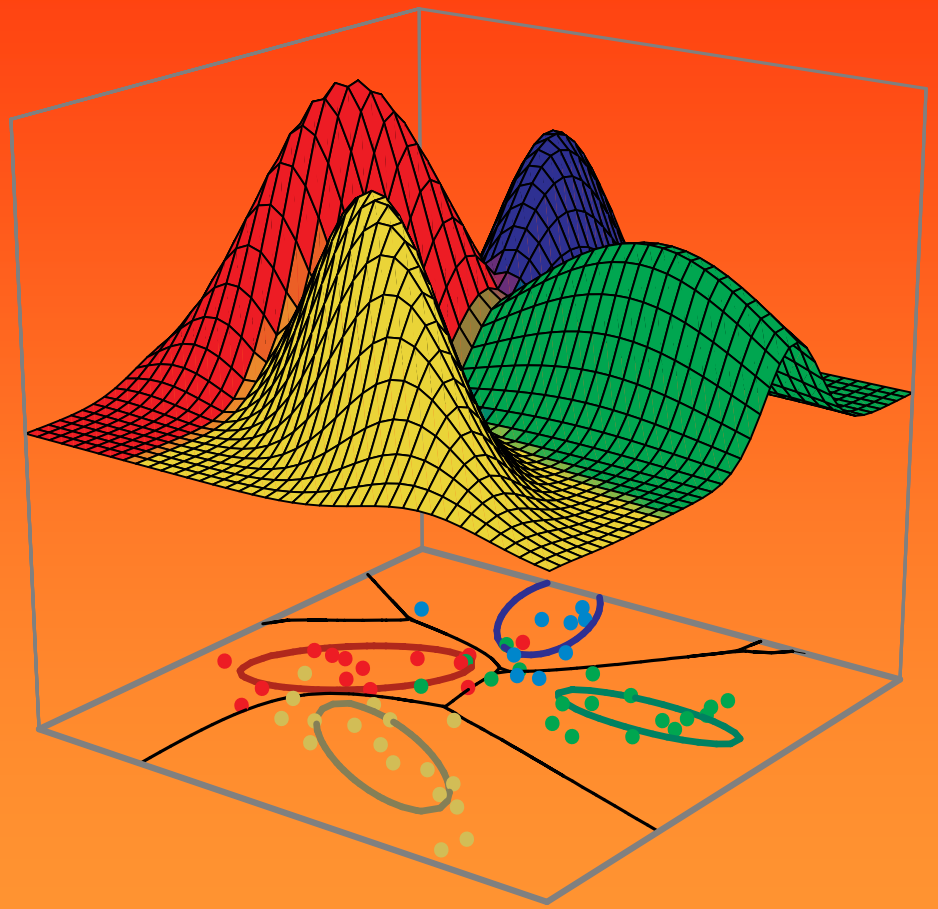


Pattern Classification

All materials in these slides were taken from
Pattern Classification (2nd ed) *by R. O. Duda, P. E. Hart and D. G. Stork, John Wiley & Sons, 2000*
with the permission of the authors and the publisher

Chapter 3: Maximum-Likelihood and Bayesian Parameter Estimation (part 2)

- Bayesian Estimation (BE)
- Bayesian Parameter Estimation: Gaussian Case
- Bayesian Parameter Estimation: General Estimation
- Problems of Dimensionality
- Computational Complexity
- Component Analysis and Discriminants
- Hidden Markov Models

- Bayesian Estimation (Bayesian learning to pattern classification problems)
  - In MLE θ was supposed fix
  - In BE θ is a random variable
  - The computation of posterior probabilities $P(\omega_i \mid x)$ lies at the heart of Bayesian classification
  - Goal: compute $P(\omega_i \mid x, D)$

    Given the sample D, Bayes formula can be written

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

- To demonstrate the preceding equation, use:

$$P(\mathbf{x}, D \mid \omega_i) = P(\mathbf{x} \mid D, \omega_i) P(D \mid \omega_i) \text{ (from def. of cond. prob.)}$$

$$P(\mathbf{x} \mid D) = \sum_j P(\mathbf{x}, \omega_j \mid D) \quad \text{(from law of total prob.)}$$

$$P(\omega_i) = P(\omega_i \mid D) \text{ (Training sample provides this!)}$$

We assume that samples in different classes are independent

Thus :

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

- Bayesian Parameter Estimation: Gaussian Case

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

- The univariate case: $P(\mu \mid D)$

  μ is the only unknown parameter

$$P(x \mid \mu) \sim N(\mu, \sigma^2)$$

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

(μ_0 and σ_0 are known!)

$$P(\mu \mid \mathsf{D}) = \frac{P(\mathsf{D} \mid \mu)P(\mu)}{\int P(\mathsf{D} \mid \mu)P(\mu)d\mu} \qquad \text{Bayes Formula}$$

$$= \alpha \prod_{k=1}^{k=n} P(x_k \mid \mu)P(\mu)$$

- But we know

$$p(x_k \mid \mu) \sim N(\mu, \sigma^2) \text{ and } p(\mu) \sim N(\mu_0, \sigma_0^2)$$

Plugging in their gaussian expressions and extracting out factors not depending on μ yields:

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

(from eq. 29 page 93)

Observation: p(μ|D) is an exponential of a quadratic

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

It is again normal! It is called a *reproducing density*

$$P(\mu \mid \mathrm{D}) \sim N(\mu_n, \sigma_n^2)$$

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

- Identifying coefficients in the top equation with that of the generic Gaussian

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

Yields expressions for μ_n and σ_n^2

$$\frac{1}{\sigma_n^2} = \frac{n}{\sigma^2} + \frac{1}{\sigma_0^2} \quad \text{and} \quad \frac{\mu_n}{\sigma_n^2} = \frac{n}{\sigma^2}\hat{\mu}_n + \frac{\mu_0}{\sigma_0^2}$$

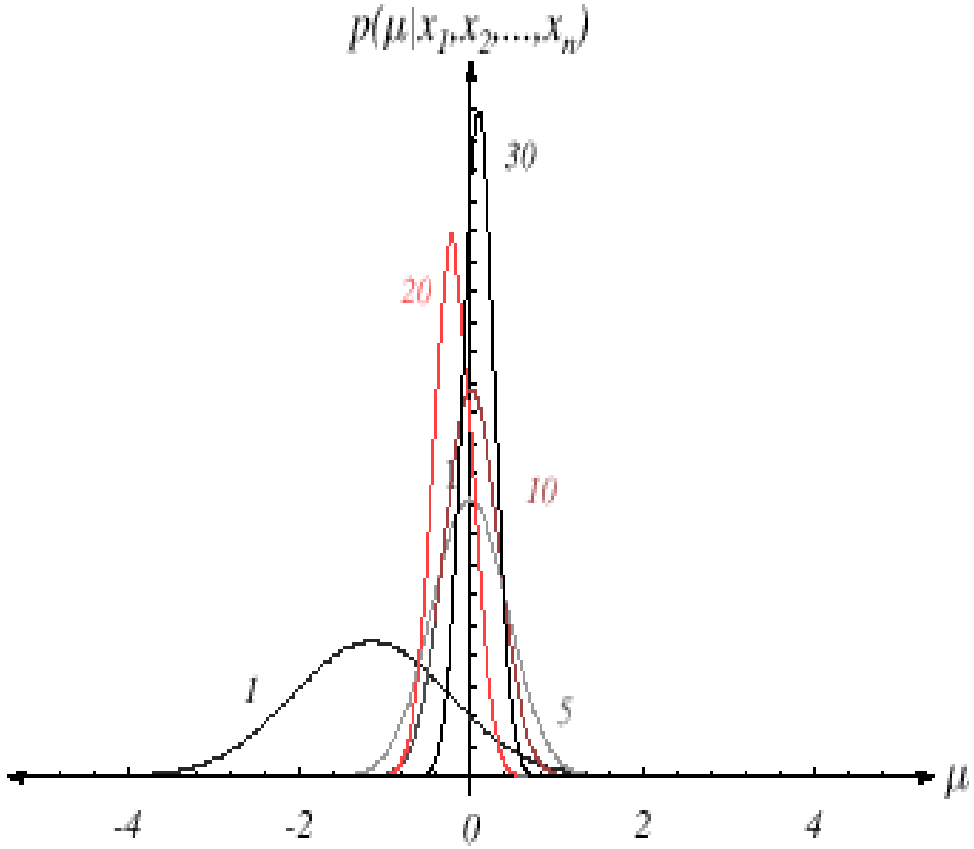
Solving for μ_n and σ_n^2 yields:

$$\mu_n = \left(\frac{n\sigma_0^2}{n_0\sigma_0^2 + \sigma^2} \right) \hat{\mu}_n + \frac{\sigma^2}{n\sigma_0^2 + \sigma^2} \mu_0$$

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

From these equations we see as *n* increases:

- the variance decreases monotonically
- the estimate of p(μ|D) becomes more peaked



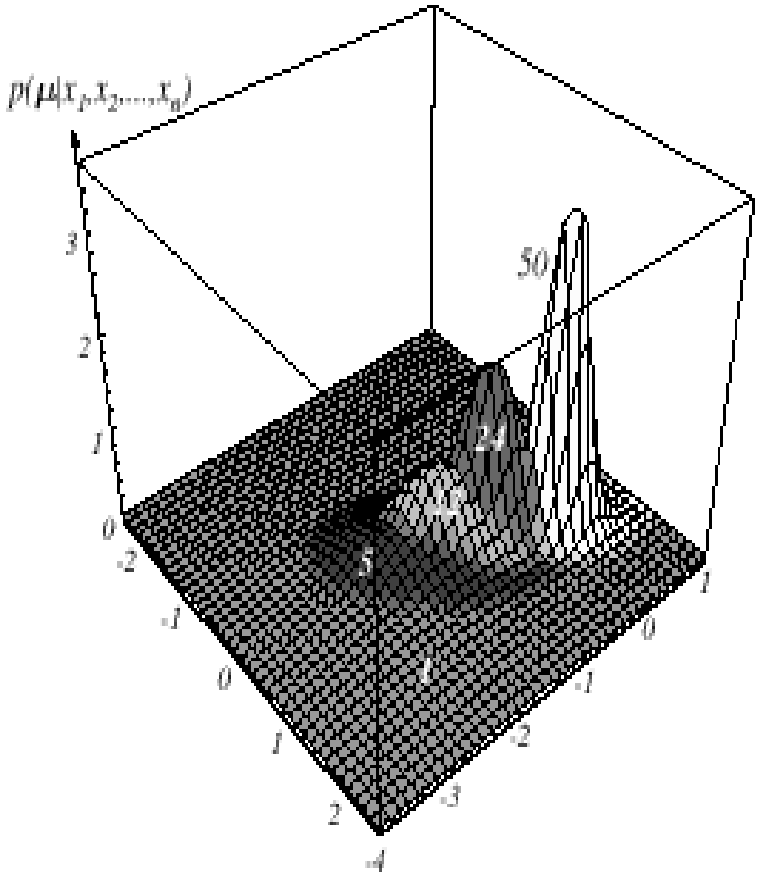


FIGURE 3.2. Bayesian learning of the mean of normal distributions in one and two dimensions. The posterior distribution estimates are labeled by the number of training samples used in the estimation. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

- The univariate case P(x | D)
  - P(μ | D) computed (in preceding discussion)
  - P(x | D) remains to be computed!

$$P(x \mid \mathrm{D}) = \int P(x \mid \mu) P(\mu \mid \mathrm{D}) d\mu \text{ is Gaussian}$$

It provides: $P(x \mid \mathrm{D}) \sim N(\mu_n, \sigma^2 + \sigma_n^2)$

We know σ^2 and how to compute μ_n and σ_n^2

(Desired class-conditional density $P(x \mid D_j, \omega_j)$)

Therefore: using $P(x \mid D_j, \omega_j)$ together with $P(\omega_j)$

And using Bayes formula, we obtain the

Bayesian classification rule:

$$\underset{\omega_j}{Max} \left[P(\omega_j \mid x, \mathrm{D} \right] \equiv \underset{\omega_j}{Max} \left[P(x \mid \omega_j, \mathrm{D}_j) P(\omega_j) \right]$$

- 3.5 Bayesian Parameter Estimation: General Theory

  - P(x | D) computation can be applied to any situation in which the unknown density can be parameterized: the basic assumptions are:

    - The form of P(x | θ) is assumed known, but the value of θ is not known exactly
    - Our knowledge about θ is assumed to be contained in a known prior density P(θ)
    - The rest of our knowledge θ is contained in a set D of n random variables x_1, x_2, …, x_n that follows P(x)

The basic problem is:

"Compute the posterior density P(θ | D)"

then "Derive P(x | D)", where $$p(\mathbf{x} \mid D) = \int p(\mathbf{x} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid D) d\boldsymbol{\theta}$$

Using Bayes formula, we have:

$$P(\boldsymbol{\theta} \mid \mathrm{D}) = \frac{P(\mathrm{D} \mid \boldsymbol{\theta}) P(\boldsymbol{\theta})}{\int P(\mathrm{D} \mid \boldsymbol{\theta}) P(\boldsymbol{\theta}) d\boldsymbol{\theta}},$$

And by the independence assumption:

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

Convergence (from notes)

- Problems of Dimensionality
  - Problems involving 50 or 100 features are common (usually binary valued)
  - Note: microarray data might entail ~20000 real-valued features
  - Classification accuracy dependant on
    - dimensionality
    - amount of training data
    - discrete vs continuous

- Case of two class multivariate normal with the same covariance
  - P(x|ω_j) ~$N(\mu_j,\Sigma)$, j=1,2
  - Statistically independent features
  - If the priors are equal then:

$$P(error) = \frac{1}{\sqrt{2\pi}} \int_{r/2}^{\infty} e^{-u^2/2}\, du \qquad \text{(Bayes error)}$$

$$where: \quad r^2 = (\mu_1 - \mu_2)^t \Sigma^{-1} (\mu_1 - \mu_2)$$

r^2 is the squared Mahalanobi s distance

$$\lim_{r \to \infty} P(error) = 0$$

- If features are ***conditionally*** independent then:

$$\Sigma = \mathbf{diag}(\sigma_1^2, \sigma_2^2, ..., \sigma_d^2)$$

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

- Do we remember what conditional independence is?
- Example for binary features:

  Let $p_i = Pr[x_i=1|\omega_1]$ then $P(x|\omega_1)$ is the product of the p_i

- Most useful features are the ones for which the difference between the means is large relative to the standard deviation
  - Doesn't require independence
- Adding independent features helps increase *r* → reduce error
- Caution: adding features increases cost & complexity of feature extractor and classifier
- It has frequently been observed in practice that, beyond a certain point, the inclusion of additional features leads to worse rather than better performance:
  - we have the wrong model !
  - we don't have enough training data to support the additional dimensions

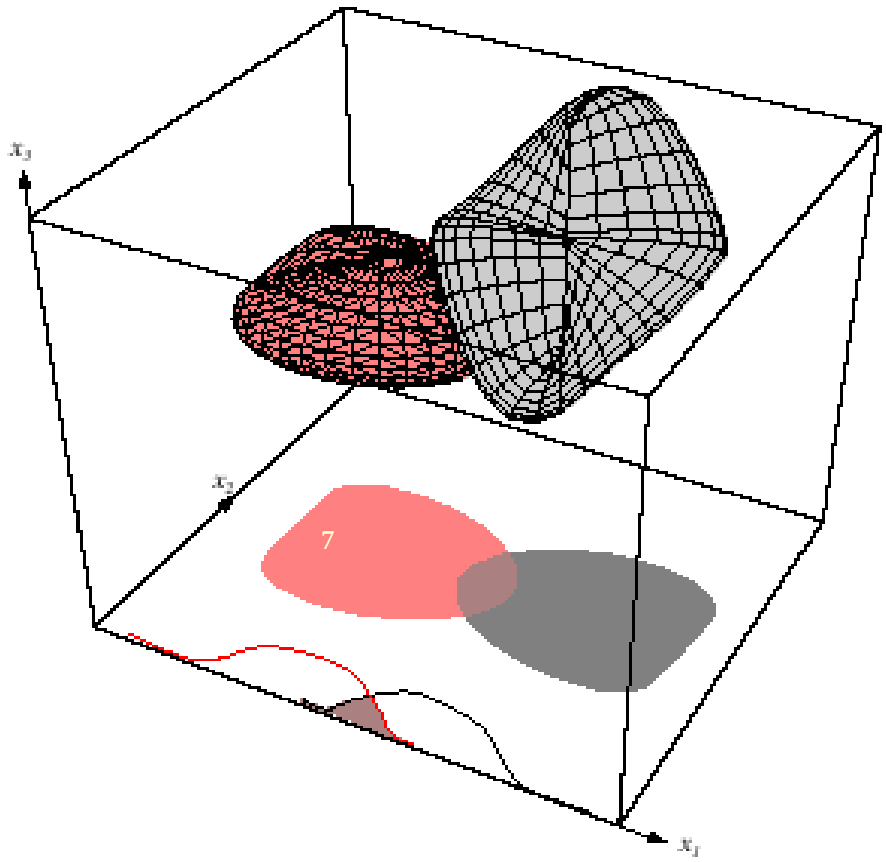


FIGURE 3.3. Two three-dimensional distributions have nonoverlapping densities, and thus in three dimensions the Bayes error vanishes. When projected to a subspace—here, the two-dimensional $x_1 - x_2$ subspace or a one-dimensional x_1 subspace—there can be greater overlap of the projected distributions, and hence greater Bayes error. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

- Computational Complexity

  - Our design methodology is affected by the computational difficulty

    - "big oh" notation

      f(x) = O(h(x)) "big oh of h(x)"

      If: $\exists (c_0, x_0) \in \Re^2;\ |f(x)| \le c_0 |h(x)|$

      (An upper bound on f(x) grows no worse than h(x) for sufficiently large x!)

      $f(x) = 2+3x+4x^2$

      $g(x) = x^2$

      $f(x) = O(x^2)$

- "big oh" is not unique!

  $f(x) = O(x^2)$; $f(x) = O(x^3)$; $f(x) = O(x^4)$

- "big theta" notation

  $f(x) = \theta(h(x))$

  If: $\exists (x_0, c_1, c_2) \in \Re^3; \forall x > x_0$

  $0 \leq c_1 g(x) \leq f(x) \leq c_2 g(x)$

  $f(x) = \theta(x^2)$ but $f(x) \neq \theta(x^3)$

- Complexity of the ML Estimation

  - Gaussian priors in *d* dimensions classifier with *n* training samples for each of *c* classes

  - For each category, we have to compute the discriminant function

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

  Total = O(d^2n)

  Total for *c* classes = O(cd^2n) ≅ O(d^2n)

  - Cost increase when *d* and *n* are large!

Overfitting

- Dimensionality of model vs size of training data
  - Issue: not enough data to support the model
  - Possible solutions:
    - Reduce model dimensionality
    - Make (possibly incorrect) assumptions to better estimate Σ

Overfitting

- Estimate better Σ
  - use data pooled from all classes
    - normalization issues
  - use pseudo-Bayesian form $\lambda\Sigma_0 + (1-\lambda)\Sigma_n$
  - "doctor" Σ by thresholding entries
    - reduces chance correlations
  - assume statistical independence
    - zero all off-diagonal elements

Shrinkage

- Shrinkage: weighted combination of common and individual covariances

$$\boldsymbol{\Sigma}_i(\alpha) = \frac{(1-\alpha)n_i\boldsymbol{\Sigma}_i + \alpha n\boldsymbol{\Sigma}}{(1-\alpha)n_i + \alpha n} \quad \text{for } 0 < \alpha < 1$$

- We can also shrink the estimate common covariances toward the identity matrix

$$\boldsymbol{\Sigma}(\beta) = (1-\beta)\boldsymbol{\Sigma} + \beta\mathbf{I} \quad \text{for} \quad 0 < \beta < 1$$

- Component Analysis and Discriminants

  - Combine features in order to reduce the dimension of the feature space
  - Linear combinations are simple to compute and tractable
  - Project high dimensional data onto a lower dimensional space
  - Two classical approaches for finding “optimal” linear transformation

    - PCA (Principal Component Analysis) “Projection that best represents the data in a least- square sense”
    - MDA (Multiple Discriminant Analysis) “Projection that best separates the data in a least-squares sense”

PCA (from notes)

- Hidden Markov Models:
  - Markov Chains
  - Goal: make a sequence of decisions
    - Processes that unfold in time, states at time t are influenced by a state at time t-1
    - Applications: speech recognition, gesture recognition, parts of speech tagging and DNA sequencing,
    - Any temporal process without memory
      $\omega^T = \{\omega(1), \omega(2), \omega(3), \ldots, \omega(T)\}$ sequence of states
      We might have $\omega^6 = \{\omega 1, \omega 4, \omega 2, \omega 2, \omega 1, \omega 4\}$
    - The system can revisit a state at different steps and not every state need to be visited

- First-order Markov models
  - Our productions of any sequence is described by the transition probabilities

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

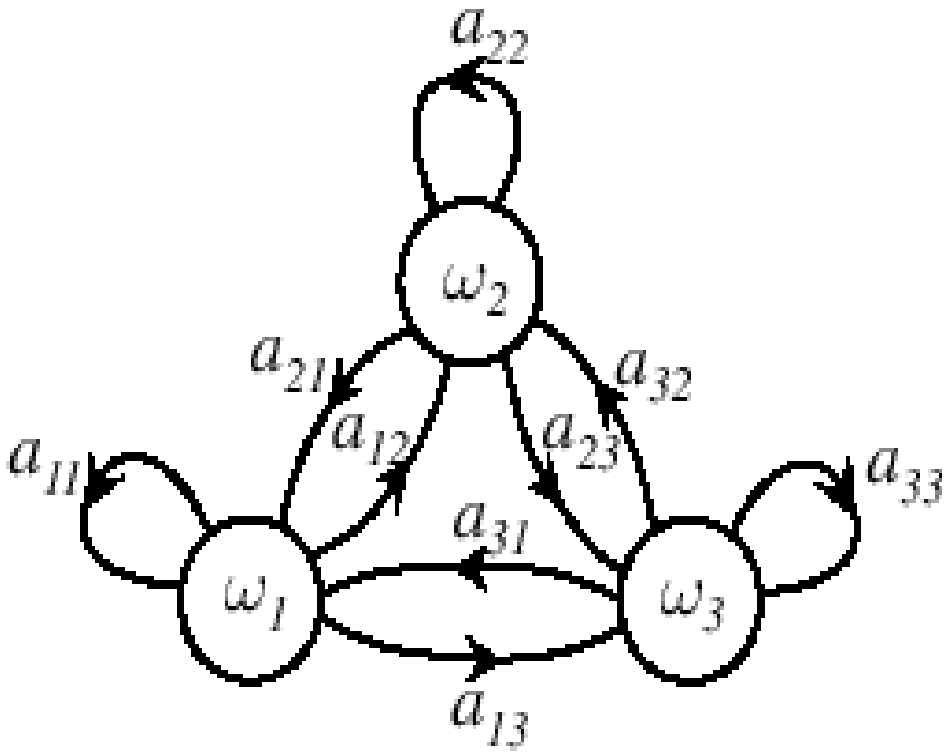


FIGURE 3.8. The discrete states, ω_i, in a basic Markov model are represented by nodes, and the transition probabilities, a_{ij}, are represented by links. In a first-order discrete-time Markov model, at any step t the full system is in a particular state $\omega(t)$. The state at step $t+1$ is a random 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.

$\theta = (a_{ij}, \omega^T)$

$P(\omega^T \mid \theta) = a_{14} \cdot a_{42} \cdot a_{22} \cdot a_{21} \cdot a_{14} \cdot P(\omega(1) = \omega_i)$

Example: speech recognition

"production of spoken words"

Production of the word: "pattern" represented by phonemes

/p/ /a/ /tt/ /er/ /n/ // (// = silent state)

Transitions from /p/ to /a/, /a/ to /tt/, /tt/ to er/, /er/ to /n/ and /n/ to a silent state