Machine learning: lecture 15

Tommi S. Jaakkola
MIT CSAIL
tommi@csail.mit.edu

Topics

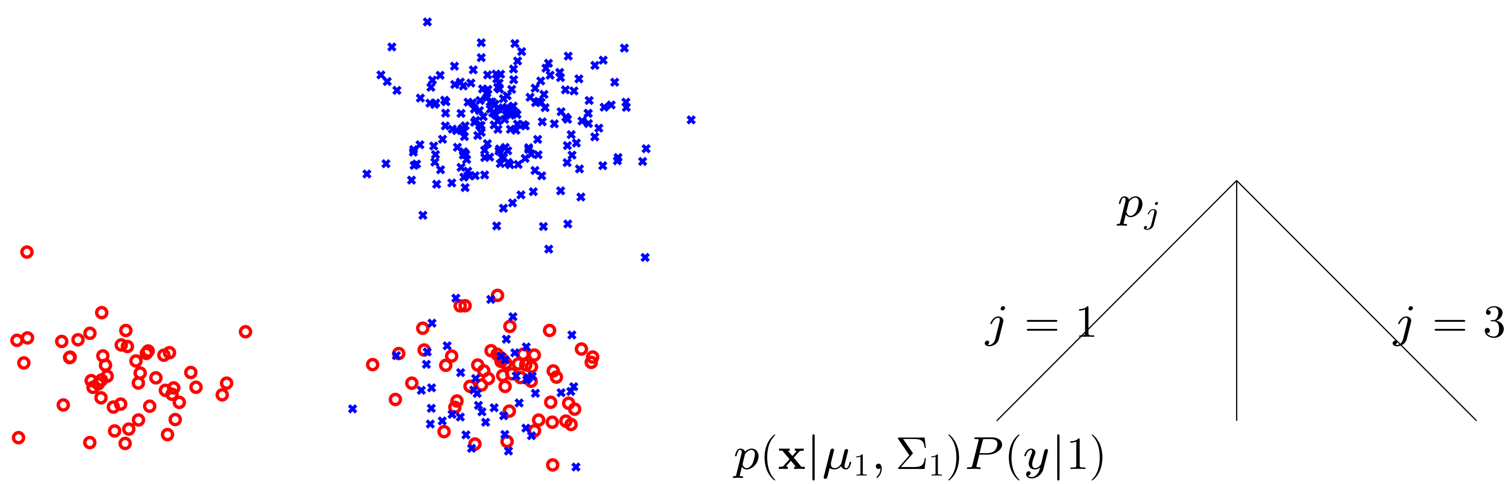
- Finite mixture classifiers
  - class conditional mixtures
  - shared components model
  - conditional mixtures (mixtures of experts)
- Non-parametric mixtures
  - Parzen windows
- Clustering

Shared component mixtures

- We can also formulate (Gaussian) mixture models where the components may be shared across class labels:

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

In other words, each component is associated with a distribution $P(y|j)$ over the class labels (each "type" has a specific distribution over labels)



Shared component mixtures: estimation

- The EM algorithm for these models only requires a few simple modifications

  E-step: the posterior assignments (responsibilities) now also depend on the class labels:

  $$\hat{p}(j|i) = \frac{p_j^{(k)} p(\mathbf{x}_i|\mu_j^{(k)}, \Sigma_j^{(k)}) P^{(k)}(y_i|j)}{p(\mathbf{x}_i, y_i|\theta^{(k)})}$$

  M-step: remains the same for the Gaussian components; in addition, we update each $P(y|j)$ according to a weighted frequency of labels assigned to component j:

  $$P^{(k+1)}(y|j) = \frac{1}{\hat{n}_j} \sum_{i=1}^{n} \hat{p}(j|i)\delta(y_i, y)$$

  where $\delta(y_i, y) = 1$ if $y_i = y$ and zero otherwise.

Shared component mixtures: example

Topics

- Finite mixture classifiers
  - class conditional mixtures
  - shared components model
  - conditional mixtures (mixtures of experts)
- Non-parametric mixtures
  - Parzen windows
- Clustering

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 mixtures we have seen so far, the selection of which expert to rely on now depends on the context (the input $\mathbf{x}$); we will call such models *mixtures of experts*.

Experts (regression)

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

$$p(y|\mathbf{x}, \theta_j) = N(y;\, \mathbf{w}_{j1}^T\mathbf{x} + w_{j0},\, \sigma_j^2)$$

where

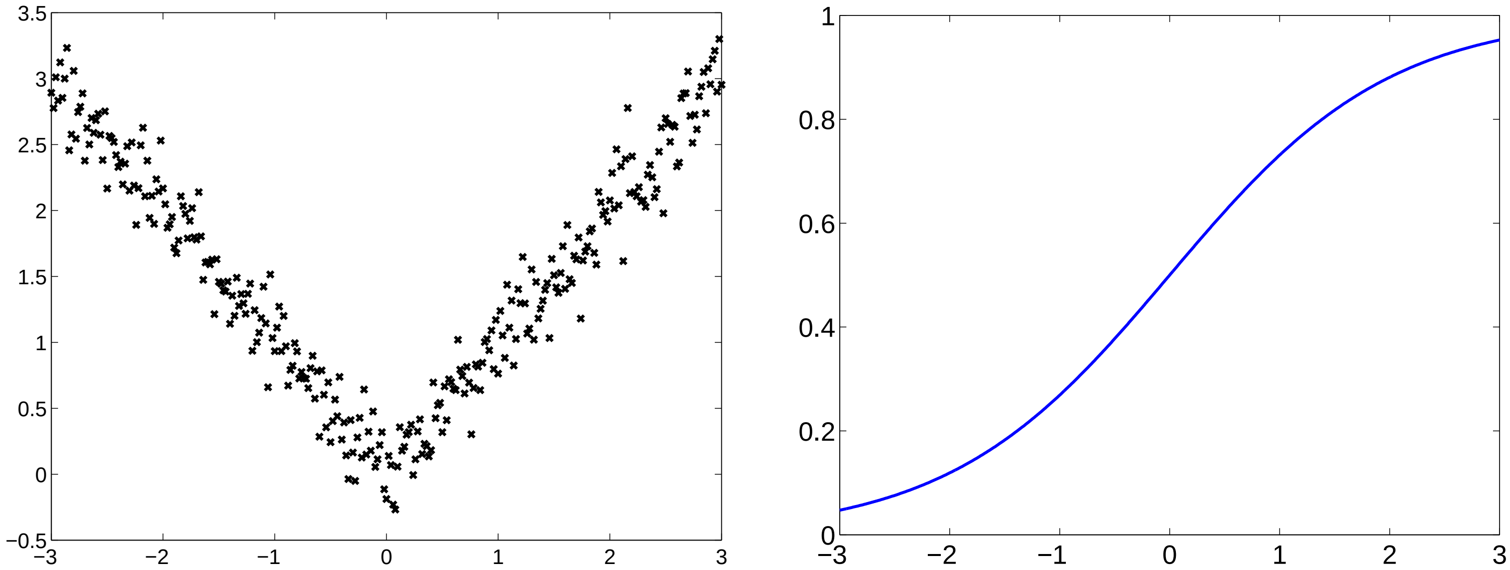
$$\begin{aligned} \text{mean of } y \text{ given } \mathbf{x} &= \mathbf{w}_{j1}^T\mathbf{x} + w_{j0} \\ \text{variance of } y \text{ given } \mathbf{x} &= \sigma_j^2 \end{aligned}$$

$\theta_j = \{\mathbf{w}_{j1}, w_{j0}, \sigma_j^2\}$ denotes the parameters of the j^{th} expert.

- We need to find an appropriate (input dependent) way of allocating tasks to these experts

Mixtures of experts

Example:



- Here we need to switch from one linear regression model to another at $x = 0$; the switch can be probabilistic.

Gating network

- A *gating network* specifies a distribution over m experts, conditionally on the input $\mathbf{x}$
- Example: when there are just two experts the gating network can be a logistic regression model

$$P(j = 1|\mathbf{x}, \eta) = g(\, \mathbf{v}_1^T \mathbf{x} + v_0 \,)$$

where $\eta = (\mathbf{v}_1, v_0)$ and $g(z) = (1 + e^{-z})^{-1}$ is the logistic function.

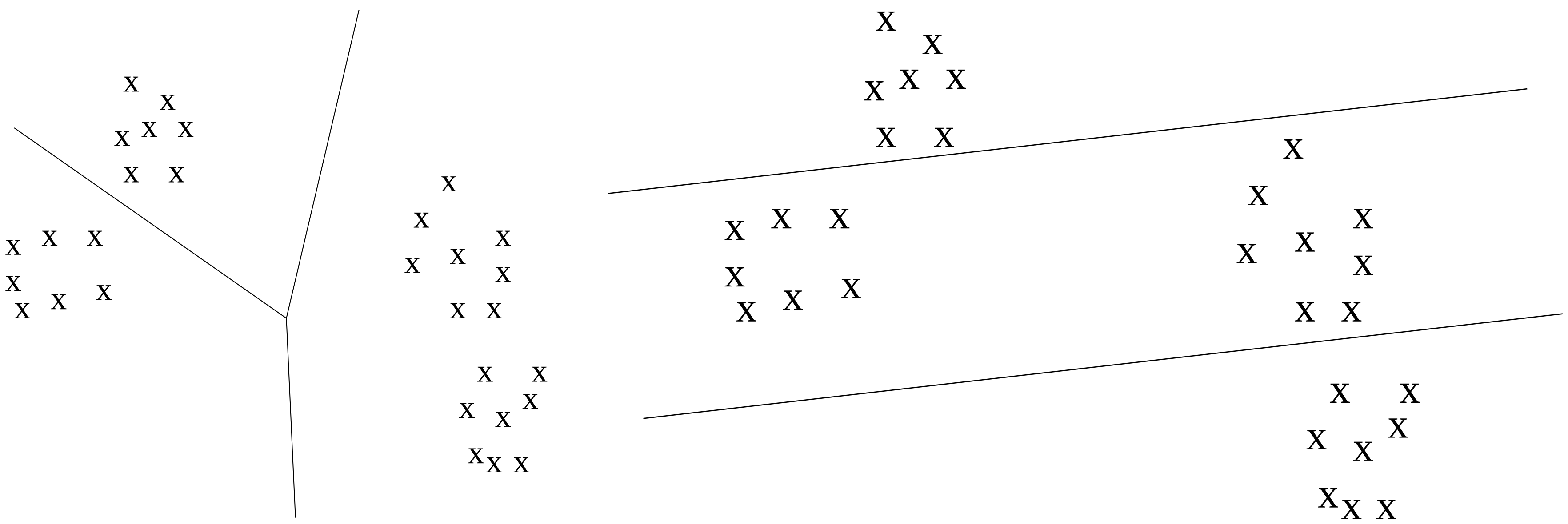
- For $m > 2$, the gating network can be a softmax model

$$P(j|\mathbf{x}, \eta) = \frac{\exp(\, \mathbf{v}_{j1}^T \mathbf{x} + v_{j0} \,)}{\sum_{j'=1}^{m} \exp(\, \mathbf{v}_{j'1}^T \mathbf{x} + v_{j'0} \,)}$$

where $\eta = \{\mathbf{v}_{11}, \ldots, \mathbf{v}_{1m}, v_{10}, \ldots, v_{m0}\}$ denotes the parameters of the gating network

Gating network: example divisions

$$P(j|\mathbf{x},\eta) = \frac{\exp(\,\mathbf{v}_{j1}^T\mathbf{x} + v_{j0}\,)}{\sum_{j'=1}^{m}\exp(\,\mathbf{v}_{j'1}^T\mathbf{x} + v_{j'0}\,)}$$

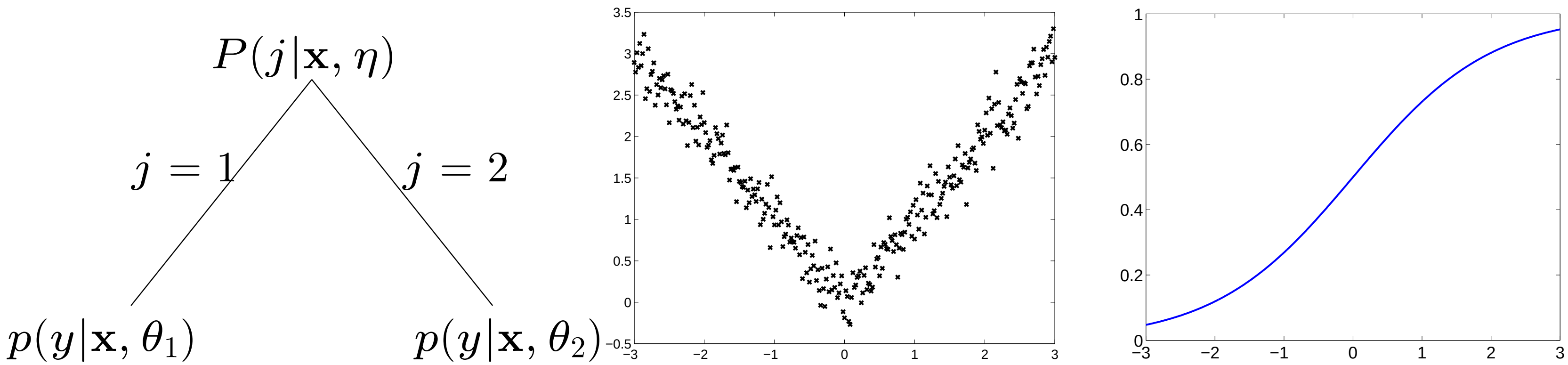


A mixture of experts model

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

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

where η specifies the parameters of the gating network (e.g., logistic) and θ_j gives the parameters of the j^{th} expert (e.g., linear regression model).



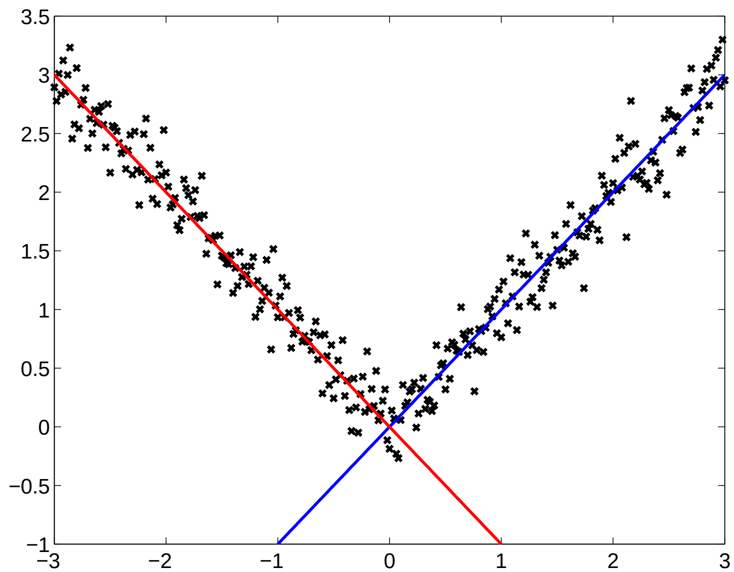
- The selection of experts is made conditionally on the input

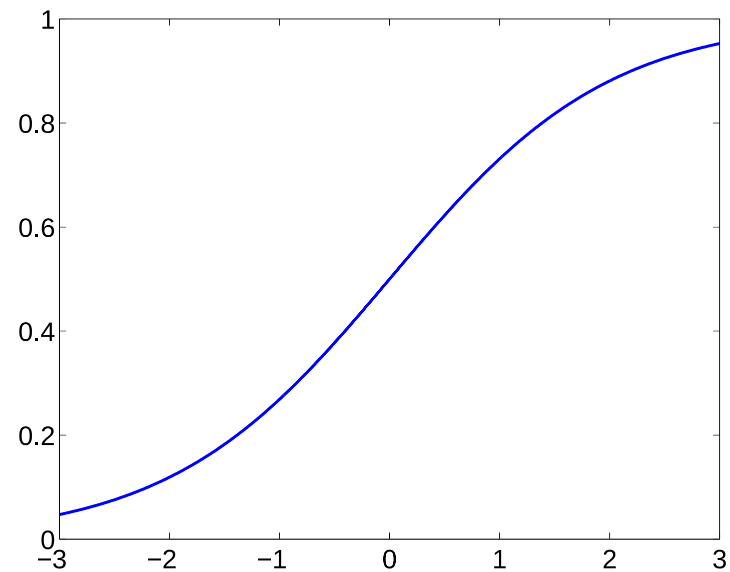
A mixture of experts model: estimation

- 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:

$$\hat{p}(j|i) = P(j|\mathbf{x}_i, y_i, \eta^{(k)}, \theta^{(k)})$$

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





EM for mixtures of experts

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

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

1. For each expert j: find $\theta_j^{(k+1)}$ that maximize

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

   (e.g., linear regression, weighted training set)

2. For the gating network: find $\eta^{(k+1)}$ that maximize

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

   (e.g., logistic regression, weighted training set)

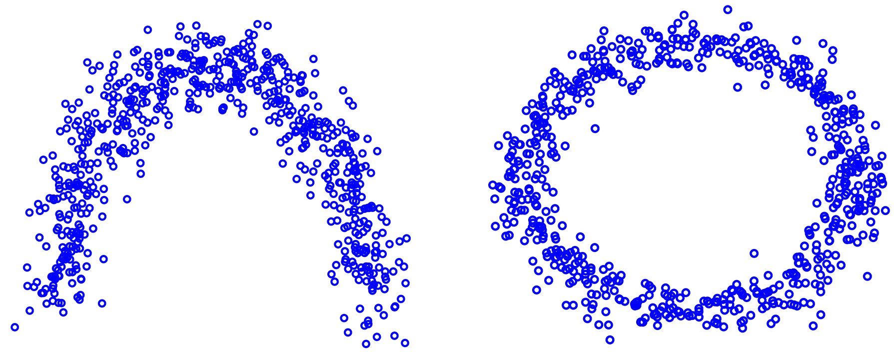
Mixtures of experts: demo

Topics

- Finite mixture classifiers
  - class conditional mixtures
  - shared components model
  - conditional mixtures (mixtures of experts)
- Non-parametric mixtures
  - Parzen windows
- Clustering

Beyond parametric density models

- More mixture densities



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

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

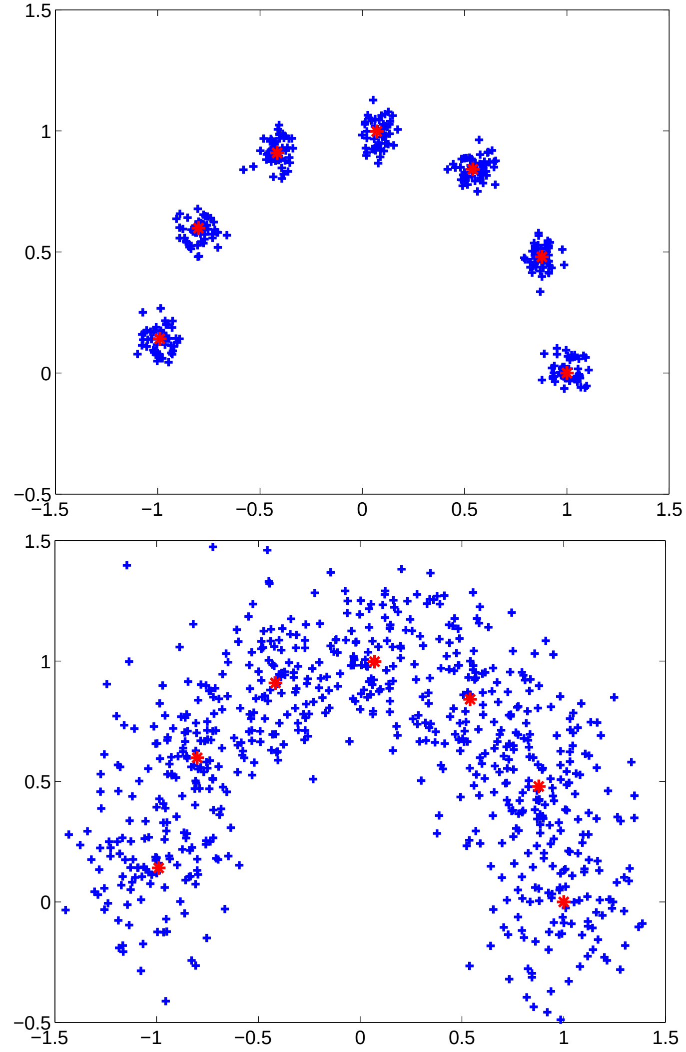
Non-parametric densities

- 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.

The single parameter σ^2 controls the smoothness of the resulting density estimate



1-dim case: Parzen windows

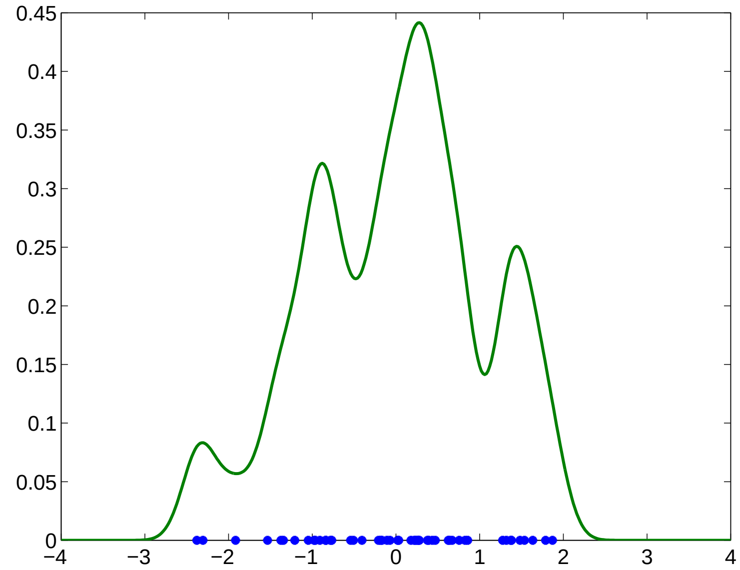
- We place a smooth Gaussian (or other) bump on each training example

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

where the "kernel function" is

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

(very different from SVM kernels).



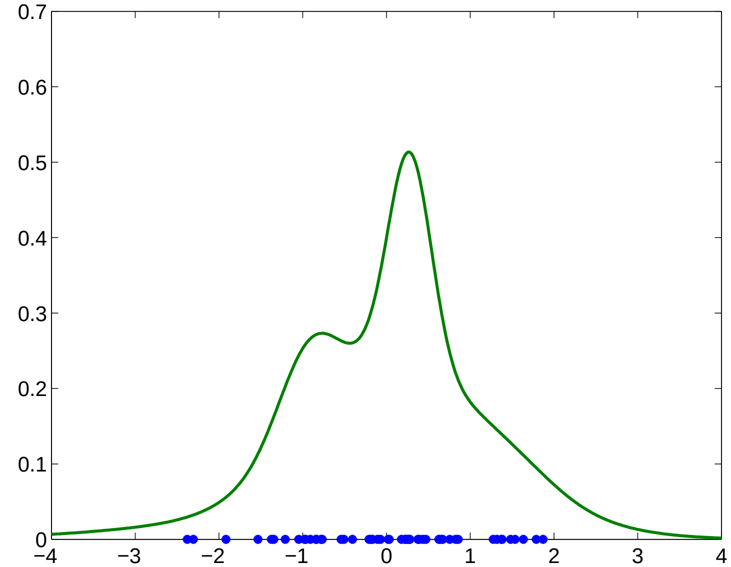
$n = 50,\ \sigma = 0.02$

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

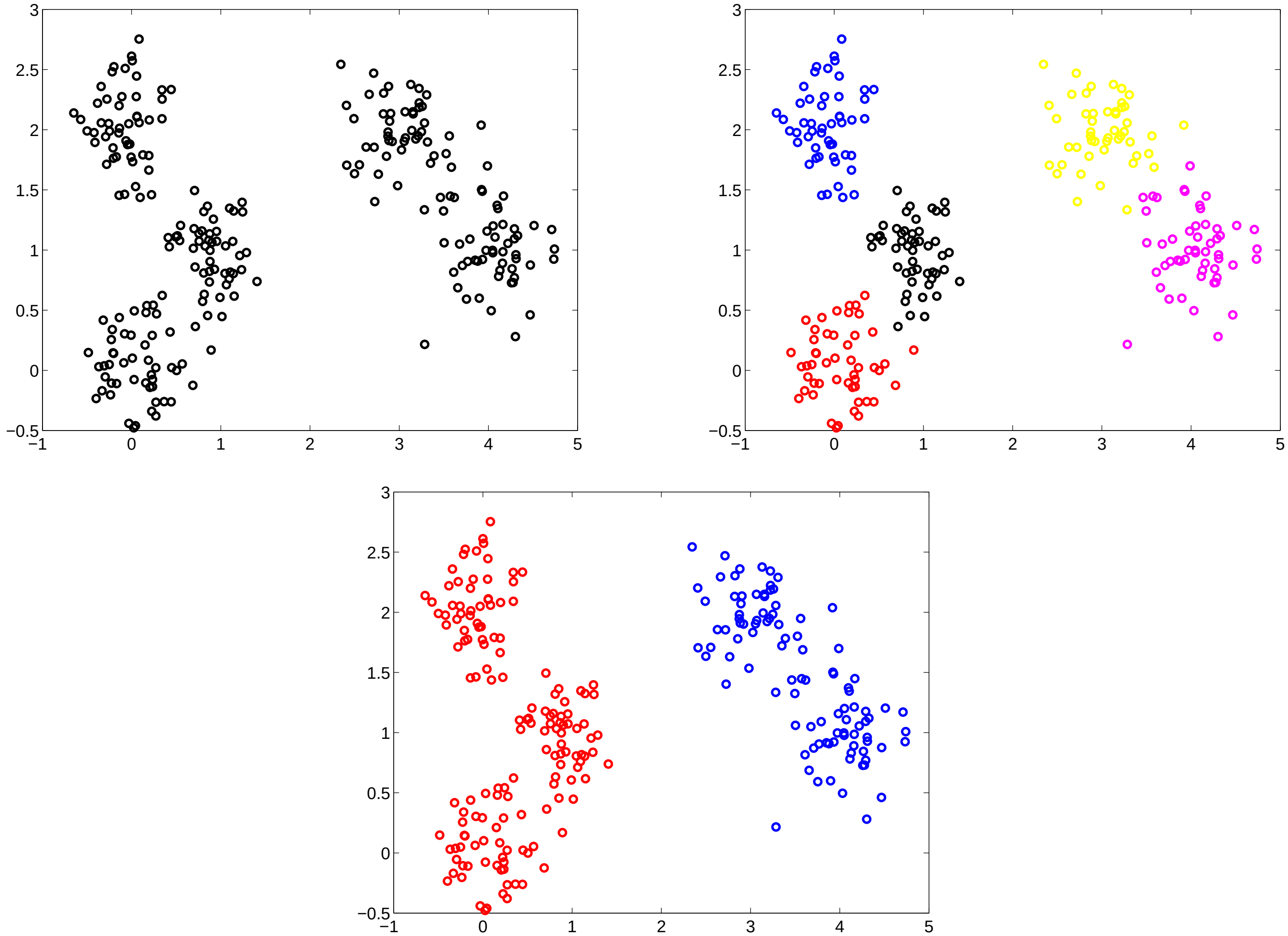
$$CV(\sigma) = \sum_{i=1}^{n} \log \hat{p}_{-i}(x_i;\sigma)$$

Topics

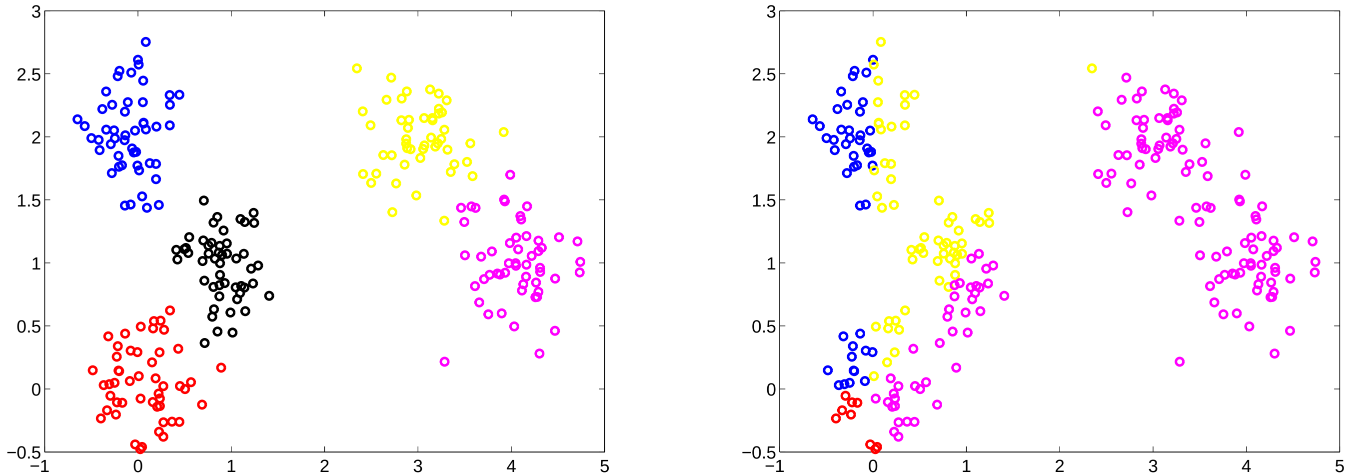
- Finite mixture classifiers
  - class conditional mixtures
  - shared components model
  - conditional mixtures (mixtures of experts)
- Non-parametric mixtures
  - Parzen windows
- Clustering

Finding structure in the data: clustering

- We can find structure in the data by isolating groups of examples that are similar in some well-defined sense



Clustering: metric



- Clustering results are crucially dependent on the measure of similarity (or distance) between the "points" to be clustered