6.867 Machine learning and neural networks

Tommi Jaakkola
MIT AI Lab
tommi@ai.mit.edu

Lecture 11: model selection, density estimation

Topics

- Model selection cont'd
  - Structural risk minimization
  - Example
- Density estimation
  - Parametric, mixture models
  - Estimation via the EM algorithm

Structural risk minimization

- We have a nested sequence of models of increasing complexity; complexity measured in terms of VC-dimension (or refinements)

  Model 1 $d_{VC} = d_1$
  Model 2 $d_{VC} = d_2$
  Model 3 $d_{VC} = d_3$

  where $d_1 \leq d_2 \leq d_3 \leq \ldots$

- Basic formulation: we derive an upper *bound* on the expected loss

  $$\text{Expected loss} \leq \text{Empirical loss} + \text{Complexity penalty}$$

  and select the model that gives the lowest *bound*.

Example

- Models of increasing complexity

Model 1	$K(\mathbf{x}_1, \mathbf{x}_2) = (1 + (\mathbf{x}_1^T \mathbf{x}_2))$
Model 2	$K(\mathbf{x}_1, \mathbf{x}_2) = (1 + (\mathbf{x}_1^T \mathbf{x}_2))^2$
Model 3	$K(\mathbf{x}_1, \mathbf{x}_2) = (1 + (\mathbf{x}_1^T \mathbf{x}_2))^3$
...	...

- These are nested, i.e.,

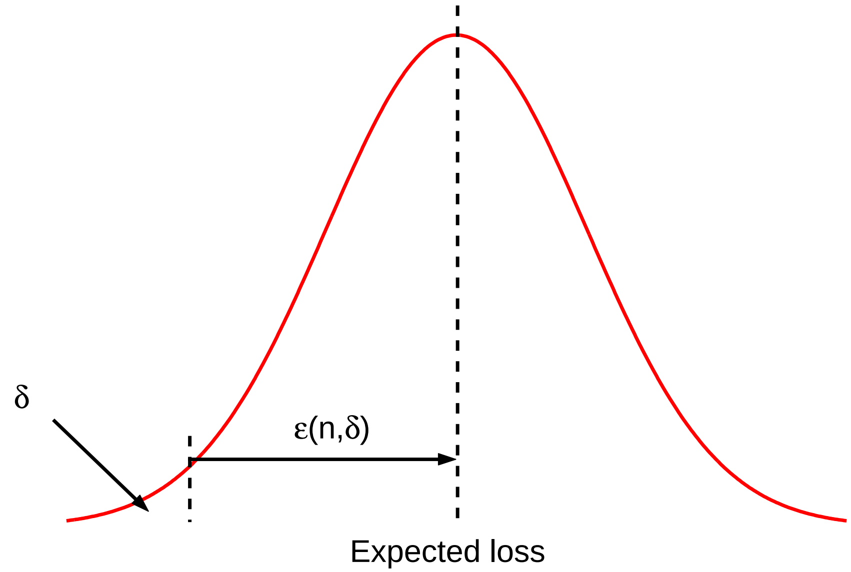
$$F_1 \subseteq F_2 \subseteq F_3 \subseteq \ldots$$

  where F_k refers to the set of possible decision boundaries that the model k can represent.

- Still need to derive the criterion...

Bounds on expected loss

- A single fixed classifier $h(\mathbf{x})$, n training points



With probability at least $1-\delta$ over the choice of the training set

$$\overbrace{E\{\,\text{Loss}(y,h(\mathbf{x}))\,\}}^{\text{Expected loss}} \;\leq\; \overbrace{\frac{1}{n}\sum_{i=1}^{n}\text{Loss}(y_i,h(\mathbf{x}_i))}^{\text{Empirical loss}} + \overbrace{\epsilon(n,\delta)}^{\text{sampling penalty}}$$

- For the bound to be valid uniformly for all classifiers in the set F, we have to include the VC-dim

Structural risk minimization

- Finite VC-dimension gives us some guarantees about how close the empirical loss is to the expected loss

  With probability at least $1-\delta$ over the choice of the training set, for all $h \in F_k$

$$\overbrace{E\{\,\text{Loss}(y, h(\mathbf{x}))\,\}}^{\text{Expected loss}} \;\leq\; \overbrace{\frac{1}{n}\sum_{i=1}^{n} \text{Loss}(y_i, h(\mathbf{x}_i))}^{\text{Empirical loss}} + \overbrace{\epsilon(n, \delta, d_k)}^{\text{Complexity penalty}}$$

  where

$$\begin{aligned} d_k &= \text{VC-dimension of model (set of hypothesis) } k \\ \delta &= \text{Confidence parameter (probability of failure)} \end{aligned}$$

- We find model k that has the lowest bound on the expected loss

Structural risk minimization cont'd

- For our zero-one loss (classification error), we can derive the following complexity penalty (Vapnik 1995):

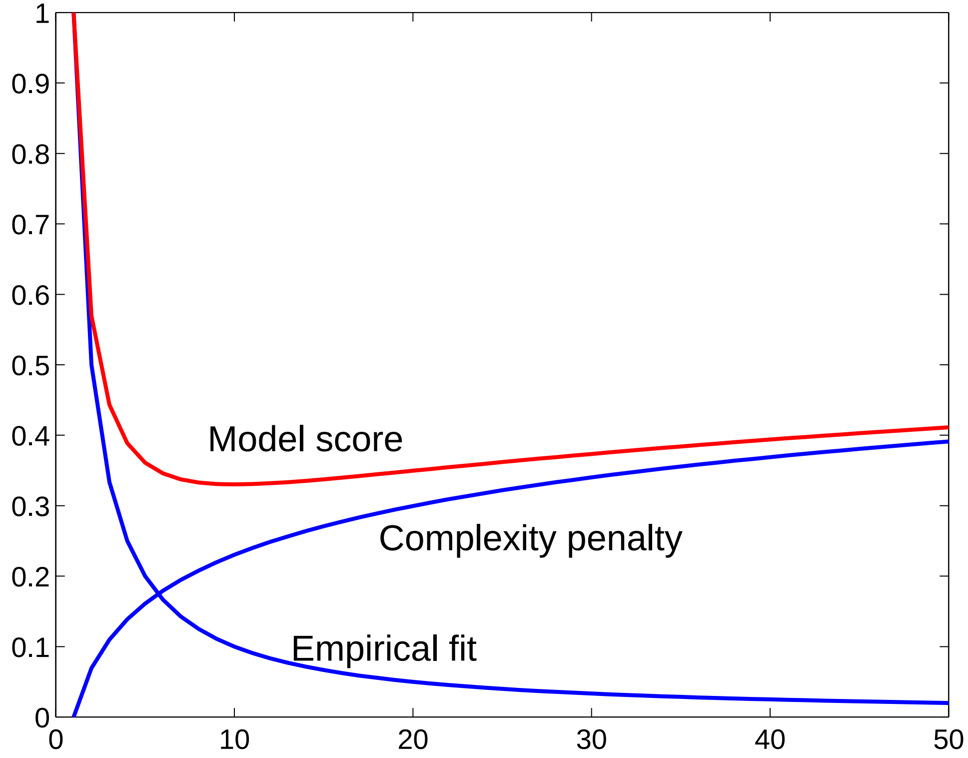
$$\epsilon(n, \delta, d) = \sqrt{\frac{d_{VC}(\log(2n/d_{VC}) + 1) + \log(1/(4\delta))}{n}}$$

1. This is an increasing function of d_{VC}
2. Increases as δ decreases
3. Decreases as a function of n

(this is not the only choice...)

Structural risk minimization cont'd

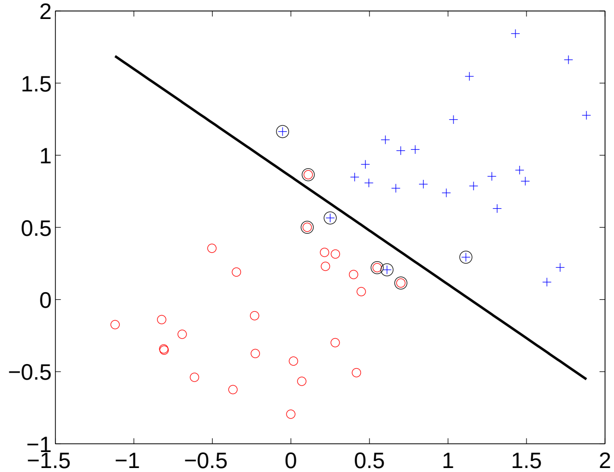
- Competition of terms...
  1. Empirical loss decreases with increasing d_{VC}
  2. Complexity penalty increases with increasing d_{VC}



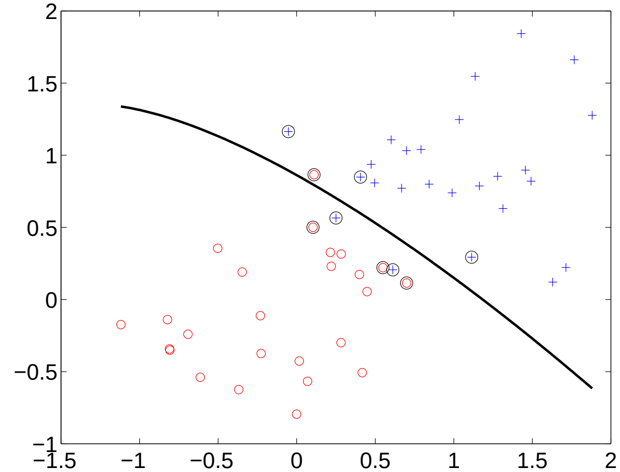
- We find the minimum of the combined score

Structural risk minimization: example

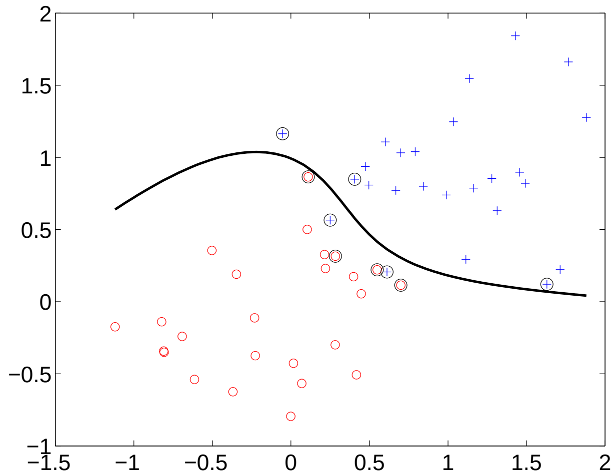
- The same problem as in the previous lecture



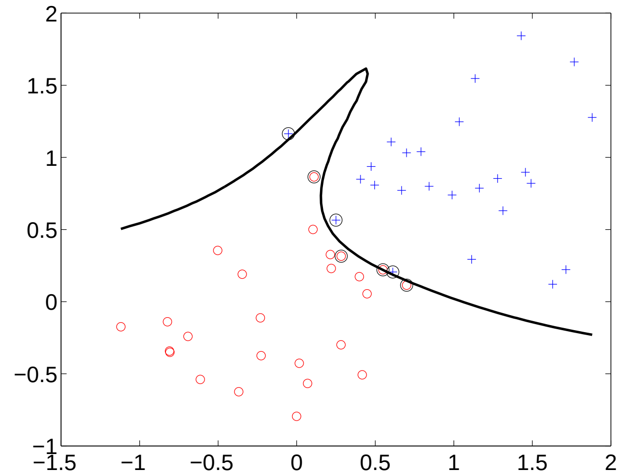
linear



2^{nd} order polynomial



4^{th} order polynomial



8^{th} order polynomial

Structural risk minimization: example cont'd

- Number of training examples $n = 50$, confidence parameter $\delta = 0.05$.

Model	d_{VC}	Empirical fit	Complexity penalty $\epsilon(n, \delta, d_{VC})$
1^{st} order	3	0.06	0.5501
2^{nd} order	6	0.06	0.6999
4^{th} order	15	0.04	0.9494
8^{th} order	45	0.02	1.2849

- Structural risk minimization would clearly select the simplest (linear) model in this case.

Topics

- Density estimation
  - Parametric, mixture models
  - Estimation via the EM algorithm

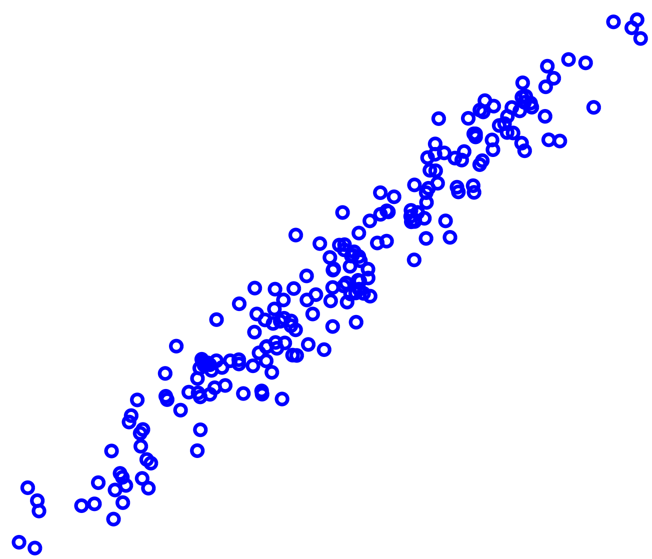
Parametric density models

- Probability model = a class of probability distributions
- Example: a simple multivariate Gaussian model

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

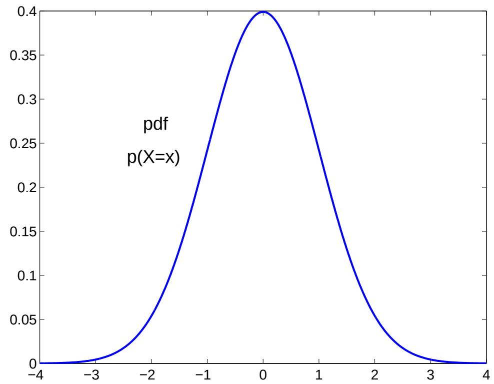
- This is a *generative model* in the sense that we can generate $\mathbf{x}$'s

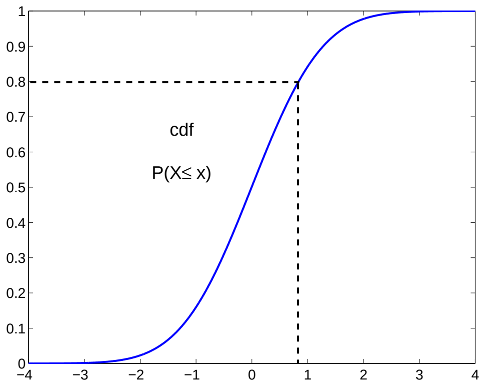
  How do I generate a sample from a specific multivariate Gaussian distribution?



Gaussian samples

- 1-dimensional Gaussian *probability density function* (pdf) $P(x|\mu, \Sigma)$ and *cumulative distribution function* (cdf)





- To draw a sample from a Gaussian, we can invert the cumulative distribution function $F(x) = \int_{-\infty}^{x} P(z|\mu, \Sigma) dz$:

$$u \sim \text{Uniform}(0, 1) \;\;\Rightarrow\;\; x = F^{-1}(u) \sim P(x|\mu, \Sigma)$$

- A multivariate sample can be constructed from multiple independent one dimensional Gaussian samples $\mathbf{z} = [z_1, \ldots, z_d]^T$:

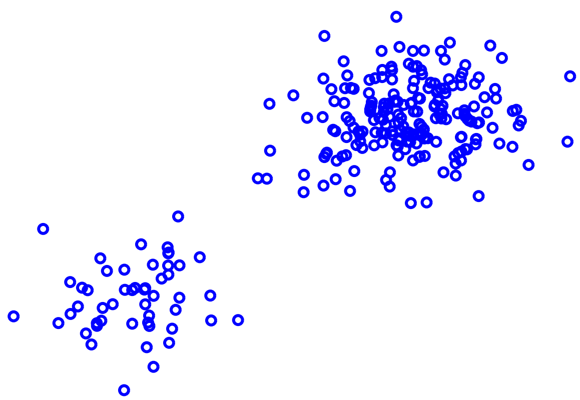
$$\mathbf{x} = \Sigma^{1/2}\mathbf{z} + \mu \;\;\Rightarrow\;\; \mathbf{x} \sim P(x|\mu, \Sigma)$$

Parametric density models

- A mixture of Gaussians model

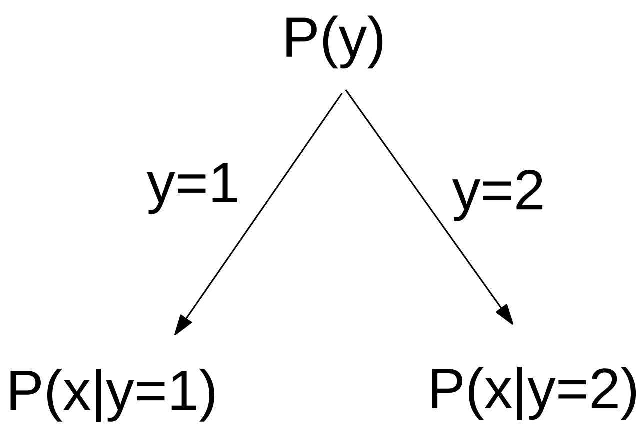
$$P(\mathbf{x}|\theta) = \sum_{i=1}^{k} p_j \, P(\mathbf{x}|\mu_j, \Sigma_j)$$

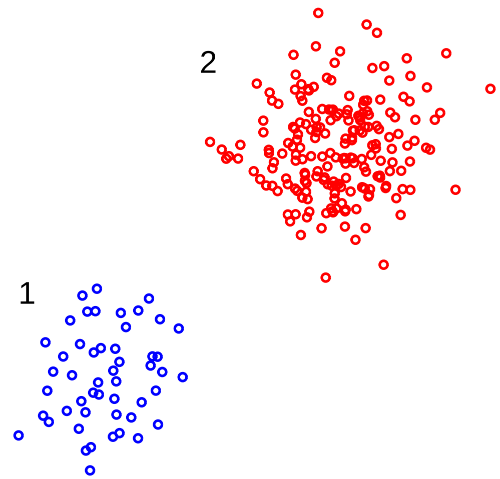
where $\theta = \{p_1, \ldots, p_k, \mu_1, \ldots, \mu_k, \Sigma_1, \ldots, \Sigma_k\}$ contains all the parameters of the mixture model. $\{p_j\}$ are known as *mixing proportions or coefficients*.



Mixture density

- Data generation process:





$$\begin{aligned} P(\mathbf{x}) &= \sum_{j=1,2} P(y=j) \cdot P(\mathbf{x}|y=j) \quad \text{(generic mixture)} \\ &= \sum_{j=1,2} p_j \cdot P(\mathbf{x}|\mu_j, \Sigma_j) \quad \text{(mixture of Gaussians)} \end{aligned}$$

(exclusive events, additive probabilities)

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

Mixture density

- For any $\mathbf{x}$, we do not know which mixture component generated it but we assume one of them did.

$$P(\mathbf{x}) = \sum_{j=1,2} P(y=j) \cdot P(\mathbf{x}|y=j)$$

- What is the posterior probability that $\mathbf{x}$ was generated by the first mixture component?

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

- This posterior probability solves a *credit assignment* problem

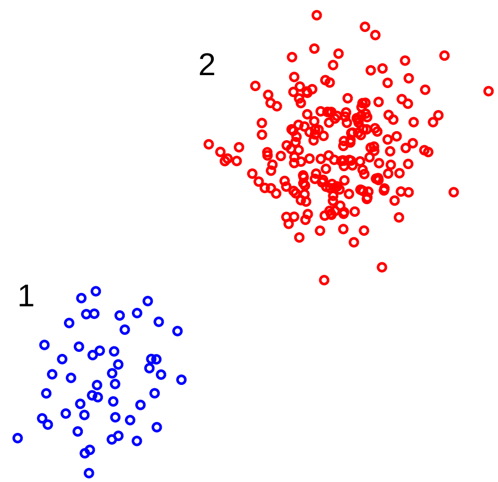
Mixture density estimation

(For simplicity, we'll look at only maximum likelihood estimation)

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

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

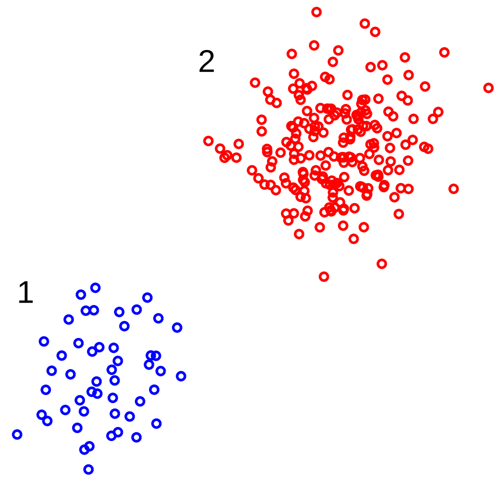
- If each example $\mathbf{x}_i$ in the training set were labeled $y_i = 1, 2$ according to which mixture component (1 or 2) generated it, then the estimation would be easy.



- Labeled examples $\Rightarrow$ no credit assignment problem

Mixture density estimation

If the examples were labeled, we could estimate each Gaussian independently of each other



- Separately for $j = 1, 2$

$$\hat{n}_j \leftarrow \sum_{i:y_i=j} 1 = \#\text{ of examples labeled } j$$

$$\hat{p}_j \leftarrow \frac{\hat{n}_j}{n}$$

$$\hat{\mu}_j \leftarrow \frac{1}{\hat{n}_j} \sum_{i:y_i=j} \mathbf{x}_i$$

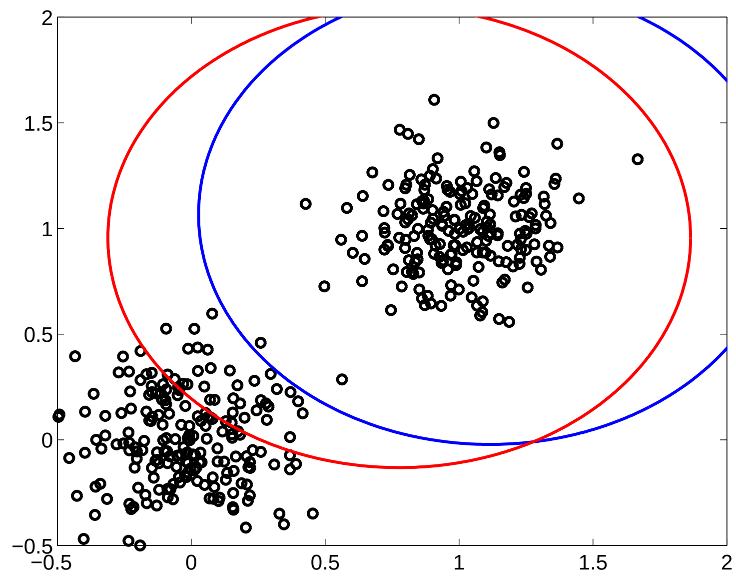
$$\hat{\Sigma}_j \leftarrow \frac{1}{\hat{n}_j} \sum_{i:y_i=j} (\mathbf{x}_i - \hat{\mu}_j)(\mathbf{x}_i - \hat{\mu}_j)^T$$

Mixture density estimation: credit assignment

- Of course we don't have such a labels ... but we can guess what the labels might be based on our current mixture distribution
- We get soft labels, posterior probabilities of which Gaussian generated which example:

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

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



The EM algorithm

E-step: First we perform a soft reassignment of examples based on the current mixture distribution, i.e., we compute

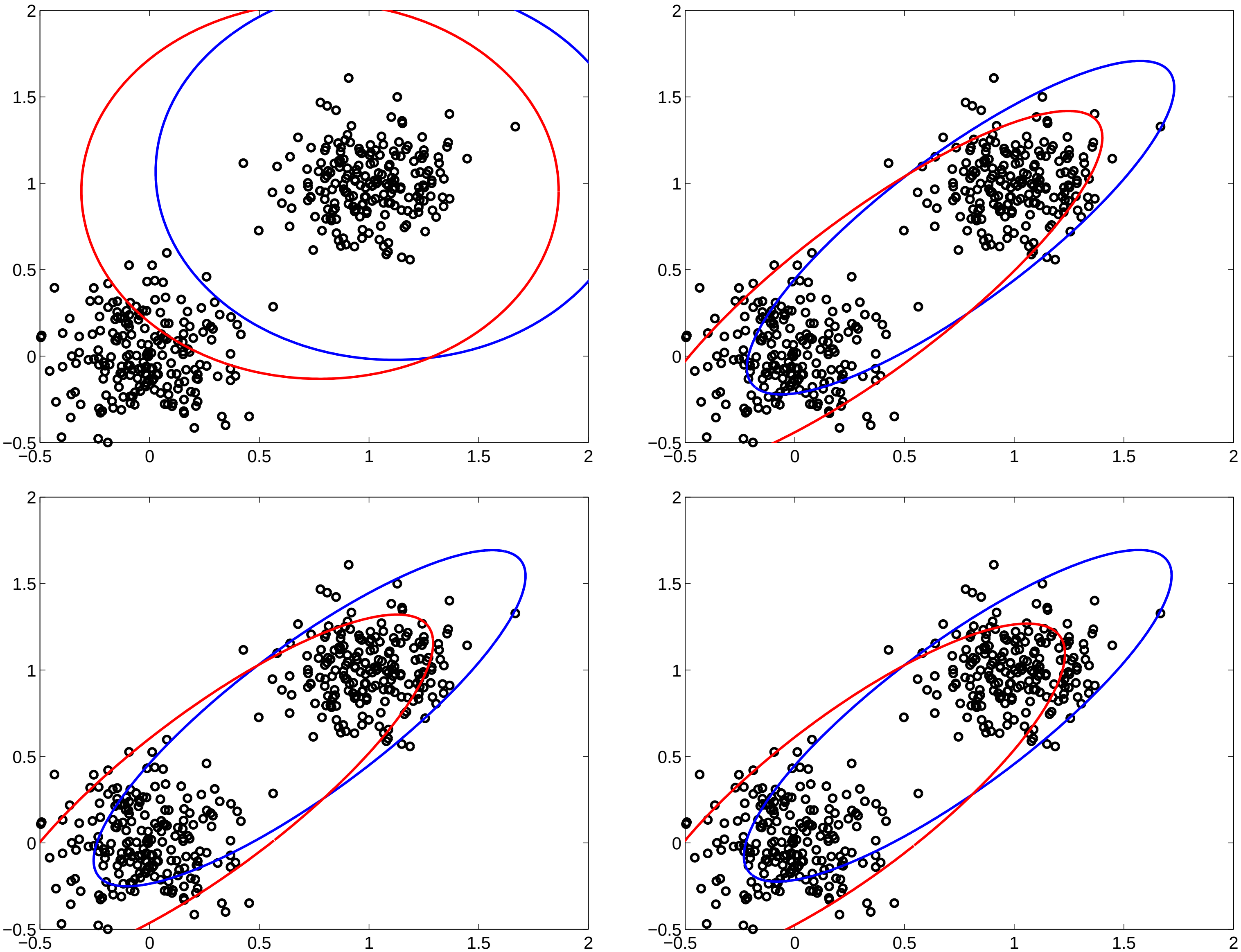
$$\hat{p}(j|i) \leftarrow P(y_i = j|\mathbf{x}_i, \theta), \text{ for all } j = 1,2 \text{ and } i = 1,\ldots,n$$

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

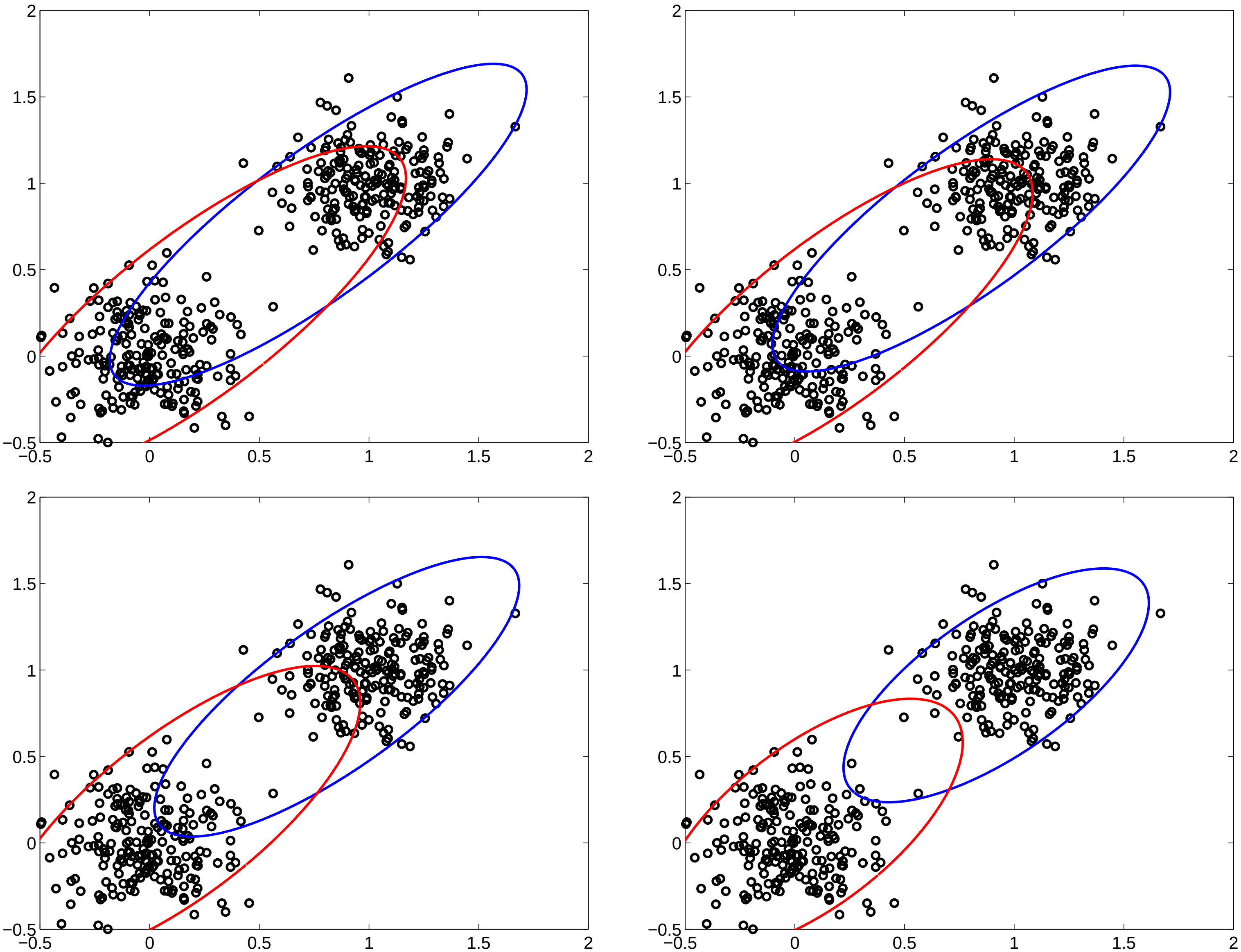
$$\begin{aligned}
\hat{n}_j &\leftarrow \sum_{i=1}^{n} \hat{p}(j|i) = \text{Soft \# of examples labeled } j \\
\hat{p}_j &\leftarrow \frac{\hat{n}_j}{n} \\
\hat{\mu}_j &\leftarrow \frac{1}{\hat{n}_j} \sum_{i=1}^{n} \hat{p}(j|i)\, \mathbf{x}_i \\
\hat{\Sigma}_j &\leftarrow \frac{1}{\hat{n}_j} \sum_{i=1}^{n} \hat{p}(j|i)\, (\mathbf{x}_i - \hat{\mu}_j)(\mathbf{x}_i - \hat{\mu}_j)^T
\end{aligned}$$

where $j = 1, 2$.

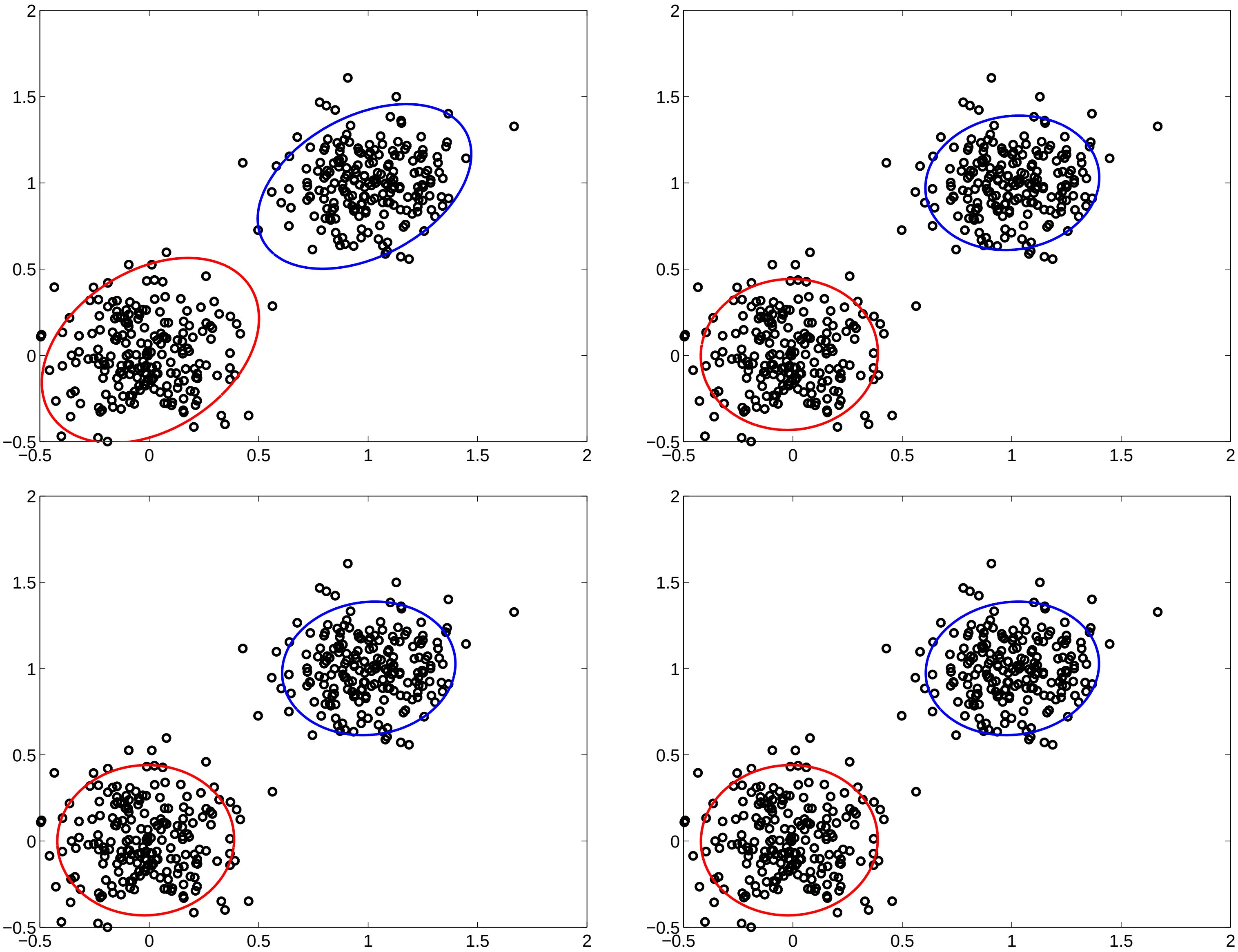
Mixture density estimation: example



Mixture density estimation



Mixture density estimation



The EM-algorithm

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

$$P(\text{ data } | \theta) = \prod_{i=1}^{n} \left[p_1 \, P(\mathbf{x}_i | \mu_1, \Sigma_1) + p_2 \, P(\mathbf{x}_i | \mu_2, \Sigma_2) \right]$$

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