

CS 559: Machine Learning Fundamentals and Applications (Midterm Recap)

Instructor: Philippos Mordohai
Webpage: www.cs.stevens.edu/~mordohai
E-mail: Philippos.Mordohai@stevens.edu
Office: Lieb 215

Midterm

- October 21 (first part of the class)
 - Open books/notes
 - No graphing calculators

Outline

- **Probability theory**
- Bayes decision theory
- Maximum-likelihood and Bayesian parameter estimation
- Expectation maximization
- Non-parametric techniques
- Hidden Markov models
- Principal component analysis

Pairs of Discrete Random Variables

- Let x and y be two discrete r.v.
- For each possible pair of values, we can define a *joint probability* $p_{ij} = \Pr[x=x_i, y=y_j]$
- We can also define a *joint probability mass function* $P(x,y)$ which offers a complete characterization of the pair of r.v.

$$P_x(x) = \sum_{y \in Y} P(x, y)$$

Marginal distributions

$$P_y(y) = \sum_{x \in X} P(x, y)$$

Note that P_x and P_y are different functions

Conditional Probability

- When two r.v. are not independent, knowing one allows better estimate of the other (e.g. outside temperature, season)

$$\Pr[x = x_i | y = y_j] = \frac{\Pr[x = x_i, y = y_j]}{\Pr[y = y_j]}$$

- If independent $P(x|y)=P(x)$

Law of Total Probability

- If an event A can occur in m different ways and if these m different ways are mutually exclusive, then the probability of A occurring is the sum of the probabilities of the sub-events

$$P(X = x_i) = \sum_j P(X = x_i | Y = y_j)P(Y = y_j)$$

Bayes Rule

$$P(x | y) = \frac{P(x, y)}{P(y)} = \frac{P(y | x)P(x)}{\sum_{x \in X} P(x, y)}$$

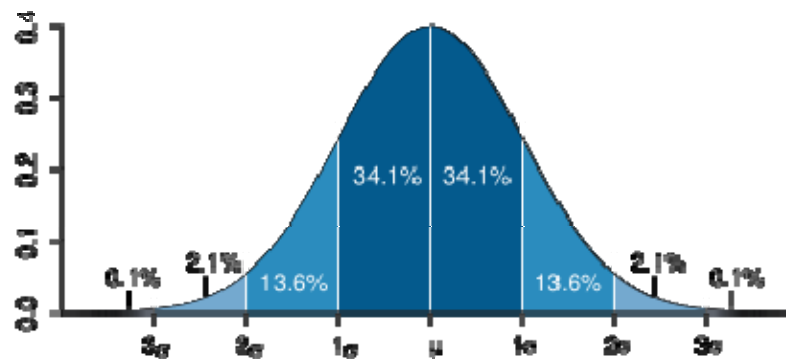
$$\text{posterior} = \frac{\text{likelihood} * \text{prior}}{\text{evidence}}$$

- x is the unknown cause
- y is the observed evidence
- Denominator often omitted (maximum a posteriori solution)
- Bayes rule shows how probability of x changes after we have observed y

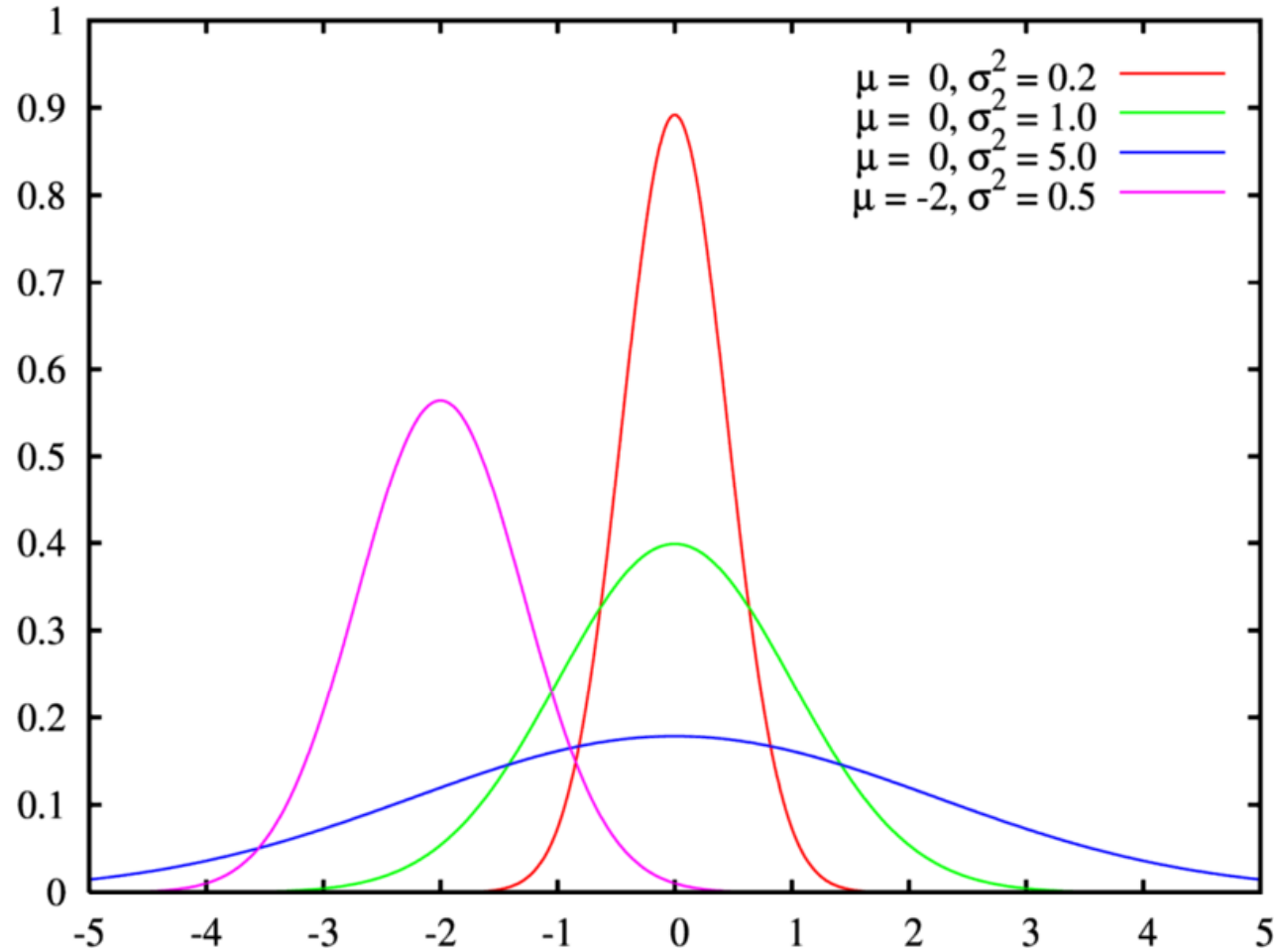
Normal (Gaussian) Distribution

- Central Limit Theorem: under various conditions, the distribution of the sum of d independent random variables approaches a limiting form known as **the normal distribution**

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} = N(\mu, \sigma^2)$$



Normal (Gaussian) Distribution



Outline

- Probability theory
- **Bayes decision theory**
- Maximum-likelihood and Bayesian parameter estimation
- Expectation maximization
- Non-parametric techniques
- Hidden Markov models
- Principal component analysis

Bayes Decision Theory

- Probability distributions for the categories are known
- Do not need training data
- Can design optimal classifier
- **Very rare in real life**

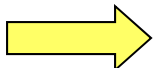
Decision Rules

- Decision rule with only the prior information
 - Decide ω_1 if $P(\omega_1) > P(\omega_2)$ otherwise decide ω_2
 - Prior comes from prior knowledge, no data have been seen yet
 - If there is a reliable source prior knowledge, it should be used
- Use of the class-conditional information
- $p(x | \omega_1)$ and $p(x | \omega_2)$ describe the difference in lightness between populations

Decision using Posteriors

- Decision given the posterior probabilities

X is an observation for which:

if $P(\omega_1 | x) > P(\omega_2 | x)$  True state of nature = ω_1

if $P(\omega_1 | x) < P(\omega_2 | x)$  True state of nature = ω_2

Therefore:

whenever we observe a particular x , the probability of error is :

$P(\text{error} | x) = P(\omega_1 | x)$ if we decide ω_2

$P(\text{error} | x) = P(\omega_2 | x)$ if we decide ω_1

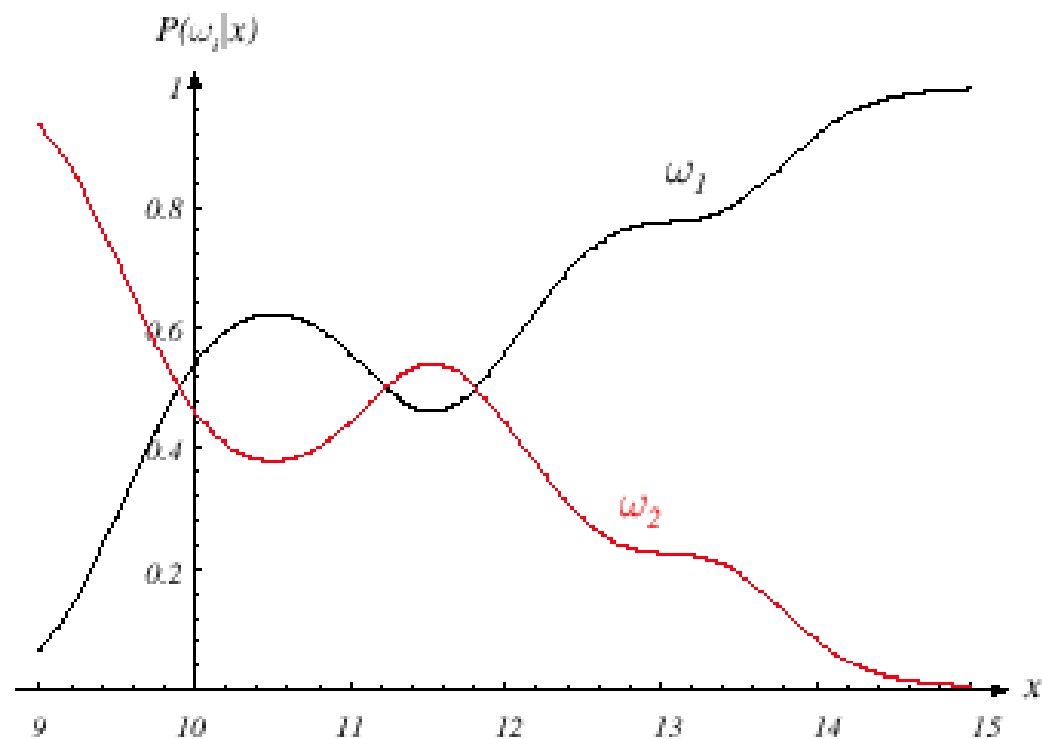


FIGURE 2.2. Posterior probabilities for the particular priors $P(\omega_1) = 2/3$ and $P(\omega_2) = 1/3$ for the class-conditional probability densities shown in Fig. 2.1. Thus in this case, given that a pattern is measured to have feature value $x = 14$, the probability it is in category ω_2 is roughly 0.08, and that it is in ω_1 is 0.92. At every x , the posteriors sum to 1.0. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Minimizing Risk

- Let $\{\omega_1, \omega_2, \dots, \omega_c\}$ be the set of c states of nature (or “categories”)
- Let $\{\alpha_1, \alpha_2, \dots, \alpha_a\}$ be the set of possible actions
- Let $\lambda(\alpha_i | \omega_j)$ be the loss incurred for taking action α_i when the state of nature is ω_j

Overall Risk

R is the expected loss associated with a given decision rule

Minimizing $R \iff$ Minimizing $R(\alpha_i | x)$ for $i = 1, \dots, a$
(select action α that minimizes risk as a function of x)

$$R(\alpha_i | x) = \sum_{j=1}^{j=c} \lambda(\alpha_i | \omega_j) P(\omega_j | x)$$

for $i = 1, \dots, a$

Select the action α_i for which $R(\alpha_i | x)$ is minimum

R is minimum and R in this case is called the **Bayes risk** = best performance that can be achieved

Conditional Risk

- Two-category classification

α_1 : decide ω_1

α_2 : decide ω_2

$$\lambda_{ij} = \lambda(\alpha_i / \omega_j)$$

loss incurred for deciding ω_i when the true state of nature is ω_j

Conditional risk:

$$R(\alpha_1 | \mathbf{x}) = \lambda_{11}P(\omega_1 | \mathbf{x}) + \lambda_{12}P(\omega_2 | \mathbf{x})$$

$$R(\alpha_2 | \mathbf{x}) = \lambda_{21}P(\omega_1 | \mathbf{x}) + \lambda_{22}P(\omega_2 | \mathbf{x})$$

Decision Rule

Our rule is the following:

if $R(\alpha_1 | \mathbf{x}) < R(\alpha_2 | \mathbf{x})$

action α_1 : decide ω_1

This results in the equivalent rule:

decide ω_1 if:

$(\lambda_{21} - \lambda_{11}) P(\mathbf{x} | \omega_1) P(\omega_1) > (\lambda_{12} - \lambda_{22}) P(\mathbf{x} | \omega_2) P(\omega_2)$

and decide ω_2 otherwise

Likelihood ratio

The preceding rule is equivalent to the following rule:

$$\textit{if } \frac{P(\mathbf{x} / \omega_1)}{P(\mathbf{x} / \omega_2)} > \frac{\lambda_{12} - \lambda_{22}}{\lambda_{21} - \lambda_{11}} \cdot \frac{P(\omega_2)}{P(\omega_1)}$$

Then take action α_1 (decide ω_1)

Otherwise take action α_2 (decide ω_2)

The Zero-one Loss Function

- Zero-one loss function:

$$\lambda(\alpha_i, \omega_j) = \begin{cases} 0 & i = j \\ 1 & i \neq j \end{cases} \quad i, j = 1, \dots, c$$

Therefore, the conditional risk is:

$$\begin{aligned} R(\alpha_i | x) &= \sum_{j=1}^{j=c} \lambda(\alpha_i | \omega_j) P(\omega_j | x) \\ &= \sum_{j \neq i} P(\omega_j | x) = 1 - P(\omega_i | x) \end{aligned}$$

- The risk corresponding to this loss function is the average probability of error

Classifiers, Discriminant Functions and Decision Surfaces

- The multi-category case
 - Set of discriminant functions $g_i(x)$, $i = 1, \dots, c$
 - The classifier assigns a feature vector x to class ω_i if:

$$g_i(x) > g_j(x) \quad \forall j \neq i$$

Max Discriminant Functions

- Let $g_i(x) = -R(\alpha_i | x)$
(max. discriminant corresponds to min. risk)

- For the minimum error rate, we take

$$g_i(x) = P(\omega_i | x)$$

(max. discriminant corresponds to max. posterior)

$$g_i(x) \equiv P(x | \omega_i) P(\omega_i)$$

$$g_i(x) = \ln P(x | \omega_i) + \ln P(\omega_i)$$

(ln: natural logarithm)

Decision Regions

- Feature space divided into c decision regions

if $g_i(x) > g_j(x) \forall j \neq i$ then x is in \mathcal{R}_i

(\mathcal{R}_i means assign x to ω_i)

- The two-category case

– A classifier is a “dichotomizer” that has two discriminant functions g_1 and g_2

Let $g(x) \equiv g_1(x) - g_2(x)$

Decide ω_1 if $g(x) > 0$; Otherwise decide ω_2

Discriminant Functions for the Normal Density

- We saw that the minimum error-rate classification can be achieved by the discriminant function

$$g_i(x) = \ln P(x | \omega_i) + \ln P(\omega_i)$$

- Case of multivariate normal

$$g_i(x) = -\frac{1}{2}(x - \mu_i)^t \sum_i^{-1}(x - \mu_i) - \frac{d}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma_i| + \ln P(\omega_i)$$

- **Case** $\Sigma_i = \sigma^2 \mathbf{I}$ (\mathbf{I} stands for the identity matrix)

$$g_i(x) = w_i^t x + w_{i0} \text{ (linear discriminant function)}$$

where :

$$w_i = \frac{\mu_i}{\sigma^2}; \quad w_{i0} = -\frac{1}{2\sigma^2} \mu_i^t \mu_i + \ln P(\omega_i)$$

(w_{i0} is called the threshold for the i th category)

– The hyperplane separating \mathcal{R}_i and \mathcal{R}_j

$$g_i(x) = w_i^t x + w_{i0} \quad \text{and} \quad g_j(x) = w_j^t x + w_{j0}$$

Decision boundary : $g_i(x) = g_j(x)$

$$w^t (x - x_0) = 0$$

$$w = \mu_i - \mu_j$$

$$x_0 = \frac{1}{2}(\mu_i + \mu_j) - \frac{\sigma^2}{\|\mu_i - \mu_j\|^2} \ln \frac{P(\omega_i)}{P(\omega_j)} (\mu_i - \mu_j)$$

always orthogonal to the line linking the means

if $P(\omega_i) = P(\omega_j)$ then $x_0 = \frac{1}{2}(\mu_i + \mu_j)$

- **Case $\Sigma_i = \Sigma$** (covariances of all classes are identical but arbitrary!)

– Hyperplane separating \mathcal{R}_i and \mathcal{R}_j

$$w_i = \Sigma^{-1} \mu_i$$

$$x_0 = \frac{1}{2}(\mu_i + \mu_j) - \frac{\ln[P(\omega_i) / P(\omega_j)]}{(\mu_i - \mu_j)^t \Sigma^{-1} (\mu_i - \mu_j)} \cdot (\mu_i - \mu_j)$$

(the hyperplane separating \mathcal{R}_i and \mathcal{R}_j is generally not orthogonal to the line between the means)

- Case $\Sigma_i = \text{arbitrary}$

- The covariance matrices are different for each category

$$g_i(\mathbf{x}) = \mathbf{x}^t \mathbf{W}_i \mathbf{x} + \mathbf{w}_i^t \mathbf{x} = w_{i0}$$

where :

$$\mathbf{W}_i = -\frac{1}{2} \Sigma_i^{-1}$$

$$\mathbf{w}_i = \Sigma_i^{-1} \boldsymbol{\mu}_i$$

$$w_{i0} = -\frac{1}{2} \boldsymbol{\mu}_i^t \Sigma_i^{-1} \boldsymbol{\mu}_i - \frac{1}{2} \ln |\Sigma_i| + \ln P(\omega_i)$$

(**Hyperquadrics** which are: hyperplanes, pairs of hyperplanes, hyperspheres, hyperellipsoids, hyperparaboloids, hyperhyperboloids)

Outline

- Probability theory
- Bayes decision theory
- **Maximum-likelihood and Bayesian parameter estimation**
- Expectation maximization
- Non-parametric techniques
- Hidden Markov models
- Principal component analysis

Introduction

- Data availability in a Bayesian framework
 - We could design an optimal classifier if we knew:
 - $p(\omega_i)$ (priors)
 - $p(x | \omega_i)$ (class-conditional densities)

Unfortunately, we rarely have this complete information!
- Design a classifier from a training sample
 - No problem with prior estimation
 - Samples are often too small for class-conditional estimation (large dimension of feature space)

Parameter Estimation

- Use a priori information about the problem
- E.g.: Normality of $p(x | \omega_i)$

$$p(x | \omega_i) \sim N(\mu_i, \Sigma_i)$$

- Simplify problem
 - From estimating unknown distribution function
 - To estimating parameters

Parameter Estimation

- Parameters in ML estimation are fixed but unknown!
- Best parameters are obtained by maximizing the probability of obtaining the samples observed
- Bayesian methods view the parameters as random variables having some known distribution
- In either approach, we use $p(\omega_i | \mathbf{x})$ for our classification rule

Independence Across Classes

- For each class ω_i we have a proposed density $p_i(x | \omega_i)$ with unknown parameters θ_i which we need to estimate
- Since we assumed independence of data across the classes, estimation is an identical procedure for all classes
- To simplify notation, we drop sub-indexes and say that we need to estimate parameters θ for density $p(x)$

MLE

- Use the information provided by the training samples to estimate $\theta = (\theta_1, \theta_2, \dots, \theta_c)$
 - each θ_i ($i = 1, 2, \dots, c$) is associated with each category
- Suppose that D contains n samples, x_1, x_2, \dots, x_n

$$p(D | \theta) = \prod_{k=1}^{k=n} p(x_k | \theta)$$

- $p(D | \theta)$ is called the likelihood of θ w.r.t the set of samples
- ML estimate of θ is, by definition the value $\hat{\theta}$ that maximizes $p(D | \theta)$

“It is the value of θ that best agrees with the actually observed training sample”

Optimal estimation

- Let $\theta = (\theta_1, \theta_2, \dots, \theta_p)^t$ and let ∇_{θ} be the gradient operator

$$\nabla_{\theta} = \left[\frac{\partial}{\partial \theta_1}, \frac{\partial}{\partial \theta_2}, \dots, \frac{\partial}{\partial \theta_p} \right]^t$$

- We define $l(\theta)$ as the log-likelihood function

$$l(\theta) = \ln p(D | \theta)$$

- New problem statement: $\hat{\theta} = \arg \max l(\theta)$
determine θ that maximizes the log-likelihood

Necessary Condition for an Optimum

$$\nabla_{\theta} l = \sum_{k=1}^{k=n} \nabla_{\theta} \ln p(x_k | \theta)$$

$$\nabla_{\theta} l = 0$$

- Example of ML estimation: unknown μ and σ (univariate)

$$\theta = (\theta_1, \theta_2) = (\mu, \sigma^2)$$

$$l = \ln p(x_k | \theta) = -\frac{1}{2} \ln 2\pi\theta_2 - \frac{1}{2\theta_2} (x_k - \theta_1)^2$$

$$\nabla_{\theta} l = \begin{pmatrix} \frac{\partial}{\partial \theta_1} (\ln p(x_k | \theta)) \\ \frac{\partial}{\partial \theta_2} (\ln p(x_k | \theta)) \end{pmatrix} = 0$$

$$\begin{cases} \frac{1}{\theta_2} (x_k - \theta_1) = 0 \\ -\frac{1}{2\theta_2} + \frac{(x_k - \theta_1)^2}{2\theta_2^2} = 0 \end{cases}$$

Bayesian Estimation

- Recall that for the MAP classifier we find the class ω_i that maximizes the posterior $p(\omega|D)$
- By analogy, a reasonable estimate of θ is the one that maximizes the posterior $p(\theta|D)$
- But θ is not our final goal, our final goal is the unknown $p(x)$
- Therefore a better thing to do is to maximize $p(x|D)$, this is as close as we can come to the unknown $p(x)$!

Estimation of $p(x|D)$

- From the definition of joint distribution:

$$p(x | D) = \int p(x, \theta | D) d\theta$$

- Using the definition of conditional probability:

$$p(x | D) = \int p(x | \theta, D) p(\theta | D) d\theta$$

- But $p(x|\theta, D)=p(x|\theta)$ since $p(x|\theta)$ is completely specified by θ

$$p(x | D) = \int \overset{\textit{known}}{p(x | \theta)} \overset{\textit{unknown}}{p(\theta | D)} d\theta$$

Estimation of $p(x|D)$

- Using Bayes formula:

$$p(\theta | D) = \frac{p(D | \theta)p(\theta)}{\int p(D | \theta)p(\theta)d\theta} \quad p(D | \theta) = \prod_{k=1}^n p(x_k | \theta)$$

- $p(x|D)$ can be computed

$$p(x | D) = \int p(x | \theta) \frac{\prod_{k=1}^n p(x_k | \theta)p(\theta)}{\int \prod_{k=1}^n p(x_k | \theta)p(\theta)d\theta} d\theta$$

Bayesian Parameter Estimation: Gaussian Case

Goal: Estimate θ using the a-posteriori density $P(\theta | D)$

- The univariate case: $p(\mu | D)$
 μ is the only unknown parameter

$$p(x | \mu) \sim N(\mu, \sigma^2)$$

$$p(\mu) \sim N(\mu_0, \sigma_0^2)$$

Exact form of distribution is not important.
Having a known form is important

μ_0 and σ_0 are known

μ_0 is best guess for μ , σ_0 is uncertainty of guess

Bayesian Parameter Estimation: Gaussian Case

$$\mu_n = \left(\frac{n\sigma_0^2}{n\sigma_0^2 + \sigma^2} \right) \hat{\mu}_n + \frac{\sigma^2}{n\sigma_0^2 + \sigma^2} \mu_0$$

$$\text{and } \sigma_n^2 = \frac{\sigma_0^2 \sigma^2}{n\sigma_0^2 + \sigma^2}$$

- μ is linear combination of empirical and prior information
- σ decreases as more data becomes available

Outline

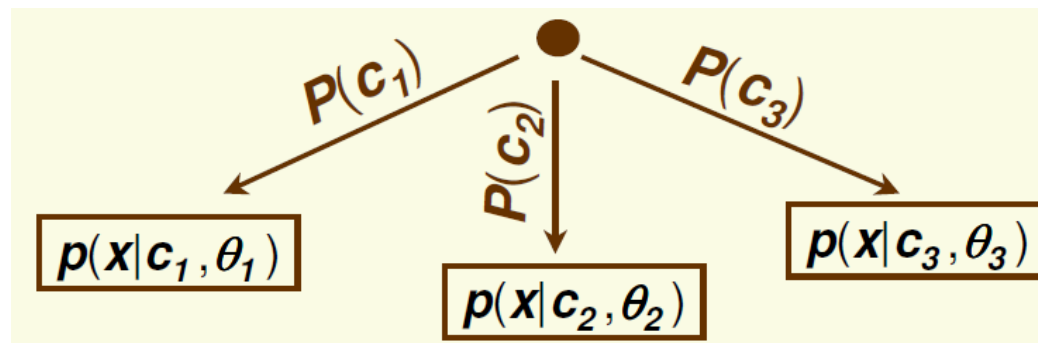
- Probability theory
- Bayes decision theory
- Maximum-likelihood and Bayesian parameter estimation
- **Expectation maximization**
- Non-parametric techniques
- Hidden Markov models
- Principal component analysis

Mixture Density Model

- Model data with mixture density

$$p(x|\theta) = \sum_{j=1}^m \underbrace{p(x|c_j, \theta_j)}_{\text{component densities}} \underbrace{P(c_j)}_{\text{mixing parameters}}$$

- where $\theta = \{\theta_1, \dots, \theta_m\}$
- $P(c_1) + P(c_2) + \dots + P(c_m) = 1$
- To generate a sample from distribution $p(x|\theta)$:
 - first select class j with probability $P(c_j)$
 - then generate x according to probability law $p(x|c_j, \theta_j)$



Mixture Density

- Before EM, let's look at the mixture density again

$$p(x | \theta, \rho) = \sum_{j=1}^m p(x | c_j, \theta_j) \rho_j$$

- Suppose we know how to estimate $\theta_1, \dots, \theta_m$ and ρ_1, \dots, ρ_m
- Estimating the class of x is easy with MAP, maximize:

$$p(x | c_i, \theta_i) P(c_i) = p(x | c_i, \theta_i) \rho_i$$

- Suppose we know the class of samples x_1, \dots, x_n
 - This is just the supervised learning case, so estimating $\theta_1, \dots, \theta_m$ and ρ_1, \dots, ρ_m is easy

$$\hat{\theta}_i = \operatorname{argmax}_{\theta_i} [\ln p(D_i | \theta_i)] \quad \hat{\rho}_i = \frac{|D_i|}{n}$$

- This is an example of chicken-and-egg problem
 - The EM algorithm approaches this problem by adding “hidden” variables

EM: Hidden Variables for Mixture Density

- For i in $[1, n]$, k in $[1, m]$, define hidden variables $z_i^{(k)}$

$$z_i^{(k)} = \begin{cases} 1 & \text{if sample } i \text{ was generated by component } k \\ 0 & \text{otherwise} \end{cases}$$

$$x_i \rightarrow \{x_i, z_i^{(1)}, \dots, z_i^{(m)}\}$$

- $z_i^{(k)}$ are indicator random variables, they indicate which Gaussian component generated sample x_i

EM: Hidden Variables for Mixture Density

- Let $z_i = \{z_i^{(1)}, \dots, z_i^{(m)}\}$, be indicator r.v. corresponding to sample x_i
- Conditioned on z_i , the distribution of x_i is Gaussian

$$p(x_i | z_i, \theta) \sim N(\mu_k, \sigma^2)$$

- where k is s.t. $z_i^{(k)} = 1$

The EM Algorithm

- start with initial parameters $\theta^{(0)}$
- iterate the following 2 steps until convergence
 - E. compute the expectation of the log likelihood with respect to current estimate $\theta^{(t)}$ and X
 - Let's call it $Q(\theta | \theta^{(t)})$

$$Q(\theta | \theta^{(t)}) = E_Z[\ln p(X, Z | \theta) | X, \theta^{(t)}]$$

M. maximize $Q(\theta | \theta^{(t)})$

$$\theta^{(t+1)} = \underset{\theta}{\operatorname{argmax}} Q(\theta | \theta^{(t)})$$

The EM Algorithm

- For the general case of multivariate Gaussians with unknown means and variances

- E step

$$\mathbf{E}_Z[\mathbf{z}_i^{(k)}] = \frac{\rho_k p(\mathbf{x} | \mu_k, \Sigma_k)}{\sum_{j=1}^m \rho_j p(\mathbf{x} | \mu_j, \Sigma_j)}$$

$$p(\mathbf{x} | \mu_k, \Sigma_k) = \frac{1}{(2\pi)^{d/2} |\Sigma_k^{-1}|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \mu_k)^t \Sigma_k^{-1}(\mathbf{x} - \mu_k)\right]$$

- M step

$$\rho_k = \frac{1}{n} \sum_{i=1}^n \mathbf{E}_Z[\mathbf{z}_i^{(k)}]$$

$$\mu_k = \frac{\sum_{i=1}^n \mathbf{E}_Z[\mathbf{z}_i^{(k)}] \mathbf{x}_i}{\sum_{i=1}^n \mathbf{E}_Z[\mathbf{z}_i^{(k)}]}$$

$$\Sigma_k = \frac{\sum_{i=1}^n \mathbf{E}_Z[\mathbf{z}_i^{(k)}] (\mathbf{x}_i - \mu_k)(\mathbf{x}_i - \mu_k)^T}{\sum_{i=1}^n \mathbf{E}_Z[\mathbf{z}_i^{(k)}]}$$

Outline

- Probability theory
- Bayes decision theory
- Maximum-likelihood and Bayesian parameter estimation
- Expectation maximization
- **Non-parametric techniques**
- Hidden Markov models
- Principal component analysis

Introduction

- All parametric densities are unimodal (have a single local maximum), whereas many practical problems involve multi-modal densities
- Non-parametric procedures can be used with arbitrary distributions and without the assumption that the forms of the underlying densities are known
- There are two types of non-parametric methods:
 - Estimate $P(x | \omega_j)$
 - Bypass density function and go directly to posterior probability estimation

Density Estimation

- Probability that a vector x will fall in region R is:

$$P = \int_{\mathcal{R}} p(x') dx' \quad (1)$$

- P is a smoothed (or averaged) version of the density function $p(x)$ if we have a sample of size n ; therefore, the probability that k points fall in R is then:

$$P_k = \binom{n}{k} P^k (1-P)^{n-k} \quad (2)$$

and the expected value for k is:

$$E(k) = nP \quad (3)$$

ML Estimate

ML estimation of $P = \theta$

$\text{Max}_{\theta}(P_k / \theta)$ is reached for $\hat{\theta} = \frac{k}{n} \cong P$

Therefore, the ratio k/n is a good estimate for the probability P and hence for the density function $p(x)$ (for large n)

Assumptions

$p(x)$ is continuous and the region \mathcal{R} is so small that p does not vary significantly within it, we can write:

$$\int_{\mathcal{R}} p(x') dx' \cong p(x) V \quad (4)$$

where x is a point within \mathcal{R} and V the volume enclosed by \mathcal{R} .

Combining equation (1) , (3) and (4) yields: $p(x) \cong \frac{k / n}{V}$

- The volume V needs to approach 0, if we want to use this estimate
 - Practically, V cannot be allowed to become small since the number of samples is always limited
 - One will have to accept a certain amount of variance in the ratio k/n
 - Theoretically, if an unlimited number of samples is available, we can circumvent this difficulty

To estimate the density of x , we form a sequence of regions

R_1, R_2, \dots containing x : the first region contains one sample, the second two samples and so on.

Let V_n be the volume of R_n , k_n the number of samples falling in R_n and $p_n(x)$ be the n^{th} estimate for $p(x)$:

$$p_n(x) = (k_n/n)/V_n \quad (7)$$

Three necessary conditions should apply if we want $p_n(x)$ to converge to $p(x)$:

$$1) \lim_{n \rightarrow \infty} V_n = 0$$

$$2) \lim_{n \rightarrow \infty} k_n = \infty$$

$$3) \lim_{n \rightarrow \infty} k_n / n = 0$$

There are two different ways of obtaining sequences of regions that satisfy these conditions:

(a) Shrink an initial region where $V_n = 1/\sqrt{n}$ and show that

$$p_n(x) \xrightarrow{n \rightarrow \infty} p(x)$$

This is called “the Parzen-window estimation method”

(b) Specify k_n as some function of n , such as $k_n = \sqrt{n}$; the volume V_n is grown until it encloses k_n neighbors of x . This is called “the k_n -nearest neighbor estimation method”

Parzen Windows

- The Parzen-window approach to estimate densities assumes that the region \mathcal{R}_n is a d-dimensional hypercube

$$V_n = h_n^d \text{ (} h_n \text{ : length of the edge of } \mathcal{R}_n \text{)}$$

Let $\varphi(u)$ be the following window function :

$$\varphi(u) = \begin{cases} 1 & |u_j| \leq \frac{1}{2} \quad j = 1, \dots, d \\ 0 & \text{otherwise} \end{cases}$$

- $\varphi((x-x_i)/h_n)$ is equal to unity if x_i falls within the hypercube of volume V_n centered at x and equal to zero otherwise

– The number of samples in this hypercube is:

$$k_n = \sum_{i=1}^{i=n} \varphi\left(\frac{x - x_i}{h_n}\right)$$

By substituting k_n in equation (7), we obtain the following estimate:

$$\mathbf{p}_n(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{i=n} \frac{1}{\mathbf{v}_n} \varphi\left(\frac{\mathbf{x} - \mathbf{x}_i}{\mathbf{h}_n}\right)$$

$P_n(x)$ estimates $p(x)$ as an average of functions of x and the samples $\{x_i\}$ ($i = 1, \dots, n$). These functions φ can be general

Window Functions

- Conditions for estimating legitimate density function
 - Non-negative $\varphi(x) \geq 0$
 - Integrate to 1

$$\int \varphi(x) dx = 1$$

- In other words, the window function should be a probability density function

Classification

- In classifiers based on Parzen-window estimation:
 - We estimate the densities for each category and classify a test point by the label corresponding to the maximum posterior
 - The decision region for a Parzen-window classifier depends upon the choice of window function as illustrated in the following figure

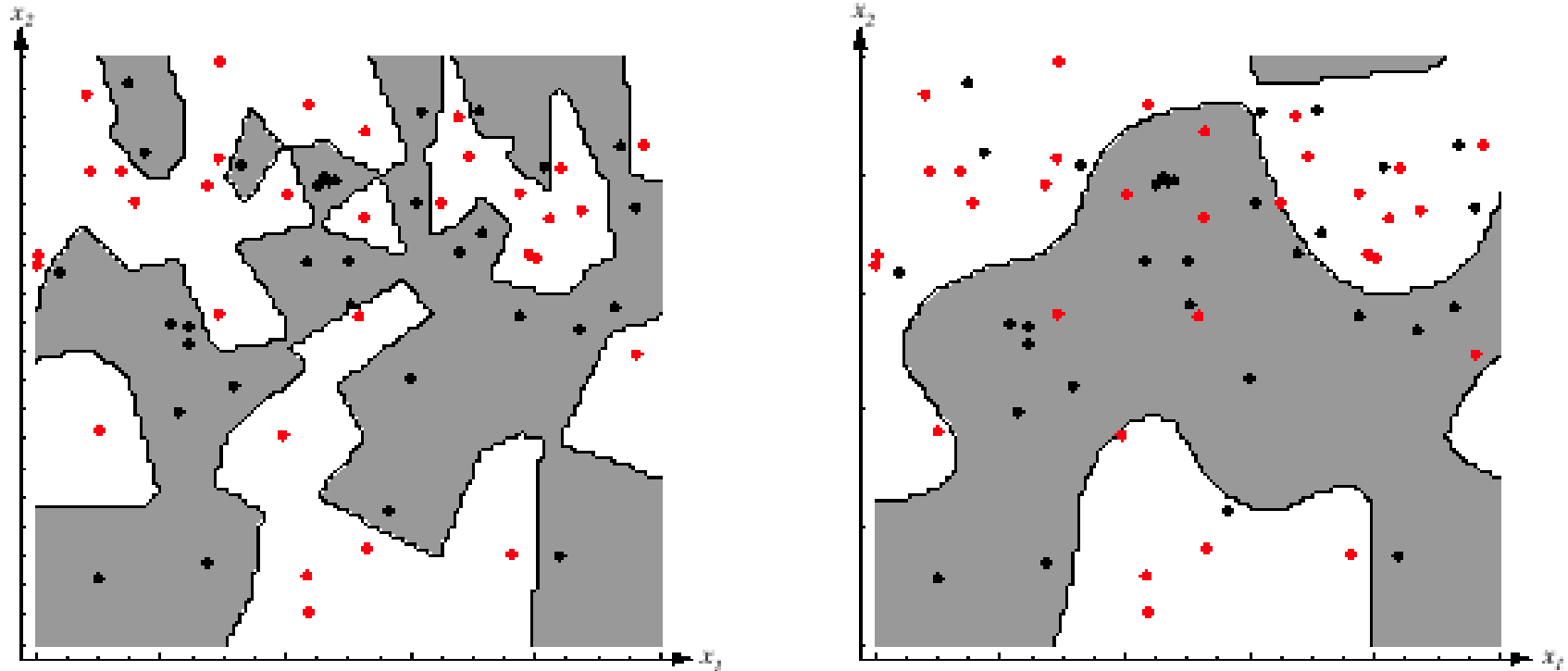


FIGURE 4.8. The decision boundaries in a two-dimensional Parzen-window dichotomizer depend on the window width h . At the left a small h leads to boundaries that are more complicated than for large h on same data set, shown at the right. Apparently, for these data a small h would be appropriate for the upper region, while a large h would be appropriate for the lower region; no single window width is ideal overall. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Remember discussion on overfitting

K - Nearest Neighbor Estimation

- **Goal:** a solution for the problem of the unknown “best” window function
 - Let the cell volume be a function of the training data
 - Center a cell about x and let it grow until it captures k_n samples ($k_n = f(n)$)
 - k_n are called the k_n nearest-neighbors of x
- **Benefits**
 - If density is high near x , the cell will be small which provides a good resolution
 - If density is low, the cell will grow large and stop when higher density regions are reached

We can obtain a family of estimates by setting $k_n = k_1/\sqrt{n}$ and choosing different values for k_1

Estimation of Posterior Probabilities

- **Goal:** estimate $P(\omega_i / \mathbf{x})$ from a set of n labeled samples
- Place a cell of volume V around \mathbf{x} and capture k samples
- k_i samples amongst k turned out to be labeled ω_i then:

$$p_n(\mathbf{x}, \omega_i) = k_i / nV$$

An estimate for $p_n(\omega_i / \mathbf{x})$ is:

$$p_n(\omega_i / \mathbf{x}) = \frac{p_n(\mathbf{x}, \omega_i)}{\sum_{j=1}^{j=c} p_n(\mathbf{x}, \omega_j)} = \frac{k_i}{k}$$

- k_i/k is the fraction of the samples within the cell that are labeled ω_j
 - For minimum error rate, the most frequently represented category within the cell is selected
- => This is equivalent to posterior estimation
- If k is large and the cell sufficiently small, the performance will approach the best possible

The Nearest-Neighbor Rule

- Let $D_n = \{x_1, x_2, \dots, x_n\}$ be a set of n labeled prototypes
- Let $x' \in D_n$ be the closest prototype to a test point x then the nearest-neighbor rule for classifying x is to assign it the label associated with x'
- The nearest-neighbor rule leads to an error rate greater than the minimum possible: the Bayes rate
- If the number of prototype is large (unlimited), the error rate of the nearest-neighbor classifier is never worse than twice the Bayes rate (it can be proven!)
- If $n \rightarrow \infty$, it is always possible to find x' sufficiently close so that:
$$P(\omega_j | x') \sim P(\omega_j | x)$$

Outline

- Probability theory
- Bayes decision theory
- Maximum-likelihood and Bayesian parameter estimation
- Expectation maximization
- Non-parametric techniques
- **Hidden Markov models**
- Principal component analysis

Definition

Doubly stochastic process with an underlying stochastic process that is not observable (hidden), but can only be observed through another set of stochastic processes that produce the sequence of observed symbols.

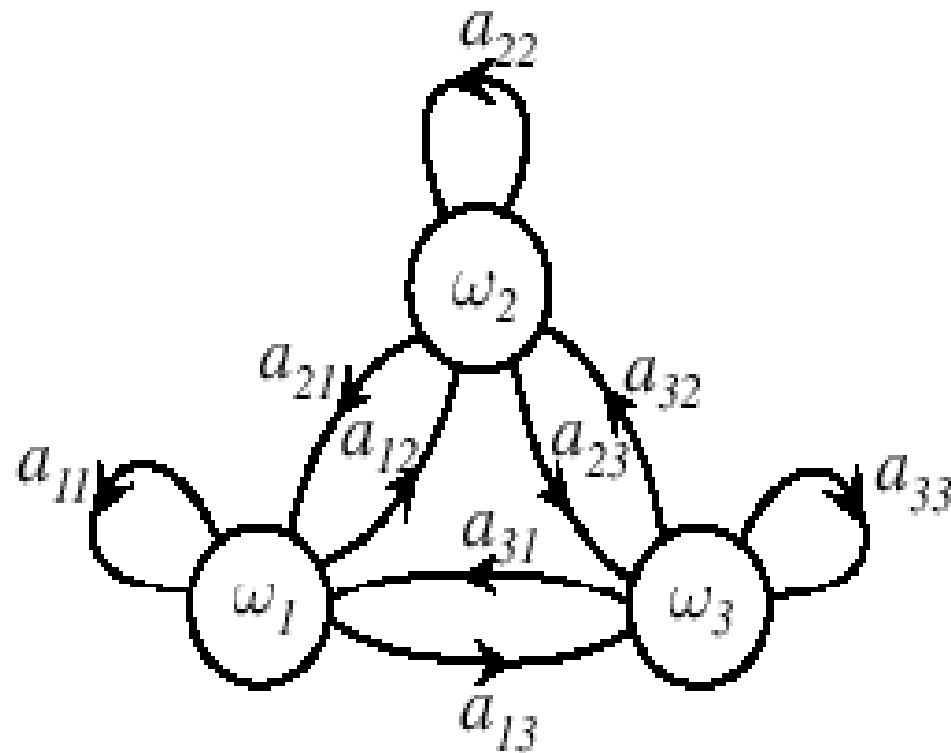


FIGURE 3.8. The discrete states, ω_i , in a basic Markov model are represented by nodes, and the transition probabilities, a_{ij} , are represented by links. In a first-order discrete-time Markov model, at any step t the full system is in a particular state $\omega(t)$. The state at step $t + 1$ is a random function that depends solely on the state at step t and the transition probabilities. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

The Evaluation Problem

The probability that the model produces a sequence V^T of visible states is:

$$P(V^T) = \sum_{r=1}^{r_{\max}} P(V^T | \omega_r^T) P(\omega_r^T)$$

where each r indexes a particular sequence $\omega_r^T = \{\omega(1), \omega(2), \dots, \omega(T)\}$ of T hidden states.

$$(1) \quad P(V^T | \omega_r^T) = \prod_{t=1}^{t=T} P(v(t) | \omega(t))$$

$$(2) \quad P(\omega_r^T) = \prod_{t=1}^{t=T} P(\omega(t) | \omega(t-1))$$

Using equations (1) and (2), we can write:

$$P(V^T) = \sum_{r=1}^{r_{\max}} \prod_{t=1}^{t=T} P(v(t) | \omega(t)) P(\omega(t) | \omega(t-1))$$

Interpretation: The probability that we observe the particular sequence of T visible states V^T is equal to the sum over all r_{\max} possible sequences of hidden states of the conditional probability that the system has made a particular transition multiplied by the probability that it then emitted the visible symbol in our target sequence.

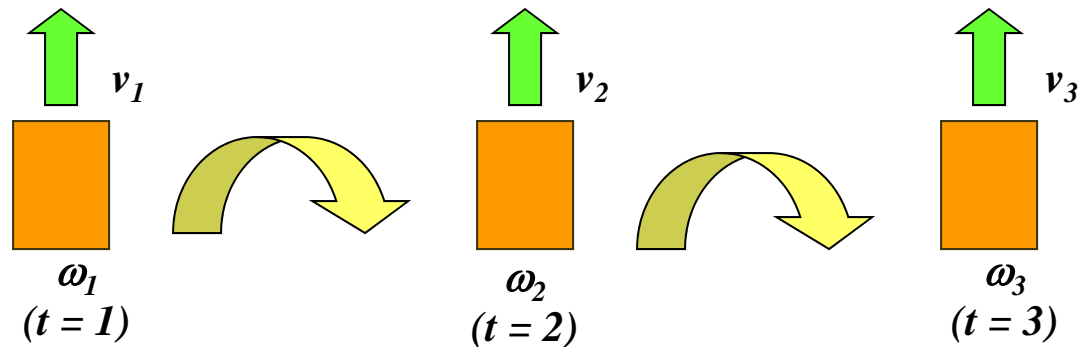
Example: Let $\omega_1, \omega_2, \omega_3$ be the hidden states; v_1, v_2, v_3 be the visible states

and $V^3 = \{v_1, v_2, v_3\}$ is the sequence of visible states

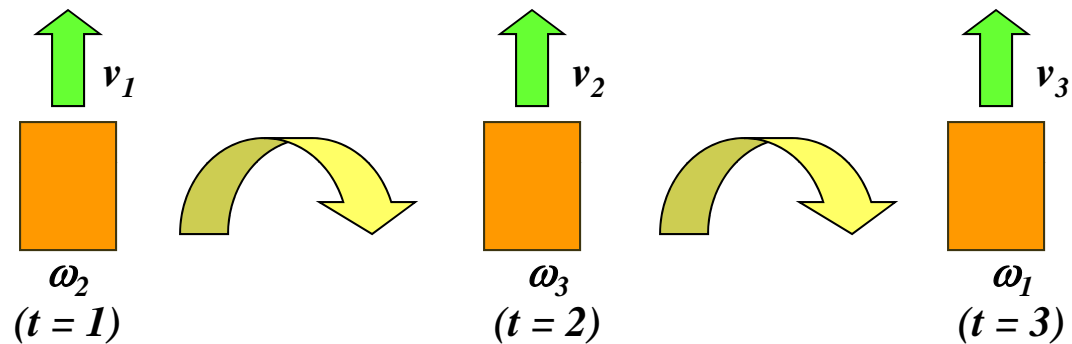
$$P(\{v_1, v_2, v_3\}) = P(\omega_1)P(v_1 | \omega_1)P(\omega_2 | \omega_1)P(v_2 | \omega_2)P(\omega_3 | \omega_2)P(v_3 | \omega_3) \\ + \dots + \text{(possible terms in the sum = all possible } (3^3 = 27) \text{ cases !)}$$

In general $r_{\max} = c^T$, where c is the number of states

First possibility:



Second Possibility:



$$P(\{v_1, v_2, v_3\}) = P(\omega_2)P(v_1 | \omega_2)P(\omega_3 | \omega_2)P(v_2 | \omega_3)P(\omega_1 | \omega_3)P(v_3 | \omega_1) + \dots +$$

Therefore:

$$P(\{v_1, v_2, v_3\}) = \sum_{\substack{\text{possible sequence} \\ \text{of hidden states}}} \prod_{t=1}^{t=3} P(v(t) | \omega(t))P(\omega(t) | \omega(t-1))$$

Algorithm

1. Initialize: $t \leftarrow 0$, a_{ij} , b_{jk} , visible sequence V^T , $a_j(0)$
2. for $t \leftarrow t+1$
3. $a_j(t) \leftarrow b_{jk} v(t) \sum a_i(t-1) a_{jj}$
4. until $t=T$
5. return $P(V^T) \leftarrow a_0(T)$

$a_j(t)$: probability of being in state ω_j at step t ,
having generated first t elements of V^T

$a_0(T)$ is probability of sequence ending at
known final state

Note: Typo in caption of Fig. 3.10 in DHS. See errata.

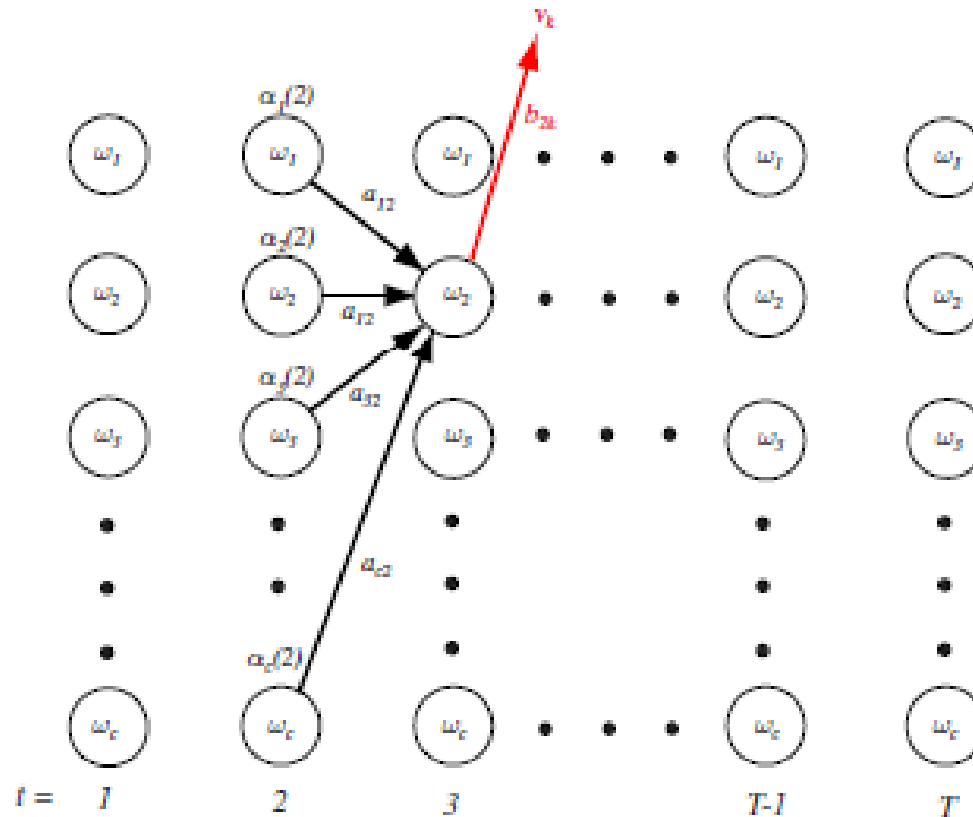


FIGURE 3.10. The computation of probabilities by the Forward algorithm can be visualized by means of a trellis—a sort of “unfolding” of the HMM through time. Suppose we seek the probability that the HMM was in state ω_2 at $t = 3$ and generated the observed visible symbol up through that step (including the observed visible symbol v_k). The probability the HMM was in state $\omega_j(t = 2)$ and generated the observed sequence through $t = 2$ is $\alpha_j(2)$ for $j = 1, 2, \dots, c$. To find $\alpha_2(3)$ we must sum these and multiply the probability that state ω_2 emitted the observed symbol v_k . Formally, for this particular illustration we have $\alpha_2(3) = b_{2k} \sum_{j=1}^c \alpha_j(2) a_{j2}$. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Outline

- Probability theory
- Bayes decision theory
- Maximum-likelihood and Bayesian parameter estimation
- Expectation maximization
- Non-parametric techniques
- Hidden Markov models
- **Principal component analysis**

Goal of PCA

- Compute the most meaningful basis to re-express a noisy data set
- Hope that this new basis will filter out the noise and reveal hidden structure
- In toy example:
 - Determine that the dynamics are along a single axis
 - Determine the important axis

Change of Basis

- X is original data ($m \times n$, $m=6$, $n=7200$)
- Let Y be another $m \times n$ matrix such that $Y=PX$
- P is a matrix that transforms X into Y
 - Geometrically it is a rotation and stretch
 - The rows of P $\{p_1, \dots, p_m\}$ are the new basis vectors for the columns of X
 - Each element of y_i is a dot product of x_i with the corresponding row of P (a projection of x_i onto p_j)

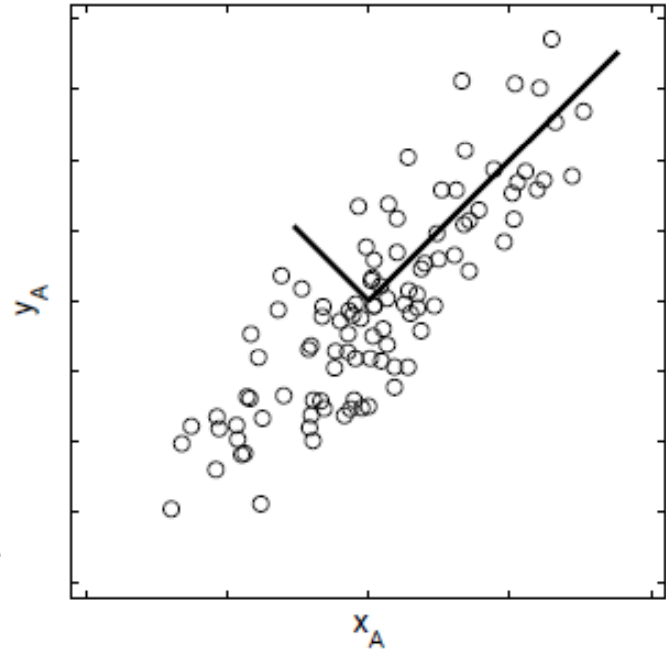
$$PX = \begin{bmatrix} p_1 \\ \vdots \\ p_m \end{bmatrix} \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}$$
$$Y = \begin{bmatrix} p_1 \cdot x_1 & \cdots & p_1 \cdot x_n \\ \vdots & \ddots & \vdots \\ p_m \cdot x_1 & \cdots & p_m \cdot x_n \end{bmatrix}$$
$$y_i = \begin{bmatrix} p_1 \cdot x_i \\ \vdots \\ p_m \cdot x_i \end{bmatrix}$$

How to find an Appropriate Change of Basis?

- The row vectors $\{p_1, \dots, p_m\}$ will become the *principal components* of X
- What is the best way to re-express X ?
- What features would we like Y to exhibit?

- If we call X “garbled data”, garbling in a linear system can refer to three things:
 - Noise
 - Rotation
 - Redundancy

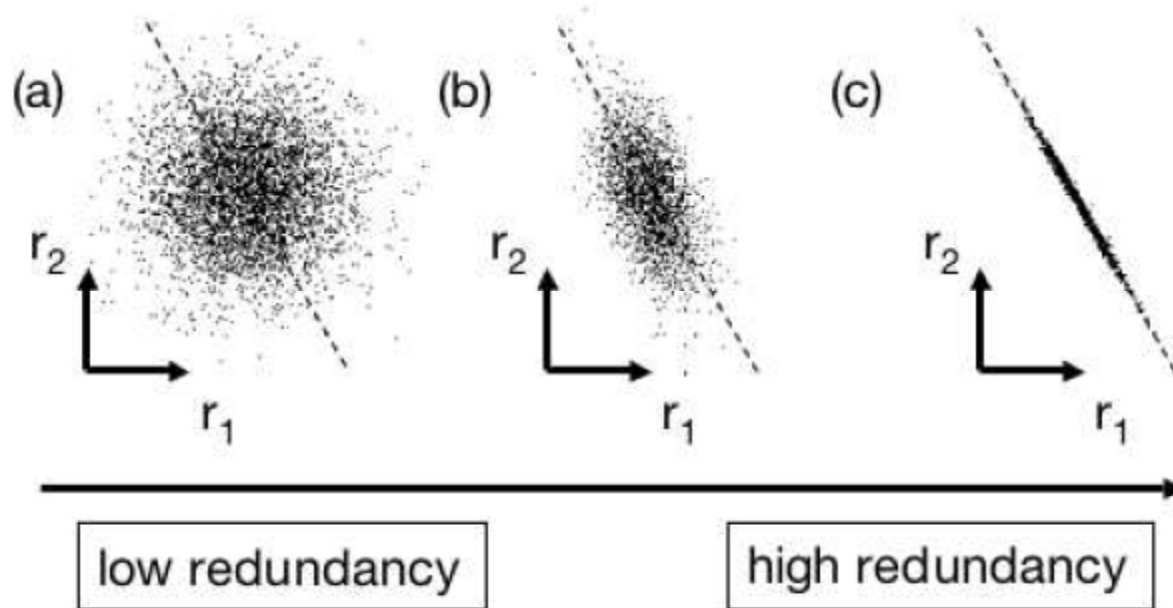
- Ball travels in straight line
 - Any deviation must be noise
- Variance due to signal and noise are indicated in diagram
- SNR: ratio of the two lengths
 - “Fatness” of data corresponds to noise
- Assumption: **directions of largest variance in measurement vector space contain dynamics of interest**



Redundancy

- Is it necessary to record 2 variables for the ball-spring system?
- Is it necessary to use 3 cameras?

Redundancy spectrum for 2 variables



Sketch of Algorithm

- Pick vector in n -D space along which variance is maximal and save as p_1
- Pick another direction along which variance is maximized among directions perpendicular to p_1
- Repeat until k principal components have been selected

Basic PCA Algorithm

- Start from $m \times n$ data matrix X
 - m data points (samples over time)
 - n measurement types
- Re-center: subtract mean from each row of X
- Compute covariance matrix:
 - $\Sigma = X_c^T X_c$
- Compute eigenvectors and eigenvalues of Σ
- Principal components: k eigenvectors with highest eigenvalues

Note: Covariance matrix is $n \times n$ (measurement types)
(But there may be exceptions)

SVD

- Efficiently finds top k eigenvectors
 - Much faster than eigen-decomposition
- Write $X = U S V^T$
 - X : data matrix, one row per datapoint
 - U : weight matrix, one row per datapoint - coordinates of x^i in eigen-space
 - S : singular value matrix, diagonal matrix
 - in our setting each entry is eigenvalue λ_j of Σ
 - V^T : singular vector matrix
 - in our setting each row is eigenvector v_j of Σ