Probability Density Estimation

Traditional approach to pattern recognition:

1) estimate (from data) $P(C_i|x)$ (inference)

2) make optimal decisions

In practice we estimate $p(x|C_i)$ rather than $P(C_i|x)$!

- Parametric methods and ML estimates
- Semi-parametric methods: Mixture Models and EM
- Non-parametric methods:
  - histograms
  - kernel methods
  - k-nearest neighbors

Parametric density estimation

Key idea:

- Assume that p(x) can be expressed by a formula that involves some parameters Θ, p(x, Θ) (e.g., Normal(μ, σ), Poisson(λ), etc.)

- Find (using your data) "optimal" values of these parameters

- “OPTIMAL” = Maximum Likelihood Principle:
  "optimal values are those that maximize
  the “likelihood” of observing data":

$$L(\Theta) = \prod_{x \in X} p(x \mid \Theta) \qquad \text{(Likelihood (maximize!))}$$

$$-\ln L(\Theta) = -\sum_{x \in X} \ln p(x \mid \Theta) \quad \text{(Negative Log Likelihood (minimize!))}$$

- For most known distributions formulas for optimal values are known

Example: Normal Distribution

Consider a set of 10 numbers:

0.8165 0.7627 0.7075 0.7352 0.6303 0.8696 0.7059 0.8797 0.7264 0.7872

Assuming that they come from a normal distribution with unknown parameters mu and sigma, how can we find "most likely" values of these parameters?

For mu=0.5 and sigma=0.1 we get [>> pdfs=normpdf(x,0.5,0.1)]:

0.0267 0.1266 0.4633 0.2512 1.7059 0.0043 0.4789 0.0030 0.3075 0.0646

Their product is **8.1093e-011** (very unlikely!)

For mu=0.6 and sigma=0.2 we get [>> pdfs=normpdf(x,0.6,0.2)]:

1.1103 1.4329 1.7264 1.5875 1.9719 0.8040 1.7338 0.7502 1.6336 1.2874

Their product is **18.9067** (more likely!)

Example: Normal Distribution

Trying mu=0.1:0.1:1 and sigma=0.1:0.1:1 we get the matrix log(prod(normpdf(x,mu,sigma))):

-208.0754	-48.5730	-21.8065	-13.8960	-11.1344	-10.2453	-10.1514	-10.4253	-10.8754	-11.4085
-146.8654	-33.2705	-15.0054	-10.0703	-8.6860	-8.5451	-8.9023	-9.4689	-10.1198	-10.7964
-95.6554	-20.4680	-9.3154	-6.8697	-6.6376	-7.1226	-7.8572	-8.6688	-9.4875	-10.2843
-54.4454	-10.1655	-4.7365	-4.2941	-4.9892	-5.9778	-7.0161	-8.0249	-8.9788	-9.8722
-23.2354	-2.3630	-1.2688	-2.3435	-3.7408	-5.1109	-6.3792	-7.5372	-8.5935	-9.5601
-2.0254	2.9395	1.0879	-1.0178	-2.8924	-4.5217	-5.9463	-7.2058	-8.3316	-9.3480
9.1845	5.7420	2.3335	-0.3172	-2.4440	-4.2103	-5.7176	-7.0306	-8.1932	-9.2359
10.3945	6.0445	2.4679	-0.2416	-2.3956	-4.1767	-5.6929	-7.0117	-8.1783	-9.2238
1.6045	3.8470	1.4912	-0.7910	-2.7472	-4.4209	-5.8723	-7.1491	-8.2868	-9.3117
-17.1855	-0.8505	-0.5965	-1.9654	-3.4988	-4.9429	-6.2557	-7.4427	-8.5188	-9.4996

Thus mu around 0.7-0.8 and sigma around 0.1 are good guesses…

In general we have to find an optimum of a function of two variables: mu, sigma.

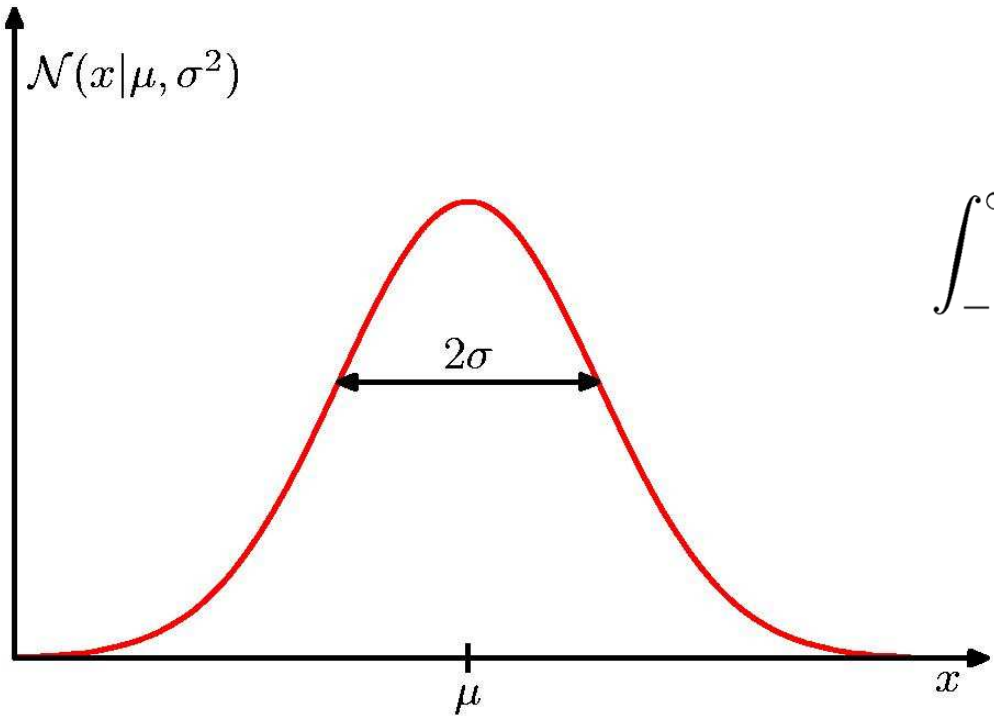
Accidentally (!?), it happens that this optimum is reached at:

mu = mean(x); sigma=std(x)

[In our case: mu= 0.7621; sigma =0.0778]

The Gaussian Distribution

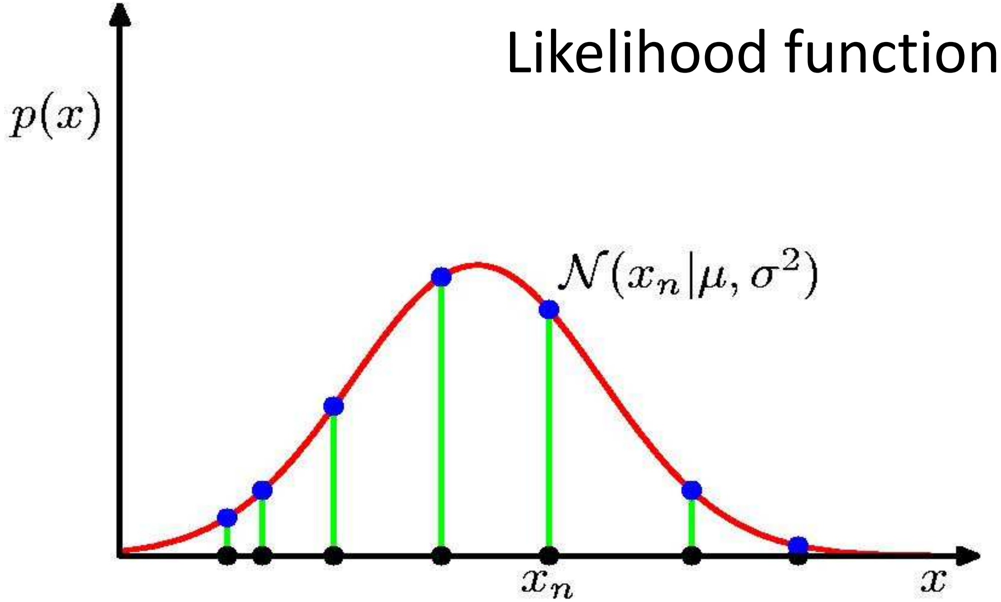
$$\mathcal{N}\left(x|\mu,\sigma^2\right) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$



$$\mathcal{N}(x|\mu,\sigma^2) > 0$$

$$\int_{-\infty}^{\infty} \mathcal{N}\left(x|\mu,\sigma^2\right) \mathrm{d}x = 1$$

Gaussian Parameter Estimation



$$p(\mathbf{x}|\mu, \sigma^2) = \prod_{n=1}^{N} \mathcal{N}\left(x_n|\mu, \sigma^2\right)$$

Maximum (Log) Likelihood

$$\mathcal{N}\left(x|\mu,\sigma^2\right) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$

$$\ln p\left(\mathbf{x}|\mu,\sigma^2\right) = -\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-\mu)^2 - \frac{N}{2}\ln\sigma^2 - \frac{N}{2}\ln(2\pi)$$

$$\mu_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N} x_n \qquad \sigma^2_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N}(x_n-\mu_{\mathrm{ML}})^2$$

The likelihood function is a function of mu and sigma, so its minimum can be found analytically ***(try it!!!)***

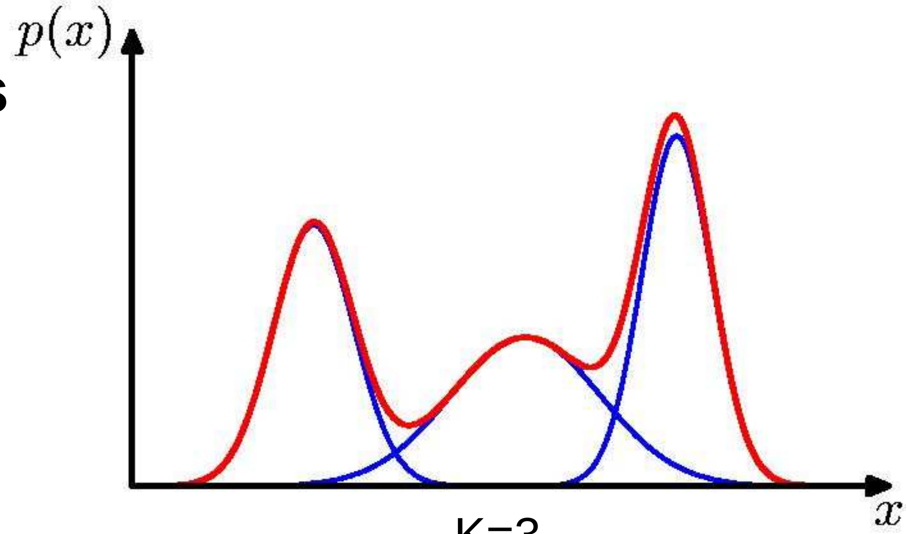
Mixtures of Gaussians

Combine simple models into a complex model:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \underbrace{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}_{\text{Component}}$$

Mixing coefficient

$$\forall k : \pi_k \geqslant 0 \qquad \sum_{k=1}^{K} \pi_k = 1$$



K=3

Expectation Maximization (EM): the main algorithm for estimating parameters of mixtures

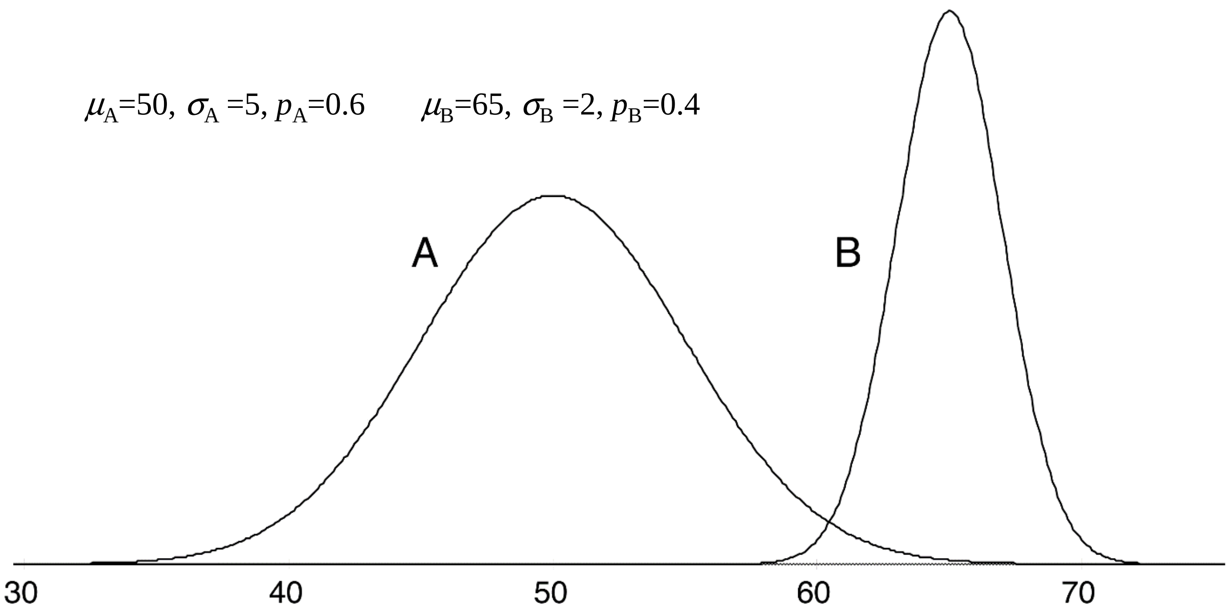
An example of a mixture models

data

A	51	B	62	B	64	A	48	A	39	A	51
A	43	A	47	A	51	B	64	B	62	A	48
B	62	A	52	A	52	A	51	B	64	B	64
B	64	B	64	B	62	B	63	A	52	A	42
A	45	A	51	A	49	A	43	B	63	A	48
A	42	B	65	A	48	B	65	B	64	A	41
A	46	A	48	B	62	B	66	A	48		
A	45	A	49	A	43	B	65	B	64		
A	45	A	46	A	40	A	46	A	48		

model

μ_A=50, σ_A =5, p_A=0.6 μ_B=65, σ_B =2, p_B=0.4



Problem formulation

given:

- data: x_1, x_2, …. (measurements)

- “meta-knowledge”: the data is a mixture of 2 normal distributions $N(m_A, s_A)$, $N(m_B, s_B)$

problem:

find values of p_A, m_A, s_A and p_B, m_B, s_B
(without knowing the labels!!!)

How???

=> Expectation Maximization Algorithm

Expectation Maximization Algorithm (EM)

Initialization:

start with *some* values of all parameters

Iteration:

- E-step:

  given parameter values calculate “probabilistic labels”:

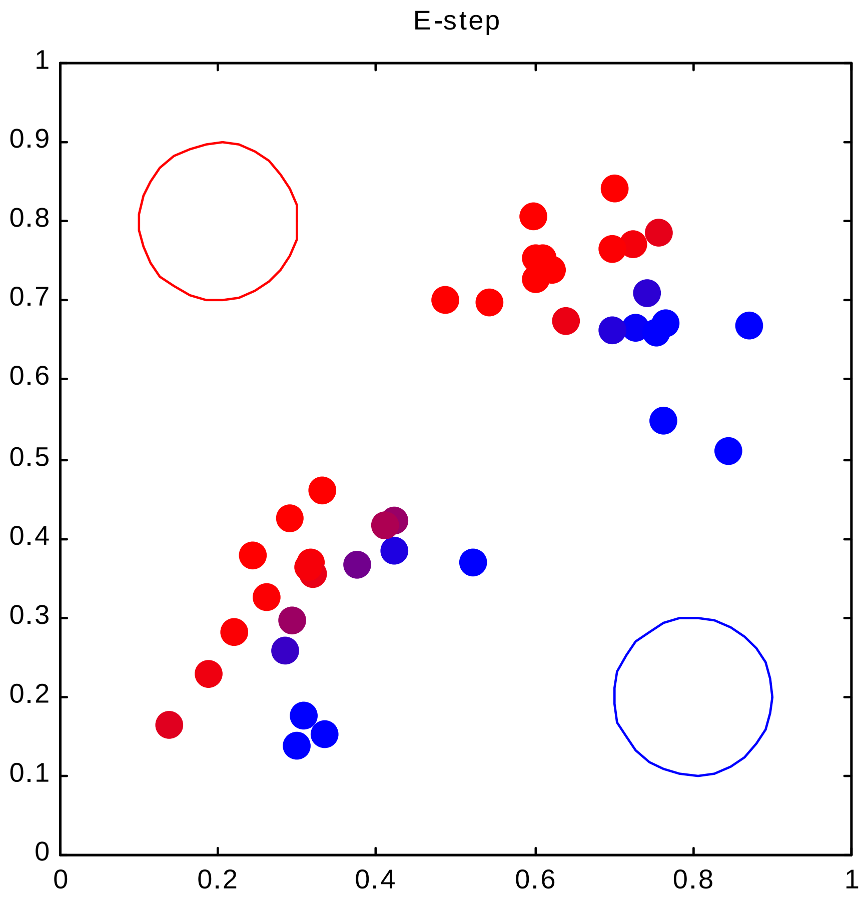
  for each observation x find P(A|x) and P(B|x)

- M-step:

  given “probabilistic labels” re-compute values of all

  parameters (mixing coefficients, means, std’s)

Demo EM in Matlab (demgmm1.m)



Expectation Maximization Algorithm (EM)

Initialization:

start with *some* values of p_A, m_A, s_A and p_B, m_B, s_B
(just guess some values!)

Iteration:

- E-step:

calculate P(class|x), where class is A or B

P(class|x)=p(x|class)*P(class)/p(x); i.e.:

$P(A|x)=p_A * p(x|A)= p_A * N(x; m_A, s_A)$

$P(B|x)=p_B * p(x|B)= p_B * N(x; m_B, s_B)$

Normalize both terms to make sure that they sum up to 1!

Expectation Maximization Algorithm (EM)

M-step:

given "probabilistic labels" find new values of

parameters: p_A, m_A, s_A and p_B, m_B, s_B

p_A =sum(P(A|x))/N; p_B =sum(P(B|x))/N;

m_A =sum(P(A|x)*x)/sum(P(A|x));
m_B =sum(P(B|x)*x)/sum(P(B|x));

s_A =sum(P(A|x)*(x- m_A)^2)/sum(P(A|x)); s_A = sqrt(s_A);
s_B =sum(P(B|x)*(x- m_B)^2)/sum(P(B|x)); s_B = sqrt(s_B);

EM Algorithm: summary

Initialization: set parameters at random

Repeat:

E-step:

calculate P(class|x), where class is A or B

(probabilities of being in A or B)

M-step: re-estimate model parameters

given “probabilistic labels” find new values of

parameters: p_A, m_A, s_A and p_B, m_B, s_B

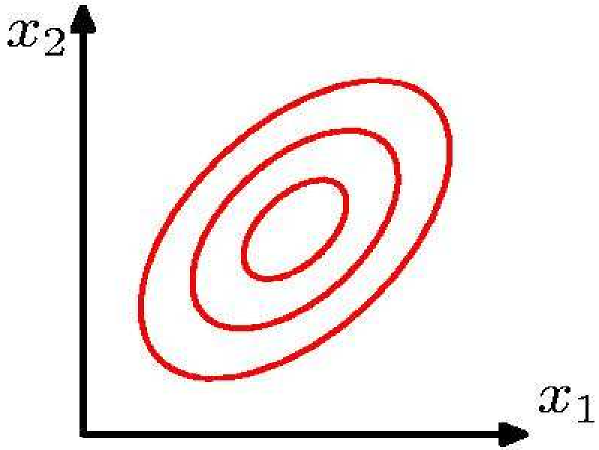
till convergence

The Multivariate Gaussian

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{ -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu}) \right\}$$

the covariance matrix (DxD)

the mean (Dx1)



Geometry of Multivariate Gaussian

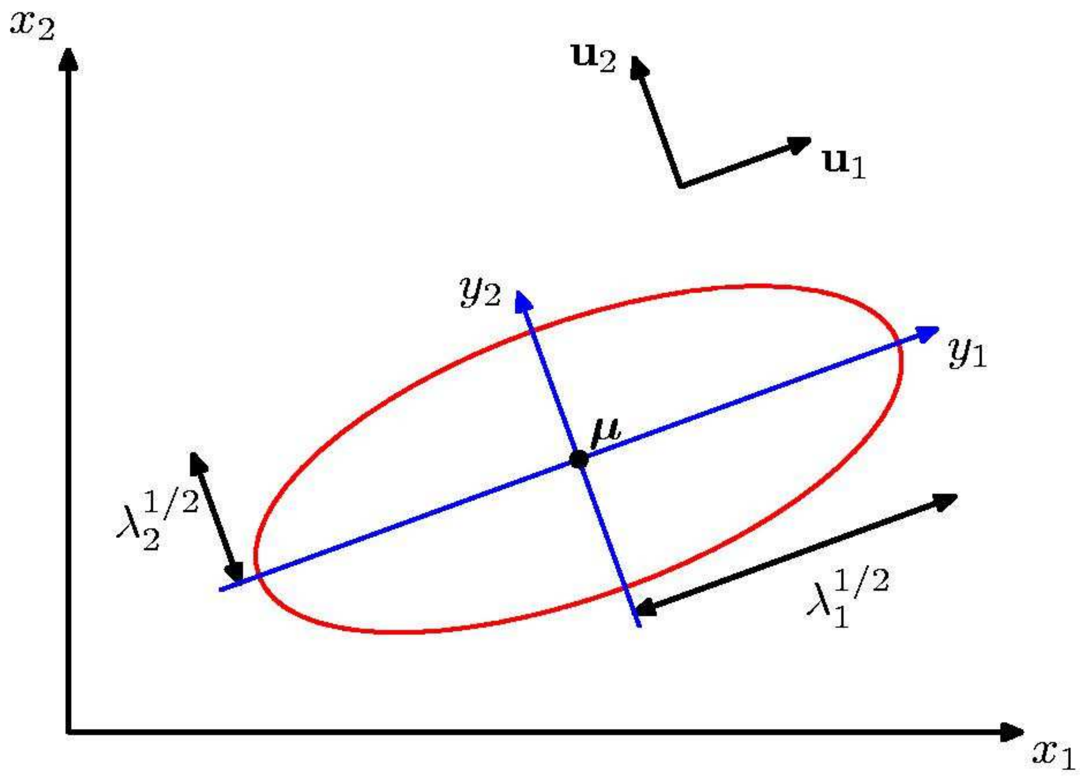
$$\Delta^2 = (\mathbf{x} - \boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})$$

Mahalanobis distance
(points with the same likelihood)

$$\boldsymbol{\Sigma}^{-1} = \sum_{i=1}^{D} \frac{1}{\lambda_i} \mathbf{u}_i \mathbf{u}_i^{\mathrm{T}}$$

$$\Delta^2 = \sum_{i=1}^{D} \frac{y_i^2}{\lambda_i}$$

$$y_i = \mathbf{u}_i^{\mathrm{T}} (\mathbf{x} - \boldsymbol{\mu})$$

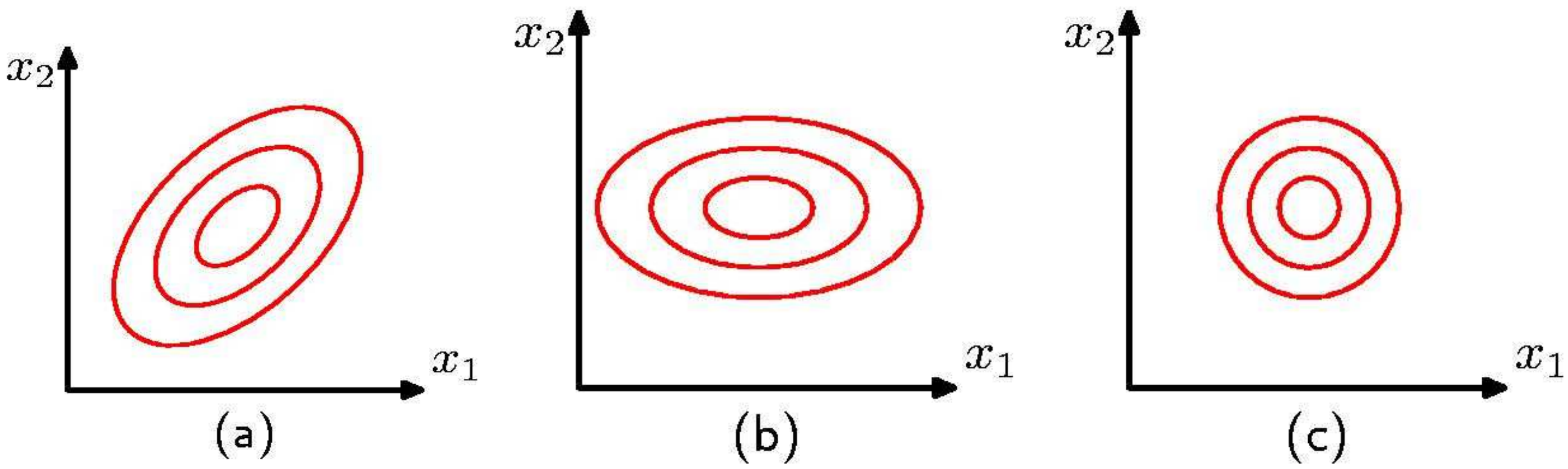


Three cases of Multivariate Gaussian

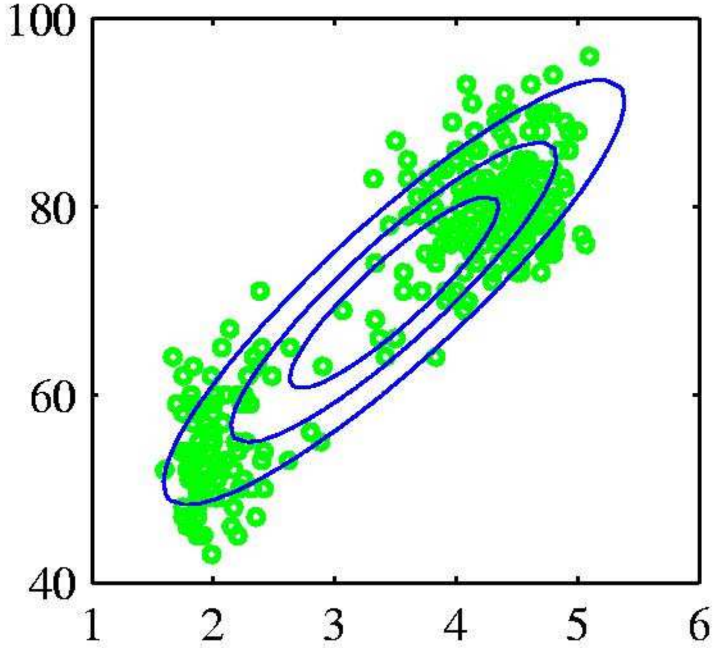
(a) general case: Σ is a symmetric, positive definite matrix ((d(d+1)/2 parameters)

(b) diagonal: Σ is diagonal (d parameters)

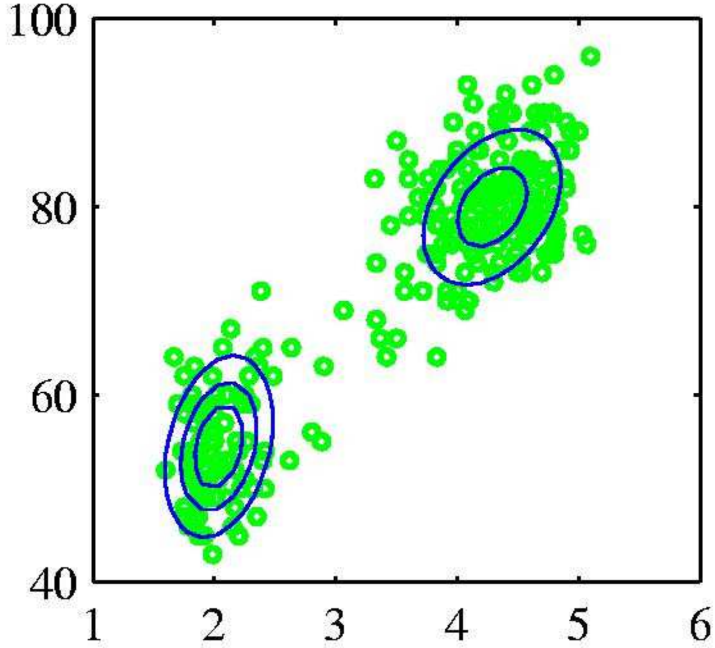
(c) circular: Σ is diagonal and all elements are the same (1 parameter),



Semi-parametric: mixture models

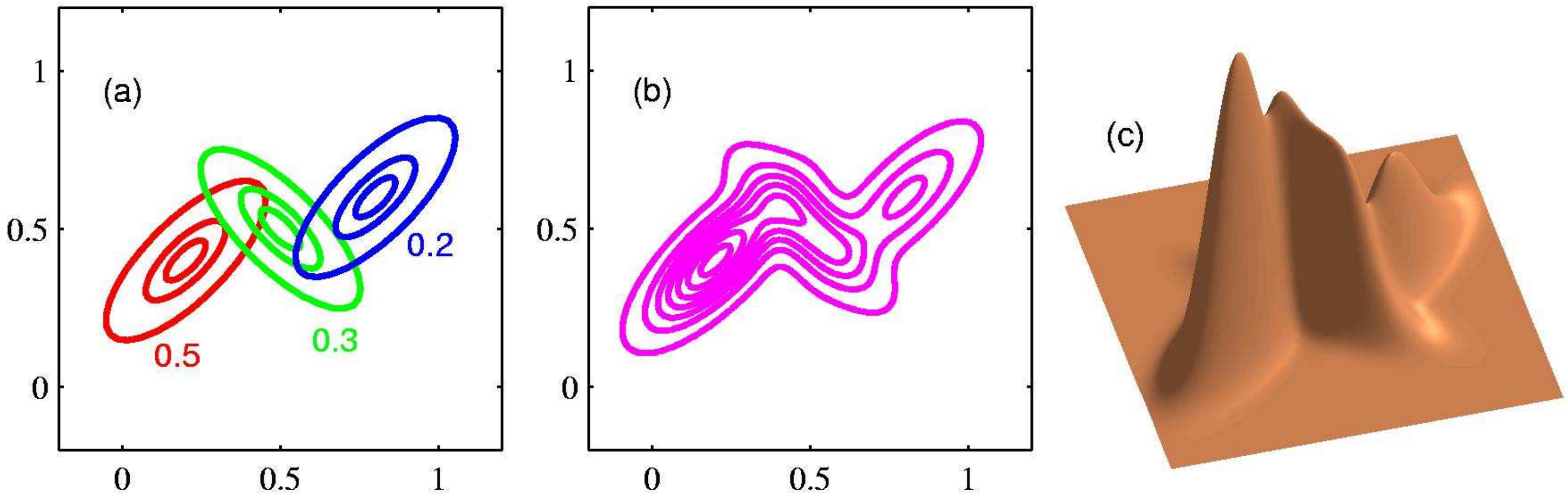


Single Gaussian



Mixture of two Gaussians

Mixtures of Gaussians



Nonparametric Methods

- Parametric distribution models are simple to estimate, but:
  - which "known" distribution to take?
  - does such a "known" distribution exist?
  - what about "combinations of several distributions"?

- Nonparametric approaches make few assumptions about the overall shape of the distribution being modelled.

Histograms

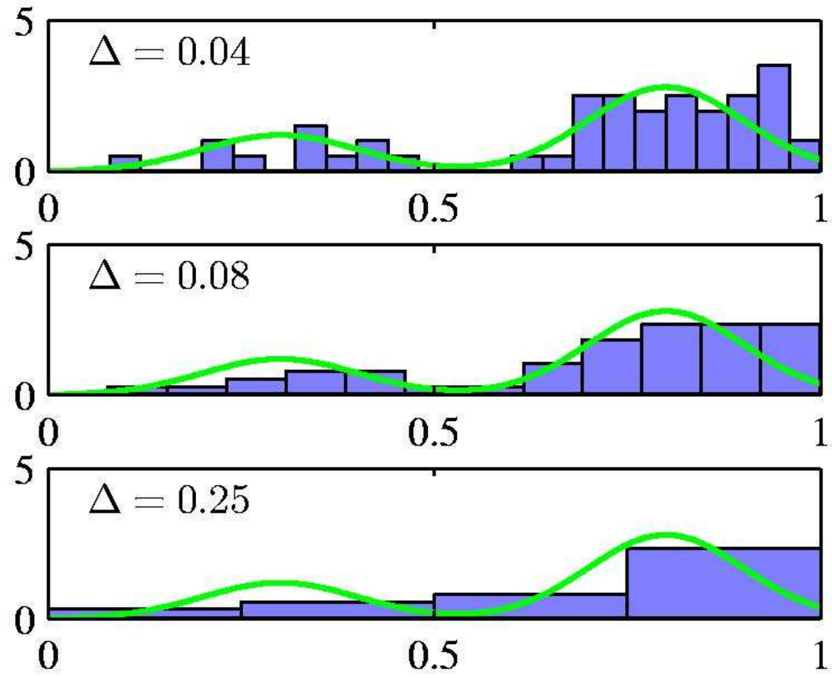
•Histogram methods partition the data space into distinct bins with widths Δ_i and count the number of observations, n_i, in each bin.

$$p_i = \frac{n_i}{N\Delta_i}$$

•Often, the same width is used for all bins, $\Delta_i = \Delta$.

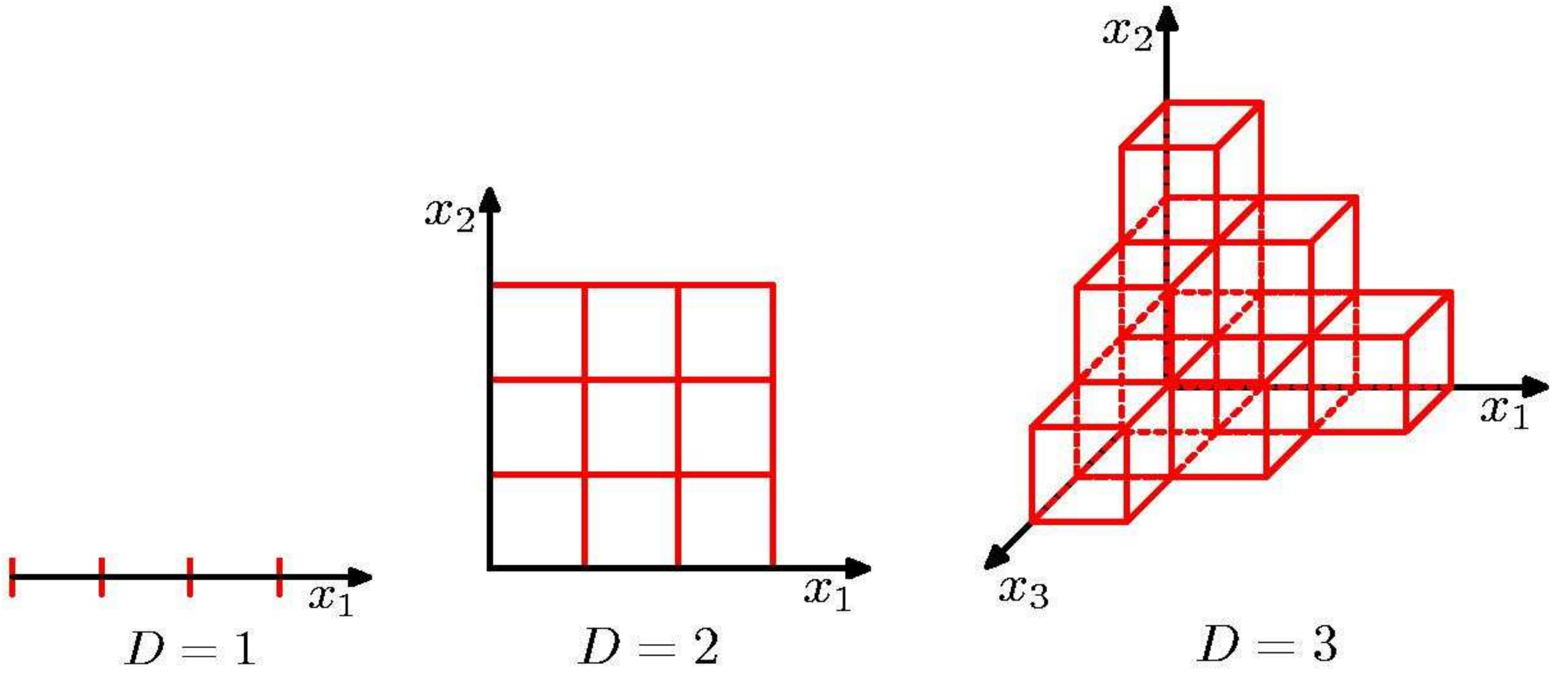
• Δ acts as a smoothing parameter *(how to chose it?)*

•Doesn't work for highly dimensional data!!!



• In a D-dimensional space, using M bins in each dimension will require M^D bins!

Curse of Dimensionality



Kernel Methods

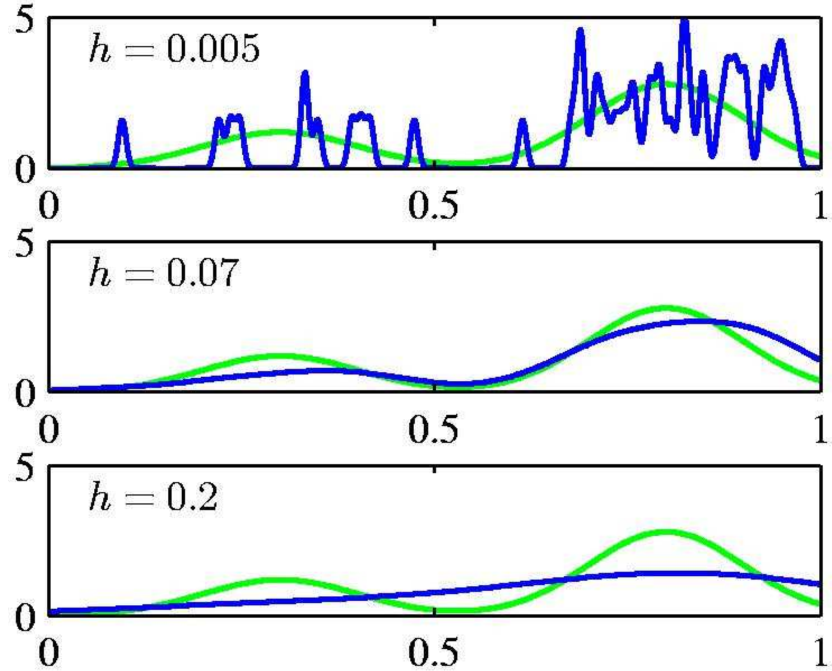
Surround each data point by
a “smooth kernel”, e.g.
a Gaussian

$$p(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{(2\pi h^2)^{D/2}} \exp\left\{-\frac{\|\mathbf{x}-\mathbf{x}_n\|^2}{2h^2}\right\}$$

Any kernel such that

$$\begin{aligned} k(\mathbf{u}) &\geqslant 0, \\ \int k(\mathbf{u})\,\mathrm{d}\mathbf{u} &= 1 \end{aligned}$$

will work.



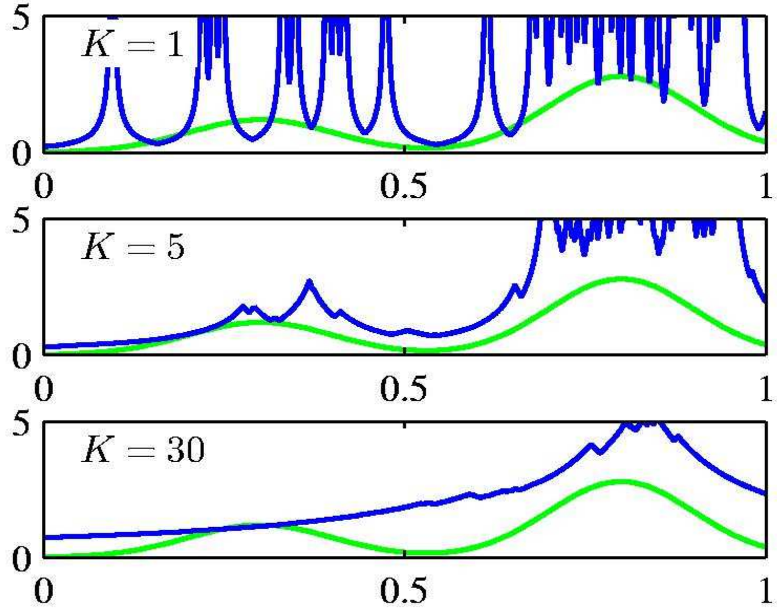
h acts as a smoother.

Nearest Neighbour Density Estimation

Fix K, estimate V from the data!

Consider a hypersphere centred on x and let it grow to a volume, V*, that includes K of the given N data points. Then

$$p(\mathbf{x}) \simeq \frac{K}{NV^{\star}}.$$



K acts as a smoother

Conclusions

- Parametric models: easy to compute (formulas), very efficient in terms of storage and computation.
  Key problem: how do we know which distribution to chose???
- Histograms are good for low dimensional data, but:
  - **location and size of bins?**
  - **“jumps” in density function**
  - **Curse of dimensionality**
- Nonparametric models (kernel- or NN- based) require storing the entire data set; computationally expensive; difficult choice of smoothing parameters
- Mixture models: attractive; computationally expensive, few design decisions needed (component distributions, number of components, initialization, stop criterion)

To Remember:

- Parametric models, density, likelihood, LogLikelihood
- Multidimensional Gaussians: covariance matrix, 3 cases (circular, diagonal, general), Mahalanobis distance
- Histograms, Curse of dimensionality
- Kernel methods: Parzen windows, Gaussians, smoothing parameter
- K-NN density estimation; relation of K-NN classifier to Bayes classifier
- Mixture models and the EM algorithm