6.891 Machine learning and neural networks

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

LECTURE 13 (October 24, 2000)

Mid-term is *this* Thursday

Problem set #3 is available (finally)

Topics

- Continuing last lecture a bit ...
  - semi-supervised clustering (information bottleneck)
- Other mixture models
  - hierarchical mixture models
  - mixtures of experts
  - hierarchical mixtures of experts
- Markov models
- Hidden markov models

Semi-supervised clustering

- Let's assume we have identified the *relevant* information for clustering the examples

  For each $i = 1, \ldots, n$:

$\mathbf{x}_i$	Training example (e.g., a text document)
$P(y\|\mathbf{x}_i)$	Relevance information per example (e.g., word distribution)

- We wish to cluster the examples into larger groups without loosing the relevance information (in this case word frequency)
- Documents with similar word frequencies should be put into the same cluster

Semi-supervised clustering cont'd

- We derive a metric for clusters of examples (documents) $\{\mathbf{x}_i\}$ based on the relevance information $\{P(y|\mathbf{x}_i)\}$ (word frequency)

  The word frequencies for a cluster C come from randomly picking a document in the cluster

$$\hat{P}(y = j|C) = \frac{1}{|C|} \sum_{i \in C} P(y = j|\mathbf{x}_i) \tag{1}$$

$$\hat{P}(C) = \frac{|C|}{n}, \tag{2}$$

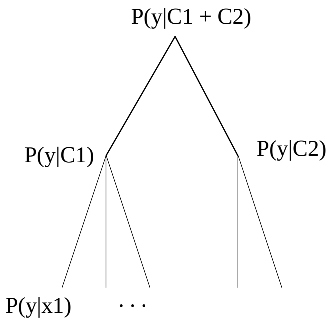
  The distance between the clusters measures how much information we loose about the words if the clusters are merged

$$d(C_l, C_k) = \frac{|C_l| + |C_k|}{n} \cdot I(\,y\,;\text{cluster identity}\,) \tag{3}$$

Semi-supervised clustering cont'd

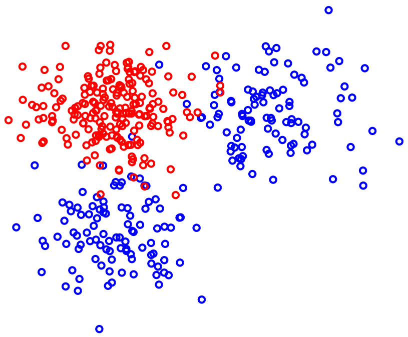
The distance between the clusters measures how much information we loose about the words if the clusters are merged

$$d(C_l, C_k) = \frac{|C_l| + |C_k|}{n} \cdot I(\, y\, ;\, \text{cluster identity}\,) \qquad (4)$$



Semi-supervised clustering: example

- Suppose we have a set of labeled examples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$



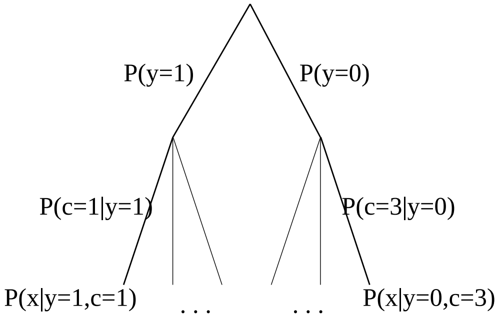
- We can take the label as the relevance variable.

$$P(y|\mathbf{x}_i) = 1, \;\text{ if } y = y_i \text{ and zero otherwise} \qquad (5)$$

Hierarchical mixture models

- We have already used a hierarchical mixture model in the context of the digit recognition problem (and in clustering)

  Data generation model:



First level: class distinction

Second level: class contingent components (e.g., style)

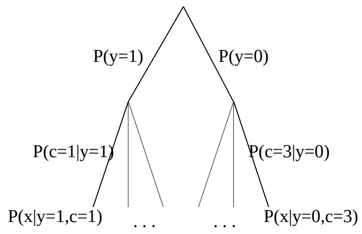
Hierarchical mixture models cont'd

- The hierarchy may not in general represent class distinction $\Rightarrow$ the top level division may also be unknown for all training examples $\mathbf{x}$

  E-step: We need to compute posterior probabilities over the possible paths in the tree

$$\hat{P}(j,k|i) \leftarrow \overbrace{P(y=j|\mathbf{x})}^{\text{First level}} \; \overbrace{P(c=k|y=j,\mathbf{x})}^{\text{Second level}}, \qquad (6)$$

  where $j = 1, 2$ and $k = 1, 2, 3$. In general the tree need not be symmetric.



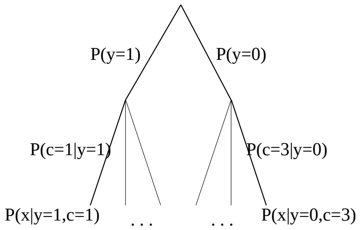
Hierarchical mixture models cont'd

- The posterior over the first division

$$P(y=j|\mathbf{x}) \;=\; \frac{P(\mathbf{x}|y=j)P(y=j)}{\sum_{j'=1}^{2} P(\mathbf{x}|y=j')P(y=j')} \qquad (7)$$

where the probability of generating $\mathbf{x}$ from the $y=j$ branch is

$$P(\mathbf{x}|y=j) = \sum_{k=1}^{3} P(c=k|y=j)\, P(\mathbf{x}|y=j,c=k) \qquad (8)$$

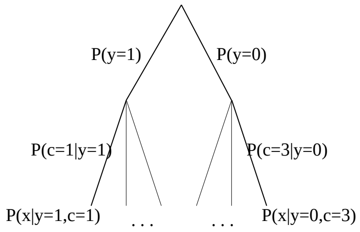


Hierarchical mixture models cont'd

- The conditional posterior over the second division

$$P(c=k|y=j,\mathbf{x}) \;=\; \frac{P(\mathbf{x}|y=j,c=k)P(c=k|y=j)}{\sum_{k'=1}^{3} P(\mathbf{x}|y=j,c=k')P(c=k'|y=j)}$$

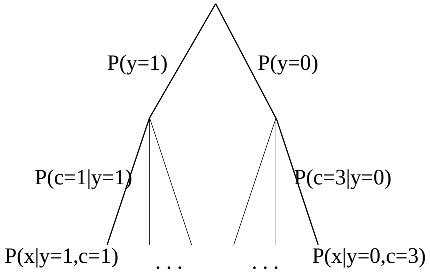
- Note that the normalization term equals $P(\mathbf{x}|y=j)$, the probability of generating $\mathbf{x}$ from one of the branches after $y=j$.
- This is a term we needed for evaluating the previous posterior $\Rightarrow$ perhaps we can evaluate these probabilities by propagating information in the tree?



Propagation in hierarchical mixture models

- Bottom-up phase:

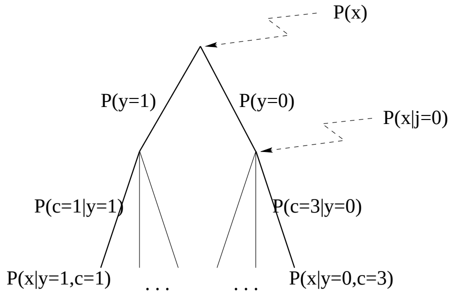
$$\begin{aligned} P(\mathbf{x}|y=j) &= \sum_{k=1}^{3} P(\mathbf{x}|y=j,c=k)P(c=k|y=j), \;\; j=0,1 \\ P(\mathbf{x}) &= \sum_{j=0}^{1} P(\mathbf{x}|y=j)P(y=j) \end{aligned}$$



Propagation in hierarchical mixture models cont'd

- Top-down phase:

$$\begin{aligned} P(y=j|\mathbf{x}) &= \frac{P(\mathbf{x}|y=j)P(y=j)}{P(\mathbf{x})} \\ P(c=k,y=j|\mathbf{x}) &= P(c=k|y=j,\mathbf{x}) \times P(y=j|\mathbf{x}) \\ &= \left[\frac{P(\mathbf{x}|y=j,c=k)P(c=k|y=j)}{P(\mathbf{x}|y=j)}\right] \times P(y=j|\mathbf{x}) \end{aligned}$$



Mixtures of experts

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

  Examples:
  1. Dealing with various styles in handwritten character recognition
  2. Dealing with dialect/accent in speech recognition

  etc.
- Each sub-problem can be solved by a specific "expert"
- The selection of which expert to rely on must depend on the context (i.e., the input $\mathbf{x}$)

Experts

- Suppose we have several "experts" or component regression models generating conditional Gaussian outputs

$$P(y|\mathbf{x}, \theta_i) = N(y;\, \mathbf{w}_i^T\mathbf{x} + w_{i0},\, \sigma_i^2) \tag{9}$$

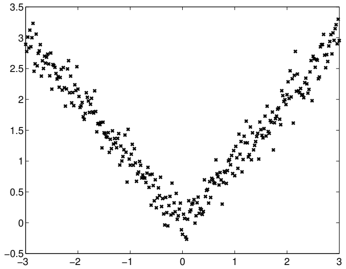
  where

$$\begin{aligned} \text{mean of } y \text{ given } \mathbf{x} &= \mathbf{w}_i^T\mathbf{x} + w_{i0} \\ \text{variance of } y \text{ given } \mathbf{x} &= \sigma_i^2 \end{aligned}$$

  We use $\theta_i = \{\mathbf{w}_i, w_{i0}, \sigma_i^2\}$ to denote 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 $\mathbf{x}$

Gating network

- A simple gating network is a probability distribution over the choice of the experts conditional on the input $\mathbf{x}$
- Example: in case of two experts (0 and 1), the gating network can be a logistic regression model

$$P(\text{expert} = 1|\mathbf{x}, \mathbf{v}_1, v_{10}) = g(\,\mathbf{v}_1^T\mathbf{x} + v_{10}\,) \tag{10}$$

  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

$$P(\text{expert} = j|\mathbf{x}, \eta) = \frac{\exp(\,\mathbf{v}_j^T\mathbf{x} + v_{j0}\,)}{\sum_{j'=1}^{m} \exp(\,\mathbf{v}_{j'}^T\mathbf{x} + v_{j'0}\,)} \tag{11}$$

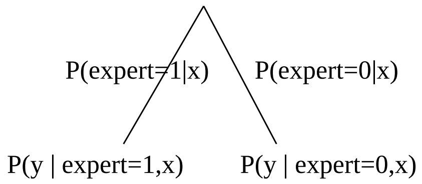
  where $\eta = \{\mathbf{v}_1, \ldots, \mathbf{v}_m, v_{10}, \ldots, v_{m0}\}$ are the parameters in the gating network

Mixtures of experts model

- The probability distribution over the output y given $\mathbf{x}$ is

$$P(y|\mathbf{x},\theta,\eta) = \sum_{j=1}^{m} P(\text{expert} = j|\mathbf{x},\eta)\, P(y|\mathbf{x},\theta_j) \tag{12}$$

- This can generate piece-wise linear outputs with Gaussian noise
- The allocation of experts is made conditionally on the input
- Only a single expert is assumed to be responsible for any specific input output mapping



Estimation of mixtures of experts

- Again, if someone labeled the training examples in terms of which expert generated which output, estimation would be much simpler
- In other words, if we have $\{(\mathbf{x}_1,k_1,y_1),\ldots,(\mathbf{x}_n,k_n,y_n)\}$, where k_i indicates which expert is responsible for which output:
  1. Separately for each expert j

     $$\text{Find } \theta_j \text{ that maximize} \quad \sum_{i=1:\,k_i=j}^{n} \log P(y_i|\mathbf{x}_i,\theta_j) \tag{13}$$

     This is linear regression using all the data points labeled j.
  2. For the gating network

     $$\text{Find } \eta \text{ that maximize} \quad \sum_{i=1}^{n} \log P(\text{expert} = k_i|\mathbf{x}_i,\eta)$$

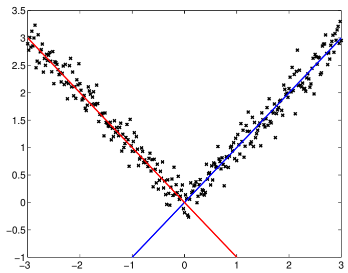
     This is a softmax regression problem (cf. problem set #2).

Estimation of mixtures of experts

- Similarly to mixture models, we now have to evaluate the posterior probability (given $\mathbf{x}_i$ AND y_i)b that the output came from a particular expert:

$$\hat{P}(j|i) \;\leftarrow\; P(\text{expert} = j|\mathbf{x}_i,y_i,\eta,\theta) \tag{14}$$

$$= \frac{P(\text{expert} = j|\mathbf{x}_i,\eta)\, P(y_i|\mathbf{x}_i,\theta_j)}{\sum_{j'=1}^{m} P(\text{expert} = j'|\mathbf{x}_i,\eta)\, P(y_i|\mathbf{x}_i,\theta_{j'})} \tag{15}$$



Estimation of mixtures of experts

E-step: evaluate the posterior probabilities $\hat{P}(j|i)$ that expert j is reponsible for training example i

M-step:

1. Separately for each expert j

   $$\text{Find } \theta_j \text{ that maximize} \quad \sum_{i=1}^{n} \hat{P}(j|i)\, \log P(y_i|\mathbf{x}_i,\theta_j) \tag{16}$$

   (weighted linear regression)
2. For the gating network

   $$\text{Find } \eta \text{ that maximize} \quad \sum_{i=1}^{n}\sum_{j=1}^{m} \hat{P}(j|i)\, \log P(\text{expert} = j|\mathbf{x}_i,\eta)$$

   (weighted softmax regression)

Mixtures of experts: additional considerations

- Softmax gating network

$$P(\text{expert} = j|\mathbf{x}, \eta) = \frac{\exp(\, \mathbf{v}_j^T\mathbf{x} + v_{j0}\,)}{\sum_{j'=1}^{m} \exp(\, \mathbf{v}_{j'}^T\mathbf{x} + v_{j'0}\,)} \tag{17}$$

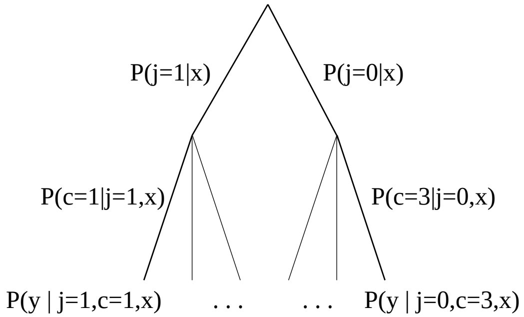
- Gaussian gating network

$$P(\text{expert} = j|\mathbf{x}, \eta) = \frac{\exp(\, -\frac{1}{2}(\mathbf{x} - \mathbf{v}_j)^T\Sigma_j^{-1}(\mathbf{x} - \mathbf{v}_j)\,)}{\sum_{j'=1}^{m} \exp(\, -\frac{1}{2}(\mathbf{x} - \mathbf{v}_{j'})^T\Sigma_{j'}^{-1}(\mathbf{x} - \mathbf{v}_{j'})\,)} \tag{18}$$

What if $\Sigma_1 = \ldots = \Sigma_m$? Are these still different?

Hierarchical mixtures of experts

- The "gates" can be arranged hierarchically:



where for example:

$$P(c = k|j = 1, \mathbf{x}, \eta_j) = \frac{\exp(\, \mathbf{v}_{1k}^T\mathbf{x} + v_{1k0}\,)}{\sum_{k'=1}^{3} \exp(\, \mathbf{v}_{1k'}^T\mathbf{x} + v_{1k'0}\,)} \tag{19}$$

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