

[illegible]

清华大学出版社

计算机图像处理技术与算法

陈 纯 编著

清华大学出版社

北 京

内 容 简 介

本书是取材于数字图像处理领域几十年的发展成果,包括传统的灰度图像处理和最新的彩色图像处理内容,结合多年的教学、科研和实践经验编写而成。

本书既全面介绍了数字图像处理的理论系统,又强调它的实践价值。首先介绍了隐式马尔可夫模型、预测模型、采样理论、图像变换等基本理论知识,然后逐章介绍图像的编码和文件格式,图像的增强、恢复和分割等内容,最后两章介绍了彩色图像处理的基本内容和基于内容的图像检索技术。

本书可作为高校计算机、生医工程等有关图像处理专业的高年级学生和研究生教材或者作为从事图像处理相关领域科研人员的参考书。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

计算机图像处理技术与算法/陈纯编著. —北京:清华大学出版社,2003

ISBN 7-302-06598-5

I. 计... II. 陈... III. 计算机应用—图像处理 IV. TP391.41

中国版本图书馆CIP数据核字(2003)第032617号

出 版 者: 清华大学出版社(北京清华大学学研大厦,邮编 100084)

[http:// www.tup.tsinghua.edu.cn](http://www.tup.tsinghua.edu.cn)

责任编辑: 苗建强

印 刷 者: 北京鑫海金澳胶印有限公司

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 **印张:** 12.5 **字数:** 283 千字

版 次: 2003 年 7 月第 1 版 2003 年 7 月第 1 次印刷

书 号: ISBN 7-302-06598-5/TP·4941

印 数: 0001~4000

定 价: 18.00 元

前 言

数字图像处理是一门高速发展的学科。从 20 世纪 50 年代到今天, 这门学科一直致力于利用计算机来进行图像的解析和处理, 并且取得了丰硕的成果, 其应用领域遍及众多行业、各学科。反过来, 在各领域的应用又对图像处理技术提出更高的要求, 推动着这门学科更快地发展。

图像处理的一个典型应用是遥感图像处理中的地理地图绘制、地质勘探、森林资源调查、环境污染监督、海洋检测等。另外在生物医学工程领域, MRI、CT 和 B 超等基于图像处理技术的无损诊断设备已成为医院的常规诊断设备。在普通消费领域, VCD、DVD 等基于图像压缩技术的设备也已成为人们的普通娱乐设施。

一方面, 图像处理技术是一个具有很强理论背景的研究领域, 它是一门综合多学科的成果而发展起来的学科, 需要广泛的基础知识, 包括计算机科学、数字信号处理、随机过程处理和统计数学、矩阵分析、信息论、控制论和最优化理论等。另一方面, 它又是与实际应用紧密结合的学科。一些成熟稳定的图像处理的算法, 已成为当今信息工业标准的一部分, 如从 20 世纪 80 年代开始规划, 至 20 世纪 90 年代发表的 H.263、JPEG、MPEG-1、MPEG-2、MPEG-4 等图像压缩和传输标准。

本书可以作为计算机图像处理的基础教材。它系统地介绍了数字图像处理的基本概念和系统知识, 并重视将图像处理技术应用于实践, 对很多重要算法都有程序实现, 程序代码附于光盘中, 对提高学生的动手能力有很大的帮助。

在内容上, 本书前面几章介绍了传统的图像处理知识, 包括隐式马尔可夫模型、预测模型、采样理论、图像变换等基本的理论知识以及图像的编码和文件格式、图像的增强、恢复和分割等内容。

本书还反映了图像处理的最新发展, 在最后两章中具体讲述了彩色图像处理中的一些典型问题、方法以及基于内容的最新图像检索技术。

本书的完成得到了卜佳俊、王强两位博士的大力帮助, 同时李琥、黄峥等在程序的编写上做了很多的工作, 在此对他们表示衷心的感谢。

陈 纯

2003 年 3 月于求是园

目 录

第 1 章 绪论	1
1.1 图像处理技术的发展历史及现状	1
1.2 图像的数学模型	2
1.2.1 图像的正交模型	3
1.2.2 隐式马尔可夫模型	3
1.2.3 预测模型	10
1.3 图像的采样与亚采样	11
1.3.1 一维连续信号的采样	12
1.3.2 二维连续图像信号的采样	14
1.3.3 图像的亚采样	15
第 2 章 图像变换	17
2.1 正交变换	18
2.1.1 连续函数集合的正交性	18
2.1.2 二维离散正交变换	19
2.2 二维傅立叶变换	20
2.2.1 一维信号的傅立叶级数	20
2.2.2 一维信号的傅立叶变换	20
2.2.3 一维离散信号的频谱	22
2.2.4 一维信号有限离散傅立叶变换	23
2.2.5 二维傅立叶变换	24
2.3 离散 K-L 变换	25
2.3.1 正交变换的物理意义	25
2.3.2 离散 K-L 变换	26
2.4 离散余弦变换	28
2.4.1 离散余弦变换	28
2.4.2 快速 DCT 算法	29
2.5 小波变换	32
2.5.1 短时傅立叶变换	32
2.5.2 连续小波变换	34
2.5.3 离散小波变换	36

2.5.4	多分辨率分析.....	37
2.5.5	小波系数分解的快速算法——Mallat 算法.....	38
2.5.6	二维小波的多分辨率分析及 Mallat 算法.....	41
第 3 章	图像编码及文件格式.....	44
3.1	无损压缩技术.....	45
3.2	有损压缩技术.....	48
3.2.1	图像量化.....	48
3.2.2	预测编码.....	52
3.2.3	变换编码.....	53
3.2.4	分形编码.....	56
3.3	图像文件格式.....	61
3.3.1	BMP 文件格式.....	61
3.3.2	GIF 文件格式.....	64
3.3.3	JPEG 文件格式.....	69
3.3.4	TIFF 文件格式.....	72
第 4 章	图像增强和恢复.....	75
4.1	空域增强法.....	75
4.1.1	灰度变换.....	75
4.1.2	直方图变换.....	77
4.1.3	图像中的脉冲噪声模型.....	79
4.1.4	邻域平均法.....	79
4.1.5	中值滤波.....	81
4.1.6	图像锐化.....	82
4.2	频域增强法.....	86
4.2.1	低通滤波器.....	86
4.2.2	同态滤波.....	87
4.2.3	高通滤波器.....	88
4.3	图像退化模型.....	89
4.4	逆滤波.....	92
4.5	维纳滤波.....	93
4.6	卡尔曼滤波.....	94
第 5 章	图像分割.....	97
5.1	图像描述.....	97
5.1.1	区域边界的描述.....	98
5.1.2	区域的描述.....	102
5.2	灰度阈值分割法.....	104

5.3 基于纹理的分割方法	107
5.3.1 纹理分析的自相关函数方法	107
5.3.2 纹理分割——Hurst 系数	109
5.3.3 灰度共生矩阵的纹理分析	109
5.4 区域生长法	112
5.4.1 区域生长法	113
5.4.2 区域分割与合并	113
5.4.3 Hough 变换	114
5.5 数学形态学	115
5.5.1 基本运算	115
5.5.2 腐蚀和膨胀的衍生运算	117
第 6 章 彩色图像处理	119
6.1 颜色模型	119
6.1.1 RGB 颜色模型	120
6.1.2 CMY 颜色模型	120
6.1.3 HSV 颜色模型	121
6.1.4 CIE (国际照明委员会) 颜色模型	122
6.1.5 照片 YCC (照片 CD) 颜色模型	124
6.1.6 各种颜色模型之间的转换算法	125
6.2 色彩量化技术	127
6.2.1 LBG	127
6.2.2 DSQ	127
6.3 分色技术	128
6.3.1 四色分色技术	128
6.3.2 专色分色技术	131
6.4 色彩调和和技术	134
6.4.1 SD 法	134
6.4.2 Ostwald 色彩调和论	135
6.4.3 Moon&Spencer 色彩调和论	137
6.4.4 基于知识的色彩调和模型	140
6.5 彩色图像处理编程及技巧	142
6.5.1 百叶窗效果	142
6.5.2 Water paper (水彩画纸) 效果	143
6.5.3 ZigZag 效果	144

第7章 基于内容的图像检索	145
7.1 引言.....	145
7.2 基于内容图像检索的内容分析和检索技术.....	146
7.2.1 颜色分析和检索技术.....	147
7.2.2 形状分析和检索技术.....	152
7.2.3 纹理分析和检索技术.....	157
7.2.4 空间关系分析和检索技术.....	163
7.3 知识和语义技术.....	166
7.4 人机交互学习技术.....	169
7.4.1 参数调整方法.....	170
7.4.2 聚类分析方法.....	171
7.4.3 概率学习方法.....	171
7.4.4 神经网络方法.....	173
7.5 相似性度量.....	173
7.5.1 相似心理学.....	173
7.5.2 向量相似性度量函数.....	175
7.5.3 特征集合相似性度量.....	178
7.5.4 EMD (Earth Mover's Distance) 距离.....	179
7.6 数据库中的特征索引技术.....	181
7.7 性能评价指标.....	183
7.8 典型系统介绍.....	184
7.8.1 系统构成.....	184
7.8.2 典型系统简介.....	186
7.9 现状和发展.....	189

第 1 章 绪 论

1.1 图像处理技术的发展历史及现状

人们所接受到的信息 90%以上是通过眼睛看到的图像来获取,这说明图像信息对人有多么重要。随着电子计算机的出现,人们试图用计算机技术来进行对图像的分析处理,包括对视觉、遥感图片等的理解、分析。到 20 世纪 50 年代,计算机数字图像处理技术迅速发展,形成了新的学科。到 20 世纪 80 年代止,图像处理的研究方向主要集中于:用图像变换和数学模型来表征图像信号,图像滤波和图像恢复,静止和视频图像的压缩,计算机硬件、图像获取设备和显示设备的不断改进,为图像处理研究构筑了不断发展的基础。到 20 世纪 80 年代中期,各种高性能的工作站和个人电脑使得图像处理研究和技术不再是大机构或大型学术团体的专利。20 世纪 90 年代,多媒体技术对图像处理技术,包括图像信息的压缩和图像文件的传送等提出了更高的要求。

随着技术的发展,数码相机、数字摄影机、数字电视机等大量数字化设备不断涌现,从一方面讲,这些是图像处理技术应用的成果,反过来,这些设备的广泛应用,也促使图像处理技术不断向前发展,图像数据库和多媒体技术也推动着这一领域不断向前发展。

- 从应用领域看,图像处理技术不再局限于电子工程领域,在地理、医学、刑事侦察、气象等学科领域也广为应用;
- 从可处理的光谱频段看,其处理范围已从可见光谱扩展到各个波段,如遥感图像的多光谱图像处理、雷达波段的侧视雷达遥感、红外光波段的图像处理(如夜视仪、热像仪)、CT 和 MRI 图像处理;
- 从图像处理技术本身来看,研究的技术领域已包括纹理和图形形状分析与识别、运动检测与估计、图像处理并行系统、图像处理技术的软硬件设计等。

图形处理技术是一个具有很强理论背景的研究领域,是基于线性代数、数理统计和物理学等理论的学科,但实质上又与实际应用紧密结合。一些稳定的图像处理算法,已经成功地应用到消费产品中。在研究人员的努力之下,随着成熟的图像处理算法的广泛应用,其中一些已成为工业标准,如从 20 世纪 80 年代开始规划,至 20 世纪 90 年代发表的 H263、JPEG、MPEG-1、MPEG-2、MPEG-4 等图像压缩和传输标准,使得图像处理技术

在信息工业成为不可或缺的一部分，并在人们日常生活中广为应用。近年来，图像的小波处理技术已经成熟，随之提出了 JPEG-2000 标准，以小波变换代替原来的 DCT 变换，因为小波变换具有时频局部性，而傅立叶变换不具有这个优点，而且研究人员提出了和傅立叶变换一样的快速算法。

图像处理技术随着理论基础的更新而不断发展。在早期，图像压缩技术主要依赖于传统的信息论，如 Huffman 编码、LZW 压缩算法，这类无损压缩的压缩比一般为 1 到 2 倍左右；但是对于有损压缩，如 DCT 变换，则压缩比可以达到 30 倍左右，这些图像压缩技术是基于像素，并没有充分利用图像本身的信息。而新一代图像压缩技术则注重图像信号的内部特征，充分利用这些特征和人类视觉系统的特点来进行高重构图像质量、高压缩比的压缩。如图像分形压缩技术充分利用图像内部的自相似性，利用一个迭代函数来产生分形图像以达到图像压缩的目的。新一代图像压缩技术中，一种重要的方法是图像子带编码。图像子带编码的基本思想是将图像信号的频带用一组分析滤波器分割成各个子带信号，针对每个不同的子带信号，按照它的统计特性分配不同的编码器和比特率。这样做的好处在于压缩误差仅仅局限于各个子带信号中，互不影响，并且可以根据人眼视觉系统频率区域敏感的特征，给各个不同的子带信号分配不同的比特率，不但可以获得高压缩比，而且重构图像的主观视觉效果比较好。其实小波变换，金字塔多尺度分割算法都可以归属于图像子带编码技术。

我们在前面只简单介绍了与图像处理技术有关的几种新方法和新技术。20 世纪 90 年代，随着信息社会的到来，图像处理技术进入一个更加迅猛发展的阶段。特别是多媒体技术、通信技术、信息存储技术和国际因特网（Internet）为代表的计算机网络技术的加速发展和广泛普及以及高清晰度电视（HDTV）的深入应用研究，更加推广了图像处理技术的研究与发展。

1.2 图像的数学模型

研究者们一直试图用数学模型来完整地描述图像，从而解析地表达图像分布的情况，然后用数学工具来对图像进行研究。但是目前还没有一种图像的数学模型能完整地描述所有图像的信息分布的重要特征。每一种模型刻画了图像分布的某一类特征，不同的图像处理应用领域有不同的数学模型。

按信号在时间轴上的连续性，图像的数学模型可分为连续模型和离散模型；按信号的统计特性来分类，可以分为平稳和非平稳的数学模型；按模型的数学特性来分，则可分为

确定性模型和统计模型。以下介绍确定性模型和统计模型。确定性模型是假定图像可以用解析数学表达来描述,经常用矩阵形式来表示。而统计模型则刻画一些信号的统计特征,如高斯模型、自回归模型、隐式马尔可夫模型等。

以下简单介绍图像处理中常用的几种数学模型。

1.2.1 图像的正交模型

正交模型是图像处理中的最为常用的数学工具。图像的行与行之间、列与列之间、分块与分块之间有很强的相关性,可以把图像信号看作是一个很复杂的数学函数。例如,在图像分析和处理中,常把一幅图像展开成一组完备的正交基函数之和,这就是所谓的酉变换(Unitary Transform)。

关于正交模型的详细内容将在第2章进行讨论。

1.2.2 隐式马尔可夫模型

隐式马尔可夫模型(Hidden Markov Model, HMM)是图像处理中应用较多的模型,在图像分割、模式识别和图像恢复中都用到这个模型。隐式马尔可夫模型较为复杂,先介绍离散的马尔可夫过程,也即马尔可夫链。

1.2.2.1 离散马尔可夫过程

假设系统有 N 个状态, $s_1, s_2, \dots, s_N$, 系统在任何时刻必处在这 N 个状态的一个, 系统在 t 时刻的真实状态为 q_t , 针对离散、一阶马尔可夫链: 系统在某一时刻处于某一特殊状态的概率只由系统前一时刻的状态所决定, 即

$$P[q_t = s_j | q_{t-1} = s_i, q_{t-2} = s_k, \dots] = P[q_t = s_j | q_{t-1} = s_i] \quad (1.1)$$

考虑式(1.1)右边的值与右边无关的情况, 得到一组状态转移概率:

$$a_{ij} = P[q_t = s_j | q_{t-1} = s_i], \quad 1 \leq i, j \leq N \quad (1.2)$$

它们满足以下条件:

$$a_{ij} \geq 0 \quad (1.3)$$

$$\sum_{j=1}^N a_{ij} = 1 \quad (1.4)$$

此模型称为显式马尔可夫模型(Observable Markov Model), 每个状态对应于一个物理事件, 这个模型的输出就是系统在某一时刻所处的状态。在实际应用中我们一般考虑的

是因果模型，即系统有初始状态：

$$\pi_i = P[q_1 = s_i], \quad 1 \leq i, j \leq N \quad (1.5)$$

1.2.2.2 隐式马尔可夫模型

但上述模型过于简单，因此在实际应用中有许多限制。现在考虑一个更复杂的模型，离散马尔可夫模型中系统的输出与系统所处状态一一对应，而现在所要考虑的是系统的输出，即观察值是系统状态的概率函数，这是一个双重嵌套的随机过程。在这种模型中，内在状态序列是不可见的，只能通过系统输出的观察序列来观测。

一个隐式马尔可夫模型所需考虑的基本参数有：

1. N —— 模型状态个数，状态集为 $\{s_1, s_2, \dots, s_N\}$ ， t 时刻系统状态为 q_t 。
2. M —— 每个状态所能产生的观察值个数，在上面所举的例子中，观察值即硬币的正反面。
3. $A = \{a_{ij}\}$ —— 状态转移概率分布， $a_{ij} = P[q_t = s_j | q_{t-1} = s_i], 1 \leq i, j \leq N$ 。
4. $B = \{b_j(k)\}$ —— 在状态 j 条件下，观察值 v_k 的概率分布，即

$$b_j(k) = P[v_k \text{ att} | q_t = s_j], \quad 1 \leq j \leq N, \quad 1 \leq k \leq M$$

5. $\pi = \{\pi_i\}$ —— 初始状态分布， $\pi_i = P[q_1 = s_i], \quad 1 \leq i \leq N$ 。

HMM 模型参数集可以简单地表示为 $\lambda = (A, B, \pi)$ 。利用隐式马尔可夫模型实际问题大致可以归为 3 类：

1. 给定观察序列 $O = O_1 O_2 O_3 \dots O_T$ 和模型参数集合 $\lambda = (A, B, \pi)$ ，计算 $P(O | \lambda)$ 。

首先来看在给定一个状态序列 $Q = q_1 q_2 q_3 \dots q_T$ 下，产生观察序列 $O = O_1 O_2 O_3 \dots O_T$ 的概率，并且假设各个观察之间统计意义上是独立的，也就是说在某一时刻系统在某特殊状态下产生某观察与其他时刻无关，即

$$P(O | Q, \lambda) = \prod_{t=1}^T P(O_t | q_t, \lambda) = b_{q_1}(O_1) b_{q_2}(O_2) \dots b_{q_T}(O_T) \quad (1.6)$$

而在模型 λ 下，产生这样一串状态序列 $Q = q_1 q_2 q_3 \dots q_T$ 的概率为：

$$P(Q | \lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \dots a_{q_{T-1} q_T} \quad (1.7)$$

现在计算给定模型 λ 下，观察序列 O 和状态序列 Q 的联合概率，即

$$P(O, Q | \lambda) = P(O | Q, \lambda) \cdot P(Q | \lambda) \quad (1.8)$$

为了计算 $P(O | \lambda)$ ，需要对上式求各种可能状态序列下之和，即

$$\begin{aligned} P(O | \lambda) &= \sum_{all Q} P(O, Q | \lambda) = \sum_{all Q} P(O | Q, \lambda) P(Q | \lambda) \\ &= \sum_{q_1 q_2 \dots q_T} \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) \dots a_{q_{T-1} q_T} b_{q_T}(O_T) \end{aligned} \quad (1.9)$$

上述给出了求 $P(\mathbf{O}|\lambda)$ 的过程, 但如果直接用(1.9)式来计算 $P(\mathbf{O}|\lambda)$, 所需的计算次数为 $2T \cdot N^T$, 因为针对每一时刻 t ($t=1,2,\dots,T$) 有 N 种可能状态, 所以所有可能状态序列数为 N^T , 而针对每一种状态序列, 需做 $2T$ 次乘[精确地说, 需要 $(2T-1) \cdot N^T$ 次乘, N^T-1 次加]。这样的计算量是非常庞大的, 即使对于数值较小的 N 和 T , 也是一个天文数字, 如 $N=5$ (5 个状态), $T=100$, 所需计算的数量级为 $2 \times 100 \times 5^{100} \approx 10^{72}$ 次运算。

很明显, 我们需要找到一个有效算法 (是指在多项式时 $2 \times 100 \times 5^{100} \approx 10^{72}$ 间内能完成的算法)。常用的算法是前向-后向算法 (Forward-Backward Algorithm)。

考虑一个前向变量 $\alpha_t(i)$, 定义为

$$\alpha_t(i) = P(O_1, O_2, \dots, O_t, q_t = s_i | \lambda) \quad (1.10)$$

它的物理含义是给定模型 λ , 计算从开始到 t 时刻的观察序列为 $O_1 O_2 \dots O_t$, 而 t 时刻系统的状态为 s_i 的概率。可以采用递归形式来计算 $\alpha_t(i)$:

$$(1) \text{ 初始化 } \alpha_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N \quad (1.11)$$

$$(2) \text{ 递归计算 } \alpha_{t+1}(j) = \left[\sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(O_{t+1}), \quad 1 \leq t \leq T-1, \quad 1 \leq j \leq N \quad (1.12)$$

$$(3) \text{ 最终结果 } P(\mathbf{O}|\lambda) = \sum_{i=1}^N \alpha_T(i) \quad (1.13)$$

上述由 t 时刻 s_i 状态到 $t+1$ 时刻 s_j 状态的递归过程可由图 1.1 说明, 而整个过程可以用图 1.2 表示。

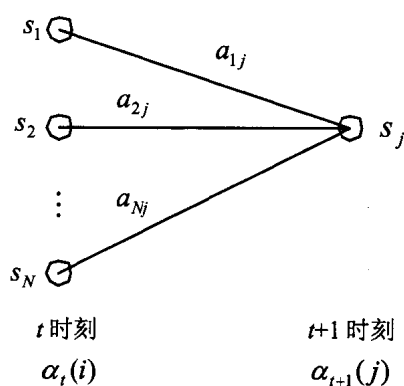


图 1.1 前向变量递归过程

从整个过程可以看到所需计算量降为 N^2T 次乘, 精确地说, 需要 $N(N+1)(T-1) + N$ 次乘、 $N(N-1)(T-1)$ 次加。同样当有 5 个状态 ($N=5$)、 $T=100$ 时, 只需约 3000 次运算, 与 10^{72} 次运算比较, 运算量减少了 10^{69} 个数量级。

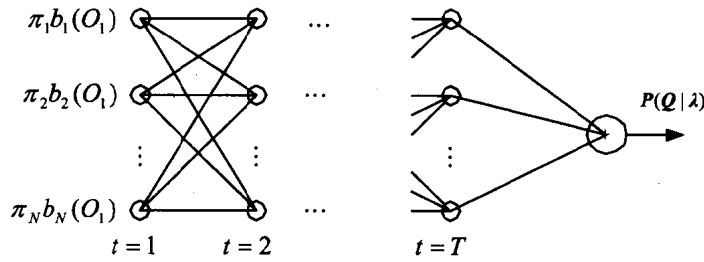


图 1.2 整个递归过程

同样道理，可以定义一个后向变量 $\beta_t(i)$ ：

$$\beta_t(i) = P(O_{t+1}O_{t+2} \cdots O_T | q_t = s_i, \lambda) \quad (1.14)$$

它的物理含义是给定模型 λ ，计算从 $t+1$ 时刻到 T 时刻的观察序列为 $O_{t+1}O_{t+2} \cdots O_T$ ，而 t 时刻系统的状态为 s_i 的概率。也可以采用递归形式来计算 $\beta_t(i)$ ：

$$(1) \text{ 初始化 } \beta_t(i) = 1, 1 \leq i \leq N \quad (1.15)$$

$$(2) \text{ 递归推导 } \beta_t(i) = \sum_{j=1}^N a_{ij} b_j(O_{t+1}) \beta_{t+1}(j), t = T-1, T-2, \dots, 1, 1 \leq i \leq N \quad (1.16)$$

$$(3) \text{ 最终结果 } P(O|\lambda) = \sum_{i=1}^N \beta_1(i) \quad (1.17)$$

同样计算 $\beta_t(i) (1 \leq t \leq T, 1 \leq i \leq N)$ 所需的计算量为 N^2T 。计算过程可以用图 1.3 表示。前向与后向算法对解决 HMM 模型中后两个问题有很大的帮助。

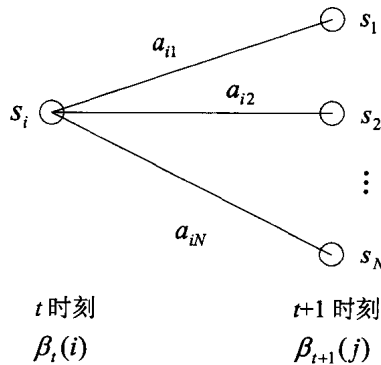


图 1.3 后向变量的递归过程

2. 给定 HMM 模型，寻找一个与给定观察序列相匹配的最优的状态序列。

最优标准不一样，所选的最优状态序列也不同。在此选择最优状态序列的标准为最可能出现的状态序列，以这样的标准解决问题 2，定义如下变量：

$$\gamma_t(i) = P\{q_t = s_i | O, \lambda\} \quad (1.18)$$

它的物理含义是给定模型 λ 和观察序列 $\mathbf{O} = O_1 O_2 O_3 \cdots O_T$ 的条件下, 计算 t 时刻状态为 s_i 的概率。可以利用前面所讲的前向和后向变量来计算 $\gamma_t(i)$

$$\gamma_t(i) = \frac{P(q_t = s_i, \mathbf{O} | \lambda)}{P(\mathbf{O} | \lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^N \alpha_t(i)\beta_t(i)} \quad (1.19)$$

由于分母 $\sum_{i=1}^N \alpha_t(i)\beta_t(i)$ 的归一化作用, 使得

$$\sum_{i=1}^N \gamma_t(i) = 1 \quad (1.20)$$

运用 $\gamma_t(i)$, 可以选择 t 时刻最优状态 q_t ,

$$q_t = \arg \max_{1 \leq i \leq N} [\gamma_t(i)], \quad 1 \leq t \leq T \quad (1.21)$$

式(1.21)虽然能选择出某一时刻最可能出现的状态, 但在有些特殊的情况下会出现问题, 如通过式(1.21)选择出某一状态序列, 在 t 时刻的状态为 s_i , $t+1$ 时刻的状态为 s_j , 但在这个 HMM 模型中, $a_{ij} = 0$, 也就是说系统不能从状态 s_i 转移到状态 s_j , 通过式(1.21)列不是一个有效状态序列。其原因是上述思想孤立地考虑各个时刻的状态, 而没有考虑各个时刻状态之间的关联性。

解决上述出现的问题, 就要修正选择的优化标准, 可以考虑状态对, 如 (q_t, q_{t+1}) 出现的概率, 或更多个状态序列出现的概率。在此选择一个最优状态序列(路径), 寻找这样一条最优路径, 常用的方法是用动态规划法, 在 HMM 模型中一种常见的算法是 Viterbi 算法。

Viterbi 算法的思想是: 给定观察序列 $\mathbf{O} = O_1 O_2 O_3 \cdots O_T$, 寻找一个最优状态序列 $\mathbf{Q} = q_1 q_2 q_3 \cdots q_T$, 定义如下变量:

$$\delta_t(i) = \max_{q_1 q_2 \cdots q_{t-1}} P[q_1 q_2 \cdots q_t = i, O_1 O_2 \cdots O_t | \lambda] \quad (1.22)$$

算法递归推导过程为:

$$\delta_{t+1}(i) = [\max_j \delta_t(j) a_{ji}] \cdot b_j(O_{t+1}) \quad (1.23)$$

算法步骤如下:

$$(1) \text{ 初始化 } \delta_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N \quad (1.24)$$

$$\psi_1(i) = 0, \quad 1 \leq i \leq N \quad (1.25)$$

$$(2) \text{ 递归推导 } \delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(O_t), \quad 2 \leq t \leq T, \quad 1 \leq j \leq N \quad (1.26)$$

$$\psi_t(j) = \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}], \quad 2 \leq t \leq T, \quad 1 \leq j \leq N \quad (1.27)$$

$$(3) \text{ 最终结果 } P^* = \max_{1 \leq i \leq N} [\delta_T(i)] \quad (1.28)$$

$$q_T^* = \arg \max_{1 \leq i \leq N} [\delta_T(i)] \quad (1.29)$$

3. 怎样调整模型参数集合 $\lambda = (A, B, \pi)$, 使得 $P(O|\lambda)$ 最大?

到目前还没有解析方法来解决这个问题, 只能通过一组训练样本来训练模型, 使 $P(O|\lambda)$ 最大, 但一般情况下, 只能得到局部最优解。现在所用的算法为 Baum-Welch 算法[有时也称为 EM (Expectation Modification) 算法]。

为了修正 HMM 模型参数, 定义概率变量, 即给定模型 λ 和观察序列 O , 系统在 t 时刻处于状态 s_i 而在 $t+1$ 时刻处于 s_j 状态的概率:

$$\xi_t(i, j) = P\{q_t = s_i, q_{t+1} = s_j | O, \lambda\} \quad (1.30)$$

上述概率公式可以用图 1.4 表示。

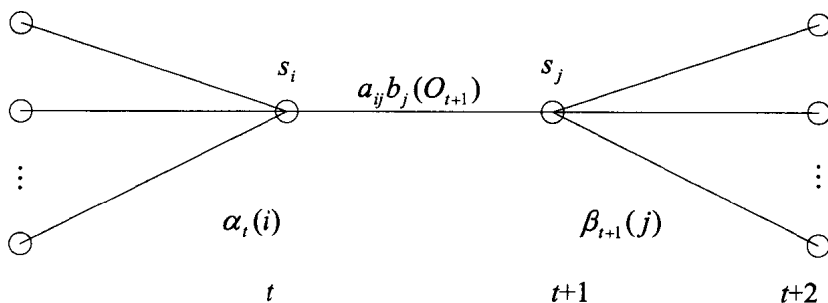


图 1.4 $\xi_t(i, j)$ 的计算图示

从前面定义的前向变量和后向变量, 可以得到:

$$\begin{aligned} \xi_t(i, j) &= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)} \\ &= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)} \end{aligned} \quad (1.31)$$

而前面提到的变量 $\gamma_t(i)$ (给定模型 λ 和观察序列 $O = O_1 O_2 O_3 \cdots O_T$ 的条件下, 计算 t 时刻状态为 s_i 的概率) 可以由 $\xi_t(i, j)$ 对所有状态 j 求和得到:

$$\gamma_t(i) = \sum_{j=1}^N \xi_t(i, j) \quad (1.32)$$

运用上述这些变量，可以估计 HMM 模型参数：

$$\bar{\pi}_i = \text{系统在初始时刻处于状态 } i \text{ 的期望值} = \gamma_1(i) \quad (1.33)$$

$$\bar{a}_{ij} = \frac{\text{系统从状态 } i \text{ 转移到状态 } j \text{ 的期望值}}{\text{系统由状态 } i \text{ 转移出去的期望值}} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad (1.34)$$

$$\bar{b}_j(k) = \frac{\text{系统处于状态 } j \text{ 而观察值为 } v_k \text{ 的期望值}}{\text{系统处于状态 } j \text{ 的期望值}} = \frac{\sum_{t=1}^T \gamma_t(j)}{\sum_{t=1}^T \gamma_t(j)} \quad (1.35)$$

如果目前模型参数为 $\lambda = (A, B, \pi)$ ，运用式(1.33)~式(1.35)来修正模型参数 A, B, π ，得到新模型 $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$ ，使得 $P(O|\bar{\lambda}) > P(O|\lambda)$ 。上式修正过程还受一些统计条件的限制，如：

$$\sum_{i=1}^N \bar{\pi}_i = 1 \quad (1.36)$$

$$\sum_{j=1}^N \bar{a}_{ij} = 1, \quad 1 \leq i \leq N \quad (1.37)$$

$$\sum_{k=1}^N \bar{b}_j(k) = 1, \quad 1 \leq j \leq N \quad (1.38)$$

前面简单地介绍了隐式马尔可夫模型的基本原理和一些基本算法，而隐式马尔可夫模型经常运用于图像分割和图像恢复之中。

如果需要为一组图像设计图像处理算法，我们把一幅特定的图像看成是含有相同像素的可能的图像集合中的一个样本，这个图像集合产生于一个二维随机场，这个随机场可以用均值函数和协方差函数来表征；另一方面也可以把这一图像集合看成是由一线性系统在白噪声或已知谱密度函数的随机序列的驱动下产生的输出。

如果不考虑图像中行与行之间（或列与列之间）的相关性，而是把图像中行与行头尾相接或列与列头尾相接形成一维信号 u_k ，还可以用自回归模型来描述它，如：

$$u_k = \sum_{n=1}^p a_n u_{k-n} + \varepsilon_k, \quad E\varepsilon_k = 0, \quad E\varepsilon_k \varepsilon_l = \beta^2 \delta_{k,l} \quad (1.39)$$

其中 a_n 为模型系数， ε_k 是一个零均值白噪声随机序列。这种模型常应用于预测编码（DPCM）、递归滤波及图像恢复中。