6.891 Machine learning and neural networks

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

LECTURE 11 (October 17, 2000)

Topics

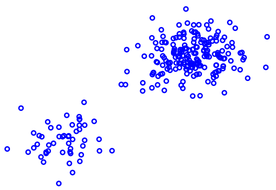
- Density estimation
  - Parametric, mixture models
  - Estimation via the EM algorithm
  - Mixture models in classification
  - K-means clustering
  - Non-parametric models

Parametric density models

- A mixture of Gaussians model

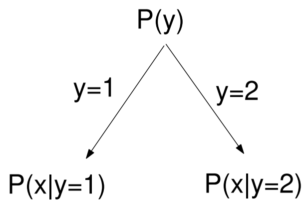
$$P(\mathbf{x}|\theta) = \sum_{i=1}^{k} p_j \, P(\mathbf{x}|\mu_j, \Sigma_j) \tag{1}$$

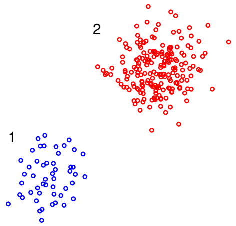
where $\theta = \{p_1, \ldots, p_k, \mu_1, \ldots, \mu_k, \Sigma_1, \ldots, \Sigma_k\}$ contains all the parameters of the mixture model. $\{p_j\}$ are known as *mixing proportions or coefficients*.



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.

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

- 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

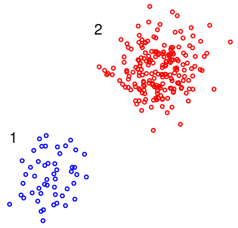
Mixture density estimation

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

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

$$P(\mathbf{x}|\theta) = p_1\, P(\mathbf{x}|\mu_1, \Sigma_1) + p_2\, P(\mathbf{x}|\mu_2, \Sigma_2) \quad (3)$$

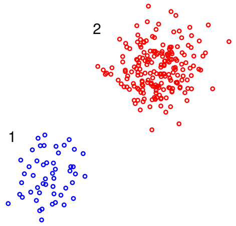
- If each example $\mathbf{x}_i$ in the training set were labeled $y_i = 1, 2$ according to which mixture component (1 or 2) generated it, then the estimation would be easy.



- 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 $j = 1, 2$

$$\hat{n}_j \leftarrow \sum_{i: y_i = j} 1 = \#\text{ of examples labeled } j \quad (4)$$

$$\hat{p}_j \leftarrow \frac{\hat{n}_j}{n} \quad (5)$$

$$\hat{\mu}_j \leftarrow \frac{1}{\hat{n}_j} \sum_{i: y_i = j} \mathbf{x}_i \quad (6)$$

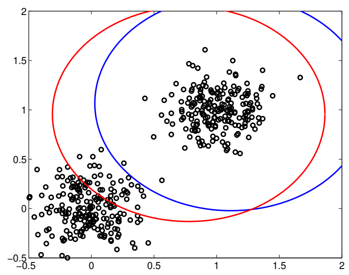
$$\hat{\Sigma}_j \leftarrow \frac{1}{\hat{n}_j} \sum_{i: y_i = j} (\mathbf{x}_i - \hat{\mu}_j)(\mathbf{x}_i - \hat{\mu}_j)^T \quad (7)$$

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:

$$\hat{p}(j|i) \leftarrow P(y_i = j|\mathbf{x}_i, \theta) \text{ for all } j = 1, 2 \text{ and } i = 1, \ldots, n \quad (8)$$

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

$$\hat{p}(j|i) \leftarrow P(y_i = j|\mathbf{x}_i, \theta), \quad \text{for all } j = 1,2 \text{ and } i = 1,\ldots,n \tag{9}$$

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

$$\hat{n}_j \leftarrow \sum_{i=1}^{n} \hat{p}(j|i) = \text{Soft \# of examples labeled } j \tag{10}$$

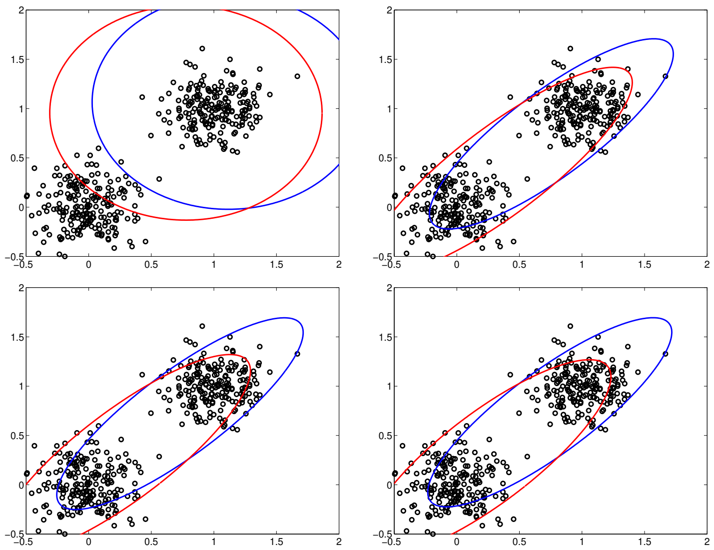
$$\hat{p}_j \leftarrow \frac{\hat{n}_j}{n} \tag{11}$$

$$\hat{\mu}_j \leftarrow \frac{1}{\hat{n}_j} \sum_{i=1}^{n} \hat{p}(j|i)\,\mathbf{x}_i \tag{12}$$

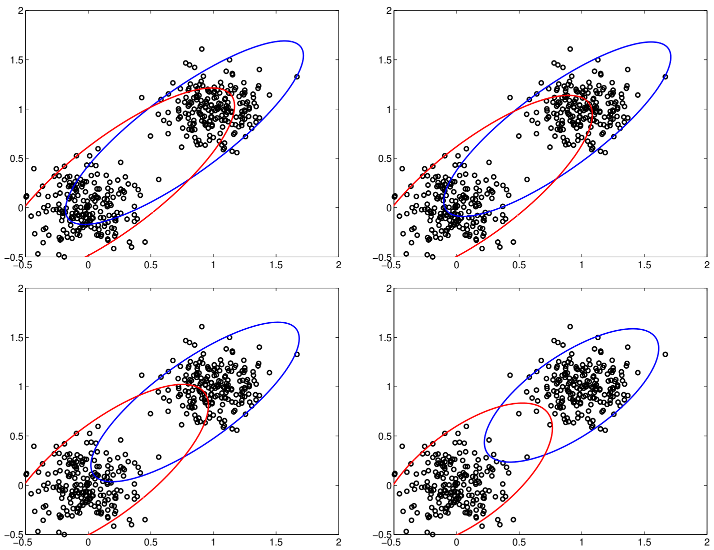
$$\hat{\Sigma}_j \leftarrow \frac{1}{\hat{n}_j} \sum_{i=1}^{n} \hat{p}(j|i)\,(\mathbf{x}_i - \hat{\mu}_j)(\mathbf{x}_i - \hat{\mu}_j)^T \tag{13}$$

where $j = 1,2$.

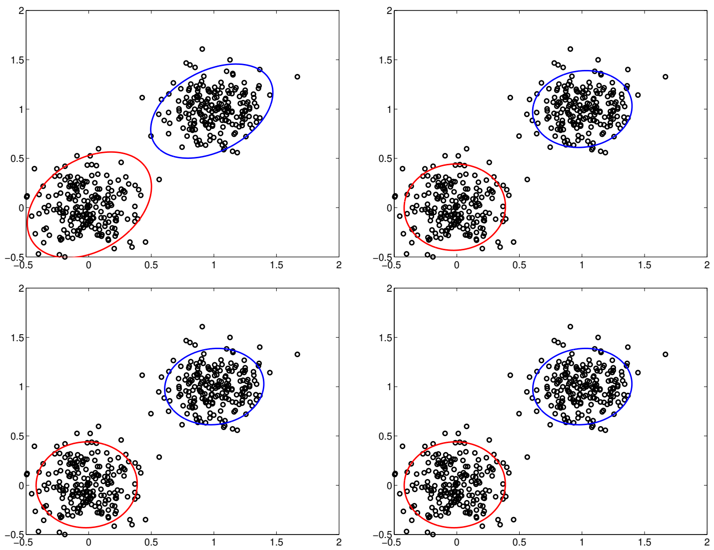
Mixture density estimation: example



Mixture density estimation



Mixture density estimation

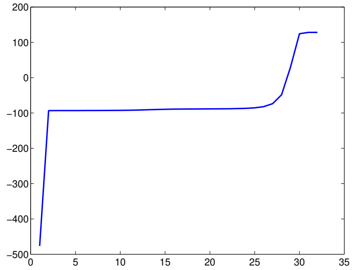


The EM-algorithm

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

$$P(\text{ data } | \theta) = \prod_{i=1}^{n} \left[p_1 \, P(\mathbf{x}_i | \mu_1, \Sigma_1) + p_2 \, P(\mathbf{x}_i | \mu_2, \Sigma_2) \right] \qquad (14)$$

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



Classification example

- Consider the digit recognition problem from problem sets #1 and #2.

  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 (with varying numbers of mixture components).

$$\text{Class 1: } P(\mathbf{x}|\hat{\theta}_1), \quad \text{Class 0: } P(\mathbf{x}|\hat{\theta}_0) \qquad (15)$$

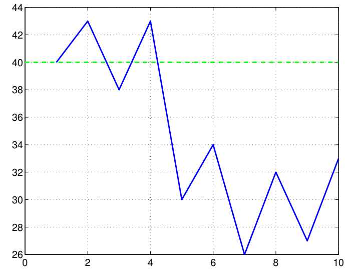
- Classification rule is based on the posterior class probability, or, equivalently, based on the log-likelihood ratio:

$$\text{Class} = 1 \text{ if } \quad \log \frac{P(\mathbf{x}|\hat{\theta}_1)}{P(\mathbf{x}|\hat{\theta}_0)} > 0 \quad \text{and Class} = 0 \text{ otherwise} \qquad (16)$$

(we are assuming that each digit is equally likely a priori)

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 the digit models



- What's wrong with this figure?

Classification example cont'd

- Question: How could we possibly get accurate estimates for even one covariance matrix ($64 * 65/2 = 2080$ parameters) based on only $n = 50$ training examples?
- Answer: regularization.

  We assign a prior distribution ($\sim$ Wishart) over each covariance matrix

$$P(\Sigma | S, n') \propto \frac{1}{|\Sigma|^{n'/2}} \exp\left(-\frac{n'}{2} \text{Trace}(\Sigma^{-1} S) \right) \qquad (17)$$

(written here in a bit non-standard way)

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

Classification example cont'd

- As a result, we maximize the penalized log-likelihood

$$J(\theta) = \sum_{i=1}^{n} \log\left[p_1\, P(\mathbf{x}_i|\mu_1,\Sigma_1) + p_2\, P(\mathbf{x}_i|\mu_2,\Sigma_2)\right] + \log P(\Sigma_1|S,n') + \log P(\Sigma_2|S,n') \quad (18)$$

- Adding such regularization changes the EM-algorithm only slightly:

- The E-step remains as before
- The M-step is modified only in terms of the covariance updates:

$$\hat{\Sigma}_j \leftarrow \frac{1}{\hat{n}_j + n'}\left[\sum_{i=1}^{n} \hat{p}(j|i)\,(\mathbf{x}_i - \hat{\mu}_j)(\mathbf{x}_i - \hat{\mu}_j)^T + n'S\right] \quad (19)$$

for $j = 1, 2$.

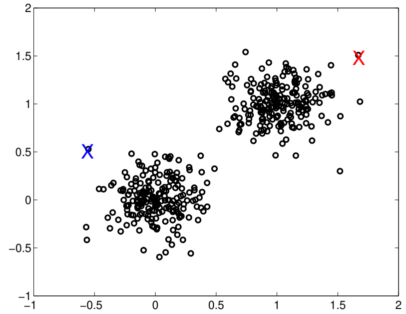
K-means clustering

- A simpler (hard assignment) version of the EM algorithm for mixture models

  Initialize: pick k random points as cluster centers

  "E-step:" reallocate each point to its closest cluster center

  "M-step:" recompute the cluster centers on the basis of the allocated points (sample mean)



K-means clustering cont'd

\+ Computationally simple

\+ Fast convergence

\+ Approximates a mixture model with k components

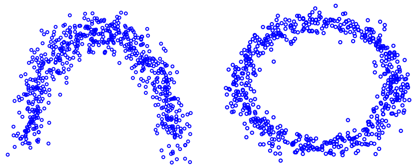
– accurately approximates only spherical Gaussians in a mixture model (no elongated clusters)

– susceptible to locally optimal solutions

– does not provide a generative probability density (unless reinterpreted post hoc)

Beyond parametric density models

- More mixture densities



- We can approximate almost any distribution by including more and more components in the mixture model

$$P(\mathbf{x}|\theta) = \sum_{j=1}^{k} p_j\, P(\mathbf{x}|\mu_j, \Sigma_j) \quad (20)$$

Non-parametric densities

- We can assign one mixture component for each example (Parzen window):

$$\hat{P}_{parzen}(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} P(\mathbf{x}|\mathbf{x}_i, \sigma^2 I) \tag{21}$$

  where n is the number of examples.

- The covariance matrices for the bumps $\Sigma_i = \sigma^2 \cdot I$ are all equal and spherical. The single parameter σ^2 now controls the smoothness of the density estimate