Chapter 3 Maximum-Likelihood and Bayesian Parameter Estimation (7,10)

- Problems of Dimensionality
- Computational Complexity
- Hidden Markov Models

3.7 Problems of Dimensionality

- Features of entries of the data (samples) are statistically independent.
  - Classification accuracy depends upon the dimensionality and the amount of training data
  - Case of two classes: the likehood function is multivariate normal with the same covariance
  - The two classes have a same prior.

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

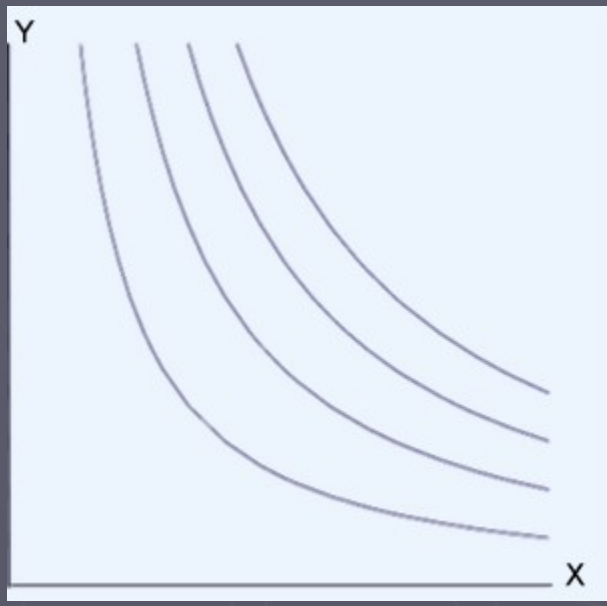
$$where: \quad r^2 = (\mu_1 - \mu_2)^t \Sigma^{-1} (\mu_1 - \mu_2)$$

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

- If features are independent then:

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

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



- Most useful features are the ones for which the difference between the means is large relative to the standard deviation
- It has frequently been observed in practice that, beyond a certain point, the inclusion of additional features leads to worse rather than better performance: this is owing to complex factors.

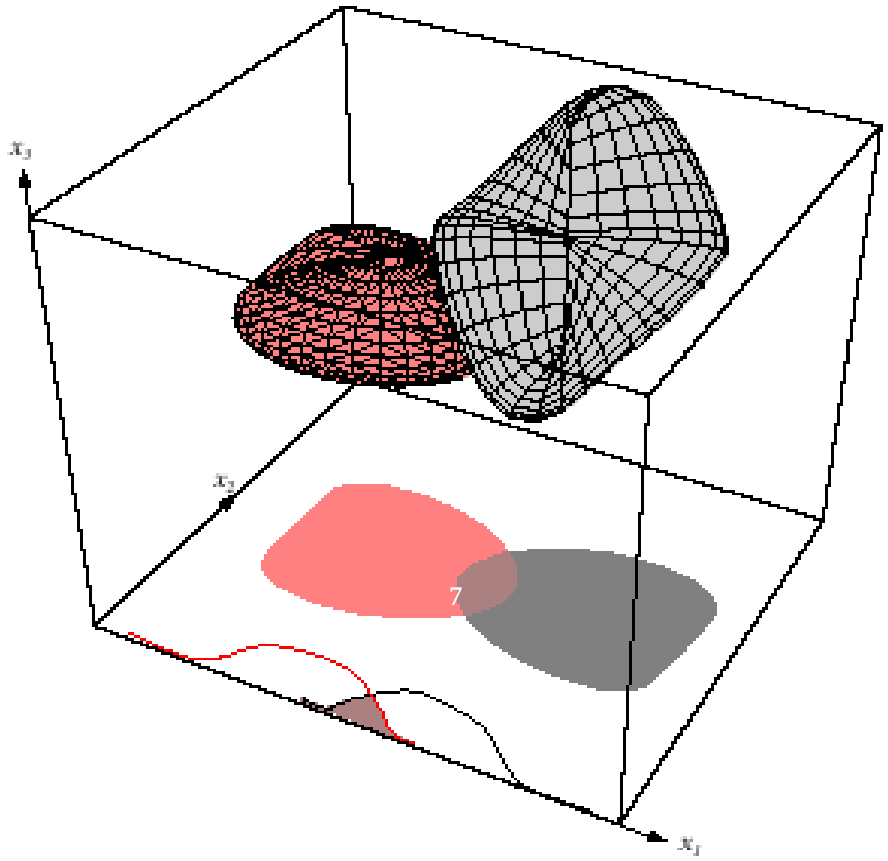
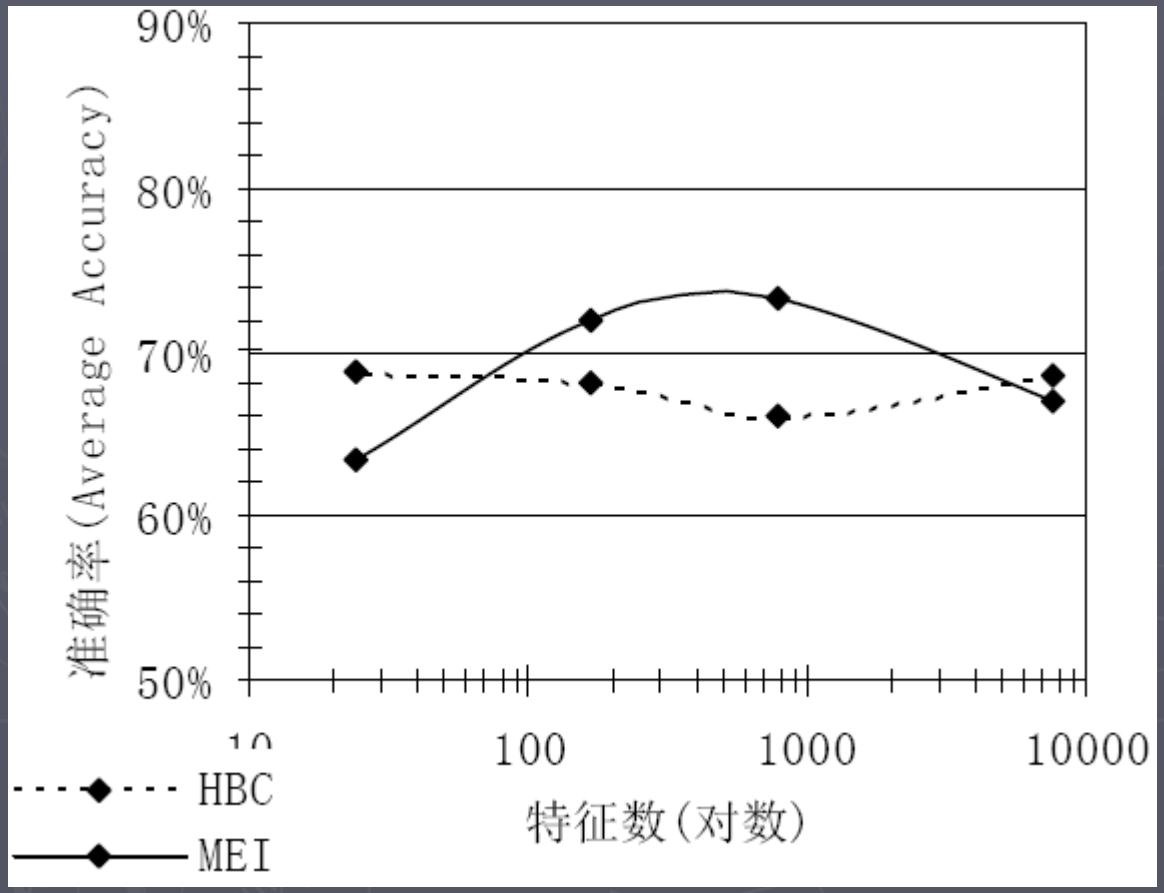


FIGURE 3.3. Two three-dimensional distributions have nonoverlapping densities, and thus in three dimensions the Bayes error vanishes. When projected to a subspace—here, the two-dimensional $x_1 - x_2$ subspace or a one-dimensional x_1 subspace—there can be greater overlap of the projected distributions, 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.

One example of document clustering



Too many features may be harmful

Way to reduce features: feature selection

Google "Feature Selection"

学术搜索 找到约 255,000 条结果 （用时0.16秒）

时间不限
2013以来
2012以来
2009以来
自定义范围...

按相关性排序
按日期排序

搜索所有网页
中文网页
简体中文网页

☑ 包括专利
☑ 包含引用

✉ 创建快讯

小提示： 只搜索**中文(简体)**结果，可在 学术搜索设置. 指定搜索语言

[PDF] A comparative study on feature selection in text categoriza
Y Yang, JO Pedersen - ICML, 1997 - faculty.cs.byu.edu
Abstract This paper is a comparative study of feature selection methods in statistical learning of text categorization. The focus is on aggressive dimensionality reduction. Five methods were evaluated, including term selection based on document frequency (DF), information ...
被引用次数：4066 相关文章 所有 32 个版本 引用 更多

An introduction to variable and feature selection
I Guyon, A Elisseeff - The Journal of Machine Learning Research, 2003 - dl.acm.org
Abstract Variable and feature selection have become the focus of much research in areas of application for which datasets with tens or hundreds of thousands of variables are available. These areas include text processing of internet documents, gene expression array ...
被引用次数：4749 相关文章 所有 119 个版本 引用

The feature selection problem: Traditional methods and a new algorithm
K Kira, LA Rendell - AAAI, 1992 - aaai.org
For real-world concept learning problems, feature selection is important to speed up learning and to improve concept quality. We review and analyze past approaches to feature selection and note their strengths and weaknesses. We then introduce and theoretically ...
被引用次数：928 相关文章 所有 2 个版本 引用

Floating search methods in feature selection
P Pudil, J Novovičová, J Kittler - Pattern recognition letters, 1994 - Elsevier
Abstract Sequential search methods characterized by a dynamically changing number of features included or eliminated at each step, henceforth "floating" methods, are presented.

- 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:

$$\exists (c_0, x_0) \in \Re^2;\ |f(x)| \le c_0 |h(x)| \text{ for all } x > x_0$$

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

      $g(x) = x^2$

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

- "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:

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

$$0 \le c_1 h(x) \le f(x) \le c_2 h(x)$$

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

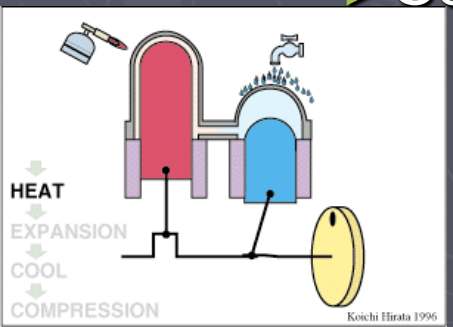
- Complexity of the ML Estimation

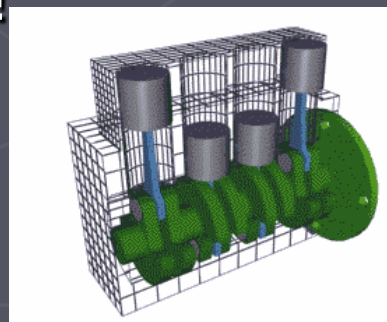
  - Gaussian priors in d dimensions classifier with n training samples for each of c classes
  - For each category, we have to compute the discriminant function

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

  Total = $O(d^2.n)$

  - Total for c classes = $O(cd^2.n) \cong O(d^2.n)$
  - Cost increase when d and n are large!





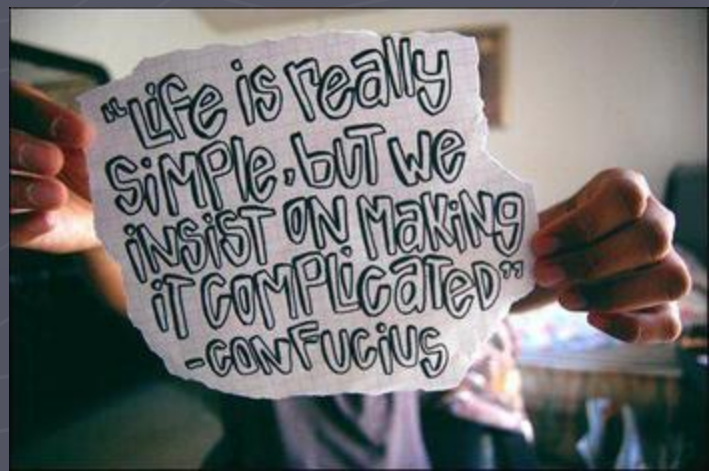
Overfitting

- Samples are inadequate
  - Reduce dimensionality (select a subset of features or combine features)
  - All c classes share the same covariance matrix (can better evaluate the covariance)
  - Look for a better estimate for covariance matrix

  Pseudo-Bayesian estimation:

$$\lambda\Sigma_0 + (1-\lambda)\hat{\Sigma}$$

  - An extreme case: statistical independence
- How to get better performance if statistical dependence
  - Sufficient data
  - Prevent overfitting



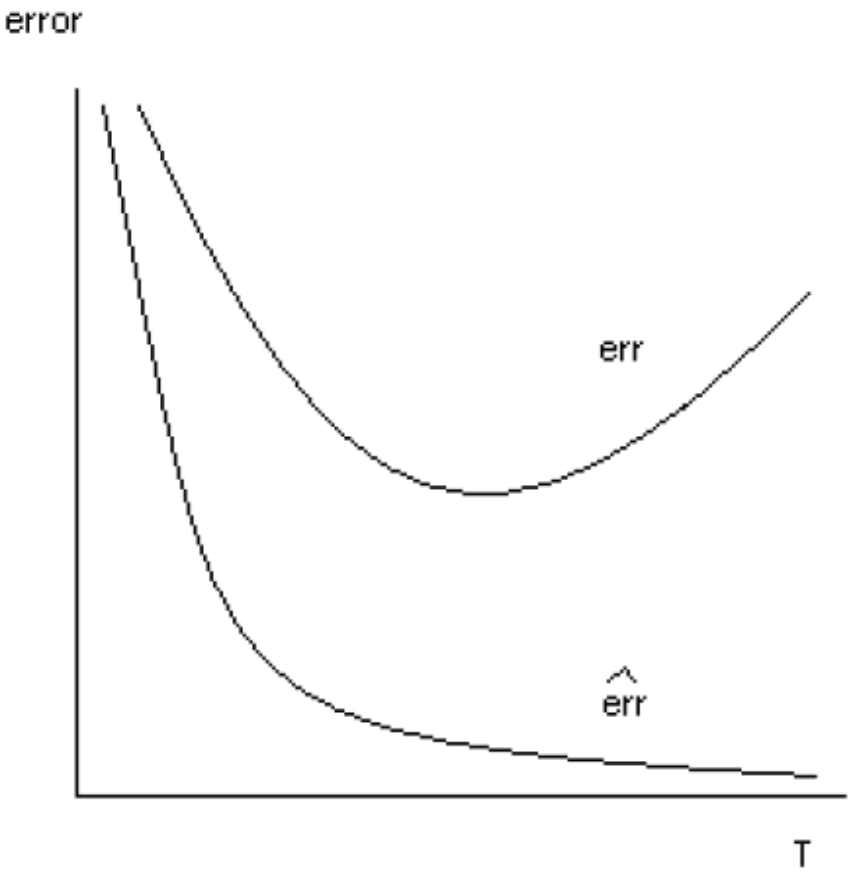
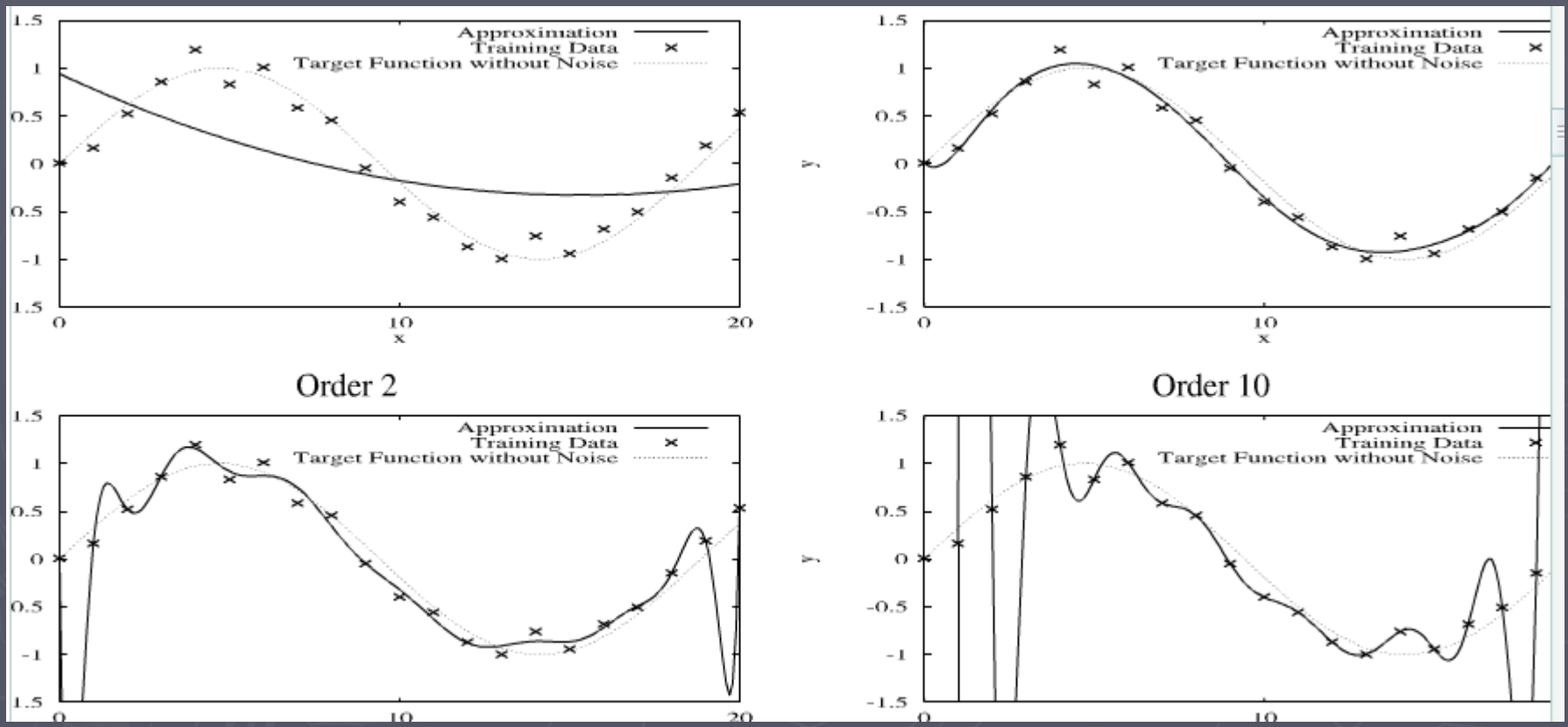


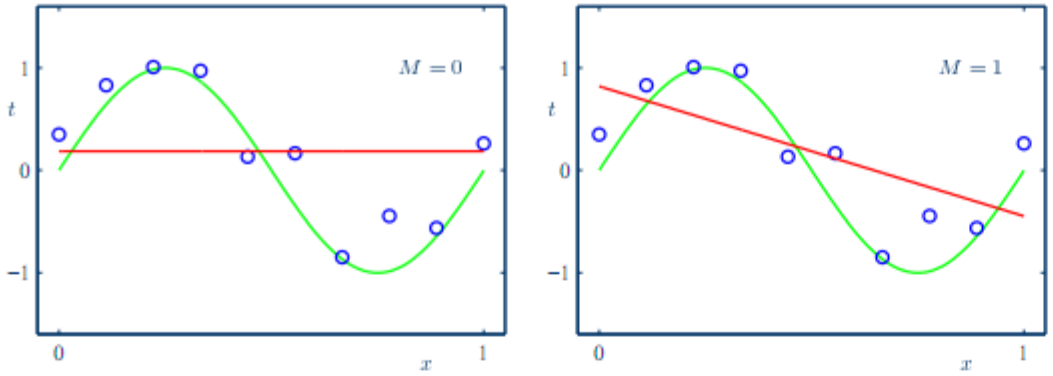
Figure 1: Expected generalization behavior due to overfitting.

Possible overfitting in neural networks

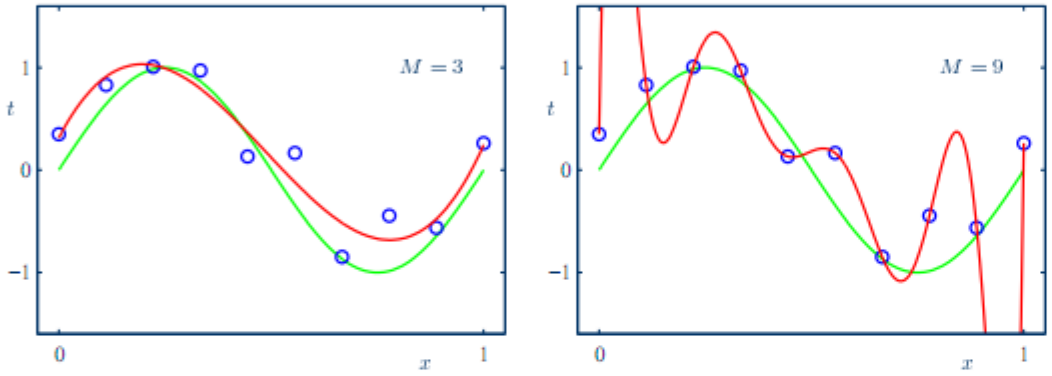


- Another example: Page 93 (In Chinese edition)

Classifiers with proper low complexity is favored



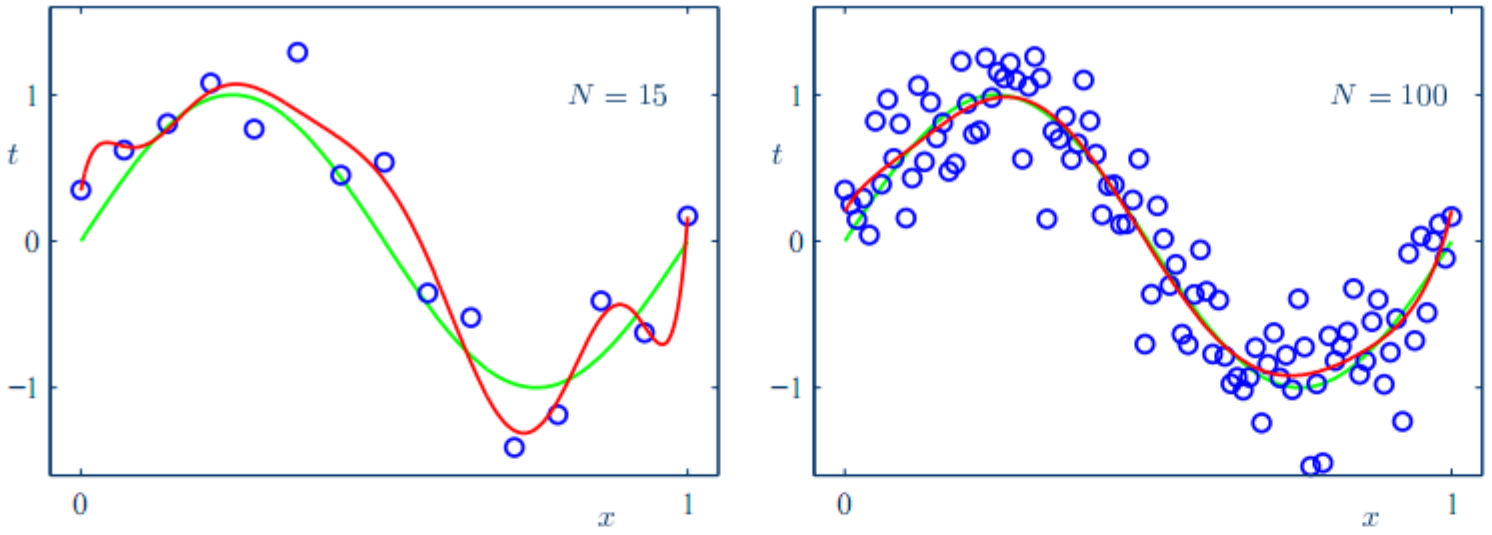
(a) 0'th order polynomial (b) 1'st order polynomial



(c) 3'rd order polynomial (d) 9'th order polynomial

Figure 19: Polynomial curve fitting: plots of polynomials having various orders, shown as red curves, fitted to the set of 10 sample points.

More training samples are better

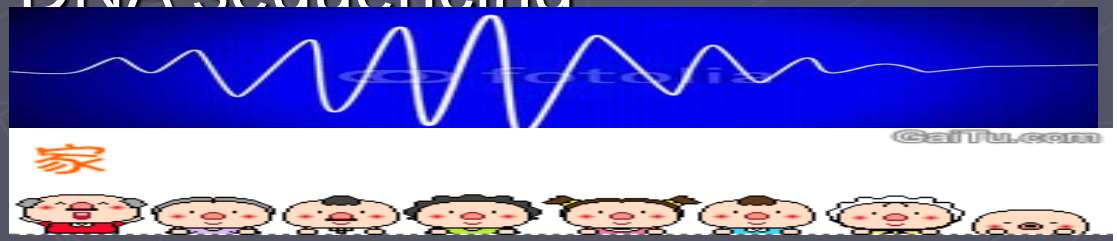


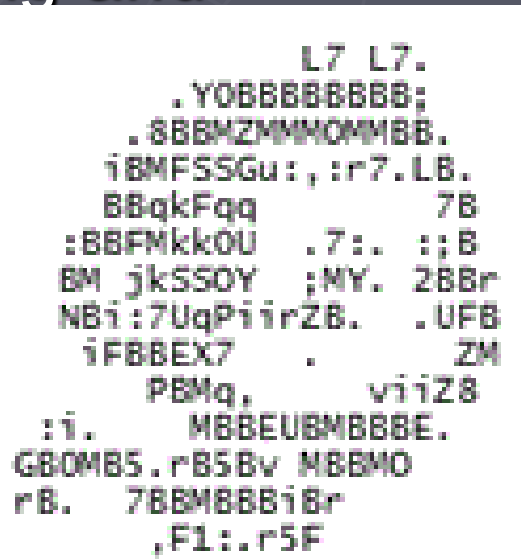
(a) 15 sample points (b) 100 sample points

Figure 20: Polynomial curve fitting: plots of 9'th order polynomials fitted to 15 and 100 sample points.

3.10 Hidden Markov Models

- Goal: make a sequence of decisions
  - A process 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





- First-order Markov models
  - A sequence of states at successive times

    $\omega^T = \{\omega(1), \omega(2), \omega(3), \ldots, \omega(T)\}$

    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
  - Our productions of any sequence is described by the transition probabilities-time independent

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

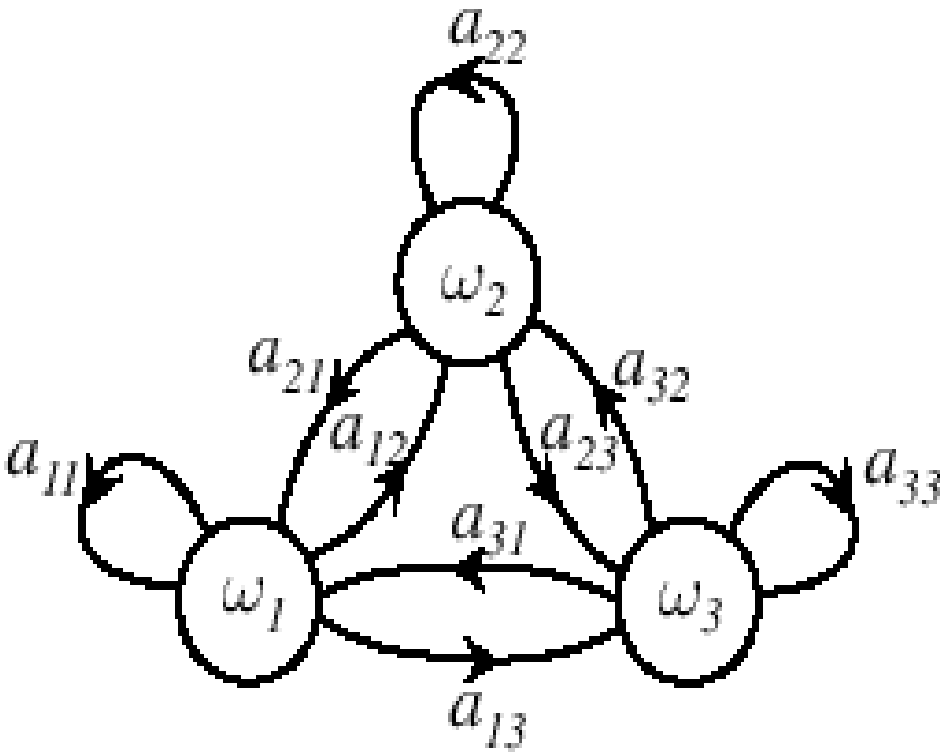


FIGURE 3.8. The discrete states, ω_i, in a basic Markov model are represented by nodes, and the transition probabilities, a_{ij}, are represented by links. In a first-order discrete-time Markov model, at any step t the full system is in a particular state $\omega(t)$. The state at step $t+1$ is a random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

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

  $\omega^T = \omega^6 = \{\omega 1, \omega 4, \omega 2, \omega 2, \omega 1, \omega 4\}$

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

  $P(\omega(1) = \omega_i)$
- First-order Markov Model: the probability at t+1 depends only on the sates at t
- 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

- First-Order Hidden Markov Models (HMM)
  - A state $\omega(t)$ *emits some visible symbol* $v(t)$ *, the sequence of such visible symbol is :*

    $V^T = \{v(1), v(2), v(3), \ldots, v(T)\}$
  - In any state $\omega_j(t)$ *we have a probability of emitting a particular visible state* $v_k(t)$ *,* ω_j *are unobservable, such a full model is HMM*
  - *In HMM* a_{ij} *is the transition probabilities among hidden states and* b_{jk} *is the probability of the emission of a visible state.*

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

    $\sum a_{ij} = 1$ *for all* i

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

    $\sum b_{jk} = 1$ *for all* j

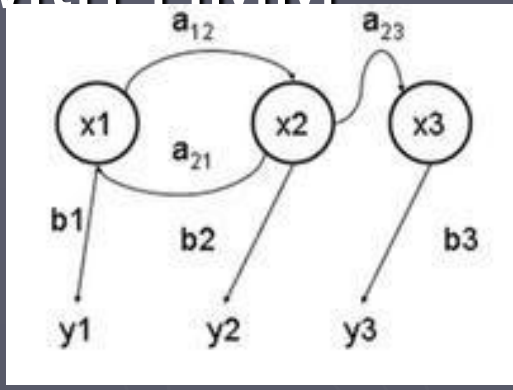
张学良将军在一次接受记者采访的时候,讲起他小时候学英语的一段趣事:我父亲很想给我请英文教师,英文教师是外交署一个英文科长,这个人我很想念他,他是香港新约书院的。他是广东人,说广东国语。我跟你说个笑话,nine,就是九,他说九(狗),我听说是狗,他说九,我当说狗。那时候

- Three problems are associated with HMM
  - The evaluation problem

    a_{ij}, b_{jk} ⟶ probability of V^T
  - The decoding problem

    V^T ⟶ ω^T （analogy: listen and write） P 108
  - The learning problem

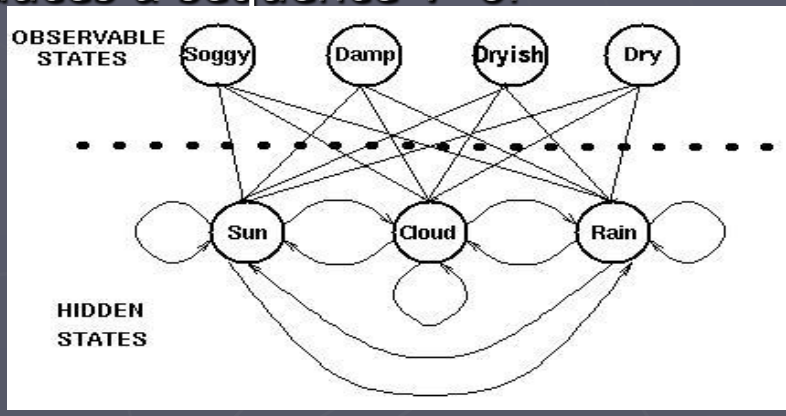
    V^T ⟶ a_{ij}, b_{jk}



1. 一只狗（九）在狂吠
2. 那儿有九（狗）个人

The evaluation problem

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



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

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

hidden states.

$$(1) \qquad P(V^T \mid \omega_r^T) = \prod_{t=1}^{t=T} P(v(t) \mid \omega_r(t))$$

$$(2) \qquad \mathrm{P}(\omega_{\mathrm{r}}^{\mathrm{T}}) = \prod_{t=1}^{t=T} P(\omega_r(t) \mid \omega_r(t-1))$$

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

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

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.

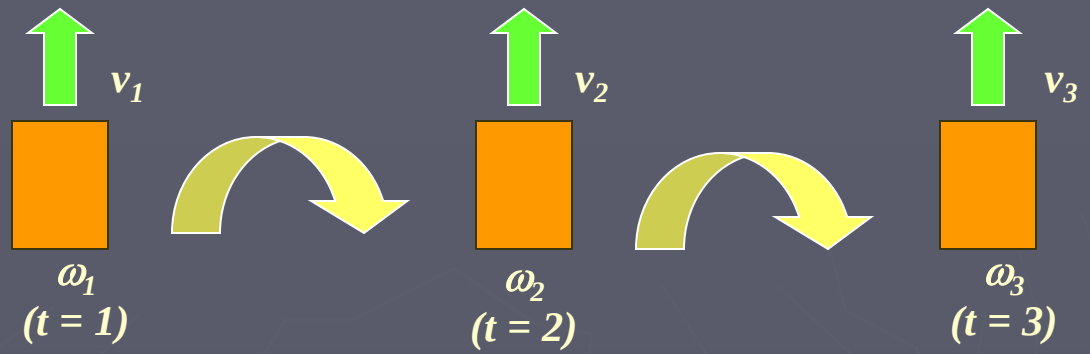
Example: Let $\omega_1, \omega_2, \omega_3$ be the hidden states; v_1, v_2, v_3 be the visible states

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

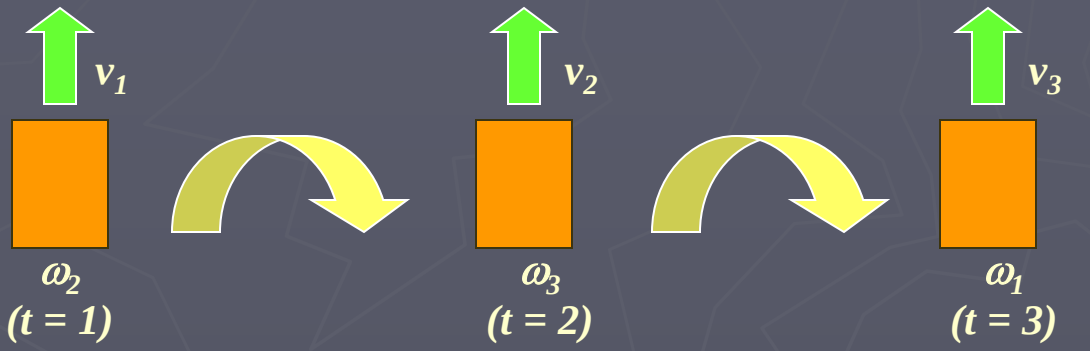
$P(\{v_1, v_2, v_3\}) = P(\omega_1).P(v_1 \mid \omega_1).P(\omega_2 \mid \omega_1).P(v_2 \mid \omega_2).P(\omega_3 \mid \omega_2).P(v_3 \mid \omega_3)$

$+ \ldots +$ (possible terms in the sum= all possible (3^3= 27) cases !)

First case:



Second case :



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

$P(\omega_2).P(v_1 \mid \omega_2).P(\omega_3 \mid \omega_2).P(v_2 \mid \omega_3).P(\omega_1 \mid \omega_3).P(v_3 \mid \omega_1) + \ldots +$

Therefore:

$$P(\{v_1, v_2, v_3\}) = \sum_{\substack{\textit{possible}\ \text{sequence} \\ \text{r of hidden states}}} \prod_{t=1}^{t=3} P(v(t) \mid \omega_r(t)).P(\omega_r(t) \mid \omega_r(t-1))$$

- HMM Forward algorithm

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

b_{jk}(v(t)): emission probability b_{jk} selected by the visible state at time t with visible state v(t).

$\alpha_j(t)$: probability that HMM is in hidden state ω_j *at step t having generated the first t elements of* V^T

- HMM Forward algorithm

  Initialize t=0, a_{ij}, b_{jk}, V^T, $\alpha_j(0)$

  for t=t+1

$$\alpha_j(t) \leftarrow b_{jk} v(t) \Sigma_{i=1}^{c} \alpha_i(t-1) a_{ij}$$

  Until t=T

  return $P(V^T) \leftarrow \alpha_0(T)$ for the final state

  end

- A left-to-right HMM

►The decoding problem (optimal state sequence)

Given a sequence of visible states V^T, the decoding problem is to find the most probable sequence of hidden states.

This problem can be expressed mathematically as:

find the single "best" state sequence (hidden states)

$$\hat{\omega}(1),\hat{\omega}(2),...,\hat{\omega}(T) \text{ such that :}$$

$$\hat{\omega}(1),\hat{\omega}(2),...,\hat{\omega}(T) = \underset{\omega(1),\omega(2),...,\omega(T)}{\arg\max} P[\omega(1),\omega(2),...,\omega(T),v(1),v(2),...,V(T)|\lambda]$$

Note that the summation disappeared, since we want to find Only one unique best case !

Where: $\lambda = [\pi, A, B]$

$\pi = P(\omega(1) = \omega)$ *(initial state probability)*

$A = a_{ij} = P(\omega(t+1) = j \mid \omega(t) = i)$

$B = b_{jk} = P(v(t) = k \mid \omega(t) = j)$

In the preceding example, this computation corresponds to the selection of the best path amongst:

$\{\omega_1(t = 1), \omega_2(t = 2), \omega_3(t = 3)\}$, $\{\omega_2(t = 1), \omega_3(t = 2), \omega_1(t = 3)\}$

$\{\omega_3(t = 1), \omega_1(t = 2), \omega_2(t = 3)\}$, $\{\omega_3(t = 1), \omega_2(t = 2), \omega_1(t = 3)\}$

$\{\omega_2(t = 1), \omega_1(t = 2), \omega_3(t = 3)\}$……

HMM decoding algorithm

begin initialize $\text{Path} \leftarrow \{\}, \text{t} \leftarrow 0$

for $\text{t} \leftarrow \text{t}+1$

$\text{j} \leftarrow 1$

for $\text{j} \leftarrow \text{j}+1$

$$\alpha_{\text{j}}(\text{t}) \leftarrow \text{b}_{\text{jk}}\,\text{v}\,(\text{t})\sum_{\text{i}=1}^{\text{c}} \alpha_{\text{i}}(t-1)a_{ij}$$

until $\text{j} = \text{c}$

$\text{j}' \leftarrow \arg\max_{\text{j}} \alpha_{\text{j}}$

Append $\omega_{\text{j}'}$ to Path

until $\text{t} = \text{T}$

return Path

end

►The learning problem (parameter estimation)

This third problem consists of determining a method to adjust the model parameters $\lambda = [\pi, A, B]$ to satisfy a certain optimization criterion. We need to find the best model

$$\hat{\lambda} = [\hat{\pi}, \hat{A}, \hat{B}]$$

Such that to maximize the probability of the observation sequence:

$$\max_{\lambda} P(V^T \mid \lambda)$$

We use an iterative procedure such as Baum-Welch or Gradient to find this local optimum

- **Parameter Updates:**

Forward-Backward Algorithm

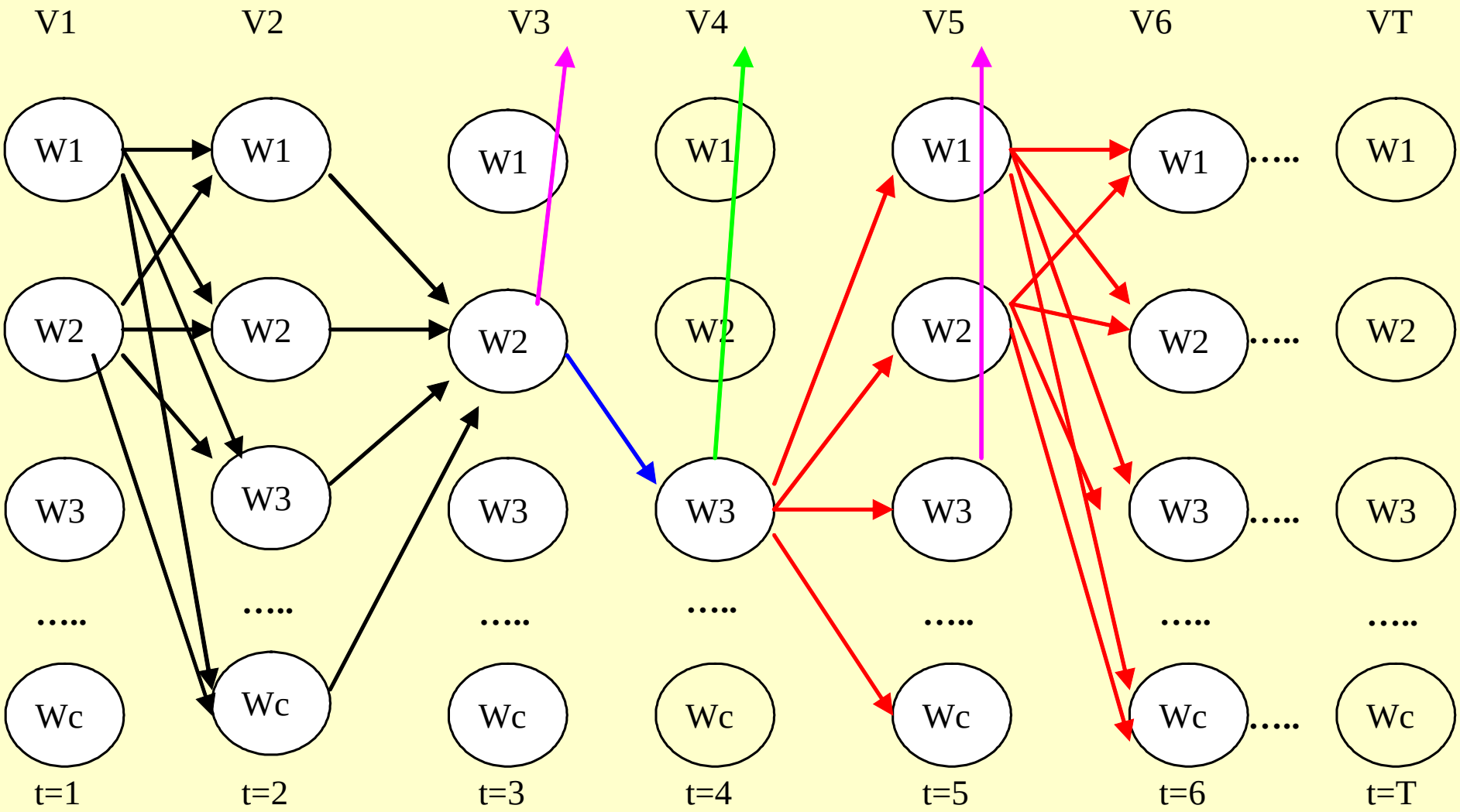
$$\beta_i(t) = \begin{cases} 0 & \text{t = T and } \omega_i(t) \neq \omega_0 \\ 1 & \text{t = T and } \omega_i(t) \neq \omega_0 \\ [\Sigma_j \beta_j(t+1) a_{ij}] b_{jk} v(t+1) & \text{otherwise} \end{cases}$$

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

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

$$\hat{b}_{jk} = \frac{\sum_{\substack{t=1 \\ v(t)=v_k}}^{T} \sum_l \gamma_{jl}(t)}{\sum_{t=1}^{T} \sum_l \gamma_{jl}(t)}$$

- $\alpha_i(t)$= P(model generates visible sequence up to step t given hidden state $\omega_i(t)$)
- $\beta_i(t)$= P(model will generate the sequence from t+1 to T given $\omega_i(t)$)
- $\gamma_{ij}(t)$ is the probability from $\omega_i(t-1)$ to $\omega_j(t)$



$$\gamma_{23}(4) = \frac{\alpha_2(3) a_{23} b_{34} \beta_3(4)}{p(V^T \mid \theta)}$$

Parameters Learning Algorithm

Begin initialize

a_{ij}, b_{jk}, training sequence V^T,
convergence criterion (cc), z=0

Do z=z+1

compute $\hat{a}(z)$ from a(z-1) and b(z-1)

compute $\hat{b}(z)$ from a(z-1) and b(z-1)

$a_{ij}(z)=\hat{a}_{ij}(z-1)$

$b_{jk}(z)=\hat{b}_{jk}(z-1)$

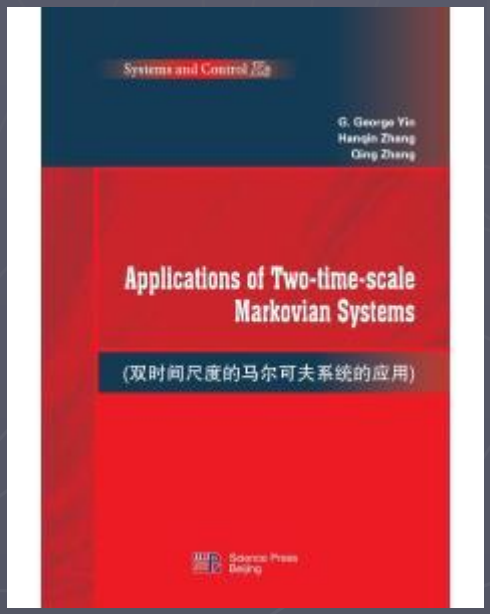
Until $\max\{a_{ij}(z)-a_{ij}(z-1), b_{jk}(z)-b_{jk}(z-1)\} < cc$

Return $a_{ij}=a_{ij}(z)$; $b_{jk}=b_{jk}(z)$

End


Application of
Wavelet Domain Markov Random Field to Image Processing
小波域马尔可夫随机场
在图像处理中的应用
李旭超 著


An Introduction
to
Markov Decision Processes
马尔可夫
决策过程引论


Systems and Control
G. George Yin
Hongjin Zhang
Qing Zhang
Applications of Two-time-scale
Markovian Systems
(双时间尺度的马尔可夫系统的应用)
Science Press
Beijing