Topics

- A bit more general view of the EM algorithm
  - regularized mixtures

- Extensions of mixture models
  - hierarchical mixture models
  - conditional mixture models: mixtures of experts
Mixture models: review

- A two component Gaussian mixture model:

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

where \( \theta = \{p_1, p_2, \mu_1, \mu_2, \Sigma_1, \Sigma_2\} \).

- Only iterative solutions are available for finding the parameters that maximize the log-likelihood

\[ l(\theta; D) = \sum_{i=1}^{n} \log p(x_i|\theta) \]

where \( D = \{x_1, \ldots, x_n\} \).

The estimation involves resolving which mixture component should be responsible for which data point.
The EM algorithm

- The EM-algorithm finds a local maximum of \( l(\theta; D) \)

**E-step:** evaluate the expected complete log-likelihood

\[
J(\theta; \theta^{(t)}) = \sum_{i=1}^{n} E_{j \sim P(j|x_i, \theta^{(t)})} \log \left( p_j p(x_i | \mu_j, \Sigma_j) \right)
\]

\[
= \sum_{i=1}^{n} \sum_{j=1,2} P(j|x_i, \theta^{(t)}) \log \left( p_j p(x_i | \mu_j, \Sigma_j) \right)
\]

**M-step:** find the new parameters by maximizing the expected complete log-likelihood

\[
\theta^{(t+1)} \leftarrow \arg \max_{\theta} J(\theta; \theta^{(t)})
\]
Regularized EM algorithm

- To maximize a penalized (regularized) log-likelihood

\[
l'(\theta; D) = \sum_{i=1}^{n} \log p(x_i|\theta) + \log p(\theta)
\]

we only need to modify the M-step of the EM-algorithm. Specifically, in the M-step, we find \( \theta \) that maximize a penalized expected complete log-likelihood:

\[
J(\theta; \theta^{(t)}) = \sum_{i=1}^{n} E_{j \sim P(j|x_i,\theta^{(t)})} \log \left( p_j p(x_i|\mu_j, \Sigma_j) \right)
+ \log p(p_1, p_2) + \log p(\Sigma_1) + \log p(\Sigma_1)
\]

where, for example, \( p(p_1, p_2) \) could be a Dirichlet and each \( p(\Sigma_j) \) a Wishart prior.
Regularized EM: demo
Selecting the number of components

As a simple strategy for selecting the appropriate number of mixture components, we can find $k$ that minimize the following asymptotic approximation to the description length:

$$DL \approx - \log p(\text{data}|\hat{\theta}_k) + \frac{d_k}{2} \log(n)$$

where $n$ is the number of training points, $\hat{\theta}_k$ is the maximum likelihood parameter estimate for the $k$-component mixture, and $d_k$ is the (effective) number of parameters in the $k$-mixture.
Extensions: hierarchical mixture models

- We have already used hierarchical mixture models in the digit classification problem.

Data generation model:

\[
P(y=1) P(y=0) \\
P(c=1|y=1) P(c=3|y=0) \\
P(x|y=0,c=3)P(x|y=1,c=1) \ldots \ldots
\]

It is not necessary for the top level division to be “observable” as it is in this classification context.
Hierarchical mixture models cont’d

To estimate such hierarchical models from data, we have to resolve which leaf (path) in the tree is responsible for generating which data point.

Only the E-step needs to be revised: the expectation over assignments is now taken with respect to

\[
P(y = j, c = k|\mathbf{x}) = P(y = j|\mathbf{x}) P(c = k|y = j, \mathbf{x}),
\]

For example, for a hierarchical mixture of Gaussians, we evaluate

\[
J(\theta; \theta^{(t)}) = \sum_{i=1}^{n} E_{(j,k) \sim P(j,k|x_i,\theta^{(t)})} \log \left( p_j p_{k|j} p(\mathbf{x}_i|\mu_{j,k}, \Sigma_{j,k}) \right)
\]

where \( p_j \) and \( p_{k|j} \) are the prior selection probabilities.
Hierarchical mixture models cont’d

- Arranging the mixture components into a hierarchy is useful only with additional “topological” constraints. The hierarchical mixture (as stated previously) is otherwise equivalent to a flat mixture.

To adequately reveal any hierarchical organization in the data, we have to prime the model to find such structure.

- initialize parameters similarly within branches
- tying parameters, etc.
Conditional mixtures: mixtures of experts

- Many regression or classification problems can be decomposed into smaller (easier) sub problems

- Examples:
  1. Dealing with style in handwritten character recognition
  2. Dealing with dialect/accent in speech recognition etc.

- Each sub-problem could be solved by a specific “expert”

- Unlike in ordinary mixtures, the selection of which expert to rely on must depend on the context (i.e., the input $x$)
Experts

• Suppose we have several “experts” or component regression models generating conditional Gaussian outputs

\[
P(y|\mathbf{x}, \theta_i) = N(y; \mathbf{w}_i^T \mathbf{x} + w_{i0}, \sigma_i^2)
\]

where

mean of \( y \) given \( \mathbf{x} \) = \( \mathbf{w}_i^T \mathbf{x} + w_{i0} \)

variance of \( y \) given \( \mathbf{x} \) = \( \sigma_i^2 \)

\( \theta_i = \{ \mathbf{w}_i, w_{i0}, \sigma_i^2 \} \) denotes the parameters of the \( i^{th} \) expert.

• We need to find an appropriate way of allocating tasks to these experts (linear regression models)
Mixtures of experts

Example:

- Here we need a switch or a gating network that selects the appropriate expert (linear regression model) as a function of the input $x$. 
Gating network

- A simple gating network is a probability distribution over the choice of the experts conditional on the input $x$

- Example: in case of two experts (0 and 1), the gating network can be a logistic regression model

$$P(\text{expert} = 1|x, v, v_0) = g(v^T x + v_0)$$

where $g(z) = (1 + e^{-z})^{-1}$ is the logistic function.

- In case of $m > 2$ experts, the gating network can be a softmax model

$$P(\text{expert} = j|x, \eta) = \frac{\exp(v_j^T x + v_{j0})}{\sum_{j'=1}^m \exp(v_{j'}^T x + v_{j'0})}$$

where $\eta = \{v_1, \ldots, v_m, v_{10}, \ldots, v_{m0}\}$ are the parameters in the gating network
Gating network cont’d

\[ P(\text{expert} = j \mid x, \eta) = \frac{\exp(\mathbf{v}_j^T x + v_{j0})}{\sum_{j' = 1}^{m} \exp(\mathbf{v}_{j'}^T x + v_{j'0})} \]
Mixtures of experts model

- The probability distribution over the (regression) output $y$ given the input $x$ is a conditional mixture model

$$P(y|x, \theta, \eta) = \sum_{j=1}^{m} P(\text{expert} = j|x, \eta) \cdot P(y|x, \theta_j)$$

where $\eta$ defines the parameters of the gating network (e.g., logistic) and $\theta_j$ are the parameters of each expert (e.g., linear regression model).

- The allocation of experts is made conditionally on the input
Estimation of mixtures of experts

- The estimation would be again easy if we had the assignment of which expert should account for which training example.
- In other words, if we had \( \{(x_1, k_1, y_1), \ldots, (x_n, k_n, y_n)\} \), where \( k_i \) indicates the expert assigned to the \( i^{th} \) example.

1. Separately for each expert \( j \)
   
   Find \( \theta_j \) that maximize
   
   \[
   \sum_{i=1: k_i = j}^{n} \log P(y_i | x_i, \theta_j)
   \]

   (linear regression based on points “labeled” \( j \))

2. For the gating network
   
   Find \( \eta \) that maximize
   
   \[
   \sum_{i=1}^{n} \log P(\text{expert} = k_i | x_i, \eta)
   \]

   (softmax regression problem to predict the assignments)
Similarly to mixture models, we now have to evaluate the posterior probability (here given both $x_i$ AND $y_i$) that the output came from a particular expert:

$$
\hat{p}(j|i) \leftarrow P(\text{expert} = j|x_i, y_i, \eta, \theta)
= \frac{P(\text{expert} = j|x_i, \eta) P(y_i|x_i, \theta_j)}{\sum_{j'=1}^{m} P(\text{expert} = j'|x_i, \eta) P(y_i|x_i, \theta_{j'})}
$$
Estimation of mixtures of experts

**E-step:** evaluate the posterior probabilities \( \hat{p}(j|i) \) that partially assign experts to training examples

**M-step(s):**

1. Separately for each expert \( j \)
   
   Find \( \theta_j \) that maximize
   \[
   \sum_{i=1}^{n} \hat{p}(j|i) \log P(y_i|x_i, \theta_j)
   \]
   (weighted linear regression)

2. For the gating network

   Find \( \eta \) that maximize
   \[
   \sum_{i=1}^{n} \sum_{j=1}^{m} \hat{p}(j|i) \log P(\text{expert} = j|x_i, \eta)
   \]
   (weighted softmax regression)
Mixtures of experts: demo
Mixtures of experts: additional considerations

- Softmax gating network

\[
P(\text{expert} = j | x, \eta) = \frac{\exp( v_j^T x + v_{j0} )}{\sum_{j'=1}^m \exp( v_{j'}^T x + v_{j'0} )}
\]

- Gaussian gating network

\[
P(\text{expert} = j | x, \eta) = \frac{\exp\left( -\frac{1}{2} (x - v_j)^T \Sigma_j^{-1} (x - v_j) \right)}{\sum_{j'=1}^m \exp\left( -\frac{1}{2} (x - v_{j'})^T \Sigma_{j'}^{-1} (x - v_{j'}) \right)}
\]

What if \( \Sigma_1 = \ldots = \Sigma_m \)? Are these still different?
Hierarchical mixtures of experts

- The “gates” can be arranged hierarchically:

\[
P(j=1|x) \quad P(j=0|x) \\
P(c=3|j=0,x) P(c=1|j=1,x) \\
P(y | j=0,c=3,x) \quad \ldots \quad P(y | j=1,c=1,x)
\]

where for example:

\[
P(c = k | j = 1, x, \eta_j) = \frac{\exp(v_{1k}^T x + v_{1k0})}{\sum_{k' = 1}^{3} \exp(v_{1k'}^T x + v_{1k'0})}
\]

- We can estimate these with the EM-algorithm similarly to hierarchical mixture models