6.867 Machine learning and neural networks

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

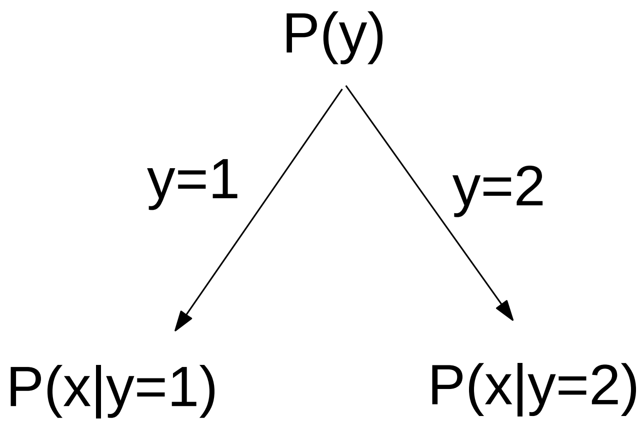
Lecture 12: mixtures, hierarchies, and experts

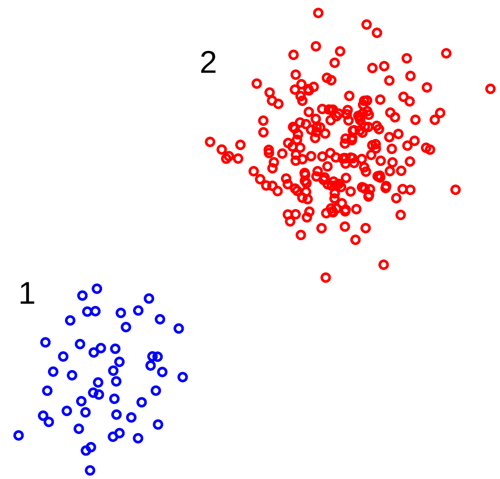
Topics

- Density estimation
  - Mixture models in classification, example
  - Hierarchical mixture models, estimation
- Conditional density models
  - experts, mixtures of experts

Review: 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

Review: the EM algorithm

E-step: First we perform a soft reassignment of examples based on the current mixture distribution, i.e., we compute

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

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

For each $j = 1, 2$, we maximize the likelihood of the corresponding weighted training set

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

Review: the EM algorithm

E-step: First we perform a soft reassignment of examples based on the current mixture distribution, i.e., we compute

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

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

$$\begin{aligned}
\hat{n}_j &= \sum_{i=1}^{n} \hat{p}(j|i) = \text{Soft \# of examples labeled } j \\
\hat{\mu}_j &\leftarrow \frac{1}{\hat{n}_j} \sum_{i=1}^{n} \hat{p}(j|i)\, \mathbf{x}_i \\
\hat{\Sigma}_j &\leftarrow \frac{1}{\hat{n}_j} \sum_{i=1}^{n} \hat{p}(j|i)\, (\mathbf{x}_i - \hat{\mu}_j)(\mathbf{x}_i - \hat{\mu}_j)^T
\end{aligned}$$

where $j = 1, 2$.

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 (with varying numbers of mixture components).

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

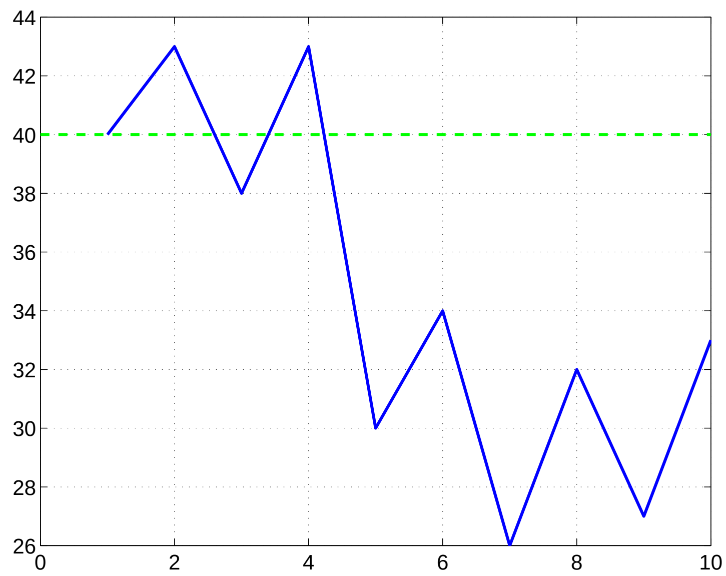
- Classification rule is based on the posterior class probability, or, equivalently, based on the log-likelihood ratio:

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

  (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 each digit model



- Anything wrong with this figure?

Classification example cont'd

- A single covariance matrix has $64 * 65/2 = 2080$ parameters, we have $n = 50$ training examples...
- We can regularize the model

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

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

  (written here in a bit non-standard way)

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

Classification example cont'd

- In the resulting M-step we maximize the penalized log-likelihood of the (weighted) training set

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

- Adding such a regularization penalty changes the estimation of the covariance matrix only slightly

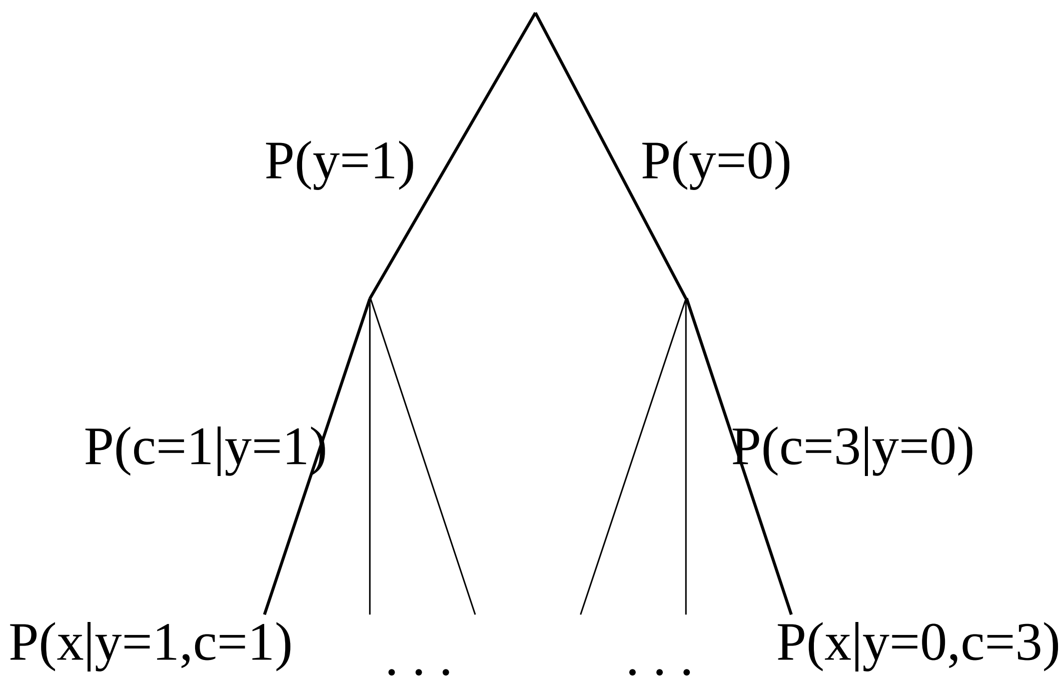
$$\hat{\Sigma}_j \;\leftarrow\; \frac{1}{\hat{n}_j + n'} \left[\sum_{i=1}^{n} \hat{p}(j|i)\,(\mathbf{x}_i - \hat{\mu}_j)(\mathbf{x}_i - \hat{\mu}_j)^T + n'S \right]$$

The remaining parts of the M-step are as before. Note that the E-step is unaffected (though the resulting values for the soft assignments will change)

Hierarchical mixture models

- We have already used a hierarchical mixture model in the digit recognition problem

  Data generation model:



  First level: class distinction

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

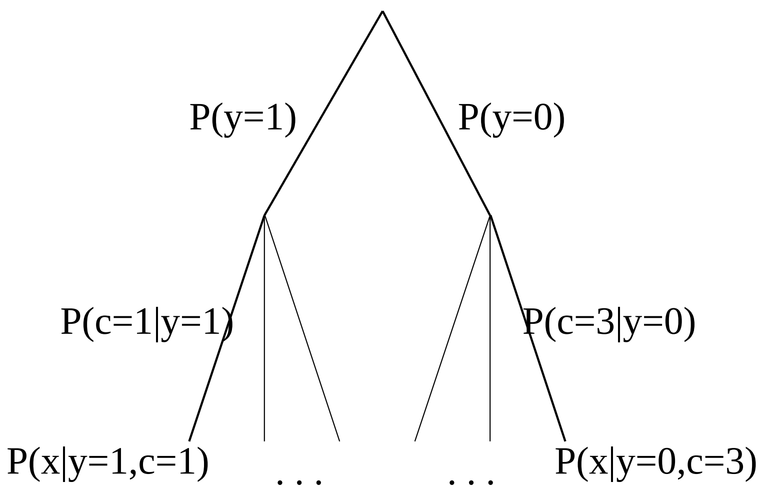
Hierarchical mixture models cont'd

- The hierarchy may not in general represent class distinction $\Rightarrow$ the top level division may also be unobserved for all training examples

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

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

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



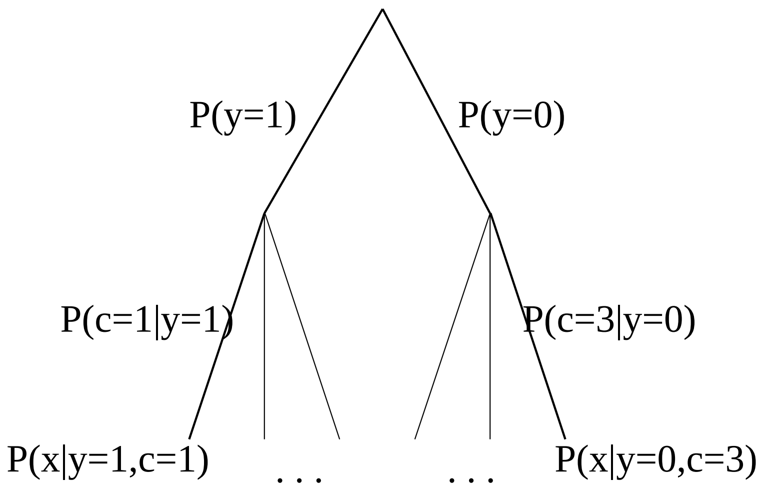
Hierarchical mixture models cont'd

- The posterior over the first division

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

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

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

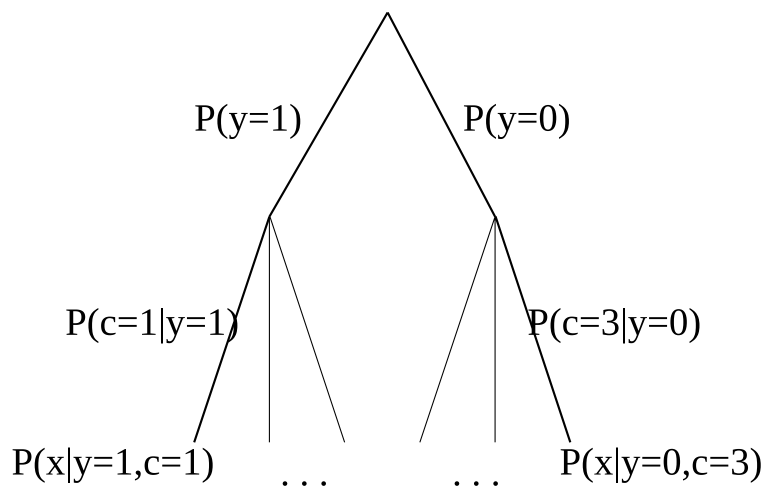


Hierarchical mixture models cont'd

- The conditional posterior over the second division

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

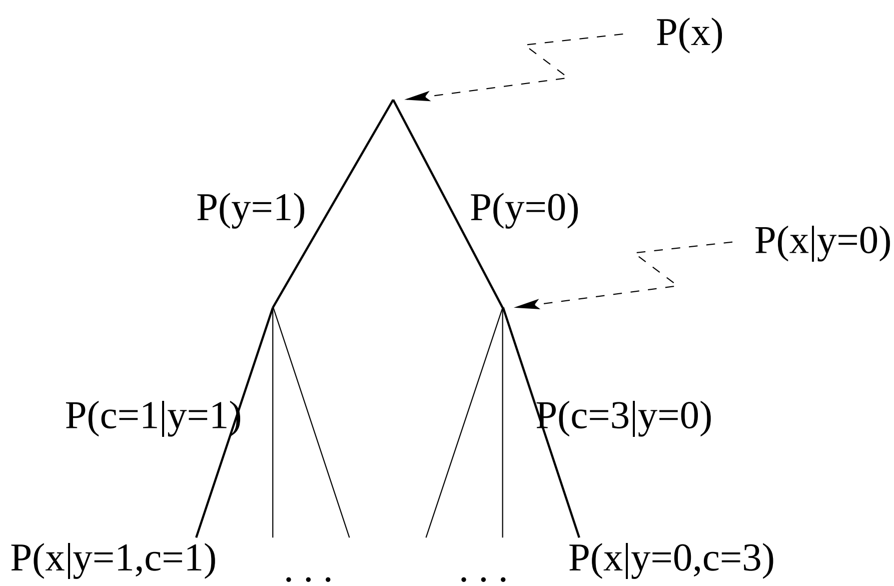
- Note that the normalization term equals $P(\mathbf{x}|y = j)$, the probability of generating $\mathbf{x}$ from one of the branches after $y = j$.
- This is a term we needed for evaluating the previous posterior $\Rightarrow$ perhaps we can evaluate these probabilities by propagating information in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



Propagation in hierarchical mixture models

- Bottom-up phase:

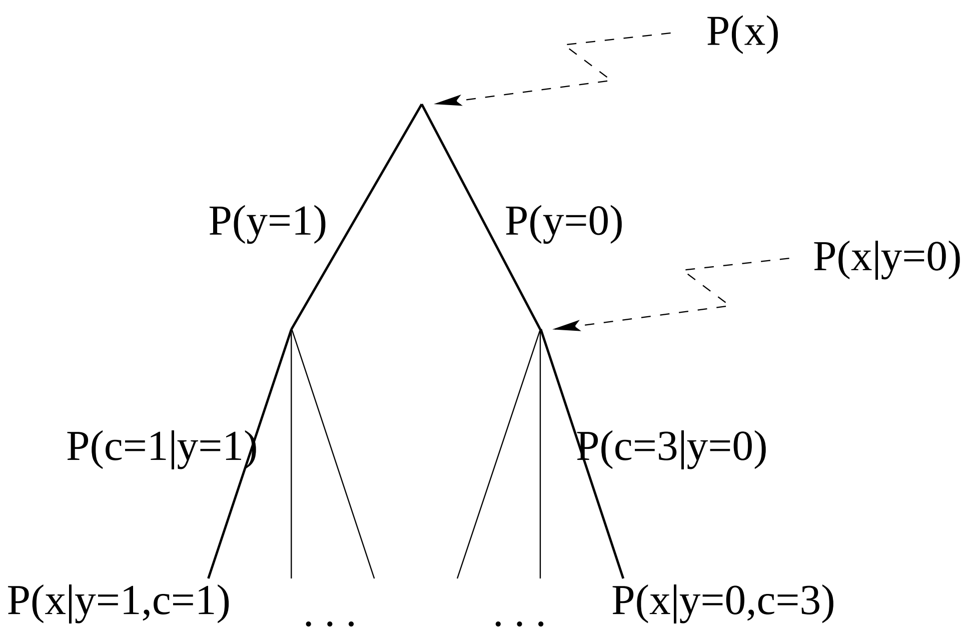
$$
\begin{aligned}
P(\mathbf{x}|y=j) &= \sum_{k=1}^{3} P(\mathbf{x}|y=j,c=k)P(c=k|y=j), \quad j=0,1 \\
P(\mathbf{x}) &= \sum_{j=0}^{1} P(\mathbf{x}|y=j)P(y=j)
\end{aligned}
$$



Propagation in hierarchical mixture models cont'd

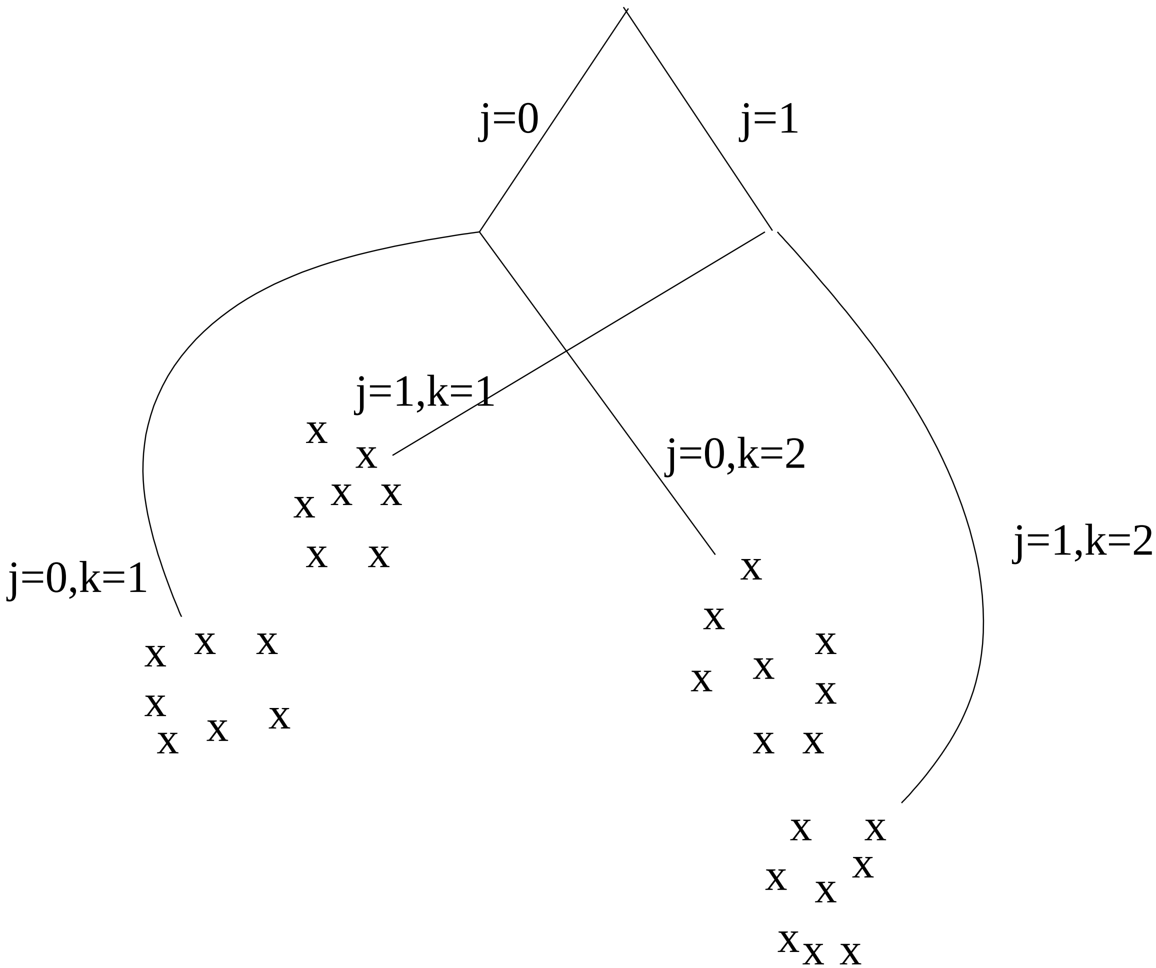
- Top-down phase:

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



Hierarchical mixture models

- Can this happen?



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

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

  where

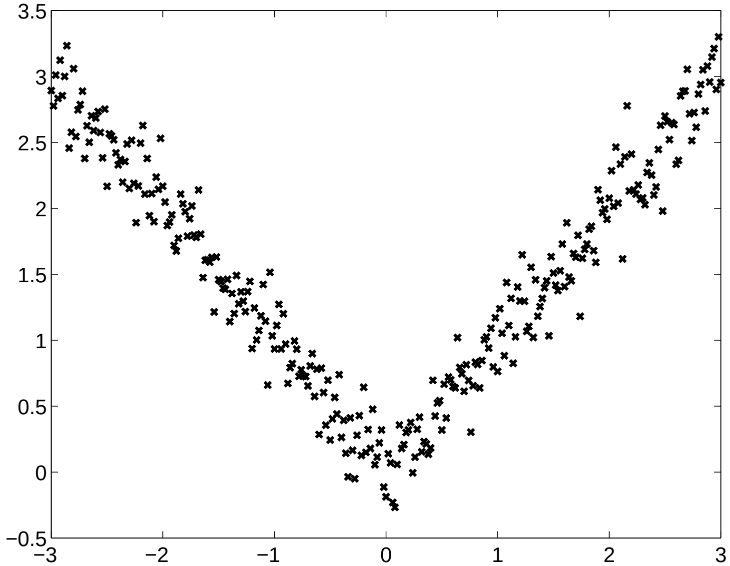
$$\begin{aligned}\text{mean of } y \text{ given } \mathbf{x} &= \mathbf{w}_i^T\mathbf{x} + w_{i0} \\ \text{variance of } y \text{ given } \mathbf{x} &= \sigma_i^2\end{aligned}$$

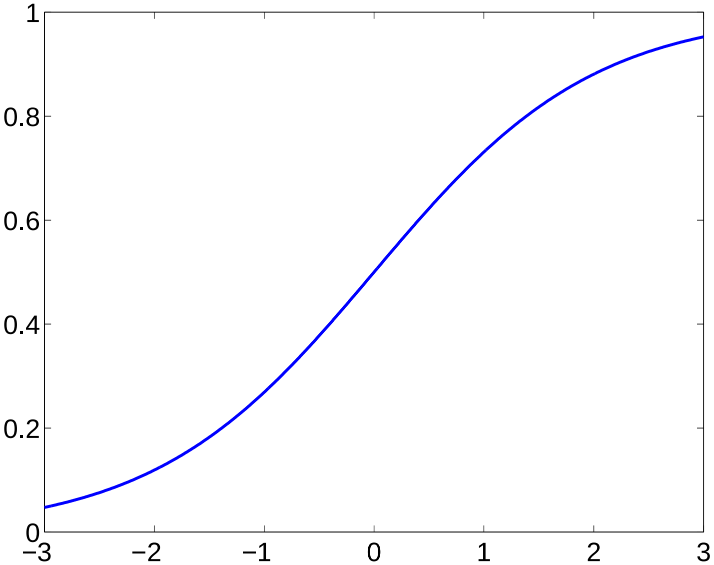
  We use $\theta_i = \{\mathbf{w}_i, w_{i0}, \sigma_i^2\}$ to denote the parameters of the i^{th} expert.

- 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

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

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

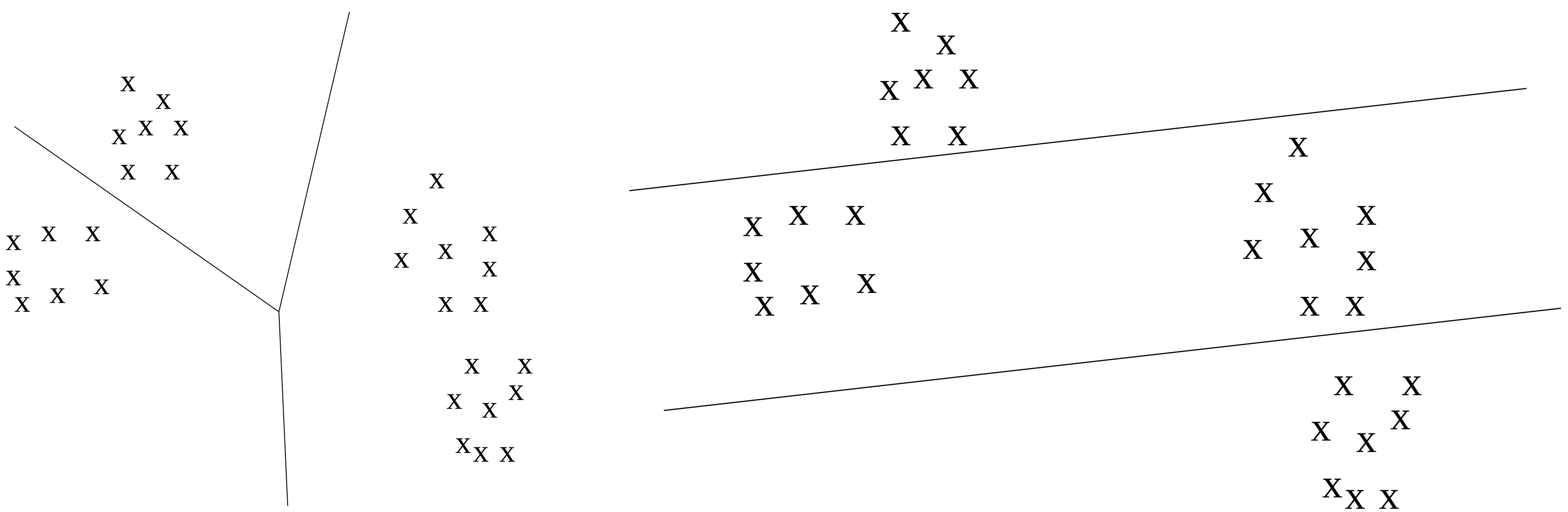
- In case of $m > 2$ experts, the gating network can be a softmax model

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

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

Gating network cont'd

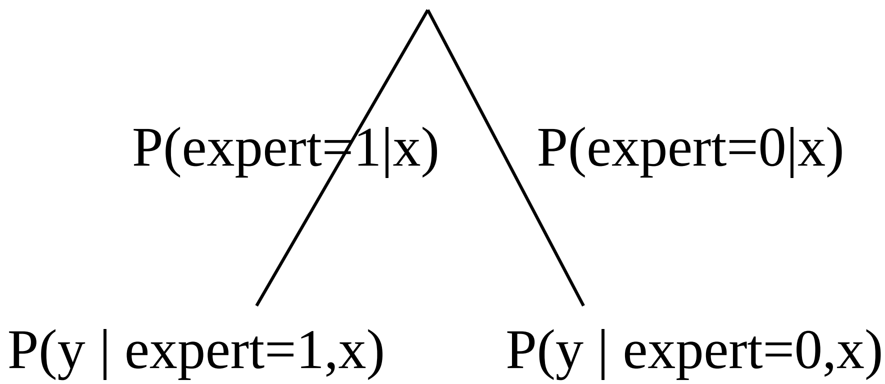
$$P(\text{expert} = j|\mathbf{x}, \eta) = \frac{\exp(\,\mathbf{v}_j^T\mathbf{x} + v_{j0}\,)}{\sum_{j'=1}^{m} \exp(\,\mathbf{v}_{j'}^T\mathbf{x} + v_{j'0}\,)}$$

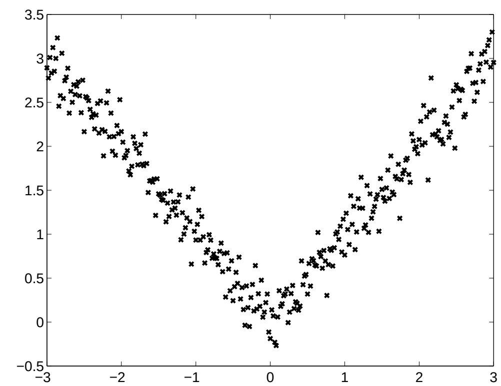


Mixtures of experts model

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

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





- 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