6.867 Machine learning and neural networks

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

Lecture 14: experts, non-parametric density estimation

Topics

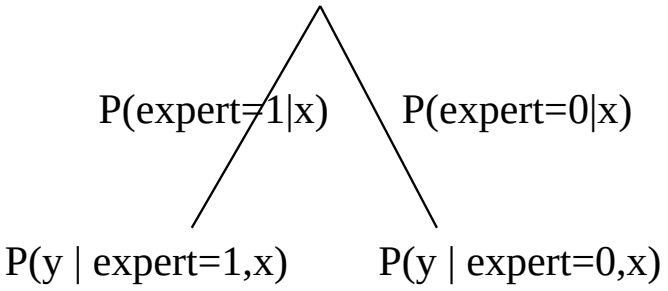
- Conditional density models
  - mixtures of experts, estimation
  - hierarchical mixtures of experts
- Non-parametric density estimation
  - Parzen windows
  - Global, local kernel width

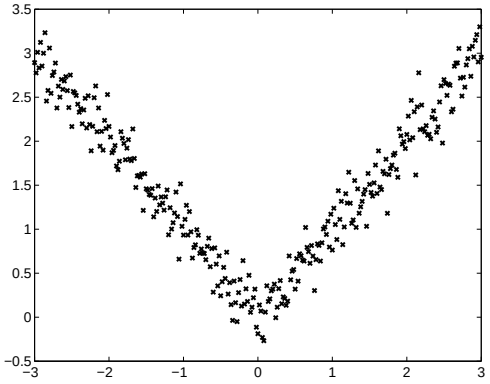
Mixtures of experts model

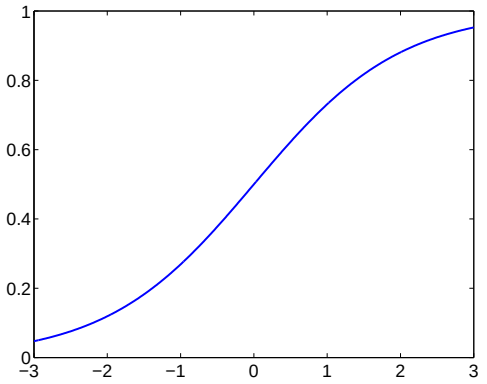
- The probability distribution over the (regression) output y given the input $\mathbf{x}$ is a conditional mixture model

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

  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
- Only a single expert is assumed to be responsible for any specific input output mapping

Estimation of mixtures of experts

- The estimation would be easy if we had the assignment of which expert should account for which training example
- In other words, if we had $\{(\mathbf{x}_1, k_1, y_1), \ldots, (\mathbf{x}_n, k_n, y_n)\}$, where k_i indicates the expert assigned to the i^{th} example

  1. Separately for each expert j

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

     This is linear regression using all the data points "labeled" j.

  2. For the gating network

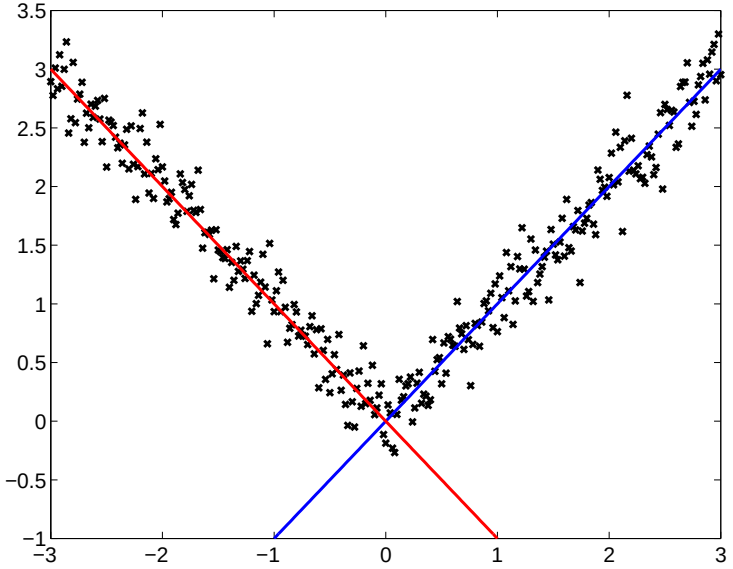
     $$\text{Find } \eta \text{ that maximize } \sum_{i=1}^{n} \log P(\text{expert} = k_i|\mathbf{x}_i, \eta)$$

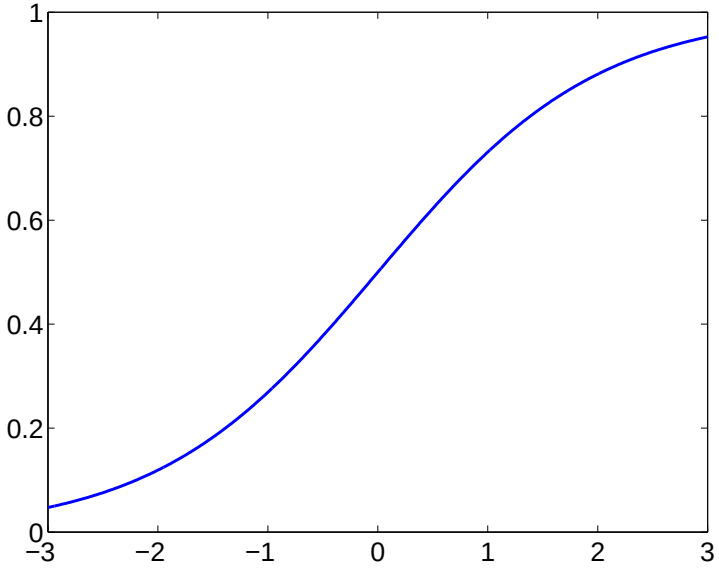
     This is a softmax regression problem

Estimation of mixtures of experts

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

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





Estimation of mixtures of experts

E-step: evaluate the posterior probabilities $\hat{P}(j|i)$ that softly assign experts to training examples

M-step:

1. Separately for each expert j

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

   (weighted linear regression)

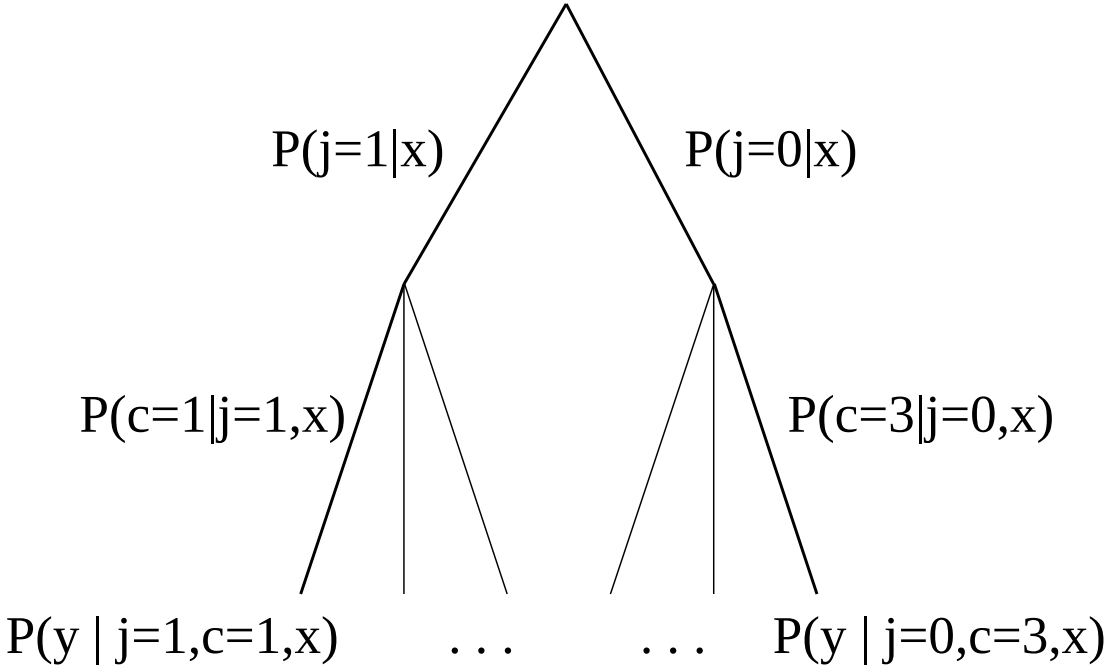
2. For the gating network

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

   (weighted softmax regression)

Hierarchical mixtures of experts

- The "gates" can be arranged hierarchically:



  where for example:

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

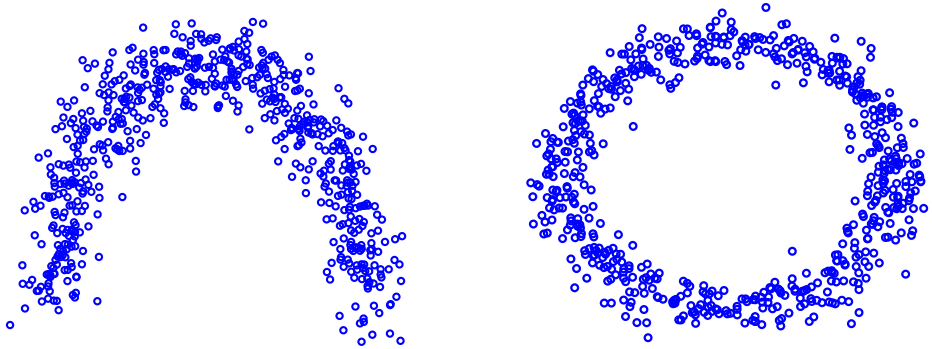
- We can estimate these with the EM-algorithm similarly to hierarchical mixture models

Topics

- Non-parametric density estimation
  - Parzen windows
  - Global, local kernel width

Beyond parametric density models

- More mixture densities



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

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

Non-parametric densities

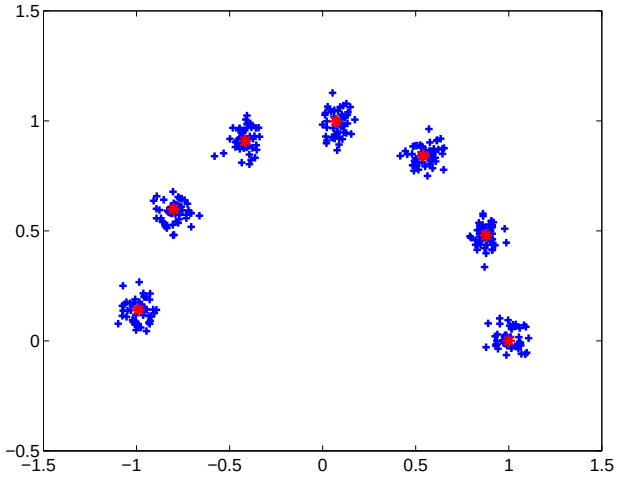
- In the limit, we can center one mixture component (e.g., Gaussian) at each example (Parzen windows):

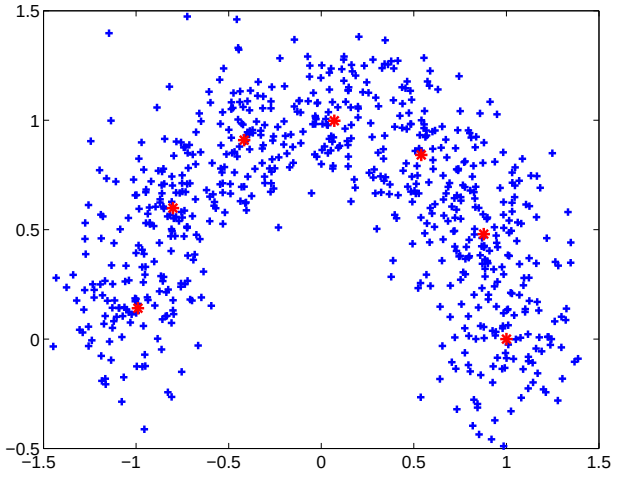
$$\hat{P}_n(\mathbf{x};\sigma) = \frac{1}{n}\sum_{i=1}^{n} P(\mathbf{x}|\mu_i, \sigma^2 I)$$

  where $\mu_i = \mathbf{x}_i,\ i = 1,\ldots,n.$

- The covariance matrices for the components $\Sigma_i = \sigma^2 \cdot I$ are all equal and spherical.

- The single parameter σ now controls the smoothness of the density estimate





One dimensional case

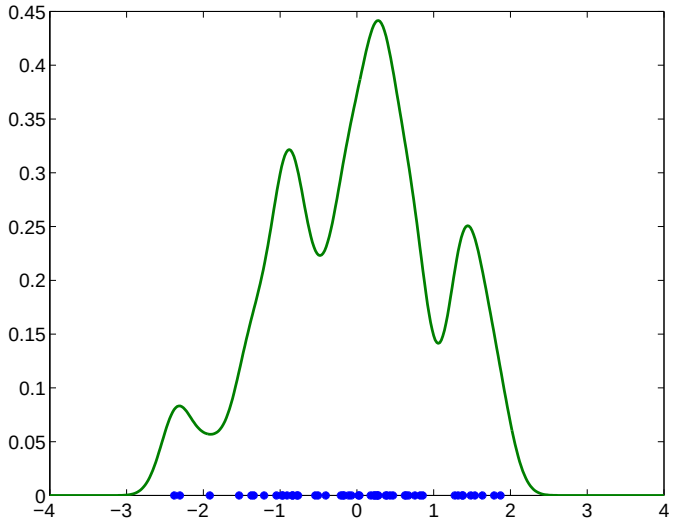
- The non-parametric estimate is typically written as

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

  where $K(z) = \exp(-z^2/2)/\sqrt{2\pi}$ is known as the kernel function (very different from SVM kernels).

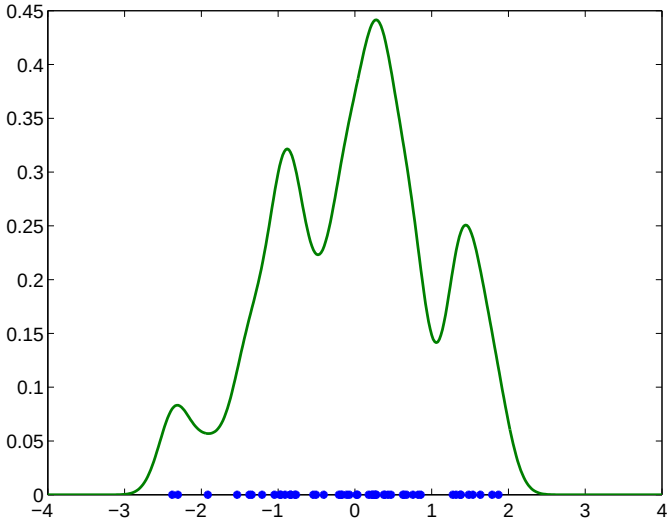
- The kernel width parameter σ controls the smoothness of the estimate

  Example: $n = 50$, $\sigma = 0.02$



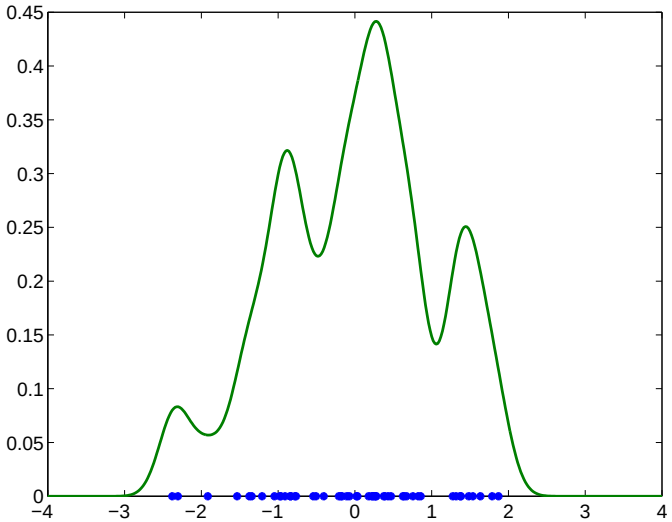
Optimal (global) kernel width

- How do we choose the kernel width (smoothing parameter) σ?



Optimal (global) kernel width

- How do we choose the kernel width (smoothing parameter) σ?



- A general solution: cross-validation

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

  We can find σ that maximizes the log-likelihood of the left-out examples

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

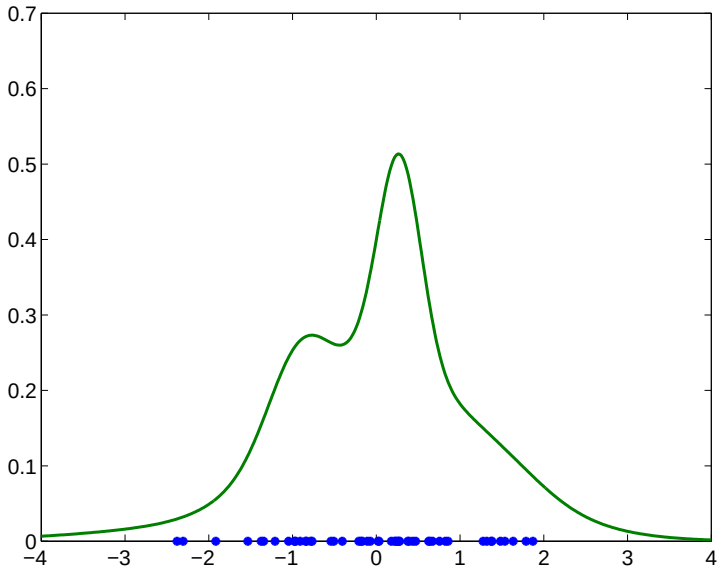
Variable kernel width

- We can also adjust 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

Example: $n = 50$, $k = 10$



How do we choose k in this case?