Lecture 3: Probability, Statistics and Pattern Recognition

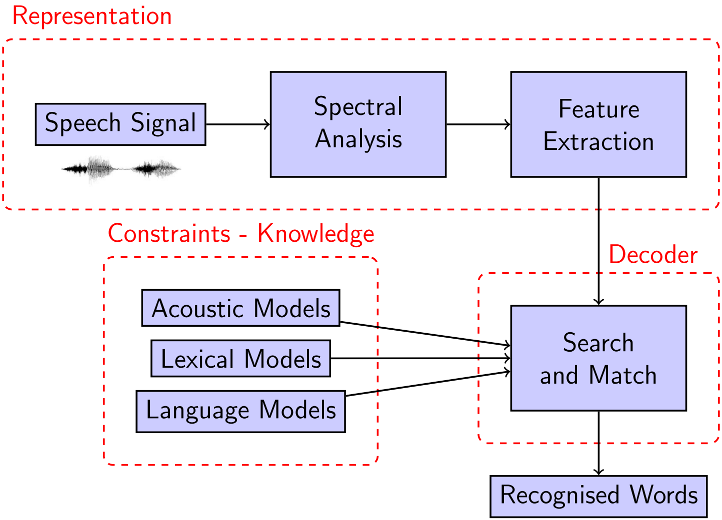
DT2118 Speech and Speaker Recognition

Giampiero Salvi

KTH/CSC/TMH giampi@kth.se

VT 2015

Components of ASR System



Different views on probabilities

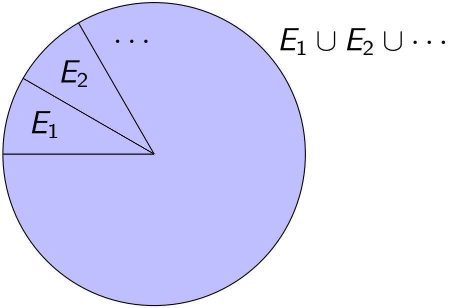
Axiomatic	defines axioms and derives properties
Classical	number of ways something can happen over total number of things that can happen (e.g. dice)
Logical	same, but weight the different ways
Frequency	frequency of success in repeated experiments
Propensity	
Subjective	degree of belief

Axiomatic view on probabilities (Kolmogorov)

Given an event E in a event space F

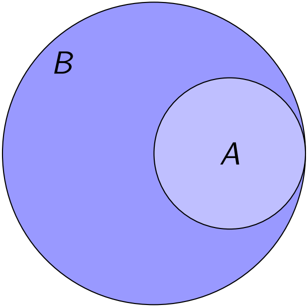
1. $P(E) \geq 0$ for all $E \in F$
2. sure event Ω: $P(\Omega) = 1$
3. $E_1, E_2, \ldots$ countable sequence of pairwise disjoint events, then

$$P(E_1 \cup E_2 \cup \cdots) = \sum_{i=1}^{\infty} P(E_i)$$



Consequences

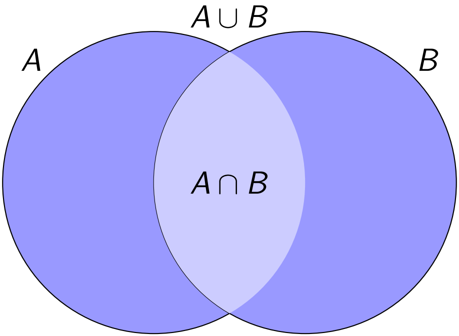
1. Monotonicity: $P(A) \leq P(B)$ if $A \subseteq B$



2. Empty set $\emptyset$: $P(\emptyset) = 0$
3. Bounds: $0 \leq P(E) \leq 1$ for all $E \in F$

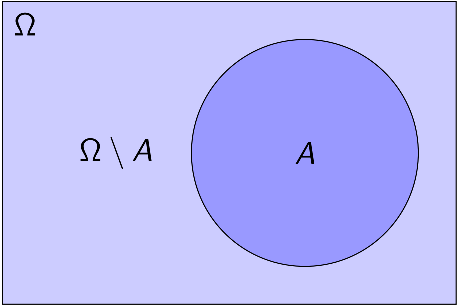
More Consequences: Addition

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$



More Consequences: Negation

$$P(\bar{A}) = P(\Omega \setminus A) = 1 - P(A)$$



Conditional Probabilities

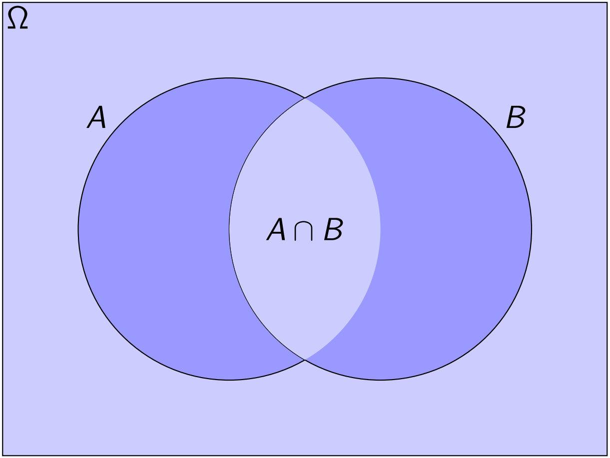
$$P(A|B)$$

The probability of event A when we *know* that event B has happened

Note: different from the probability that event A *and* event B happen

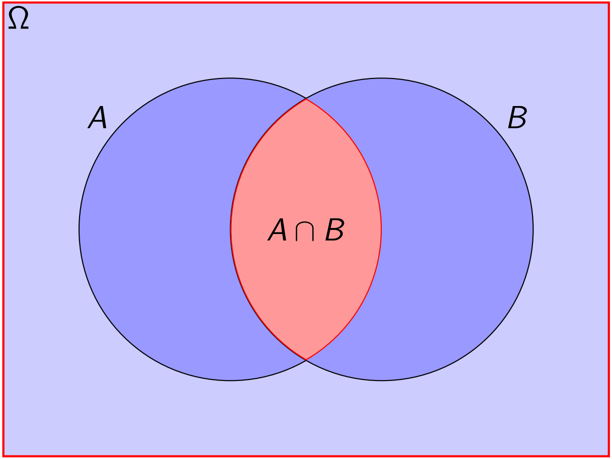
Conditional Probabilities

$$P(A|B) \neq P(A \cap B)$$



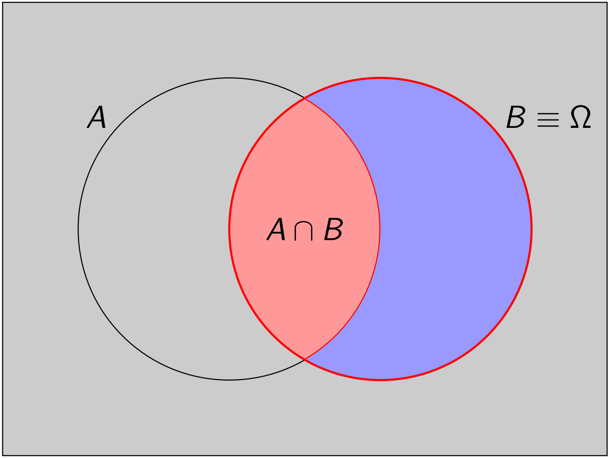
Conditional Probabilities

$$P(A|B) \neq P(A \cap B)$$



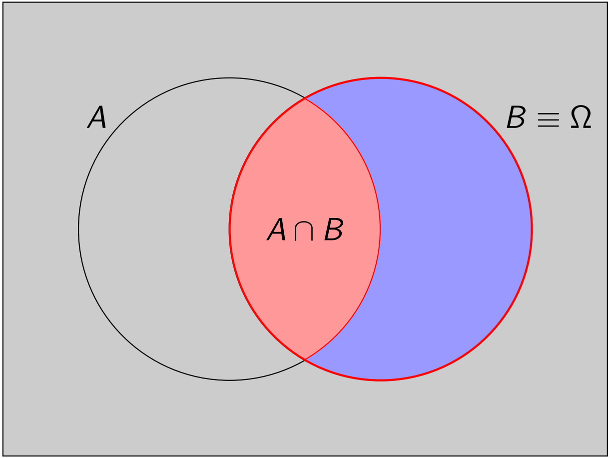
Conditional Probabilities

$$P(A|B) \neq P(A \cap B)$$



Conditional Probabilities

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$



Bayes' Rule

if

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

then

$$P(A \cap B) = P(A|B)P(B) = P(B|A)P(A)$$

and

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

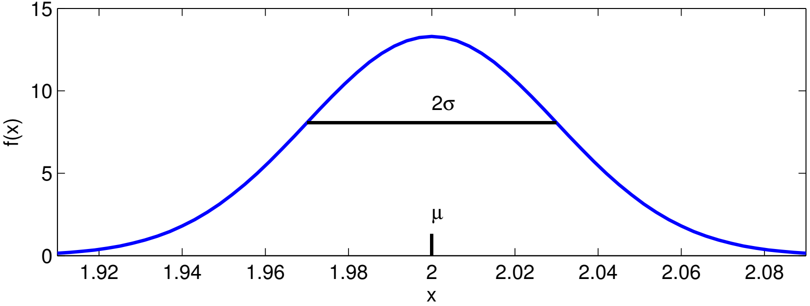
Discrete vs Continuous variables

- Discrete events: either 1, 2, 3, 4, 5, or 6.
- Discrete probability distribution $p(x) = P(d = x)$
- $P(d = 1) = 1/6$ (fair dice)

- Any real number (theoretically infinite)
- Distribution function (PDF) $f(x)$ (NOT PROBABILITY!!!)
- $P(t = 36.6) = 0$
- $P(36.6 < t < 36.7) = 0.1$

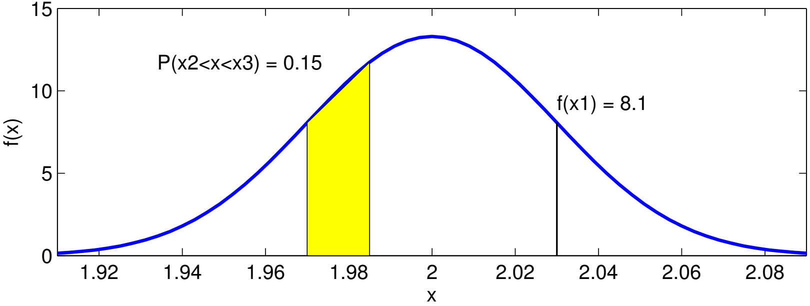
Gaussian distributions: One-dimensional

$$f(x|\mu, \sigma^2) = N(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$$



Gaussian distributions: One-dimensional

$$f(x|\mu,\sigma^2) = N(\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$$



Bayes rule with continuous variables

- Discrete case:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

- Continuous case (not probabilities)

$$P(A|x) = \frac{f(x|A)P(A)}{f(x)}$$

- Continuous case (probabilities)

$$P(A|x) = \frac{f(x|A)dxP(A)}{f(x)dx}$$

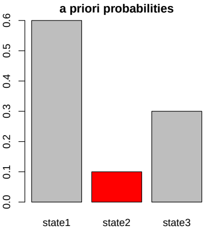
Gaussian distributions: d Dimensions

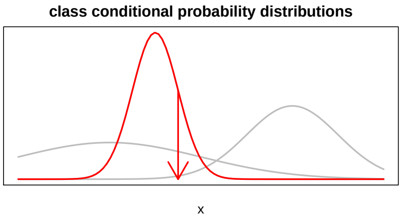
$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_d \end{bmatrix} \quad \mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \dots \\ \mu_d \end{bmatrix} \qquad \Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \dots & \sigma_{1d} \\ \sigma_{21} & \dots & & \\ \dots & & & \\ \sigma_{d1} & \dots & & \sigma_{dd} \end{bmatrix}$$

$$f(\mathbf{x}|\mu, \Sigma) = \frac{exp\left[-\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1}(\mathbf{x}-\mu)\right]}{(2\pi)^{\frac{d}{2}} \left|\Sigma\right|^{\frac{1}{2}}}$$

The Probabilistic Model of Classification

- "Nature" assumes one of c states ω_j with *a priori* probability $P(\omega_j)$
- When in state ω_j, "nature" emits observations $\hat{\mathbf{x}}$ with distribution $p(\mathbf{x}|\omega_j)$

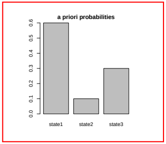


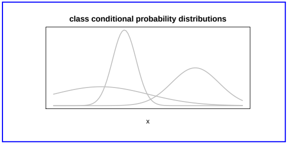


Problem

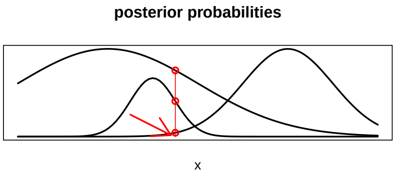
- If I observe $\hat{\mathbf{x}}$ and I know $P(\omega_j)$ and $p(\mathbf{x}|\omega_j)$ for each j
- what can I say about the state of "nature" ω_j?

Bayes decision theory

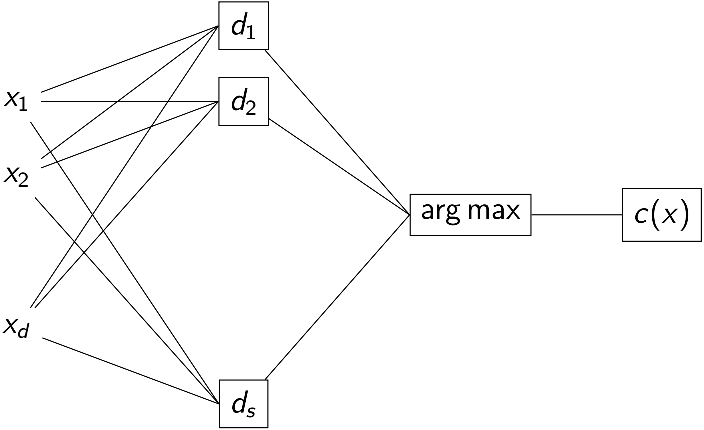




$$P(\omega_j|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_j)\ P(\omega_j)}{p(\mathbf{x})}$$



Classifiers: Discriminant Functions



$$d_i(\mathbf{x}) = p(\mathbf{x}|\omega_i)\ P(\omega_i)$$

Classifiers: Decision Boundaries

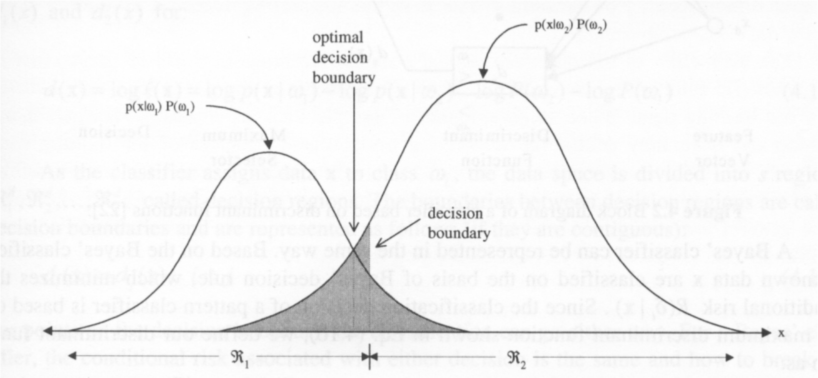


Figure from Huang, Acero, Hon.

Decision Boundaries in Two Dimensions

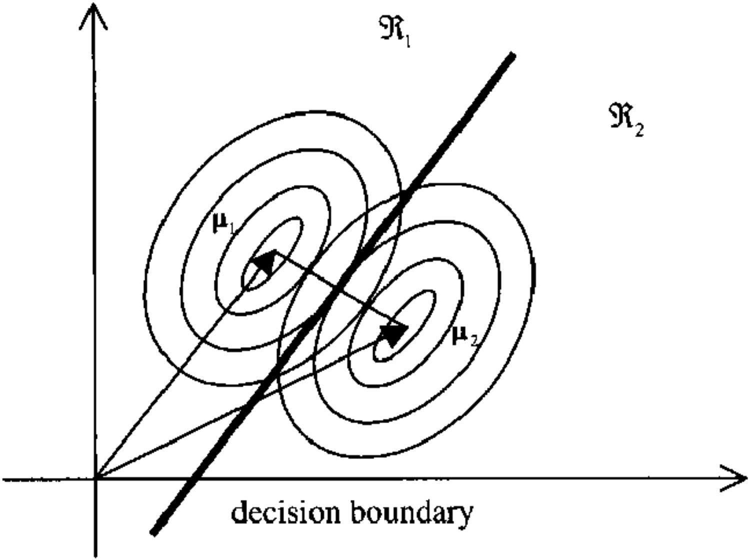


Figure from Huang, Acero, Hon.

Bayes' Rule and Pattern Recognition

$A =$ words, $B =$ sounds:

- During training we know the words and can compute $P(\text{sounds}|\text{words})$ using frequentist approach (repeated observations)
- during recognition we want
$\widehat{\text{words}} = \arg\max P(\text{words}|\text{sounds})$
- using Bayes' rule:

$$P(\text{words}|\text{sounds}) = \frac{P(\text{sounds}|\text{words})P(\text{words})}{P(\text{sounds})}$$

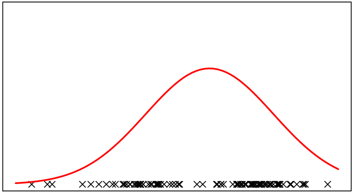
where
$P(\text{words})$: *a priori* probability of the words (Language Model)
$P(\text{sounds})$: *a priori* probability of the sounds (constant, can be ignored)

Estimation Theory

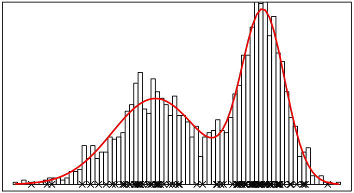
- so far we assumed we know $P(\omega_j)$ and $p(\mathbf{x}|\omega_j)$
- how can we obtain them from collections of data?
- this is the subject of Estimation Theory

Parametric vs Non-Parametric Estimation

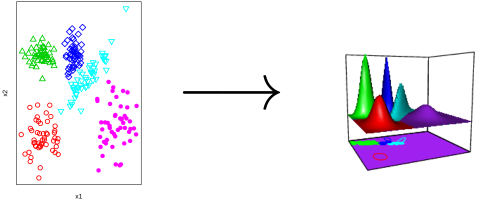
Parametric



non parametric



Parameter estimation

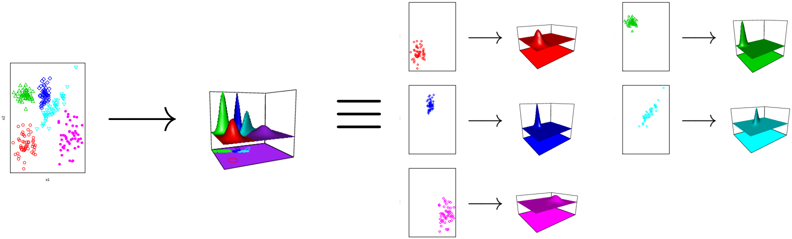


Assumptions:

- samples from class ω_i do not influence estimate for class $\omega_j,\ i \neq j$
- samples from the same class are independent and identically distributed (i.i.d.)

Parameter estimation (cont.)

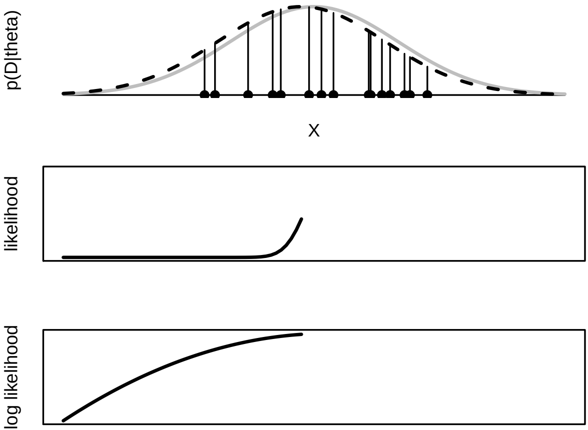
- class independence assumption:



- Maximum likelihood estimation
- Maximum a posteriori estimation
- Bayesian estimation

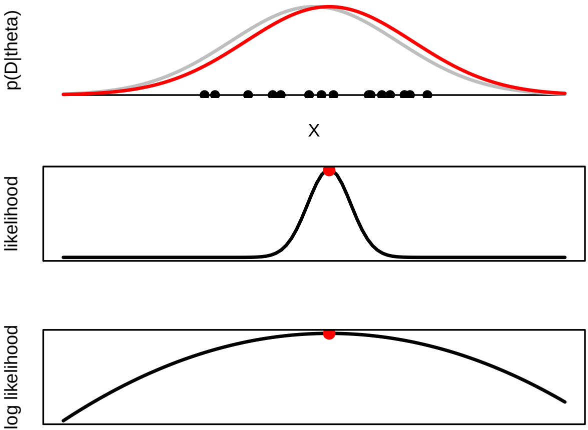
Maximum likelihood estimation

- Find parameter vector $\hat{\theta}$ that maximises $p(\mathcal{D}|\theta)$ with $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$
- i.i.d. $\rightarrow p(\mathcal{D}|\theta) = \prod_{k=1}^{n} p(\mathbf{x}_k|\theta)$



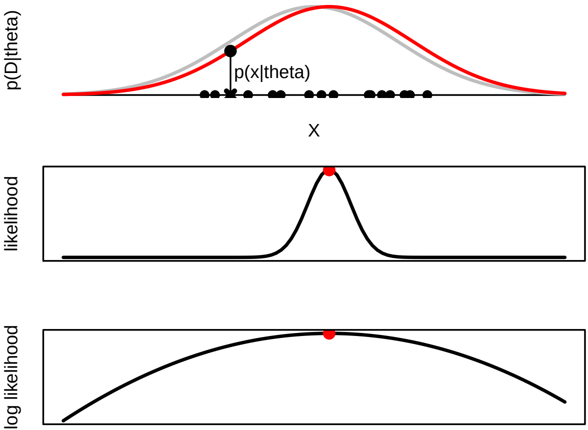
Maximum likelihood estimation

- Find parameter vector $\hat{\theta}$ that maximises $p(\mathcal{D}|\theta)$ with $\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$
- i.i.d. $\rightarrow p(\mathcal{D}|\theta) = \prod_{k=1}^{n} p(\mathbf{x}_k|\theta)$



Maximum likelihood estimation

- Find parameter vector $\hat{\theta}$ that maximises $p(\mathcal{D}|\theta)$ with $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$
- i.i.d. $\rightarrow p(\mathcal{D}|\theta) = \prod_{k=1}^{n} p(\mathbf{x}_k|\theta)$



ML estimation of Gaussian mean

$$N(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right], \text{ with } \theta = \{\mu,\sigma^2\}$$

Log-likelihood of data (i.i.d. samples):

$$\log P(\mathcal{D}|\theta) = \sum_{i=1}^{N} \log N(x_i|\mu,\sigma^2) = -N\log\left(\sqrt{2\pi\sigma}\right) - \sum_{i=1}^{N} \frac{(x_i-\mu)^2}{2\sigma^2}$$

$$0 = \frac{d\log P(\mathcal{D}|\theta)}{d\mu} = \sum_{i=1}^{N} \frac{(x_i-\mu)}{\sigma^2} = \frac{\sum_{i=1}^{N} x_i - N\mu}{\sigma^2} \iff$$

$$\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

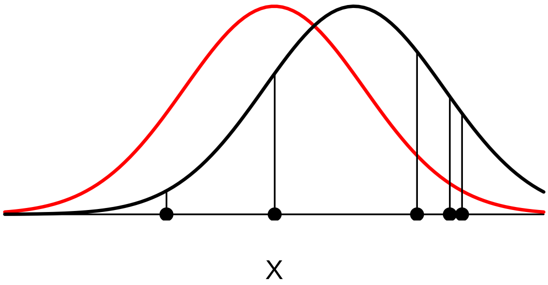
ML estimation of Gaussian parameters

$$\begin{aligned}\hat{\mu} &= \frac{1}{N}\sum_{i=1}^{N} x_i \\ \hat{\sigma}^2 &= \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{\mu})^2\end{aligned}$$

- same result by minimizing the sum of square errors!
- but we make assumptions explicit

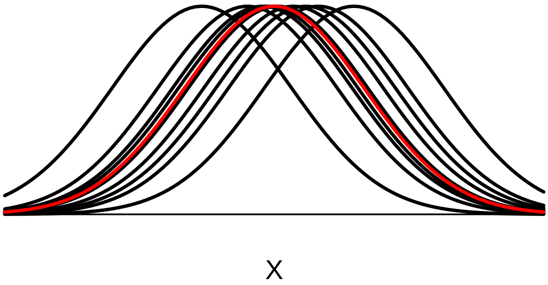
Problem: few data points

10 repetitions with 5 points each



Problem: few data points

10 repetitions with 5 points each



Maximum a Posteriori Estimation

$$\hat{\mu}, \hat{\sigma}^2 = \arg\max_{\mu,\sigma^2} \left[\prod_{i=1}^{N} P(x_i|\mu, \sigma^2) P(\mu, \sigma^2) \right]$$

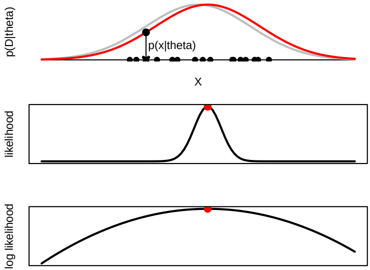
where the prior $P(\mu, \sigma^2)$ needs a nice mathematical form for closed solution

$$\begin{aligned} \hat{\mu}_{\text{MAP}} &= \frac{N}{N+\gamma}\,\hat{\mu}_{\text{ML}} + \frac{\gamma}{N+\gamma}\,\delta \\ \hat{\sigma}^2_{\text{MAP}} &= \frac{N}{N+3+2\alpha}\hat{\sigma}^2_{\text{ML}} + \frac{2\beta + \gamma(\delta + \hat{\mu}_{\text{MAP}})^2}{N+3+2\alpha} \end{aligned}$$

where $\alpha, \beta, \gamma, \delta$ are parameters of the prior distribution

ML, MAP and Point Estimates

- Both ML and MAP produce point estimates of θ
- Assumption: there is a true value for θ
- advantage: once $\hat{\theta}$ is found, everything is known



Overfitting

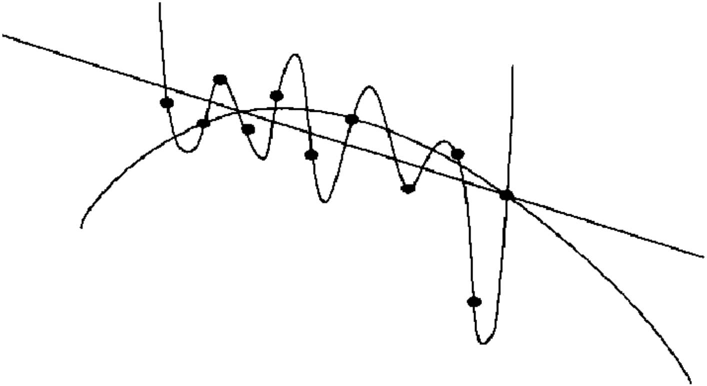


Figure from Huang, Acero, Hon.

Overfitting: Phoneme Discrimination

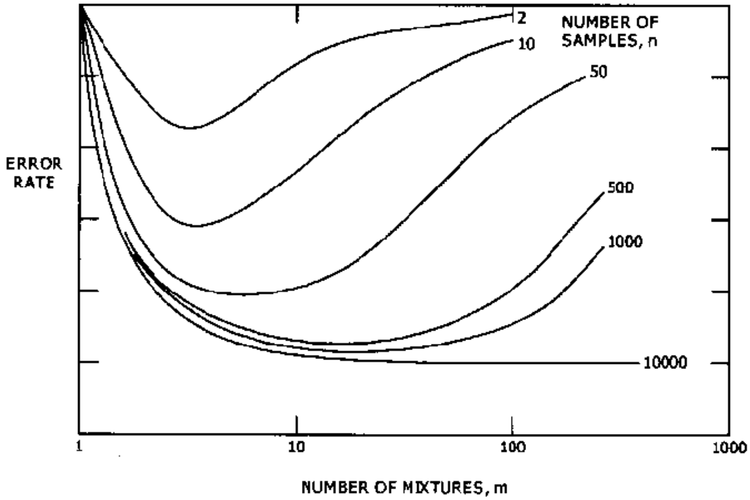


Figure from Huang, Acero, Hon.

Bayesian estimation

- Consider θ as a random variable
- characterize θ with the posterior distribution $P(\theta|\mathcal{D})$ given the data

$$\begin{array}{llcl} \text{ML:} & \mathcal{D} & \rightarrow & \hat{\theta}_{\text{ML}} \\ \text{MAP:} & \mathcal{D}, P(\theta) & \rightarrow & \hat{\theta}_{\text{MAP}} \\ \text{Bayes:} & \mathcal{D}, P(\theta) & \rightarrow & P(\theta|\mathcal{D}) \end{array}$$

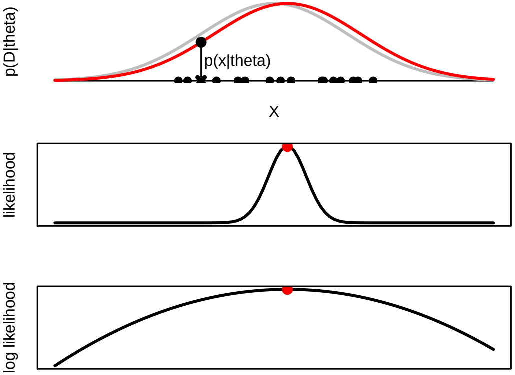
- for new data points, instead of $P(\mathbf{x}_{\text{new}}|\hat{\theta}_{\text{ML}})$ or $P(\mathbf{x}_{\text{new}}|\hat{\theta}_{\text{MAP}})$, compute:

$$P(\mathbf{x}_{\text{new}}|\mathcal{D}) = \int_{\theta\in\Theta} P(\mathbf{x}_{\text{new}}|\theta)P(\theta|\mathcal{D})d\theta$$

Bayesian estimation (cont.)

- we can compute $p(\mathbf{x}|\mathcal{D})$ instead of $p(\mathbf{x}|\hat{\theta})$
  - integrate the joint density $p(\mathbf{x},\theta|\mathcal{D}) = p(\mathbf{x}|\theta)p(\theta|\mathcal{D})$

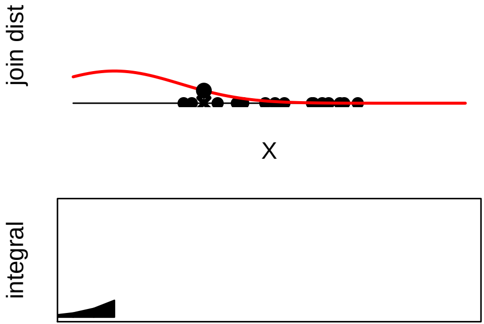
$$p(\mathbf{x}|\hat{\theta})$$



Bayesian estimation

- we can compute $p(\mathbf{x}|\mathcal{D})$ instead of $p(\mathbf{x}|\hat{\theta})$
  - integrate the joint density $p(\mathbf{x},\theta|\mathcal{D}) = p(\mathbf{x}|\theta)p(\theta|\mathcal{D})$

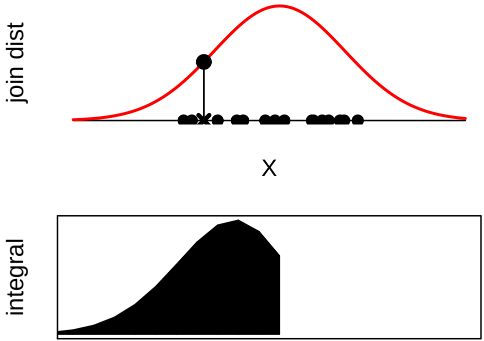
$$p(\mathbf{x}|\mathcal{D}) = \int p(\mathbf{x}|\theta)p(\theta|\mathcal{D})d\theta$$



Bayesian estimation

- we can compute $p(\mathbf{x}|\mathcal{D})$ instead of $p(\mathbf{x}|\hat{\theta})$
  - integrate the joint density $p(\mathbf{x},\theta|\mathcal{D}) = p(\mathbf{x}|\theta)p(\theta|\mathcal{D})$

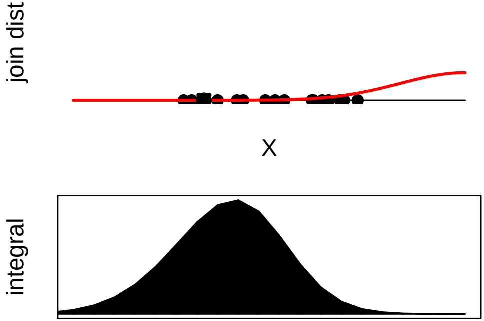
$$p(\mathbf{x}|\mathcal{D}) = \int p(\mathbf{x}|\theta)p(\theta|\mathcal{D})d\theta$$



Bayesian estimation

- we can compute $p(\mathbf{x}|\mathcal{D})$ instead of $p(\mathbf{x}|\hat{\theta})$
  - integrate the joint density $p(\mathbf{x}, \theta|\mathcal{D}) = p(\mathbf{x}|\theta)p(\theta|\mathcal{D})$

$$p(\mathbf{x}|\mathcal{D}) = \int p(\mathbf{x}|\theta)p(\theta|\mathcal{D})d\theta$$



Bayesian estimation (cont.)

Pros:

- better use of the data
- makes a priori assumptions explicit
- easily implemented recursively
  - use posterior $p(\theta|\mathcal{D})$ as new prior
- reduce overfitting

Cons:

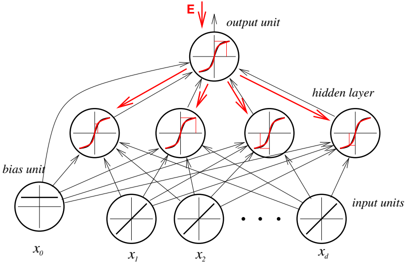
- definition of noninformative priors can be tricky
- often requires numerical integration

Other Training Strategies: Discriminative Training

- Maximum Mutual Information Estimation
- Minimum Error Rate Estimation
- Neural Networks

Multi layer neural networks

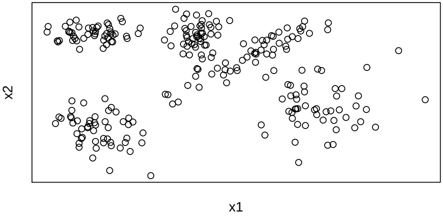
Multi layer
neural networks



- Backpropagation algorithm

Unsupervised Learning

- so far we assumed we knew the class ω_i for each data point
- what if we don't?
- class independence assumption loses meaning



Vector Quantisation, K-Means

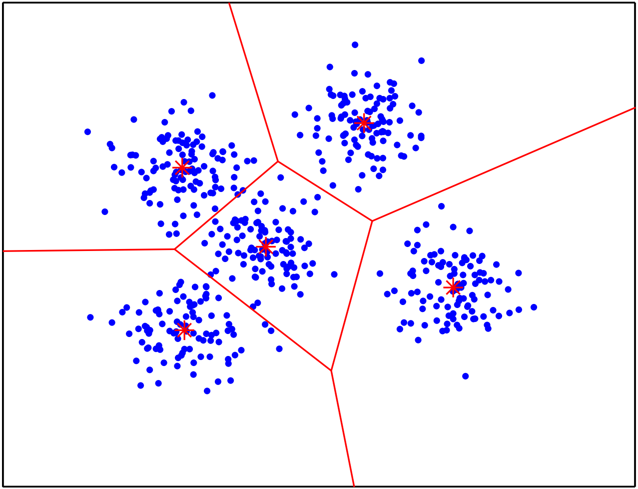
- describes each class with a centroid
- a point belongs to a class if the corresponding centroid is closest (Euclidean distance)
- iterative procedure
- guaranteed to converge
- not guaranteed to find the optimal solution
- used in vector quantization

K-means: algorithm

Data: k (number of desired clusters), n data points $\mathbf{x}_i$
Result: k clusters
initialization: assign initial value to k centroids $\mathbf{c}_i$;
repeat
    assign each point $\mathbf{x}_i$ to closest centroid $\mathbf{c}_j$;
    compute new centroids as mean of each group of points;
until *centroids do not change*;
return k clusters;

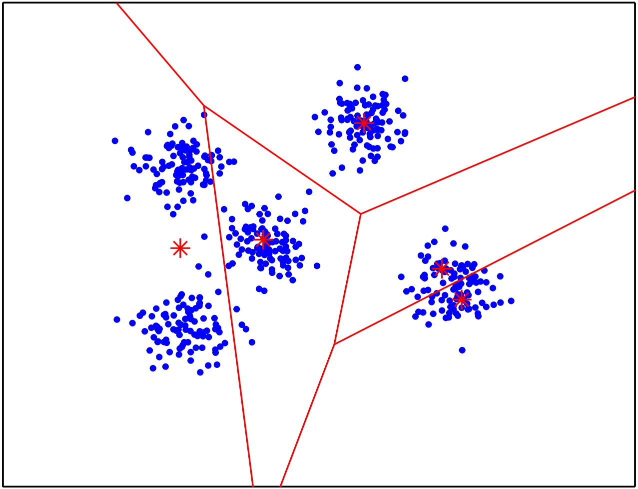
K-means: example

iteration 20, update clusters



K-means: sensitivity to initial conditions

iteration 20, update clusters



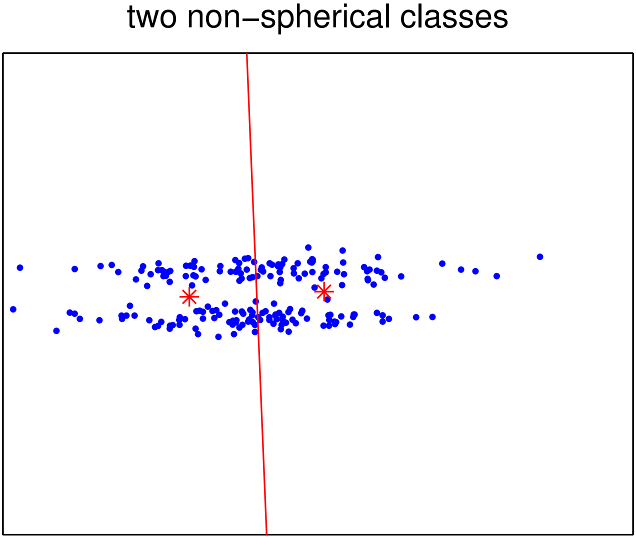
Solution: LBG Algorithm

- Linde–Buzo–Gray
- start with one centroid
- adjust to mean
- split centroid (with ϵ)
- K-means
- split again…

K-means: limits of Euclidean distance

- the Euclidean distance is isotropic (same in all directions in $\mathbb{R}^p$)
- this favours spherical clusters
- the size of the clusters is controlled by their distance

K-means: non-spherical classes



Probabilistic Clustering

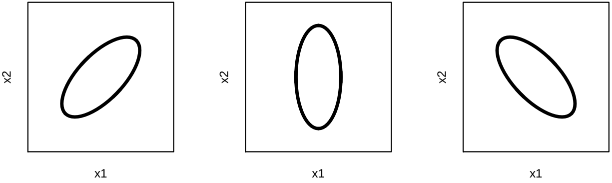
- model data as a mixture of probability distributions (Gaussian)
- each distribution corresponds to a cluster
- clustering corresponds to parameter estimation

Gaussian distributions

$$f_k(\mathbf{x}_i|\mu_k, \Sigma_k) = \frac{exp\left\{-\frac{1}{2}(\mathbf{x}_i - \mu_k)^T \Sigma_k^{-1}(\mathbf{x}_i - \mu_k)\right\}}{(2\pi)^{\frac{p}{2}} \left|\Sigma_k\right|^{\frac{1}{2}}}$$

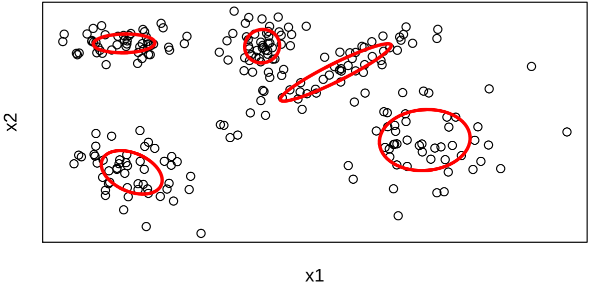
Eigenvalue decomposition of the covariance matrix:

$$\Sigma_k = \lambda_k D_k A_k D_k^T$$



Mixture of Gaussian distributions

Σ_k	Distribution	Volume	Shape	Orientation
λI	Spherical	Equal	Equal	N/A
$\lambda_k I$	Spherical	Variable	Equal	N/A
λDAD^T	Ellipsoidal	Equal	Equal	Equal
$\lambda D_k A D_k^T$	Ellipsoidal	Equal	Equal	Variable
$\lambda_k D_k A D_k^T$	Ellipsoidal	Variable	Equal	Variable
$\lambda_k D_k A_k D_k^T$	Ellipsoidal	Variable	Variable	Variable



Fitting the model

- given the data $D = \{\mathbf{x}_i\}$
- given a certain model $\mathcal{M}$ and its parameters θ
- maximize the model fit to the data as expressed by the likelihood

$$\mathcal{L} = p(D|\theta)$$

Unsupervised Case

- release class independence assumption:
- learn the mixture at once
- problem of missing data
- solution: Expectation Maximization

Expectation Maximization

- let $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ be the data (observations) drawn from K distributions (known)
- we call $z_j \in [1, K]$ the index of the Gaussian that generated the point $\mathbf{x}_j$ (unknown)
- the combination of $\mathbf{x}$ and $\mathbf{z}$ is called the complete data
- the probability that the ith Gaussian generates a particular $\mathbf{x}$ is proportional to

$$p(\mathbf{x}|z = i, \theta) = \mathcal{N}(\mu_i, \Sigma_i)$$

Expectation Maximization 2

- the task is to estimate the unknown parameters

$$\begin{aligned}\theta \; = \; & \{\mu_1, \ldots, \mu_K, \Sigma_1, \ldots, \Sigma_K, \\ & P(z=1), \ldots, P(z=K)\}\end{aligned}$$

- to do so we iterate between improving our knowledge about $\mathbf{z}$ and improving the estimate of θ given this knowledge

EM: formulation

E-step estimate the probability of z given the observation and the current model:

$$P(z_j = i|\mathbf{x}_j, \theta_t)$$

M-step 1) compute the expected log-likelihood of the complete data $(\mathbf{x}, \mathbf{z})$

$$Q(\theta) = E_z\left[\ln \prod_{j=1}^{n} p(\mathbf{x}_j, z|\theta)|\mathbf{x}_j\right]$$

2) maximize $Q(\theta)$ with respect to the model parameters θ

EM and GM: properties

- the variance in the GM model must be constrained (to avoid infinite likelihood)
- EM is guaranteed to converge to a *local* maximum of the complete data likelihood
- the initial conditions play an important role (as with K-means)
- GM and EM are quivalent to K-means when the covariances are all equal to the identity matrix
- equal covariances lead to linear discriminants

Expectation Maximization

Fitting model parameters with missing (latent) variables

$$P(\mathbf{x}|\theta) = \sum_{k=1}^{K} \pi_k P(x|\theta_k),$$

$$\text{with } \theta = \{\pi_1, \ldots, \pi_k, \theta_1, \ldots, \theta_K\}$$

- very general idea (applies to many different probabilistic models)
- augment the data with the missing variables: h_{ik} probability of assignment of each data point x_i to each component of the mixture k
- optimize the Likelihood of the complete data:

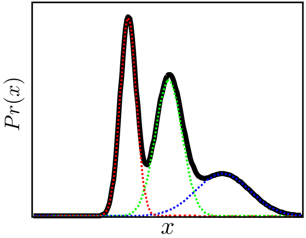
$$P(\mathbf{x}, \mathbf{h}|\theta)$$

Mixture of Gaussians

This distribution is a weight sum of K Gaussian distributions

$$P(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x; \mu_k, \sigma_k^2)$$

where $\pi_1 + \cdots + \pi_K = 1$
and $\pi_k > 0$ $(k = 1, \ldots, K)$.



This model can describe **complex multi-modal** probability distributions by combining simpler distributions.

Mixture of Gaussians

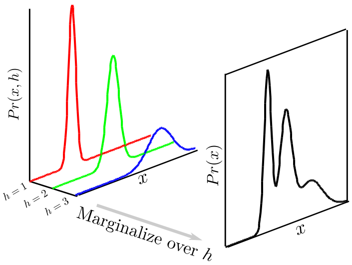
$$P(x) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(x; \mu_k, \sigma_k^2)$$

- Learning the parameters of this model from training data $x_1, \ldots, x_n$ is not trivial - using the usual straightforward maximum likelihood approach.
- Instead learn parameters using the **Expectation-Maximization** (EM) algorithm.

Mixture of Gaussians as a marginalization

We can interpret the Mixture of Gaussians model with the introduction of a discrete hidden/latent variable h and $P(x, h)$:

$$P(x) = \sum_{k=1}^{K} P(x, h = k) = \sum_{k=1}^{K} P(x \mid h = k) P(h = k)$$
$$= \sum_{k=1}^{K} \pi_k \mathcal{N}(x; \mu_k, \sigma_k^2)$$



← **mixture density**

Figures taken from **Computer Vision: models, learning and inference** by Simon Prince.

EM for two Gaussians

Assume: We know the pdf of x has this form:

$$P(x) = \pi_1 \mathcal{N}(x; \mu_1, \sigma_1^2) + \pi_2 \mathcal{N}(x; \mu_2, \sigma_2^2)$$

where $\pi_1 + \pi_2 = 1$ and $\pi_k > 0$ for components $k = 1, 2$.

Unknown: Values of the parameters (Many!)

$$\Theta = (\pi_1, \mu_1, \sigma_1, \mu_2, \sigma_2).$$

Have: Observed n samples $x_1, \ldots, x_n$ drawn from $P(x)$.

Want to: Estimate Θ from $x_1, \ldots, x_n$.

How would it be possible to get them all???

EM for two Gaussians

For each sample x_i introduce a *hidden variable* h_i

$$h_i = \begin{cases} 1 & \text{if sample } x_i \text{ was drawn from } \mathcal{N}(x; \mu_1, \sigma_1^2) \\ 2 & \text{if sample } x_i \text{ was drawn from } \mathcal{N}(x; \mu_2, \sigma_2^2) \end{cases}$$

and come up with initial values

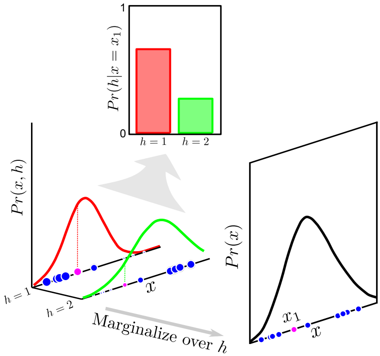
$$\Theta^{(0)} = \left(\pi_1^{(0)}, \mu_1^{(0)}, \sigma_1^{(0)}, \mu_2^{(0)}, \sigma_2^{(0)}\right)$$

for each of the parameters.

EM is an *iterative algorithm* which updates $\Theta^{(t)}$ using the following two steps...

EM for two Gaussians: E-step

The responsibility of k-th Gaussian for each sample x (indicated by the size of the projected data point)



Look at each sample x along hidden variable h in the E-step

Figure from **Computer Vision: models, learning and inference** by Simon Prince.

EM for two Gaussians: E-step (cont.)

E-step: Compute the *"posterior probability"* that x_i was generated by component k given the current estimate of the parameters $\Theta^{(t)}$. (responsibilities)

for $i = 1, \ldots n$

for $k = 1, 2$

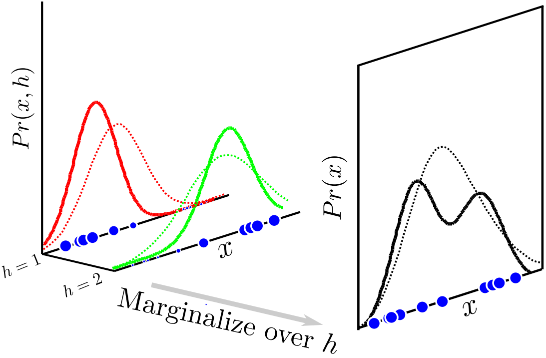
$$\gamma_{ik}^{(t)} = P(h_i = k \mid x_i, \Theta^{(t)})$$

$$= \frac{\pi_k^{(t)} \mathcal{N}(x_i; \mu_k^{(t)}, \sigma_k^{(t)})}{\pi_1^{(t)} \mathcal{N}(x_i; \mu_1^{(t)}, \sigma_1^{(t)}) + \pi_2^{(t)} \mathcal{N}(x_i; \mu_2^{(t)}, \sigma_2^{(t)})}$$

Note: $\gamma_{i1}^{(t)} + \gamma_{i2}^{(t)} = 1$ and $\pi_1 + \pi_2 = 1$

EM for two Gaussians: M-step

Fitting the Gaussian model for each of k-th constinuetnt.
Sample x_i contributes according to the responsibility γ_{ik}.



(dashed and solid lines for fit before and after update)

Look along samples x for each h in the M-step

Figure from **Computer Vision: models, learning and inference** by Simon Prince.

EM for two Gaussians: M-step (cont.)

M-step: Compute the *Maximum Likelihood* of the parameters of the mixture model given out data's membership distribution, the $\gamma_i^{(t)}$'s:

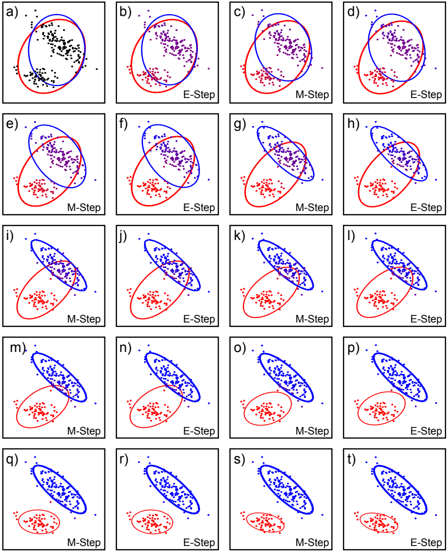
for $k = 1, 2$

$$\mu_k^{(t+1)} = \frac{\sum_{i=1}^n \gamma_{ik}^{(t)} x_i}{\sum_{i=1}^n \gamma_{ik}^{(t)}},$$

$$\sigma_k^{(t+1)} = \sqrt{\frac{\sum_{i=1}^n \gamma_{ik}^{(t)} (x_i - \mu_k^{(t+1)})^2}{\sum_{i=1}^n \gamma_{ik}^{(t)}}},$$

$$\pi_k^{(t+1)} = \frac{\sum_{i=1}^n \gamma_{ik}^{(t)}}{n}.$$

EM in practice



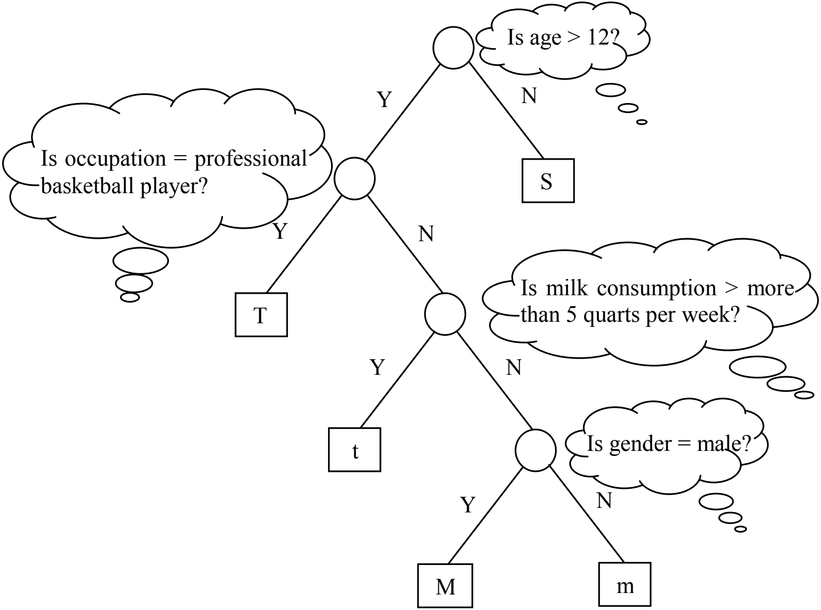
Classification and Regression Tree (CART)

gender	age	occupation	milk consumption (litres/day)	height (meters)
male	23	basketball player	1.0	2.0
female	22	student	0.5	1.6
male	13	student	0.2	1.3
female	8	student	0.5	1.2
female	72	retired	0.1	1.7
...	...	...	...	...

Classification and Regression Tree (CART)

- Binary decision tree
- An automatic and data-driven framework to construct a decision process based on objective criteria
- Handles data samples with mixed types, nonstandard structures
- Handles missing data, robust to outliers and mislabeled data samples
- Used in speech recognition for model tying

Classification and Regression Tree (CART)



Steps in constructing a CART

- Find set of questions
- Put all training samples in root
- Recursive algorithm
  - Find the best combination of question and node. Split the node into two new nodes
  - Move the corresponding data into the new nodes
  - Repeat until right-sized tree is obtained
- Greedy algorithm, only locally optimal, splitting without regard to subsequent splits

Defining questions

Data described by $\mathbf{x} = (x_1, x_2, \ldots, x_d)$

- one question per variable (singleton questions)
- If x_i discrete with values in $\{c_1, \ldots, c_K\}$, questions in the form: is $x_i \in S$?, with S subset of the values.
- If x_i continuous, questions in the form: is $x_i \leq c$?, with c real number.
- in both cases, finite number of questions for a dataset

Splitting Criterium

- we want data points in each leaf to be homogeneous
- Find the pair of node and question for which the split gives largest improvement
- Examples:
  1. Largest decrease in class entropy
  2. Largest decrease in squared error from a regression of the data in the node