Machine learning: lecture 12

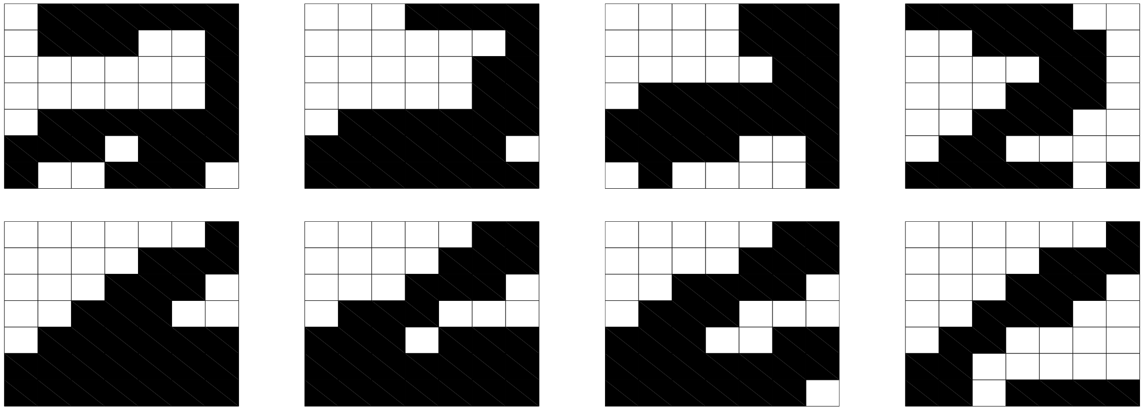
Tommi S. Jaakkola
MIT AI Lab
tommi@ai.mit.edu

Topics

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

Why density estimation

The digits again...



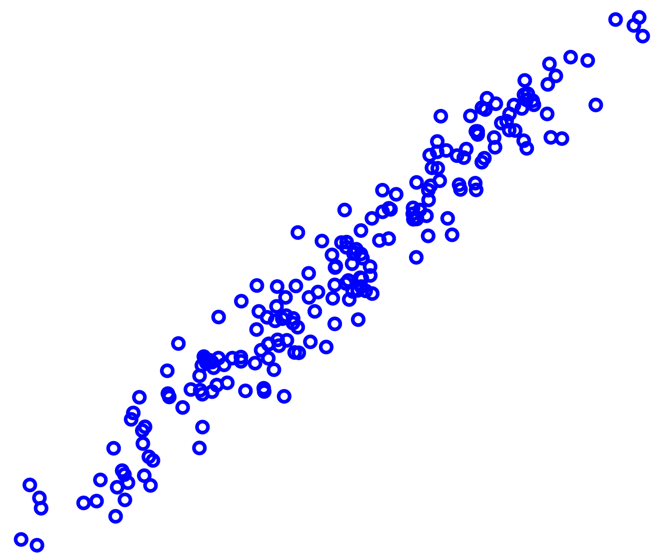
- Possible uses:
  - understanding the generation process of examples
  - clustering
  - classification via class-conditional densities
  - inference based on incomplete observations

Parametric density models

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

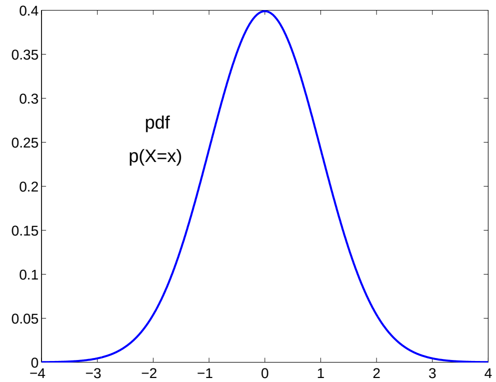
$$p(\mathbf{x}|\mu,\Sigma) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp\{ -\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu) \}$$

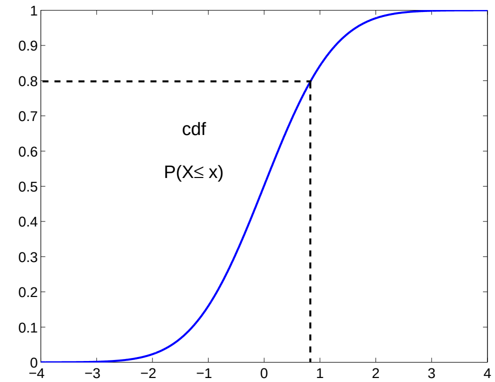
- This is a *generative model* in the sense that we can generate $\mathbf{x}$'s



Sampling from a Gaussian

- 1-dimensional Gaussian *probability density function* (pdf) $p(x|\mu, \sigma^2)$ and the corresponding *cumulative distribution function* (cdf) $F_{\mu,\sigma^2}(x)$





- To draw a sample from a Gaussian, we can invert the cumulative distribution function

$$F_{\mu,\sigma^2}(x) = \int_{-\infty}^{x} p(z|\mu, \sigma^2)dz$$

$$u \sim \text{Uniform}(0, 1) \Rightarrow x = F^{-1}_{\mu,\sigma^2}(u) \sim p(x|\mu, \sigma^2)$$

Multi-variate Gaussian samples

- A multivariate sample can be constructed from multiple independent one dimensional Gaussian samples:

$$
\begin{aligned}
z_i &\sim p(z_i|\mu = 0, \sigma^2 = 1), \;\; \mathbf{z} = [z_1, \ldots, z_d]^T \\
\mathbf{x} &= \Sigma^{1/2}\mathbf{z} + \mu
\end{aligned}
$$

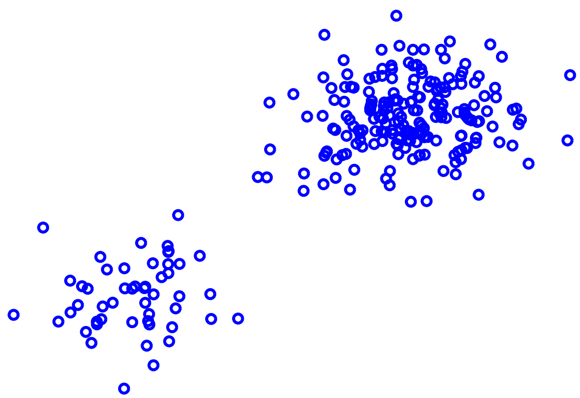
In this case $\mathbf{x} \sim p(\mathbf{x}|\mu, \Sigma)$.

Multi-variate density estimation

- A mixture of Gaussians model

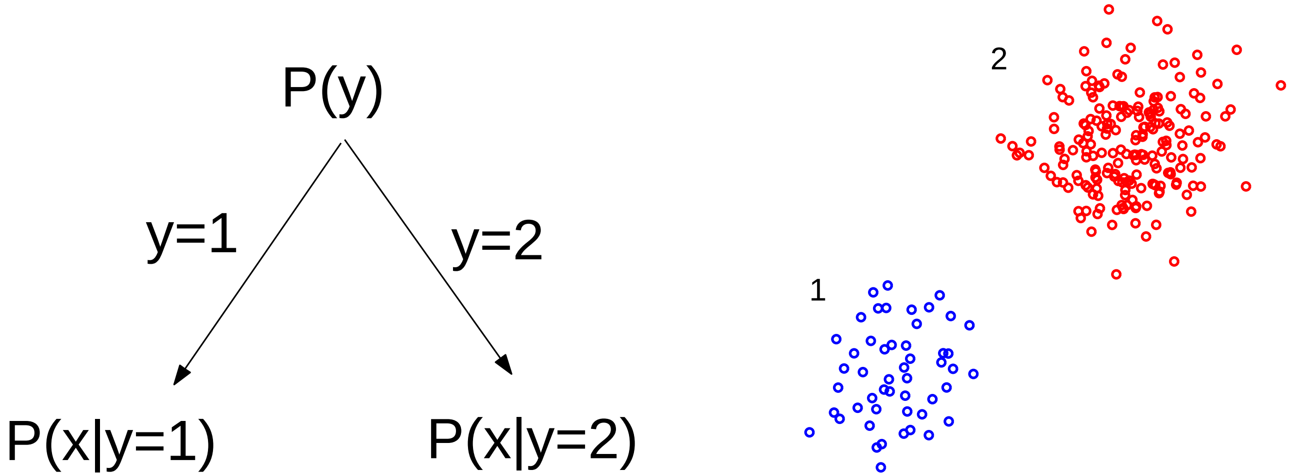
$$p(\mathbf{x}|\theta) = \sum_{i=1}^{k} p_j \, p(\mathbf{x}|\mu_j, \Sigma_j)$$

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

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

- This solves a *credit assignment* problem

Mixture density: posterior sampling

- Consider sampling $\mathbf{x}$ from the mixture density, then y from the posterior over the components given $\mathbf{x}$, and finally $\mathbf{x}'$ from the component density indicated by y:

$$\begin{aligned} \mathbf{x} &\sim p(\mathbf{x}|\theta) \\ y &\sim P(y|\mathbf{x}, \theta) \\ \mathbf{x}' &\sim p(\mathbf{x}'|y, \theta) \end{aligned}$$

Is y a fair sample from the prior distribution $P(y)$?

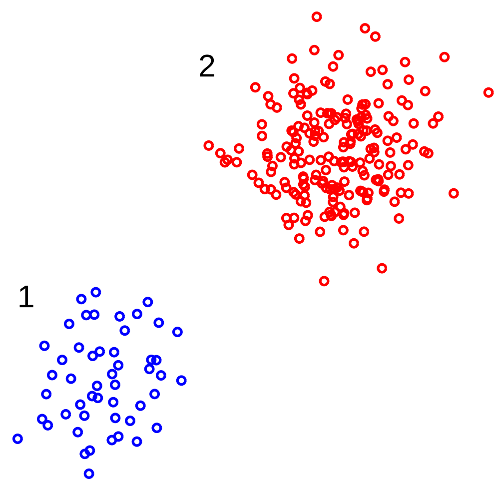
Is $\mathbf{x}'$ a fair sample from the mixture density $p(\mathbf{x}'|\theta)$?

Mixture density estimation

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

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

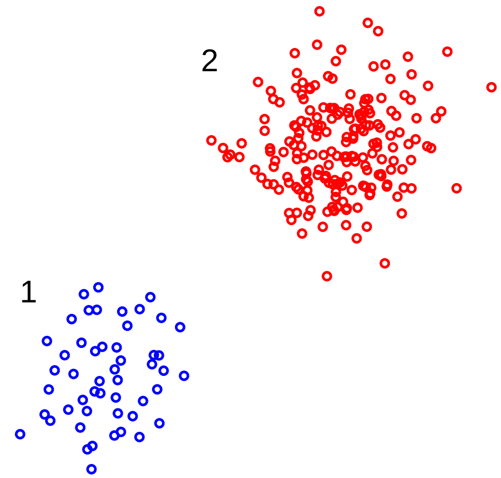
- If each example $\mathbf{x}_i$ in the training set were labeled $y_i = 1, 2$ according to which mixture component (1 or 2) had generated it, then the estimation would be easy.



- Labeled examples $\Rightarrow$ no credit assignment problem

Mixture density estimation

When examples are already assigned to mixture components (labeled), we can estimate each Gaussian independently



- If $\hat{n}_j$ is the number of examples labeled j, then for each $j = 1, 2$ we set

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

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

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

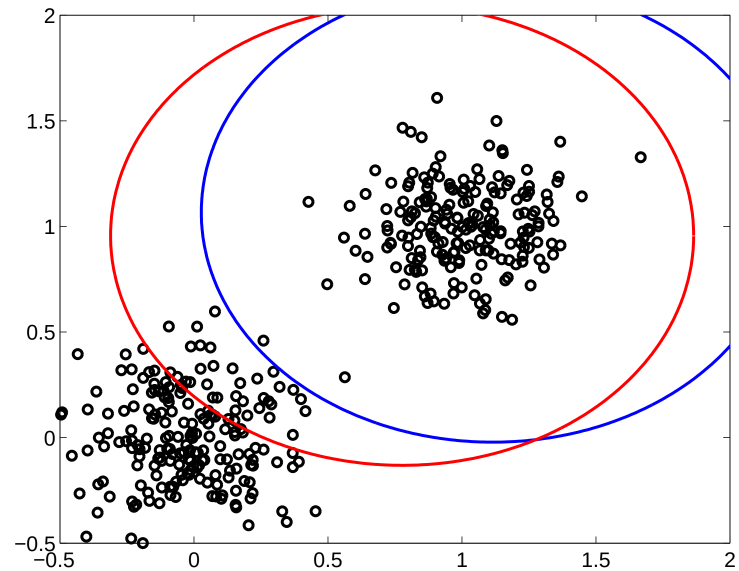
Mixture density estimation: credit assignment

- Of course we don't have such 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.

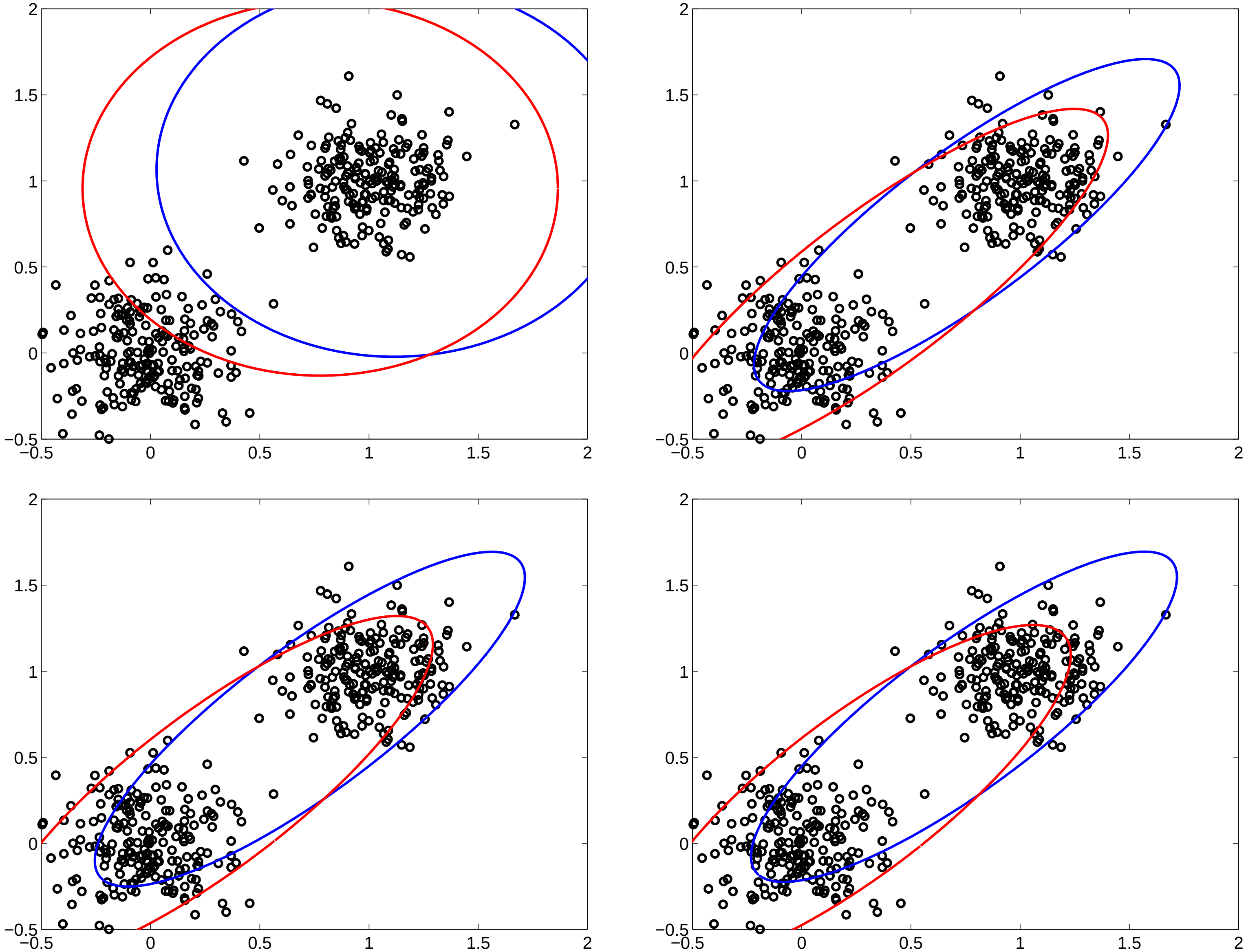
$$\hat{n}_j \leftarrow \sum_{i=1}^{n} \hat{p}(j|i) = \text{Soft \# of examples labeled } j$$

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

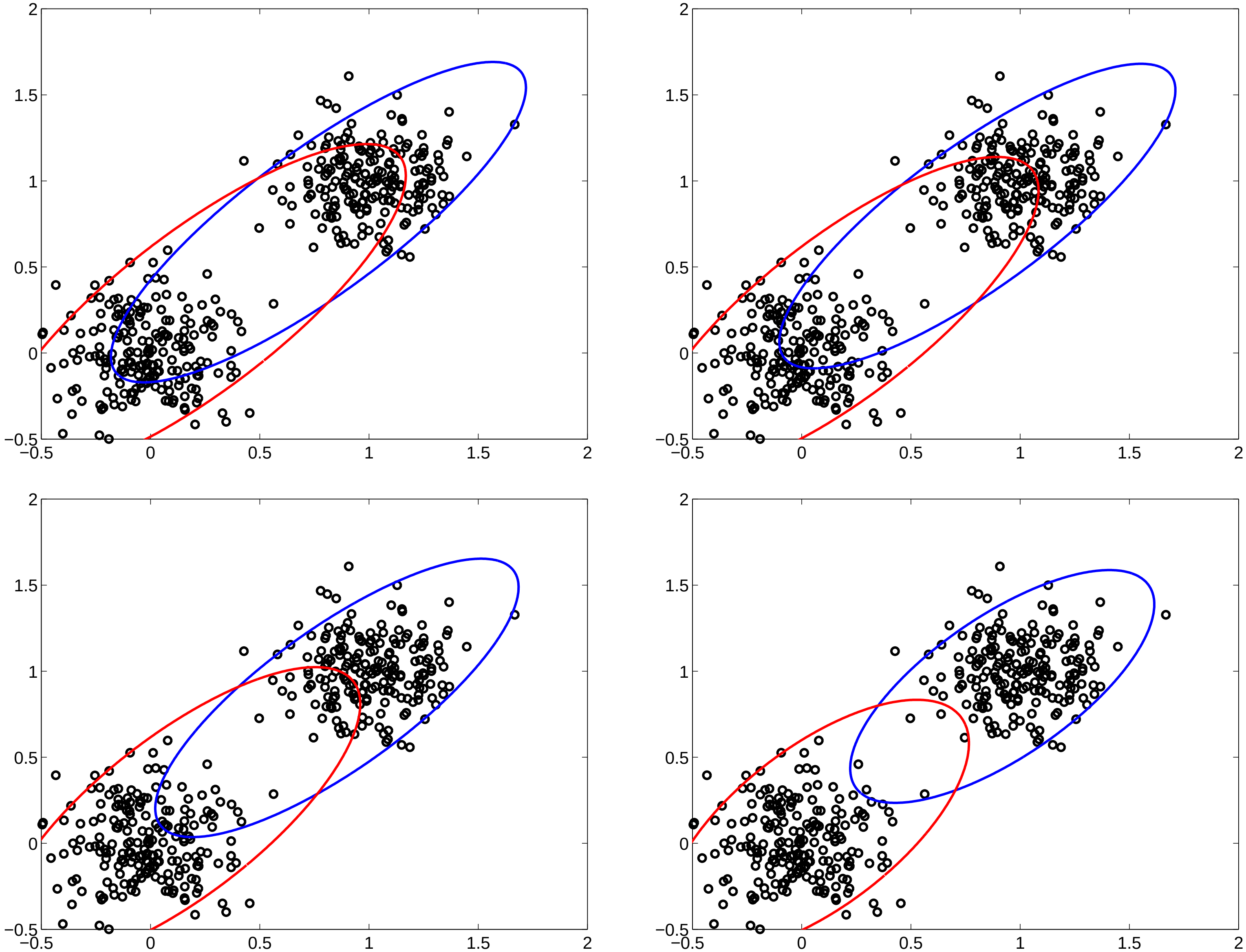
$$\hat{\mu}_j \leftarrow \frac{1}{\hat{n}_j} \sum_{i=1}^{n} \hat{p}(j|i)\, \mathbf{x}_i$$

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

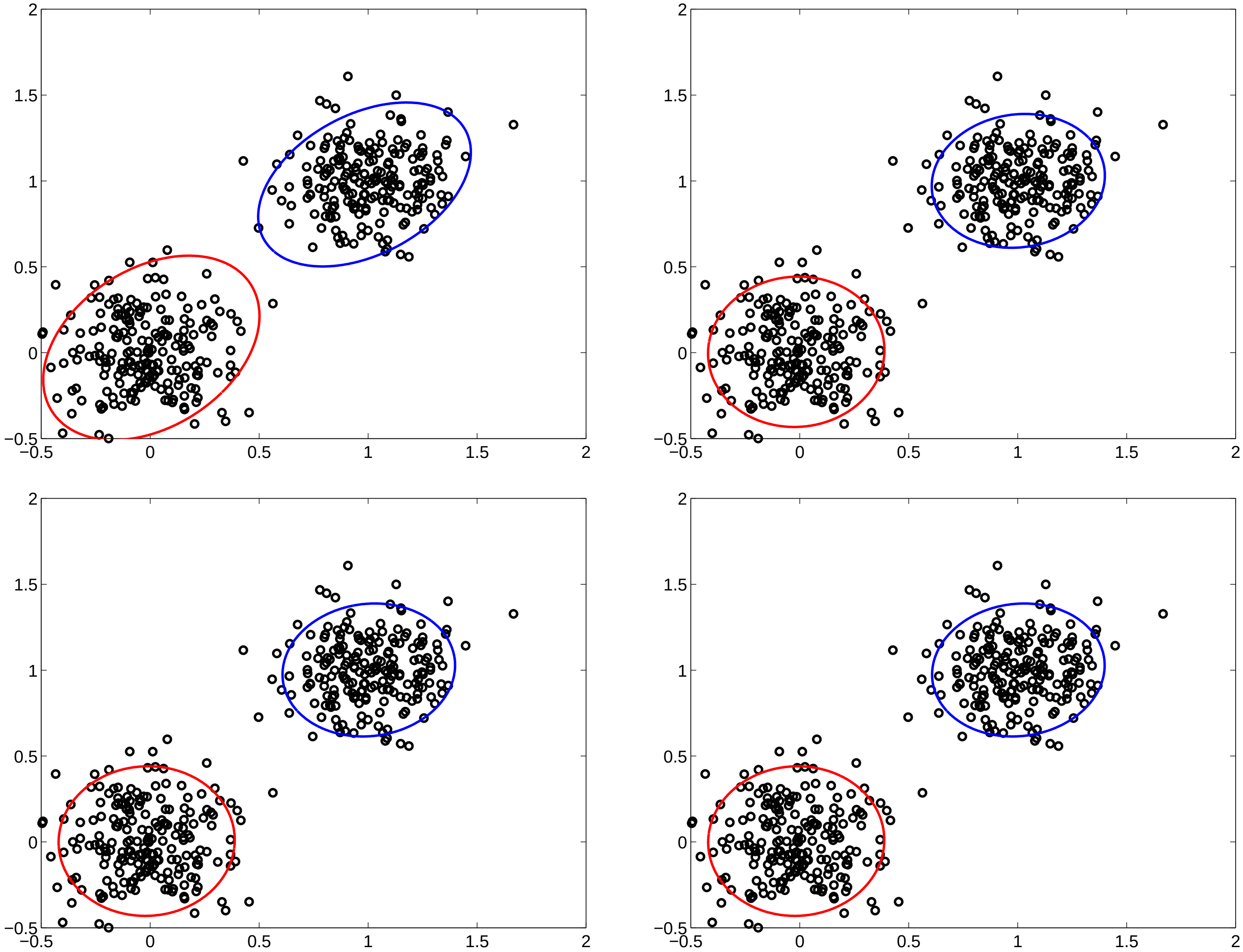
Mixture density estimation: example



Mixture density estimation



Mixture density estimation

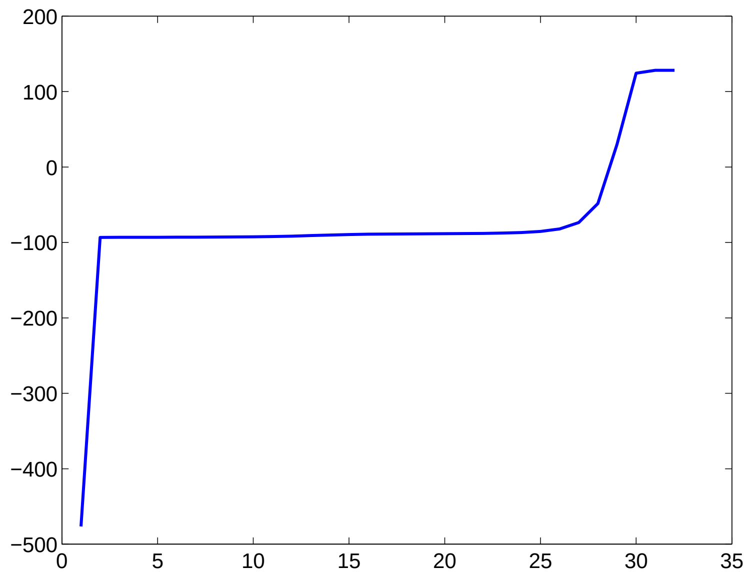


The EM-algorithm

- 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.



Demo

Classification example

- A digit recognition problem (8x8 binary digits)
  Training set $n = 100$ $(50$ examples of each digit).
  Test set $n = 400$ $(200$ examples of each digit).
- We estimate a mixture of Gaussians model separately for each type of digit

  Class 1: $P(\mathbf{x}|\theta_1)$, (e.g., a 3-component mixture density)

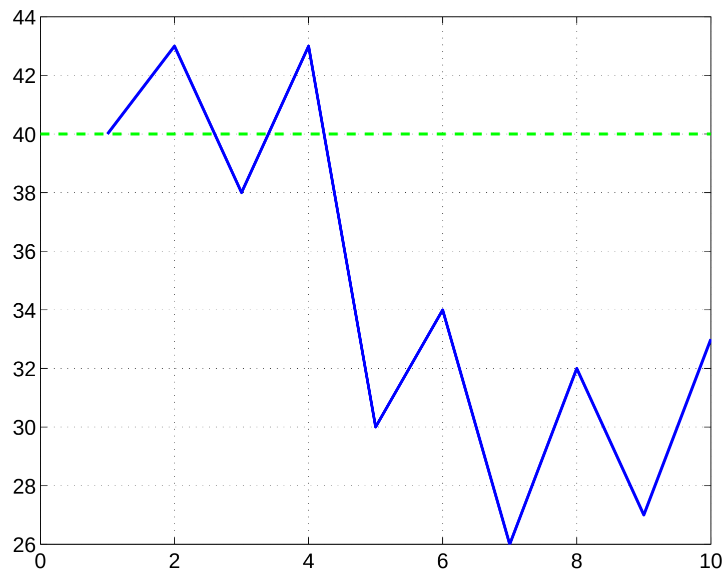
  Class 0: $P(\mathbf{x}|\theta_0)$, (e.g., a 3-component mixture density)

- Assuming the examples in each class are equally likely a priori, we will classify new examples $\mathbf{x}$ according to

  Class = 1 if $\log \frac{P(\mathbf{x}|\hat{\theta}_1)}{P(\mathbf{x}|\hat{\theta}_0)} > 0$ and Class = 0 otherwise

Classification example cont'd

- The figure gives the number of missclassified examples on the test set as a function of the number of mixture components in each class-conditional model



- Anything wrong with this figure?

Classification example cont'd

- A single covariance matrix has $64 * 65/2 = 2080$ parameters but we have only $n = 50$ training examples...

Classification example cont'd

- A single covariance matrix has $64 * 65/2 = 2080$ parameters but we have only $n = 50$ training examples...
- We can regularize the model by assigning a prior distribution over the parameters.

  We use a Wishart prior over each covariance matrix

$$P(\Sigma|S, n') \propto \frac{1}{|\Sigma|^{n'/2}} \exp\left(-\frac{n'}{2} \text{Trace}(\Sigma^{-1} S) \right)$$

  (written here in a bit non-standard way)

$$\begin{aligned} S &= \text{“prior” covariance matrix} \\ n' &= \text{equivalent sample size} \end{aligned}$$

Regularized EM

- E-step is unaffected (though the resulting values for the soft assignments will change)
- In the M-step we maximize instead a penalized log-likelihood of the (weighted) training set:

$$\sum_{i=1}^{n} \hat{P}(j|i) \, \log P(\mathbf{x}_i|\mu_j, \Sigma_j) + \log P(\Sigma_j|S, n')$$

  where j denotes the component (e.g., $j = 1, 2, 3$)

- Adding such a regularization penalty changes the resulting covariance estimate only slightly

$$\hat{\Sigma}_j \;\leftarrow\; \frac{1}{\hat{n}_j + n'} \left[\sum_{i=1}^{n} \hat{p}(j|i) \, (\mathbf{x}_i - \hat{\mu}_j)(\mathbf{x}_i - \hat{\mu}_j)^T + n'S \right]$$