Machine learning: lecture 15

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

Topics

- Mixture models
  - selecting the number of components
  - non-parametric mixtures
  - conditional mixtures: mixtures of experts

The number of mixture components

- As a simple strategy for selecting the appropriate number of mixture components, we can find m that minimizes the overall description length:

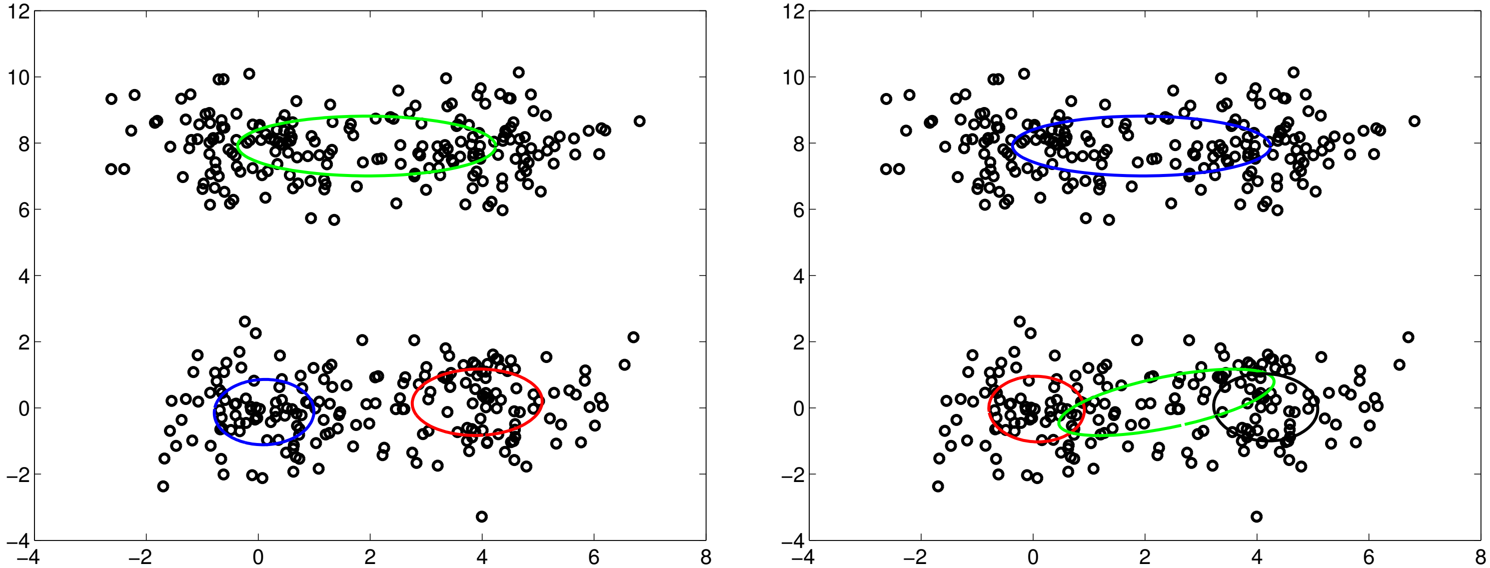
$$\text{DL} \approx -\log p(\text{data}|\hat{\theta}_m) + \frac{d_m}{2}\log(n)$$

where n is the number of training points, $\hat{\theta}_m$ are the maximum likelihood parameters for the m-component mixture, and d_m is the (effective) number of parameters in the m-mixture.

- This asymptotic approximation is also known as BIC (Bayesian Information Criterion)

The number of mixture components: example

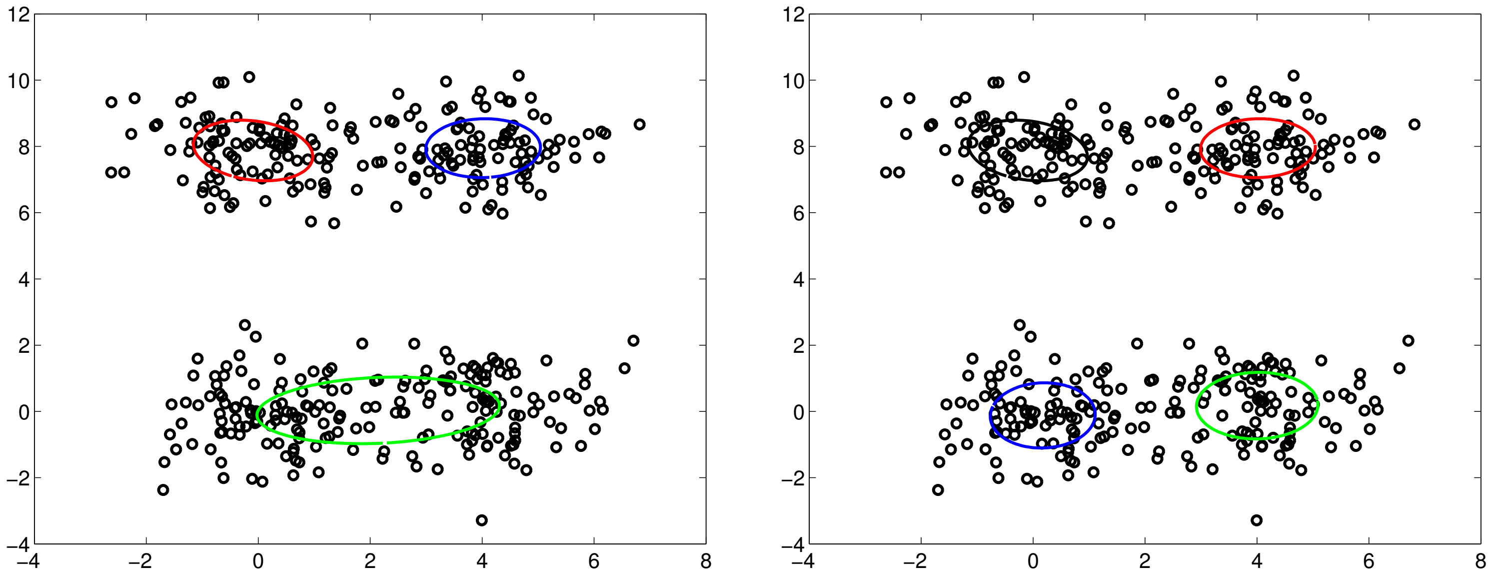
- Typical cases




```
m=1, -logP(data)=2017.38, penalty=14.98, DL=2032.36
m=2, -logP(data)=1712.69, penalty=32.95, DL=1745.65
m=3, -logP(data)=1711.40, penalty=50.93, DL=1762.32
m=4, -logP(data)=1682.06, penalty=68.90, DL=1750.97
```

The number of mixture components: example

- The best cases (out of many runs):



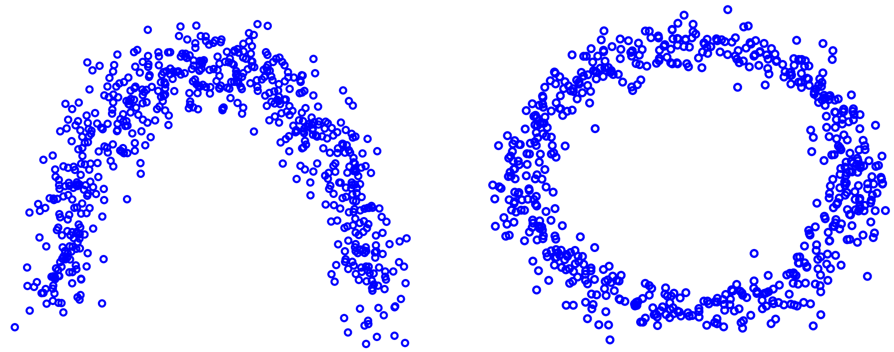
```
m=1, -logP(data)=2017.38, penalty=14.98, DL=2032.36
m=2, -logP(data)=1712.69, penalty=32.95, DL=1745.65
m=3, -logP(data)=1678.56, penalty=50.93, DL=1729.49
m=4, -logP(data)=1649.08, penalty=68.90, DL=1717.98
```

Topics

- Mixture models
  - selecting the number of components
  - non-parametric mixtures
  - conditional mixtures: mixtures of experts

Beyond parametric density models

- More mixture densities



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

$$p(\mathbf{x}|\theta) = \sum_{j=1}^{m} p_j \, p(\mathbf{x}|\mu_j, \Sigma_j)$$

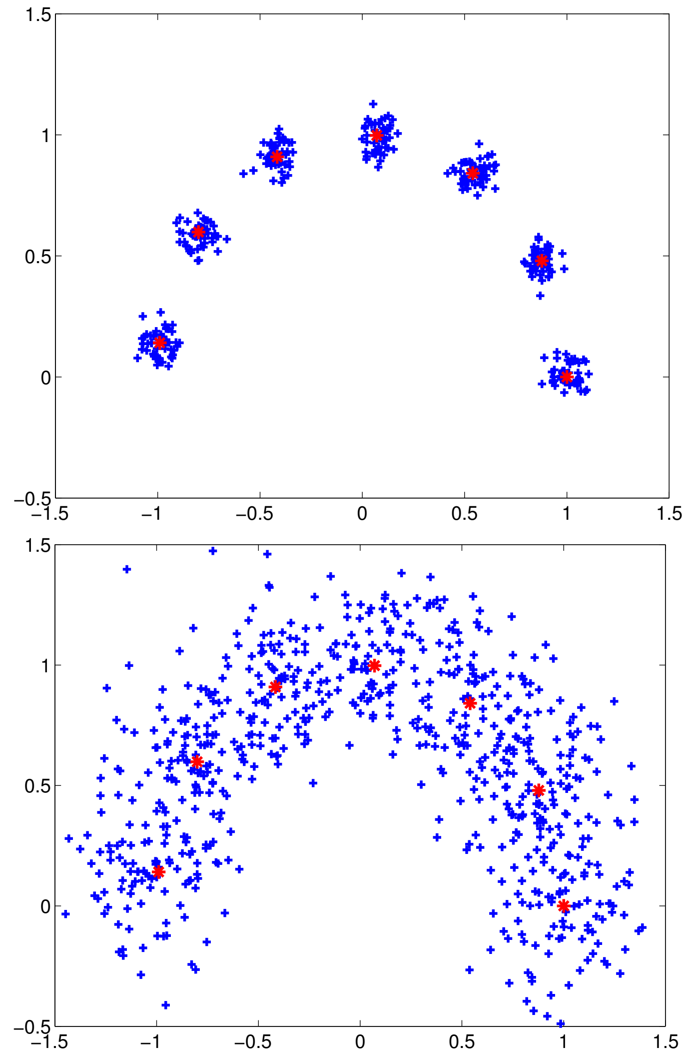
Non-parametric densities

- We can even introduce one mixture component (Gaussian) per training example

$$\hat{p}(\mathbf{x};\sigma^2) = \frac{1}{n}\sum_{i=1}^{n} p(\mathbf{x}|\mathbf{x}_i,\sigma^2 I)$$

where n is the number of examples.

Here the covariances are all equal and spherical; the single parameter σ^2 controls the smoothness of the resulting density estimate



1-dim case: Parzen windows

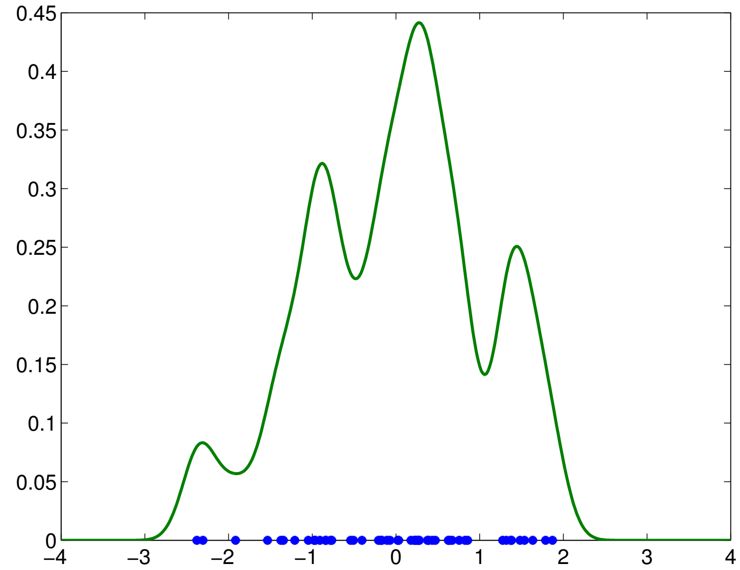
- We place a smooth Gaussian (or other) bump on each training example

$$\hat{p}_n(x;\sigma) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{\sigma} K\left(\frac{x - x_i}{\sigma}\right), \text{ where}$$

where the "kernel function" is

$$K(z) = \frac{1}{\sqrt{2\pi}} \exp(-z^2/2)$$

(very different from SVM kernels).



$n = 50, \ \sigma = 0.02$

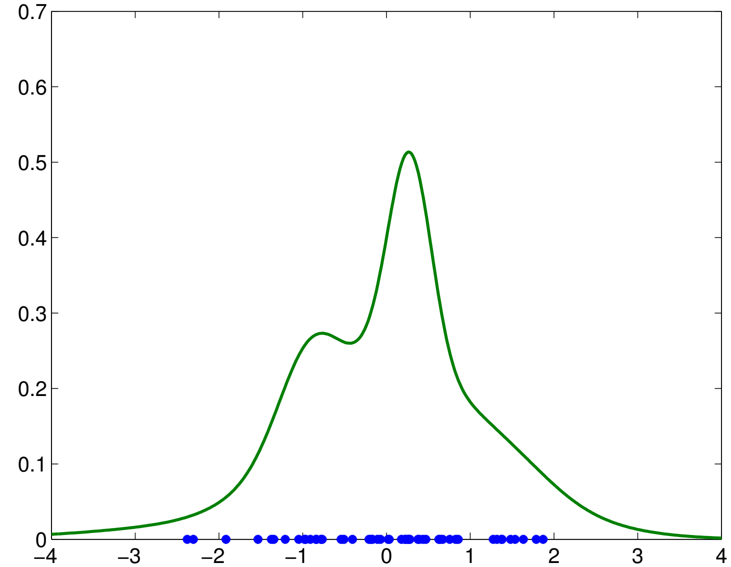
Parzen windows: variable kernel width

- We can also set the kernel width locally

  k-nearest neighbor choice: let d_{ik} be the distance from x_i to its k^{th} nearest neighbor

$$\hat{p}_n(x;k) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{d_{ik}} K\left(\frac{x - x_i}{d_{ik}}\right)$$

- The estimate is smoother where there are only few data points



Parzen windows: optimal kernel width

- We still have to set the kernel width σ or the number of nearest neighbors k
- A practical solution: cross-validation

  Let $\hat{p}_{-i}(x;\sigma)$ be a parzen windows density estimate constructed on the basis of $n-1$ training examples leaving out x_i.

  We select σ (or similarly k) that maximizes the leave-one-out log-likelihood

$$CV(\sigma) = \sum_{i=1}^{n} \log \hat{p}_{-i}(x_i;\sigma)$$

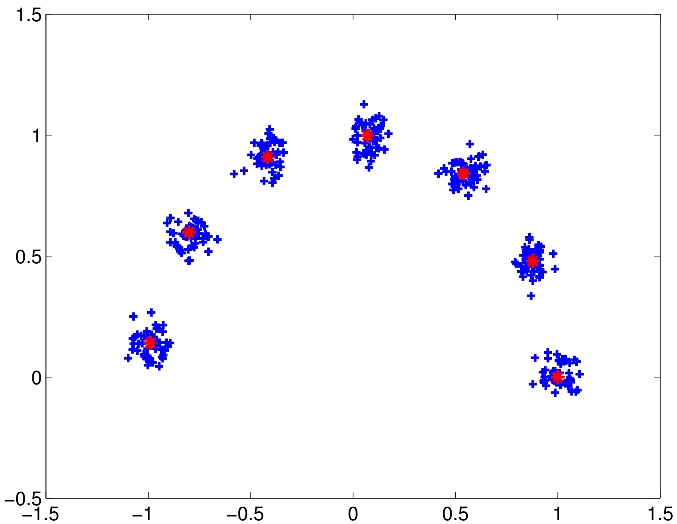
Parzen windows: multi-dimensional case

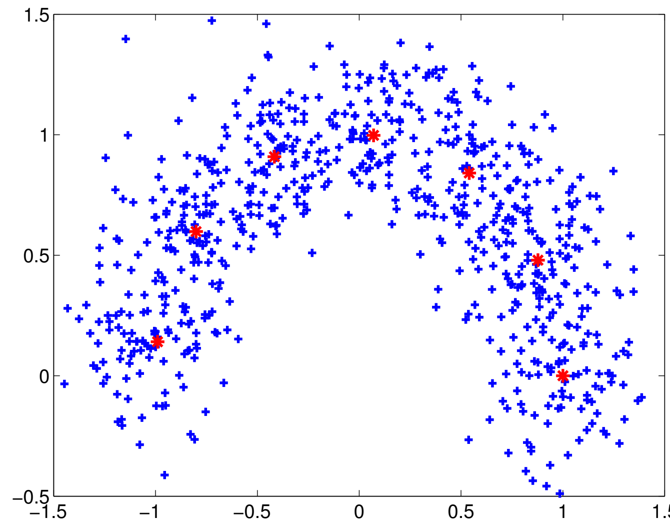
- Multi-dimensional Parzen windows estimate:

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

  where n is the number of examples.

- The covariance matrices are all equal and spherical. The single parameter σ controls the smoothness of the density estimate and can be set analogously





Topics

- Mixture models
  - selecting the number of components
  - non-parametric mixtures
  - conditional mixtures: mixtures of experts

Conditional mixtures

- Many regression or classification problems can be decomposed into smaller (easier) sub problems
- Examples:
  - style in handwritten character recognition
  - dialect/accent in speech recognition

  etc.
- Each sub-problem could be solved by a specific but relatively simple "expert"
- Unlike in ordinary mixtures, the selection of which expert to rely on must now depend on the context (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

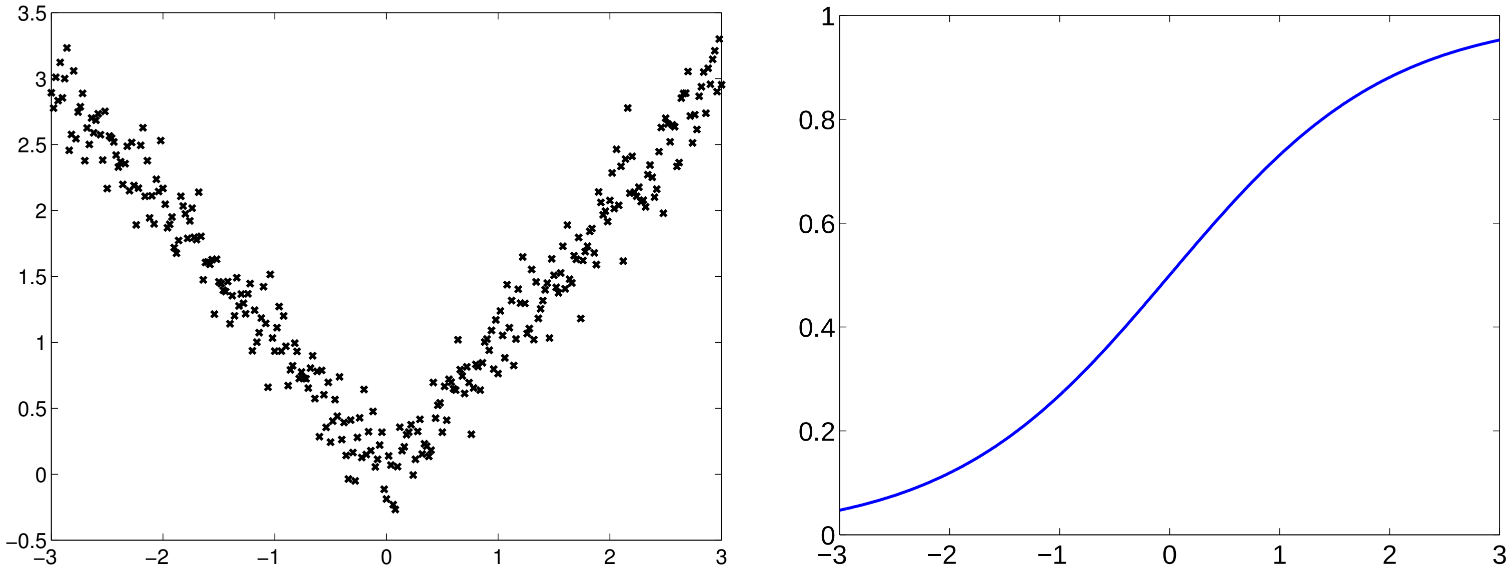
$$\begin{aligned} \text{mean of } y \text{ given } \mathbf{x} &= \mathbf{w}_i^T\mathbf{x} + w_{i0} \\ \text{variance of } y \text{ given } \mathbf{x} &= \sigma_i^2 \end{aligned}$$

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

Gating network

- A simple gating network is a probability distribution over the choice of the expert, conditional on the input $\mathbf{x}$

- Example: in case of just two experts (say 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.

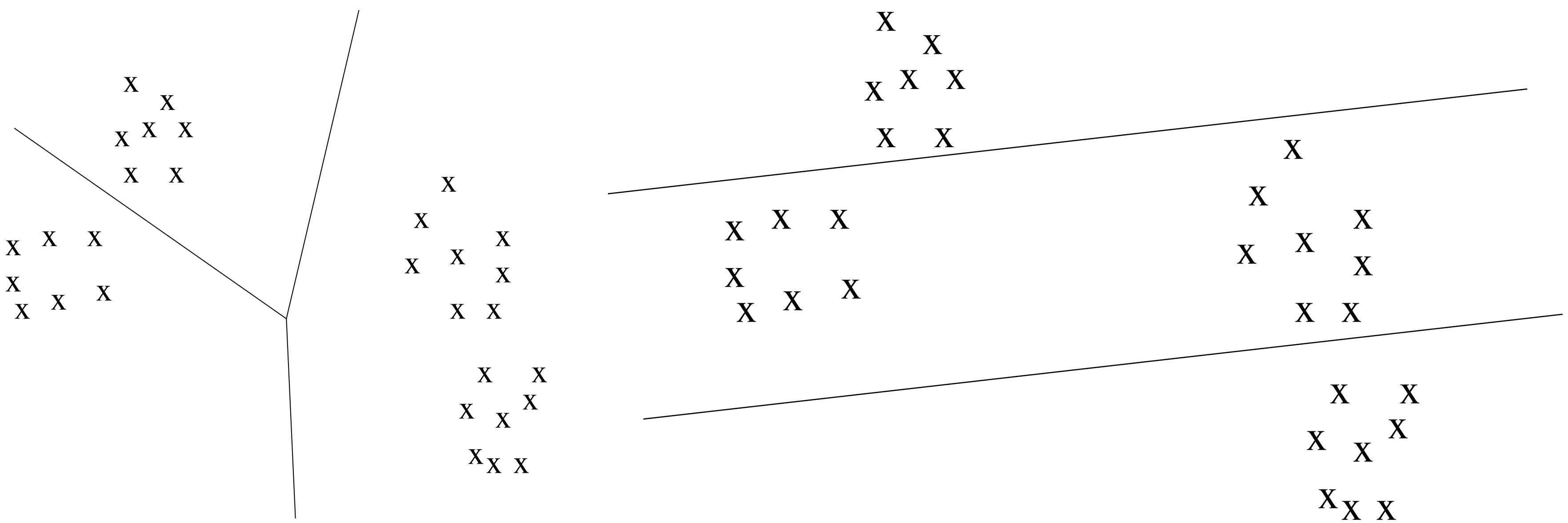
- When the number of experts $m > 2$, the gating network can be a softmax model

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

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

Gating network: example divisions

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

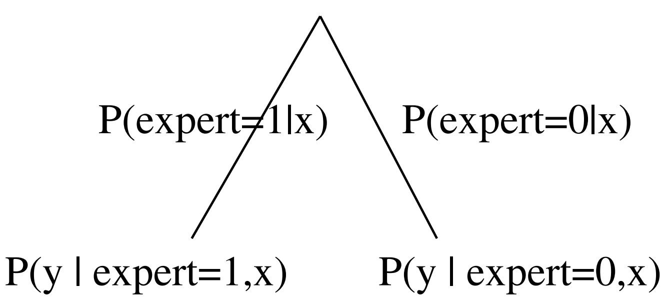


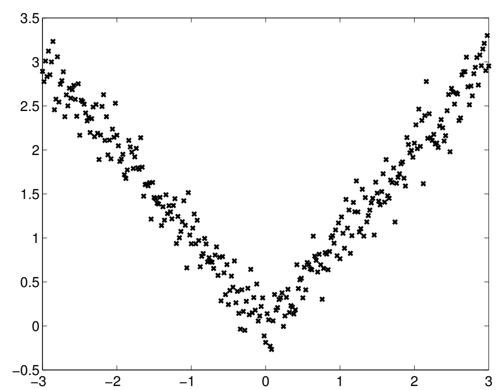
Mixtures of experts model

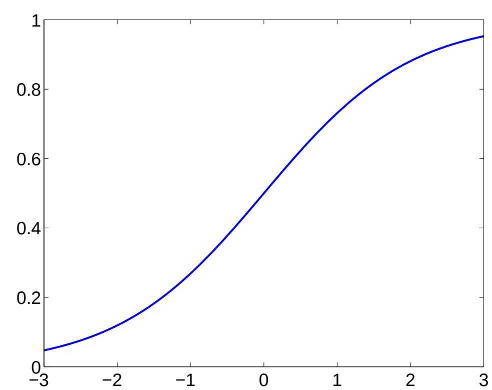
- The distribution over possible outputs y given an input $\mathbf{x}$ is a conditional mixture model

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

where η defines the parameters of the gating network (e.g., logistic) and θ_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 knew which expert should account for which training example

  Let $\delta(j|i)$ be indicator functions over the choice of the experts such that $\delta(k|i) = 1$ if expert k was responsible for training example i. If we knew $\delta(j|i)$, then

1. Separately for each expert j: find θ_j that maximize

$$\sum_{i=1}^{n} \delta(j|i) \log P(y_i|\mathbf{x}_i, \theta_j)$$

   (linear regression based on points "labeled" j)

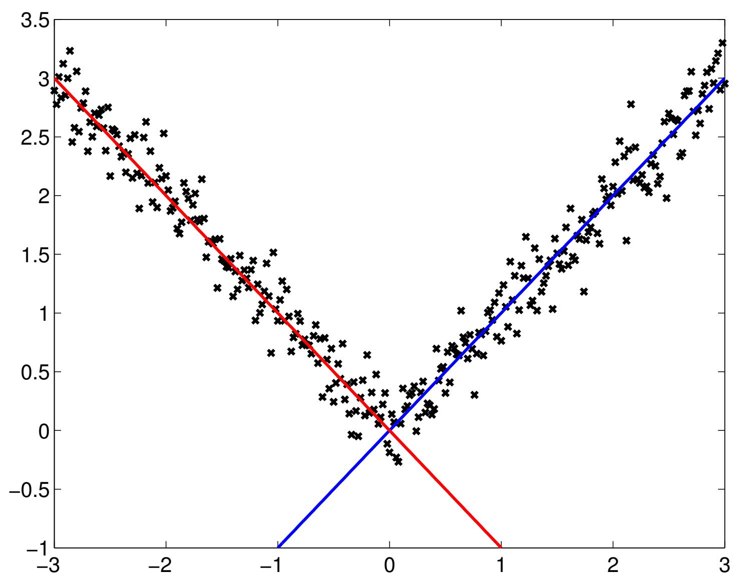
2. Find η to reproduce the assignments: maximize

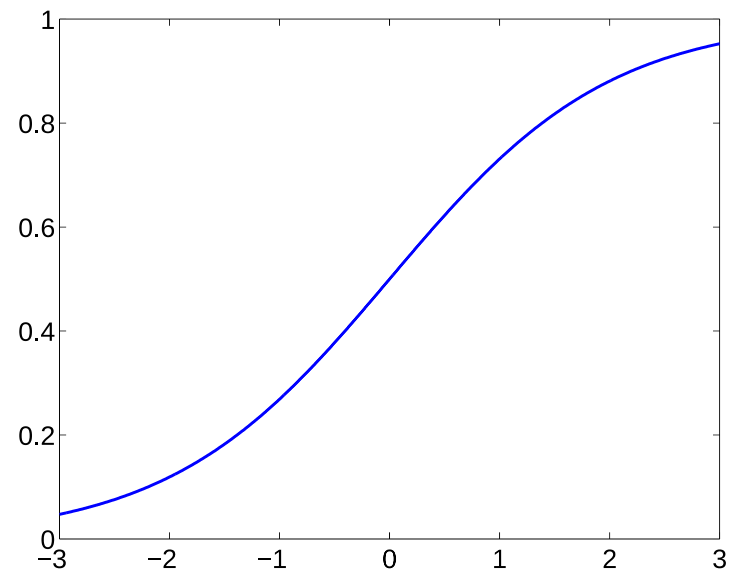
$$\sum_{i=1}^{n} \sum_{j=1}^{m} \delta(j|i) \log P(\text{expert} = j|\mathbf{x}_i, \eta)$$

Estimation of mixtures of experts

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

$$\begin{aligned}\hat{p}(j|i) \quad \leftarrow \quad & P(\text{expert} = j|\mathbf{x}_i, y_i, \eta, \theta) \\ = & \frac{P(\text{expert} = j|\mathbf{x}_i, \eta)\, P(y_i|\mathbf{x}_i, \theta_j)}{\sum_{j'=1}^{m} P(\text{expert} = j'|\mathbf{x}_i, \eta)\, P(y_i|\mathbf{x}_i, \theta_{j'})}\end{aligned}$$





EM for mixtures of experts

E-step: evaluate the posterior assignment probabilities $\hat{p}(j|i)$

M-step(s): separately re-estimate the experts and the gating network based on the posterior assignments:

1. For each expert j: find θ_j that maximize

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

2. For the gating network: find η that maximize

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

Mixtures of experts: demo