Machine Learning from Data

Lecture 19: Spring 2021

Today's Lecture

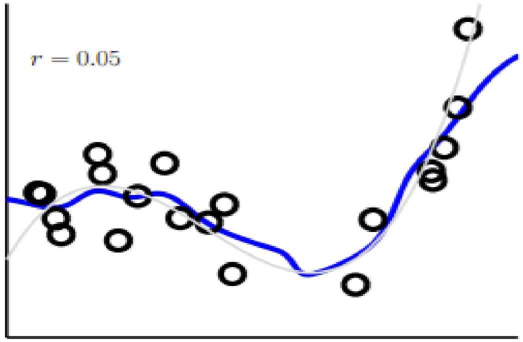
- Unsupervised Learning
- K-means clustering
- Probability Density Estimation
- Gaussian Mixture Models

RECAP: # Radial Basis Functions

Nonparametric RBF

$$g(\mathbf{x}) = \sum_{n=1}^{N} \left(\frac{\alpha_n(\mathbf{x})}{\sum_{m=1}^{N} \alpha_m(\mathbf{x})} \right) \cdot y_n$$

$$\alpha_n(\mathbf{x}) = \phi\left(\frac{\|\mathbf{x}-\mathbf{x}_n\|}{r}\right) \qquad \text{(bump on } \mathbf{x}\text{)}$$



No Training

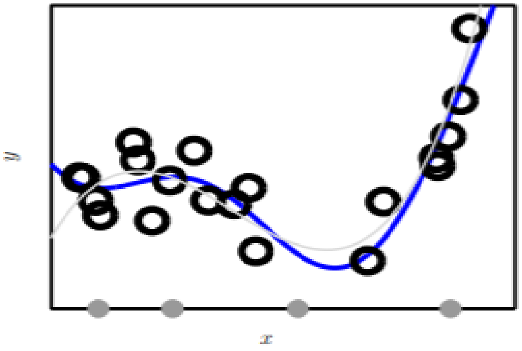
Parametric k-RBF-Network

$$h(\mathbf{x}) = w_0 + \sum_{j=1}^{k} w_j \cdot \phi\left(\frac{\|\mathbf{x}-\boldsymbol{\mu}_j\|}{r}\right) = \mathbf{w}^{\mathrm{T}}\boldsymbol{\Phi}(\mathbf{x})$$

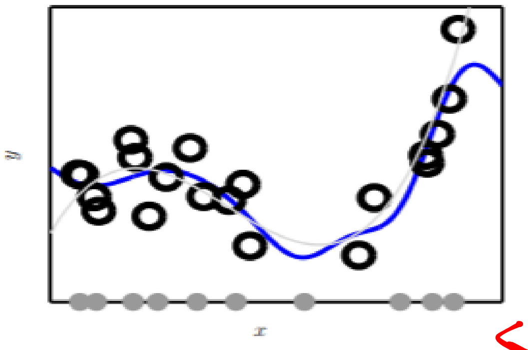
(bump on $\boldsymbol{\mu}_j$)

linear model given $\boldsymbol{\mu}_j$

choose $\boldsymbol{\mu}_j$ as centers of k-clusters of data



$k = 4, r = \frac{1}{k}$



$k = 10$, regularized

Unsupervised Learning

x y

x'

- Preprocessor to organize the data for supervised learning:

  Organize data for faster nearest neighbor search

  Determine centers for RBF bumps.

- Important to be able to organize the data to identify patterns.

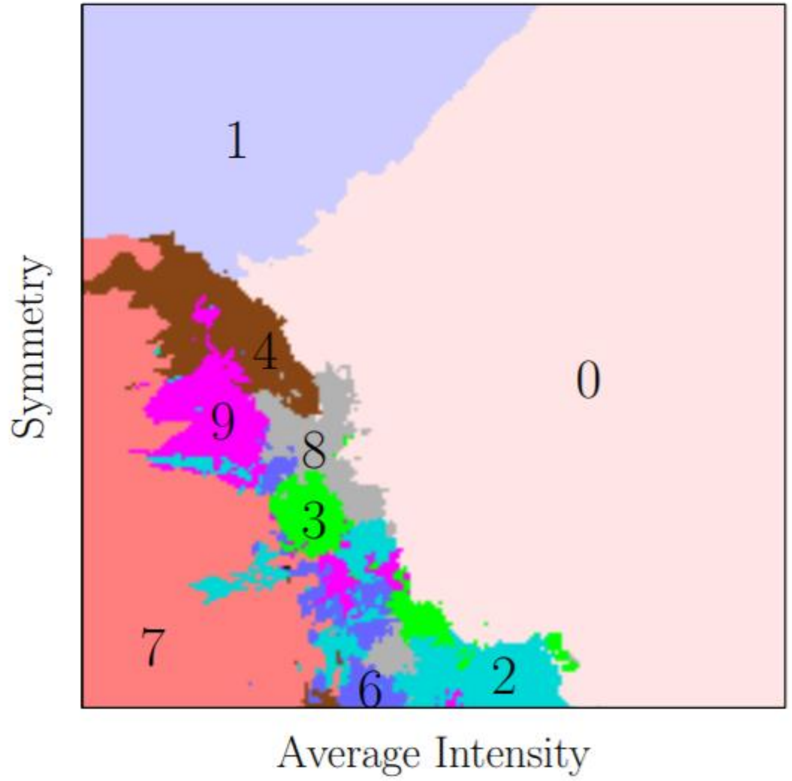
  Learn the patterns in data, e.g. the patterns in a language before getting into a supervised setting.

  amazon.com organizes books into categories

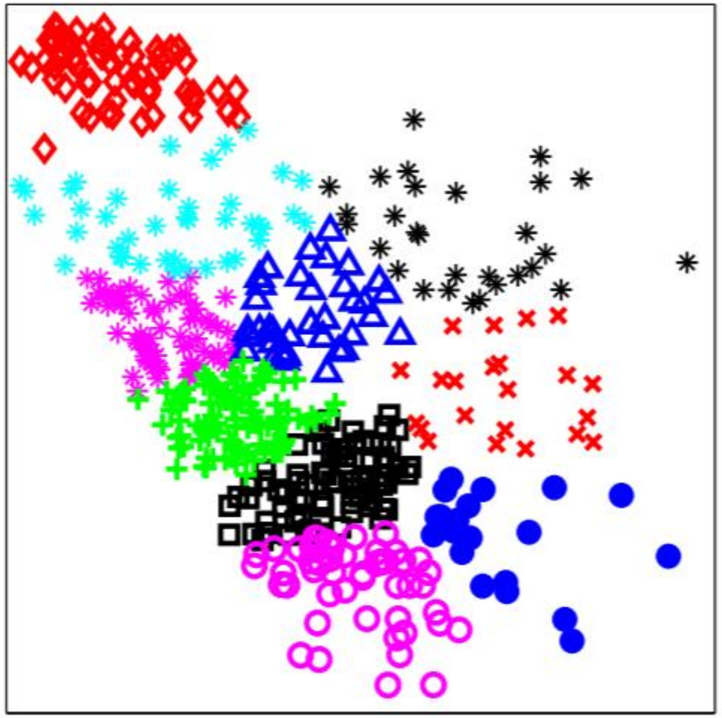
Clustering Digits

21-NN rule, 10 Classes

labels



10 Clustering of Data



Clustering [Unsupervised]

→ Assume we are given k-subsets.

→ $S_1 \; S_2 \cdots\cdots S_k$

i) Disjoint (Non overlapping)

ii) Complete (covers entire dataset)

→ $S_i \cap S_j = \emptyset \quad i \neq j$

→ $\bigcup_{i=1}^{k} S_i = D$

Trivial

$D = [x_1, x_2 \cdots x_N]$

$\downarrow$ OUTPUT

Cluster data into some subsets

$\downarrow$

We need to know how many?

What makes clustering good?

1) Tightly bound — within cluster. ✓

2) Well separated - between cluster.

$$S_1 \; S_2 \; \dots \; S_K \Longrightarrow \mu_j$$

$$\underline{E_j} = \sum_{x_i \in S_j} \|x_i - \mu_j\|^2 \longrightarrow \text{Spread of cluster.}$$

$$E = \sum_{j=1}^{K} E_j \longrightarrow E \simeq \text{small}$$

K-means clustering error.

$$\underline{\text{Minimize}} \longrightarrow \boxed{E = \sum_{n=1}^{N} \|\underline{x_n} - \mu(x_n)\|^2}$$

Output $\rightarrow$ $\left.\begin{matrix} S_1 & S_2 & \cdots & S_k \\ \mu_1 & \mu_2 & & \mu_k \end{matrix}\right\}$ NP-hard

Alternating Algorithm / Lloyd's

i) $\mu_1, \mu_2 \ldots \mu_k$ (given) $\rightarrow$ greedily

ii) $S_1, S_2 \ldots S_k$ (given)

$$E_j = \sum_{x \in S_j} \| x - \mu_j \|^2$$

μ_j^*

$$\mu_j^* = \frac{\sum_{x \in S_j} x}{|S_j|}$$

Lloyd's Algorithm

1) Find the centers $\mu_1, \mu_2 \dots \mu_k$. How?

→ Randomly selected. (first step)

OR → Mean vector in S_j

2) Assign data to centers greedily.

$\arg\min \|x_n - \mu_j\|^2$

Repeat Until convergence.

Convergence

Change $\rightarrow$ Error $\rightarrow$ Improving.

$\downarrow$

decreasing

$\rightarrow$ local minima.

Lloyd's Algorithm for k-Means Clustering

Digits

$$E_{\text{in}}(S_1,\ldots,S_k;\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_k)=\sum_{n=1}^{N}\|\mathbf{x}_n-\boldsymbol{\mu}(\mathbf{x}_n)\|^2$$

1: **Initialize** Pick well separated centers $\boldsymbol{\mu}_j$.

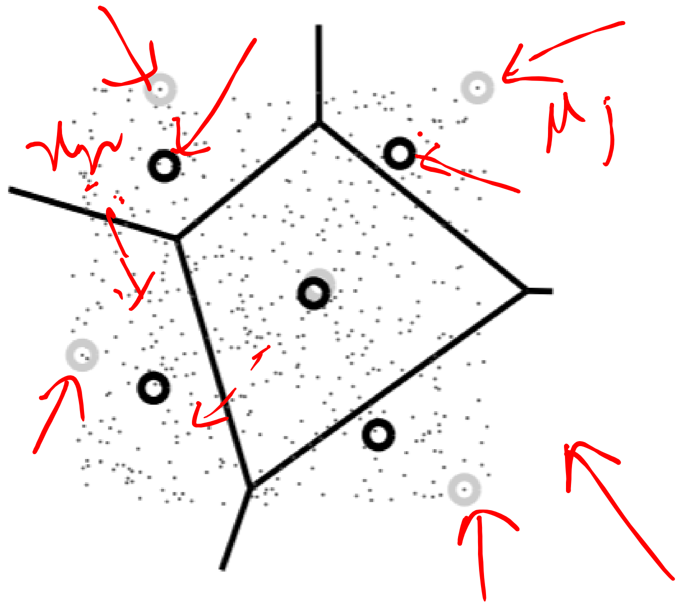
2: **Update** S_j to be all points closest $\boldsymbol{\mu}_j$.

$$S_j \leftarrow \{\mathbf{x}_n : \|\mathbf{x}_n-\boldsymbol{\mu}_j\| \le \|\mathbf{x}_n-\boldsymbol{\mu}_\ell\| \text{ for } \ell=1,\ldots,k\}.$$

3: **Update** $\boldsymbol{\mu}_j$ to the centroid of S_j.

$$\boldsymbol{\mu}_j \leftarrow \frac{1}{|S_j|}\sum_{\mathbf{x}_n\in S_j}\mathbf{x}_n$$

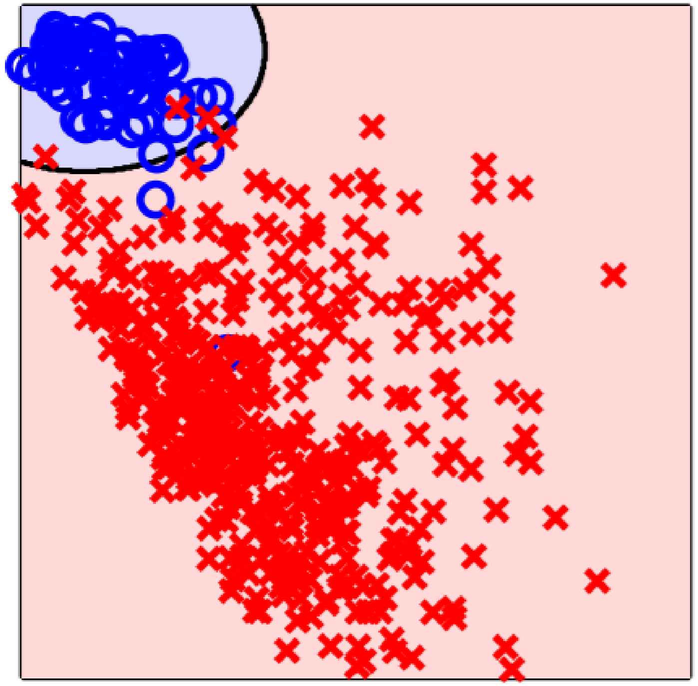
4: Repeat steps 2 and 3 until E_{in} stops decreasing.



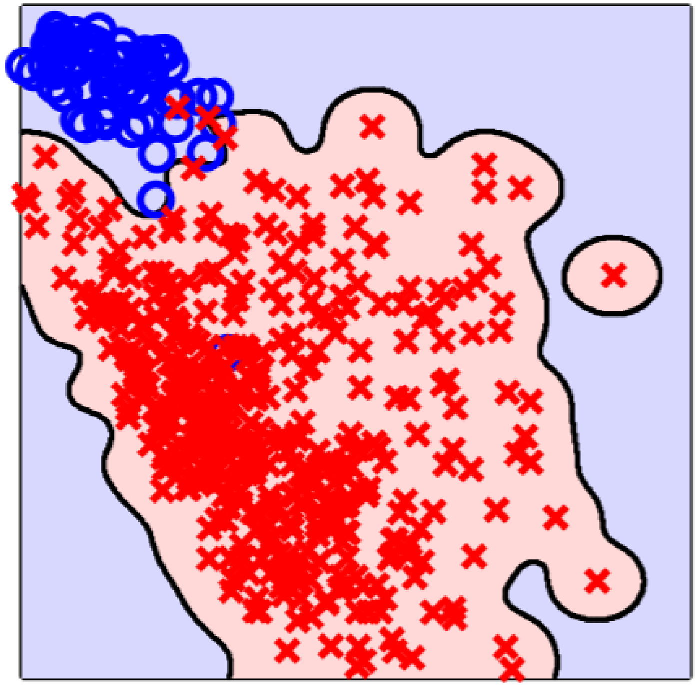
Application to k-RBF-Network

1 vs Not 1

10-center RBF-network



300-center RBF-network



Overfitting

Regularization

Choosing k - knowledge of problem (10 digits) or CV.

Probability Density Estimation

Data $\rightarrow P(x)$

Text

$$P(\mathbf{x})$$

$P(\mathbf{x})$ measures how likely it is to generate inputs *similar* to $\mathbf{x}$.

Similarity

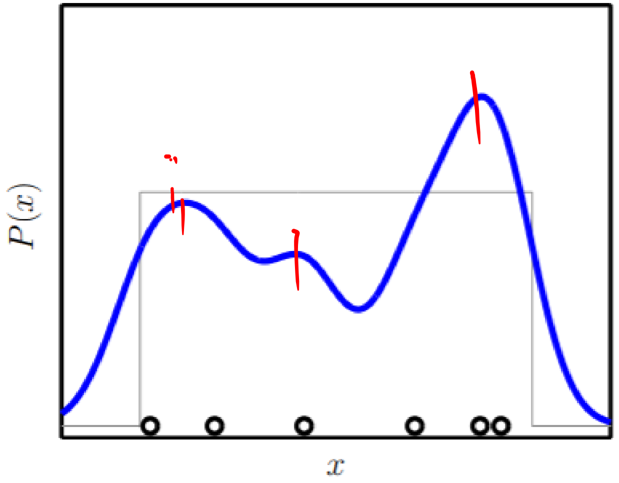
Estimating $P(\mathbf{x})$ results in a 'softer/finer' representation than clustering

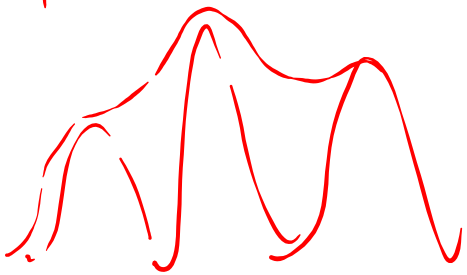
Clusters are regions of high probability.

Parzen Windows – RBF density estimation

Normalized prob. density

Basic idea: put a bump of 'size' (volume) $\frac{1}{N}$ on each data point.





x_i

N

1

$\frac{1}{N}$

$$\hat{P}(\mathbf{x}) = \frac{1}{Nr^d}\sum_{i=1}^{N} \phi\left(\frac{\|\mathbf{x}-\mathbf{x}_i\|}{r}\right)$$

scale

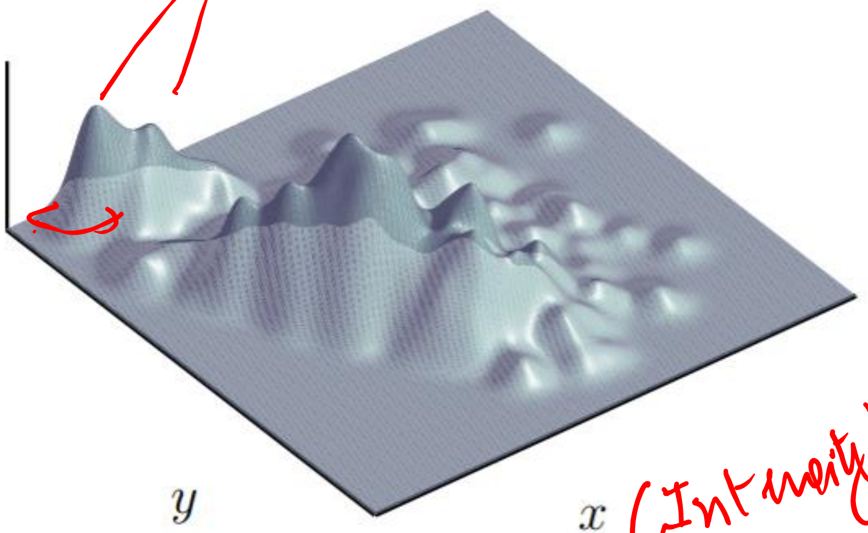
$\phi(z) = \frac{1}{(2\pi)^{d/2}} e^{-\frac{1}{2}z^2}$

Digits Data

Non-parametric RBF

RBF Density Estimate

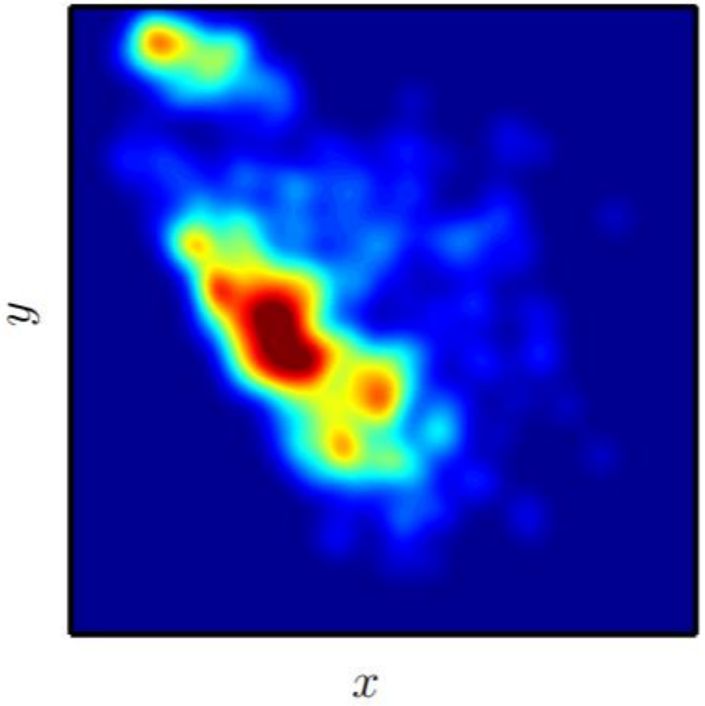
high prob.



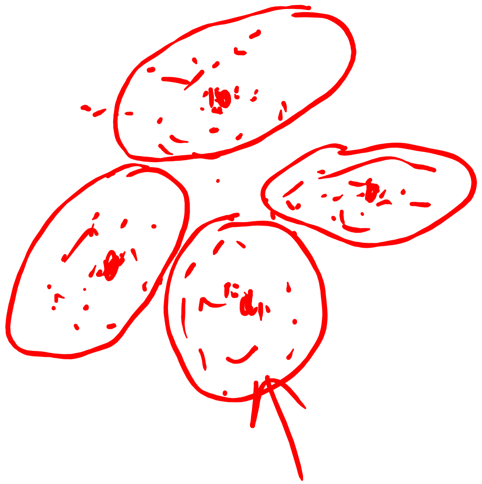
symmetry

(Intensity)

Density Contours



Gaussian Mixture Model (GMM)



$k = 4$

$\mu_1 \quad \mu_2 \quad \mu_3 \quad \mu_4$

Shape : Covariance Matrix

$\Sigma_1 \quad \Sigma_2 \quad \Sigma_3 \quad \Sigma_4$

Bump weights

$\omega_1 \quad \omega_2 \quad \omega_3 \quad \omega_4$

gaussian density

GMM

$$\underbrace{\hat{P}(x) = \sum_{k=1}^{K} \omega_k \, N(x, \mu_k, \Sigma_k)}$$

$$N(x, \mu, \Sigma)$$

mean

covariance matrix

$x_1 \quad x_2 \cdots x_M$

$$N(x, \mu, \Sigma) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} e^{-1/2 (x-\mu)^T \Sigma^{-1} (x-\mu)}$$

$$\frac{1}{M} \sum x_i = E[x] = \mu$$

$$\frac{1}{M} \sum x_i x_i^T \Rightarrow E[x x^T] = \mu \mu^T + \Sigma$$

$$\hat{P}(x) = \sum_{k=1}^{K} \omega_k \, N(x, \mu_k, \Sigma_k), \quad \omega_k \geq 0, \quad \sum_{k=1}^{K} \omega_k = 1$$

$$(\omega_j, \mu_k, \Sigma_j)$$

weights, centers, shapes.

Pick in such a way → max. the prob. to generate the data $x_1, x_2, \ldots x_L$

Maximum Likelihood.

Expectation Maximization

1) We knew which x_n belongs to which bump

$\longrightarrow \mu_j$ & Σ_j & w_j

2) If we knew $\{w_j\ \mu_j\ \Sigma_j\} \longrightarrow$ which x_n belongs to j

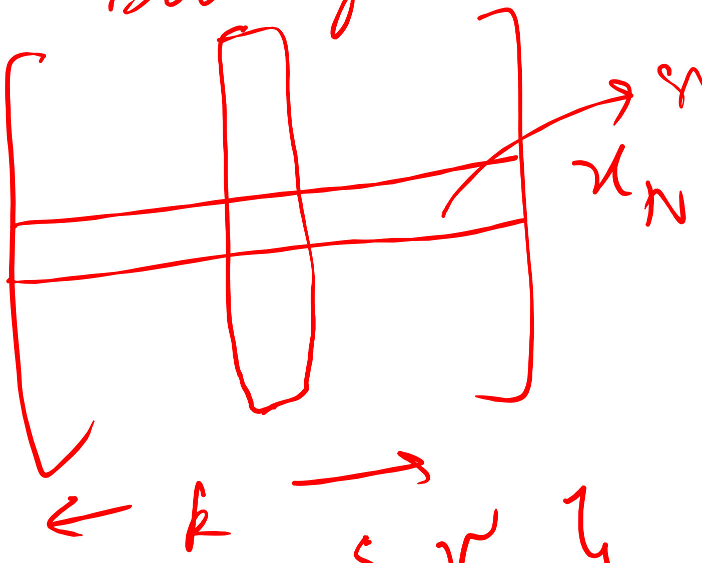
Algo

1) Initialization:

membership variable r_{nj} = The fraction of data point x_n that belongs to bump j

$r_{nj} \geq 0$

$\sum_j r_{nj} = 1$

$r = N$ 

Assume we know $\{r_{nj}\}$

$$N_j = \sum_{n=1}^{N} r_{nj}$$

$$N = \sum_j N_j$$

Volume $\omega_j = \frac{N_j}{N}$

$$\mu_j = \frac{\sum_{n=1}^{N} r_{nj} x_n}{N_j}$$

$$\Sigma_j = \frac{1}{N_j} \sum_{n=1}^{N} r_{nj} x_n x_n^T - \mu_j \mu_j^T$$

(2) Update $r_{nj} = \frac{P[x_n \text{ came from bump } j]}{\sum_j P[x_n \text{ came from bump } j]}$

$$= \frac{\omega_j N(x_n \mu_j \Sigma_j)}{\sum_j \omega_j N(x_n \mu_j \Sigma_j)}$$

(3) Repeat Until convergence

E-M Algorithm

E-M Algorithm for GMMs:

1: Start with estimates for the bump membership γ_{nj}.

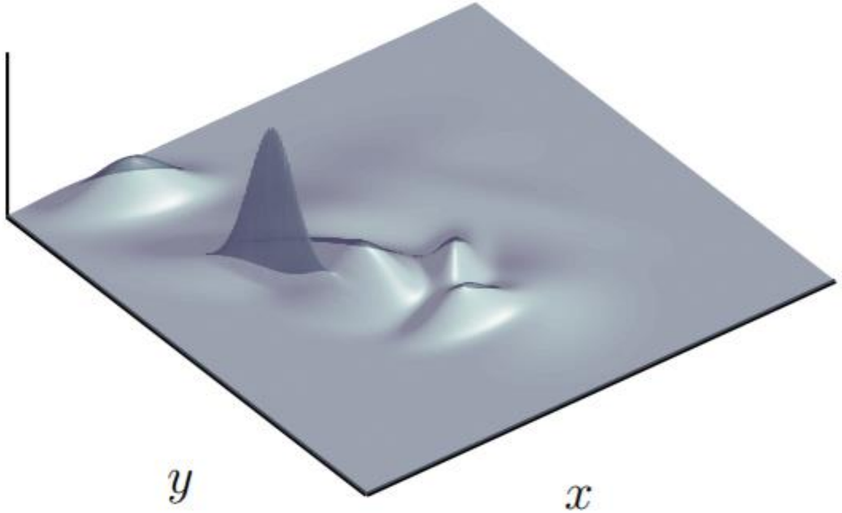
2: Estimate $w_j, \boldsymbol{\mu}_j, \Sigma_j$ given the bump memberships.

3: Update the bump memberships given $w_j, \boldsymbol{\mu}_j, \Sigma_j$;

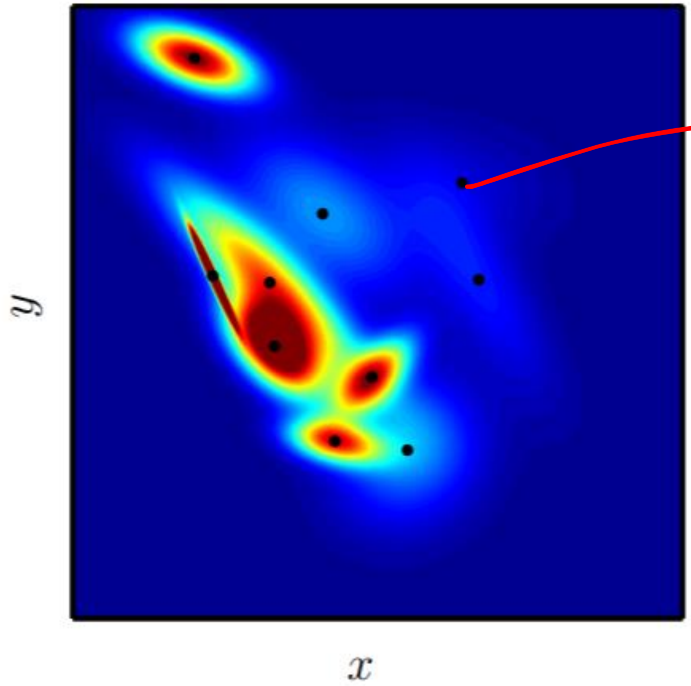
4: Iterate to step 2 until convergence.

GMM on Digits Data

10-center GMM



Density Contours



centers

Thanks!