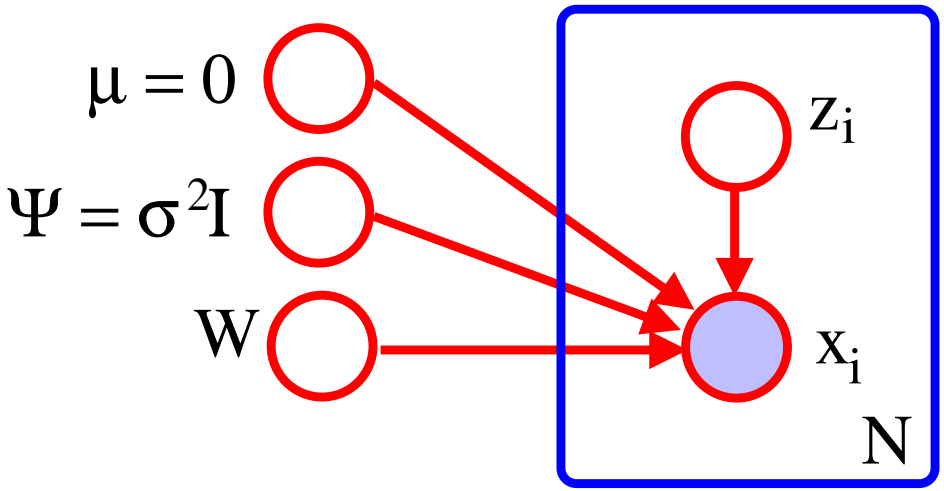
$$\left(\boldsymbol{x}_1 \; \cdots \; \boldsymbol{x}_n \right) \approx \left(\text{  } \right) \begin{bmatrix} | & & | \\ \boldsymbol{z}_1 & \cdots & \boldsymbol{z}_n \\ | & & | \end{bmatrix}$$

Idea: $\boldsymbol{z}_i$ more 'meaningful' representation of i-th face than $\boldsymbol{x}_i$

Can use $\boldsymbol{z}_i$ for nearest-neighbor classification

Much faster when $p \gg k$.

Probabilistic PCA



- Assuming $\Psi = \sigma^2 I$ and centered data in the FA model
  $\rightsquigarrow$ likelihood

$$p(\boldsymbol{x}_i|\boldsymbol{z}_i, \boldsymbol{\theta}) = \mathcal{N}(W\boldsymbol{z}_i, \sigma^2 I).$$

Probabilistic PCA

- (Tipping & Bishop 1999): Maxima of the likelihood are given by

$$\hat{W} = V(\Lambda - \sigma^2 I)^{\frac{1}{2}} R,$$

where R is an arbitrary orthogonal matrix,
columns of V: first k eigenvectors of $S = \frac{1}{n} X^t X$,
Λ: diagonal matrix of eigenvalues.

- As $\sigma^2 \to 0$, we have $\hat{W} \to V$, **as in classical PCA** (for $R = \Lambda^{-\frac{1}{2}}$).
- Projections $\boldsymbol{z}_i$: Posterior over the latent factors:

$$\begin{aligned} p(\boldsymbol{z}_i | \boldsymbol{x}_i, \hat{\boldsymbol{\theta}}) &= \mathcal{N}(\boldsymbol{z}_i | \hat{\boldsymbol{m}}_i, \sigma^2 \hat{F}^{-1}) \\ \hat{F} &= \sigma^2 I + \hat{W}^t \hat{W} \\ \boldsymbol{m}_i &= \hat{F}^{-1} \hat{W}^t \boldsymbol{x}_i \end{aligned}$$

For $\sigma^2 \to 0$, $\boldsymbol{z}_i \to \boldsymbol{m}_i$ and $\boldsymbol{m}_i \to V^t \boldsymbol{x}_i \rightsquigarrow$ orthogonal projection of the data onto the column space of V, **as in classical PCA.**