Machine learning: lecture 13

Tommi S. Jaakkola
MIT CSAIL
tommi@csail.mit.edu

Topics

- Complexity, compression, and model selection
  - description length
  - minimum description length principle
- Probabilistic modeling
  - mixture models, EM algorithm

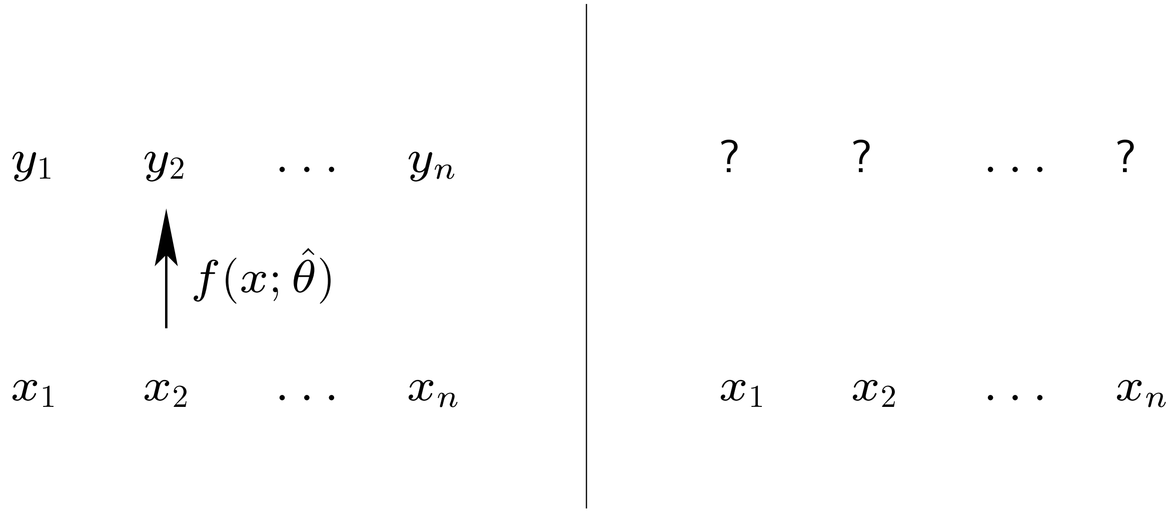
Data compression and model selection

- We can alternatively view model selection as a problem of finding the best way of communicating the available data

$$
\begin{array}{cccc|cccc}
y_1 & y_2 & \ldots & y_n & ? & ? & \ldots & ? \\
x_1 & x_2 & \ldots & x_n & x_1 & x_2 & \ldots & x_n
\end{array}
$$

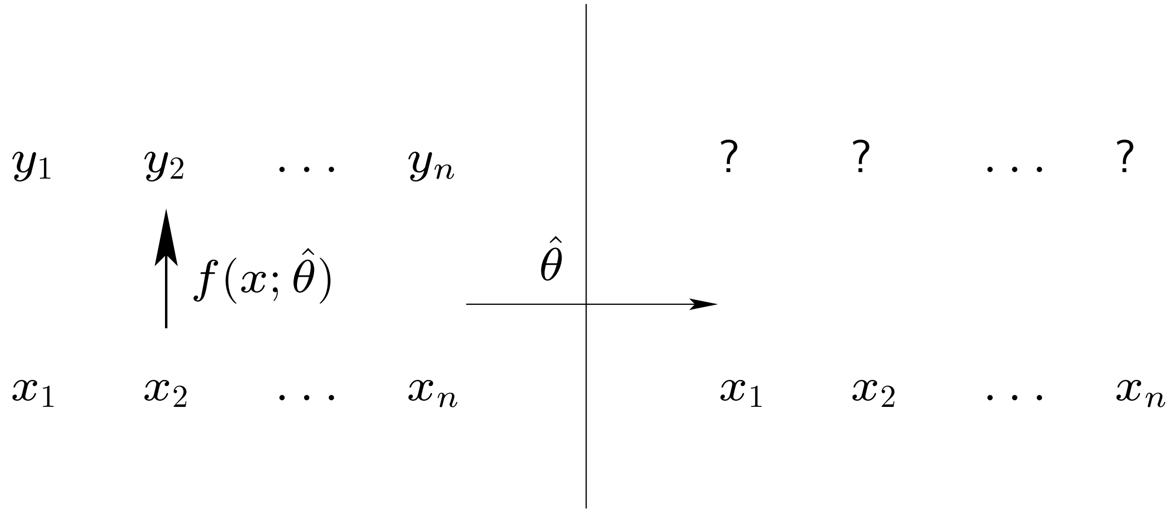
Data compression and model selection

- We can alternatively view model selection as a problem of finding the best way of communicating the available data



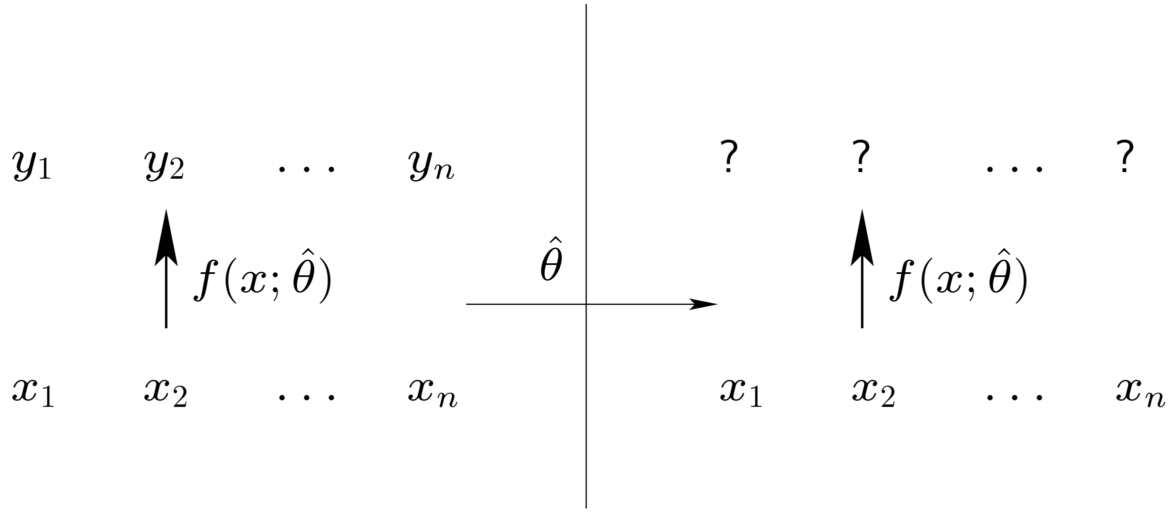
Data compression and model selection

- We can alternatively view model selection as a problem of finding the best way of communicating the available data



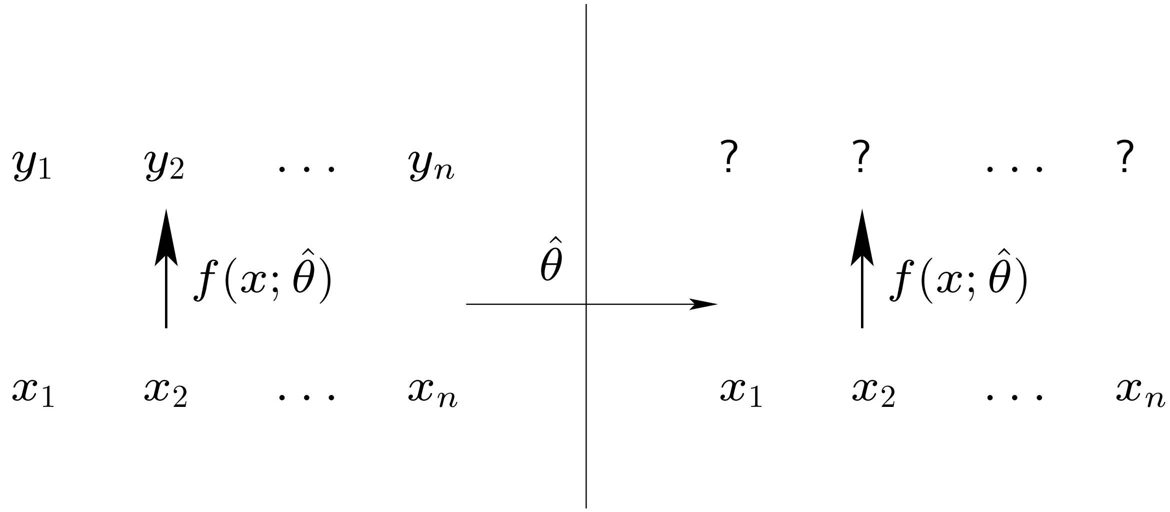
Data compression and model selection

- We can alternatively view model selection as a problem of finding the best way of communicating the available data



Data compression and model selection

- We can alternatively view model selection as a problem of finding the best way of communicating the available data



What is shared between the sender and the receiver?

– input examples $\mathbf{x}_1, \ldots, \mathbf{x}_n$

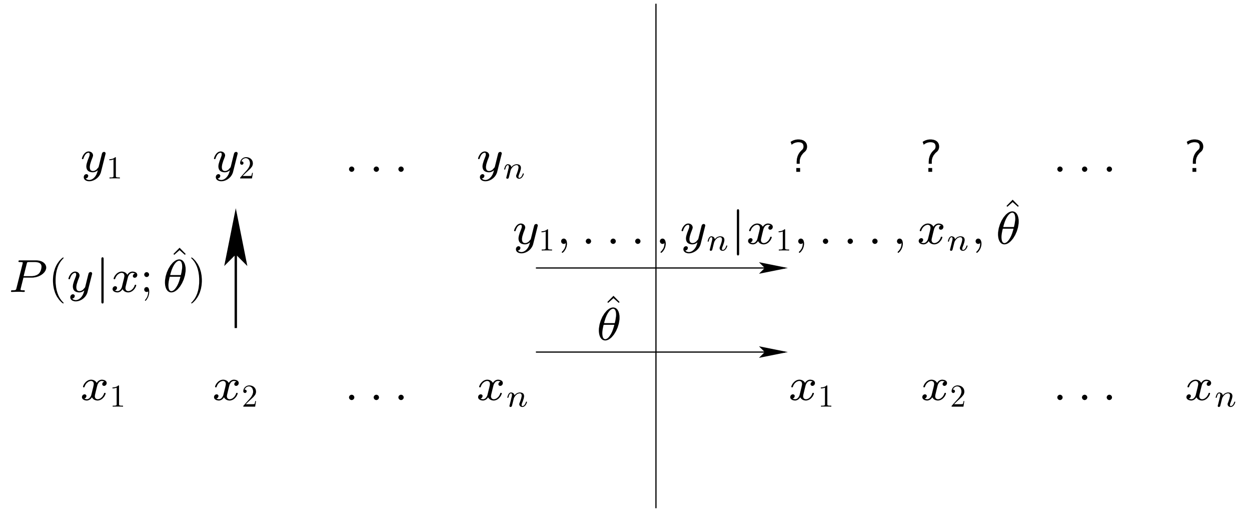
– knowledge of function classes

What needs to be communicated?

– anything pertaining to the labels $y_1, \ldots, y_n$

Data compression and model selection

- To communicate the labels effectively we need to cast the problem in probabilistic terms

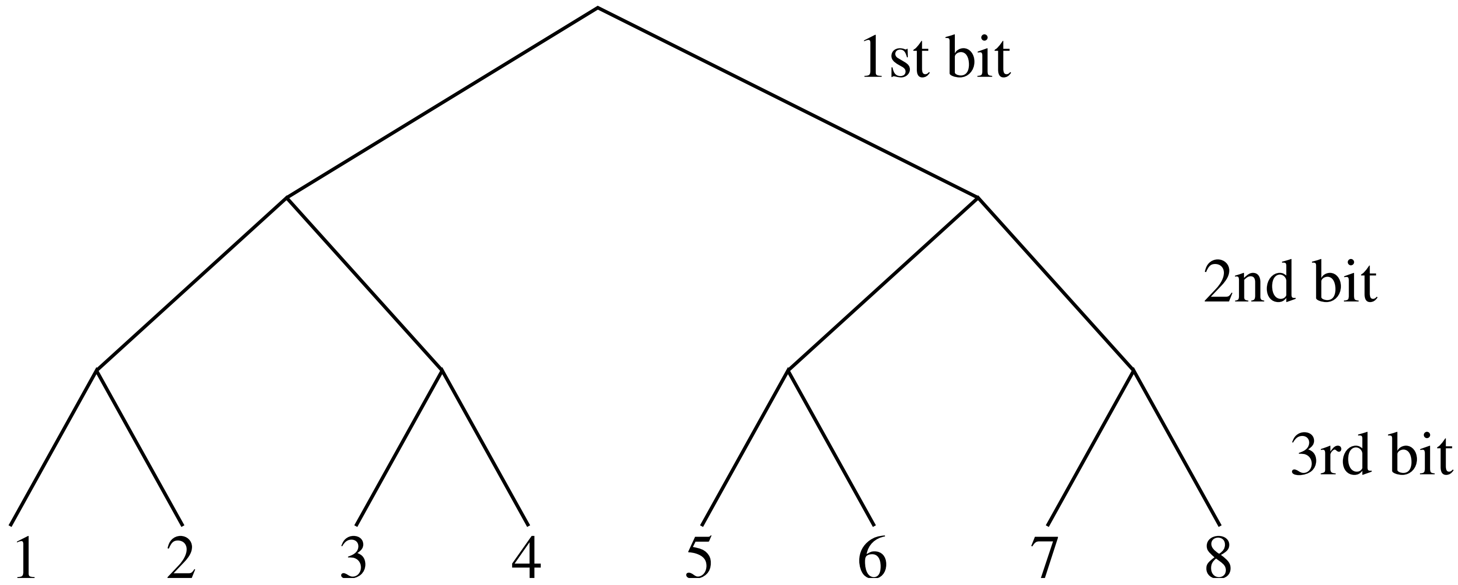


- The communication cost in bits depends on how well the model can predict the data as well as how hard it is to describe the model itself (complexity)

$$\text{Total \# of bits} = \text{bits to describe the data given the model} + \text{bits to describe the model}$$

Bits and probabilities

- How many bits do we need to communicate a specific selection out of a set of eight equally likely choices?



We need $-\log_2 P(y) = -\log_2(1/8) = 3$ bits to describe each y.

Description length

- How many bits do we need to describe

  0111111111110111001111111111111110111111

- If we assume that the bits $\{y_i\}$ in the sequence are independent random draws from P, where $P(y = 1) = 0.5$, then

$$\sum_{i=1}^{40} (-\log_2 P(y_i)) = 40\text{bits}$$

- If we assume instead that $P(y = 1) = 0.9$, then

$$\sum_{i=1}^{40} (-\log_2 P(y_i)) \approx 22\text{bits}$$

- What we assume matters a great deal.

Conditional description length

- We can also describe outcomes conditionally, i.e., determine the number of bits we need to specify y given $\mathbf{x}$

$$\begin{array}{ccccc} y_1 & y_2 & y_3 & y_4 & \ldots \\ \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4 & \ldots \end{array}$$

  Assuming the labels are generated from a conditional distribution $P(y|\mathbf{x},\theta)$, we need

$$\sum_i (-\log_2 P(y_i|\mathbf{x}_i,\theta))$$

  bits to describe the outcomes (labels).

- The actual number of bits may vary considerably as a function of the parameters θ.

Description length cont'd

- We can of course find $\hat{\theta} \in \Theta$ (the maximum likelihood parameter estimate) that minimizes the number of bits needed to describe the labels given examples

$$\sum_i \Big(- \log_2 P(y_i|\mathbf{x}_i, \hat{\theta}) \Big)$$

The minimizing $\hat{\theta}$ depends on the labels and needs to be communicated as well.

Description length cont'd

- In addition to describing the data using $\hat{\theta}$ with

$$\sum_i \left(-\log_2 P(y_i|\mathbf{x}_i, \hat{\theta}) \right) \text{ bits},$$

we have to communicate $\hat{\theta}$.

$$\texttt{total DL} = \texttt{DL of data using } \hat{\theta} + \texttt{DL of } \hat{\theta}$$

- The description length of the parameters $\hat{\theta}$ depends on the model (the set of distributions we are considering)
  - the more choices we have, the more bits it takes to describe any specific selection

How to describe the parameters

- We need to encode the parameters up to a finite precision $\delta_k = 1/2^k$, i.e., use k significant bits (we assume here that the precision is the same for all parameters)
- With the help of a prior density $p(\theta)$, it takes us roughly speaking

$$-\log_2 \left(p(\theta_{\delta_k})\delta_k^d \right)$$

bits to describe any finite precision choice θ_{δ_k}. Here d is the dimensionality of the parameter vector θ.

How to describe the parameters cont'd

- We also need to communicate our choice of precision or k since this choice may be based on the labels.

  This takes us

  $$\log_2^*(k) \;=\; \log_2(k) + \log_2 \log_2(k) + \ldots \quad \text{bits}$$

  (based on a specific prior over integers).

Description length

- The total description length – bits needed to communicate the labels given examples – is given by the minimum of

$$\sum_i \left(-\log_2 P(y_i|\mathbf{x}_i, \theta_{\delta_k})\right) - \log_2 \left(p(\theta_{\delta_k})\delta_k^d\right) + \log_2^*(k)$$

where the minimization is taken with respect to finite precision choices θ_{δ_k} as well as the number of significant bits k.

Description length

- The total description length – bits needed to communicate the labels given examples – is given by the minimum of

$$\sum_i \left(-\log_2 P(y_i|\mathbf{x}_i, \theta_{\delta_k})\right) - \log_2 \left(p(\theta_{\delta_k})\delta_k^d\right) + \log_2^*(k)$$

where the minimization is taken with respect to finite precision choices θ_{δ_k} as well as the number of significant bits k.

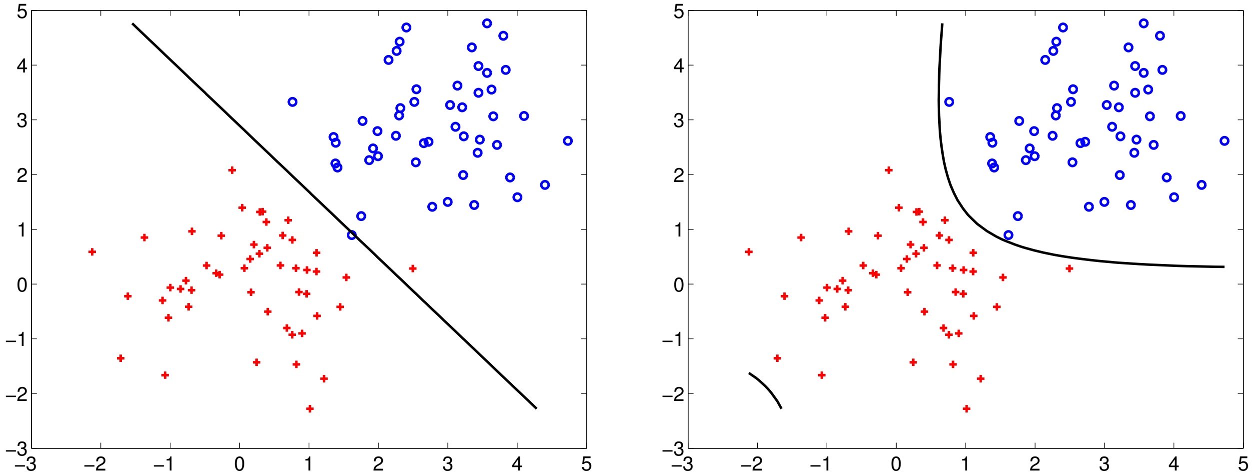
- For large n we can use the following asymptotic expansion:

$$\sum_i \left(-\log_2 P(y_i|\mathbf{x}_i, \hat{\theta})\right) + \frac{d}{2}\log_2(n)$$

where $\hat{\theta}$ is the maximum likelihood setting of the parameters and d is the effective number of parameters.

Description length: example

- Example: polynomial logistic regression, $n = 100$

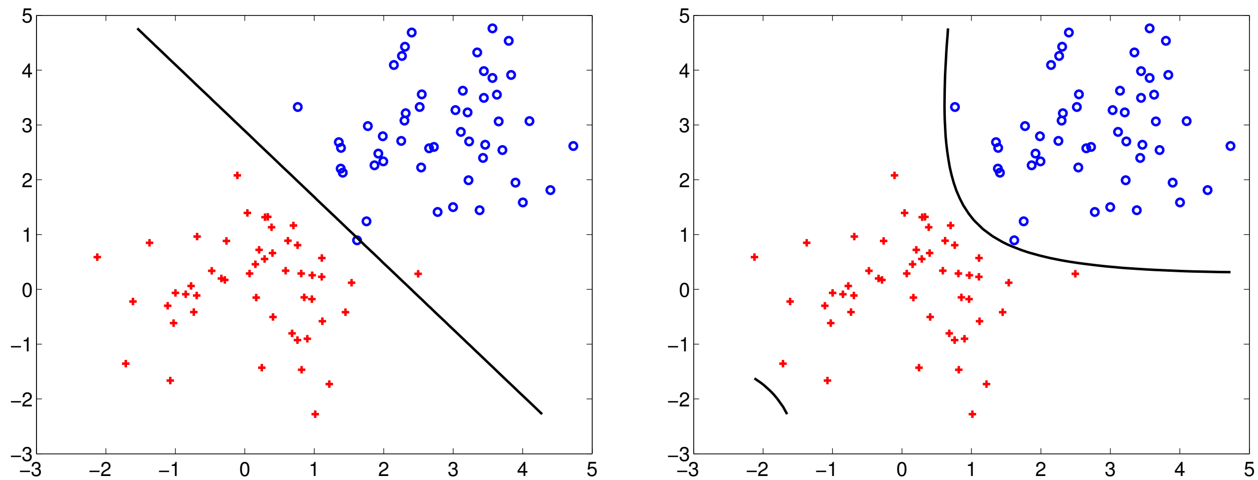


linear quadratic

$$\text{DL} = \sum_i \left(-\log_2 P(y_i|\mathbf{x}_i, \hat{\theta}) \right) + \frac{d}{2}\log_2(n)$$

Description length: example

- Example: polynomial logistic regression, $n = 100$



linear quadratic

$$\mathsf{DL} = \sum_i \left(-\log_2 P(y_i|\mathbf{x}_i, \hat{\theta}) \right) + \frac{d}{2}\log_2(n)$$

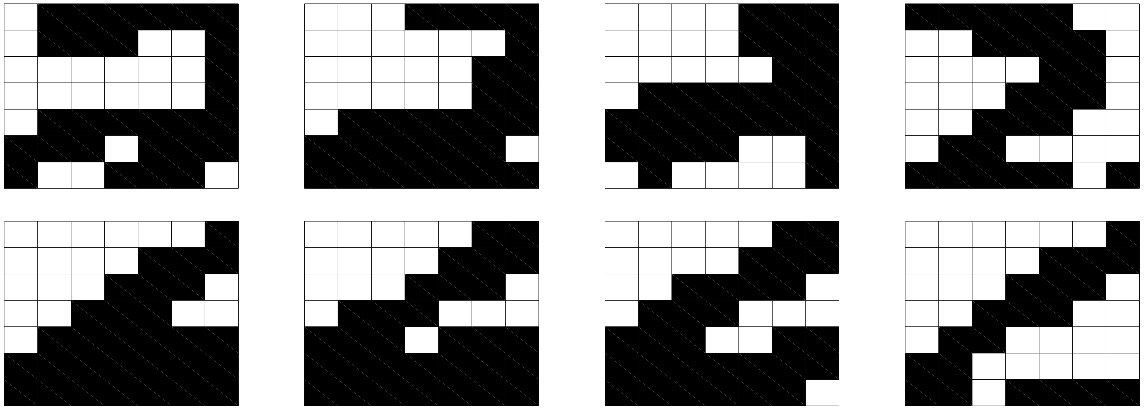
degree	# param	DL(data)	DL(model)	MDL score
1	3	5.6 bits	9.9 bits	15.5 bits
2	6	2.4 bits	19.9 bits	22.3 bits

Topics

- Complexity, compression, and model selection
  - description length
  - minimum description length principle
- Probabilistic modeling
  - mixture models, EM algorithm

Probabilistic modeling

The digits again...



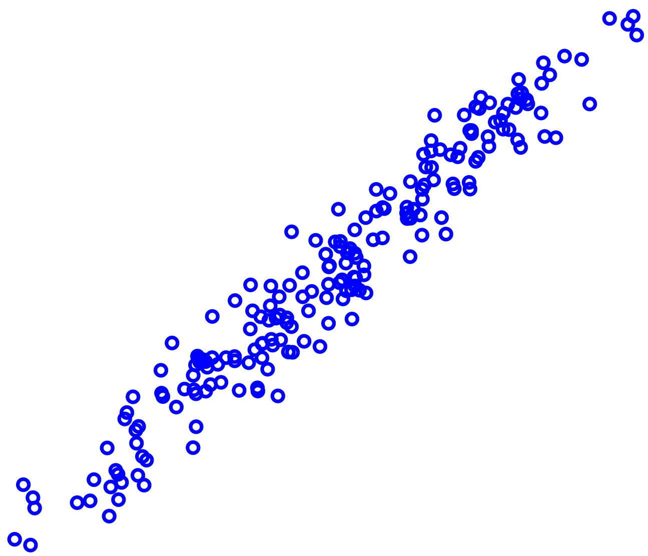
- Possible uses:
  - clustering
  - understanding the generation process of examples
  - classification via class-conditional densities
  - inference based on incomplete observations

Parametric density models

- Probability model $=$ a parameterized family of probability distributions
- Example: a simple multivariate Gaussian model

$$p(\mathbf{x}|\mu, \Sigma) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp\{ -\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu) \}$$

- This is a *generative model* in the sense that we can generate $\mathbf{x}$'s

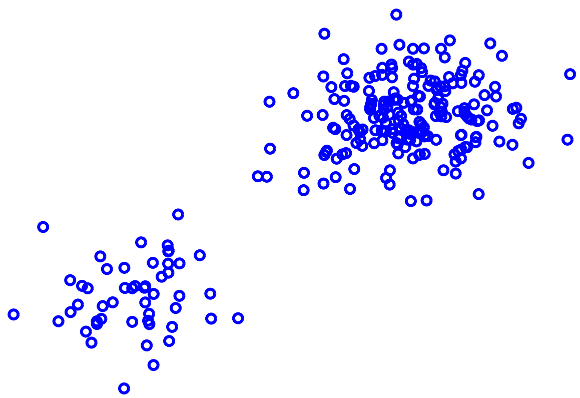


Multi-variate density estimation

- A mixture of Gaussians model

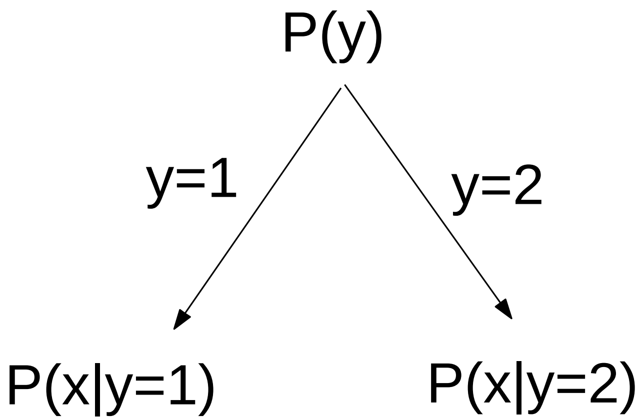
$$p(\mathbf{x}|\theta) = \sum_{i=1}^{k} p_j \, p(\mathbf{x}|\mu_j, \Sigma_j)$$

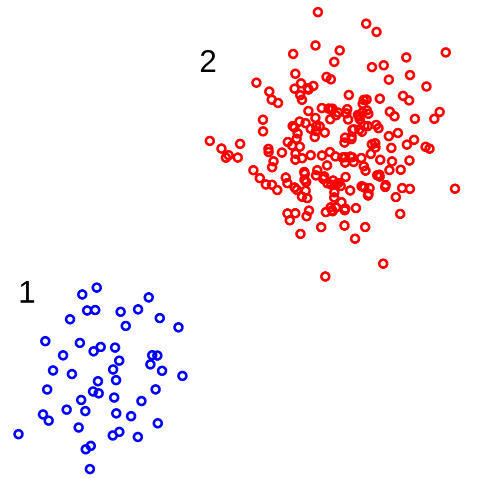
where $\theta = \{p_1, \ldots, p_k, \mu_1, \ldots, \mu_k, \Sigma_1, \ldots, \Sigma_k\}$ contains all the parameters of the mixture model. $\{p_j\}$ are known as *mixing proportions or coefficients*.



Mixture density

- Data generation process:





$$p(\mathbf{x}|\theta) = \sum_{j=1,2} P(y=j) \cdot p(\mathbf{x}|y=j) \quad \text{(generic mixture)}$$

$$= \sum_{j=1,2} p_j \cdot p(\mathbf{x}|\mu_j, \Sigma_j) \quad \text{(mixture of Gaussians)}$$

- Any data point $\mathbf{x}$ could have been generated in two ways

Mixture density

- If we are given just $\mathbf{x}$ we don't know which mixture component this example came from

$$p(\mathbf{x}|\theta) = \sum_{j=1,2} p_j p(\mathbf{x}|\mu_j, \Sigma_j)$$

- We can evaluate the posterior probability that an observed $\mathbf{x}$ was generated from the first mixture component

$$\begin{aligned} P(y=1|\mathbf{x},\theta) &= \frac{P(y=1)\cdot p(\mathbf{x}|y=1)}{\sum_{j=1,2} P(y=j)\cdot p(\mathbf{x}|y=j)} \\ &= \frac{p_1\, p(\mathbf{x}|\mu_1,\Sigma_1)}{\sum_{j=1,2} p_j\, p(\mathbf{x}|\mu_j,\Sigma_j)} \end{aligned}$$

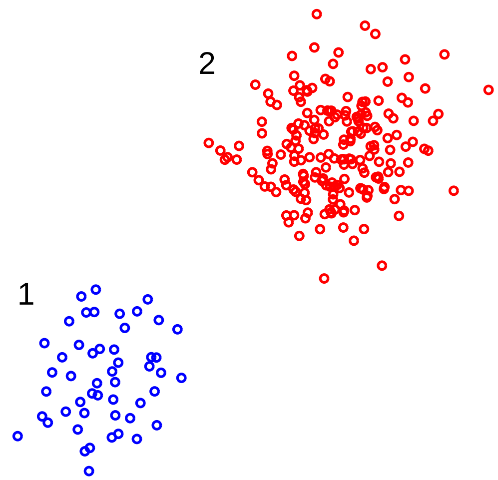
- This solves a *credit assignment* problem

Mixture density estimation

- Suppose we want to estimate a two component mixture of Gaussians model.

$$p(\mathbf{x}|\theta) = p_1\, p(\mathbf{x}|\mu_1, \Sigma_1) + p_2\, p(\mathbf{x}|\mu_2, \Sigma_2)$$

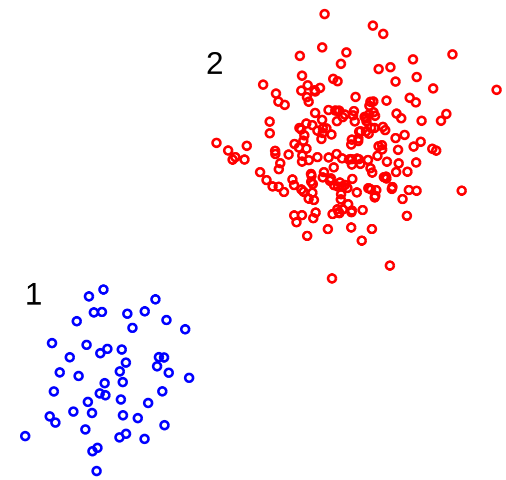
- If each example $\mathbf{x}_i$ in the training set were labeled $y_i = 1, 2$ according to which mixture component (1 or 2) had generated it, then the estimation would be easy.



- Labeled examples $\Rightarrow$ no credit assignment problem

Mixture density estimation

- When examples are labeled, we can estimate each Gaussian independently
- Let $\delta(j|i)$ be an indicator function of whether example i is labeled j. Then for each $j = 1, 2$



$$\hat{p}_j \leftarrow \frac{\hat{n}_j}{n}, \text{ where } \hat{n}_j = \sum_{i=1}^{n} \delta(j|i)$$

$$\hat{\mu}_j \leftarrow \frac{1}{\hat{n}_j} \sum_{i=1}^{n} \delta(j|i)\, \mathbf{x}_i$$

$$\hat{\Sigma}_j \leftarrow \frac{1}{\hat{n}_j} \sum_{i=1}^{n} \delta(j|i)\, (\mathbf{x}_i - \hat{\mu}_j)(\mathbf{x}_i - \hat{\mu}_j)^T$$

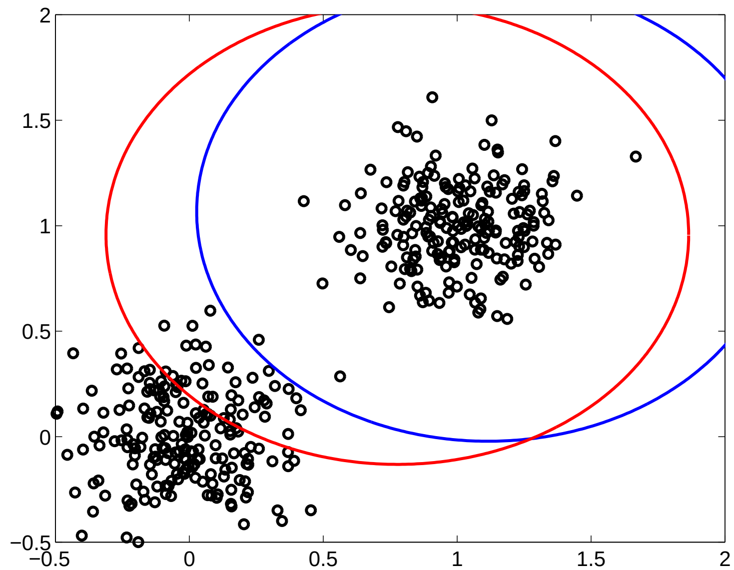
Mixture density estimation: credit assignment

- Of course we don't have such labels ... but we can guess what the labels might be based on our current mixture distribution

- We get soft labels or posterior probabilities of which Gaussian generated which example:

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

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



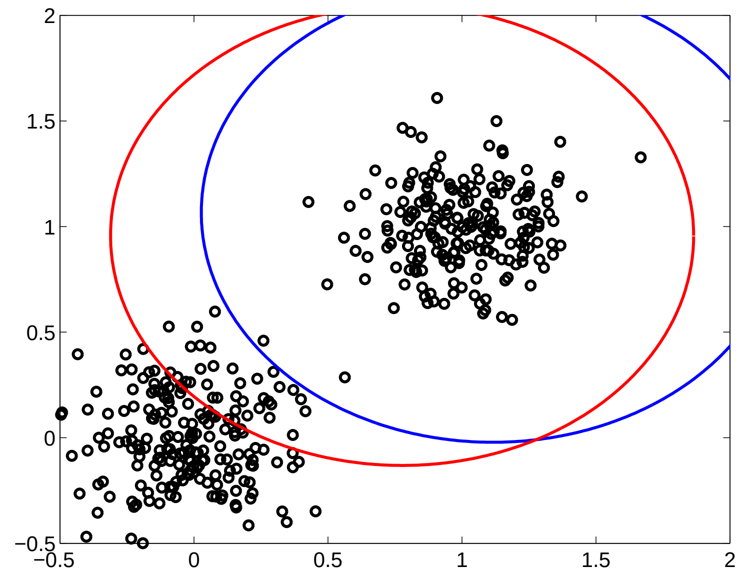
Mixture density estimation: credit assignment

- Of course we don't have such labels ... but we can guess what the labels might be based on our current mixture distribution

- We get soft labels or posterior probabilities of which Gaussian generated which example:

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

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



- When the Gaussians are almost identical (as in the figure), $\hat{p}(1|i) \approx \hat{p}(2|i)$ for almost any available point $\mathbf{x}_i$.

  Even slight differences can help us determine how we should modify the Gaussians.

The EM algorithm

E-step: softly assign examples to mixture components

$\hat{p}(j|i) \leftarrow P(y_i = j|\mathbf{x}_i, \theta)$, for all $j = 1, 2$ and $i = 1, \ldots, n$

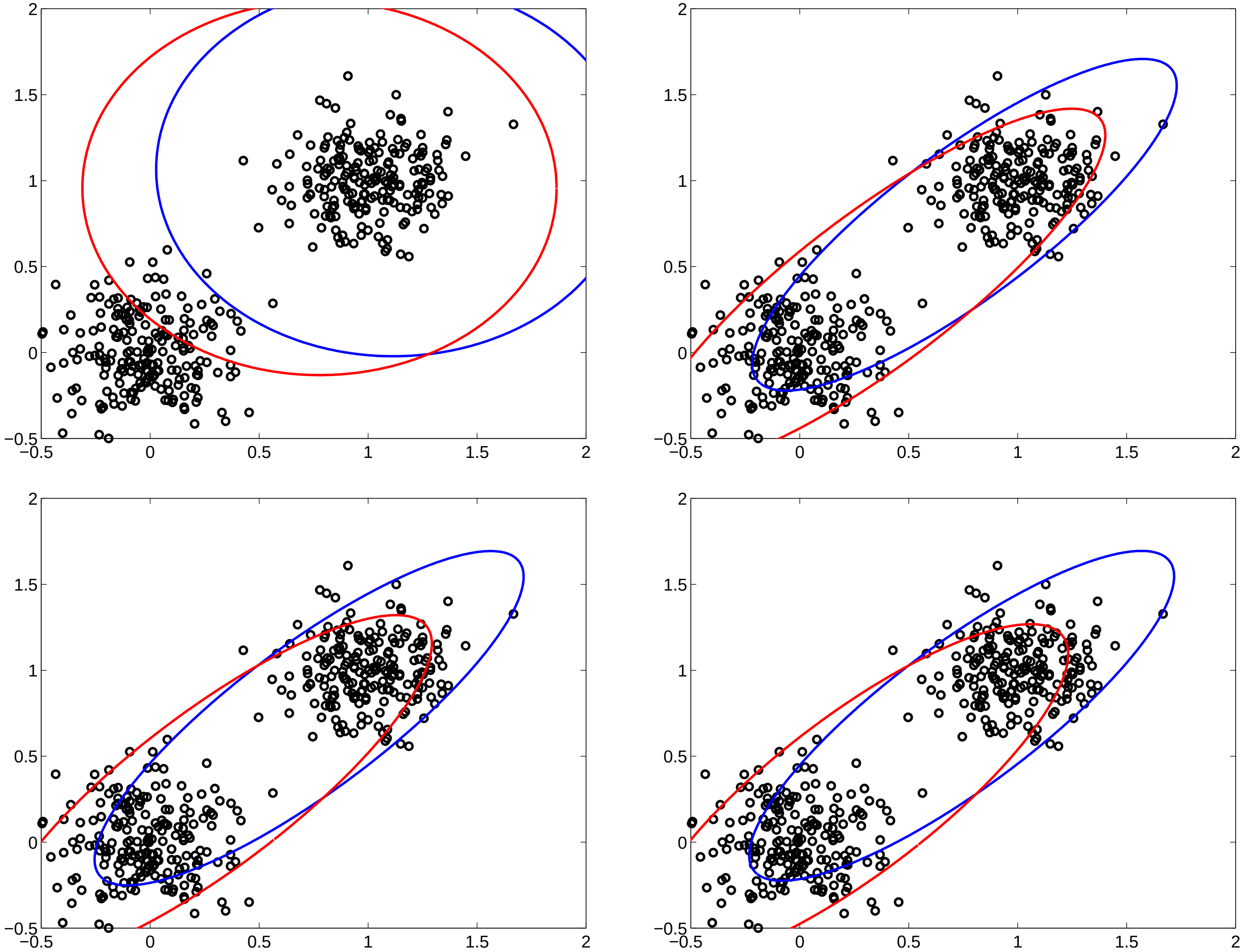
M-step: re-estimate the parameters (separately for the two Gaussians) based on the soft assignments.

$$\hat{p}_j \leftarrow \frac{\hat{n}_j}{n}, \quad \text{where } \hat{n}_j = \sum_{i=1}^{n} \hat{p}(j|i)$$

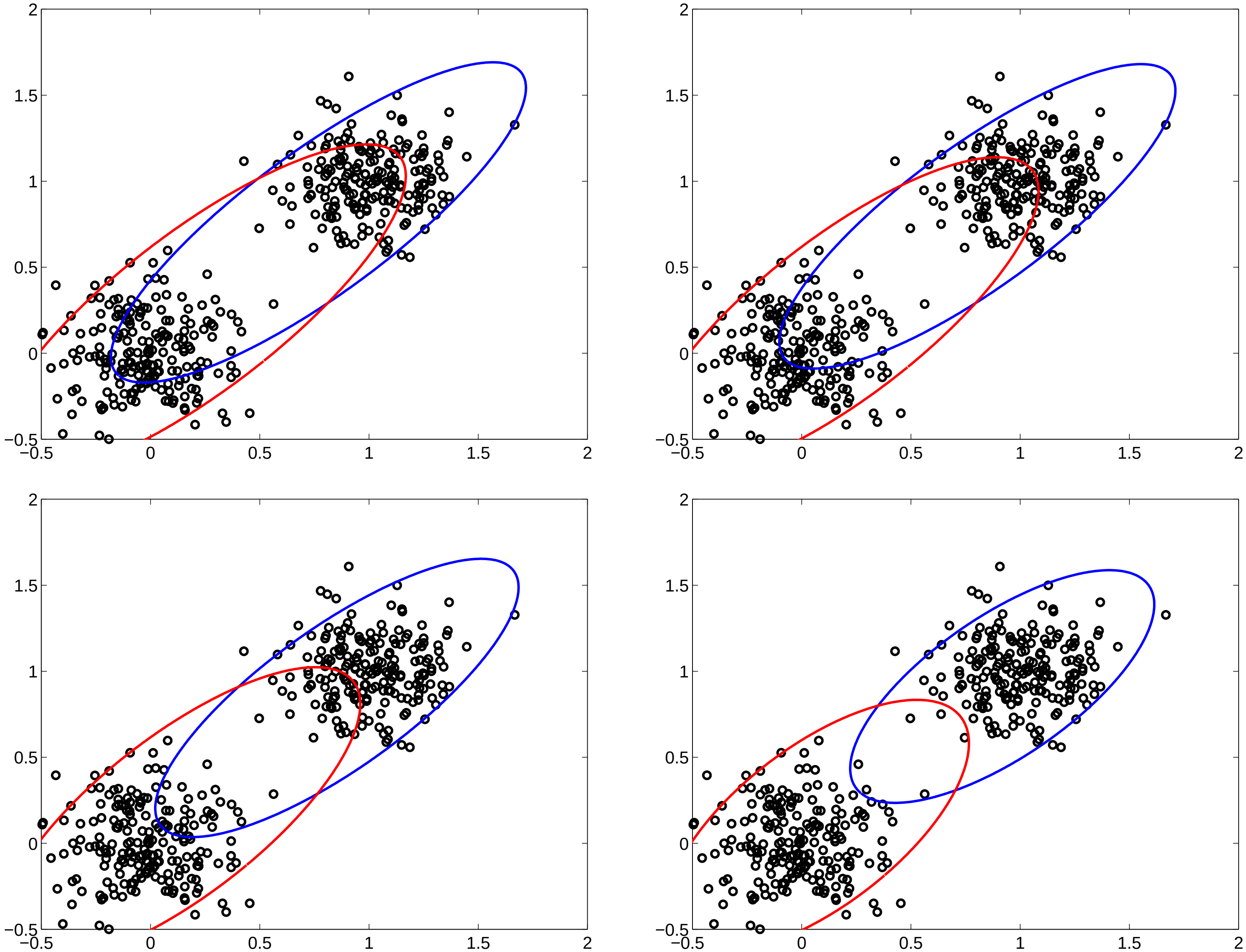
$$\hat{\mu}_j \leftarrow \frac{1}{\hat{n}_j} \sum_{i=1}^{n} \hat{p}(j|i)\, \mathbf{x}_i$$

$$\hat{\Sigma}_j \leftarrow \frac{1}{\hat{n}_j} \sum_{i=1}^{n} \hat{p}(j|i)\, (\mathbf{x}_i - \hat{\mu}_j)(\mathbf{x}_i - \hat{\mu}_j)^T$$

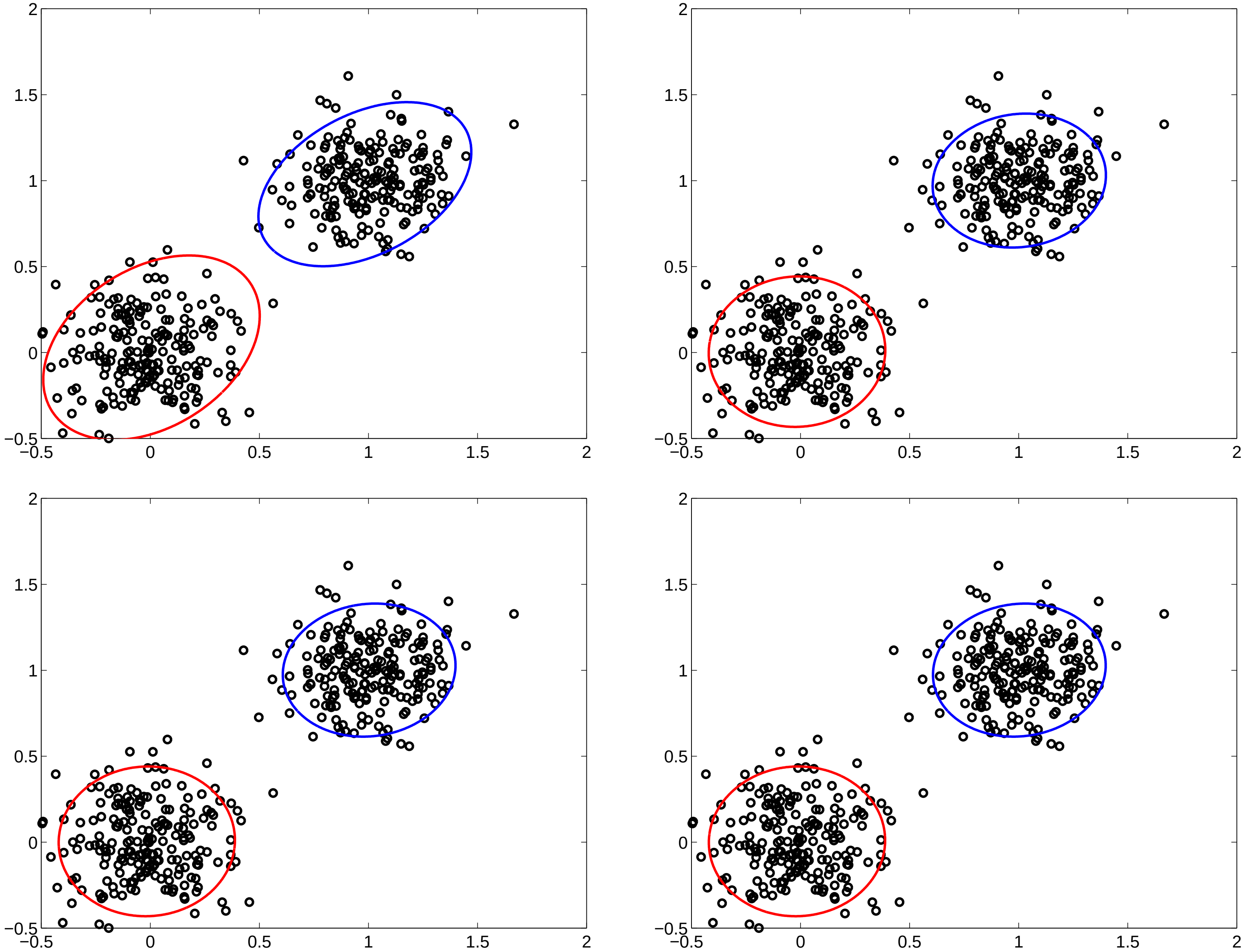
Mixture density estimation: example



Mixture density estimation



Mixture density estimation



The EM-algorithm

- Each iteration of the EM-algorithm *monotonically* increases the (log-)likelihood of the n training examples $\mathbf{x}_1, \ldots, \mathbf{x}_n$:

$$\log p(\text{ data } | \theta) = \sum_{i=1}^{n} \log \left(\overbrace{p_1 \, p(\mathbf{x}_i | \mu_1, \Sigma_1) + p_2 \, p(\mathbf{x}_i | \mu_2, \Sigma_2)}^{p(\mathbf{x}_i | \theta)} \right)$$

where $\theta = \{p_1, p_2, \mu_1, \mu_2, \Sigma_1, \Sigma_2\}$ contains all the parameters of the mixture model.