Chapter 3

Maximum-likelihood and Bayesian Parameter Estimation

Introduction

- An optimal classifier can be designed if we know $P(\omega_i)$ and $p(\mathbf{x}|\omega_i)$.
  - Complete knowledge about the probabilistic structure is rarely provided.
    - Vague and general knowledge about the situation
    - Limited number of design samples or training data
  - The problem:
    - To find some way to use this information to design or train the classifier.

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

- An approach:
  - To use the samples to estimate the unknown probabilities/densities, then use the resulting estimates as if they were the true values.
    - Estimating prior probabilities/class-conditional densities
    - The number of available samples always seems too small.
    - The dimensionality of the feature vector $\mathbf{x}$ is large.
    - If we know the number of parameters and our knowledge about the problems, the severity of these problems can be reduced.
      - □ If $p(\mathbf{x}|\omega_i)$ is a normal density, the problem becomes to estimate $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$.

Introduction

- Parameter Estimation
  - Maximum-likelihood estimation
    - The parameters are regarded as quantities whose values *are fixed but unknown*.
    - The best estimate of their value is defined to be the one that maximizes the probability of obtaining the samples actually observed.
  - Bayesian estimation
    - The parameters are regarded as *random variables* having some known prior distribution.
    - Observation of the samples converts this to a posterior density.
    - A typical effect of observing additional samples is to sharpen the a posteriori density function : *Bayesian learning*
  - Supervised/unsupervised learning
    - $P(\omega_i)$ and $p(\mathbf{x}|\omega_i)$

Maximum-likelihood Estimation

- Attractive attributes
  - Good convergence properties as the number of training samples increases
  - Simpler than alternative methods

The general principle

- $D_1, \dots, D_c$: c data sets
  - The samples in D_j have been drawn independently according to the probability law $p(\mathbf{x}|\omega_j)$

- Assume that $p(\mathbf{x}|\omega_j)$ has a known parametric form and determined by a parameter vector $\boldsymbol{\theta}_j$

  $p(\mathbf{x}|\omega_j) \sim N(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$, where $\boldsymbol{\theta}_j$ consists of the components of $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$.

  $$\Rightarrow p(\mathbf{x}|\omega_j, \boldsymbol{\theta}_j) \text{ : to show the dependence of } p(\mathbf{x}|\omega_j) \text{ on } \boldsymbol{\theta}_j.$$

- The problem is to use the information provided by the training samples to obtain good estimates for the unknown parameter vector $\boldsymbol{\theta}_j$ associated with the category.

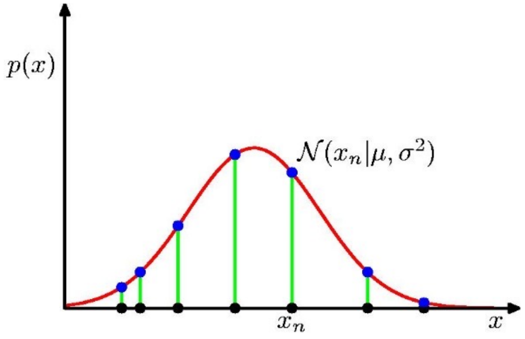
Maximum-likelihood Estimation

The general principle (cont.)

- To simplify the problem, let us assume that samples in D_i give no information about $\boldsymbol{\theta}_j$ if $i \neq j$.
  - the parameters for the different classes are functionally independent.
  - use a set D of training samples drawn independently from the probability density $p(\mathbf{x}|\boldsymbol{\theta})$ to estimate the unknown parameter vector $\boldsymbol{\theta}$.
- Suppose the D contains n samples, $\mathbf{x}_1, \ldots, \mathbf{x}_n$

$$p(D|\boldsymbol{\theta}) = \prod_{k=1}^{n} p(\mathbf{x}_k|\boldsymbol{\theta})$$

likelihood of $\boldsymbol{\theta}$ with respect to the set of samples

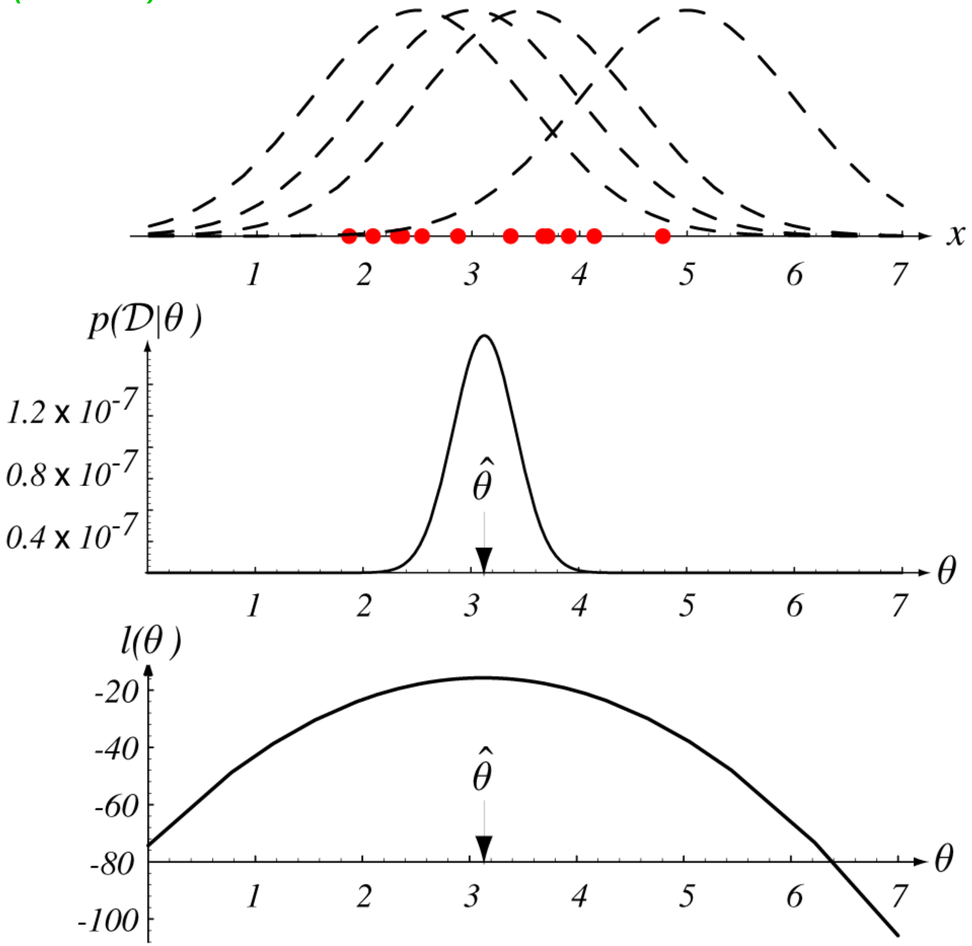


PRML Fig1.14

- $\hat{\boldsymbol{\theta}}$: the maximum-likelihood estimate of $\boldsymbol{\theta}$ is the value that maximizes $p(D|\boldsymbol{\theta})$
  - *The value that best agrees with or supports the actually observed training samples.*

Maximum-likelihood Estimation

The general principle (cont.)



Maximum-likelihood Estimation

The general principle (cont.)

- It is easier to work with logarithm of the likelihood.
- If the number of parameters to be estimated is p

$$\boldsymbol{\theta} = \left(\theta_1, \dots, \theta_p\right)^t$$

The *log-likelihood* function $l(\boldsymbol{\theta}) \equiv \ln p(D|\boldsymbol{\theta})$

$$\widehat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} l(\boldsymbol{\theta})$$

$$l(\boldsymbol{\theta}) = \sum_{k=1}^{n} \ln p(\mathbf{x}_k|\boldsymbol{\theta})$$

$$\nabla_{\boldsymbol{\theta}} l(\boldsymbol{\theta}) = \sum_{k=1}^{n} \nabla_{\boldsymbol{\theta}} \ln p(\mathbf{x}_k|\boldsymbol{\theta})$$

The gradient operator

$$\boxed{\nabla_{\boldsymbol{\theta}} \equiv \begin{bmatrix} \frac{\partial}{\partial\theta_1} \\ \vdots \\ \frac{\partial}{\partial\theta_p} \end{bmatrix}}$$

- A set of necessary conditions for the maximum-likelihood estimate for $\boldsymbol{\theta}$ can be obtained by

$$\nabla_{\boldsymbol{\theta}} l(\boldsymbol{\theta}) = \mathbf{0}$$

- The equation could represent a true global maximum, a local maximum or minimum, or an inflection point of $l(\boldsymbol{\theta})$.

Maximum-likelihood Estimation

The general principle (cont.)

- MAP (maximum a posteriori) estimation
  - A related class of estimators which find the value of θ that maximizes $l(\theta)p(\theta)$ where $p(\theta)$ describes the prior probability of different parameter values.
    - A maximum-likelihood estimations is a MAP estimator for the "flat" prior.
  - A MAP estimation finds the peak.
  - The drawback of MAP estimates is that if we choose some arbitrary nonlinear transform of the parameter space, the density will change, and the MAP solution need no longer be appropriate.

Maximum-likelihood Estimation

The Gaussian case: Unknown μ

- We consider a sample point $\mathbf{x}_k$ and find

$$\ln p(\mathbf{x}_k|\boldsymbol{\mu}) = -\frac{1}{2}\ln[(2\pi)^d|\boldsymbol{\Sigma}|] - \frac{1}{2}(\mathbf{x}_k - \boldsymbol{\mu})^t\Sigma^{-1}(\mathbf{x}_k - \boldsymbol{\mu})$$

and

$$\boldsymbol{\nabla}_{\boldsymbol{\mu}} \ln p(\mathbf{x}_k|\boldsymbol{\mu}) = \Sigma^{-1}(\mathbf{x}_k - \boldsymbol{\mu})$$

Identifying $\boldsymbol{\theta}$ with $\boldsymbol{\mu}$, the maximum-likelihood estimate for $\boldsymbol{\mu}$ must satisfy

$$\sum_{k=1}^{n} \Sigma^{-1}(\mathbf{x}_k - \boldsymbol{\mu}) = \mathbf{0}$$

$$l(\boldsymbol{\theta}) = \textstyle\sum_{k=1}^{n} \ln p(\mathbf{x}_k|\boldsymbol{\theta})$$

$$\nabla_{\boldsymbol{\theta}} l(\boldsymbol{\theta}) = \mathbf{0}$$

$$\Longrightarrow \quad \hat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{k=1}^{n} \mathbf{x}_k$$

The maximum-likelihood estimate for the unknown population mean is just the arithmetic average of the training samples – the *sample mean*.

Maximum-likelihood Estimation

The Gaussian case: Unknown μ and Σ

- The unknown parameters constitute the components of $\boldsymbol{\theta}$.

Univariate case with $\theta_1 = \mu$ and $\theta_2 = \sigma^2$:

The log-likelihood of a single point

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

and

$$\boldsymbol{\nabla_\theta} l = \boldsymbol{\nabla_\theta} l \ln p(x_k|\boldsymbol{\theta}) = \begin{bmatrix} \dfrac{1}{\theta_2}(x_k - \theta_1) \\ -\dfrac{1}{2\theta_2} + \dfrac{(x_k - \theta_1)^2}{2\theta_2^2} \end{bmatrix}$$

⇨ $\sum_{k=1}^{n} \frac{1}{\hat{\theta}_2}\left(x_k - \hat{\theta}_1\right) = 0$ and $-\sum_{k=1}^{n} \frac{1}{\hat{\theta}_2} + \sum_{k=1}^{n} \frac{\left(x_k - \hat{\theta}_1\right)^2}{\hat{\theta}_2^2} = 0$

⇨ $\hat{\mu} = \frac{1}{n}\sum_{k=1}^{n} x_k$ and $\hat{\sigma}^2 = \frac{1}{n}\sum_{k=1}^{n}(x_k - \hat{\mu})^2$

Maximum-likelihood Estimation

The Gaussian case: Unknown μ and Σ (cont.)

Multivariate case :

$$\widehat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{k=1}^{n}\mathbf{x}_k \quad \text{and} \quad \widehat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{k=1}^{n}(\mathbf{x}_k - \widehat{\boldsymbol{\mu}})(\mathbf{x}_k - \widehat{\boldsymbol{\mu}})^t$$

We find that the maximum-likelihood estimate for the mean vector is the sample mean and for the covariance matrix is the arithmetic average of the n matrices$(\mathbf{x}_k - \widehat{\boldsymbol{\mu}})(\mathbf{x}_k - \widehat{\boldsymbol{\mu}})^t$.

Bayesian Estimation

- In MLE:
  - $\boldsymbol{\theta}$ to be fixed
- In Bayesian learning:
  - $\boldsymbol{\theta}$ to be a random variable and *training data* allow us to convert a distribution on this variable into a posterior probability density.

The class-conditional densities

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

- How can we proceed when these quantities are unknown?
  - To compute $P(\omega_j|\mathbf{x})$ using all of the information at our disposal
    - □ Knowledge of the functional forms for unknown densities and ranges for the values of unknown parameters
    - □ Some from a set of training samples.

Bayesian Estimation

The class-conditional densities (cont.)

$$P(\omega_i|\mathbf{x}, D) = \frac{p(\mathbf{x}|\omega_i, D)P(\omega_i|D)}{\sum_{j=1}^{c} p\left(\mathbf{x}\middle|\omega_j, D\right)P(\omega_j|D)}$$

- The information provided by the training samples can be used to help to determine both the class-conditional densities and the prior probabilities.
- We assume that the true values of the prior probabilities are known or obtainable from a trivial calculation $P(\omega_i) = P(\omega_i|D)$.

Bayesian Estimation

The class-conditional densities (cont.)

- Supervised case
  - To separate the training samples by class into c subsets $D_1, \ldots, D_c$, with the samples in D_i belonging to ω_i.
  - The samples in D_i have no influence on $p(\mathbf{x}|\omega_j, D)$ if $i \neq j$.

$$P(\omega_i|\mathbf{x}, D) = \frac{p(\mathbf{x}|\omega_i, D)P(\omega_i|D)}{\sum_{j=1}^{c} p(\mathbf{x}|\omega_j, D)P(\omega_j|D)} \Rightarrow P(\omega_i|\mathbf{x}, D) = \frac{p(\mathbf{x}|\omega_i, D_i)P(\omega_i)}{\sum_{j=1}^{c} p(\mathbf{x}|\omega_j, D_j)P(\omega_j)}$$

  - Use a set D of samples drawn independently according to the fixed but unknown probability distribution $p(\mathbf{x})$ to determine $p(\mathbf{x}|D)$.

Bayesian Estimation

The Parameter Distribution

- Assume that $p(\mathbf{x})$ has a known parametric form. The only they unknown is the value of a parameter vector $\boldsymbol{\theta}$.
- Any information we might have about $\boldsymbol{\theta}$ prior to observing the samples is assumed to be contained in a *known* prior density $p(\boldsymbol{\theta})$. Observation of the samples converts this to a posterior density $p(\boldsymbol{\theta}|D)$, which is sharply peaked about the true value of $\boldsymbol{\theta}$.
- Problem of learning a probability density function
  - ⇨ estimating a parameter vector
  - ⇨ goal is to compute $p(\mathbf{x}|D)$

$$\boxed{\begin{aligned} p(\mathbf{x}|D) &= \int p(\mathbf{x},\boldsymbol{\theta}|D)d\boldsymbol{\theta} \\ &= \int p(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta}|D)d\boldsymbol{\theta} \end{aligned}}$$

$$p(\mathbf{x},\boldsymbol{\theta}|D) = p(\mathbf{x}|\boldsymbol{\theta},D)p(\boldsymbol{\theta}|D)$$

- The equation links the desired class-conditional density $p(\mathbf{x}|D)$ to the posterior density $p(\boldsymbol{\theta}|D)$ for the unknown parameter vector. If $p(\boldsymbol{\theta}|D)$ peaks sharply about some value $\widehat{\boldsymbol{\theta}}$, we obtain $p(\mathbf{x}|D) \cong p(\mathbf{x}|\widehat{\boldsymbol{\theta}})$.

Bayesian Parameter Estimation: Gaussian case

- To calculate a posteriori density $p(\boldsymbol{\theta}|D)$ and $p(\mathbf{x}|D)$ for the case where $p(\mathbf{x}|\boldsymbol{\mu}) \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

The Univariate Case: $p(\mu|D)$

- Consider the case where μ is the only unknown parameter.

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

- Prior knowledge we have about μ can be expressed by a known prior density $p(\mu)$

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

  - μ_0 : our best prior guess for μ.
  - σ_0^2: our uncertainty about the guess.

Bayesian Parameter Estimation: Gaussian case

The Univariate Case: $p(\mu|D)$(cont.)

- Imagine that a value is drawn for μ from a population governed by the probability law $p(\mu)$. Once the value is drawn, it becomes the true value of μ and completely determines the density for x. Suppose that n samples $x_1, \ldots, x_n$ are independently drawn from the population.

$$p(\mu|D) = \frac{p(D|\mu)p(\mu)}{\int p(D|\mu)p(\mu)d\mu}$$

$$= \alpha \prod_{k=1}^{n} p(x_k|\mu)p(\mu)$$

Shows how the observation of a set of training samples affects our ideas about true value of μ.

Bayesian Parameter Estimation: Gaussian case

The Univariate Case: $p(\mu|D)$(cont.)

- Because $p(x_k|\mu) \sim N(\mu, \sigma^2)$ and $p(\mu) \sim N(\mu_0, \sigma_0^2)$

$$p(\mu|D) = \alpha \prod_{k=1}^{n} \overbrace{\frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x_k-\mu}{\sigma}\right)^2\right]}^{p(x_k|\mu)} \overbrace{\frac{1}{\sqrt{2\pi}\sigma_0} \exp\left[-\frac{1}{2}\left(\frac{\mu-\mu_0}{\sigma_0}\right)^2\right]}^{p(\mu)}$$

$$= \alpha' \exp\left[-\frac{1}{2}\left(\sum_{k=1}^{n}\left(\frac{\mu-x_k}{\sigma}\right)^2 + \left(\frac{\mu-\mu_0}{\sigma_0}\right)^2\right)\right]$$

$$= \alpha'' \exp\left[-\frac{1}{2}\left[\left(\frac{n}{\sigma^2}+\frac{1}{\sigma_0^2}\right)\mu^2 - 2\left(\frac{1}{\sigma^2}\sum_{k=1}^{n} x_k + \frac{\mu_0}{\sigma_0^2}\right)\mu\right]\right]$$

An exponential function of a quadratic function of μ - *normal density*

Reproducing density

Bayesian Parameter Estimation: Gaussian case

The Univariate Case: $p(\mu|D)$(cont.)

- If we write $p(\mu|D) \sim N(\mu_n, \sigma_n^2)$,

$$p(\mu|D) = \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left[-\frac{1}{2}\left(\frac{\mu-\mu_n}{\sigma_n}\right)^2\right]$$

Identifying coefficients yields

$$\frac{1}{\sigma_n^2} = \frac{n}{\sigma^2} + \frac{1}{\sigma_0^2} \quad \text{and} \quad \frac{\mu}{\sigma_n^2} = \frac{n}{\sigma^2}\hat{\mu}_n + \frac{\mu_0}{\sigma_0^2} \text{ , where } \hat{\mu}_n = \frac{1}{n}\sum_{k=1}^{n} x_k$$

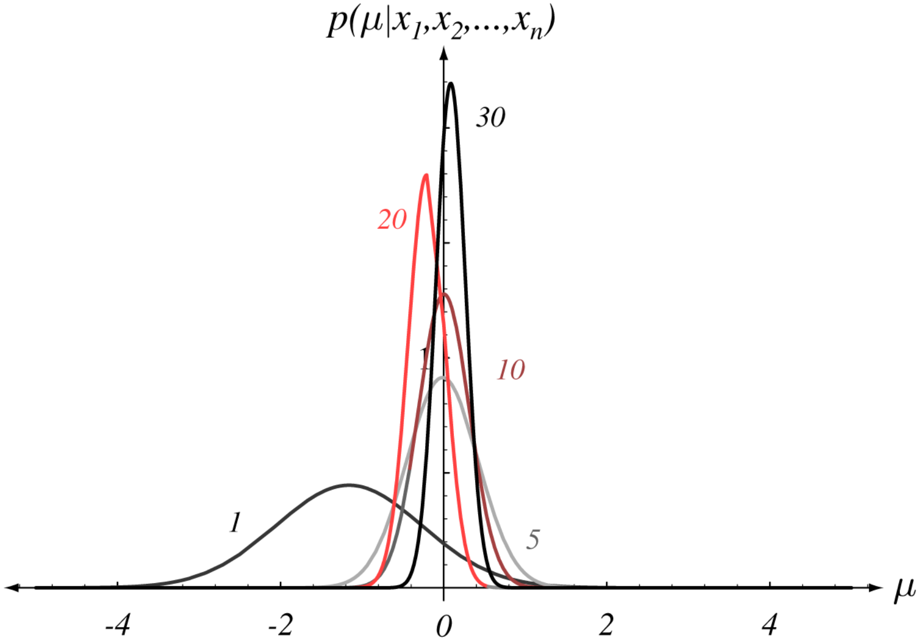
$$\Rightarrow \mu_n = \left(\frac{n\sigma_0^2}{n\sigma_0^2 + \sigma^2}\right)\hat{\mu}_n + \frac{\sigma^2}{n\sigma_0^2 + \sigma^2}\mu_0 \quad \text{and} \quad \sigma_n^2 = \frac{\sigma_0^2 \sigma}{n\sigma_0^2 + \sigma^2}$$

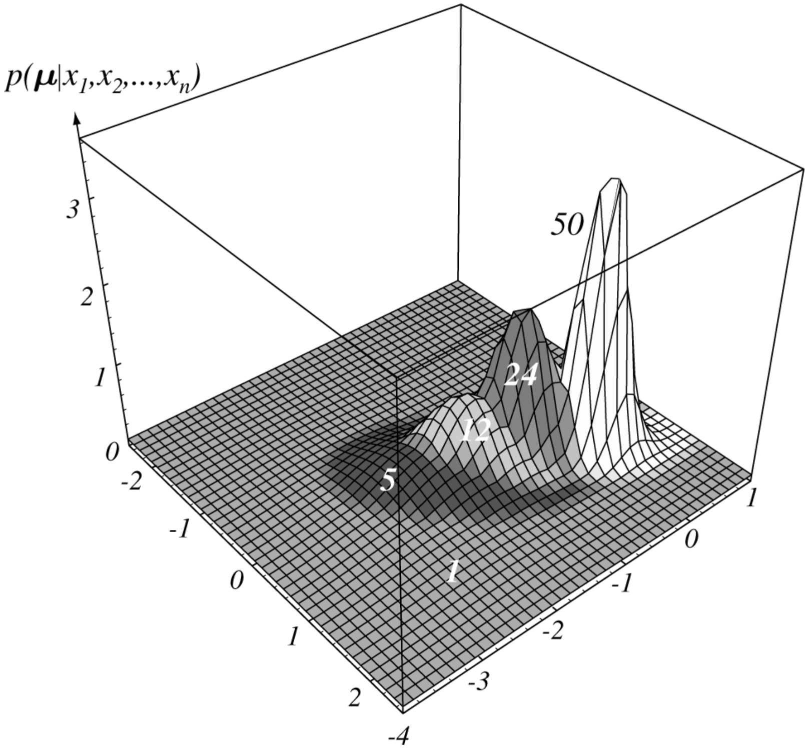
Our best guess for μ after observing n samples

- Our uncertainty about the guess: $\sigma_n^2 \Longrightarrow \sigma/n$ as $n \to \infty$
- Each additional observation decreases the uncertainty about the true value of μ.

Bayesian Parameter Estimation: Gaussian case

The Univariate Case: $p(\mu|D)$(cont.)





The posterior distribution estimates are labeled by the number of training samples used in the estimation.

Bayesian Parameter Estimation: Gaussian case

The Univariate Case: $p(x|D)$

- Having obtained the a posteriori density for the mean, $p(\mu|D)$, all that remains is to obtain the "class-conditional" density $p(x|D)$.

$$p(x|D) = \int p(x|\mu)p(\mu|D)d\mu$$

$$= \int \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right] \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left[-\frac{1}{2}\left(\frac{\mu-\mu_n}{\sigma_n}\right)^2\right] d\mu$$

$$= \frac{1}{2\pi\sigma\sigma_n} \exp\left[-\frac{1}{2}\frac{(x-\mu_n)^2}{\sigma^2+\sigma_n^2}\right] f(\sigma,\sigma_n)$$

$p(x|D)$ is normally distributed with mean μ_n and variance $\sigma^2+\sigma_n^2$: $p(x|D) \sim N(\mu_n, \sigma^2+\sigma_n^2)$

, where $f(\sigma,\sigma_n) = \int \exp\left[-\frac{1}{2}\frac{\sigma^2+\sigma_n^2}{\sigma^2\sigma_n^2}\left(\mu - \frac{\sigma_n^2 x - \sigma^2\mu}{\sigma^2+\sigma_n^2}\right)^2\right] d\mu$

$p(x|D)$ the desired class-conditional density $p(x|\omega_j, D_j)$

MLE: making point estimation for μ and $\hat{\sigma}^2$

B.E. : estimate a distribution for $p(x|D)$

Bayesian Parameter Estimation: Gaussian case

The Multivariate Case:

- Direct generalization of the univariate case.

Assume that $p(\mathbf{x}|\boldsymbol{\mu})\sim N(\boldsymbol{\mu},\boldsymbol{\Sigma})$ and $p(\boldsymbol{\mu})\sim N(\boldsymbol{\mu}_0,\boldsymbol{\Sigma}_0)$

$$p(\boldsymbol{\mu}|D) = \alpha\prod_{k=1}^{n} p(\mathbf{x}_k|\boldsymbol{\mu})p(\boldsymbol{\mu})$$

$$= \alpha' \exp\left[-\frac{1}{2}\left(\boldsymbol{\mu}^t(n\boldsymbol{\Sigma}^{-1}+\boldsymbol{\Sigma}_0^{-1})\boldsymbol{\mu} - 2\boldsymbol{\mu}^t\left(\boldsymbol{\Sigma}^{-1}\sum_{k=1}^{n}\mathbf{x}_k + \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0\right)\right)\right]$$

$$\Longrightarrow\ p(\boldsymbol{\mu}|D) = \alpha'' \exp\left[-\frac{1}{2}(\boldsymbol{\mu}-\boldsymbol{\mu}_n)^t\boldsymbol{\Sigma}_n^{-1}(\boldsymbol{\mu}-\boldsymbol{\mu}_n)\right]\ \Longrightarrow\ p(\boldsymbol{\mu}|D)\sim N(\boldsymbol{\mu}_n,\boldsymbol{\Sigma}_n)$$

$$\boxed{\boldsymbol{\mu}_n = \boldsymbol{\Sigma}_0\left(\boldsymbol{\Sigma}_0+\frac{1}{n}\boldsymbol{\Sigma}\right)^{-1}\hat{\boldsymbol{\mu}}_n + \frac{1}{n}\boldsymbol{\Sigma}\left(\boldsymbol{\Sigma}_0+\frac{1}{n}\boldsymbol{\Sigma}\right)^{-1}\boldsymbol{\mu}_0} \qquad \boxed{\boldsymbol{\Sigma}_n = \boldsymbol{\Sigma}_0\left(\boldsymbol{\Sigma}_0+\frac{1}{n}\boldsymbol{\Sigma}\right)^{-1}\frac{1}{n}\boldsymbol{\Sigma}}$$

$$\boxed{p(\mathbf{x}|D)\sim N(\boldsymbol{\mu}_n,\boldsymbol{\Sigma}+\boldsymbol{\Sigma}_n)}$$

Bayesian Parameter Estimation: General theory

- The basic assumptions:
  - The form of the density $p(\mathbf{x}|\boldsymbol{\theta})$ is assumed to be known, but the value of the parameter vector $\boldsymbol{\theta}$ is not known exactly.
  - Our initial knowledge about $\boldsymbol{\theta}$ is assumed to be contained in a known prior density $p(\boldsymbol{\theta})$.
  - The rest of out knowledge about $\boldsymbol{\theta}$ is contained in a set D of n samples $\mathbf{x}_1, \dots, \mathbf{x}_n$drawn independently according to the unknown probability density $p(\mathbf{x})$.

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

$$p(\boldsymbol{\theta}|D) = \frac{p(D|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(D|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}}$$

By the independence assumption

$$p(D|\boldsymbol{\theta}) = \prod_{k=1}^{n} p(\mathbf{x}_k|\boldsymbol{\theta})$$

Bayesian Parameter Estimation: General theory

- A number of interesting questions remain:
  - Convergence of $p(\mathbf{x}|D)$ to $p(\mathbf{x})$
  - *Difficulty of carrying out the computations* (sec 3.7.2)

To indicate explicitly the number of samples in a set $D^n = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$

$$p(D|\boldsymbol{\theta}) = \prod_{k=1}^{n} p(\mathbf{x}_k|\boldsymbol{\theta})$$

$$\Longrightarrow \quad p(D^n|\boldsymbol{\theta}) = p(\mathbf{x}_n|\boldsymbol{\theta})p(D^{n-1}|\boldsymbol{\theta})$$

$$\Longrightarrow \quad p(\boldsymbol{\theta}|D^n) = \frac{p(\mathbf{x}_n|\boldsymbol{\theta})p(\boldsymbol{\theta}|D^{n-1})}{\int p(\mathbf{x}_n|\boldsymbol{\theta})p(\boldsymbol{\theta}|D^{n-1})d\boldsymbol{\theta}}$$

See example 1 at page 98.

Recursive Bayes approach to parameter estimation.

Incremental or *on-line learning*

Bayesian Parameter Estimation: General theory

When do Maximum-likelihood and Bayes Methods differ?

- Several criteria that influence the choice:
  - Computational complexity (sec. 3.7.2)
    - ML methods are often to be preferred because they require merely differential calculus techniques (gradient search).
    - Bayes methods: complex multidimensional integration.
  - Interpretability
    - ML solution is easier to interpret and understand.
    - Bayes methods give a weighted average of models.

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

  - Confidence in the prior information
    - ML solutions must be of the assumed parametric form.
    - Bayes methods use more of the information brought to the problem than do ML methods by use of the full $p(\boldsymbol{\theta}|D)$.

Bayesian Parameter Estimation: General theory

When do Maximum-likelihood and Bayes Methods differ? (cont.)

- Three sources of classification error in our final system:
  - Bayes or Indistinguishability error
    - □ The error due to overlapping densities $p(\mathbf{x}|\omega_i)$ for different values of i. This error is an inherent property of the problem and can never be eliminated.
  - Model error
    - □ The error due to having an incorrect model. This error can only be eliminated if the designer specifies a model that includes the true model which generated the data.
  - Estimation error
    - □ The error arising from the fact that the parameters are estimated from a finite sample. This error can best be reduced by increasing the training data.

Bayesian Parameter Estimation: General theory

Noninformative Prior and Invariance

- The information about $p(\boldsymbol{\theta})$ derives from the designer's knowledge of the problem domain.
  - "c categories equally likely"
  - In a Bayesian framework, we can have a "noninformative" prior over a parameter for a single category's distribution.
  - Suppose we are using Bayesian methods to infer from data some position and scale parameters:
    - □ Translation invariance
    - □ Scale invariance

Gibbs Algorithm

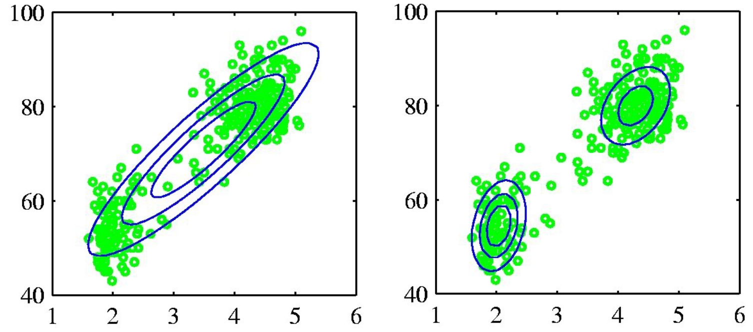
$$p(\mathbf{x}|D) = \int p(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta}|D)d\boldsymbol{\theta}$$

- The integration is very difficult.
- A simple alternative is to pick a parameter vector $\boldsymbol{\theta}$ according to $p(\boldsymbol{\theta}|D)$ and use the single value as if it were the true value.
- The Gibbs algorithm gives a misclassification error that is at most twice the expected error of the Bayes optimal classifier.

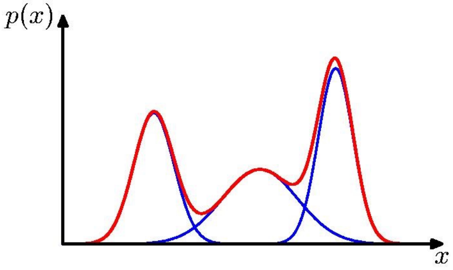
PRML 2.3.9

Mixture of Gaussians

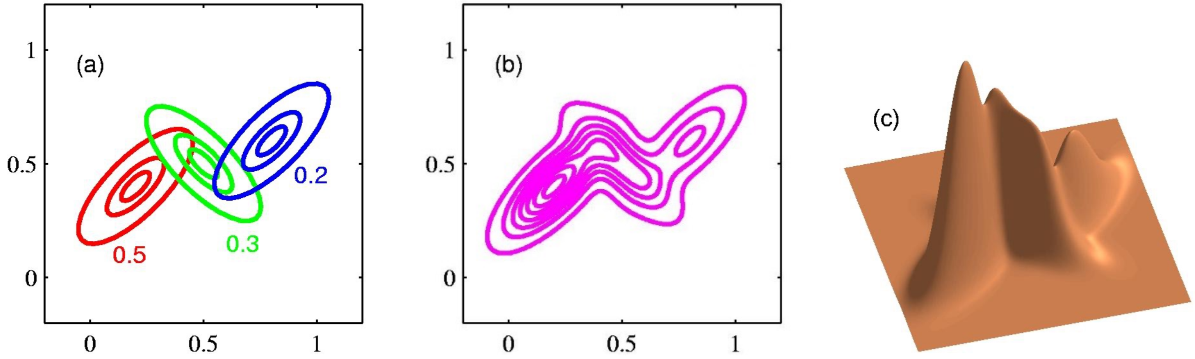
- In the data set in the figure
  - A simple Gaussian distribution is unable to capture this structure.



  - $p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k N(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$
    - $\sum_{k=1}^{K} \pi_k = 1$



  - A mixture of 3 Gaussians



Problems of Dimensionality

- We might typically believe that each feature is useful for at least some of the discrimination!
- Two issues that must be confronted:
  - How classification accuracy depends on the dimensionality (and amount of training data)
  - The computational complexity of designing the classifier.

Problems of Dimensionality

Accuracy, dimension, and training sample size

- Bayes error rate $p(\mathbf{x}|\omega_j) \sim N(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}), j = 1,2$

$$P(e) = \frac{1}{\sqrt{2\pi}} \int_{r/2}^{\infty} e^{-u^2/2}\, du \quad \text{where} \quad r^2 = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^t \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

  - $P(e)$ decreases as r increases (approaching zero as r approaches infinity)
  - In the conditionally independent case

$$\boldsymbol{\Sigma} = diag(\sigma_1^2, \ldots, \sigma_d^2)$$

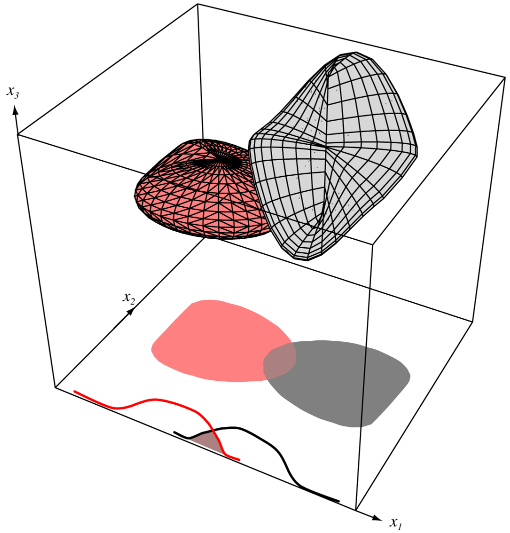
$$r^2 = \sum_{i=1}^{d} \left(\frac{\mu_{i1} - \mu_{i2}}{\sigma_i} \right)^2$$

    - This shows each feature contributes to reducing the $P(e)$
    - The most useful features:
      - □ Ones for which the difference between the means is large relative to the standard deviations.
    - No feature is useless if its means for the two classes differ.

Problems of Dimensionality

Accuracy, dimension, and training sample size (cont.)

- In general, it is natural to consider adding new features, if the performance with a given set of features is inadequate.
  - Increasing the number of features increases the cost and complexity of both the feature extractor and the classifier.
  - If the probabilistic structure of the problem were completely known, the Bayes risk could not be increased by adding new features.



Two 3-dim distributions have nonoverlapping densities, and thus in three dimensions the Bayes error vanishes. When projected to a subspace there can be greater overlap of the projected distributions, and hence greater Bayes error.

Problems of Dimensionality

Accuracy, dimension, and training sample size (cont.)

- Unfortunately, it has been observed in practice that, beyond a certain point, the inclusion of additional features leads to worse than better performance.
  - The source of the difficulty:
    - □ Wrong model (Gaussian assumption, conditional assumption)
    - □ The finite number of training samples
    - □ Inaccurate estimation of the distributions

Problems of Dimensionality

Computational complexity

- The technical notion of computational complexity
  - Page 111 and A.8 at page 633
- We are interested in the number of basic mathematical operations (additions, multiplications, and divisions), or in the time and memory needed.
  - MLE for Gaussian priors in d dimensions, with n training samples for each of c categories.

$$g(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \overset{O(dn)}{\overset{\uparrow}{\hat{\boldsymbol{\mu}}}})^t \overbrace{\hat{\boldsymbol{\Sigma}}^{-1}}^{O(nd^2)} (\mathbf{x} - \hat{\boldsymbol{\mu}}) - \overbrace{\frac{d}{2}\ln 2\pi}^{O(1)} - \overbrace{\frac{1}{2}\ln\left|\hat{\boldsymbol{\Sigma}}\right|}^{O(d^2 n)} + \overbrace{\ln P(\omega)}^{O(n)}$$

  - If we assume that $n > d$, the overall complexity of calculating an individual discriminant function is dominated by the $O(d^2 n)$.
    - Overall computational complexity: $O(cd^2 n)$.

Problems of Dimensionality

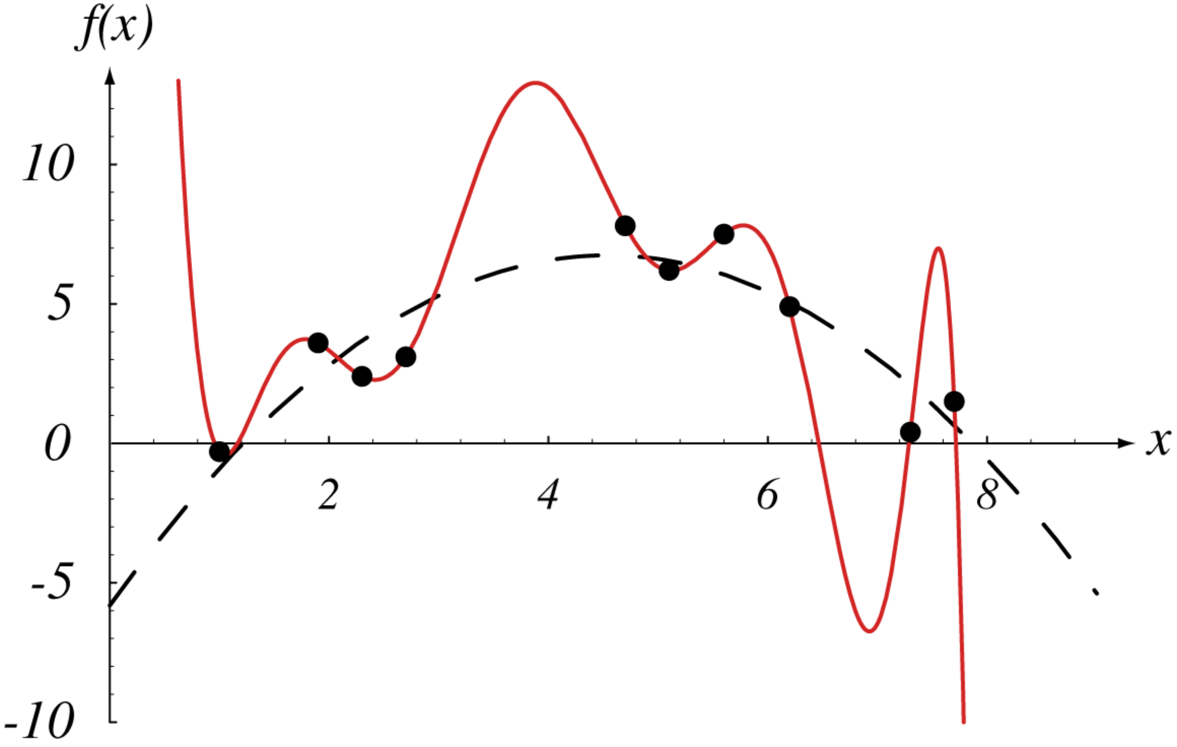
Computational complexity (cont.)

- Sometimes we stress space and time complexities which are particularly relevant when contemplating parallel implementations.
  - Time-space tradeoffs
    - Using a single processor many times, or using many processors in parallel for a short time.

- A common qualitative distinction is between
  - Polynomially complex and
  - Exponentially complex - $O(a^k)$

Problems of Dimensionality

Overfitting



The training data (black dots) were selected from a quadratic function plus Gaussian noise. The 10th-degree polynomial shown fits the data perfectly, but we desire instead the second-order function $f(x)$, because it would lead to better predictions for new samples.

Component analysis and discriminants

Principal Component Analysis(PCA)

- One approach to coping with the problem of excessive dimensionality.
  - The problem of representing all of the vectors in a set of n d-dimensional samples $\mathbf{x}_1, \dots, \mathbf{x}_n$ by a single vector $\mathbf{x}_0$.
  - The squared-error criterion function $J_0(\mathbf{x}_0)$

$$J_0(\mathbf{x}_0) = \sum_{k=1}^{n} \|\mathbf{x}_0 - \mathbf{x}_k\|^2$$

To seek the value of $\mathbf{x}_0$ that minimizes J_0

  - It's simple to show that the solution to this problem is given by $\mathbf{x}_0 = \mathbf{m}$, where $\mathbf{m}$ is the sample mean. (Eq. 79, 80)
    - Too simple? – it does not reveal any of the variability in the data.

Component analysis and discriminants

Principal Component Analysis(PCA) (cont.)

- One-dimensional representation by projecting the data onto a line running through the sample mean.
  - Let $\mathbf{e}$ be a unit vector in the direction of the line

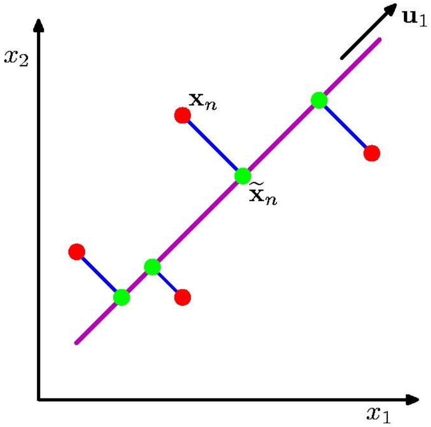
$$\mathbf{x} = \mathbf{m} + a\mathbf{e}$$

a corresponds to the distance of any point $\mathbf{x}$ from the mean $\mathbf{m}$

$$J_1(a_1, \dots, a_n, \mathbf{e}) = \sum_{k=1}^{n} \|(\mathbf{m} + a_k\mathbf{e}) - \mathbf{x}_k\|^2 = \sum_{k=1}^{n} \|a_k\mathbf{e} - (\mathbf{x}_k - \mathbf{m})\|^2$$

$$= \sum_{k=1}^{n} a_k^2\|\mathbf{e}\|^2 - 2\sum_{k=1}^{n} a_k\mathbf{e}^t(\mathbf{x}_k - \mathbf{m}) + \sum_{k=1}^{n} \|\mathbf{x}_k - \mathbf{m}\|^2$$

PRML 12.2



$$\|\mathbf{e}\| = 1$$

$$a_k = \mathbf{e}^t(\mathbf{x}_k - \mathbf{m})$$

projecting the vector $\mathbf{x}_k$onto the line in the direction of $\mathbf{e}$ that passes through the sample mean

Component analysis and discriminants

Principal Component Analysis(PCA) (cont.)

- The more interesting problem of finding the best direction $\mathbf{e}$ for the line
- Scatter Matrix

$$\mathbf{S} = \sum_{k=1}^{n} (\mathbf{x}_k - \mathbf{m})(\mathbf{x}_k - \mathbf{m})^t$$

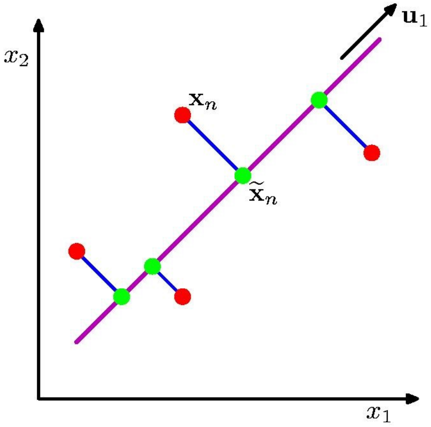
$$\begin{aligned} J_1(\mathbf{e}) &= \sum_{k=1}^{n} a_k^2 - 2\sum_{k=1}^{n} a_k^2 + \sum_{k=1}^{n} \|\mathbf{x}_k - \mathbf{m}\|^2 \\ &= -\sum_{k=1}^{n} [\mathbf{e}^t(\mathbf{x}_k - \mathbf{m})]^2 + \sum_{k=1}^{n} \|\mathbf{x}_k - \mathbf{m}\|^2 \\ &= -\sum_{k=1}^{n} \mathbf{e}^t(\mathbf{x}_k - \mathbf{m})(\mathbf{x}_k - \mathbf{m})^t\mathbf{e} + \sum_{k=1}^{n} \|\mathbf{x}_k - \mathbf{m}\|^2 \\ &= -\mathbf{e}^t\mathbf{S}\mathbf{e} + \sum_{k=1}^{n} \|\mathbf{x}_k - \mathbf{m}\|^2 \end{aligned}$$

The vector $\mathbf{e}$ that minimize J_1 also maximizes $\mathbf{e}^t\mathbf{S}\mathbf{e}$.

Component analysis and discriminants

Principal Component Analysis(PCA) (cont.)

PRML 12.2



$$u = \mathbf{e}^t\mathbf{S}\mathbf{e} - \lambda(\mathbf{e}^t\mathbf{e} - 1)$$

$$\frac{\partial u}{\partial \mathbf{e}} = 2\mathbf{S}\mathbf{e} - 2\lambda\mathbf{e}$$

$$\mathbf{S}\mathbf{e} = \lambda\mathbf{e}$$

$$\mathbf{e}^t\mathbf{S}\mathbf{e} = -\lambda\mathbf{e}^t\mathbf{e} = \lambda$$

We want to select the eigenvetors corresponding to the largest eigenvalues of the scatter matrix.

We project the data onto a line through the sample mean in the direction of the eigenvector of the scatter matrix having the largest eigenvelue.

Component analysis and discriminants

Principal Component Analysis(PCA) (cont.)

- Extension from a one-dim projection to a d'-dim projection.

$$\mathbf{x} = \mathbf{m} + \sum_{i=1}^{d'} a_i \mathbf{e}_i$$

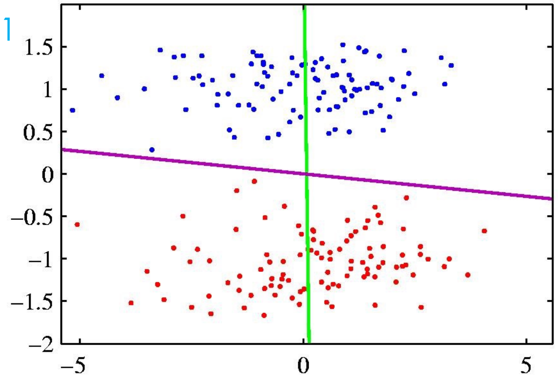
The coefficients a_i are the components of $\mathbf{x}$ in that basis, and are called the principal components

- Geometrically, if we picture the data points $\mathbf{x}_1, \dots, \mathbf{x}_n$ as forming a d-dimensional, hyperellipsoidally shaped cloud, then the eigenvectors of the scatter matrix are the principal axes of that hyperellipsoid.
- PCA reduces the dimensionality of feature space by restricting attention to those directions along which the scatter of the cloud is greatest.

Component analysis and discriminants

Fisher linear discriminant

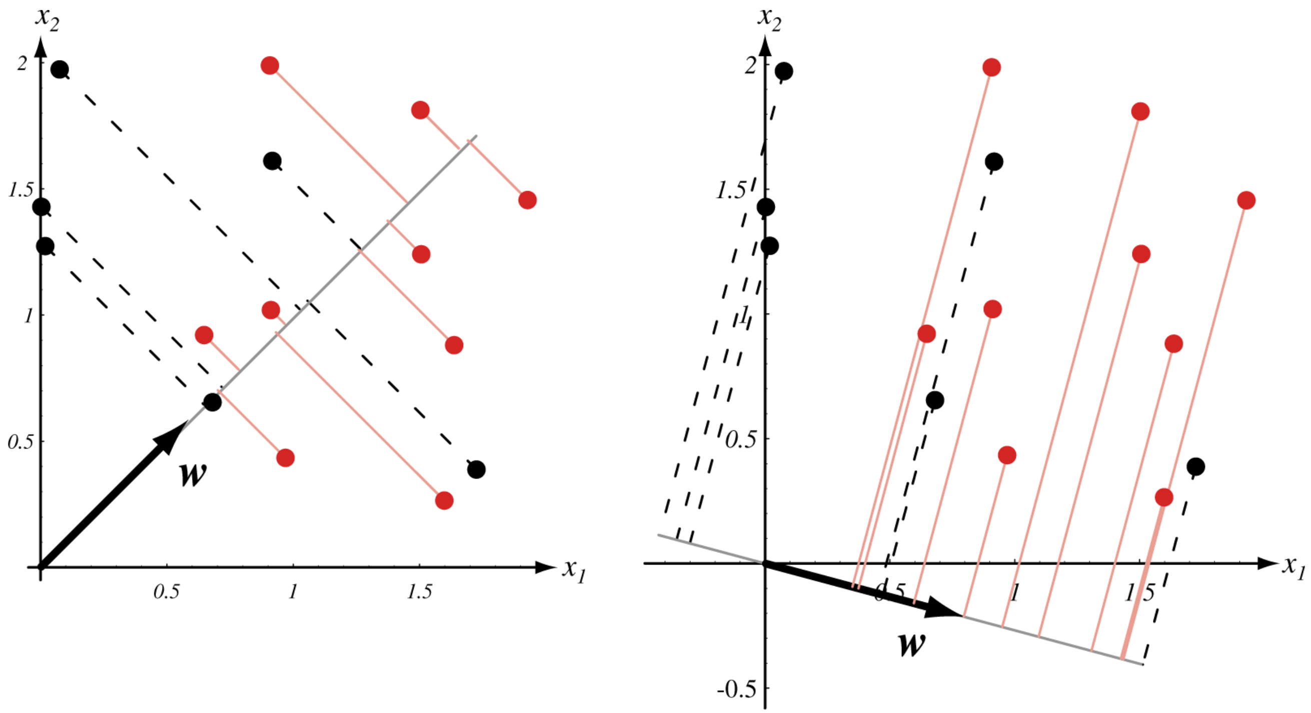
PRML 12.1



- PCA
  - Finding components that are useful for representing data
  - No reason to assume that these components must be useful for discriminating between data in different classes.
- The directions that are discarded by PCA might be exactly the directions that are needed for distinguishing between classes.
  - Q and O
- Projecting data from d-dimensions onto a line
  - If we form a linear combination of the components of $\mathbf{x}$, we obtain the scalar dot product $y = \mathbf{w}^t\mathbf{x}$.
  - By moving the line around, we might be able to find an orientation for which the projected samples are well separated.
  - The direction of $\mathbf{w}$ is important.

Component analysis and discriminants

Fisher linear discriminant (cont.)



Projection of the same set of samples onto two different lines in the directions marked $\mathbf{w}$. The figure on the right shows greater separation between the red and black projected points.

Component analysis and discriminants

Fisher linear discriminant (cont.)

- Finding the best such direction $\mathbf{w}$

$$\mathbf{m}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in D_i} \mathbf{x}$$

d-dimensional sample mean

$$\tilde{m}_i = \frac{1}{n_i} \sum_{y \in Y_i} y$$

$$= \frac{1}{n_i} \sum_{\mathbf{x} \in D_i} \mathbf{w}^t \mathbf{x} = \mathbf{w}^t \mathbf{m}_i$$

the sample mean for the projected points
It's simply the projection of $\mathbf{m}_i$.

$$|\tilde{m}_1 - \tilde{m}_2| = |\mathbf{w}^t(\mathbf{m}_1 - \mathbf{m}_2)|$$

the distance between the projected means

The difference between the means to be large relative to some measure of the standard deviation for each class: *scatter*

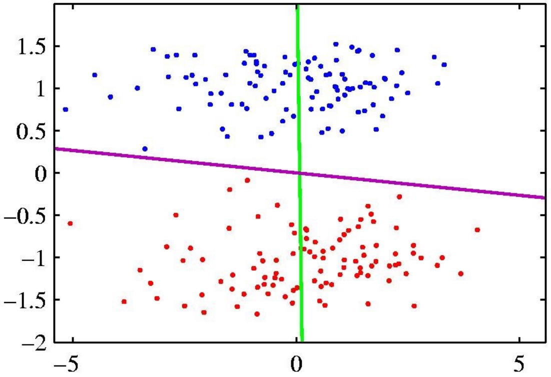
$$\tilde{s}_i^2 = \sum_{y \in Y_i} (y - \tilde{m}_i)^2$$

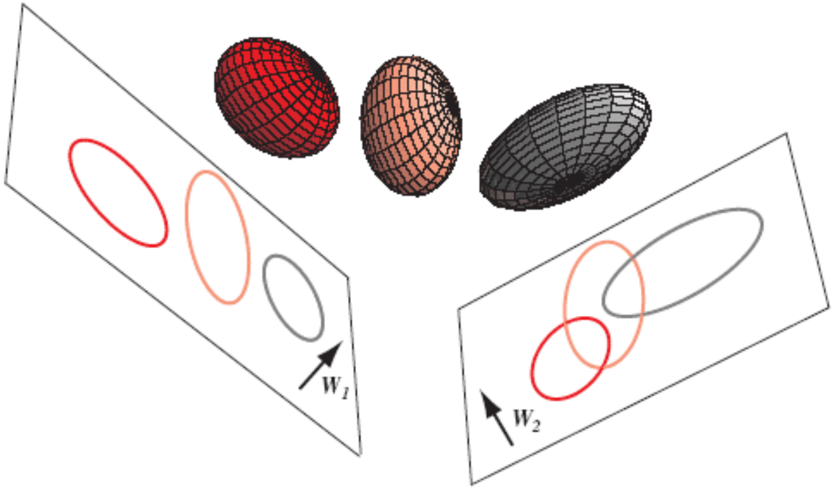
Component analysis and discriminants

Fisher linear discriminant (cont.)

- The Fisher linear discriminant employs that linear function $\mathbf{w}^t\mathbf{x}$ for which the criterion function $J(\mathbf{w}) = \frac{|\tilde{m}_1 - \tilde{m}_2|}{\tilde{s}_1^2 + \tilde{s}_2^2}$ is maximum.

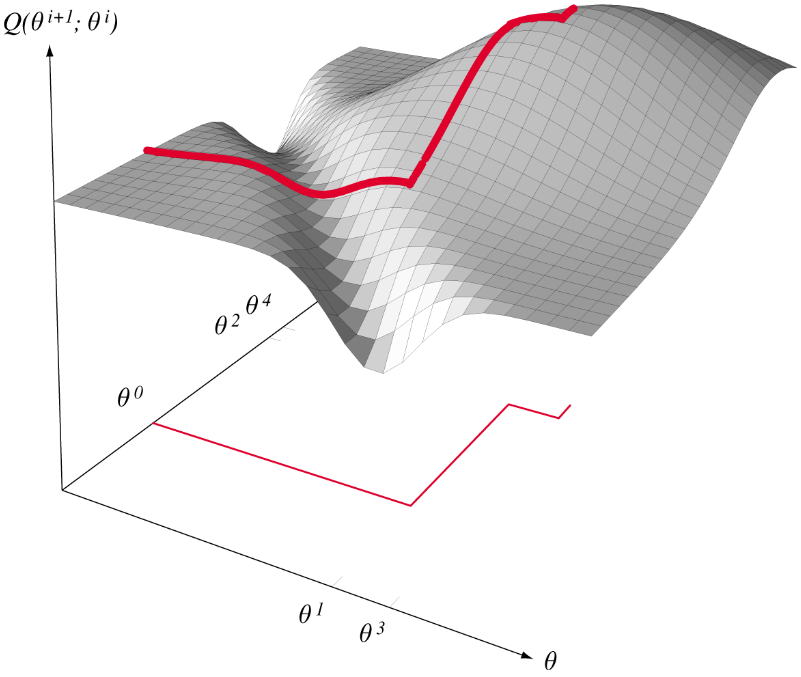
- Multiple Discriminant Analysis
  - The generalization of Fisher's linear discriminant function to c-class problem.



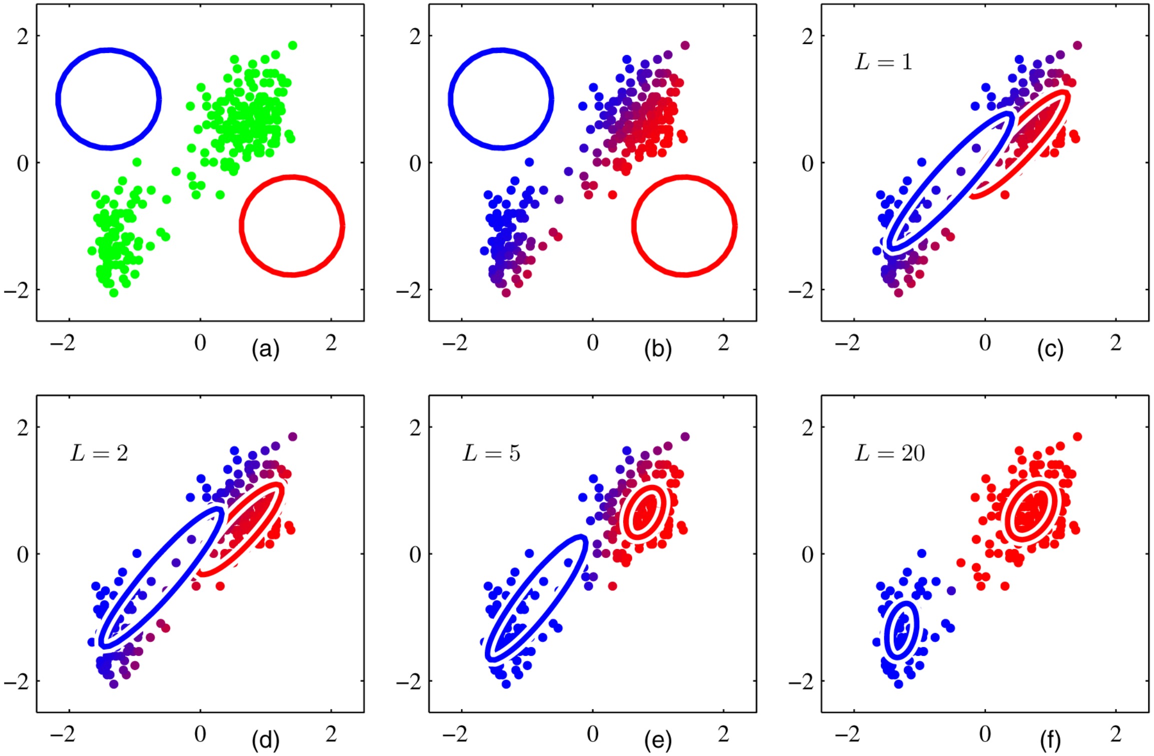


Expectation-Maximization(EM)

- Extension of maximum-likelihood techniques to permit the learning of parameters governing a distribution from training points, some of which have missing features.
- The basic idea in the EM is to iteratively estimate the likelihood given the data that is present.



See example 2 at page 126.



PRML 9.8

Hidden Markov Models (HMMs)

- Our attention to the problem of estimating the parameters in the class-conditional densities have been limited to make a *single* decision.

- The problem of making a *sequence* of decisions.
  - An inherent temporality
    - A process that unfold in time

- HMMs have found greatest use in speech recognition or gesture recognition.

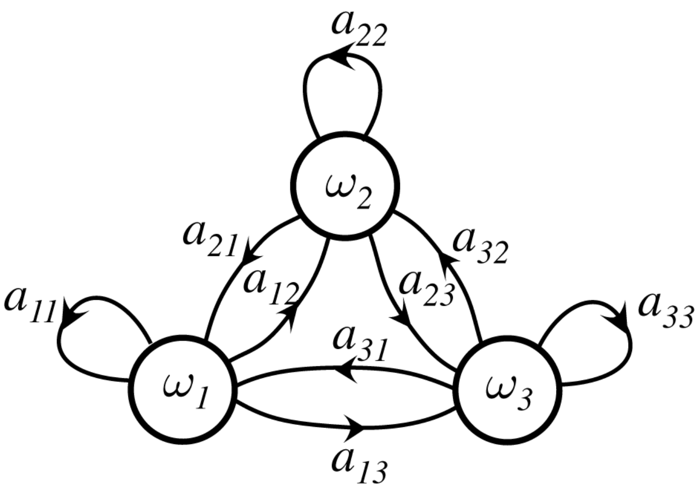
- HMMs have a number of parameters whose values are set so as to best explain training patterns for the known category.
  - A test pattern is classified by the model that has the highest posterior probability – that best explains the test pattern.

Hidden Markov Models (HMMs)

First-order Markov models

- A sequence of states at successive times;
  - $\omega(t)$: the state at any time t.
  - A sequence of length T: $\boldsymbol{\omega}^T = \{\omega(1), \omega(2), \ldots, \omega(T)\}$
  - Example: $\boldsymbol{\omega}^6 = \{\omega_1, \omega_4, \omega_2, \omega_2, \omega_1, \omega_4\}$ – the system can revisit a state a different steps

  - Transition probabilities: a_{ij} $\quad P(\omega_j(t+1) \mid \omega_i(t))$



$$P(\boldsymbol{\omega}^T \mid \boldsymbol{\theta}) = a_{14}a_{42}a_{22}a_{21}a_{14}$$

Hidden Markov Models (HMMs)

First-order Markov models (cont.)

- We assume that at every time step t the system is in a state $\omega(\mathrm{t})$ but now we also assume that it emits some (visible) symbol $v(\mathrm{t})$.
- Restrict to the case where a discrete symbol is emitted.
  - Visible states

$$\mathbf{V}^T = \{v(1), v(w), \ldots, v(T)\}$$

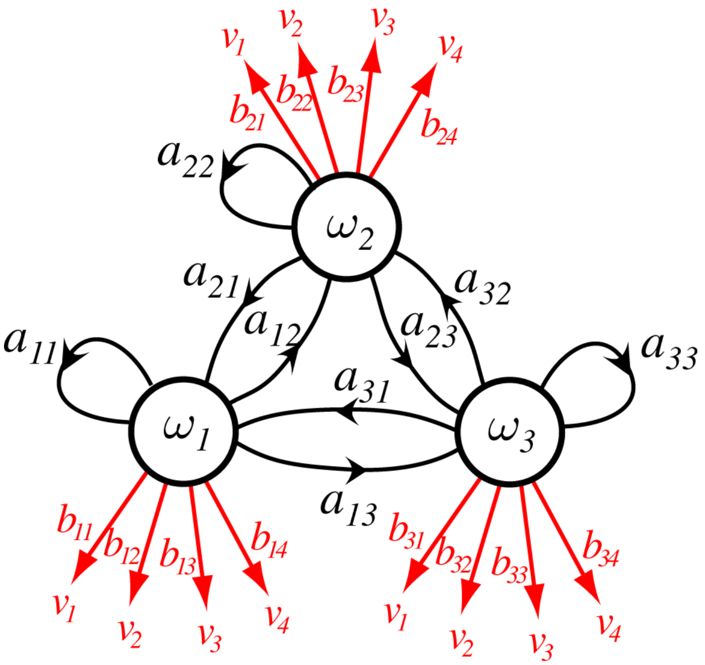
- Our model is then that in any state $\omega(\mathrm{t})$ we have a probability of emitting a particular visible state $v_{\mathrm{k}}(\mathrm{t})$.
  - emission probabilities: b_{jk}

$$P(v_k(t) \mid \omega_j(t))$$

  - We have access only to the *visible* states, while ω_{l} are *unobservable* – *hidden Markov model*

Hidden Markov Models (HMMs)

Hidden Markov model computation



Three hidden units in an HMM and the transitions between them are shown in black while the visible states and the emission probabilities of visible state are shown in red.

Causal: the probabilities depend on previous states.

Ergodic: if every one of the states has a nonzero probability of occurring given some stating state.

$$a_{ij} = P(\omega_j(t+1) \mid \omega_i(t))$$

$$b_{jk} = P(v_k(t) \mid \omega_j(t))$$

$$\sum_j a_{ij} = 1 \text{ for all } i$$

$$\sum_k b_{jk} = 1 \text{ for all } j$$

Hidden Markov Models (HMMs)

Hidden Markov model computation

- Three central issues in HMMs:
  - The Evaluation problem
    - Suppose we have an HMM, complete with transitions probabilities a_{ij} and b_{jk}. Determine the probability that a particular sequence of visible state $\mathbf{V}^T$ was generated by that model.
  - The Decoding problem
    - Suppose we have an HMM as well as a set of observations $\mathbf{V}^T$. Determine the most likely sequence of hidden state ω^T that led to those observations.
  - The Learning problem
    - Suppose we are given the coarse structure of a model (the number of states and the number of visible states) but not the probabilities a_{ij} and b_{jk}. Given a set of training observations of visible symbols, determine these parameters.

Hidden Markov Models (HMMs)

HMMs - Evaluation

$$P(\mathbf{V}^T) = \sum_{r=1}^{r_{\max}} P(\mathbf{V}^T \mid \boldsymbol{\omega}_r^T) P(\boldsymbol{\omega}_r^T)$$

where r indexes a particular sequence $\boldsymbol{\omega}_r^T = \{\omega(1), \omega(2), ..., \omega(T)\}$

$$P(\boldsymbol{\omega}_r^T) = \prod_{t=1}^{T} P(\omega(t) \mid \omega(t-1))$$

transition prob. for the hidden state

$$P(\mathbf{V}^T \mid \boldsymbol{\omega}_r^T) = \prod_{t=1}^{T} P(v(t) \mid \omega(t))$$

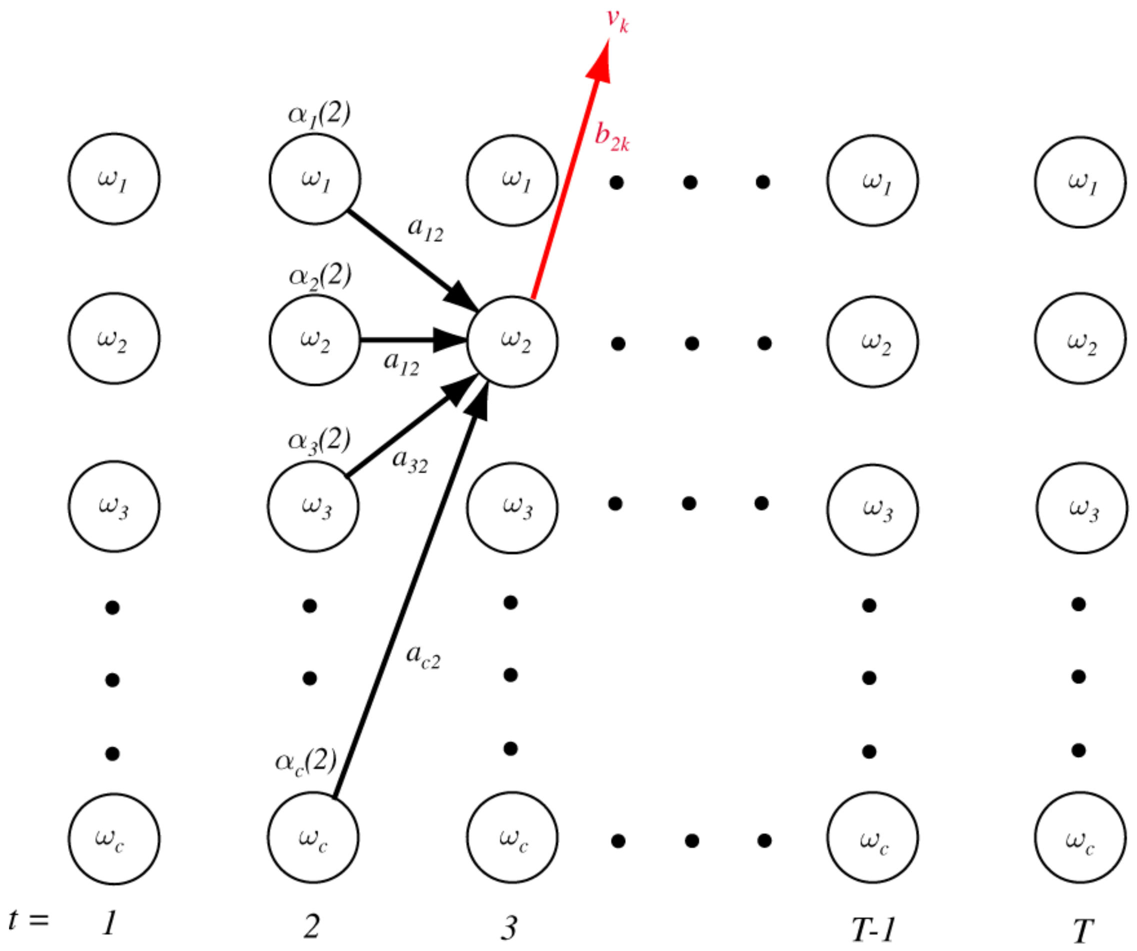
$$\Rightarrow \qquad P(\mathbf{V}^T) = \sum_{r=1}^{r_{\max}} \prod_{t=1}^{T} P(v(t) \mid \omega(t)) P(\omega(t) \mid \omega(t-1))$$

A computationally simpler algorithm (recursive)

$$\alpha_i(t) = \begin{cases} 0 & t = 0 \text{ and } i \neq \text{initial state} \\ 1 & t = 0 \text{ and } i = \text{initial state} \\ \sum_j \alpha_j(t-1) a_{ij} b_{jk} v(t) & \text{otherwise} \end{cases}$$

Hidden Markov Models (HMMs)

HMMs – Evaluation (cont.)

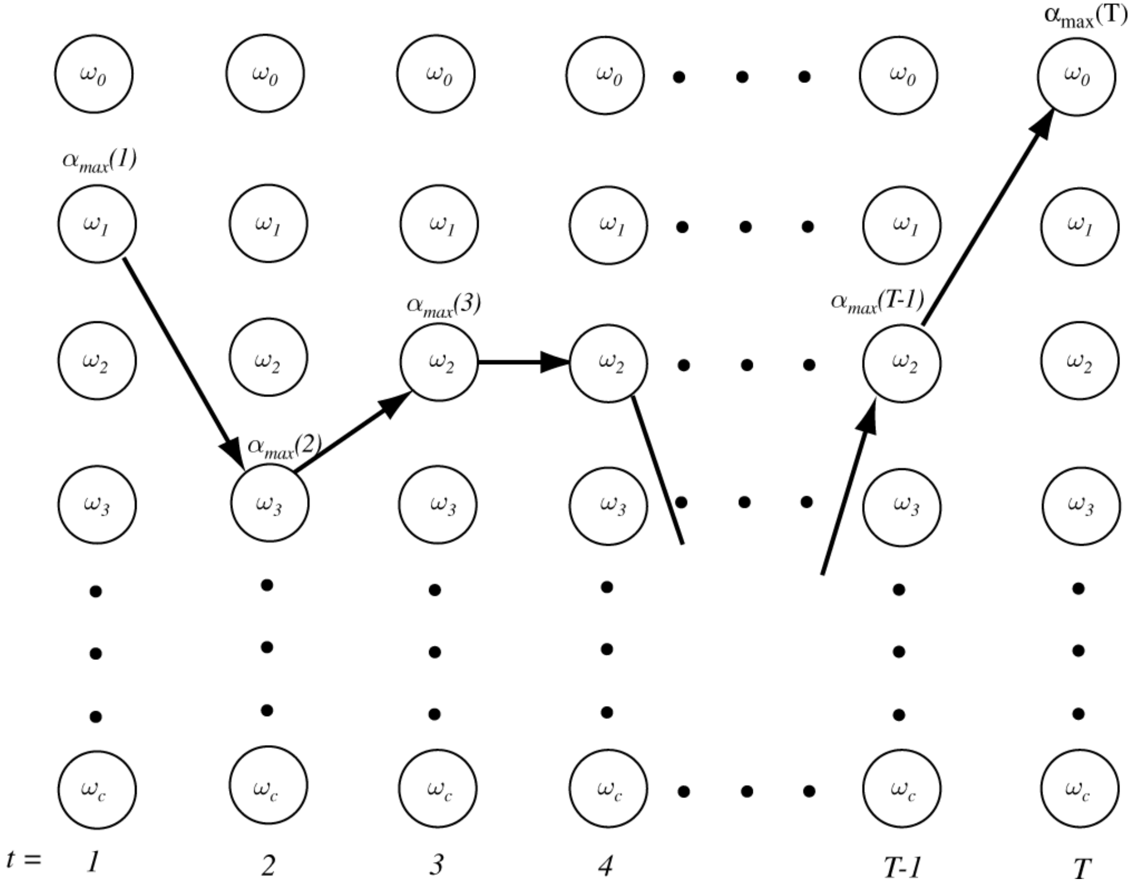


See example 3 at page 133.

Hidden Markov Models (HMMs)

HMMs – Decoding

To find the most probable sequence of hidden states.



See example 4 at page 136.

Hidden Markov Models (HMMs)

HMMs – Learning

- To determine model parameters from an ensemble of training samples.
  - a_{ij} and b_{jk}
  - No known method for obtaining the optimal or most likely set of parameters from the data.

$$\alpha_i(t) = \begin{cases} 0 & t = 0 \text{ and } i \neq \text{initial state} \\ 1 & t = 0 \text{ and } i = \text{initial state} \\ \sum_j \alpha_j(t-1) a_{ij} b_{jk} v(t) & \text{otherwise} \end{cases}$$

$$\beta_i(t) = \begin{cases} 0 & \omega_i(t) \neq \text{sequence's final state and } t = T \\ 1 & \omega_i(t) = \text{sequence's final state and } t = T \\ \beta_j(t+1) \sum_j a_{ij} b_{jk} v(t+1) & \text{otherwise} \end{cases}$$

$$\gamma_{ij}(t) = \frac{\alpha_i(t-1) a_{ij} b_{ij} \beta_j(t)}{P(\mathbf{V}^T | \boldsymbol{\theta})}$$

Hidden Markov Models (HMMs)

HMMs – Learning (cont.)

$$\hat{a}_{ij}(t) = \frac{\sum_{t=1}^{T} \gamma_{ij}(t)}{\sum_{t=1}^{T} \sum_{k} \gamma_{ik}(t)}$$

The estimate of the transition probability

$$\hat{b}_{jk} = \frac{\sum_{t=1}^{T} b_{jk}(t)}{\sum_{t=1}^{T} \sum_{k} b_{jk}(t)}$$

The estimate of the emission probability

Baum-Weltch or forward-backward algorithm