Machine learning: lecture 14

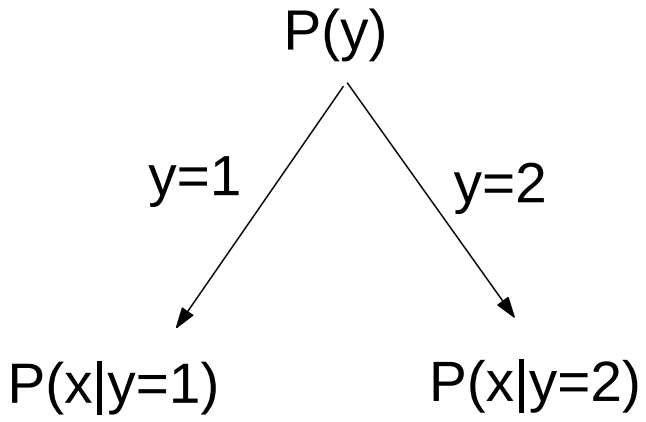
Tommi S. Jaakkola
MIT CSAIL
tommi@csail.mit.edu

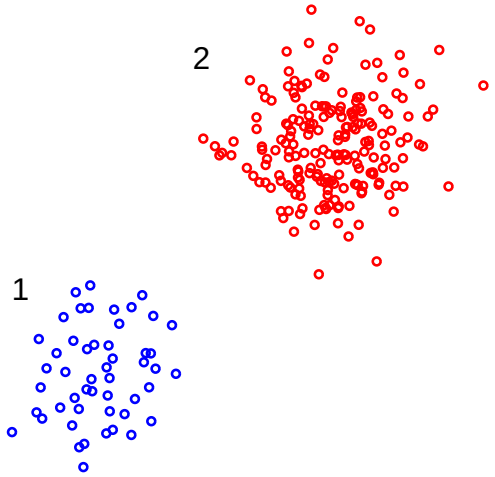
Topics

- Probabilistic modeling
  - mixture models and the EM algorithm
  - mixtures of experts, examples
  - hierarchical models

Review: mixture density

- Data generation process:





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

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

Review: the EM algorithm

E-step: softly assign examples to mixture components

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

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

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

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

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

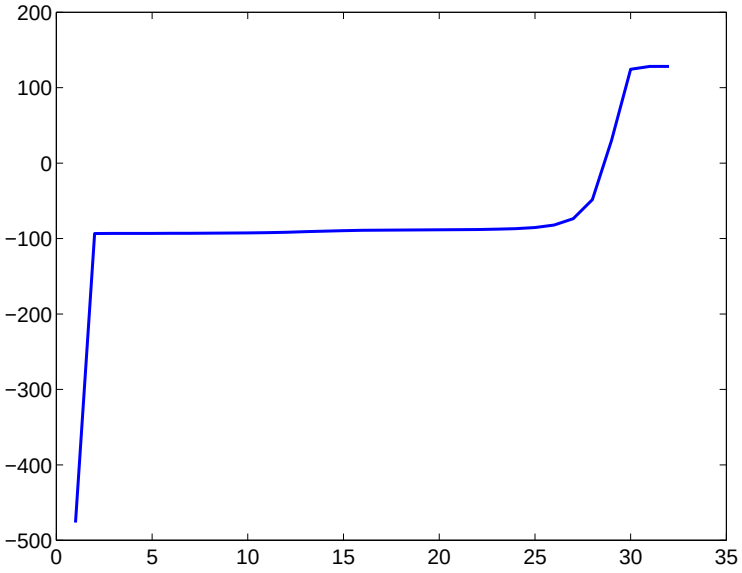
Mixture demo

The EM-algorithm

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

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

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



The EM algorithm: why does it work?

- To show that EM increases the log-likelihood of the data after each iteration we resort to the following *auxiliary* objective

$$J(q;\theta) = \sum_i \sum_{j=1,2} q(j|i) \log \left[\frac{p_j p(\mathbf{x}_i|\mu_j, \Sigma_j)}{q(j|i)} \right]$$

  where $q(j|i)$ are soft assignments of examples to mixture components. We view both q and θ as parameters to be optimized.

- We need to specify
  - how we will use this objective (algorithm)
  - how the algorithm relates to EM
  - how the objective relates to the log-likelihood

The auxiliary objective: algorithm

$$J(q;\theta) = \sum_i \sum_{j=1,2} q(j|i) \log \left[\frac{p_j p(\mathbf{x}_i|\mu_j, \Sigma_j)}{q(j|i)} \right]$$

- We maximize this objective in the following alternating steps. Let $\theta^{(0)}$ be the initial setting of the parameters, then at iteration k:

$$\begin{aligned} q^{(k)} &= \arg\max_q J(q;\theta^{(k)}) \quad \text{(E-step)} \\ \theta^{(k+1)} &= \arg\max_\theta J(q^{(k)};\theta) \quad \text{(M-step)} \end{aligned}$$

The auxiliary objective: algorithm

$$J(q;\theta) = \sum_i \sum_{j=1,2} q(j|i) \log \left[\frac{p_j p(\mathbf{x}_i|\mu_j, \Sigma_j)}{q(j|i)} \right]$$

- We maximize this objective in the following alternating steps. Let $\theta^{(0)}$ be the initial setting of the parameters, then at iteration k:

$$q^{(k)} = \arg\max_q J(q;\theta^{(k)}) \quad \text{(E-step)}$$

$$\theta^{(k+1)} = \arg\max_\theta J(q^{(k)};\theta) \quad \text{(M-step)}$$

- Clearly these steps lead to a monotonically increasing value of the objective $J(q;\theta)$

$$J(q^{(k-1)};\theta^{(k)}) \leq J(q^{(k)};\theta^{(k)}) \leq J(q^{(k)};\theta^{(k+1)})$$

Relation to EM

- Suppose we set $q(j|i)$ to be the posterior assignments $\hat{p}(j|i)$ based on the current setting of the parameters θ:

$$J(\hat{p};\theta) = \sum_i \sum_{j=1,2} \hat{p}(j|i) \log \left[\frac{p_j p(\mathbf{x}_i|\mu_j, \Sigma_j)}{\hat{p}(j|i)} \right]$$

Relation to EM

- Suppose we set $q(j|i)$ to be the posterior assignments $\hat{p}(j|i)$ based on the current setting of the parameters θ:

$$
\begin{aligned}
J(\hat{p};\theta) &= \sum_i \sum_{j=1,2} \hat{p}(j|i) \log \left[\frac{p_j p(\mathbf{x}_i|\mu_j,\Sigma_j)}{\hat{p}(j|i)} \right] \\
&= \sum_i \sum_{j=1,2} \hat{p}(j|i) \log \left[p_j p(\mathbf{x}_i|\mu_j,\Sigma_j) \right] + \text{const.}
\end{aligned}
$$

Relation to EM

- Suppose we set $q(j|i)$ to be the posterior assignments $\hat{p}(j|i)$ based on the current setting of the parameters θ:

$$
\begin{aligned}
J(\hat{p};\theta) &= \sum_i \sum_{j=1,2} \hat{p}(j|i) \log \left[\frac{p_j p(\mathbf{x}_i|\mu_j,\Sigma_j)}{\hat{p}(j|i)} \right] \\
&= \sum_i \sum_{j=1,2} \hat{p}(j|i) \log \left[p_j p(\mathbf{x}_i|\mu_j,\Sigma_j) \right] + \text{const.} \\
&= \sum_{j=1,2} \left[\sum_i \hat{p}(j|i) \log \left[p_j p(\mathbf{x}_i|\mu_j,\Sigma_j) \right] \right] + \text{const.}
\end{aligned}
$$

Relation to EM

- Suppose we set $q(j|i)$ to be the posterior assignments $\hat{p}(j|i)$ based on the current setting of the parameters θ:

$$
\begin{aligned}
J(\hat{p};\theta) &= \sum_i \sum_{j=1,2} \hat{p}(j|i) \log \left[\frac{p_j p(\mathbf{x}_i|\mu_j,\Sigma_j)}{\hat{p}(j|i)} \right] \\
&= \sum_i \sum_{j=1,2} \hat{p}(j|i) \log \left[p_j p(\mathbf{x}_i|\mu_j,\Sigma_j) \right] + \text{const.} \\
&= \sum_{j=1,2} \underbrace{\left[\sum_i \hat{p}(j|i) \log \left[p_j p(\mathbf{x}_i|\mu_j,\Sigma_j) \right] \right]}_{\text{weighted log-likelihood}} + \text{const.}
\end{aligned}
$$

The estimation of θ reduces to finding ML Gaussians based on a weighted training set, separately for each component

$\Rightarrow$ M-step of the EM-algorithm.

Relation to the log-likelihood

$$J(q;\theta) = \sum_i \sum_{j=1,2} q(j|i) \log \left[\frac{p_j p(\mathbf{x}_i|\mu_j, \Sigma_j)}{q(j|i)} \right]$$

- This objective has two relevant properties for our purposes:

$$J(q;\theta) \leq l(D;\theta) \quad (1)$$

$$J(\hat{p};\theta) = l(D;\theta) \quad (2)$$

where $\hat{p}(j|i) = P(j|\mathbf{x}_i, \theta)$ are the posterior assignments.

Proof of property 1

For any setting of $q(j|i)$

$$J(q;\theta) = \sum_i \sum_{j=1,2} q(j|i) \log \left[\frac{p_j p(\mathbf{x}_i|\mu_j, \Sigma_j)}{q(j|i)} \right]$$

Proof of property 1

For any setting of $q(j|i)$

$$
\begin{aligned}
J(q;\theta) &= \sum_i \sum_{j=1,2} q(j|i) \log \left[\frac{p_j p(\mathbf{x}_i|\mu_j, \Sigma_j)}{q(j|i)} \right] \\
&\leq \sum_i \log \left[\sum_{j=1,2} q(j|i) \frac{p_j p(\mathbf{x}_i|\mu_j, \Sigma_j)}{q(j|i)} \right]
\end{aligned}
$$

Proof of property 1

For any setting of $q(j|i)$

$$
\begin{aligned}
J(q;\theta) &= \sum_i \sum_{j=1,2} q(j|i) \log \left[\frac{p_j p(\mathbf{x}_i|\mu_j, \Sigma_j)}{q(j|i)} \right] \\
&\leq \sum_i \log \left[\sum_{j=1,2} q(j|i) \frac{p_j p(\mathbf{x}_i|\mu_j, \Sigma_j)}{q(j|i)} \right] \\
&= \sum_i \log \left[\sum_{j=1,2} p_j p(\mathbf{x}_i|\mu_j, \Sigma_j) \right]
\end{aligned}
$$

Proof of property 1

For any setting of $q(j|i)$

$$
\begin{aligned}
J(q;\theta) &= \sum_i \sum_{j=1,2} q(j|i) \log \left[\frac{p_j p(\mathbf{x}_i|\mu_j, \Sigma_j)}{q(j|i)} \right] \\
&\leq \sum_i \log \left[\sum_{j=1,2} q(j|i) \frac{p_j p(\mathbf{x}_i|\mu_j, \Sigma_j)}{q(j|i)} \right] \\
&= \sum_i \log \left[\sum_{j=1,2} p_j p(\mathbf{x}_i|\mu_j, \Sigma_j) \right] \\
&= \sum_i \log p(\mathbf{x}_i|\theta) = l(D;\theta)
\end{aligned}
$$

Proof of property 2

By setting $q(j|i)$ equal to the posterior assignments

$$\hat{p}(j|i) = \frac{p_j p(\mathbf{x}_i|\mu_j, \Sigma_j)}{\sum_{j'=1,2} p_{j'} p(\mathbf{x}_i|\mu_{j'}, \Sigma_{j'})} = \frac{p_j p(\mathbf{x}_i|\mu_j, \Sigma_j)}{p(\mathbf{x}_i|\theta)}$$

we get

$$J(\hat{p}; \theta) = \sum_i \sum_{j=1,2} \hat{p}(j|i) \log \left[\frac{p_j p(\mathbf{x}_i|\mu_j, \Sigma_j)}{\hat{p}(j|i)} \right]$$

Proof of property 2

By setting $q(j|i)$ equal to the posterior assignments

$$\hat{p}(j|i) = \frac{p_j p(\mathbf{x}_i|\mu_j, \Sigma_j)}{\sum_{j'=1,2} p_{j'} p(\mathbf{x}_i|\mu_{j'}, \Sigma_{j'})} = \frac{p_j p(\mathbf{x}_i|\mu_j, \Sigma_j)}{p(\mathbf{x}_i|\theta)}$$

we get

$$\begin{aligned} J(\hat{p};\theta) &= \sum_i \sum_{j=1,2} \hat{p}(j|i) \log \left[\frac{p_j p(\mathbf{x}_i|\mu_j, \Sigma_j)}{\hat{p}(j|i)} \right] \\ &= \sum_i \sum_{j=1,2} \hat{p}(j|i) \log \left[p(\mathbf{x}_i|\theta) \right] \end{aligned}$$

Proof of property 2

By setting $q(j|i)$ equal to the posterior assignments

$$\hat{p}(j|i) = \frac{p_j p(\mathbf{x}_i|\mu_j, \Sigma_j)}{\sum_{j'=1,2} p_{j'} p(\mathbf{x}_i|\mu_{j'}, \Sigma_{j'})} = \frac{p_j p(\mathbf{x}_i|\mu_j, \Sigma_j)}{p(\mathbf{x}_i|\theta)}$$

we get

$$\begin{aligned} J(\hat{p};\theta) &= \sum_i \sum_{j=1,2} \hat{p}(j|i) \log \left[\frac{p_j p(\mathbf{x}_i|\mu_j, \Sigma_j)}{\hat{p}(j|i)} \right] \\ &= \sum_i \sum_{j=1,2} \hat{p}(j|i) \log \left[p(\mathbf{x}_i|\theta) \right] \\ &= \sum_i \log p(\mathbf{x}_i|\theta) = l(D;\theta) \end{aligned}$$

Completing the argument

- The algorithm and properties

$$
\begin{aligned}
q^{(k)} &= \arg\max_{q} J(q; \theta^{(k)}) \quad \text{(E-step)} \\
\theta^{(k+1)} &= \arg\max_{\theta} J(q^{(k)}; \theta) \quad \text{(M-step)} \\
J(q; \theta) &\leq l(D; \theta) \quad \text{(property 1)} \\
J(\hat{p}; \theta) &= l(D; \theta) \quad \text{(property 2)}
\end{aligned}
$$

- We can now set up the following chain of inequalities

$$
l(D; \theta^{(k)}) = J(q^{(k)}; \theta^{(k)}) \quad \text{(E-step, prop. 2)}
$$

Completing the argument

- The algorithm and properties

$$
\begin{aligned}
q^{(k)} &= \arg\max_{q} J(q; \theta^{(k)}) \quad \text{(E-step)} \\
\theta^{(k+1)} &= \arg\max_{\theta} J(q^{(k)}; \theta) \quad \text{(M-step)} \\
J(q; \theta) &\leq l(D; \theta) \quad \text{(property 1)} \\
J(\hat{p}; \theta) &= l(D; \theta) \quad \text{(property 2)}
\end{aligned}
$$

- We can now set up the following chain of inequalities

$$
\begin{aligned}
l(D; \theta^{(k)}) &= J(q^{(k)}; \theta^{(k)}) \quad \text{(E-step, prop. 2)} \\
&\leq J(q^{(k)}; \theta^{(k+1)}) \quad \text{(M-step)}
\end{aligned}
$$

Completing the argument

- The algorithm and properties

$$
\begin{aligned}
q^{(k)} &= \arg\max_{q} J(q; \theta^{(k)}) \quad \text{(E-step)} \\
\theta^{(k+1)} &= \arg\max_{\theta} J(q^{(k)}; \theta) \quad \text{(M-step)} \\
J(q; \theta) &\leq l(D; \theta) \quad \text{(property 1)} \\
J(\hat{p}; \theta) &= l(D; \theta) \quad \text{(property 2)}
\end{aligned}
$$

- We can now set up the following chain of inequalities

$$
\begin{aligned}
l(D; \theta^{(k)}) &= J(q^{(k)}; \theta^{(k)}) && \text{(E-step, prop. 2)} \\
&\leq J(q^{(k)}; \theta^{(k+1)}) && \text{(M-step)} \\
&\leq J(q^{(k+1)}; \theta^{(k+1)}) && \text{(E-step)}
\end{aligned}
$$

Completing the argument

- The algorithm and properties

$$
\begin{aligned}
q^{(k)} &= \arg\max_q J(q;\theta^{(k)}) && \text{(E-step)} \\
\theta^{(k+1)} &= \arg\max_\theta J(q^{(k)};\theta) && \text{(M-step)} \\
J(q;\theta) &\leq l(D;\theta) && \text{(property 1)} \\
J(\hat{p};\theta) &= l(D;\theta) && \text{(property 2)}
\end{aligned}
$$

- We can now set up the following chain of inequalities

$$
\begin{aligned}
l(D;\theta^{(k)}) &= J(q^{(k)};\theta^{(k)}) && \text{(E-step, prop. 2)} \\
&\leq J(q^{(k)};\theta^{(k+1)}) && \text{(M-step)} \\
&\leq J(q^{(k+1)};\theta^{(k+1)}) && \text{(E-step)} \\
&= l(D;\theta^{(k+1)}) && \text{(prop. 2)}
\end{aligned}
$$

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)

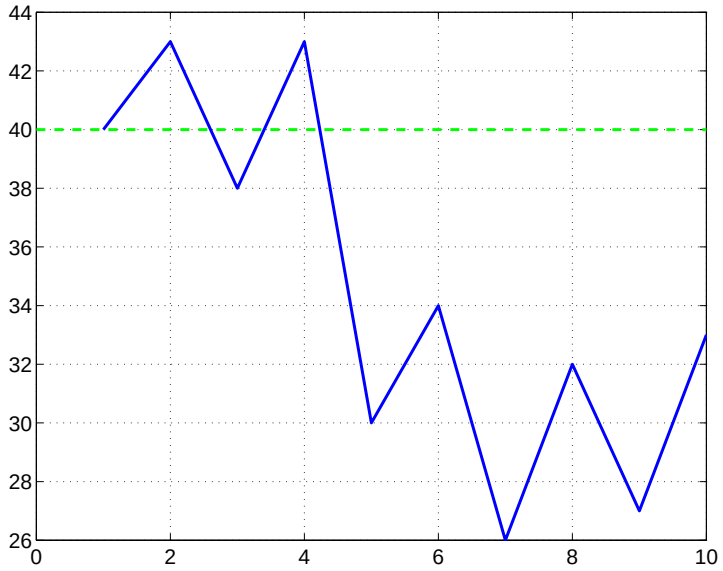
  Class 0: $P(\mathbf{x}|\theta_0)$, (e.g., a 3-component mixture density)

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

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

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, especially the covariance matrices

  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: for each component j

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

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

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

Regularized EM: demo