CS540 Machine learning
Lecture 14
Mixtures, EM,
Non-parametric models

Outline

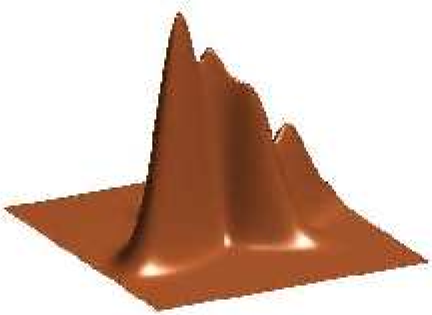
- Mixture models
- EM for mixture models
- K means clustering
- Conditional mixtures
- Kernel density estimation
- Kernel regression

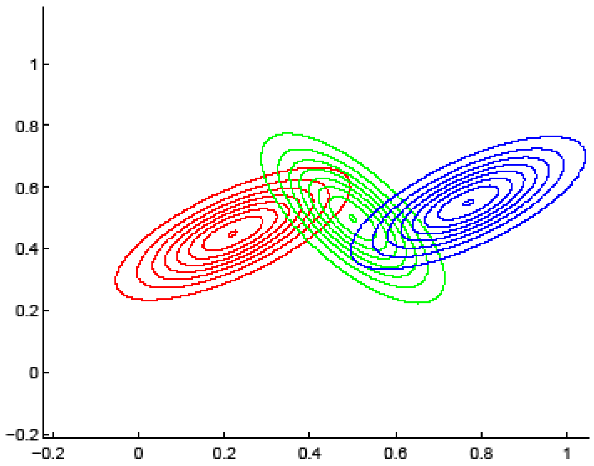
Gaussian mixture models

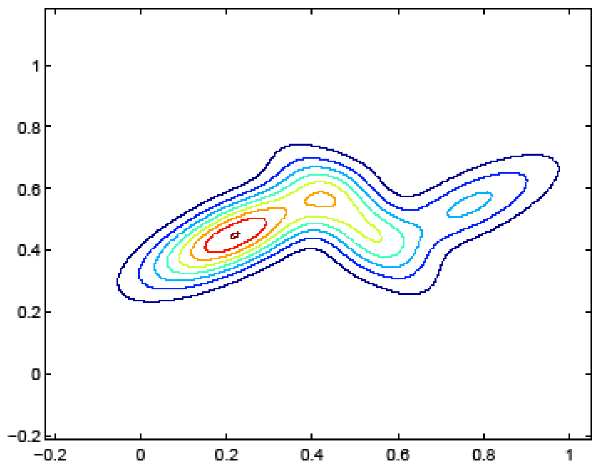
- GMM

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{k=1}^{K} p(z=k|\boldsymbol{\pi})p(\mathbf{x}|z=k,\boldsymbol{\phi}_k)$$

$$p(\mathbf{x}|z=k,\boldsymbol{\phi}_k) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)$$



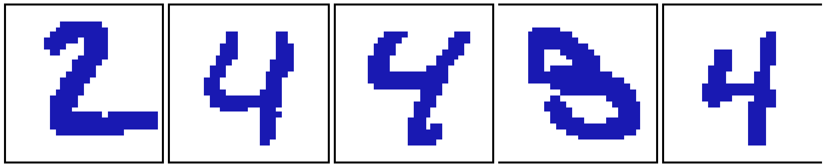


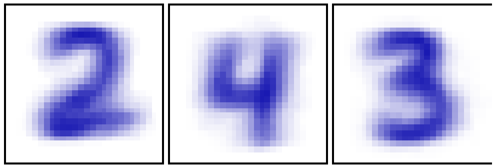


Bernoulli mixture models

- BMM

$$
\begin{aligned}
p(\mathbf{x}|\boldsymbol{\theta}) &= \sum_{k=1}^{K} p(z=k|\boldsymbol{\pi}) p(\mathbf{x}|z=k, \boldsymbol{\phi}_k) \\
p(\mathbf{x}|z=k, \boldsymbol{\mu}_k) &= \prod_{j=1}^{d} \text{Ber}(x_j|\mu_{j,k})
\end{aligned}
$$

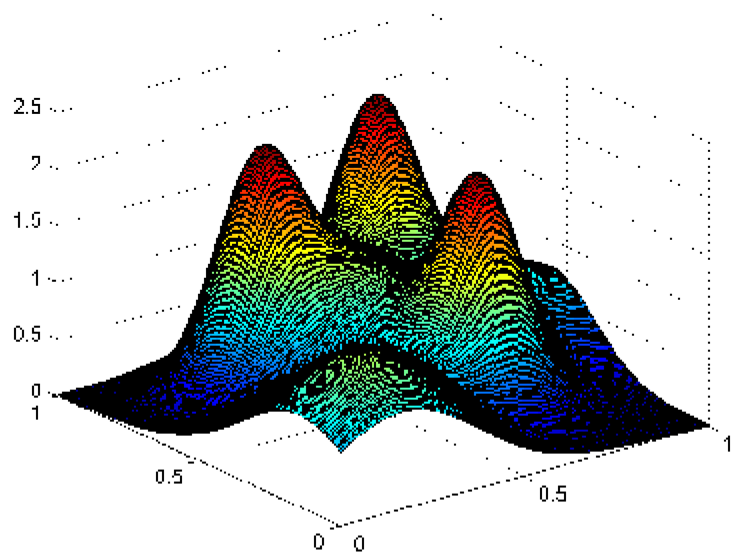




MLE for mixture models

- Hard to compute. Can find local maximum using gradient methods.

$$\ell(\theta) \quad = \quad \log p(\mathbf{x}_{1:n}|\boldsymbol{\theta}) = \sum_i \log p(\mathbf{x}_i|\boldsymbol{\theta}) = \sum_i \log \sum_{z_i} p(\mathbf{x}_i, z_i|\boldsymbol{\theta})$$



Outline

- Mixture models
- EM for mixture models
- K means clustering
- Conditional mixtures
- Kernel density estimation
- Kernel regression

Expectation Maximization

- EM is an algorithm for finding MLE or MAP for problems with hidden variables
- Key intuition: if we knew what cluster each point belonged to (i.e., the z_i variables), we could partition the data and find the MLE for each cluster separately
- E step: infer responsibility of each cluster for each data point

$$r_{ik} \quad = \quad p(z_i = k|\boldsymbol{\theta}, \mathcal{D})$$

- M step: optimize parameters using “filled in” data z
- Repeat until convergence

Expected complete data loglik

- Complete data loglik

$$
\begin{aligned}
\ell_c(\boldsymbol{\theta}) &= \log p(\mathbf{x}_{1:n}, z_{1:n}|\boldsymbol{\theta}) \\
&= \log \prod_i p(z_i|\boldsymbol{\pi}) p(\mathbf{x}_i|z_i, \boldsymbol{\phi}) \\
&= \log \prod_i \prod_k [\pi_k p(\mathbf{x}_i|\boldsymbol{\phi}_k)]^{I(z_i=k)} \\
&= \sum_i \sum_k I(z_i = k)[\log \pi_k + \log p(\mathbf{x}_i|\boldsymbol{\phi}_k)]
\end{aligned}
$$

- Expected complete data loglik

$$
\begin{aligned}
Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) &\stackrel{\text{def}}{=} E \sum_i \log p(\mathbf{x}_i, z_i|\boldsymbol{\theta}) \\
&= E \sum_i \sum_k I(z_i = k) \log[\pi_k p(\mathbf{x}_i|\boldsymbol{\phi}_k)] \\
&= \sum_i \sum_k p(z_i|\mathbf{x}_i, \theta^{old}) \log[\pi_k p(\mathbf{x}_i|\boldsymbol{\phi}_k)]
\end{aligned}
$$

EM for mixture models

- E step: compute responsibilities

$$r_{ik} \stackrel{\text{def}}{=} p(z_i = k|\mathbf{x}_i, \boldsymbol{\theta}^{old})$$

- M step: maximize wrt θ

$$\begin{aligned} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) &\stackrel{\text{def}}{=} E \sum_i \log p(\mathbf{x}_i, z_i|\boldsymbol{\theta}) \\ &= E \sum_i \sum_k I(z_i = k) \log[\pi_k p(\mathbf{x}_i|\boldsymbol{\phi}_k)] \\ &= \sum_i \sum_k p(z_i|\mathbf{x}_i, \theta^{old}) \log[\pi_k p(\mathbf{x}_i|\boldsymbol{\phi}_k)] \end{aligned}$$

$$\begin{aligned} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) &= \sum_i \sum_k r_{ik} \log \pi_k + \sum_k \sum_i r_{ik} \log p(\mathbf{x}_i|\boldsymbol{\phi}_k) \\ &= J(\boldsymbol{\pi}) + \sum_k J(\boldsymbol{\phi}_k) \end{aligned}$$

EM for GMM

- E step

$$r_{ik} = p(z_i = k|\mathbf{x}_i, \boldsymbol{\theta}^{old}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k'} \pi_{k'} \mathcal{N}(\mathbf{x}_i|\mu_{k'}, \Sigma_{k'})}$$

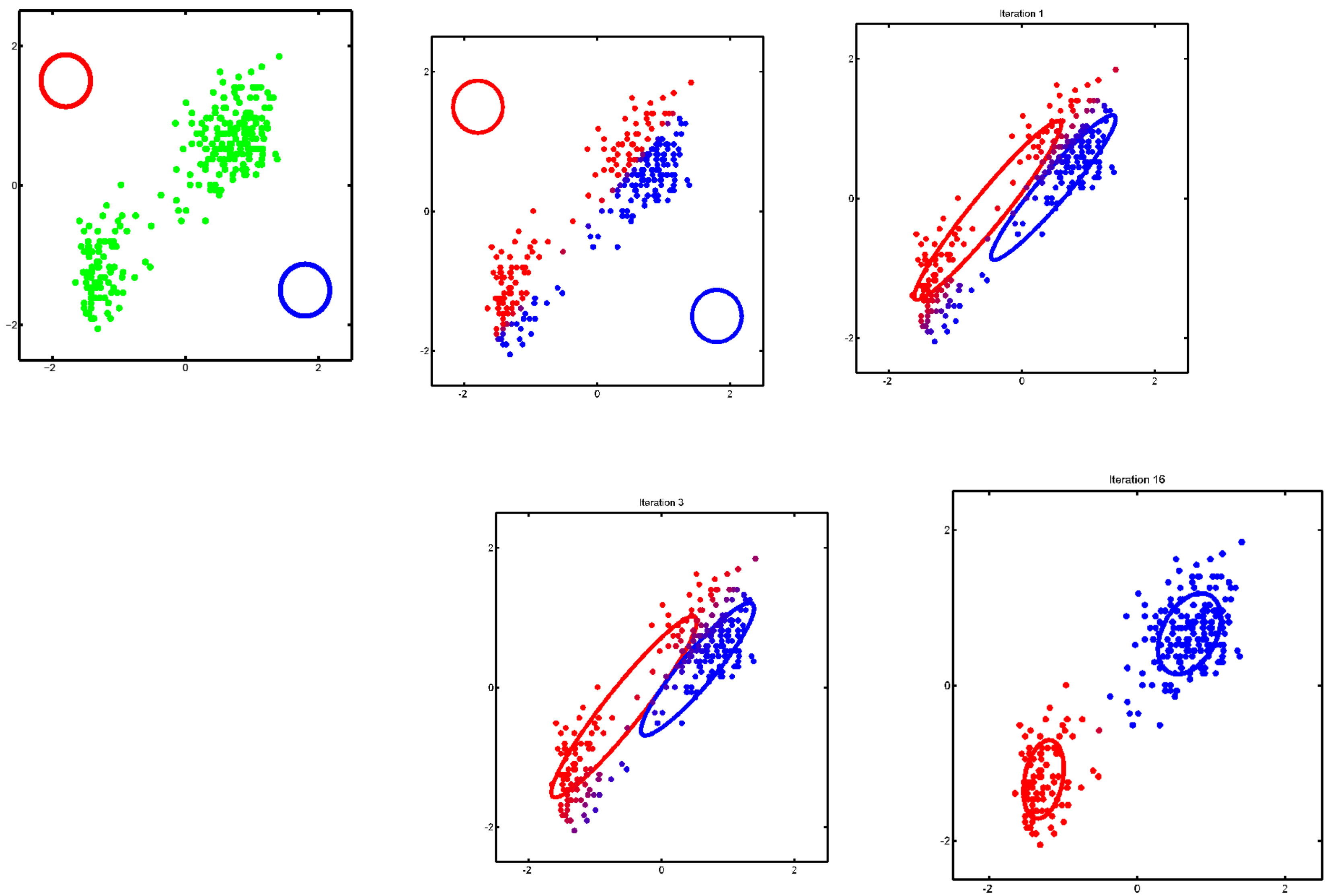
- M step

$$\begin{aligned} 0 &= \frac{\partial}{\partial \pi_j}[\sum_i \sum_k r_{ik} \log \pi_k + \lambda(1 - \sum_k \pi_k)] \\ \pi_k &= \frac{1}{n}\sum_i r_{ik} = \frac{r_k}{n} \end{aligned}$$

M step for mu, Sigma

$$
\begin{aligned}
J(\mu_k, \Sigma_k) &= -\frac{1}{2} \sum_i r_{ik} \left[\log |\boldsymbol{\Sigma}_k| + (\mathbf{x}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_k) \right] \\
0 &= \frac{\partial}{\partial \boldsymbol{\mu}_k} J(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \\
\boldsymbol{\mu}_k &= \frac{\sum_i r_{ik} \mathbf{x}_i}{\sum_i r_{ik}} \\
\boldsymbol{\Sigma}_k &= \frac{\sum_i r_{ik} (\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^T}{\sum_i r_{ik}} \\
&= \frac{\sum_i r_{ik} \mathbf{x}_i \mathbf{x}_i^T - r_k \boldsymbol{\mu}_k \boldsymbol{\mu}_k^T}{r_k}
\end{aligned}
$$

EM for GMM



EM for mixtures of Bernoullis

- E step

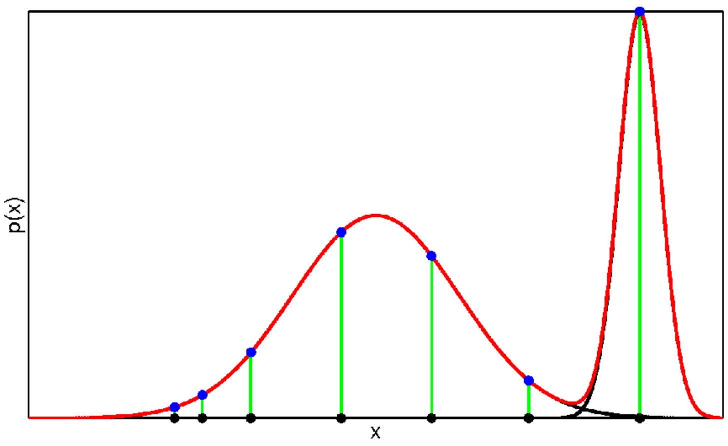
$$r_{ik} = \frac{\pi_k p(\mathbf{x}_i|\boldsymbol{\mu}_k)}{\sum_{k'} \pi_{k'} p(\mathbf{x}_i|\boldsymbol{\mu}_{k'})}$$

- M step

$$\boldsymbol{\mu}_k^{new} = \frac{\sum_{i=1}^n r_{ik}\mathbf{x}_i}{\sum_{i=1}^n r_{ik}}$$

Singularities in GMM

- Can maximize likelihood by letting σ_k -> 0 (overfitting)



EM for MAP for GMMs

- Maximize expected complete data log likelihood plus log prior

$$
\begin{aligned}
J(\boldsymbol{\theta}) &= \left[\sum_i \sum_k r_{ik} \log p(\mathbf{x}_i | z_i = k, \boldsymbol{\theta})\right] + \log p(\boldsymbol{\pi}) + \sum_k \log p(\boldsymbol{\phi}_k) \\
\boldsymbol{\pi} &\sim \text{Dir}(\alpha \mathbf{1}) \\
\pi_k &= \frac{\sum_i r_{ik} + \alpha - 1}{n + K\alpha - K} \\
p(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) &= NW(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k | \mathbf{m}_k, \eta_k, \mathbf{S}_k, \boldsymbol{\nu}_k) = \mathcal{N}(\boldsymbol{\mu}_k | \mathbf{m}_k, \eta_k \boldsymbol{\Lambda}_k) \text{Wi}(\boldsymbol{\Lambda}_k | \boldsymbol{\nu}_k, \mathbf{S}_k) \\
\boldsymbol{\mu}_k &= \frac{\eta_k \mathbf{m}_k \sum_i r_{ik} \mathbf{x}_i}{\eta_k + \sum_i r_{ik}} \\
\boldsymbol{\Lambda}_k^{-1} &= \frac{\sum_i r_{ik} (\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^T}{\sum_i r_{ik} + \nu_k - d} + \frac{\eta_k (\boldsymbol{\mu}_k - \mathbf{m}_k)(\boldsymbol{\mu}_k - \mathbf{m}_k)^T + \mathbf{S}_k}{\sum_i r_{ik} + \nu_k - d}
\end{aligned}
$$

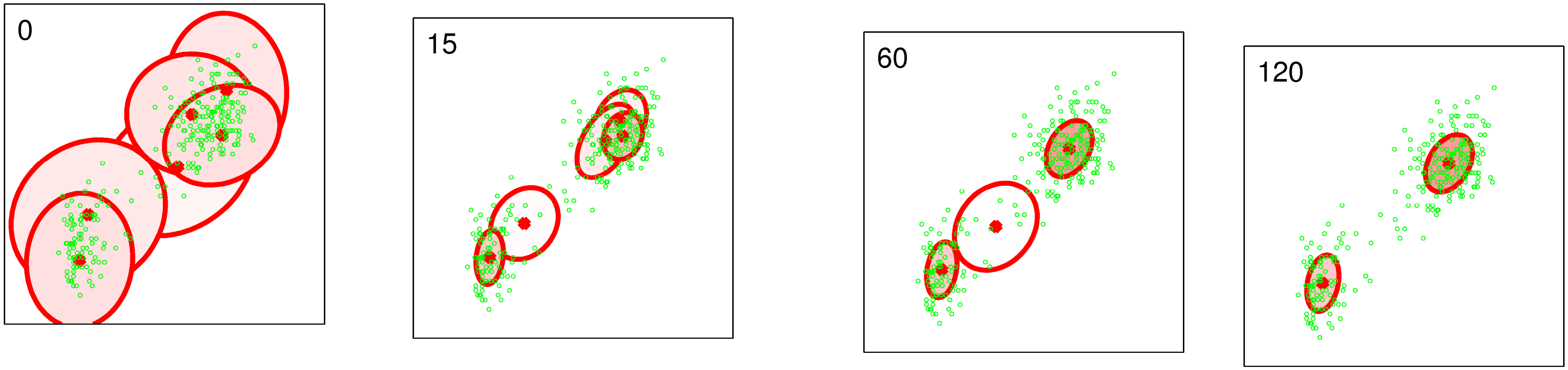
Setting hyper-parameters

- Can set $\alpha_k = 1$ (see later)
- $m_0 = 0$, $\eta_k=0$ (improper)
  $\nu_\kappa = d+2$, S_k depends on data scale

$$
\begin{aligned}
E[\mathbf{\Lambda}_k^{-1}] &= \frac{\mathbf{S}_k}{\nu_k - d - 1} \\
\mathbf{S}_k &= (\nu_k - d - 1)\frac{\hat{\sigma}^2}{K}\mathbf{I}_d \\
\mathbf{S}_k &= (\nu_k - d - 1)\frac{1}{K}\mathrm{diag}(\hat{\boldsymbol{\sigma}}_1{}^2, \ldots, \hat{\boldsymbol{\sigma}}_d{}^2)
\end{aligned}
$$

Choosing K

- Search over K, score with CV or BIC
- Or set $\alpha_k \approx 0$, and run EM once; unneeded components get responsibility 0



Other issues

- Standard to do multiple random restarts, and return best of T tries to avoid local optima
- For GMMs, common to initialize using K-means or set each μ_k to one of the data points; otherwise, random initial params
- Convergence declared if params stop changing, or if the observed data loglik stops increasing

EM for other models

- Latent variable models eg PPCA, HMMs
- MLE of a single MVN with missing values
- MLE of scale mixture model (eg student T, Lasso)
- Empirical Bayes
- Etc.

Outline

- Mixture models
- EM for mixture models
- K means clustering
- Conditional mixtures
- Kernel density estimation
- Kernel regression

K-means clustering

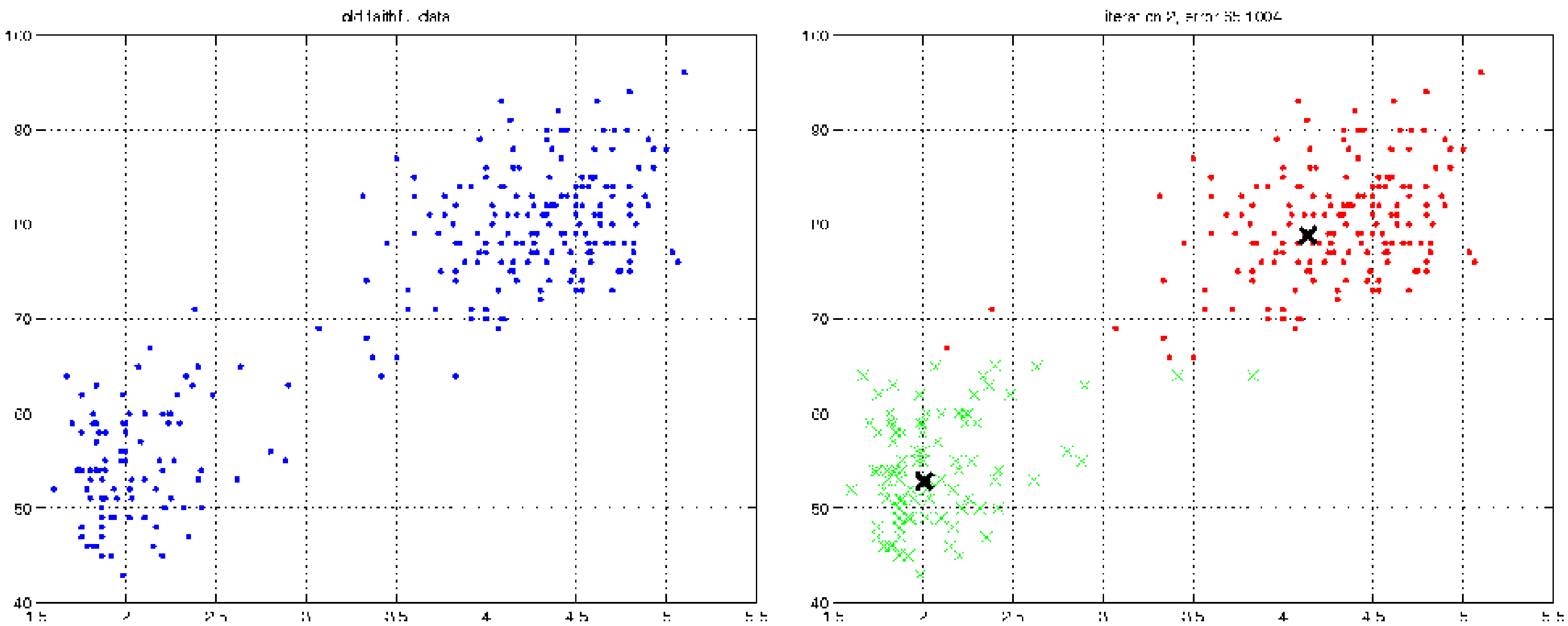
- GMM with $\Sigma_k = \sigma^2 I_d$, $\pi_k = 1/K$, only μ_k is learned
- In E step, use hard assignment (can use kd-trees to speed this up)

Algorithm 1: K-means algorithm

---

initialize $\mathbf{m}_k$, $k \leftarrow 1$ **to** K
repeat
  Assign each data point to its closest cluster center: $z_i = \arg\min_k ||\mathbf{x}_i - \mathbf{m}_k||^2$
  Update each cluster center by computing the mean of all points assigned to it: $\mathbf{m}_k = \frac{1}{n_k} \sum_{i:z_i=k} \mathbf{x}_i$
until *converged*

Vector quantization

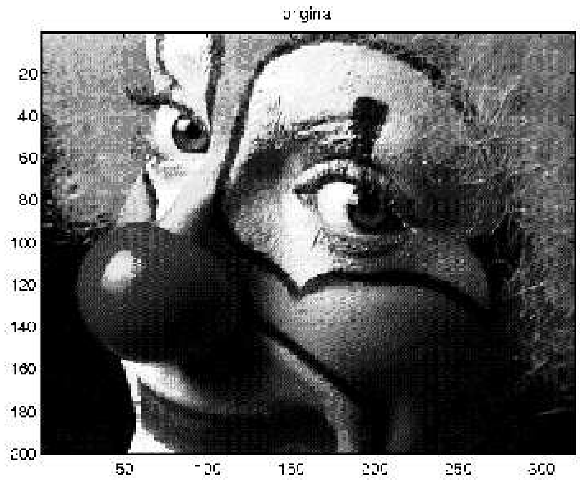


Replace each $x_i \in R^2$ with a codeword z_i in $\{1,..,K\}$

This is an index into the codebook $m_1, m_2, \ldots, m_K$ in R^2

K-means minimizes the distortion

$$J = \frac{1}{n}\sum_{i=1}^{n} ||\mathbf{x}_i - \text{decode}(\text{encode}(\mathbf{x}_i))||^2 = \frac{1}{n}\sum_{i=1}^{n} ||\mathbf{x}_i - \mathbf{m}_{z_i}||^2$$



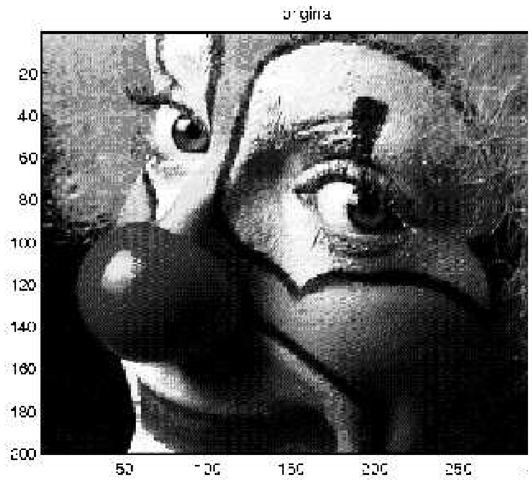
Original

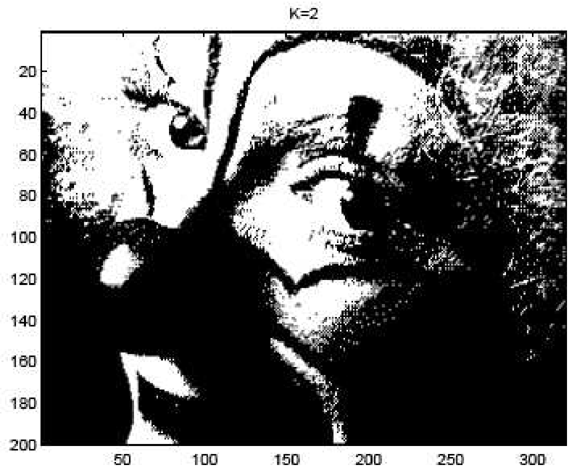
K=2

K=4

K-means minimizes the distortion

$$J = \frac{1}{n}\sum_{i=1}^{n} ||\mathbf{x}_i - \mathrm{decode}(\mathrm{encode}(\mathbf{x}_i))||^2 = \frac{1}{n}\sum_{i=1}^{n} ||\mathbf{x}_i - \mathbf{m}_{z_i}||^2$$





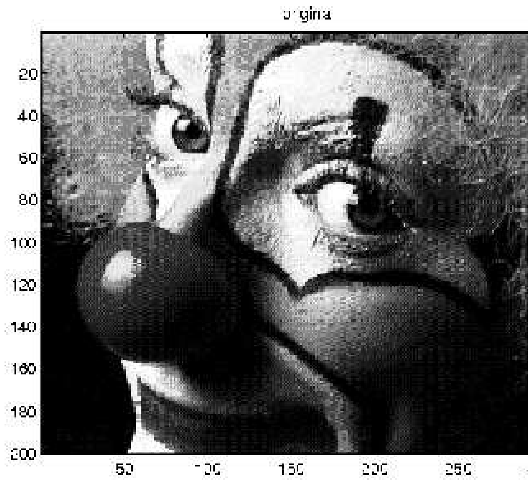
Original

K=2

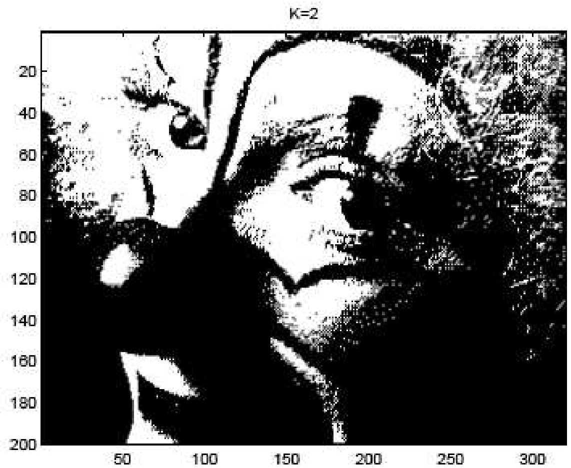
K=4

K-means minimizes the distortion

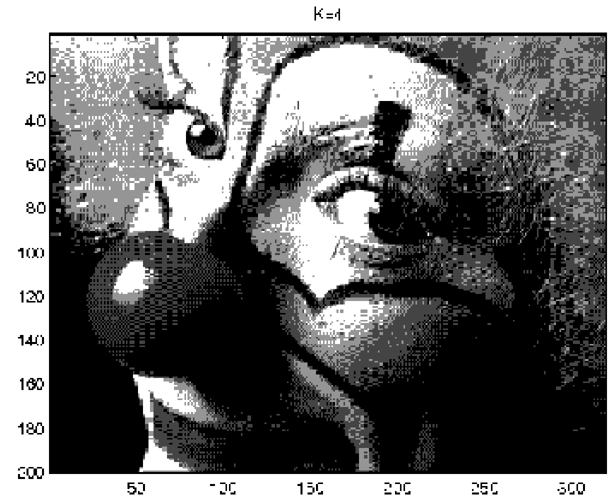
$$J = \frac{1}{n}\sum_{i=1}^{n} ||\mathbf{x}_i - \mathrm{decode}(\mathrm{encode}(\mathbf{x}_i))||^2 = \frac{1}{n}\sum_{i=1}^{n} ||\mathbf{x}_i - \mathbf{m}_{z_i}||^2$$



Original



K=2



K=4

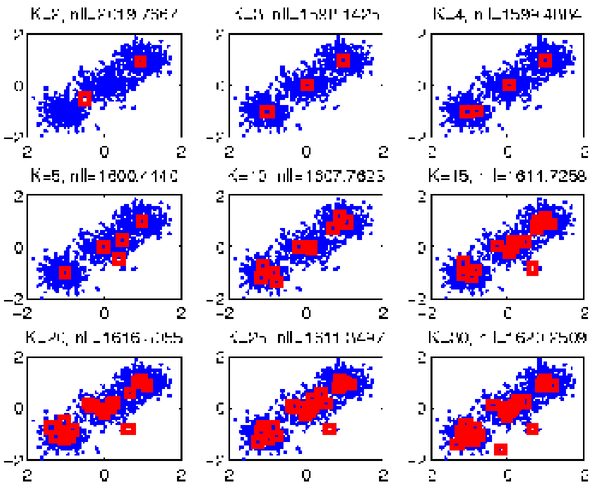
K-medoids

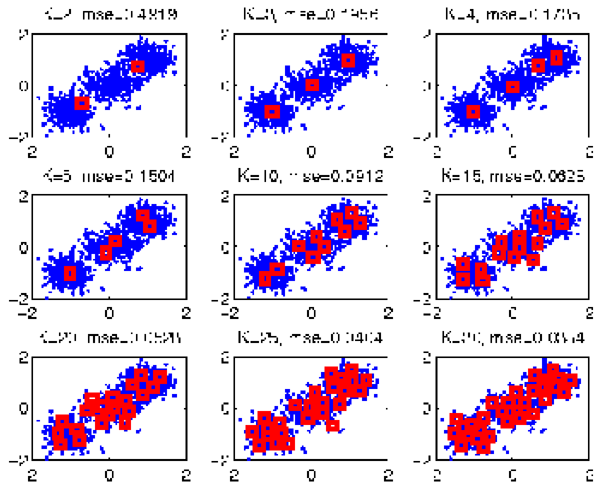
- Each cluster is represented by a single data point (prototype), rather than an average of many data points

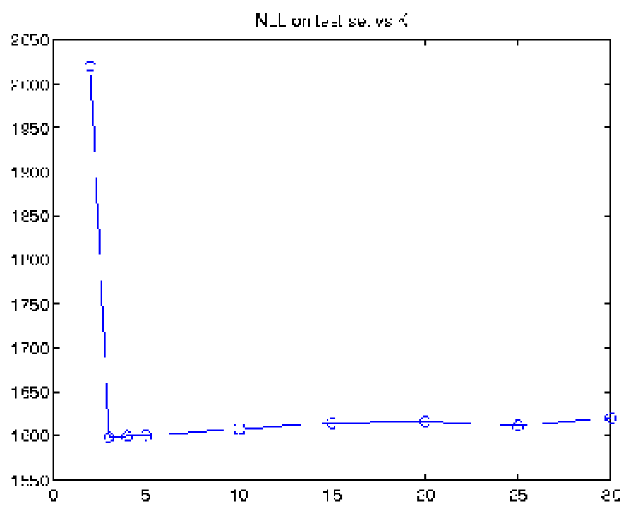
Algorithm 1: K-medoids algorithm

initialize $m_{1:K}$ as a random subset of size K from $\{1, \ldots, n\}$
repeat
    Assign each data point to its closest prototype: $z_i = \arg\min_k D(i, m(k))$
    For each cluster k, pick as prototype the point that is closest to all others: $m_k \leftarrow \arg\min_{i:z_i=k} \sum_{i':z_{i'}=k} d_{i,i'}$
until *converged*

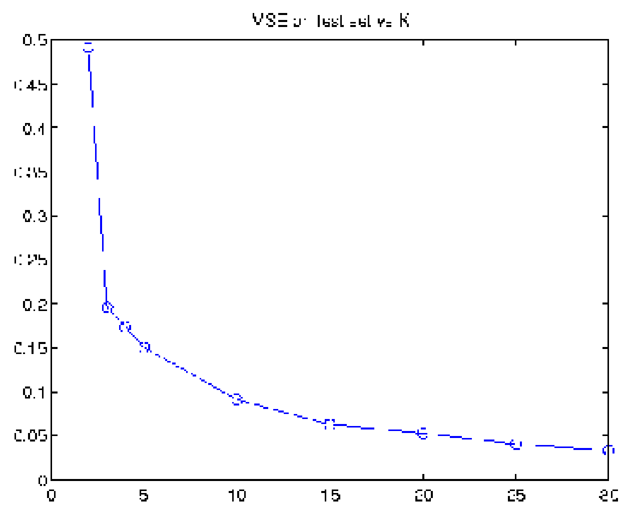
EM vs K-means







Test error vs K

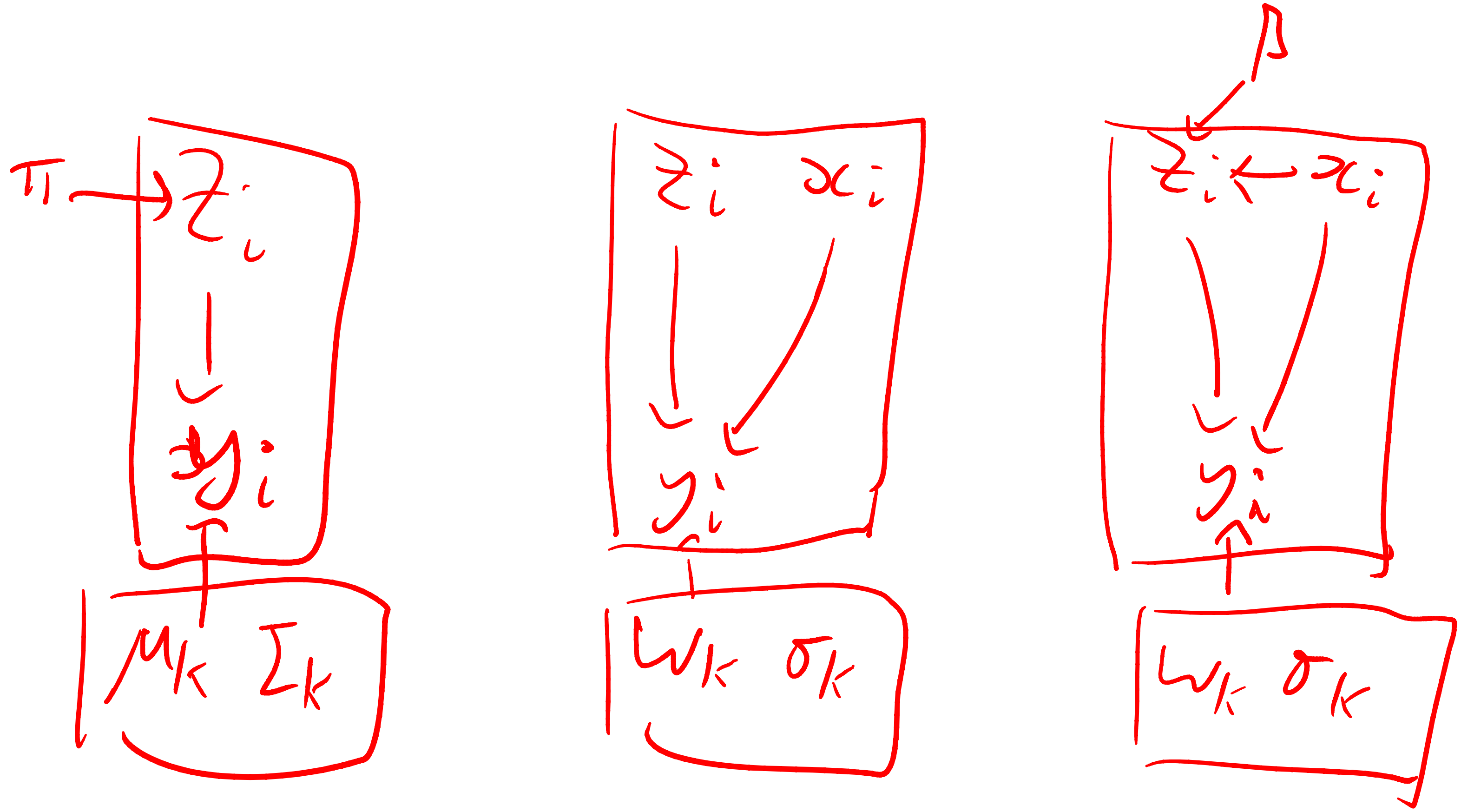


Cannot use CV to select K for K-means!!

Outline

- Mixture models
- EM for mixture models
- K means clustering
- Conditional mixtures
- Kernel density estimation
- Kernel regression

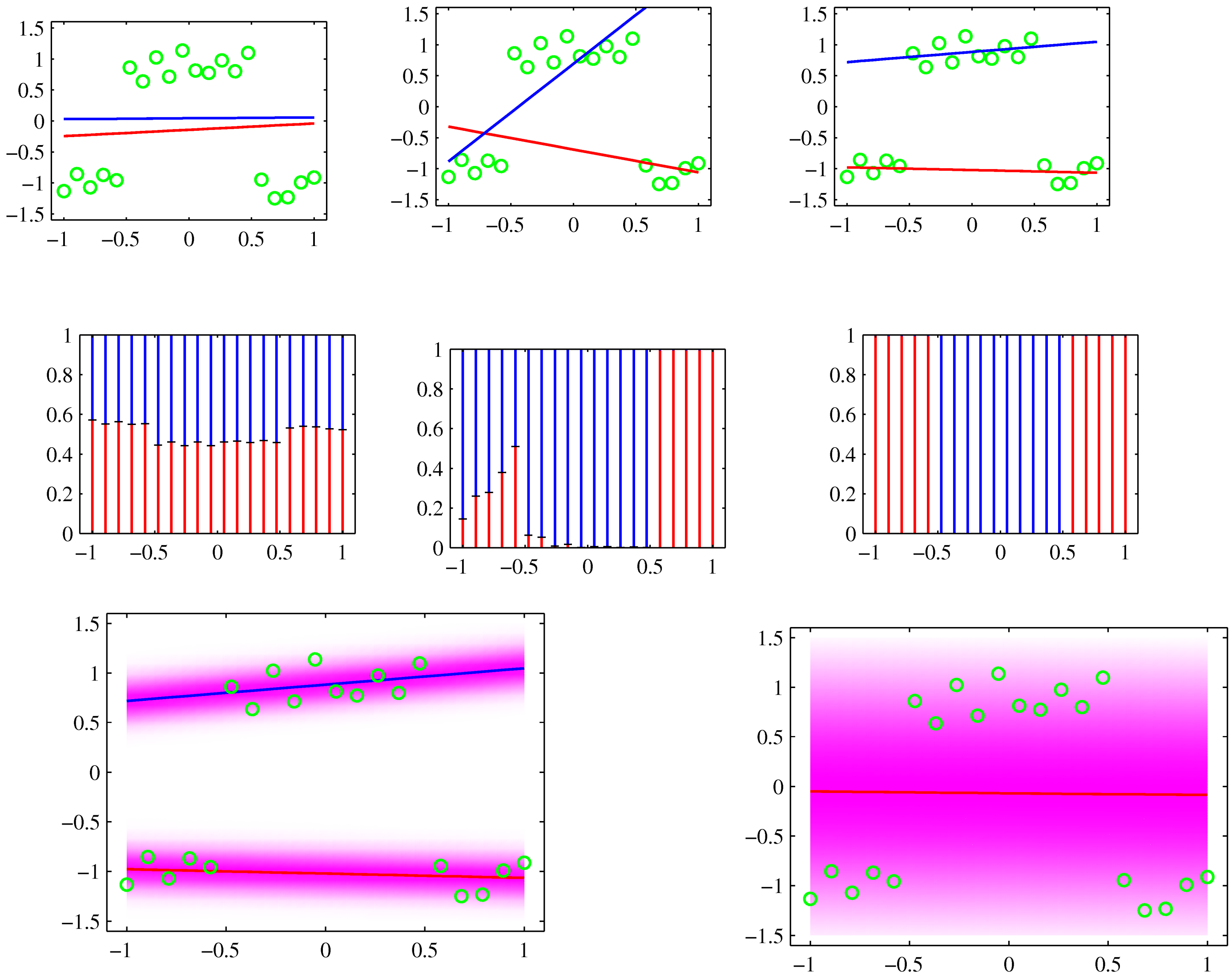
Conditional mixtures



$$p(y_i|\mathbf{x}_i, z_i = k, \mathbf{W}, \boldsymbol{\sigma}) = \mathcal{N}(y_i|\mathbf{x}_i^T \mathbf{w}_k, \sigma_k^2)$$

$$p(z_i = k|\mathbf{x}_i, \mathbf{B}) = \mathcal{S}(\mathbf{x}_i \mathbf{B})_k$$

Mixtures of linear regression

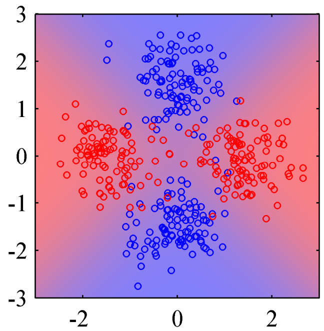


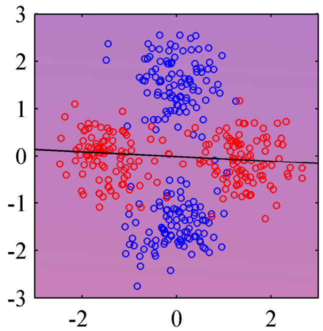
LL=-3

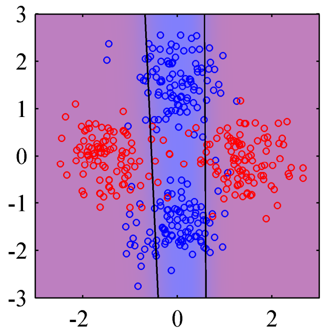
LL=-27.6

Bishop

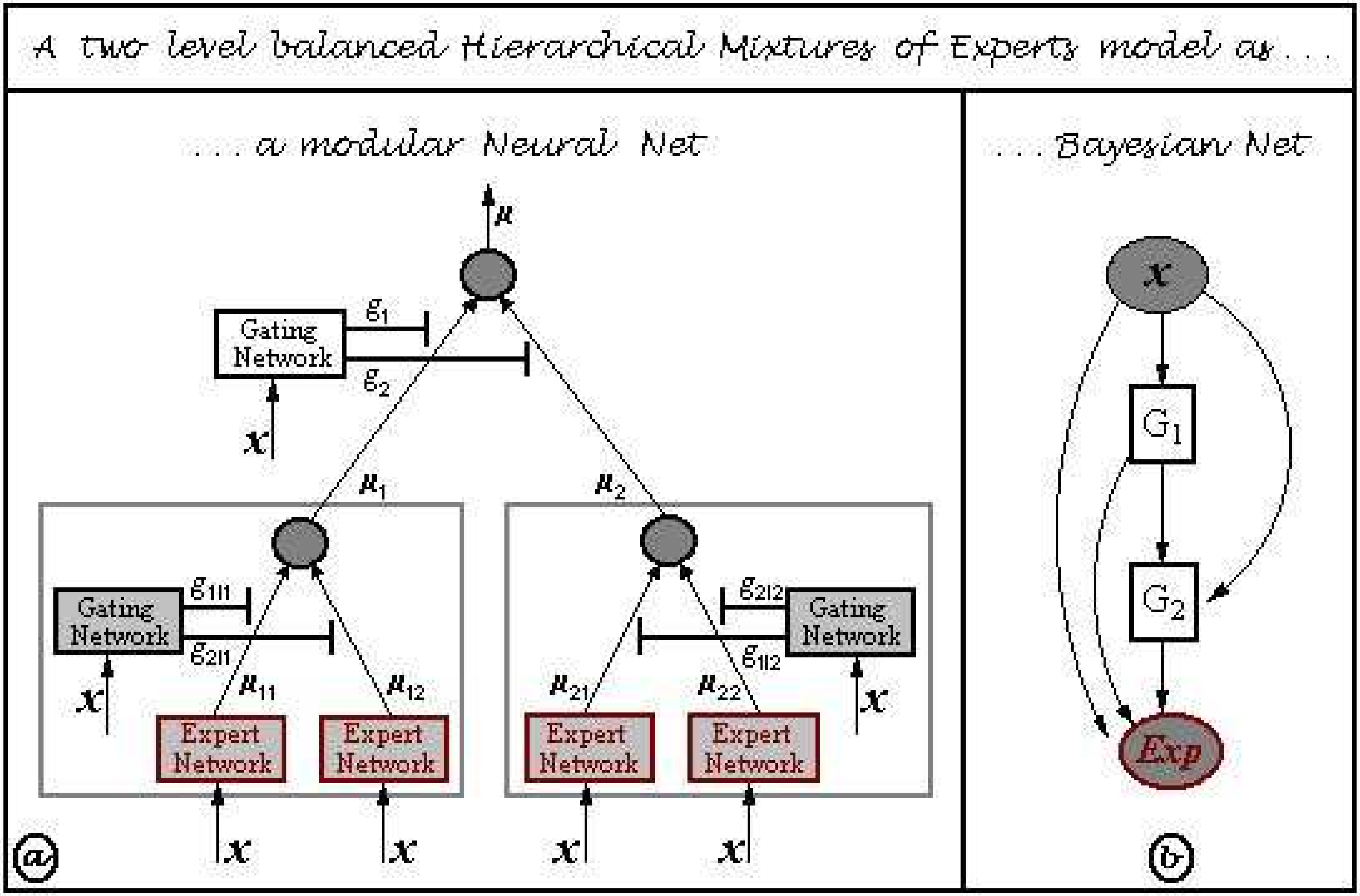
Mixtures of logistic regression





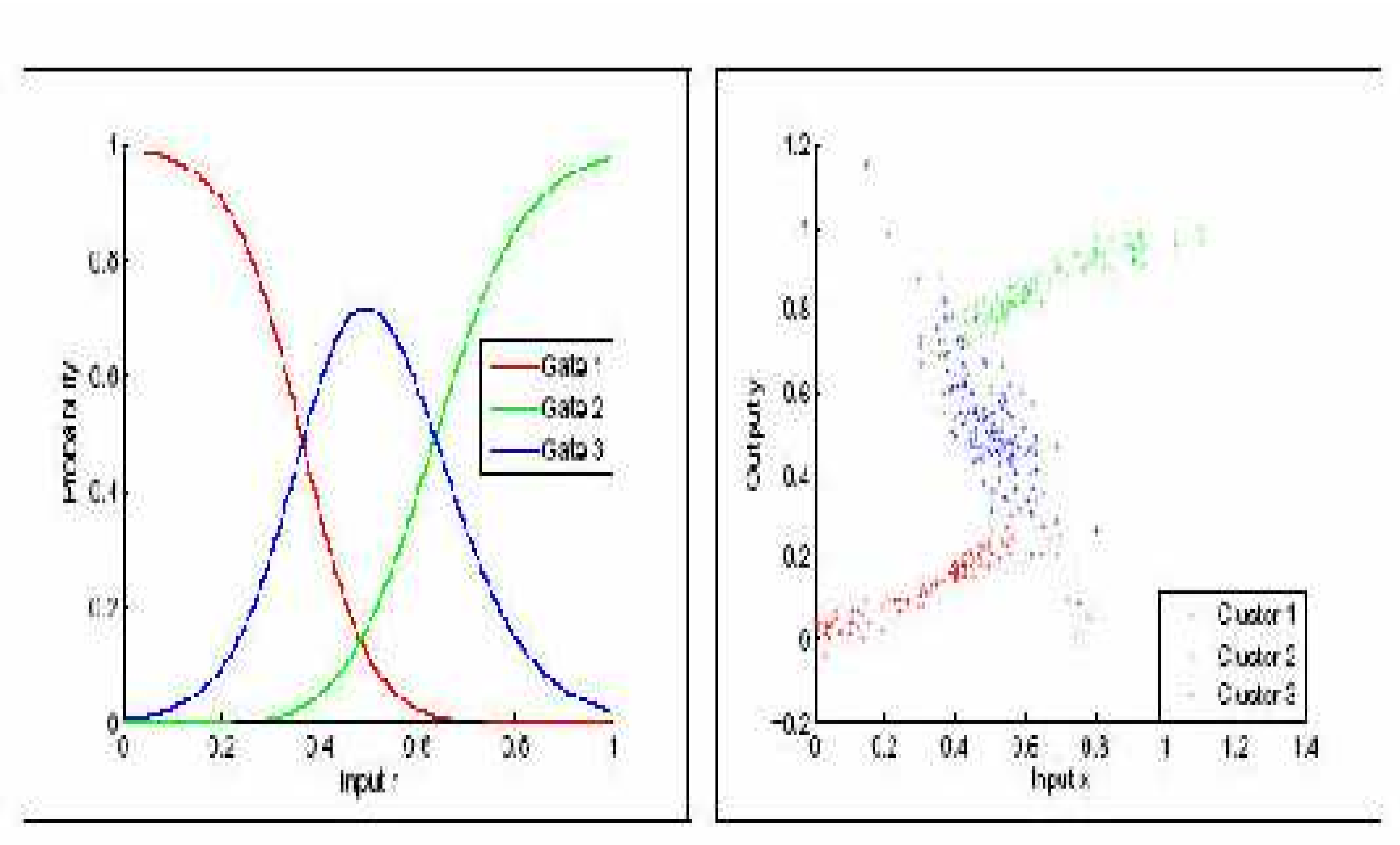


Hierarchical mixtures of experts



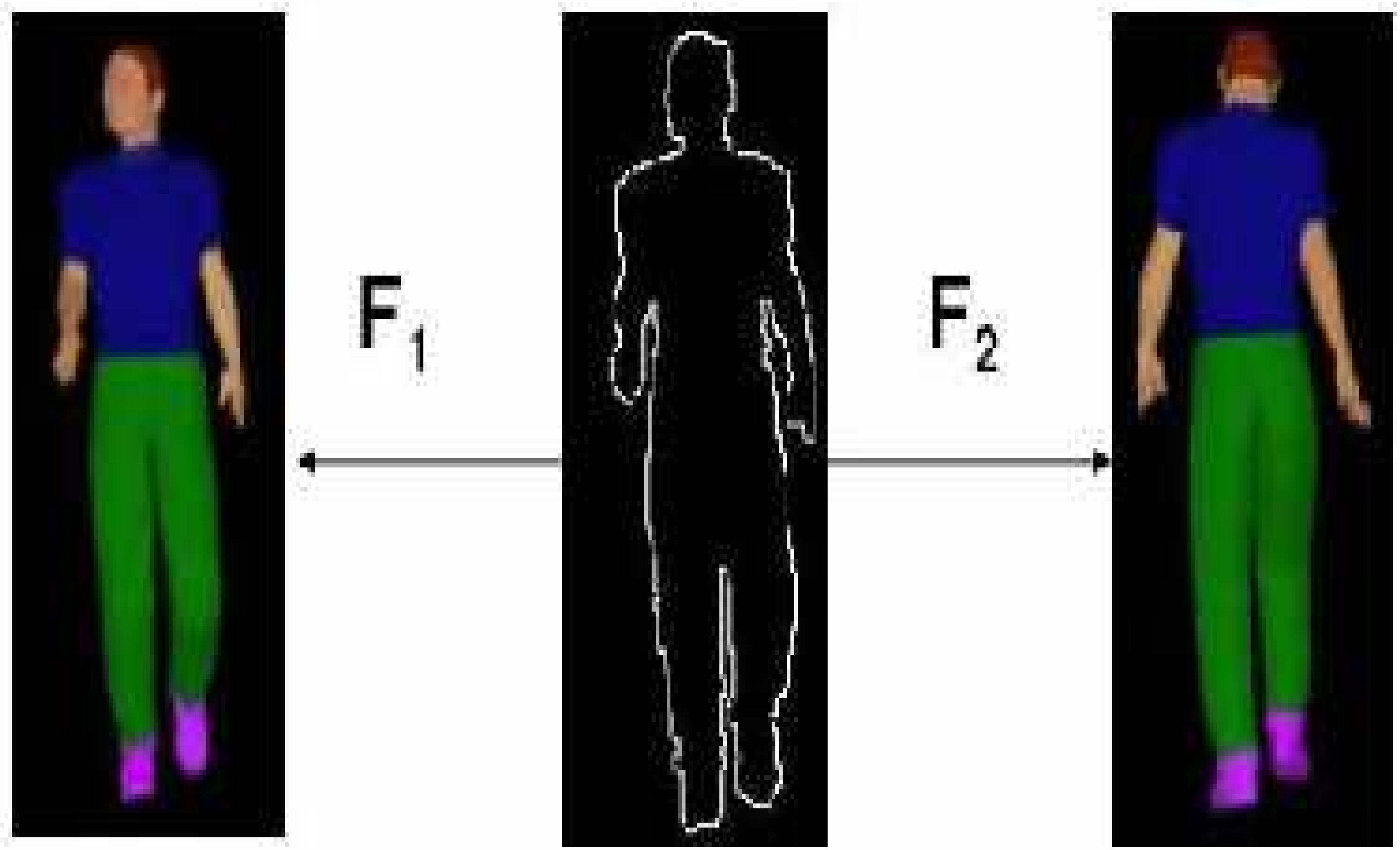
Brutti

One to many functions



Sminchisescu

Ambiguity in inferring 3d from 2d



Sminchisescu

Outline

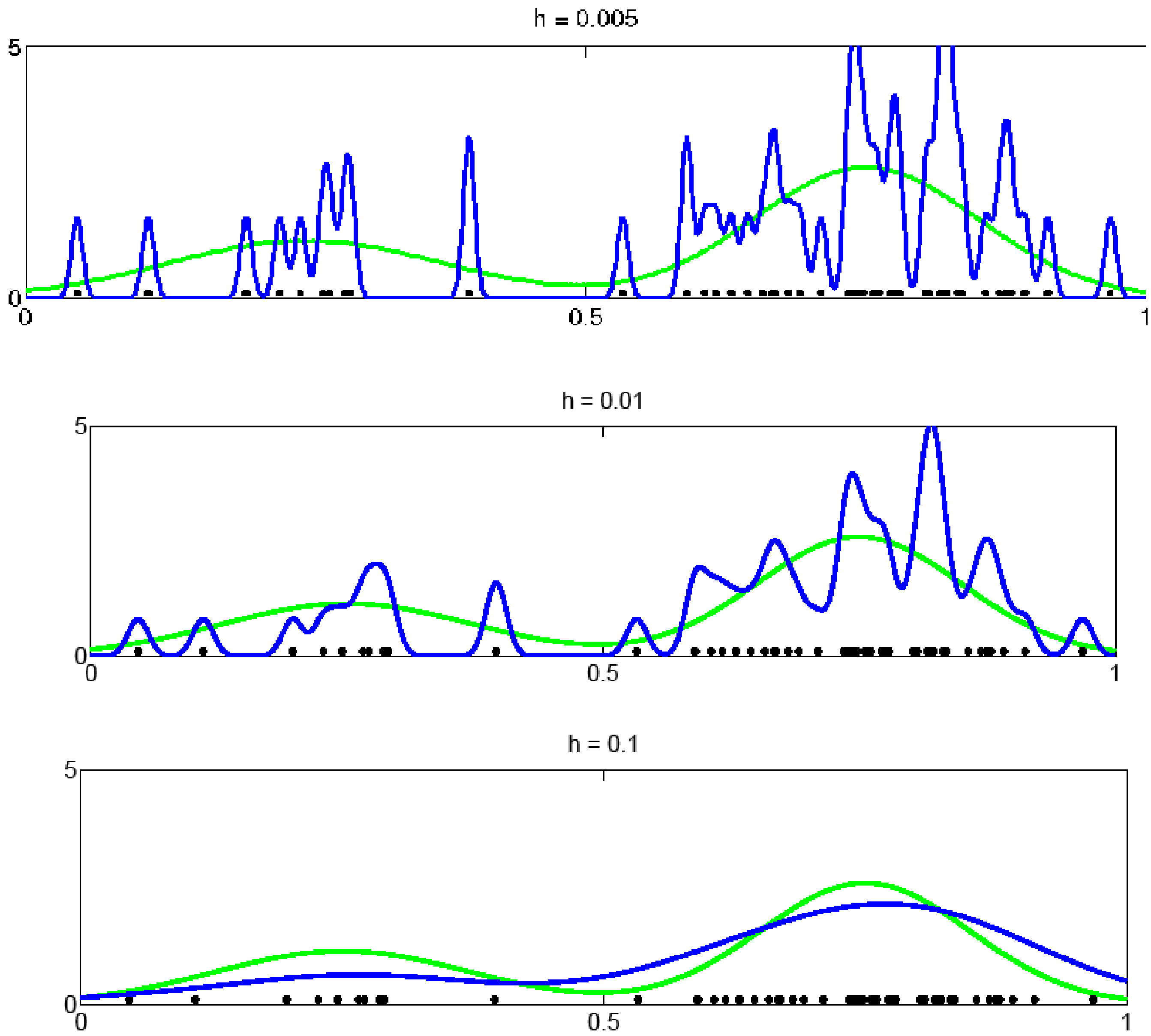
- Mixture models
- EM for mixture models
- K means clustering
- Conditional mixtures
- Kernel density estimation
- Kernel regression

Kernel density estimation

- Parzen window density estimator
- Put one centroid on each data point, $\mu_i = x_i$, and set $\pi_i = 1/n$, $\Sigma_i = h^2 I_d$

$$
\begin{aligned}
p(\mathbf{x}) &= \frac{1}{n}\sum_{i=1}^{n} k_h(\mathbf{x}-\mathbf{x}_i) \\
k_h(\mathbf{u}) &= \frac{2}{(2\pi h^2)^{d/2}} \exp[-\frac{1}{2h}||\mathbf{u}||^2]
\end{aligned}
$$

Bandwidth h



Outline

- Mixture models
- EM for mixture models
- K means clustering
- Conditional mixtures
- Kernel density estimation
- Kernel regression

Kernel regression

- Nadaraya-Watson model
- KDE on (y_i,x_i)

$$p(\mathbf{x}, y|\mathcal{D}) = \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}-\mathbf{x}_i, y-y_i)$$

$$p(y|\mathbf{x}, \mathcal{D}) = \frac{p(y, \mathbf{x}|\mathcal{D})}{\int p(y, \mathbf{x}|\mathcal{D})dy}$$

- Eg f is Gaussian

$$f(\mathbf{z}_i) = \mathcal{N}(\mathbf{z}_i|\mathbf{0}, \sigma^2\mathbf{I}_{d+1})$$

Gaussian kernel regression

$$p(y|\mathbf{x}, \mathcal{D}) = \frac{\sum_{i=1}^{n} \mathcal{N}(\mathbf{z} - \mathbf{z}_i|\mathbf{0}, \sigma^2\mathbf{I}_{d+1})}{\int \sum_{i=1}^{n} \mathcal{N}(\mathbf{z} - \mathbf{z}_i|\mathbf{0}, \sigma^2\mathbf{I}_{d+1})dy}$$

Numerator

$$\sum_{i=1}^{n} \mathcal{N}(\mathbf{x}|\mathbf{x}_i, \sigma^2\mathbf{I}_d)\mathcal{N}(y|y_i, \sigma^2)$$

Denominator

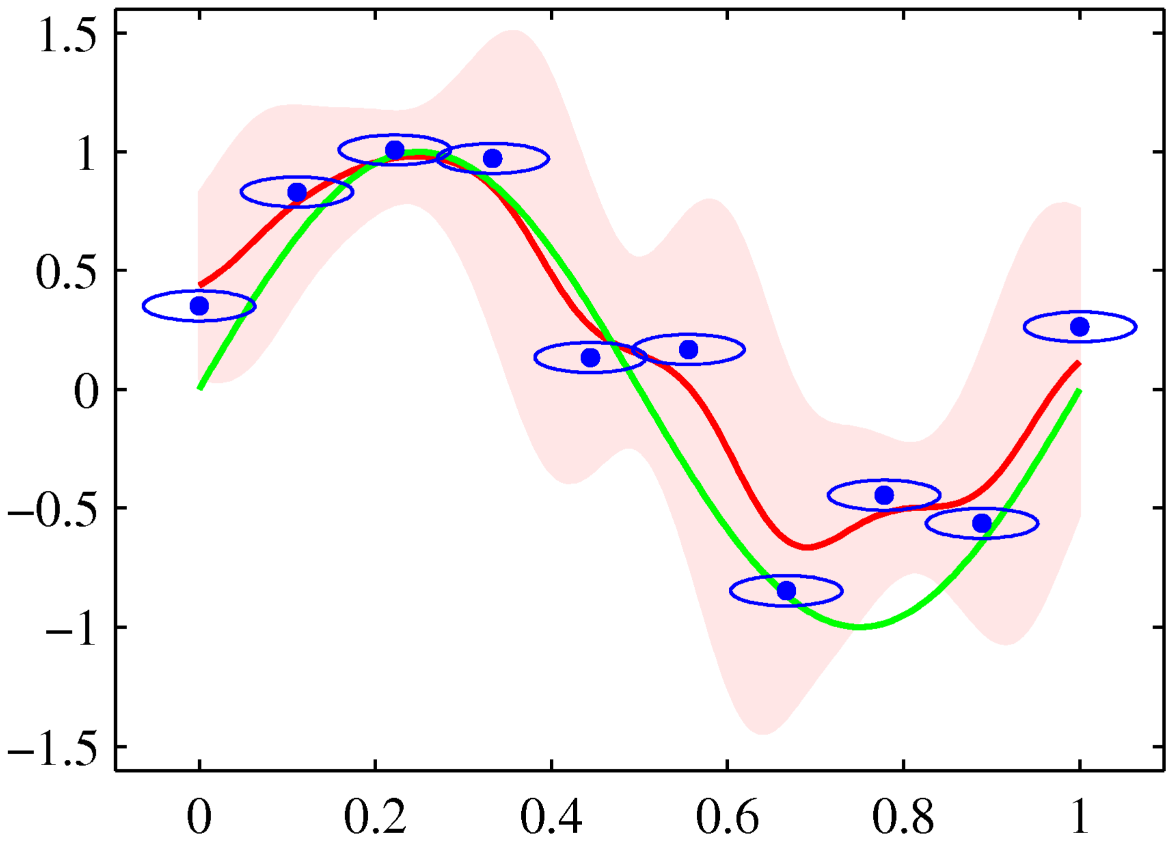
$$\int \sum_{i=1}^{n} \mathcal{N}(\begin{pmatrix} \mathbf{x} \\ y \end{pmatrix} | \begin{pmatrix} \mathbf{x}_i \\ y_i \end{pmatrix}, \sigma^2\mathbf{I}_{d+1})dy = \sum_{i=1}^{n} \mathcal{N}(\mathbf{x}|\mathbf{x}_i, \sigma^2\mathbf{I}_d)$$

Hence

$$p(y|\mathbf{x}, \mathcal{D}) = \sum_{i=1}^{n} k(\mathbf{x}, \mathbf{x}_i)\mathcal{N}(y|y_i, \sigma^2)$$

$$k(\mathbf{x}, \mathbf{x}_i) \stackrel{\text{def}}{=} \frac{\mathcal{N}(\mathbf{x}|\mathbf{x}_i, \sigma^2\mathbf{I}_d)}{\sum_{j=1}^{n} \mathcal{N}(\mathbf{x}|\mathbf{x}_j, \sigma^2\mathbf{I}_d)}$$

Gaussian kernel regression



Generative models for regression

- We can other models for p(x,y), eg finite GMM
- Harder to fit; faster at test time
- Generative models can handle missing data