统计学习方法（第一版）笔记（汇总人：天国之影） ¶

目录

1 第1章-统计学习方法概论-导读
  1.1 统计学习
  1.2 监督学习
  1.3 统计学习三要素
  1.4 模型评估与模型选择
  1.5 正则化与交叉验证
  1.6 泛化能力
  1.7 生成模型与判别模型
  1.8 分类问题
  1.9 标注问题
  1.10 回归问题
2 第1章-统计学习方法概论-泛化误差上界
  2.1 定理：泛化误差上界
  2.2 Hoeffding不等式
  2.3 证明
3 第1章-统计学习方法概论-极大似然估计和贝叶斯估计
  3.1 极大似然估计
  3.2 贝叶斯估计
  3.3 对比极大似然估计和贝叶斯估计
4 第2章-感知机模型-导读
  4.1 感知机模型
  4.2 感知机学习策略
  4.3 感知机学习算法
    4.3.1 感知机学习算法的原始形式
    4.3.2 感知机学习算法的对偶形式
    4.3.3 对比两个算法
5 第2章-感知机模型-随机梯度下降法
6 第2章-感知机模型-算法收敛性
  6.1 定理2.1 Novikoff
  6.2 解释
  6.3 证明
    6.3.1 证明（1）
    6.3.2 证明（2）
  6.4 思考
7 第3章-k近邻法-k近邻算法
  7.1 k近邻算法
  7.2 k近邻模型
    7.2.1 模型
    7.2.2 距离度量
    7.2.3 k值的选择
    7.2.4 分类决策规则
  7.3 k近邻法的实现：kd树
  7.4 回顾总结
8 第4章-朴素贝叶斯法-导读

8.1 三个分类模型的比较
8.2 朴素贝叶斯法的学习与分类
8.3 朴素贝叶斯法的参数估计
8.3.1 极大似然估计
8.3.2 贝叶斯估计
9 第4章-朴素贝叶斯法-后验概率最大化
9.1 推导
9.2 解释
10 第4章-朴素贝叶斯法-贝叶斯估计
10.1 问题描述
10.2 极大似然估计
10.3 贝叶斯估计
10.4 总结
11 第5章-决策树-导读
11.1 决策树模型与学习
11.2 特征选择
11.2.1 熵
11.2.2 信息增益
11.2.3 信息增益比
11.3 决策树的生成
11.3.1 算法5.2（ID3算法）
11.3.2 算法5.2（C4.5算法）
11.4 决策树的剪枝
11.5 CART算法
12 第5章-决策树-信息增益与基尼指数
12.1 从定性的角度了解熵与基尼指数的含义
12.2 混乱程度的理解
12.3 关系图说明
13 第6章-Logistic回归与最大熵模型-导读
13.1 Logistic回归模型
13.1.1 二项Logistic回归模型
13.1.2 二项Logistic回归模型参数估计
13.1.3 多项Logistic回归模型
13.2 最大熵模型
13.3 模型学习的最优化算法
13.4 总结
14 第6章-Logistic回归与最大熵模型-改进的迭代尺度法
14.1 改进的迭代尺度法
14.2 求解最大似然函数
15 第6章-Logistic回归与最大熵模型-拉格朗日对偶性
15.1 原始问题
15.2 拉格朗日函数
15.3 总结
15.4 定理C.1
15.5 定理C.2
15.6 定理C.3(KKT条件)
16 第7章-支持向量机（SVM）-导读
16.1 线性可分支持向量机与硬间隔最大化
16.2 线性支持向量机与软间隔最大化
16.2.1 支持向量
16.2.2 合页损失函数
16.3 非线性支持向量机与核函数

16.4 序列最小最优化算法
17 第7章-支持向量机-最大间隔分离超平面存在唯一性
17.1 证明
17.1.1 首先需要验证w, b是否满足约束条件
17.1.2 证明$w_1^* = w_2^*$
17.1.3 证明$b_1^* = b_2^*$
17.1.4 结论
18 第7章-支持向量机-软间隔最大化对偶问题
18.1 对书中第109页的一句话的理解
18.2 软间隔最大化对偶问题的可行性
18.3 对偶问题的转换与推导
18.4 求解（定理7.3）
18.5 对书中第109页的一句话的理解
18.6 软间隔最大化对偶问题的可行性
18.7 对偶问题的转换与推导
18.8 求解（定理7.3）
19 第8章-提升方法-导读
19.1 提升方法AdaBoost算法
19.1.1 提升方法的基本思路
19.1.2 AdaBoost算法（算法8.1）
19.2 AdaBoost算法的训练误差分析
19.3 AdaBoost算法的解释
19.3.1 前向分步算法
19.3.2 前向分步算法与AdaBoost
19.4 提升树
19.4.1 回归问题的提升树算法
19.4.2 对算法8.3的解释
19.4.3 梯度提升算法
20 第8章-提升方法-AdaBoost训练误差
20.1 定理8.1的证明
20.2 定理8.2的证明
21 第8章-提升方法-前向分步算法
21.1 G^*求解
21.2 α^*求解
21.3 权重$\bar{w}_{m,i}$的更新
22 第9章-EM算法及推广-导读
22.1 EM算法的引入
22.1.1 EM算法
22.1.1.1 三硬币模型
22.1.1.2 EM算法的解释
22.1.1.3 EM算法
22.1.2 EM算法的导出
22.2 EM算法的收敛性
22.3 EM算法在高斯混合模型学习中的应用
22.3.1 高斯混合模型
22.3.2 高斯混合模型的参数估计
22.4 EM算法的推广
23 第9章-EM算法及推广-EM算法的导出
24 第9章-EM算法及推广-高斯混合模型
24.1 明确隐变量，写出完全数据的对数似然函数
24.2 EM算法的E步，确定Q函数
24.3 确定EM算法的M步

25 第10章-隐马尔科夫模型(HMM)-导读
25.1 概率图模型
25.2 隐马尔可夫模型的基本概念
25.2.1 隐马尔可夫模型的定义
25.2.1.1 状态与状态之间的关系
25.2.1.2 状态与观测之间的关系
25.2.1.3 初始状态概率
25.2.2 两个基本假设
25.2.3 三个基本问题
25.3 概率计算算法
25.3.1 直接计算法
25.3.2 前向算法
25.3.3 后向算法
25.4 学习算法
25.4.1 监督学习方法
25.4.2 Baum-Welch算法（EM算法）
25.5 预测算法
25.5.1 近似算法
25.5.2 维特比算法
26 第10章-隐马尔科夫模型-前向算法
26.1 式10.17的推导
26.2 式10.16的推导
27 第10章-隐马尔科夫模型-维特比算法
28 第11章-条件随机场-导读
28.1 概率无向图模型
28.1.1 概率图模型
28.1.2 马尔可夫性
28.1.3 概率无向图模型的因子分解
28.1.4 总结
28.2 条件随机场的定义与形式
28.2.1 条件随机场的定义
28.2.2 条件随机场的参数化形式
28.2.3 条件随机场的简化形式
28.2.4 条件随机场的矩阵形式
28.3 条件随机场的概率计算问题
28.4 条件随机场的学习算法
28.4.1 条件随机场的预测算法
29 第11章-条件随机场-条件随机场的矩阵形式
29.1 推导从参数化形式转化为矩阵形式
29.2 例11.2讲解
29.3 对比矩阵M和状态转移矩阵A
30 第11章-条件随机场-拟牛顿法（附录B）
30.1 牛顿法
30.2 拟牛顿法
30.3 书上对应的两个算法

1 第1章-统计学习方法概论-导读

1.1 统计学习

监督学习的实现步骤：

1. 得到一个有限的训练数据集合
2. 确定模型的假设空间，也就是所有的备选模型
3. 确定模型选择的准则，即学习的策略
4. 实现求解最优模型的算法
5. 通过学习方法选择最优模型
6. 利用学习的最优模型对新数据进行预测或分析

学习系统： 模型假设空间、学习策略、学习算法

1.2 监督学习

训练集：$T = (x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)$
实例x的特征向量：$x = (x^{(1)}, x^{(2)}, \cdots, x^{(n)})$
模型：
1）决策函数：$Y = f(X)$
预测形式：$y = f(x)$
2）条件概率分布：$P(Y|X)$
预测形式：$\arg\max_y P(y|x)$

总结：

首先介绍了训练集，在学习系统和模型这一部分，介绍了两种模型，第一种是以决策函数来表示的，第二种是以条件概率分布来表示的。

以决策函数来表示，给定一个输入X会得到一个相对应的预测值Y。如果是以条件概率分布来表示，输入一个X得到的是相对应的Y的分布，在实际预测中，取这个分布中Y的一个众数（即条件概率最大的那个点）。

1.3 统计学习三要素

1. 模型（假设空间）：模型一共分为两种：决策函数和条件概率分布（前文已述）
   决策函数：$F = \{f|Y = f_\theta(X), \theta \in R^n\}$
   条件概率分布：$F = \{P|P_\theta(Y|X), \theta \in R^n\}$
2. 策略：第一个要素是假设空间，我们已经确定了一个备选模型的集合，从该集合中找到一个最优的模型，策略就是以什么样的标准来确定最优模型。策略体现在损失函数上，损失函数是对于每一个实例，预测值和真实值之间差别的一个惩罚。

常见的损失函数：

- 0-1损失函数：
$$L(Y, f(X)) = \begin{cases} 1, Y \neq f(X) \\ 0, Y = f(X) \end{cases}$$
一般用于分类问题。
- 平方损失函数：
$$L(Y, f(X)) = (Y - f(X))^2$$
一般由于回归问题。
- 绝对损失函数：
$$L(Y, f(X)) = |Y - f(X)|$$
也是用于回归模型。平方损失函数，对于差值较大的观测值和预测值，它的惩罚力度会更强。
- 对数似然损失函数：
$$L(Y, P(Y|X)) = -\log P(Y|X)$$
，针对条件概率函数。

现在介绍的损失函数，都是针对每一个具体的实例得到的损失，但是在学习过程中，在训练集中有多个实例，也会有多个损失函数，如何根据N个损失值来决定最优模型呢？ 下面有两个准则：
（1）经验风险最小化：

$$\min_{f \in F} \frac{1}{N} \sum_{i=1}^{N} L\left(y_i, f\left(x_i\right)\right)$$

（2）结构风险最小化：

$$\min_{f \in F} \frac{1}{N} \sum_{i=1}^{N} L\left(y_i, f\left(x_i\right)\right) + \lambda J(f)$$

，其中$J(f)$表示的是函数f的模型复杂度，平衡了经验风险和模型复杂度。

3. 算法：如何根据策略，从这一系列的备选模型中，选择一个最优的模型。

1.4 模型评估与模型选择

训练误差：

$$\frac{1}{N} \sum_{i=1}^{N} L\left(y_i, \widehat{f}\left(x_i\right)\right)$$

测试误差：

$$\frac{1}{N'} \sum_{i=1}^{N'} L\left(y_i, \widehat{f}\left(x_i\right)\right)$$

1.5 正则化与交叉验证

最小化结构风险：

$$\min_{f \in F} \frac{1}{N} \sum_{i=1}^{N} L\left(y_i, f\left(x_i\right)\right) + \lambda J(f)$$

，通过最小化结构风险，平衡训练集的拟合程度和模型复杂度。
交叉验证：将训练集分成两部分，一部分是训练集，一部分是验证集。

1.6 泛化能力

定理：泛化误差上界
对于二分类问题，当假设空间是有限个函数的集合$F = \{f_1, f_2, \cdots, f_d\}$时，对任意一个函数$f \in F$，至少以概率$1 - \delta$，以下不等式成立：

$$R(f) \leq \hat{R}(f) + \varepsilon(d, N, \delta)$$

其中

$$\varepsilon(d, N, \delta) = \sqrt{\frac{1}{2N}\left(\log d + \log \frac{1}{\delta}\right)}$$

解释：$R(f)$表示的是期望风险，$\hat{R}(f)$是f在训练集上的经验风险，期望风险代表了f在预测数据，或者说在总体数据上的一个表现。这个定理保证了当学习到的模型在训练集上的经验风险，可以用来体现这个模型在总体数据或在测试集上的风险。

N表示样本量，样本量越大，用经验风险代表期望风险的效果越好；d表示备选模型的个数，个数越多，效果越差。

注： 泛化能力不是针对一个备选模型，而是针对假设空间中的所有模型都成立。

1.7 生成模型与判别模型

生成方法：

$$P(Y|X)=\frac{P(X,Y)}{P(X)}$$

判别方法：

$$f(X)或P(Y|X)$$

1.8 分类问题

- TP——将正类预测为正类数
- FN——将正类预测为负类数
- FP——将负类预测为正类数
- TN——将负类预测为负类数

精确率：预测出来的正类中，有多大的比例是正确的

$$P=\frac{TP}{TP+FP}$$

召回率：真实情况下是正类的这些，预测准确的概率是多少

$$R=\frac{TP}{TP+FN}$$

1.9 标注问题

输入：$x=\left(x^{(1)},x^{(2)},\cdots,x^{(n)}\right)^T$

输出：$y=\left(y^{(1)},y^{(2)},\cdots,y^{(n)}\right)^T$

例如：文本分类

输入：At Microsoft Research

输出：At/O Micosoft/B Research/E

1.10 回归问题

回归问题中，输出变量（预测值）是连续值，根据输出变量值的类型可以把监督学习分为分类问题和回归问题。

2 第1章-统计学习方法概论-泛化误差上界

2.1 定理：泛化误差上界

对于二分类问题，当假设空间是有限个函数的集合$F=\{f_1,f_2,\cdots,f_d\}$时，对任意一个函数$f\in F$，至少以概率$1-\delta$，以下不等式成立：

$$R(f)\leq\hat{R}(f)+\varepsilon(d,N,\delta)$$

其中$\varepsilon(d,N,\delta)=\sqrt{\frac{1}{2N}\left(\log d+\log\frac{1}{\delta}\right)}$

2.2 Hoeffding不等式

有随机变量序列$x_1,x_2,\cdots,x_n$，定义随机变量的和$S_n=x_1+x_2+\cdots+x_n$，随机变量和的期望为$ES_n=E(\quad x_i)$，随机变量x_i能被一个区间控制住，记为$x_i\in[a_i,b_i]$，满足上述这些条件，就会有如下不等式：

$$P\left(S_n-ES_n\geqslant t\right)\leqslant\exp\left(-\frac{2t^2}{2(b_i-a_i)^2}\right)$$

这样的一个形式不太好理解，我们给该公式做一个变形。不考虑随机变量序列的和，而是考虑序列的均值x_n。

由上式可以得到：$x_n = \frac{S_n}{n}$，$E\left(x_n\right) = \frac{ES_n}{n}$

现考虑随机变量序列的均值与均值期望之间的距离大于等于t的概率为$P\left(x_n - E\left(x_n\right) \geqslant t\right)$

公式推导如下：$P\left(x_n - E\left(x_n\right) \geqslant t\right) = P(S_n - ES_n \geqslant t) \leqslant \exp\left(-\frac{2n^2t^2}{(b_i - a_i)^2}\right)$，当$n$比较大的时候，该概率为$O(e^{-n})$，当$n \to \infty$时，概率趋近于0。

得$P(X - EX \geqslant t) \leqslant \exp\left(-\frac{2n^2t^2}{(b_i - a_i)^2}\right)$，该公式中期望和均值是可以交换的，更改为

$P(EX - X \geqslant t) \leqslant \exp\left(-\frac{2n^2t^2}{(b_i - a_i)^2}\right)$

2.3 证明

现考虑二分类问题，在该问题中，从假设空间中任取一个备选模型f，这个模型在训练集上的经验风险（即为随机变量的均值）$\hat{R}(f) = \frac{1}{N} \quad L(x_i, f(x_i))$，期望风险（这个模型在测试集上的表现）为$R(f)$，将上述两个风险带入到Hoeffding不等式中，

得到$P(R(f) - \hat{R}(f) \geqslant t) \leqslant \exp(-\frac{2N^2t^2}{N}) = \exp(-2Nt^2)$，以上是得到了一个备选模型成立的情况。但是在假设空间中有d个备选模型，并不知道会从这些备选模型中选取到哪一个，所以要求这些模型在训练集上的经验风险和期望风险的差值都不大，现考虑该条件的对立面：

$P(\exists f \in F, R(f) - \hat{R}(f) \geqslant t) = P(\bigcup_{f \in F}\{R(f) - \hat{R}(f) \geqslant t\}) \leqslant \sum_{f \in F} P(R(f) = \hat{R}(f) \geqslant t) \leqslant d\exp(-2Nt^2$

对于$P(\forall f \in F, R(f) - \hat{R}(f) \leqslant t) \geqslant 1 - d\exp(-2Nt^2)$，最后将$t$换成$\varepsilon$，得到$\delta = d\exp(-2N\varepsilon^2)$，变量交换得到$\varepsilon = \sqrt{\frac{1}{2N}\left(\log d + \log\frac{1}{\delta}\right)}$。

故至少以概率$1 - \delta$有$R(f) \leq \hat{R}(f) + \varepsilon(d, N, \delta)$，其中$\varepsilon$由上式得出，定理得证。

3 第1章-统计学习方法概论-极大似然估计和贝叶斯估计

3.1 极大似然估计

在掷硬币实验中估计出现正面向上的概率θ，通过一系列的实验，就会得到很多个观测的结果，将每个观测的结果用随机变量表示出来$x_i = \begin{cases} 1 & \text{正} \\ 0 & \text{反} \end{cases}$，$x_i$满足二项分布$x_i \sim B(1, \theta)$，概率函数为$P(X = x) = \theta^x(1 - \theta)^{1-x}$。

假设已知θ，根据实验结果，得到出现这组结果的概率用L表示，其似然函数为$L(\theta) = \prod_{i=1}^{n} P(X_i = x_i|\theta)$（为什么可以写成连乘形式？因为每一次投硬币的概率都是独立的）。将概率函数带入到似然函数中，得到$L(\theta) = \prod_{i=1}^{n} \theta^{x_i}(1 - \theta)^{1-x_i}$。

极大似然估计的意义：找到一个θ使得样本出现的概率是最大的，也就是需要最大化似然函数。最大化似然函数等价于最大化这个似然函数的对数：

$$\begin{aligned}\max \ln L(\theta) &= \sum [\ln \theta^{x_i} + \ln(1-\theta)^{1-x_i}] \\ &= \sum x_i \ln \theta + (n - \sum x_i) \ln(1-\theta)\end{aligned}$$

对$\ln L(\theta)$求导：$\frac{\partial \ln L(\theta)}{\partial \theta} = \frac{x_i}{\theta} - \frac{n - x_i}{1-\theta} = 0$，得到$\hat{\theta} = \frac{x_i}{n}$。

概括来说，极大似然估计就是根据样本的概率分布，写出样本的联合概率的似然函数，然后通过最大化似然函数，得到参数的估计值。

极大似然估计完全是根据样本信息得到的参数估计，其参数估计值为$\hat{\theta} = \frac{x_i}{n}$。

3.2 贝叶斯估计

先验概率密度函数：$\pi(\theta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta^{\alpha-1}(1-\theta)^{\beta-1}$，在$[0,1]$区间中，$\theta$的概率密度函数由$\alpha$和$\beta$来决定的。这里不给$\alpha$和$\beta$赋值，在实际应用过程中，需要进行赋值的。

目前已知$\pi(\theta)$和一组样本$x_1, x_2, \cdots, x_n$，根据样本信息调整对θ分布的判断，即找到对应的后验分布：

$$\begin{aligned}p(\theta|x_i, \cdots, x_n) &= \frac{p(\theta, x_1 \cdots, x_n)}{p(x_1, \cdots, x_n)} \\ &= \frac{\pi(\theta) \cdot p(x_1|\theta) \cdots p(x_n|\theta)}{\int p(\theta, x_1, \cdots, x_n) d\theta} \\ &\propto \theta^{\alpha-1}(1-\theta)^{\beta-1} \prod \theta^{x_i}(1-\theta)^{1-x_i} \\ &= \theta^{\ x_i+\alpha-1}(1-\theta)^{n-\ x_i+\beta-1}\end{aligned}$$

上式得到的最终结果也是一个贝叶斯分布，参数为 $x_i + \alpha$ 和$n - x_i + \beta$，所以在贝叶斯估计中，首先得到的是一个关于参数给定样本信息的后验分布，然后要给出θ一个具体的值来估计它，就从这个后验分布中，找出使得后验分布的概率密度最大（即众数），可得到众数值为$\hat{\theta} = \frac{x_i + \alpha - 1}{n + \alpha + \beta - 2}$。

3.3 对比极大似然估计和贝叶斯估计

极大似然估计（MLE）：$\hat{\theta} = \frac{x_i}{n}$

贝叶斯估计（Bayes）：$\hat{\theta} = \frac{x_i + \alpha - 1}{n + \alpha + \beta - 2}$

当样本量$n \to \infty$时，贝叶斯估计的结果为$\frac{x_i}{n}$，可以看到这个结果就是极大似然估计的结果。

解释：

首先会在贝叶斯估计中，给出参数的先验信息，但是当样本量足够大的时候，之前的先验信息与样本信息相比，就非常的微不足道了。所以近似于用所有的样本信息估计θ得到的结果。那为什么还会有贝叶斯估计呢？由于之前考虑的样本量是很大的情况，现在考虑一个极端情况，假如只有一个样本，通过极大似然估计，θ只能为0或1，但在贝叶斯估计中，如果得到的样本取值为0，得到的结果是$\frac{\alpha - 1}{\alpha + \beta - 1}$，如果得到的样本取值为1，得到的结果是$\frac{\alpha}{\alpha + \beta - 1}$。

对比可得，由贝叶斯估计得到的结果不会像极大似然估计那么极端，所以当样本量小的时候，这个就是贝叶斯估计的优势所在，当样本量大的时候，这两个估计的结果是一样的。

4 第2章-感知机模型-导读

从本章开始，就要学习各种模型了，在学习每个模型的时候，最重要的是要知道这个模型的适用条件、所能解决的问题、对应的统计学习方法三要素（假设空间、策略、求解算法）

本章的感知机模型针对的是二分类问题，有一个比较强的假设，要求二分类问题是线性可分的。

4.1 感知机模型

- 输入空间：$X \subseteq R^n$
- 输入变量：$x \in X$
- 输出空间：$Y = \{+1, -1\}$
- 输出变量：$y \in \{+1, -1\}$
- 假设空间：$f(x) = \text{sign}(w \cdot x + b)$

相关说明： 输入变量x是一个n维向量，$+1$表示该实例是正类，-1表示该实例是负类。

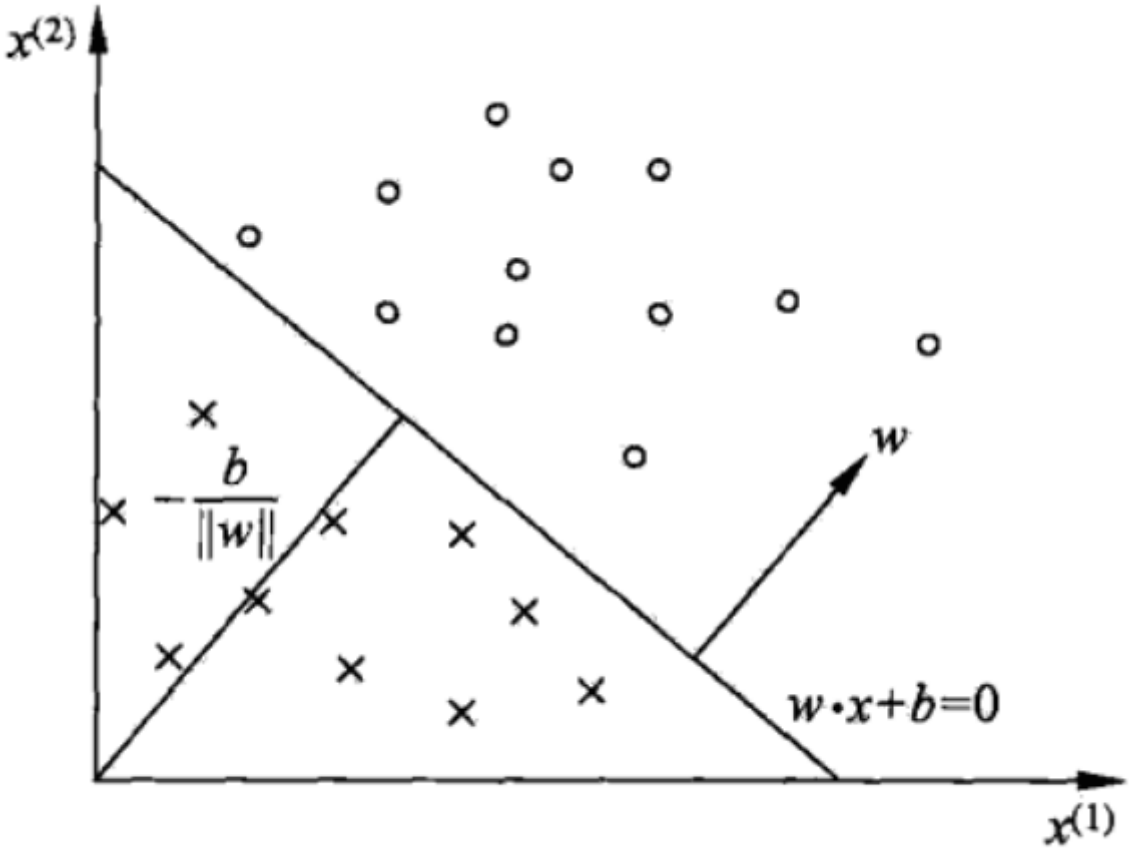

图 2.1 感知机模型

上图表示的是一个输入空间是2维的模型，故$n = 2$，图中$x^{(1)}$和$x^{(2)}$表示两个分量，然后根据训练集中每一个实例在这两个输入变量上的取值，将其划分到输入空间中对应的点，实例就是由图中的∘和×表示的，其中∘表示正类，×表示负类。

感知机模型的输入变量是线性可分的，也就是说图中的这些点可以用一条直线来分开。在直线上方的是正类，下方的是负类。该直线的表示形式为$w_1 x^{(1)} + w_2 x^{(2)} + b = 0$，该平面中所有的直线，就构成了假设空间。判断实例属于哪一类，根据如下公式：$w_1 x^{(1)} + w_2 x^{(2)} + b \begin{cases} \geqslant 0 & \text{正类} \\ < 0 & \text{负类} \end{cases}$，假设空间

$f(x) = \text{sign}(w \cdot x + b) = \begin{cases} +1 & w \cdot x + b \geqslant 0 \\ -1 & w \cdot x + b < 0 \end{cases}$，其中$w$是$n$维向量，

$w \cdot x = w_1 x^{(1)} + w_1 x^{(1)} + \cdots + w_n x^{(n)}$

总结： 在感知机模型中，假设空间是关于输入变量x的线性函数，再取其符号函数。取符号函数的目的是输出变量是需要分类的，为+1或-1。

4.2 感知机学习策略

损失函数： 误分类点到超平面的总距离$L(w,b)=-\sum_{x_i\in M}y_i(w\cdot x_i+b)$

推导：

在感知机学习中，损失函数的定义如下：判断错误的实例到直线的距离，对于任意一点x_i到一条直线的距离为$\frac{|w\cdot x_i+b|}{\|w\|}$，其中$\|w\|=\sqrt{w_1^2+\cdots+w_n^2}$，误分类点$x_i$到直线的距离等价与$\frac{-y_i(w\cdot x_i+b)}{\|w\|}$，对于所有的误分类点，$w\cdot x_i+b$是小于0的，所以需要添加-(负号)来保证该公式是大于0的。于是，对于所有的误分类点到超平面的总距离为$\sum_{x_i\in M}\frac{-y_i(w\cdot x_i+b)}{\|w\|}$，由于要算出最小值，可以去掉$\|w\|$，所以可得

$$-\sum_{x_i\in M}y_i(w\cdot x_i+b)$$

解释：

超平面：在输入变量是2维的时候，用一条直线来分类的，当输入变量是3维的时候，用一个平面类划分，当输入变量是4维的时候，用一个3维的平面来划分，这个时候，该平面被称为超平面。当输入变量是n维的时候，用一个$n-1$维的超平面来分类，所以就用超平面来表示分割平面。

4.3 感知机学习算法

4.3.1 感知机学习算法的原始形式

算法2.1（随机梯度下降法）

输入：训练数据集$T=[(x_1,y_1),\dots,(x_N,y_N))$，学习率$\eta$

1. 选取初值w_0,b_0，确定了假设空间中的一个备选模型
2. 在训练集中选取数据(x_i,y_i)
3. 如果$y_i(w\cdot x_i+b)\leqslant 0$

$$w:=w+\eta y_i x_i$$
$$b:=b+\eta y_i$$

4. 转至2，直到训练集中没有误分类的点

输出：w,b

说明： η如果小，这条更新的直线向误分类点移动程度小，如果大的话，移动程度大。

4.3.2 感知机学习算法的对偶形式

$$f(x)=\text{sign}(\sum_{j=1}^{N}\alpha_j y_j x_j\cdot x+b)$$

$$\alpha=(\alpha_1,\cdots,\alpha_N)^T$$

该假设空间是由α_j,b决定的。

算法2.2：

输入：训练数据集$T=[(x_1,y_1),\dots,(x_N,y_N))$，学习率$\eta$

1. 初值$\alpha:=0,b:=0$
2. 在训练集中选取数据(x_i,y_i)
3. 如果$y_1\left(\sum_{j=1}^{N}\alpha_j y_j x_j\cdot x+b\right)\leqslant 0$

$$\alpha_i:=\alpha_i+\eta$$
$$b:=b+\eta y_i$$

4. 转至2，直到训练集中没有误分类的点

输出：α, b

4.3.3 对比两个算法

对于算法2.2，每次只会更新两个数α_i和b，而算法2.1，每次需要更新一个向量w和一个数b，相比于算法2.2，需要更新得更多，并且每次都要计算一次$w \cdot x_i$内积，计算量也会很大。算法2.2也需要计算一个内积$x_i \cdot x_j$，但只是当前点的内积，可以提前将所有实例的输入向量的内积都算出来。

所以，算法2.2比算法2.1整体上计算量要少一点。

5 第2章-感知机模型-随机梯度下降法

感知机模型的学习策略是要最小化误分类点到分类超平面距离的和，其对应的经验风险函数是$L(w,b) = -\sum_{x_i \in M} y_i(w \cdot x_i + b)$，是一个关于$w$和$b$的函数，$w$是一个$n$维向量，$x_i$(每一个输入变量)也是一个$n$维向量，$x_i \in M$表示经验风险函数中包含的求和项只是在当前的分类超平面中被误分类的那些点，现在需要找到是该公式的值最小的w和b，也就是要最小化这个函数$L(w,b)$。

首先求出关于w的梯度：$\bigtriangledown_w L(w,b) = -\sum_{x_i \in M} y_i x_i$

再求出关于b的梯度：$\bigtriangledown_b L(w,b) = -\sum_{x_i \in M} y_i$

采用梯度下降法，对w, b进行更新：

$$w := w + \eta \sum_{x_i \in M} y_i x_i$$

$$b := b + \eta \sum_{x_i \in M} y_i$$

而随机梯度下降法，只会添加一个$y_i x_i$，为什么要采用随机梯度下降呢？因为在一般的问题里面，可能数据量比较大，每次只用一个数据来更新w和b，降低计算量；还有另外一个原因，需要更新的x_i是误分类点，在更新完之后，就不是误分类的点，所以再第二次更新w和b的时候，仍需再一次计算$y_i x_i$，浪费计算时间。

随机梯度下降法，每次只用一个实例点进行更新w和b，然后用新的w和b判断目前状态下的误分类点。

6 第2章-感知机模型-算法收敛性

6.1 定理2.1 Novikoff

假设训练数据集$T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$是线性可分的，其中$x_i \in \mathcal{X} = \mathrm{R}^n, y_i \in \mathcal{Y} = \{-1, +1\}, i = 1, 2, \cdots, N$则

（1）存在满足条件$\|\hat{w}_{opt}\| = 1$的超平面$\hat{w}_{opt} \cdot \hat{x} = w_{opt} \cdot x + b_{opt} = 0$将训练数据集完全正确分开；且存在$\gamma > 0$，对所有$i = 1, 2, \cdots, N$满足

$$y_i(\hat{w}_{opt} \cdot \hat{x}) = y_i(w_{opt} \cdot x_i + b_{opt}) \geqslant \gamma$$

（2）令$R = \max_{1 \leqslant i \leqslant N} \|\hat{x}_i\|$，则感知机算法2.1在训练数据集上的误分类次数$k$满足下面不等式

$$k \leqslant \left(\frac{R}{\gamma}\right)^2$$

6.2 解释

超平面$w_{opt} \cdot x + b_{opt} = 0$为可以将训练数据集线性可分的。

先定义$\hat{w}_{opt} = (w_{opt}^T, b_{opt})^T$，$\hat{x} = (x^T, 1)^T$

再计算$\hat{w}_{opt}$和$\hat{x}$的内积（对应部分相乘再求和）：$\hat{w}_{opt} \cdot \hat{x} = w_{opt} \cdot x + b_{opt}$

所以超平面$w_{opt} \cdot x + b_{opt} = 0$可以用$\hat{w}_{opt} \cdot \hat{x} = 0$来做等价变换。
为了让$\hat{w}_{opt}$表示唯一，对其进行约束，使得其长度等于单位长度：$\|\hat{w}_{opt}\| = 1$

6.3 证明

6.3.1 证明（1）

可知对于任意超平面都有$y_i(\hat{w}_{opt} \cdot \hat{x}_i) > 0$，所以存在$\gamma = \min_i \{y_i(\hat{w}_{opt}, x_i)\}$，于是得证。

6.3.2 证明（2）

根据这个算法，每找到一个点，都会对w进行修正，修正的总次数k值一定是有上界的，要得到这个结论，需要两个步骤。
第一步： 需要证明$\hat{w}_k \cdot \hat{w}_{opt} \geqslant k\eta\gamma$，其中$\eta$就是算法中的学习率，$\hat{w}_k$就是在第$k$个误分类点修正之后所得到的。假设$\hat{w}_0 = (0, 0, \cdots, 0)^T$

$$\begin{aligned}
\hat{w}_k \cdot \hat{w}_{opt} &= (\hat{w}_{k-1} + \eta y_i \hat{x}_i) \cdot \hat{w}_{opt} \\
&= \hat{w}_{k-1} \cdot \hat{w}_{opt} + \eta y_i \hat{x}_i \cdot \hat{w}_{opt} \\
&\geqslant \hat{w}_{k-1} \cdot \hat{w}_{opt} + \eta\gamma \\
&\geqslant \hat{w}_{k-2} \cdot \hat{w}_{opt} + \eta\gamma + \eta\gamma \\
&\cdots \\
&\geqslant \hat{w}_0 \cdot \hat{w}_{opt} + k\eta\gamma \\
&= k\eta\gamma
\end{aligned}$$

第一步得证。
第二步： 需要证明$\|\hat{w}_k\|^2$有上界的，小于某个值。
二范数的平方裂项：

$$\begin{aligned}
\|a + b\|^2 &= (a + b)^T (a + b) \\
&= \|a\|^2 + 2a \cdot b + \|b\|^2
\end{aligned}$$

开始证明：

$$\begin{aligned}
\|\hat{w}_k\|^2 &= \|\hat{w}_{k-1} + \eta y_i \hat{x}_i\|^2 \\
&= \|\hat{w}_{k-1}\|^2 + 2\eta y_i \hat{w}_{k-1} \cdot \hat{x}_i + \eta^2 \|\hat{x}_i\|^2
\end{aligned}$$

上式中间一项，由于是误分类点，可得$2\eta y_i \hat{w}_{k-1} \cdot \hat{x}_i < 0$，又由于$\|\hat{x}_i\| \leqslant R$，所以$\|\hat{w}_{k-1}\|^2 + 2\eta y_i \hat{w}_{k-1} \cdot \hat{x}_i + \eta^2 \|\hat{x}_i\|^2 \leqslant \|\hat{w}_{k-1}\|^2 + \eta^2 R^2$，根据递归，最终可以得到：

$$\begin{aligned}
\|\hat{w}_k\|^2 &\leqslant \|\hat{w}_{k-1}\|^2 + \eta^2 R^2 \\
&\leqslant \cdots \\
&\leqslant \|\hat{w}_0\|^2 + k\eta^2 R^2 \\
&= k\eta^2 R^2
\end{aligned}$$

第二步得证。

现根据上面两步继续推导：根据柯西不等式，得到$\hat{w}_k \cdot \hat{w}_{opt} \leqslant \|\hat{w}_k\| \cdot \|\hat{w}_{opt}\|$
$\because \|\hat{w}_{opt}\| = 1$
$\therefore \hat{w}_k \cdot \hat{w}_{opt} \leqslant \|\hat{w}_k\|$
由第二步可知$\|\hat{w}_k\| \leqslant \sqrt{k\eta R}$
由第一步可知$\hat{w}_k \cdot \hat{w}_{opt} \geqslant k\eta\gamma$
$\therefore k\eta\gamma \leqslant \hat{w}_k \cdot \hat{w}_{opt} \leqslant \|\hat{w}_k\| \leqslant \sqrt{k\eta R}$
$\therefore k\eta\gamma \leqslant \sqrt{k\eta^2 R^2}$
$\therefore k \leqslant \left(\frac{R}{\gamma}\right)^2$

故结论二得证，也就是说误分类次数k是有上界的，随机梯度下降法在有限步之后就可以使得所有的训练集中的实例都正确分类。

6.4 思考

现考虑在结论二的证明中得到的两步结论：

1. $\hat{w}_k \cdot \hat{w}_{opt} \geqslant k\eta\gamma$
2. $\|\hat{w}_k\|^2 \leqslant k\eta^2 R^2$

$\hat{w}_k \cdot \hat{w}_{opt}$内积是逐渐在增大的，因为$k$是逐渐增大的，$\hat{w}_k$的长度是可以被限制住的。

7 第3章-k近邻法-k近邻算法

7.1 k近邻算法

算法3.1：
输入：训练数据集$T = [(x_1, y_1), \cdots, (x_N, y_N)), x_i \in X \subseteq R^n, y_i \in Y = \{c_1, \cdots, c_K\}$，实例特征向量$x$

1. 根据给定的距离度量（欧式距离），在训练集中找到与x最近的k个点，涵盖这个k个点的领域记作$N_k(x)$
2. 在$N_k(x)$中根据分类决策规则（如多数表决）决定x的类别y

输出：实例x所属的类别y

说明：

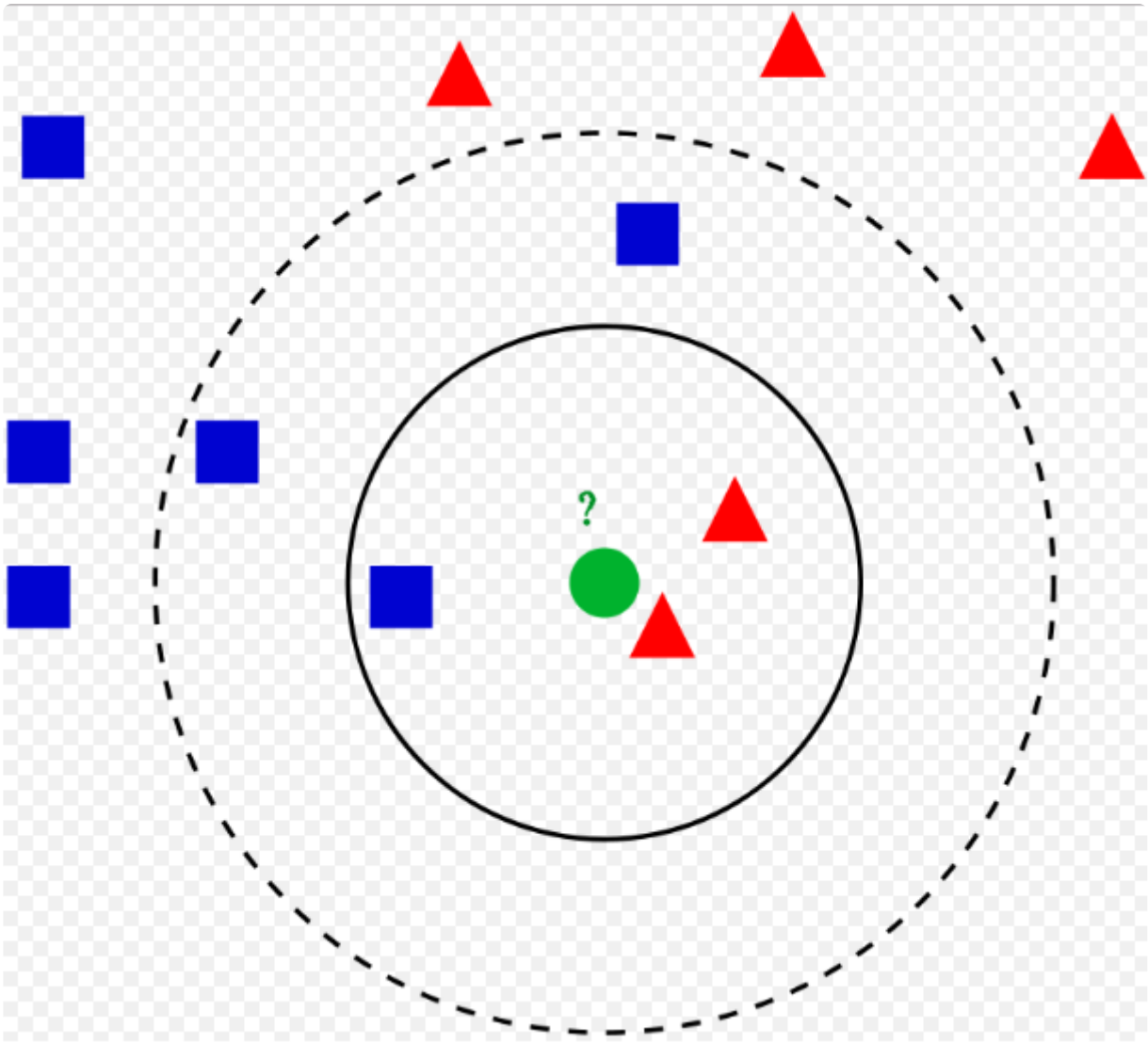

图3.1 KNN算法

对于任何一个统计学习方法来说，都需要有一个训练数据集，图中所有蓝色正方形和红色三角形代表训练集的数据，绿色的圆形表示要预测的数据，在图中，输入变量是一个二维向量（点的坐标），颜色对应了输出变量，训练集中有11个实例，判断绿色圆形属于哪一类（红色或蓝色）。如果k=3（实线圆圈）它被分配给第二类，因为有2个三角形和只有1个正方形在内侧圆圈之内。如果k=5（虚线圆圈）它被分配到第一类（3个正方形与2个三角形在外侧圆圈之内）。

7.2 k近邻模型

7.2.1 模型

k近邻方法没有显式的模型形式

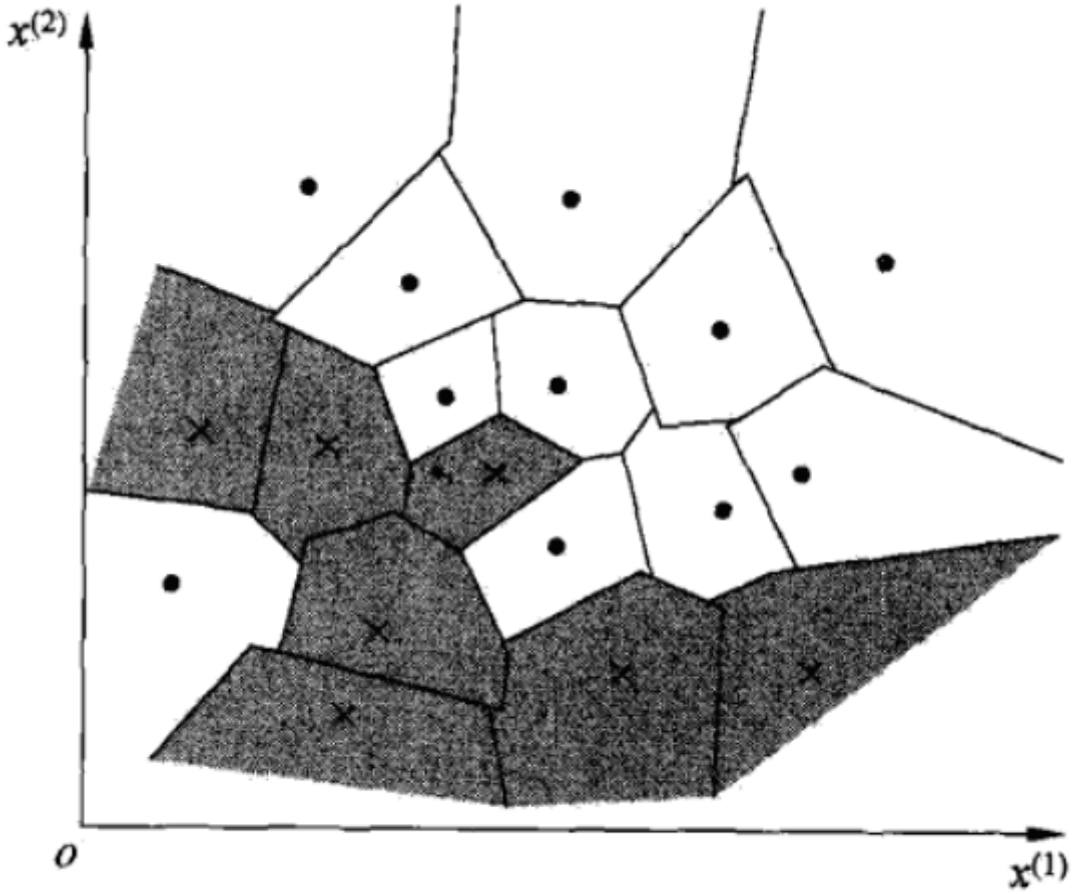

图 3.1 k 近邻法的模型对应特征空间的一个划分

图3.2 KNN模型

上图是一个二分类问题，对应的类别是•和×，在这个例子中，k取值为1，对于每一个要做判断的实例，寻找与其相邻最近的一个点，然后判定为该点的类别。对于每一个训练集的点，都有一个小的多边形区域，在该区域内的点，都属于同一个类别。

$k=1$的时候，该模型称为最近邻模型，该模型相当于在空间中做了一个划分，该划分没有显式的形式，如果要将这个划分写出来，对应的每一个训练集实例都需要很多个超平面方程来定义。对于这种没有显式形式的模型，在做分类的时候，每当得到一个新的实例x，都需要在原始的训练集数据中做搜索。

7.2.2 距离度量

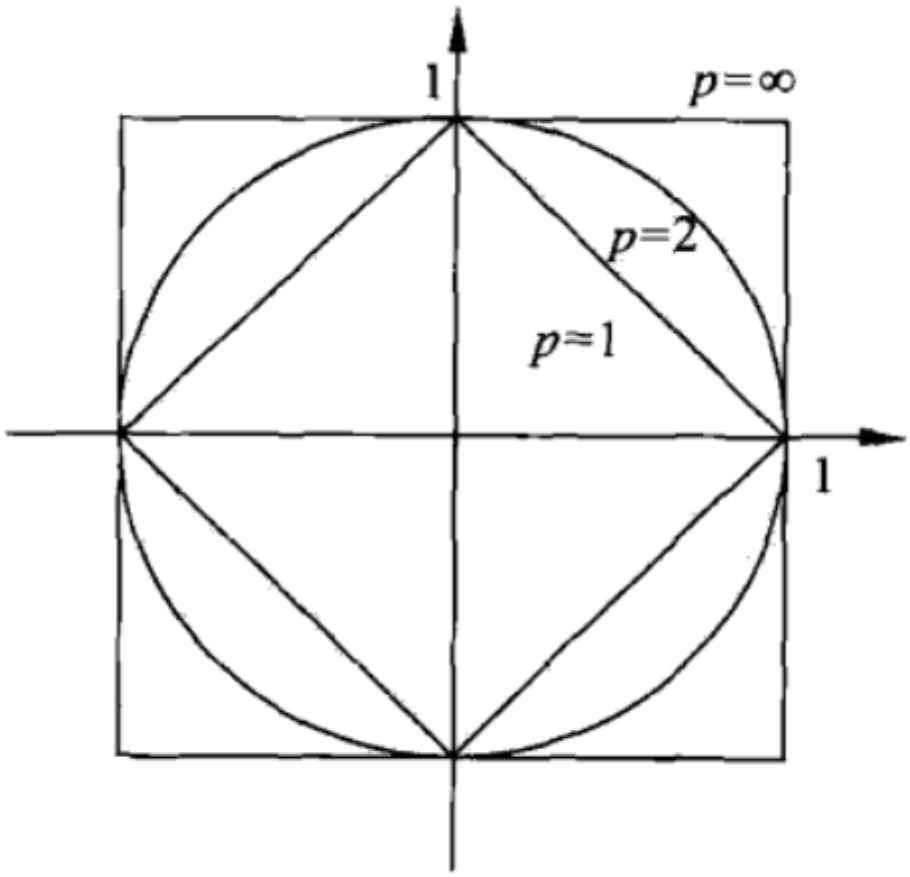

图 3.2 L_p距离间的关系

图3.3 Lp距离间的关系

用p范数$\|x - x_i\|_p$来度量，L_p距离定义如下：

$$L_p(x_i, x_j) = \left(\sum_{l=1}^{n} |x_i^{(l)} - x_j^{(l)}|^p \right)^{\frac{1}{p}}$$

通常p的取值为1，2，∞。

上图中，取原点为基准点，和该点距离相等的点，当取$p = 1$时，在正方形边上的点和原点的距离相等，这就是1-范数；当$p = 2$时，在圆形边上的点和原点距离相等，这就是2-范数；当$p = \infty$时，在最外面正方形边上的点到原点的距离相等，这就是∞-范数。最熟悉的是2-范数，即欧式距离。

那么其他两种意义是什么呢？当$p = 1$时，就是对应点的坐标差值的绝对值之和。当$p = \infty$时，绝对值最大的那个元素，该元素的绝对值就是∞-范数。

7.2.3 k值的选择

在最开始的例子中，已经描述过了（见图3.1），当$k = 3$时（实线圆圈），判定绿色点为红色类，当$k = 5$时（虚线圆圈），判定绿色点为蓝色类。那么k值应该如何选取呢？一般采用交叉验证的方法，将训练数据集分成两类（训练集和验证集），用验证集的实例，给出该实例的特征向量，然后在训练集上选取不同的k值，用k近邻模型，验证实例对应的类别，从而选出最合适的k值。

7.2.4 分类决策规则

对于分类问题，一般采用多数表决，在最近的K个邻近点中，属于哪一类的点最多，就将预测点判定为该类。对应的经验风险为$\frac{1}{k} \sum_{x_i \in N_k(x)} I(y_i \neq c_j)$。

上述公式含义：见图3.1，取$k = 3$（实线圆圈），用唯一的类别给该区域内的点打标签，假如标签为红色，

考察训练集上的损失值，易得$\frac{1}{3}\left(I(y_1 \neq \text{红}) + I(y_2 \neq \text{红}) + I(y_3 \neq \text{红})\right) = \frac{1}{3}$，以上所得为区域中的点判定为红色的损失。当判定为蓝色时，损失为$\frac{1}{3}(I_1 + I_2 + I_3) = \frac{2}{3}$，取损失值最小的类别，得到红色类别，故将该区域所有的点都判定为红色。

7.3 k近邻法的实现：kd树

本节介绍了一个kd树存储数据的方法，对于每一个要预测的实例，都需要寻找和其特征向量距离最近的那些训练集中的实例，当特征向量维度很高时，这种搜索很耗时，这就需要一种特殊的数据结构，不需要遍历训练集中的每一个实例。

7.4 回顾总结

第2章和第3章介绍的都是分类问题的方法，这两种方法都可以用于解决二分类问题，针对输入变量是连续变量。下面对比两种方法的不同：

1. 第2章的感知机模型，需要使用所有的训练集数据，找到一个可以分割两类的超平面，并要求数据是线性可分的，当然对于数据不是线性可分时，也有相应的感知机算法。当找到这个超平面，训练数据集就可以不再需要了，仅根据该超平面，就可以判定新实例的类别。总的来说，感知机模型用全部数据找到了一个全局的结构，保留这个结构并遗弃训练集数据，用该结构预测新实例。
2. 第3章的k近邻算法，并没有用全部数据寻找一个结构，所用的是一个局部的信息，所找的是要预测的这个实例离该点最近的那些点，然后使用这些实例进行预测。当使用局部信息时，需要存储全部的训练数据集。

要如何在这两种算法中选择呢？当数据具有线性可分结构时，用感知机模型更好，因为计算简单，但当数据不是线性可分时，只能使用预测点周围的信息（局部信息）进行分类判定，此时k近邻算法更好，该算法对整个数据集的结构（即模型的结构）没有那么强的假设。

8 第4章-朴素贝叶斯法-导读

朴素贝叶斯法也是用来做分类问题的，其对应的输出变量为$y \in \{c_1, c_2, \cdots, c_K\}$，输入变量为$x_i$（某些离散值），可能的取值个数为$S_i$。

在介绍朴素贝叶斯法之前，先回顾一下第1章，在该章中，把统计学习方法进行了分类，根据模型形式的不同，可以分成决策函数和条件概率分布，在决策函数中，给定一个输入X，通过一个函数f，得到预测值Y，即$Y = f(X)$。在条件概率分布中，给定一个输入X，得到输出Y的条件概率分布$P(Y|X)$，通过该分布，对Y进行决策，如果Y是分类变量时，选取概率最大时所对应的分类。

统计学习方法也可以分成生成模型和判别模型。

- 生成模型：$P(Y|X) = \frac{P(X,Y)}{P(X)}$
- 判别模型：$Y = f(X), P(Y|X)$

8.1 三个分类模型的比较

$$Y = f(x) \Rightarrow P(Y|X) \Rightarrow P(Y|X) = \frac{P(X,Y)}{P(X)}$$

1. 决策函数，首先不考虑X和Y的随机性；Y的条件概率分布，只考虑了Y的随机性，给定X时，Y有一个概率分布；第3种形式，同时考虑了X和Y的随机性，不仅有Y的条件分布、X的边际分布，还有X和Y的联合分布。所以从左到右，考虑的随机性是越来越多的。
2. 第2章-感知机模型，首先该模型属于决策函数的形式。第4章-朴素贝叶斯法，该模型直接从第1种形式到第3种形式，里面用到的最重要的公式是贝叶斯公式。

8.2 朴素贝叶斯法的学习与分类

首先举一个高考升学的例子，考查一个高中生能否考上985大学，其结果只有两种：$Y=\{c_1=\text{考上}, c_2=\text{没考上}\}$（$K$），有如下特征向量：生源地($X^{(1)}$)，是否来自重点中学($X^{(2)}$)。可得$P(X|Y=c_1)$和$P(X|Y=c_2)$两个不同的分布，即给定$Y$的取值，$X$的分布是不同的。

已知以上信息，一个考生，考上和没考上的分布是什么？这个就是Y的分布，其分布为$P(Y=c_1)$和$P(Y=c_2)$。但如果获取信息增加，例如该考生来自北京，此时Y的分布如何（即求$P(Y|X^{(1)}=\text{北京})$）？又该考生来自重点中学，此时的分布是什么（即求$P(Y|X^{(1)}=\text{北京}, X^{(2)}=\text{重点中学})$）？在贝叶斯法中，目的是通过$P(X|Y=c_1)$和$P(X|Y=c_2)$得到$P(Y|X^{(1)}=\text{北京}, X^{(2)}=\text{重点中学})$

> 生成模型：$P(Y=c_k|X=x)=\frac{P(Y=c_k)\cdot P(X=x|Y=c_k)}{P(X=x)}$
>
> 模型假设，条件独立性：$P(x=x|Y=c_k)=\prod_{i=1}^{n}P(X^{(i)}=x^{(i)}|Y=c_k)$
>
> 预测准则，后验概率最大：$y=\arg\max_{c_k}P(Y=c_k|X=x)$

根据上述算法，继续之前的举例：
先求考上的先验概率为$P(X^{(1)}=x^{(1)}, X^{(2)}=x^{(2)}|Y=c_1)$
∵ 在朴素贝叶斯法中，假设各分量之前是独立的。
∴ $P(X^{(1)}=x^{(1)}, X^{(2)}=x^{(2)}|Y=c_1)=P(X^{(1)}=x^{(1)}|Y=c_1)\cdot P(X^{(2)}=x^{(2)}|Y=c_1)$
根据生成模型，由于$P(X=x)$的值不变
∴ 只需要对$P(Y=c_k|X=x)$求取最大值

8.3 朴素贝叶斯法的参数估计

用训练集实例估计$P(Y=c_k), P(X=x|Y=c_k)$，有两种方法：

8.3.1 极大似然估计

首先求出$P(Y=c_1)=\frac{\#\{y_i=c_1\}}{N}$，#表示求个数，N表示整个训练集实例总数。
同样求得$P(Y=c_2)=\frac{\#\{y_i=c_2\}}{N}$ 现求出$P(X^{(1)}=x^{(1)}|Y=c_1)=\frac{\#\{y_i=c_1, X^{(1)}=x^{(1)}\}}{\#\{y_i=c_1\}}$
将上述值代入生成模型贝叶斯公式中，即可求出$P(Y=c_k|X=x)$

在极大似然估计中，可能会出现一种尴尬的情况，在考上的同学中，来自某一个地区的人数可能为0，此时分母$\#\{y_i=c_1\}=0$。出现这种情况的可能是当训练集中的实例比较少，分类类别比较多，就可能出现某一种类别没有对应的实例。所以这个时候，极大似然估计的效果会比较差，可换成贝叶斯估计。

8.3.2 贝叶斯估计

以下用估计Y的概率分布来举例：

为了避免乘积为0，在求取$P(Y=c_1)$时，在分子处加一项λ，同时在分母处也加上一项$K\lambda$，即为

$$P(Y=c_1)=\frac{\#\{y_i=c_1\}+\lambda}{N+K\lambda}, P(Y=c_2)=\frac{\#\{y_i=c_2\}+\lambda}{N+K\lambda}, \cdots, P(Y=c_K)=\frac{\#\{y_i=c_K\}+\lambda}{N+K\lambda}$$

所以$P(X^{(1)}=x^{(1)}|Y=c_1)=\frac{\#\left\{y_i=c_1, X^{(1)}=x^{(1)}\right\}+\lambda}{\#\{y_i=c_1\}+S_i\lambda}$

再讨论一下先验概率和后验概率：

在分类类别Y的分布，其中每一个类别的概率是$P(Y=c_i)=\theta_i$，所以 $\theta_i=1$，需要估计的是$\theta_1, \theta_2, \cdots, \theta_K$的先验分布，该分布叫做Dirichlet（狄利克雷）分布，它是β分布在多维上的一个推广，公式是

$\pi(\theta)=\frac{\Gamma(K\alpha)}{\Gamma(\alpha)^K}\theta_1^{\alpha-1}\theta_2^{\alpha-1}\cdots\theta_k^{\alpha-1}$，这就是$\theta$的分布——Dirichlet分布。

再加上样本信息y_i出现的 ，根据贝叶斯公式，求θ的后验分布，根据最大后验这个准则，就得到$\theta_1,\theta_2,\cdots,\theta_K$的贝叶斯估计。这里面出现的$\lambda$和$\alpha$是相对应的关系，并不是完全相等的关系。

9 第4章-朴素贝叶斯法-后验概率最大化

在朴素贝叶斯法这一章中，书中有一个结论：

> 朴素贝叶斯法将实例分到后验概率最大的类中，这等价于期望风险最小化。

9.1 推导

首先已知损失函数$L(Y,f(X))=\begin{cases}1, & Y\neq f(X)\\ 0, & Y=f(X)\end{cases}$，这里的$f(X)$对应于预测值，$Y$对应于真实值，需要计算的是$f(x)=\arg\max_{c_k} P(Y=c_k|X)$，本来$f(X)$是决策函数的形式，$P(Y=c_k|X)$是条件概率分布的形式，通过这样的一个等式，将两种形式联系在一起。在什么情况下，满足上式呢？当最小化期望风险时，能够满足。

期望风险为$E[L(Y,f(X))]$

$\because$ Y和$f(X)$都是离散的

$$\begin{aligned}\therefore E[L(Y,f(X))]&=\sum_Y\sum_X[L(Y,f(X))P(X,Y)]\\&=\sum_Y\sum_X[L(Y,f(X))P(X)P(Y|X)]\\&=\sum_X\left[[\sum_Y L(Y,f(X))P(Y|X)]P(X)\right]\end{aligned}$$

要最小化$E[L(Y,f(X))]$就需要最小化每一个X的 $_Y\ L(Y,f(X))P(Y|X)$一项。

$\therefore \min E[L(Y,f(X))]\Rightarrow \min\{\ _Y\ L(Y,f(X))P(Y|X)\}$

$\therefore \sum_{c_k} L(Y=c_k,f(x))P(Y=c_k|X)$

中间这项$L(Y=c_k,f(x))$是损失函数，只有当$f(x)\neq c_k$时，取值为1

$$\begin{aligned}\therefore \min\{\sum_Y L(Y,f(X))P(Y|X)\}&=\min\sum_{c_k}\left[I(f(x)\neq c_k)P(Y=c_k|X)\right]\\&=\min\sum_{c_k}\left[[1-I(f(x)=c_k)]P(Y=c_k|X)\right]\\&=\min\left\{\sum_{c_k}P(Y=c_k|X)-\sum_{c_k}\left[I(f(x)=c_k)P(Y=c_k|X)\right]\right\}\end{aligned}$$

$\because \sum_{c_k} P(Y=c_k|X)=1$

$$\begin{aligned}\therefore \min\left\{\sum_{c_k}P(Y=c_k|X)-\sum_{c_k}\left[I(f(x)=c_k)P(Y=c_k|X)\right]\right\}&=\min\left\{1-\sum_{c_k}\left[I(f(x)=c_k)P(Y=c\right.\right.\\&=\max\sum_{c_k}\left[I(f(x)=c_k)P(Y=c_k|X)\right]\end{aligned}$$

9.2 解释

对于每一个随机变量X，都有一个概率分布Y，$X \in \{c_1, c_2, \cdots, c_K\}, y \in \{p_1, p_2, \cdots, p_K\}$，对于$f(x)$而言，只能取其中的一个，假设$f(x) = c_1$时，那么输出$Y$的真实值只以概率$p_1$的可能性是$c_1$，只以概率$p_2$的可能性是$c_2$，...，只以概率$p_K$的可能性是$c_K$，当输出的真实值是$c_1$时，就预测正确，当出现其他类别时，预测错误。

所以当用$f(x)$预测Y时，Y以一定的概率判定为其中的一个类别c_k，对于$c_1, \cdots, c_K$而言，只有其中的一个k使得$I(f(x) = c_k) = 1$，因为$f(x)$只能取其中的一个值。

所以要取最大值，取其中的一项使得$I(f(x) = c_k) = 1$，要使得其最大，找一个c_k使得$P(Y = c_k|X)$最大。

$\therefore f(x) = \arg\max_{c_k} P(Y = c_k|X)$

后验概率最大化得证。

10 第4章-朴素贝叶斯法-贝叶斯估计

本章中对Y的分布的估计，以及Y的条件下X分布的估计，用到了两种估计方法：极大似然估计和贝叶斯估计。本节通过对Y分布的估计来介绍一下极大似然估计和贝叶斯估计得到的结果。

10.1 问题描述

随机变量Y的分布是离散的，可能的取值为$c_1, c_2, \cdots, c_K$，Y属于多项分布（对于每一个类别，Y的取值都有一个对应的概率θ_i），可知 $\theta_i = 1$。

下面用掷硬币的例子，Y出现的结果是正面或反面，正面出现的概率记为θ_1，反面出现的概率记为θ_2，$\theta_1 + \theta_2 = 1$，如果只有两个结果，Y为二项分布，如果有K个结果，叫做多项分布。将K个θ写在一起，记为向量$\hat{\theta}$。

$\therefore Y$的分布为$P(Y = y|\theta) = \theta_1^{I\{y=c1\}} \cdot \theta_2^{I\{y=c2\}} \cdots \theta_K^{I\{y=cK\}}$

10.2 极大似然估计

在极大似然估计中，得到了$y_1, y_2, \cdots, y_N$样本，可得到联合概率分布：其中m_i表示出现结果是c_i的次数

$$\begin{aligned} P(y_1, y_2, \cdots, y_N|\theta) &= P(y_1|\theta)P(y_2|\theta)\cdots P(y_N|\theta) \\ &= \theta_1^{m1}\theta_2^{m2}\cdots\theta_N^{mN} \end{aligned}$$

极大似然估计的思想是通过最大化这个联合概率分布，寻找θ的估计值：

$$\max P(y_1, y_2, \cdots, y_N|\theta) \Rightarrow \max lnP(\cdot) = m_1 \ln\theta_1 + m_2 \ln\theta_2 + \cdots + m_K \ln\theta_K$$

$\because \theta_1 + \theta_2 + \cdots + \theta_K = 1$

为求上式，需要引入拉格朗日乘子：$\max_{\theta_1,\cdots,\theta_K} m_1 \ln\theta_1 + m_2 \ln\theta_2 + \cdots + m_K \ln\theta_K + \lambda(\theta_1 + \cdots + \theta_K - 1)$

对上式求偏导可得：

$$\frac{m_1}{\theta_1}+\lambda=0\Rightarrow\theta_1=-\frac{m_1}{\lambda}$$
$$\frac{m_2}{\theta_2}+\lambda=0\Rightarrow\theta_2=-\frac{m_2}{\lambda}\quad\because\theta_1+\theta_2+\cdots+\theta_K=1$$
$$\vdots$$
$$\frac{m_K}{\theta_K}+\lambda=0\Rightarrow\theta_K=-\frac{m_K}{\lambda}$$

$\therefore -\frac{m_1+m_2+\cdots+m_K}{\lambda}=1\Rightarrow\lambda=-(m_1+m_2+\cdots+m_K)$，将该结果带入前面公式

$$\therefore\begin{aligned}\theta_1&=-\frac{m_1}{\lambda}=\frac{m_1}{N}\\ \theta_2&=-\frac{m_2}{\lambda}=\frac{m_2}{N}\\ &\vdots\\ \theta_K&=-\frac{m_K}{\lambda}=\frac{m_K}{N}\end{aligned}$$

10.3 贝叶斯估计

又回顾掷硬币的例子，结果为$y\in\{$正, 反$\}$，θ的先验概率分布是B（贝塔）分布，正面出现的概率为θ_1，反面出现的概率为θ_2，$p(\theta)=\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta_1^{\alpha-1}\theta_2^{\beta-1}$

如果$y\in\{c_1,\cdots,c_K\}$，对应的概率为$\theta_1,\cdots,\theta_K$，θ的先验概率分布是Dirichlet分布，

$p(\theta)=\frac{\Gamma(\alpha_1+\cdots+\alpha_K)}{\Gamma(\alpha_1)\cdots\Gamma(\alpha_K)}\theta_1^{\alpha-1}\theta_2^{\alpha-2}\cdots\theta_K^{\alpha-K}$

在Beta分布中，当$\alpha=\beta>1$时，$\theta_1=\frac{1}{2}$的概率最大，在先验信息中，认为正面和反面出现的概率，很大的可能是相等的。同样在Dirichlet分布中，$\alpha_1=\alpha_2=\cdots=\alpha_K=\alpha$，也满足如上假设。

获得样本之后，求取θ的后验概率分布

$$\begin{aligned}p(\theta|y_1,\cdots,y_N)&=\frac{p(\theta,y_1,\cdots,y_N)}{p(y_1,\cdots,y_N)}\\ &\propto p(\theta)p(y_1,\cdots,y_N|\theta)\\ &\propto\theta_1^{\alpha-1}\theta_2^{\alpha-1}\cdots\theta_K^{\alpha-1}\cdot\theta_1^{m1}\theta_2^{m2}\cdots\theta_K^{mK}\\ &\propto\theta_1^{m1+\alpha-1}\theta_2^{m2+\alpha-1}\cdots\theta_K^{mK+\alpha-1}\end{aligned}$$

上述是一个Dirichlet分布，为什么会是这个分布呢？正如看到$e^{-\theta^2+a\theta+b}$这种形式，是正态分布的感觉是一样的。

根据上面的公式，可以写出系数：$\frac{\Gamma(m_1+m_2+\cdots+m_K+K\alpha)}{\Gamma(m_1+\alpha)\Gamma(m_2+\alpha)\cdots\Gamma(m_K+\alpha)}$

$\therefore p(\theta|y_1,\cdots,y_N)=\frac{\Gamma(m_1+m_2+\cdots+m_K+K\alpha)}{\Gamma(m_1+\alpha)\Gamma(m_2+\alpha)\cdots\Gamma(m_K+\alpha)}\theta_1^{m1+\alpha-1}\theta_2^{m2+\alpha-1}\cdots\theta_K^{mK+\alpha-1}$

由于计算贝叶斯估计是计算后验概率最大值，不需要如此精确的值，所以计算的时候去掉前面的系数，即$\max\theta_1^{m1+\alpha-1}\theta_2^{m2+\alpha-1}\cdots\theta_K^{mK+\alpha-1}$

最大后验结果为：

$$\begin{aligned}\theta_1&=\frac{m_1+\alpha-1}{m_1+m_2+\cdots+m_K+K(\alpha-1)}\\ &=\frac{m_1+\alpha-1}{N+K\alpha-K}\end{aligned}$$

易得：$\theta_2=\frac{m_2+\alpha-1}{N+K\alpha-K}$，后面省略。

所以对应书上第51页中：$\lambda=\alpha-1$

10.4 总结

这一部分所讲的内容，属于书上的扩展内容，比较超纲，如果没有看懂，也没有关系，可以尝试地推导一下，如果学有余力，可以学习扩展内容：LDA(Latent Dirichlet Alocation)模型，这个模型的基础就是Dirichlet分布和贝叶斯框架，感兴趣的同学可以查一下相关资料。

11 第5章-决策树-导读

本章书上的内容看起来很多，但其实决策树是一个比较简单，而且容易可视化的统计学习模型。决策树模型既可以用来做分类问题，也可以用来做回归问题，书中主要讲的是分类问题。本章 模型的特点是输出变量Y是一个分类变量，输入变量X主要也是分类变量。

11.1 决策树模型与学习

表 5.1 贷款申请样本数据表

ID	年龄	有工作	有自己的房子	信贷情况	类别
1	青年	否	否	一般	否
2	青年	否	否	好	否
3	青年	是	否	好	是
4	青年	是	是	一般	是
5	青年	否	否	一般	否
6	中年	否	否	一般	否
7	中年	否	否	好	否
8	中年	是	是	好	是
9	中年	否	是	非常好	是
10	中年	否	是	非常好	是
11	老年	否	是	非常好	是
12	老年	否	是	好	是
13	老年	是	否	好	是
14	老年	是	否	非常好	是
15	老年	否	否	一般	否

表5-1 贷款申请样本数据表

该表一共有15条数据（实例），“类别”为输出变量Y，是一个二分类，分别对应“否”和“是”，“否”表示此人贷款申请失败（不被批准），“是”表示此人贷款申请成功（被批准），对于每一个示例，有4个输入变量，即X是4维的，年龄分为青年、中年、 老牛，有工作分为是和否，有自己的房子分为是和否，信贷情况分为一般、好、非常好。

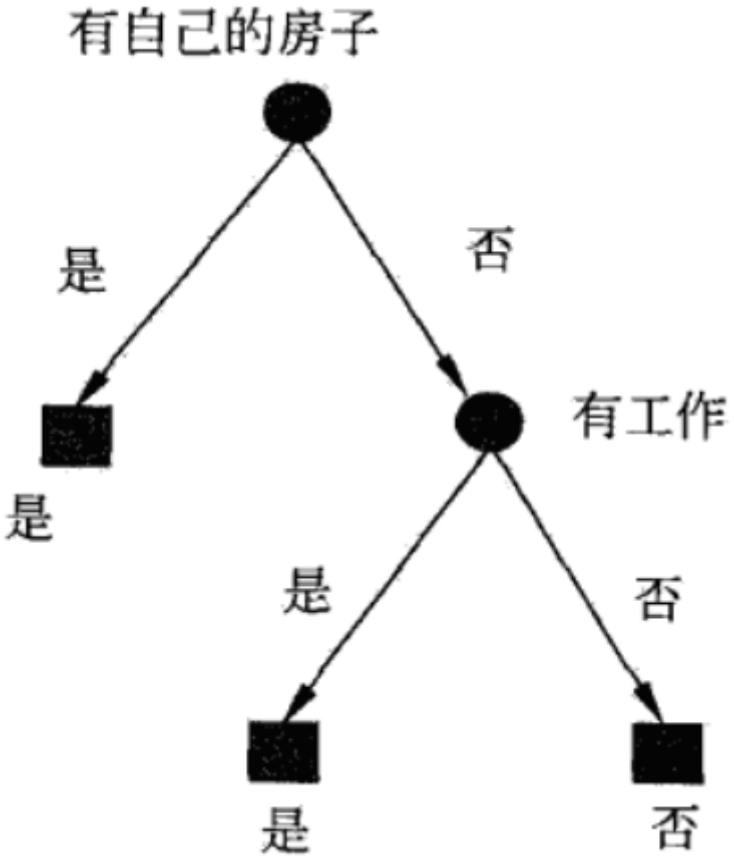

图5-1 决策树生成

决策树模型最后要得到的为上面这个图，根据上图，任意用一个实例，可以判定数据的预测类别。图中有以下几个术语：

- 根结点：图中“有自己的房子”这个结点为根结点。
- 内部结点：图中“有工作”这个结点为内部结点。
- 叶子结点：图中“是”和“否”的方形结点为叶子结点。

根据训练集中的输入变量，首先找到一个根结点，通过根结点判断所属类别，分成不同的分支，在分支下面继续寻找下一个用来分类的输入变量（即内部结点），以上是决策树模型类别判定过程描述。

虽然决策树模型有非常好的可视化性质，但是也有一些问题，需要通过哪些准则来寻找根结点？当寻找根结点时，直观上说，要一个分类效果最好，可以通过特征选择。树状结构停止的标准是什么？其实并不需要将训练实例中所有的特征都放在决策树中，假设有非常多的特征，如果将它们都放入决策树中，决策树中对应的每一个叶子结点会对应于训练集中的每一个实例，也就是说，该决策树在训练集上表现很好，很容易过拟合。

决策树中比较重要的两个问题：第一个是如何选择分类的特征，第二个是如何避免过拟合。

11.2 特征选择

11.2.1 熵

熵：$H(X) = -\sum_{i=1}^{n} p_i \log p_i$

在本章中，讨论的都是离散型的随机变量（即可分类的），对于分类变量，假如有n个不同的类别，每一个类别出现的概率是p_i，熵为$H(X) = -\sum_{i=1}^{n} p_i \log p_i$。其中$\log$是以2为底的对数，熵衡量的是随机变量的混乱程度。

现根据前面的例子，可以看到有15个人的数据，在最终的分类变量Y上的离散程度。观察类别：一共出现了

6个“否”，9个“是”；由于p_i是由样本求出的，H(X)也称为经验熵。

15个实例在Y上面的经验熵为：$-\left(\frac{6}{15}\log\frac{6}{15}+\frac{9}{15}\log\frac{9}{15}\right)$

现选择一个特征，将数据分组，希望的目标是按照此特征，分组出来的数据全部都是“是”，另一组全部都是“否”，计算两个分组的经验熵：$-1\log 1=0$，两组的经验熵都是0。但是很难选择一个特征，将两类数据完全分开，这样就需要对比每一个特征对样本数据的混乱程度（熵）。

以下用一个特征举例，比如“有工作”这个特征，按照这个特征将数据集分成两个部分，有工作的有5人，最后申请贷款成功的有5人，没有工作的有10人，最后申请贷款成功的有4人，申请贷款失败的有6人。条件熵为

$$\frac{5}{10}\times -1\log 1+\frac{10}{15}\times\left(-(\frac{6}{10}\log\frac{6}{10}+\frac{4}{10}\log\frac{4}{10})\right)$$

11.2.2 信息增益

信息增益：$g(D,A)=H(D)-H(D|A)$

给定A的条件下D的熵$H(D|A)$，之前没有对数据进行分类，经验熵为$H(D)$，现在根据“有工作”这个特征将数据分类，在这一过程中，数据混乱的减少程度，叫做信息增益。

当选择第一个根结点时，在所有的特征中，寻找一个信息增益最大的特征，作为根结点，一步一步地从剩下的特征中选择信息增益最大的，作为下一个内部结点。虽然该特征选择方法比较好理解，但是有一个缺点，如果有一个特征A，其类别非常多，现有15个数据实例，特征A有15个不同的类别，导致会有15个分支，对应结点的熵为0，条件熵为0，信息增益会很大，很容易出现过拟合。针对这个情况，对信息增益做了改进，采用信息增益比来选择特征。

11.2.3 信息增益比

信息增益比：$g_R(D,A)=\frac{g(D,A)}{H_A(D)}$

根据上述公式，可以看到分子是信息增益，而分母是$H_A(D)$，考虑到了特征A本身的混乱程度（熵）。根据上一个例子，将样本数据按照“有工作”分成了两类，有工作为5个，没工作有10个，可得

$H_A(D)=-(\frac{5}{15}\log\frac{5}{15}+\frac{10}{15}\log\frac{10}{15})$，类别分得越多，混乱程度越大（即熵值越大）。

11.3 决策树的生成

11.3.1 算法5.2（ID3算法）

输入：训练数据集D，特征集A，阈值ε

输出：决策树T

（1）若D中所有实例属于同一类C_k，则T为单结点树，并将类C_k作为该结点的类标记，返回T；

（2）若$A\neq\emptyset$，则T为单结点树，并将D中实例数最大的类C_k作为该结点的类标记，返回T；

（3）否则，按算法5.1计算A中各特征对D的**信息增益**，选择**信息增益**最大的特征A；

（4）如果A_g的信息增益小于阈值ε，则置T为单结点树，并将D中实例数最大的类C_g作为该结点的类标记，返回T；

（5）否则，对A_g的每一个可能值a_i，依$A_g=a_i$将D分割为若干非空子集D_i，将D_i中实例数最大的类作为标记，构建子结点，由结点及其子结点构成树T，返回T；

（6）对第i个子结点，以D_i为训练集，以$A-\{A_g\}$为特征集，递归地调用（1）~（5），得到子树T_i，返回T_i。

11.3.2 算法5.2（C4.5算法）

输入：训练数据集D，特征集A，阈值ε

输出：决策树T

（1）若D中所有实例属于同一类C_k，则T为单结点树，并将类C_k作为该结点的类标记，返回T；

（2）若$A \neq \emptyset$，则T为单结点树，并将D中实例数最大的类C_k作为该结点的类标记，返回T；

（3）否则，按公式$g_R(D, A) = \dfrac{g(D, A)}{H_A(D)}$计算$A$中各特征对$D$的**信息增益比**，选择**信息增益比**最大的特征A；

（4）如果A_g的信息增益小于阈值ε，则置T为单结点树，并将D中实例数最大的类C_g作为该结点的类标记，返回T；

（5）否则，对A_g的每一个可能值a_i，依$A_g = a_i$将D分割为若干非空子集D_i，将D_i中实例数最大的类作为标记，构建子结点，由结点及其子结点构成树T，返回T；

（6）对第i个子结点，以D_i为训练集，以$A - \{A_g\}$为特征集，递归地调用（1）~（5），得到子树T_i，返回T_i。

11.4 决策树的剪枝

前面两个算法都通过限定阈值，避免生成的决策树很深，造成过拟合。这种方法叫做预剪枝，也就是说，在树的生成过程中，已经考虑了不生成很宽很深的树，有一些分支被剪掉了。书中5.4节，所讲的是后剪枝，当生成整个树之后，如何再砍掉一些分支，让决策树更精简，增强泛化能力。

决策树损失函数：$C_\alpha(T) = C(T) + \alpha|T|$，其中$C(T) = \sum_{t=1}^{|T|} N_t H_t(T)$

说明： α表示对树规模的一个惩罚；$|T|$表示这个决策树有多少个叶子结点，例如图5-1中，叶子结点有3个，当叶子结点数越大，说明决策树模型越复杂，其泛化能力差，$\alpha|T|$表示模型的复杂度；$C(T)$表示模型对训练集数据的拟合程度，用熵来度量，当叶子结点上面的条件熵越小时，所对应的损失越小，熵越小，代表数据越整齐，混乱程度越低，当模型用于分类的时候，分类效果更好。

首先计算原始$C_\alpha(T)$，再剪枝，计算剪枝之后的$C_\alpha(T')$，如果$C_\alpha(T') < C_\alpha(T)$，则可以剪枝，如果$C_\alpha(T') \geqslant C_\alpha(T)$，则不可以剪枝。

11.5 CART算法

这个算法对应的就是一个二叉树，上面两种算法（ID3和C4.5）可以按照类别个数进行分类，但是这个算法，不论特征有多少类别，都划分成两类。依然以之前的贷款申请为例，按照“年龄”，有“青年”、“中年”、“老年”，但按照该算法，可分为“青年”和“非青年”，或者为“中年”和“非中年”。那么要如何选择类别呢？提出了一个新的判别方法，称为基尼指数。

基尼指数：$\mathrm{Gini}(p) = \sum_{k=1}^{K} p_k(1 - p_k)$

在特征A的条件下，集合D的基尼指数：$\mathrm{Gini}(D, A) = \dfrac{|D_1|}{|D|}\mathrm{Gini}(D_1) + \dfrac{|D_2|}{|D|}\mathrm{Gini}(D_2)$

在CRAT算法中，选择$\mathrm{Gini}(D, A)$最小的特征来进行决策树的分支，这个时候就不用再考虑特征A对于集合D的混乱程度，因为在CRAT算法中，只划分两类，不会导致信息增益很大。

12 第5章-决策树-信息增益与基尼指数

12.1 从定性的角度了解熵与基尼指数的含义

根据公式：

- 熵：$H(x) = -\Sigma p_i \ln p_i$
- 基尼指数：$Gini(p) = \sum_i p_i(t - p_i)$

这两个指标都是衡量一个随机变量的离散程度（混乱程度），提到离散程度，比较熟悉的是方差。

例如随机变量样本值为$x_1, x_2, \cdots, x_n$，方差为$\frac{1}{n}\sum_{i=1}^{n}(x_i - x)^2$，如果样本中有两个或者多个相同时，另记$x_j$，对于$j$而言，有$n_j$个样本是相同的取值，则方差为$\sum_j \frac{n_j}{n}(x_j - x)^2$，记$p_j = \frac{n_j}{n}$表示$x_j$对应的样本出现的频率，方差为$\sum_j (x_j - x)^2 p_j$，这种形式一般用于$x$是连续的随机变量，或者虽然$x$为离散的随机变量，但是其取值的大小是有一定含义的，所以在衡量这个离散程度时，会考虑x_j到其均值的距离，但是对于熵而言，熵衡量的是离散随机变量，这些随机变量取值的大小是没有意义的。

所以对于离散随机变量，一般采用熵或者基尼指数来作为衡量离散程度（混乱程度）的指标。

12.2 混乱程度的理解

举一个二分类的例子：掷硬币，投掷硬币以$\frac{1}{2}$的概率出现正面，以$\frac{1}{2}$的概率出现反面，当预测掷硬币出现的结果时，很不确定的判断掷硬币的结果。假设有一个非常不均匀的硬币，已知出现正面的概率为$\frac{8}{9}$，出现反面的概率为$\frac{1}{9}$，这个时候，正面出现的概率占了绝对的优势，此时预测掷硬币的结果，很有信心确定本次结果是正面，表明该信息没有那么混乱。

12.3 关系图说明

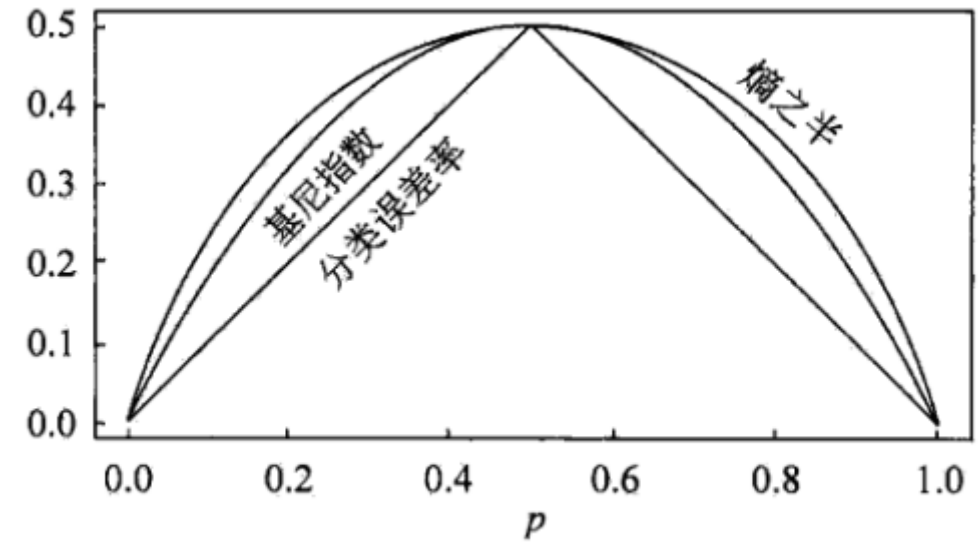

图 5.7　二类分类中基尼指数、熵之半和分类误差率的关系

图5-3 二类分类中基尼指数、熵之半和分类误差率的关系

如果把出现正面的概率记为p，观察图5-3，当$p = 0$（掷硬币出现反面）或$p = 1$（掷硬币出现正面）时，属于确定性事件，此时没有任何混乱，对应的基尼指数和熵之半都为0；随着p逐渐变大到0.5时，当前信息是最混乱的，对应的基尼指数和熵之半都为0.5；当p从0.5逐渐变大到1时，也就是出现正面的概率要超过0.5，并且越来越大，此时信息又逐渐不混乱，所以基尼指数和熵之半在减小。

图中还出现了另一个指标：分类误差率，当$p = 0.5$时，预测硬币出现是正面的事件概率为0.5，也就是说有50%的可能预测正确或错误；当出现正面的概率是0.3时，即$p = 0.3$，此时肯定会预测硬币出现结果为反面，预测错误的概率为0.3，所以当$p = 0.3$时，分类误差率为0.3；当$p = 0.8$是，出现正面的概率是0.8，此时肯定会预测硬币出现结果为正面，预测错误的概率为0.2，所以当$p = 0.8$时，分类误差率为0.2。

13 第6章-Logistic回归与最大熵模型-导读

本章一共讲了两个模型：Logistic回归与最大熵模型，这两个模型都是用来做分类问题的，具有以下几个共同点：（1）它们都属于概率模型，该模型要寻找的是给定一个x，得到输出变量Y的概率分布$P(Y|x)$，如果是二分类，Y取值为0或1，如果是多分类，Y有K个不同的类别。（2）它们都属于对数线性模型，对概率分布$P(Y|x)$取对数，可得$\ln P(Y|x) = w \cdot x$关于x的线性函数，如果$Y = c_i$，则$w = w_i$。两个模型之前的区别是Logistic回归属于判别模型，最大熵模型属于生成模型。在最大熵模型中，不仅x有随机性，Y也具有随机性，是一个随机变量。

13.1 Logistic回归模型

13.1.1 二项Logistic回归模型

给定输入变量x，输出变量为$y \in \{0, 1\}$，将$y = 1$的概率记作$\pi(x) = P(Y = 1|x) \in [0, 1]$，上述已经介绍了，这个模型是一个线性模型，可以用$w \cdot x$线性 函数来表示，由于$w \cdot x \in (-\infty, +\infty)$，那么要如何将$\pi(x)$与$w \cdot x$对应起来呢？这就需要用到一个变换，该变换称为Logit变换。

Logit变换：$\text{logit}(\pi(x)) = \log \dfrac{\pi(x)}{1 - \pi(x)} \in (-\infty, +\infty)$

$\therefore \ln \dfrac{\pi(x)}{1 - \pi(x)} = w \cdot x$，这个就是Logistic回归模型的一个形式。

$\therefore \pi(x) = \dfrac{\exp(w \cdot x)}{1 + \exp(w \cdot x)}$，其中$\pi(x)$就是给定$x$的条件下，$Y = 1$的概率。

所以可得下面两个公式：$\log \dfrac{P(Y = 1|x)}{1 - P(Y = 1|x)} = w \cdot x$

$$P(Y = 1|x) = \frac{\exp(w \cdot x)}{1 + \exp(w \cdot x)}$$

有了这个模型之后，需要求解参数w，一旦求出w，那么任意给定一个输入变量x，就可以得到$Y = 1$的概率，如果该概率值大于0.5，就将该类判定为1，如果小于0.5，将该类判定为0。

求解w使用的方法是**极大似然估计**，给定参数w，求样本的联合概率密度，最大化该联合概率，从而求出参数w。

13.1.2 二项Logistic回归模型参数估计

已知$y_i \in \{0, 1\}$，$\pi(x)$

可得$P_w(y|x) = \pi(x)^y[1 - \pi(x)]^{1-y}$

似然函数为$L(w) = \prod_{i=1}^{N} \pi(x)^{y_i}[1 - \pi(x)]^{1-y_i}$

最大化似然函数：

$$\begin{aligned} \max \ln L(w) &= \sum_{i=1}^{N} \left\{ y_i \ln \pi(x) + (1 - y_i) \ln \left[1 - \pi(x)\right] \right\} \\ &= \sum_{i=1}^{N} \left\{ y_i (w \cdot x_i) - \ln \left[1 + \exp(w \cdot x_i)\right] \right\} \end{aligned}$$

使用梯度下降法（见第2章-感知机模型-随机梯度下降法），其中求解过程中略有区别，在感知机模型中，求解的是极小值min，给定w的初值，是向着w的$-\nabla$的方向更新的；但是在Logistic回归模型中，求解的是极大值max，给定w的初值，需要向着w的$+\nabla$的方向更新。

13.1.3 多项Logistic回归模型

在二分类Logistic回归模型中，$\ln \frac{\pi(x)}{1-\pi(x)} \Rightarrow \ln \frac{P(Y=1|x)}{P(Y=0|x)}$。

在多分类Logistic回归模型中，Y取值为$1, \cdots, K$，那么Logit变换是$\ln \frac{p_(Y=k|x)}{P(Y=K|x)} = w_k \cdot x \in (-\infty, +\infty)$，因为$k$的取值为$1, \cdots, K-1$，所以$w$变为$w_k$，一共需要求取$K-1$个参数向量。

所以可得下面两个公式：

$$P(Y=k|x) = \frac{\exp(w_k \cdot x)}{1 + \sum_{k=1}^{K-1} \exp(w_k \cdot x)}, k = 1, \cdots, K-1$$

$$P(Y=K|x) = \frac{1}{1 + \sum_{k=1}^{K-1} \exp(w_k \cdot x)}$$

13.2 最大熵模型

最大熵模型在自然语言处理中是一个很常用的模型。

最大熵模型原理：在满足约束条件的模型集合中选择熵最大的模型。

之前介绍过熵的概念，表示数据的混乱程度。为什么要选择熵最大？因为在不知道任何信息的情况下，假设数据在各个取值上面是比较平均的（即比较混乱的）。

书上例题6.1：假设随机变量X有5个取值$\{A, B, C, D, E\}$，要估计取各个值的概率$P(A), P(B), P(C), P(D), P(E)$。

解答：

可知$P(A) + P(B) + P(C) + P(D) + P(E) = 1$，$P(k) \geqslant 0, k \in \{A, B, C, D, E\}$

根据最大熵原理，可得$P(A) = P(B) = P(C) = P(D) = P(E) = \frac{1}{5}$

有时，能从一些先验知识中得到一些对概率值的约束条件，例如：

$$P(A) + P(B) = \frac{3}{10}$$

$$P(A) + P(B) + P(C) + P(D) + P(E) = 1$$

可以写成上述问题的优化问题：

$$\min \quad -H(P) = \sum_{i=1}^{5} P(y_i) \log P(y_i)$$

$$\text{s.t.} \quad P(y_1) + P(y_2) = \tilde{P}(y_1) + \tilde{P}(y_2) = \frac{3}{10}$$

$$\sum_{i=1}^{5} P(y_i) = \sum_{i=1}^{5} \tilde{P}(y_i) = 1$$

条件熵是给定一个随机变量x，在已知x的信息下，求随机变量y_i的混乱程度。可得$H(y|x) = -\sum_{y} P(y|x) \ln P(y|x)$，由于在所有训练数据集中$x$的取值是不一样的，只需要求解$x$关于$y$的期望，可得$H(P) = E_x H(y|x) = -\sum_{x,y} P(x)P(y|x) \ln P(y|x)$，其中$P(x)$可以用样本上的分布$\tilde{P}(x)$进行替换，所以可得如下公式：

$$H(P) = -\sum_{x,y} \tilde{P}(x)P(y|x) \log P(y|x)$$

特征函数：

$$f(x, y) = \begin{cases} 1, x\text{与}y\text{满足某一事实} \\ 0, \text{否则} \end{cases}$$

下面举例说明：
以英汉翻译为例，对于英语中的“take”，它对应汉语的翻译有：

- “抓住”: The mother takes her child by the hand. 母亲抓住孩子的手。
- “拿走”：Take the book home. 把书拿回家。
- “乘坐”：to take a bus to work. 乘坐公共汽车上班。
- “量”: Take your temperature. 量一量你的体温。
- “装”：The suitcase wouldn't take another thing. 这个衣箱不能装别的东西了。
- “花费”：It takes a lot of money to buy a house. 买一所房子要花一大笔钱。
- “理解、领会”：How do you take this package? 你怎么理解这段话？

以上可知这是Y的不同类别，训练集中的实例x就是说的话（英语），对应的y是对应汉语的翻译，这些实例对应的翻译y是不一样的，从已经获得的样本中发现，如果take后面有一个单词bus，那就将take翻译为“乘坐”，也就是说$y=$ 乘坐，x中的take后面有一个单词bus。这就是观察到的一个特征，然后用特征函数来描述这个信息。

可得到特征函数：

$$f(x,y)=\begin{cases}1, \text{if} & y=\text{“乘坐” and next(x)} = \text{“bus”}\\ 0\end{cases}$$

会观察到很多这样的信息，一般用$f_i(x,y)$来表示，其中$i=1,\cdots,n$，将上述函数加到最大熵的约束条件中，让该特征在训练集实例（即样本）上出现的概率等于该特征在总体中出现的概率，那要如何用数学语言描述呢？因为$f_i(x,y)$是一个二值函数，在总体中出现的概率可以用期望$E_{P(x,y)}(f_i(x,y))$表示，可得：

$$E_{P(x,y)}(f_i(x,y))=E_{\tilde{P}(x,y)}(f_i(x,y))$$

最大熵模型对应的最优化问题：

$$\begin{aligned}\max_{P\in C}\quad & H(P)=-\sum_{x,y}\tilde{P}(x)P(y|x)\log P(y|x)\\ \text{s.t.}\quad & E_P(f_i)=E_{\tilde{P}}(f_i),\quad i=1,2,\cdots,n\\ & \sum_y P(y|x)=1\end{aligned}$$

求出的结果是$P_w(y|x)=\frac{1}{z_w(x)}\exp\Big(\sum_{i=1}^{n}w_i f_i(x,y)\Big)$，其中$w=\arg\max L_{\tilde{P}}(P_w)=\log\prod_{(x,y)}P(y|x)^{\tilde{P}(x,y)}$

直观理解：给定输入变量x，分别求解y取不同值时，对应的概率分布，$\sum_{i=1}^{n}w_i f_i(x,y)$表示$n$个特征在给定的$(x,y)$上出现的次数，$(x,y)$满足几个特征，$w_i$表示该特征的重要程度，当满足的特征越多，并且这些特征越重要，$P_w(y|x)$的概率越大。

13.3 模型学习的最优化算法

改进的迭代尺度法：求最大化对数似然函数，该对数似然函数为

$L(w)=\sum_{x,y}\tilde{P}(x,y)\sum_{i=1}^{n}w_i f_i(x,y)-\sum_x\tilde{P}(x)\log Z_w(x)$，其中$Z_w(x)$是归一化的系数，为了保证给定$x$的条件下$y$的概率和等于1。

拟牛顿法：

$$\min_{w\in R^n} f(w)=\sum_x \tilde{P}(x)\log\sum_y \exp\Big(\sum_{i=1}^n w_i f_i(x,y)\Big)-\sum_{x,y}\tilde{P}(x,y)\sum_{i=1}^n w_i f_i(x,y)$$

13.4 总结

最大熵模型，和之前介绍的模型有一个很明显的区别：在之前的模型中，直接用输入变量X向量中每一个值的信息，但是在最大熵模型中，采用的是x和y之间的特征关系，并不是直接用x的取值。

14 第6章-Logistic回归与最大熵模型-改进的迭代尺度法

14.1 改进的迭代尺度法

最大熵模型的形式是一个指数形式，需要对w进行参数估计，求解w的方法是最大似然估计，其似然函数为

$$L(w)=\sum_{x,y}\Big[\tilde{P}(x,y)\sum_{i=1}^n w_i f_i(x,y)\Big]-\sum_x\Big[\tilde{p}(x)\ln z_w(x)\Big]$$

其中$\tilde{P}(x,y)$是x和y的经验分布，根据训练集当中特定的x和y的个数占总训练集实例的比值；w_i为所求，一共需要求解n个w；$f_i(x,y)$是已经给定的特征函数，取值为0或1；$Z_w(x)$表示关于给定x的y条件分布的归一化系数，$Z_w(x)=\sum_y \exp\Big[\sum_i w_i f_i(x,y)\Big]$，在该公式中，也存在$w$。

这个函数$L(w)$是关于w的，函数形式比较复杂，有指数上面的w并还要取对数，直接对其求导比较难求，使用了迭代的方法求导。首先给w初值，然后更新w，使$L(w)$的值不断增大，从而求得$L(w)$的最大值。

14.2 求解最大似然函数

假设存在一个δ，使得$w\to w+\delta$。对于给定的经验分布$\tilde{P}(x,y)$，模型参数从w到$w+\delta$，对数似然函数的改变量为：

$$L(w+\delta)-L(w)=\sum_{x,y}\tilde{P}(x,y)\sum_{i=1}^n\delta_i f_i(x,y)-\sum_x\tilde{P}(x)\ln\frac{Z_{w+\delta}(x)}{Z_w(x)}$$

通过更新w变为$w+\delta$，使得似然函数变大，变大的值为上述公式，故求解等号后面的最大值。

$\because -\ln\alpha\geqslant 1-\alpha,\alpha>0$

仅观察这项：$-\sum_x\tilde{P}(x)\ln\frac{Z_{w+\delta}(x)}{Z_w(x)}\geqslant\sum_x\tilde{P}(x)\Big[1-\frac{Z_{w+\delta}(x)}{Z_w(x)}\Big]$

$$=1-\sum_x\tilde{P}(x)\frac{Z_{w+\delta}(x)}{Z_w(x)}$$

$$\because \frac{Z_{w+\delta}(x)}{Z_w(x)}=\frac{1}{Z_w(x)}\cdot\sum_y\exp\Big(\sum_{i=1}^n(w_i+\delta)f_i(x,y)\Big)$$

$$=\frac{1}{Z_w(x)}\cdot\sum_y\exp\Big[\sum_{i=1}^n w_i f_i(x,y)+\sum_{i=1}^n\delta_i f_i(x,y)\Big]$$

$$=\frac{1}{Z_w(x)}\cdot\sum_y\Big[\exp\sum_{i=1}^n w_i f_i(x,y)\cdot\exp\sum_{i=1}^n\delta_i f_i(x,y)\Big]$$

$$=\sum_y\frac{1}{Z_w(x)}\Big[\exp\sum_{i=1}^n w_i f_i(x,y)\cdot\exp\sum_{i=1}^n\delta_i f_i(x,y)\Big]$$

$\because$最大熵模型为$P_w(y|x)=\frac{1}{Z_w(x)}\exp\Big(\sum_{i=1}^n w_i f_i(x,y)\Big)$

$$\therefore \frac{Z_{w+\delta}(x)}{Z_w(x)} = \sum_y \frac{1}{Z_w(x)} \Big[\exp \sum_{i=1}^{n} w_i f_i(x,y) \cdot \exp \sum_{i=1}^{n} \delta_i f_i(x,y) \Big]$$

$$= \sum_y \Big[P_w(y|x) \exp \Big(\sum_{i=1}^{n} \delta_i f_i(x,y) \Big) \Big]$$

将上式代入并整理：$L(w+\delta) - L(w) = \sum_{x,y} \tilde{P}(x,y) \sum_{i=1}^{n} \delta_i f_i(x,y) - \sum_x \tilde{P}(x) \ln \frac{Z_{w+\delta}(x)}{Z_w(x)}$

$$\geqslant \sum_{x,y} \tilde{P}(x,y) \sum_{i=1}^{n} \delta_i f_i(x,y) + 1 - \sum_x \tilde{P}(x) \sum_y \Big[P_w(y|x) \exp \Big(\sum_{i=1}^{n}$$

$$= A(\delta|w)$$

目前得到了改变量的下界，如果想让该值最大，就最大化$A(\delta|w)$值。

$\because (e^{\ \delta_i f_i})' = e^{\ \delta_i f_i} \cdot f_i$，求导之后依然有其他的$\delta_i$分量，但是希望对$\delta_i$求导之后能得到只关于$\delta_i$的函数，使得$g(\delta_i) = 0$，需要对$\exp \Big(\sum_{i=1}^{n} \delta_i f_i(x,y) \Big)$再进行变换，需要用到Jesson不等式。

> **Jesson不等式：**
> 对一个凸函数$\phi(x)$，已知权重a_i，　$a_i = 1$ ，下列不等式成立：
> $$\phi(\sum_i a_i x_i) \leqslant \sum_i a_i \phi(x_i)$$

根据Jesson不等式，可得：

$$\exp \Big(\sum_i \delta_i f_i(x,y) \Big) = \exp(\sum_i \frac{f_i(x,y)}{f^{\#}(x,y)} f^{\#}(x,y) \delta_i)$$

$$\leqslant \sum_i \frac{f_i(x,y)}{f^{\#}(x,y)} \exp(f^{\#}(x,y) \delta_i)$$

$$\therefore A(\delta|w) \geqslant \sum_{x,y} \tilde{P}(x,y) \sum_{i=1}^{n} \delta_i f_i(x,y) + 1 - \sum_x \tilde{P}(x) \sum_y \Big[P_w(y|x) \sum_i \frac{f_i(x,y)}{f^{\#}(x,y)} \exp(f^{\#}(x,y) \delta_i) \Big] = B(\delta|$$

经过上述放缩，$B(\delta|w)$是对数似然函数改变量的一个新的下界。对$B(\delta|w)$求导，使得导数等于0。

> 再考虑迭代尺度法，收敛的条件是$L(w+\delta)$和$L(w)$的差值是接近0的，即最大化没有提升空间了，当$L(w+\delta) - L(w) = 0$，则$\delta = 0$。
>
> 会有下面思考：放缩了两次，怎样保证最大化下界，最后收敛时，可以使得$L(w+\delta) - L(w)$最大化呢？可将$\delta = 0$带入下界公式中，观察得到的值是否为0。当$\delta = 0$时，满足$A(\delta|w) = B(\delta|w) = 0$。

求$B(\delta|w)$对δ_i的偏导数，并令偏导数为0可得：

$$\sum_{x,y} \tilde{P}(x) P_w(y|x) f_i(x,y) \exp(\delta_i f^{\#}(x,y)) = E_{\tilde{P}}(f_i)$$

求解使得该等式成立的δ_i，没有一个显示的形式，对于这样一个方程，要如何寻找零点？此时可以用牛顿迭代法。

上述问题变为：已知$g(\delta_i) = \sum_{x,y} \tilde{P}(x) P_w(y|x) f_i(x,y) \exp(\delta_i f^{\#}(x,y)) - E_{\tilde{P}}(f_i)$，令$g(\delta_i) = 0$，求解$\delta_i$。

求解步骤：先给出δ_i的初值，更新$\delta_i^{(k+1)} = \delta_i^{(k)} - \frac{g(\delta_i^{(k)})}{g'(\delta_i^{(k)})}$

15 第6章-Logistic回归与最大熵模型-拉格朗日对偶性

拉格朗日对偶性是用在解优化问题中的一个性质，在第6章最大熵模型的推导过程和第7章SVM都用到了这个性质，主要是按照附录C中的内容来介绍。

15.1 原始问题

优化问题的一般形式，对于任意一个优化问题，都有一个需要优化的目标函数为f，需要优化的变量为x，最小化$f(x)$，会有一些约束条件，有不等式约束$c_i(x) \leqslant 0, i = 1, 2, \cdots, k$，也有等式约束$h_j(x) = 0, j = 1, 2, \cdots, l$，很多问题并不是两个约束都有，比如第6章中，只有等式约束（即$k = 0$），第7章SVM中，只有不等式约束（即$l = 0$）。

原始问题P：

$$\begin{aligned} \min_{x \in R^n} \quad & f(x) \\ \text{s.t.} \quad & c_i(x) \leqslant 0, i = 1, 2, \cdots, k \\ & h_j(x) = 0, j = 1, 2, \cdots, l \end{aligned}$$

15.2 拉格朗日函数

在不等式约束中，有多少个不等式就有多少个α，有多少个等式就有多少个β。

$$L(x, \alpha, \beta) = f(x) + \sum_{i=1}^{k} \alpha_i c_i(x) + \sum_{j=1}^{l} \beta_j h_j(x)$$

其中$\alpha_i \geqslant 0$目标函数f，优化变量x，可行域为满足所有约束条件的x，最优值为x^*，对应的目标函数为$p^* = f(x^*)$。

结论1： 可以用拉格朗日函数的极小极大问题表示为原始问题，记为

$$P = \min_x \max_{\alpha, \beta} L(x, \alpha, \beta) = \min_x \begin{cases} f(x) & c_i(x) \leqslant 0, h_j(x) = 0 \\ \infty & \text{其他} \end{cases}$$

结论2： 对偶问题D可以用拉格朗日函数的极大极小问题表示，记为

$$\begin{aligned} \max_{\alpha, \beta} \min_x \quad & L(x, \alpha, \beta) \\ \text{s.t.} \quad & \alpha_i \geqslant 0, i = 1, 2, \cdots, k \end{aligned}$$

其中，最优解为α^*, β^*，对应的值记为d^*。

15.3 总结

已知原始的最优化问题，根据拉格朗日函数定义出了原始问题的对偶问题，原始问题是关于x求目标函数的极小值，对偶问题是关于α，β求目标函数的极大值，原始问题相当于极小极大化拉格朗日函数，对偶问题相当于极大极小化拉格朗日函数。

15.4 定理C.1

本定理是讲了原始问题p^*和对偶问题d^*的关系。若原始问题和对偶问题都有最优解，则：

$$d^* = \max_{\alpha, \beta: \alpha \geqslant 0} \min_x L(x, \alpha, \beta) \leqslant \min_x \max_{\alpha, \beta: \alpha \geqslant 0} L(x, \alpha, \beta) = p^*$$

证明：

$$\because d^* = \max_{\alpha,\beta} \min_{x} L(x,\alpha,\beta)$$

$$\leqslant \max_{\alpha,\beta} \min_{x\in可行域} L(x,\alpha,\beta)$$

$$\leqslant \max_{\alpha,\beta} \min_{x\in可行域} f(x)$$

$$\leqslant \min_{x\in可行域} f(x) = p^*$$

$$\therefore d^* \leqslant p^*$$

直观理解： 对偶问题的最优解是一个原始问题极小值的下界。

15.5 定理C.2

要使$d^* = p^*$，需要满足两个条件：（1）原问题是凸优化问题；（2）原问题满足Slater条件，$d^* = p^*$称为强对偶性，$d^* \leqslant p^*$称为弱对偶性。定理C.2给出了对偶性的一个充分条件。

优化问题就是在一个可行域里，求解目标函数的最优值，当可行域是一个凸集，目标函数$f(x)$为凸函数时，该问题就是**凸优化问题**，也就是说，在一个凸集上求解凸函数的极小值问题。

凸集是从某空间中的任意取两个点，这两个点连成的线段上面的每一个点依然在这个点集中。

凸函数是在这个函数上，任取两个点，所连成的线段在这个函数上方。当$C_i(x)$是一个凸函数，$h_j(x)$函数是关于x的线性函数（仿射函数）时，可行域是一个**凸集**。

Slater条件是针对约束中，不等式约束的一个限制，之前所讲的每一个不等式约束，所满足的x可以看成一个集合，在凸优化问题中，每一个满足这个条件的x都是一个凸集，如果x同时满足所有不等式约束，那么所有不等式约束对应凸集的交集依然是一个凸集，Slater条件要求这些凸集的交集是有内点的，不仅仅只是边界，边界中还有一些点。

Slater条件相对比较宽松，一般只需要确定原始问题是凸优化问题即可，那么强对偶性就是成立的，可以通过求解对偶问题，求解原问题的最优值。以上条件可以得到$d^* = p^* = L(x^*,\alpha^*,\beta^*)$。定理C.2告诉我们，在什么样的条件下可以通过对偶问题求解原始问题。

15.6 定理C.3(KKT条件)

当原始问题满足强对偶性时，才可以使用KKT条件。

KKT条件如下：

$$\nabla_x L(x^*,\alpha^*,\beta^*) = 0$$

$$\alpha_i^* c_i(x^*) = 0$$

$$c_i(x^*) \leqslant 0$$

$$\alpha_i^* \geqslant 0$$

$$h_j(x^*) = 0$$

其中，$\nabla_x L(x^*,\alpha^*,\beta^*) = 0$为拉格朗日函数直观地对$x$求导等于0，$c_i(x^*) \leqslant 0$和$h_j(x^*) = 0$u是原始问题的约束，$\alpha_i^* \geqslant 0$是对偶问题的约束，$\alpha_i^* c_i(x^*) = 0$称为互补松弛条件，通过KKT条件，求出原始问题和对偶问题的解。

16 第7章-支持向量机（SVM）-导读

首先回顾一下感知机模型，感知机模型是当数据线性可分时，如何用一个超平面区分两类不同的数据。对于上述情况，支持向量机和感知机是非常相似的，它们的差别只在于决策函数（损失函数）的不同。

7.1节介绍的就是线性可分的情况，会和感知机比较，当线性可分时，这就是一个最简单的情况，在7.2节会介绍如何处理数据线性不可分的情况。当一个线性模型，对数据的分类效果不好时，但数据可以用一个曲面进行分隔，将数据的输入变量做变换（即输入空间和特征空间不是完全一致的），在7.3节中，可以用核函数的方法。

16.1 线性可分支持向量机与硬间隔最大化

假设空间：$w \cdot x + b = 0$
决策函数：$f(x) = \text{sign}(w^* \cdot x + b^*)$，输出的类别是$\{+1, -1\}$。

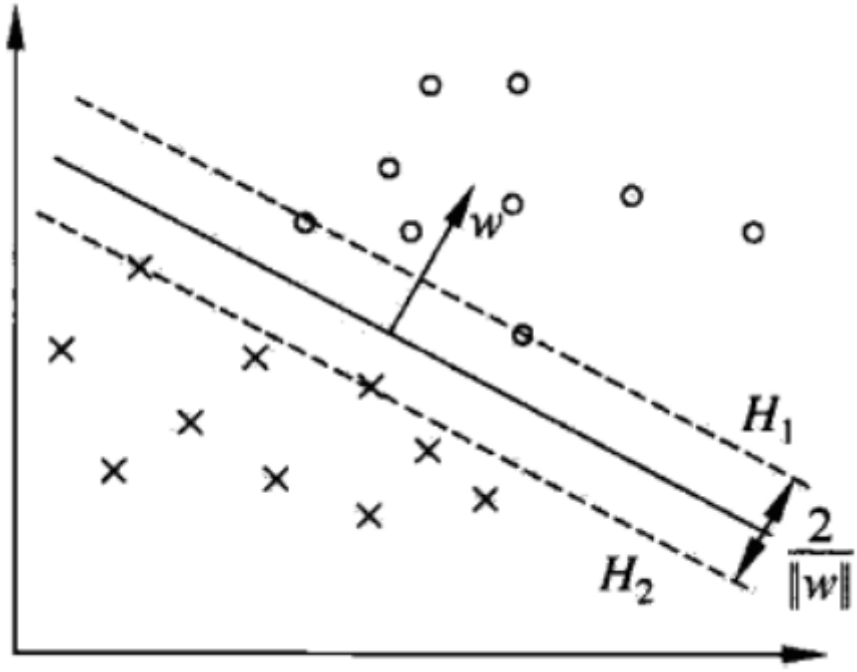

图7-1 支持向量

如图7-1所示，圆点和×点为两种不同的类别，在感知机模型中，只要找到一个超平面把这两组数据分开就可以，得到的超平面不是唯一的，在SVM模型中，得到的超平面是唯一的，模型选择的依据就是**硬间隔最大化**，这个是什么意思呢？当选择一个超平面将两组数据分开，在二维问题中，这个超平面是一条直线（如图7-1中的实线），每一个点到该直线都有一个距离，使得每个点到直线的距离都比较大，因此这条直线就是唯一的，所有这些距离中最小的距离的点组成的平面称为**支撑超平面**（如图7-1的虚线），这些点被称为**支持向量**，图中实线为**分离超平面**，两个支撑超平面之间的距离称为**硬间隔**。“硬”表示所有的点不能在支撑超平面中间，**硬间隔最大化**就是支持向量机模型选择的标准。

如何使得硬间隔最大化？书中讲述了两个概念：函数间隔和几何间隔。假设有一个超平面$w \cdot x + b = 0$，某一个实例为x^*，函数间隔为$|w \cdot x^* + b|$，几何间隔为$\dfrac{|w \cdot x^* + b|}{\|w\|}$。

书中描述要使得硬间隔最大化，使用几何间隔，而不用函数间隔。一个点到不同超平面的距离，不能用函数间隔表示。

∵ $|w \cdot x + b|$可以转换为$y_i(w \cdot x_i + b)$
∴ 几何间隔可以转化为$\dfrac{y_i(w \cdot x_i + b)}{\|w\|}$
∵ 硬间隔最大化表示点到超平面距离最小的点的距离最大
∴ **硬间隔最大化公式**为

$$\max_{w,b} \min_{i} \frac{y_i(w \cdot x_i + b)}{\|w\|}$$

将$\dfrac{1}{\|w\|}$提取出来，变为

$$\max_{w,b} \frac{1}{\|w\|} \min_{i} y_i(w \cdot x_i + b)$$

令$y_i(w \cdot x_i + b) \geqslant 1$，可以转化为

$$\begin{aligned}\max_{w,b} \quad & \frac{1}{\|w\|} \\ \text{s.t.} \quad & y_i(w \cdot x_i + b) \geqslant 1\end{aligned}$$

∵ 最大化$\frac{1}{\|w\|}$和最小化$\frac{1}{2}\|w\|^2$是等价的

最后线性可分的支持向量机的最优化问题为

$$\begin{aligned}\min_{w,b} \quad & \frac{1}{2}\|w\|^2 \\ \text{s.t.} \quad & y_i(w \cdot x_i + b) - 1 \geqslant 0\end{aligned}$$

对偶问题：

$$\begin{aligned}\min_{\alpha} \quad & \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^{N}\alpha_i \\ \text{s.t.} \quad & \sum_{i=1}^{N}\alpha_i y_i = 0 \\ & \alpha_i \geqslant 0, \quad i = 1, 2, \cdots, N\end{aligned}$$

最优解：

$$w^* = \sum_{i=1}^{N}\alpha_i^* y_i x_i$$

$$b^* = y_j - \sum_{i=1}^{N}\alpha_i^* y_i (x_i \cdot x_j)$$

16.2 线性支持向量机与软间隔最大化

16.2.1 支持向量

最优化问题：

$$\begin{aligned}\min_{w,b,\xi} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i \\ \text{s.t.} \quad & y_i\,(w \cdot x_i + b) \geqslant 1 - \xi_i, \quad i = 1, 2, \cdots, N \\ & \xi_i \geqslant 0, \quad i = 1, 2, \cdots, N\end{aligned}$$

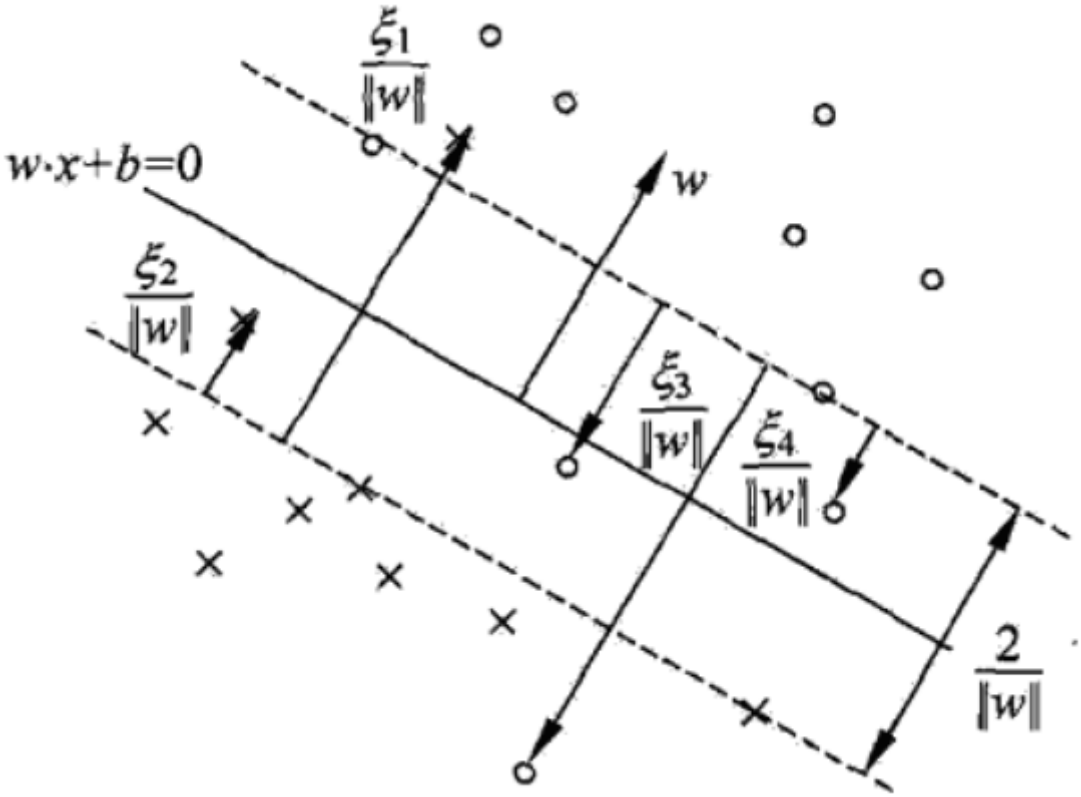

图7-2 软间隔的支持向量

软间隔最大化就是允许数据点出现在两个支撑超平面之间（如图7-2），加入了惩罚项，对误分类点进行惩罚，公式中$\frac{1}{2}\|w\|^2$表示两个支撑超平面之间的几何间隔尽可能大，会纳入更多的数据点，其中也会有误分类的数据点，数据点越多，惩罚$C\sum_{i=1}^{N}\xi_i$就越大，C为了衡量支撑超平面之间的间隔。当C取∞时，该问题与7.1节的问题是一样的。

16.2.2 合页损失函数

最优化问题：

$$\sum_{i=1}^{N}[1-y_i(w\cdot x_i+b)]_+ + \lambda\|w\|^2$$

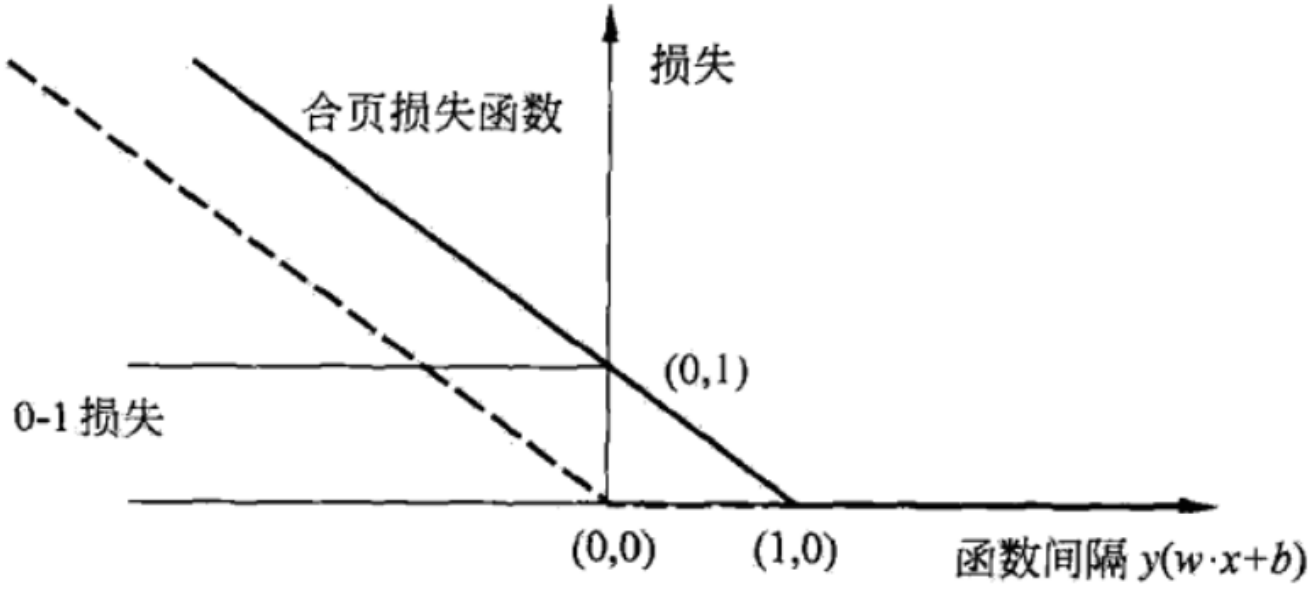

图7-3 合页损失函数

图7-3中，横轴表示函数间隔，函数间隔$> 1-\xi_i$，ξ_i表示为惩罚，$\xi_i \geqslant 0$，当函数间隔$\geqslant 1$时，$\xi_i = 0$（即没有惩罚），当函数间隔< 1，$\xi = 1 - g(w \cdot x + b)$，在图中表示为实线，其函数为$[1 - y_i(w \cdot x_i + b)]_+$

对偶问题：

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^{N}\alpha_i \\ \text{s.t.} \quad & \sum_{i=1}^{N}\alpha_i y_i = 0 \\ & 0 \leqslant \alpha_i \leqslant C, \quad i = 1, 2, \cdots, N \end{aligned}$$

参数的最优解：

$$w^* = \sum_{i=1}^{N}\alpha_i^* y_i x_i$$

$$b^* = y_j - \sum_{i=1}^{N} y_i \alpha_i^* (x_i \cdot x_j)$$

16.3 非线性支持向量机与核函数

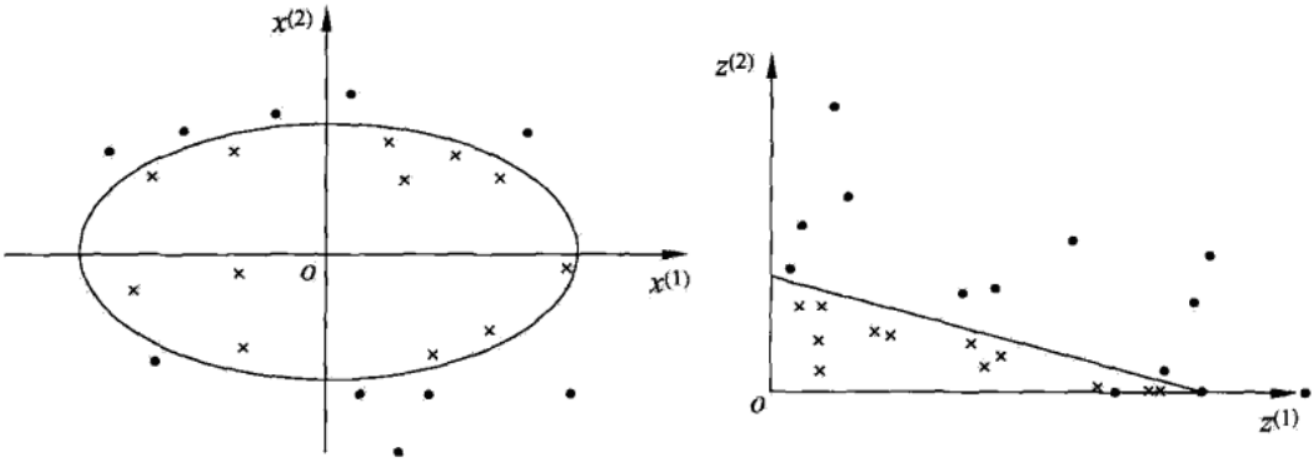

图7-4 非线性分类问题与核技巧示例

7-4左图中，$x^{(1)}$和$x^{(2)}$是二维输入变量，分为•点和×点两类，没有办法用一条直线很好的分隔数据，从图中可以看到，×点离原点近，•点离原点远，将坐标进行变换$z=\phi(x),\phi(x)=x^2$，故
$z^{(1)}=(x^{(1)})^2, z^{(2)}=(x^{(2)})^2$，最后可以得到7-4右图。可见右图是线性可分的，可以用超平面进行分隔，用新的z进行支持向量机，因为在支持向量机的对偶形式中出现的是内积形式，现在得到的内积为
$k(x_i,x_j)=\phi(x_i)\cdot\phi(x_j)$。
对偶问题：

$$\begin{aligned}\min_{\alpha} \quad & \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K(x_i,x_j)-\sum_{i=1}^{N}\alpha_i \\ \text{s.t.} \quad & \sum_{i=1}^{N}\alpha_i y_i=0 \\ & 0\leqslant\alpha_i\leqslant C, \quad i=1,2,\cdots,N\end{aligned}$$

决策函数：

$$f(x)=\text{sign}\left(\sum_{i=1}^{N}\alpha_i^* y_i K(x\cdot x_i)+b^*\right)$$

原始形式的最优化问题：

$$\begin{aligned}\min \quad & \frac{1}{2}\|w\|^2+C\sum_i\xi \\ \text{s.t.} \quad & y_i[w\cdot\phi(x)]\geqslant 1-\xi_i \\ & \xi_i\geqslant 0\end{aligned}$$

分离超平面为$w\cdot\phi(x)+b=0$
缺点： 不知道用什么样的曲面能更好地分隔数据，选择核函数，并没有很强的依据，只能依靠经验去试验，也就是说用核函数技巧，可以解决曲面分类的一些问题，但同时用什么样的曲面去解决，这也是一个新的问题。

16.4 序列最小最优化算法

对偶问题：

$$\begin{aligned}\min_{\alpha} \quad & \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K(x_i,x_j)-\sum_{i=1}^{N}\alpha_i \\ \text{s.t.} \quad & \sum_{i=1}^{N}\alpha_i y_i=0 \\ & 0\leqslant\alpha_i\leqslant C, \quad i=1,2,\cdots,N\end{aligned}$$

需要优化的变量是α_i有N个，当数据量很大的时候，需要优化的变量非常多，很难计算，所以不优化所有的变量，每一次优化其中的一部分变量。在该算法中，每次优化两个变量。

17 第7章-支持向量机-最大间隔分离超平面存在唯一性

最大分离超平面是由一个最优化问题得到的，最优化问题为：

$$\begin{aligned}\min_{w,b} \quad & \frac{1}{2}\|w\|^2 \\ \text{s.t.} \quad & y_i(w\cdot x_i+b)-1\geqslant 0\end{aligned}$$

存在并且最优解是唯一的。

记最优解为w^*,b^*，证明过程分为两个部分，第一部分是存在性，这一部分比较简单就略过了。第二部分是唯一性，用的是反证法。首先假设最优解不是唯一的，两个解分别为$(w_1^*,b_1^*),(w_2^*,b_2^*)$，推导
$w_1^*=w_2^*,b_1^*=b_2^*$，这样就可以证明最优解是唯一的。

17.1 证明

已知w_1^*,w_2^*都是最优化问题的解
$\therefore \|w_1^*\|=\|w_2^*\|=c$，$c$是一个常数
构造另一组解：$w=\dfrac{w_1^*+w_2^*}{2},b=\dfrac{b_1^*+b_2^*}{2}$
现假设w_1^*,b_1^*,w_2^*,b_2^*都已经求出，可得w,b都是确定的。

17.1.1 首先需要验证w,b是否满足约束条件

约束条件：$y_i(\dfrac{w_1^*+w_2^*}{2}\cdot x_i+\dfrac{b_1^*+b_2^*}{2})-1\geqslant 0$

$$\begin{aligned}\therefore \quad & y_i(\frac{w_1^*+w_2^*}{2}\cdot x_i+\frac{b_1^*+b_2^*}{2})-1 \\ =&\frac{1}{2}\left[\left(y_i(w_1^*\cdot x_i+b_1^*)-1\right)+\left(y_i(w_2^*\cdot x_i+b_2^*)-1\right)\right]\end{aligned}$$

$\because w_1^*,b_1^*,w_2^*,b_2^*$满足约束条件
$y_i(w_1^*\cdot x_i+b_1^*)-1\geqslant 0, y_i(w_2^*\cdot x_i+b_2^*)-1\geqslant 0$
$\therefore y_i(\dfrac{w_1^*+w_2^*}{2}\cdot x_i+\dfrac{b_1^*+b_2^*}{2})-1\geqslant 0$
可得w,b在可行域中，满足所有的N个约束条件

17.1.2 证明$w_1^*=w_2^*$

易得$\|w_1^*\| \leqslant \|w\|$，不然$w_1^*$就不是最优解了。

$\therefore c \leqslant \|w\| = \|\frac{1}{2}w_1^* + \frac{1}{2}w_2^*\|$

根据**三角不等式** $\|\frac{1}{2}w_1^* + \frac{1}{2}w_2^*\| \leqslant \frac{1}{2}\|w_1^*\| + \frac{1}{2}\|w_2^*\|$

$\therefore c \leqslant \|w\| = \|\frac{1}{2}w_1^* + \frac{1}{2}w_2^*\| \leqslant \frac{1}{2}\|w_1^*\| + \frac{1}{2}\|w_2^*\| = c$

根据**夹逼定理** $\|\frac{1}{2}w_1^* + \frac{1}{2}w_2^*\| = \frac{1}{2}\|w_1^*\| + \frac{1}{2}\|w_2^*\|$

根据**向量相加原理** ，只有w_1^*与w_2^*是同方向的，才能满足上式。

$\therefore w_1^* = \lambda w_2^*, \lambda \geqslant 0$

$\because \|w_1^*\| = \|w_2^*\| = c$

$\therefore \lambda = 1$

$\therefore w_1^* = w_2^*$

17.1.3 证明$b_1^* = b_2^*$

可以把两个最优解写成$(w^*, b_1^*), (w^*, b_2^*)$

先观察在最优化问题中b出现的位置，可以看到b出现在不等式约束中，假如想找到一个关于b的等式，那就需要寻找使得不等式约束变成等式约束的那些约束。

那什么时候是等式约束呢？位于支撑超平面上的点就可以满足等式约束，目前有两个最优解，就有两个不同的分离超平面，由于w是相同的，就表示斜率是一样的，但是b_1^*和b_2^*不同，这两个平面有不同的位置（平移关系），这两个分离超平面都会各有两个支撑超平面。

在(w, b_1^*)中，正类+1的点记为x_1'，负类-1的点记为x_1''，在(w, b_2^*)中，正类+1的点记为x_2'，负类-1的点记为x_2''，可知x_2'位于一个支撑超平面（正类+1）上方，x_2''位于一个支撑超平面（负类-1）下方，同理x_1'位于一个支撑超平面（正类+1）上方，x_1''位于一个支撑超平面（负类-1）下方。

由于x_1'与x_2'在同一个支撑超平面（正类+1）上

$\therefore w^* \cdot x_1' + b_1^* - 1 = 0, w^* \cdot x_2' + b_1^* - 1 \geqslant 0$

$\therefore w^* \cdot x_1' \leqslant w^* \cdot x_2'$

$\therefore w^* \cdot x_2' + b_2^* - 1 = 0, w^* \cdot x_1' + b_2^* - 1 \geqslant 0$

$\therefore w^* \cdot x_2' \leqslant w^* \cdot x_1'$

根据$w^* \cdot x_1' \leqslant w^* \cdot x_2'$和$w^* \cdot x_2' \leqslant w^* \cdot x_1'$，可得$w^* \cdot (x_1' - x_2') = 0$，同理可得$w^* \cdot (x_1'' - x_2'') = 0$

$$\because \begin{matrix} w^* \cdot x_1' + b_1^* - 1 = 0 \\ -(w^* \cdot x_1'' + b_1^*) - 1 = 0 \end{matrix}$$

$$\therefore \begin{matrix} b_1^* = 1 - w* \cdot x_1' \\ b_1^* = -1 - w* \cdot x_1'' \end{matrix}$$

$\therefore b_1^* = \frac{1}{2}(w^* \cdot x_1' + w^* \cdot x_1'')$

同理可得$b_2^* = \frac{1}{2}(w^* \cdot x_2' + w^* \cdot x_2'')$ 由于需要证明$b_1^* = b_2^*$，故求$b_1^* - b_2^* = 0$

$\therefore b_1^* - b_2^* = -\frac{1}{2}[w^* \cdot (x_1' - x_2') + w^* \cdot (x_1'' - x_2'')]$

$\because w^* \cdot (x_1' - x_2') = 0, w^* \cdot (x_1'' - x_2'') = 0$

$\therefore -\frac{1}{2}[w^* \cdot (x_1' - x_2') + w^* \cdot (x_1'' - x_2'')] = 0$

$\therefore b_1^* - b_2^* = 0$ 即 $b_1^* = b_2^*$

17.1.4 结论

根据$w_1^* = w_2^*$，$b_1^* = b_2^*$，可得两个最优解$(w_1^*, b_1^*), (w_2^*, b_2^*)$是相同的，解的唯一性得证。

18 第7章-支持向量机-软间隔最大化对偶问题

拉格朗日对偶性

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i \\ \text{s.t.} \quad & y_i(w\cdot x_i + b) \geqslant 1-\xi, i=1,\cdots,N \\ & \xi_i \geqslant 0, i=1,\cdots,N \end{aligned}$$

18.1 对书中第109页的一句话的理解

在求解这个问题之前，首先解释书中第109页的一句话

> 该问题的最优解w是唯一的，但b的解可能不唯一，而是存在一个区间。

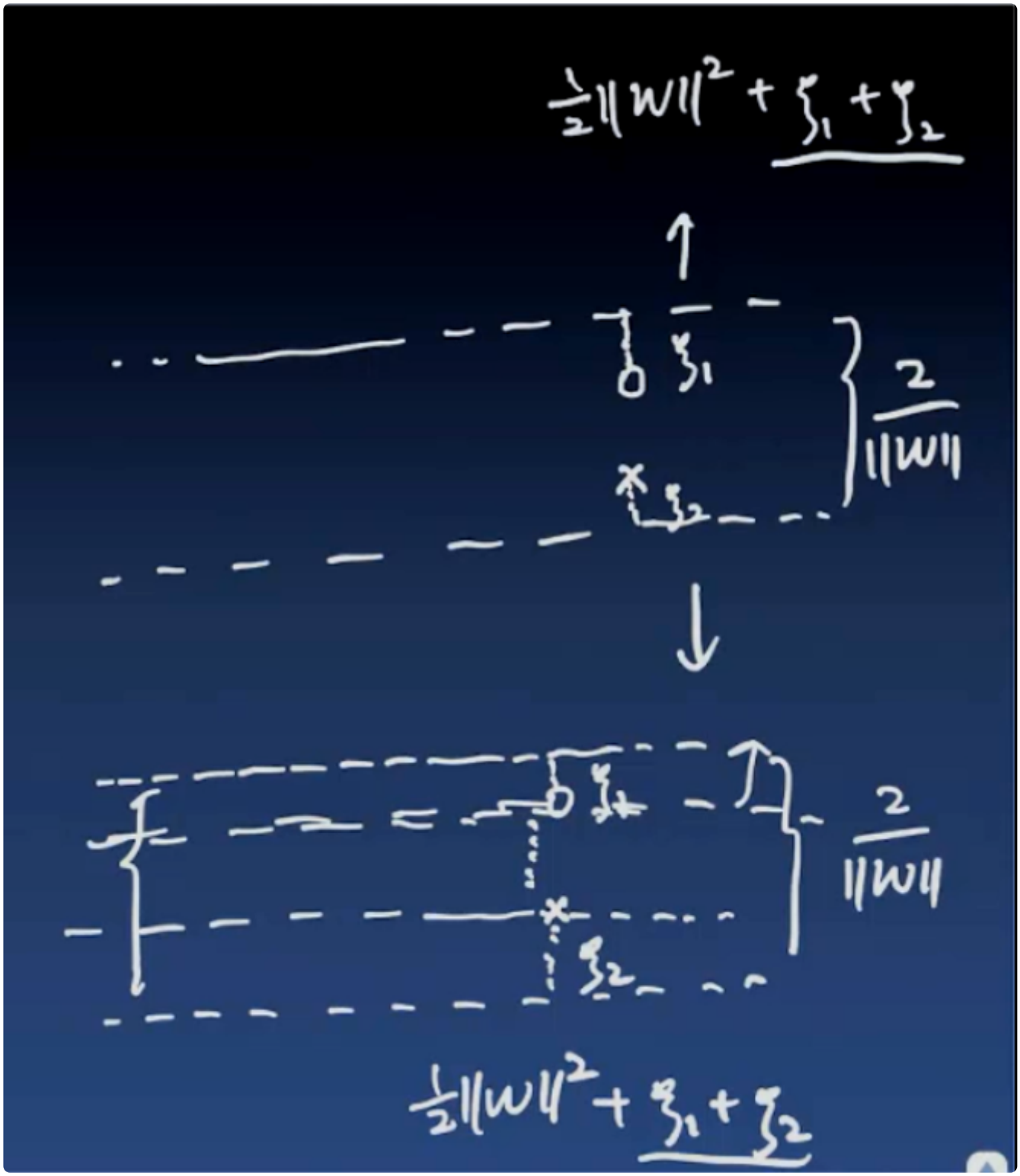

图7-5 w的唯一性

w是唯一的，说明得到的分离超平面的斜率是唯一的。从图7-5中可以看到，将支撑超平面平移之后，$\xi_1+\xi_2$的和是不变的，平移的范围就是从原始位置，移动到最近的误分类点的距离，记为b，所以b的解是不唯一的。从本例中推测一下，ξ的解也是不唯一的，但是$\sum_{i=1}^{N}\xi_i$是不变的，这就是对书中该句话的理解。

在硬间隔最大化中，解是唯一的，但是在软间隔中，不能说解是唯一的。

18.2 软间隔最大化对偶问题的可行性

原始问题是一个凸优化问题（目标函数是一个凸函数，约束条件是一个凸集）。

$\because \frac{1}{2}\|w\| = \frac{1}{2}w^T \cdot w$

可知$\frac{1}{2}w^T \cdot w$是一个凸函数。

> $f(x) = x$即是一个凸函数，也是一个凹函数。因为任取两个点，该点连线不在函数的下方，即为凸函数，同理可得也是凹函数。

$\therefore f(\xi) = \xi$是一个凸函数，有N个这样的凸函数。

$\therefore \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i$是凸函数，即目标函数是凸函数。

$\because y_i(w \cdot x_i + b) \geqslant 1 - \xi \Rightarrow 1 - \xi_i - y_i(w \cdot x_i + b) \leqslant 0$

上式是一个超平面，满足超平面一侧的点的条件，即满足线性函数$\leqslant 0$的点组成的集合就是凸集，同理约束条件$-\xi_i \leqslant 0$，这个也是关于ξ_i的线性函数，同样满足该函数的点组成的集合也是凸集。那么$2N$个凸集的交集依然是凸集，所以约束条件对应的可行域依然是一个凸集。那么**在一个凸集上求解一个凸函数的最小值问题，这就是一个凸优化问题。**

在拉格朗日对偶性（第6章-Logistic回归与最大熵模型-拉格朗日对偶性）中说过，对于凸优化问题来说，对偶问题的解与原始问题的解（目标函数最优值）是相等的，并且满足KKT条件。

18.3 对偶问题的转换与推导

拉格朗日函数：

$$\begin{aligned} & L(w, b, \xi, \alpha, \mu) \\ = \quad & \frac{1}{2}\|w\|^2 + C\sum \xi_i + \sum \alpha_i[1 - \xi_i - y_i(w \cdot x_i + b)] + \sum \mu_i(-\xi_i) \\ = \quad & \frac{1}{2}\|w\|^2 + C\sum \xi_i - \sum \alpha_i[y_i(w \cdot x_i + b) - 1 + \xi_i] - \sum \mu_i\xi_i \end{aligned}$$

其中$\alpha_i \geqslant 0, \mu_i \geqslant 0$

下面由拉格朗日函数写出对偶问题，首先分别求出w, b, ξ的梯度，并令梯度等于0：

$$\begin{aligned} \nabla_w L(w, b, \xi, \alpha, \mu) &= w - \alpha_i y_i x_i = 0 \Rightarrow w = \alpha_i y_i x_i \\ \nabla_b L(w, b, \xi, \alpha, \mu) &= - \alpha_i y_i = 0 \\ \nabla_{\xi_i} L(w, b, \xi, \alpha, \mu) &= C - \alpha_i - \mu_i = 0 \end{aligned}$$

将上述解代入原始问题中，可得

$$\begin{aligned} & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i \\ = \quad & \frac{1}{2}(\sum \alpha_i y_i x_i) \cdot (\sum \alpha_i y_i x_i) + C\sum \xi_i - (\sum \alpha_i y_i x_i) \cdot (\sum \alpha_i y_i x_i) - b\sum \alpha_i y_i + \sum \alpha_i - \sum \alpha_i \xi_i - \\ = \quad & -\frac{1}{2}(\sum \alpha_i y_i x_i) \cdot (\sum \alpha_i y_i x_i) + \sum \alpha_i \\ = \quad & -\frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) + \sum \alpha_i \end{aligned}$$

所以对偶问题为

$$
\begin{aligned}
\max_{\alpha} \quad & -\frac{1}{2}\sum_{i}\sum_{j}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j) + \sum \alpha_i \\
\text{s.t.} \quad & \alpha_i y_i = 0 \\
& C - \alpha_i - \mu_i = 0 \\
& \alpha_i \geqslant 0 \\
& \mu_i \geqslant 0, i = 1, 2, \cdots, N
\end{aligned}
$$

因为原始问题是凸优化问题，解满足KKT条件。
第一个KKT条件为梯度等于0：

$$
\begin{aligned}
\nabla_w L(w, b, \xi, \alpha, \mu) &= w - \quad \alpha_i y_i x_i = 0 \Rightarrow w = \quad \alpha_i y_i x_i \\
\nabla_b L(w, b, \xi, \alpha, \mu) &= - \quad \alpha_i y_i = 0 \\
\nabla_{\xi_i} L(w, b, \xi, \alpha, \mu) &= C - \alpha_i - \mu_i = 0
\end{aligned}
$$

第二个KKT条件为互补松弛条件：

$$
\begin{aligned}
\alpha_i \left[y_i (w \cdot x_i + b) - 1 + \xi_i \right] = 0 \\
\mu_i \xi_i = 0
\end{aligned}
$$

第三个KKT条件为原始问题的不等式约束：

$$
\begin{aligned}
y_i (w \cdot x_i + b) \geqslant 1 - \xi \\
\xi_i \geqslant 0
\end{aligned}
$$

第四个KKT条件为原始问题的等式约束，在这个例子中没有等式约束。
第五个KKT条件是关于拉格朗日乘子的约束条件：

$$
\begin{aligned}
\alpha_i \geqslant 0 \\
\mu_i \geqslant 0
\end{aligned}
$$

18.4 求解（定理7.3）

如果原始问题的最优解为w^*, b^*, ξ_i*，对偶问题的最优解为α_i^*, μ_i^*，这些最优解是满足上述KKT条件。

一旦根据对偶问题求出α_i^*后，可以根据$w = \quad \alpha_i y_i x_i$ 求出w^*，求解b^*，需要在KKT条件中找到关于b的等式约束$\alpha_i \left[y_i (w \cdot x_i + b) - 1 + \xi_i \right] = 0$，要求解出$b^*$，必须$\alpha_i \neq 0$，又由于$0 < \alpha_i < C$和$C - \alpha_i - \mu_i = 0 \Rightarrow \alpha_i = C - \mu_i < C$，可以得到$\mu_i > 0$，代入$\mu_i \xi_i = 0$，可以得到$\xi_i = 0$，再代入公式$\alpha_i \left[y_i (w \cdot x_i + b) - 1 + \xi_i \right] = 0$中，得到$y_i (w \cdot x_i + b) - 1 = 0$，可以得到$b^*$。

以上解释了为什么需要α_i^*满足$0 < \alpha_i < C$。## 第7章-支持向量机-软间隔最大化对偶问题

拉格朗日对偶性

$$
\begin{aligned}
\min_{w,b,\xi} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i \\
\text{s.t.} \quad & y_i (w \cdot x_i + b) \geqslant 1 - \xi, i = 1, \cdots, N \\
& \xi_i \geqslant 0, i = 1, \cdots, N
\end{aligned}
$$

18.5 对书中第109页的一句话的理解

在求解这个问题之前，首先解释书中第109页的一句话

> 该问题的最优解w是唯一的，但b的解可能不唯一，而是存在一个区间。

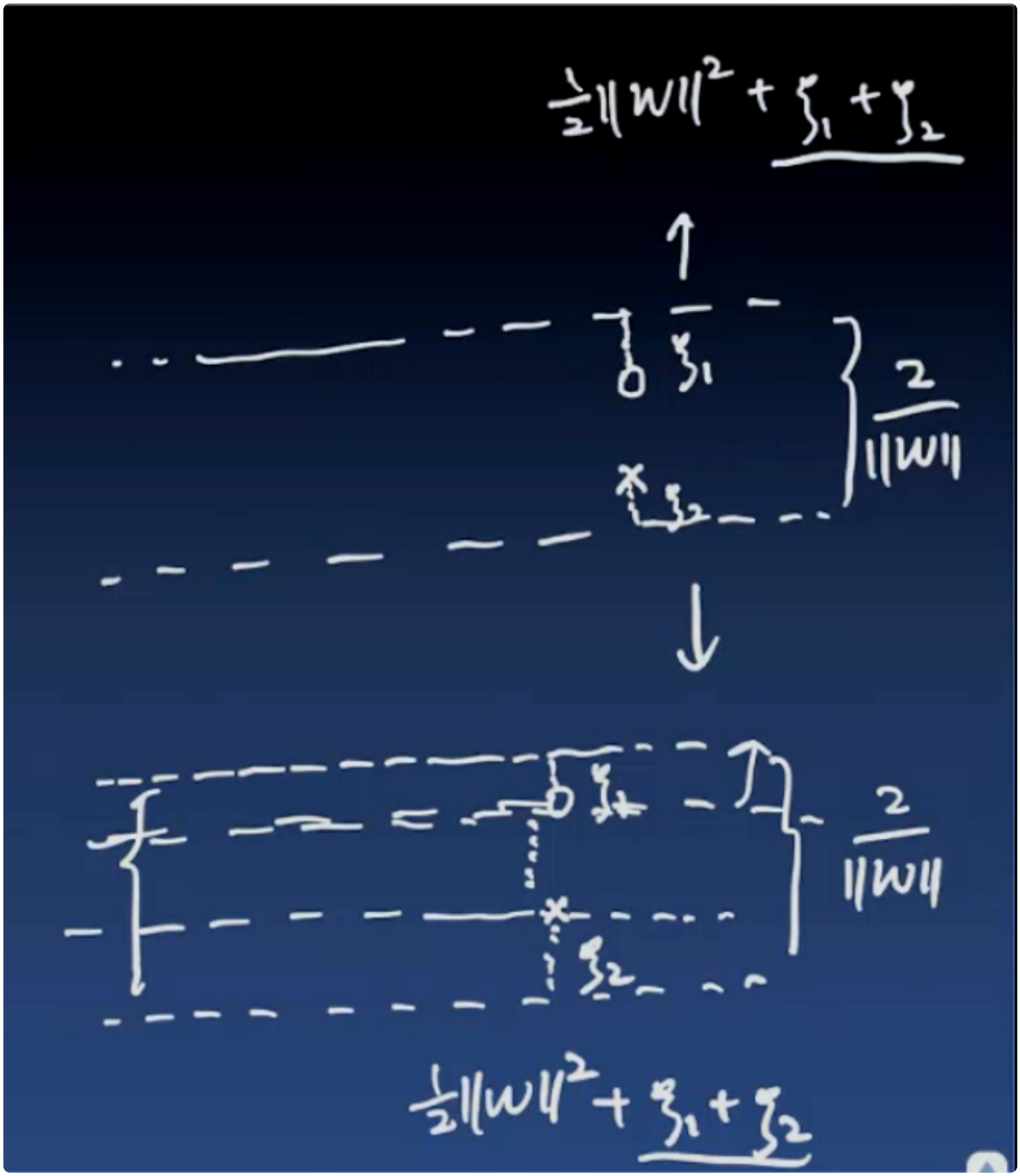

图7-5 w的唯一性

w是唯一的，说明得到的分离超平面的斜率是唯一的。从图7-5中可以看到，将支撑超平面平移之后，$\xi_1+\xi_2$的和是不变的，平移的范围就是从原始位置，移动到最近的误分类点的距离，记为b，所以b的解是不唯一的。从本例中推测一下，ξ的解也是不唯一的，但是$\sum_{i=1}^{N}\xi_i$是不变的，这就是对书中该句话的理解。

在硬间隔最大化中，解是唯一的，但是在软间隔中，不能说解是唯一的。

18.6 软间隔最大化对偶问题的可行性

原始问题是一个凸优化问题（目标 函数是一个凸函数，约束条件是一个凸集）。

$\because \frac{1}{2}\|w\| = \frac{1}{2}w^T \cdot w$

可知$\frac{1}{2}w^T \cdot w$是一个凸函数。

> $f(x) = x$即是一个凸函数，也是一个凹函数。因为任取两个点，该点连线不在函数的下方，即为凸函数，同理可得也是凹函数。

$\therefore f(\xi) = \xi$是一个凸函数，有N个这样的凸函数。

$\therefore \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i$是凸函数，即目标函数是凸函数。

$\because y_i(w \cdot x_i + b) \geqslant 1 - \xi \Rightarrow 1 - \xi_i - y_i(w \cdot x_i + b) \leqslant 0$

上式是一个超平面，满足超平面一侧的点的条件，即满足线性函数$\leqslant 0$的点组成的集合就是凸集，同理约束条件$-\xi_i \leqslant 0$，这个也是关于ξ_i的线性函数，同样满足该函数的点组成的集合也是凸集。那么$2N$个凸集的交集依然是凸集，所以约束条件对应的可行域依然是一个凸集。那么**在一个凸集上求解一个凸函数的最小值问题，这就是一个凸优化问题。**

在拉格朗日对偶性（第6章-Logistic回归与最大熵模型-拉格朗日对偶性）中说过，对于凸优化问题来说，对偶问题的解与原始问题的解（目标函数最优值）是相等的，并且满足KKT条件。

18.7 对偶问题的转换与推导

拉格朗日函数：

$$\begin{aligned} & L(w, b, \xi, \alpha, \mu) \\ = \quad & \frac{1}{2}\|w\|^2 + C\sum \xi_i + \sum \alpha_i[1 - \xi_i - y_i(w \cdot x_i + b)] + \sum \mu_i(-\xi_i) \\ = \quad & \frac{1}{2}\|w\|^2 + C\sum \xi_i - \sum \alpha_i[y_i(w \cdot x_i + b) - 1 + \xi_i] - \sum \mu_i\xi_i \end{aligned}$$

其中$\alpha_i \geqslant 0, \mu_i \geqslant 0$

下面由拉格朗日函数写出对偶问题，首先分别求出w, b, ξ的梯度，并令梯度等于0：

$$\begin{aligned} \nabla_w L(w, b, \xi, \alpha, \mu) &= w - \alpha_i y_i x_i = 0 \Rightarrow w = \alpha_i y_i x_i \\ \nabla_b L(w, b, \xi, \alpha, \mu) &= - \alpha_i y_i = 0 \\ \nabla_{\xi_i} L(w, b, \xi, \alpha, \mu) &= C - \alpha_i - \mu_i = 0 \end{aligned}$$

将上述解代入原始问题中，可得

$$\begin{aligned} & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i \\ = \quad & \frac{1}{2}(\sum \alpha_i y_i x_i) \cdot (\sum \alpha_i y_i x_i) + C\sum \xi_i - (\sum \alpha_i y_i x_i) \cdot (\sum \alpha_i y_i x_i) - b\sum \alpha_i y_i + \sum \alpha_i - \sum \alpha_i \xi_i - \\ = \quad & -\frac{1}{2}(\sum \alpha_i y_i x_i) \cdot (\sum \alpha_i y_i x_i) + \sum \alpha_i \\ = \quad & -\frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) + \sum \alpha_i \end{aligned}$$

所以对偶问题为

$$
\begin{aligned}
\max_{\alpha} \quad & -\frac{1}{2}\sum_{i}\sum_{j}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j) + \sum \alpha_i \\
\text{s.t.} \quad & \alpha_i y_i = 0 \\
& C - \alpha_i - \mu_i = 0 \\
& \alpha_i \geqslant 0 \\
& \mu_i \geqslant 0, i = 1, 2, \cdots, N
\end{aligned}
$$

因为原始问题是凸优化问题，解满足KKT条件。
第一个KKT条件为梯度等于0：

$$
\begin{aligned}
\nabla_w L(w, b, \xi, \alpha, \mu) &= w - \quad \alpha_i y_i x_i = 0 \Rightarrow w = \quad \alpha_i y_i x_i \\
\nabla_b L(w, b, \xi, \alpha, \mu) &= - \quad \alpha_i y_i = 0 \\
\nabla_{\xi_i} L(w, b, \xi, \alpha, \mu) &= C - \alpha_i - \mu_i = 0
\end{aligned}
$$

第二个KKT条件为互补松弛条件：

$$
\begin{aligned}
\alpha_i \left[y_i (w \cdot x_i + b) - 1 + \xi_i \right] = 0 \\
\mu_i \xi_i = 0
\end{aligned}
$$

第三个KKT条件为原始问题的不等式约束：

$$
\begin{aligned}
y_i (w \cdot x_i + b) \geqslant 1 - \xi \\
\xi_i \geqslant 0
\end{aligned}
$$

第四个KKT条件为原始问题的等式约束，在这个例子中没有等式约束。
第五个KKT条件是关于拉格朗日乘子的约束条件：

$$
\begin{aligned}
\alpha_i \geqslant 0 \\
\mu_i \geqslant 0
\end{aligned}
$$

18.8 求解（定理7.3）

如果原始问题的最优解为w^*, b^*, ξ_i*，对偶问题的最优解为α_i^*, μ_i^*，这些最优解是满足上述KKT条件。

一旦根据对偶问题求出α_i^*后，可以根据$w = \quad \alpha_i y_i x_i$ 求出w^*，求解b^*，需要在KKT条件中找到关于b的等式约束$\alpha_i \left[y_i (w \cdot x_i + b) - 1 + \xi_i \right] = 0$，要求解出$b^*$，必须$\alpha_i \neq 0$，又由于$0 < \alpha_i < C$和$C - \alpha_i - \mu_i = 0 \Rightarrow \alpha_i = C - \mu_i < C$，可以得到$\mu_i > 0$，代入$\mu_i \xi_i = 0$，可以得到$\xi_i = 0$，再代入公式$\alpha_i \left[y_i (w \cdot x_i + b) - 1 + \xi_i \right] = 0$中，得到$y_i (w \cdot x_i + b) - 1 = 0$，可以得到$b^*$。

以上解释了为什么需要α_i^*满足$0 < \alpha_i < C$。

19 第8章-提升方法-导读

19.1 提升方法AdaBoost算法

19.1.1 提升方法的基本思路

这一部分提出了几个概念：弱可学习、强可学习、PAC学习框架

- **PAC学习框架：**在第1章讲的泛化误差上界的定理，也就是说，可以用一个训练误差和一个小的ε以一定的概率控制泛化误差。
- **弱可学习：**可以找到一个方法，用这个方法预测输出变量，对于分类问题，会比随机猜测的效果略好。
- **强可学习：**可以学习的效果很好。
- **提升方法的基本思路：**如果能找到一个弱可学习算法，那就可以将该算法提升为强可学习算法。

提升方法属于集成学习中的一种，集成学习就是用一些比较简单的模型，将它们综合起来构成一个复杂的模型。

集成学习两个主要类别：序列方法、并行方法。

- **序列方法：**当我们学到一个模型，再学下一个模型时，下一个模型依赖上一个模型的结果。
- **并行方法：**可以同时学很多模型，这些模型之间不会相互影响。

第8章中介绍的方法都属于序列方法，该章中首先介绍了AdaBoost算法，这个算法是一个非常重要的序列方法，提出得比较早，具有很强的理论支撑；之后就是提升树，AdaBoost主要解决二分类问题，提升树分为回归树提升方法和分类树提升方法，既可以用分类树提升方法解决多分类问题，也可以用回归树提升方法解决回归问题（当输出变量是连续变量时，该问题称为回归问题）。

19.1.2 AdaBoost算法（算法8.1）

AdaBoost算法用于解决分类问题，输出变量$y \in \{-1, +1\}$，当用原始数据学习完一个模型后，那这个学习的结果如何对第二个学习的模型产生影响呢？这个时候，用第一个模型的学习效果在训练数据集上的一个表现，该表现分为整体表现（如果有多个模型，第一个模型在训练数据集上的整体表现非常好，在最终的分类器加权中，整体效果越好的模型，其权重就越大）和在单个样本上的表现效果（如果在其中一个样本上表现效果好，在训练下一个模型时，就可以较少地考察表现好的样本点，更多考察表现效果差的样本点，这个就决定了在下一个模型中，每一个样本点的权重），以上就是AdaBoost的基本思路。

输入：训练数据集$T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$，其中$x_i \in \mathcal{X} \subseteq \mathbf{R}^n$, $y_i \in \mathcal{Y} = \{-1, +1\}$；弱学习算法；

输出：最终分类器$G(x)$。

(1)初始化训练数据的权值分布

$$D_1 = (w_{11}, \cdots, w_{1i}, \cdots, w_{1N}), \quad w_{1i} = \frac{1}{N}, \quad i = 1, 2, \cdots, N$$

(2)对$m = 1, 2, \cdots, M$表示学习每个小分类器的过程。

(a)使用具有权值分布D_m的训练数据集学习，得到基本分类器

$$G_m(x) : \mathcal{X} \to \{-1, +1\}$$

(b)计算$G_m(x)$在训练数据集上的分类误差率（衡量模型效果）

$$e_m = \sum_{i=1}^{N} P(G_m(x_i) \neq y_i) = \sum_{i=1}^{N} w_{mi} I(G_m(x_i) \neq y_i)$$

(c)计算$G_m(x)$的系数

$$\alpha_m = \frac{1}{2} \ln \frac{1 - e_m}{e_m}$$

(d)更新训练数据集的权值分布

$$D_{m+1} = (w_{m+1,1}, \cdots, w_{m+1,i}, \cdots, w_{m+1,N})$$

$$w_{m+1,i} = \frac{w_{mi}}{Z_m} \exp(-\alpha_m y_i G_m(x_i)), \quad i = 1, 2, \cdots, N$$

其中，Z_m是规范化因子

$$Z_m = \sum_{i=1}^{N} w_{mi} \exp(-\alpha_m y_i G_m(x_i))$$

它使D_{m+1}成为一个概率分布。

(3)构建基本分类器的线性组合

$$f(x) = \sum_{m=1}^{M} \alpha_m G_m(x)$$

得到最终分类器

$$G(x) = \text{sign}(f(x)) = \text{sign}\Big(\sum_{m=1}^{M} \alpha_m G_m(x)\Big)$$

AdaBoost算法所用的每一个基本分类器都是非常简单的模型，算法中并没有规定要使用什么样子的分类器，这个可以自己选择，只要分类器有效果就可以。

19.2 AdaBoost算法的训练误差分析

定理8.1（AdaBoost的训练误差界）AdaBoost算法最终分类器的训练误差界为

$$\frac{1}{N}\sum_{i=1}^{N} I(G(x_i) \neq y_i) \leqslant \frac{1}{N}\sum_{i} \exp(-y_i f(x_i)) = \prod_{m} Z_m$$

定理8.2（二类分类问题AdaBoost的训练误差界）

$$\prod_{m=1}^{M} Z_m = \prod_{m=1}^{M} [2\sqrt{e_m(1-e_m)}] = \prod_{m=1}^{M} \sqrt{(1-4\gamma_m^2)} \leqslant \exp\Big(-2\sum_{m=1}^{M}\gamma_m^2\Big)$$

，其中，$\gamma_m = \frac{1}{2} - e_m$

e_m越小，说明当前分类器效果越好，γ_m越大，$\frac{1}{2}$可以看做是分类问题中，随机猜测分类结果的最大误差率。γ_m可以解释成当前的基本分类器对于随机猜测结果的提升程度。提升程度越大，γ_m越大，上界$\exp\Big(-2\sum_{m=1}^{M}\gamma_m^2\Big)$就越小，训练误差随着迭代次数的增加而越小，减小的速度是以指数速度变小的，也就是说，训练次数增多的时候，训练误差降低的速度会变快。

19.3 AdaBoost算法的解释

在本节中，将AdaBoost算法用第1章的统计学习方法的三要素分析该模型、策略、算法。

> 问题：二分类问题
> 模型：加法模型$f(x) = \sum_{m=1}^{M} \beta_m b(x;\gamma_m)$
> 策略：损失函数为指数函数$L(y, f(x)) = \exp[-yf(x)]$
> 算法：前向分步算法

AdaBoost从模型形式上看，该模型是一个加法模型$f(x) = \sum_{m=1}^{M} \beta_m b(x;\gamma_m)$，其中$b(x;\gamma_m)$为基函数，该基函数可能是关于$x$的线性函数，也可能是二次函数，这个函数的形式不变，但是有一个参数γ_m，在每一个求和项中，γ_m是不一样的，求解该模型，M即求和项项数是已知的，基函数$b(x;\gamma_m)$也是给定的，所要求解的就是β_m和γ_m。

AdaBoost的损失函数是一个指数函数$L(y, f(x)) = \exp[-yf(x)]$，二分类中$y \in \{-1, +1\}, f(x) \in \{-1, +1\}$，如果预测值等于真实值（预测正确即$y = f(x)$），$yf(x) = 1$，损失为$e^{-1}$，如果不相等（预测错误即$y \neq f(x)$），损失为$e$，这个与我们平常看到的损失不一致，这里即使预测正确，也是有损失的，预测错误比预测正确的损失大。

19.3.1 前向分步算法

输入：训练数据集$T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$，损失函数$L(y, f(x))$，基函数集$\{b(x;\gamma)\}$；
输出：加法模型$f(x)$
(1)初始化$f_0(x) = 0$

(2)对$m=1,2,\cdots,M$

(a)极小化损失函数

$$(\beta_m,\gamma_m)=\arg\min_{\beta,\gamma}\sum_{i=1}^{N}L(y_i,f_{m-1}(x_i)+\beta b(x_i;\gamma))$$

(b)更新

$$f_m(x)=f_{m-1}(x)+\beta_m b(x;\gamma_m)$$

(3)得到加法模型

$$f(x)=f_M(x)=\sum_{m=1}^{M}\beta_m b(x;\gamma_m)$$

这样，前向分步算法将同时求解从$m=1$到M所有参数β_m,γ_m的优化问题简化为逐次求解各个β_m,γ_m的优化问题。

19.3.2 前向分步算法与AdaBoost

有了模型和策略，当拟合加法模型时，需要求解很多的参数，有一个比较简单的算法（前向分步算法），已知加法模型$f(x)$由M项求和组成的，用迭代的方法求解$f(x)$，初始化$f_0(x)=0$，现考虑$f_1(x)=f_0(x)+\beta_1 b(x;\gamma_1)$，求解$\gamma_1,\beta_1$，使得$\gamma_1^*,\beta_1^*=\arg\max_{\gamma,\beta}\frac{1}{N}\sum_{i=1}^{N}L(y_i,f_1(x_i))$，可以得到$f_1(x)=f_0(x)+\beta_1^* b(x;\gamma_1^*)$，下一次更新为$f_2(x)=f_1(x)+\beta_2 b(x;\gamma_2)$，求解$\beta_2,\gamma_2$，依然利用经验风险最小这个准则，$\gamma_2^*,\beta_2^*=\arg\max_{\gamma,\beta}\frac{1}{N}\sum_{i=1}^{N}L(y_i,f_2(x_i))$，得到$f_2(x)=f_1(x)+\beta_2^* b(x;\gamma_2^*)$，以此类推，可以求得$f(x)=f_M(x)$。其中，每步中 得到的$b(x;\gamma_m)$就是$G_m(x)$，$\beta_m$就是$\alpha_m$。

当用这样的框架重新分析AdaBoost算法之后，可以进行很多的变形，当损失函数不同，很多AdaBoost变形，都是通过替换框架中的策略得到的。

19.4 提升树

提升树与AdaBoost算法的思路是非常像的，也是考虑一个加法模型，AdaBoost算法中没有规定所用的分类器，提升树中，基本分类器为分类树或回归树（分类树用来做分类问题，回归树用来做回归问题），所用的算法依然是前向分步算法。对于分类树的提升树，如果是二分类问题，该算法和AdaBoost算法是等价的，所以本节中没有介绍分类问题的提升树算法。

基本分类器：分类树或回归树

提升树模型：$f_M(x)=\sum_{m=1}^{M}T(x;\Theta_m)$

前向分步算法：

$$f_m(x)=f_{m-1}(x)+T(x;\Theta_m)$$

$$\hat{\Theta}_m=\arg\max\sum_{i=1}^{N}L(y_i,f_{m-1}(x_i)+T(x_i;\Theta_m))$$

19.4.1 回归问题的提升树算法

提升树模型是由M个基本分类器构成的，每一个基本分类器都是一个回归树$T(x;\Theta_m)$，回归树中的Θ_m都包含了哪些项？首先对应了一个空间上的划分（每一个叶子结点），还包括了每一个叶子结点的拟合值c，对于前向分步算法，首先初始化$f_0(x)=0$，考察加法模型的第一项$f_1(x)=f_0(x)+T(x;\Theta_1)$，求解$\Theta_1$采用最小化经验风险$\frac{1}{N}\sum_{i=1}^{N}L(y_i,f_1(x_i))$，对于提升树，一般采用平方误差损失$L(y,f(x))=(y-f(x))^2$

> 算法8.3（回归问题的提升树算法-平方误差损失）
> 输入：训练数据集$T=\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\},x_i\in\mathcal{X}\subseteq\mathrm{R}^n,y_i\in\mathcal{Y}\subseteq\mathrm{R}$
> 输出：提升树$f_M(x)$
> (1)初始化f_0(x)=0
> (2)对$m=1,2,\cdots,M$
> (a)计算残差$r_{mi}=y_i-f_{m-1}(x_i),i=1,2,\cdots,N$
> (b)拟合残差r_{mi}学习一个回归树，得到$T(x;\Theta_m)$
> (c)更新$f_m(x)=f_{m-1}(x)+T(x;\Theta_m)$
> (3)得到回归问题提升树
> $$f_M(x)=\sum_{m=1}^{M}T(x;\Theta_m)$$

19.4.2 对算法8.3的解释

首先初始化$f_0(x)=0$，对于第1棵树直接拟合得到$y,f_0(x)$，通过最小化y_i和$f_0(x_i)$的平方误差损失求解Θ_1，然后得到回归树模型$T(x;\Theta_1)$，用训练数据集训练，求出预测值，通过预测值和真实值计算每一个样本点上的残差r_1，当求解下一个棵树时，直接用残差$r_{1,i}$学习回归树（即残差r_1为y），继续用r_1和$f_1(x)$的平方误差损失最小求解Θ_2，得到第2棵树$T(x;\Theta_2)$，目前有了一个新的模型$f_2(x)=T(x;\Theta_1)+T(x;\Theta_2)$，继续计算残差，然后拟合第3棵树，最终得到第M棵树$f_M(x)$。

为什么这里是通过拟合残差来学习回归树呢？书中也给出了推导，对于任意一个样本点，y和$f(x)$拟合之间的损失为$L(y,f(x))=(y-f(x))^2$，在前向分步算法中，第M步得到的$f_m(x)=y-f_{m-1}(x)+T(x;\Theta_m)$

$$\begin{aligned}\therefore\quad & L(y,f(x))\\ =\quad & \left[y-f_{m-1}(x)-T(x;\Theta_m)\right]^2\\ =\quad & [r_{m-1}-T(x;\Theta_m)]^2\\ =\quad & L(r_{m-1},T(x;\Theta_m))\end{aligned}$$

最小化$L(y,f_{m-1}(x)+T(x;\Theta_m))$时，相当于最小化$L(r_{m-1},T(x;\Theta_m))$，所以用整体样本中$y$和$f_m(x)$的经验风险最小化等价于用上一步残差$r_{m-1}$拟合第$M$步的回归树，可求解出$\Theta_m$，$\Theta_m$包含两个部分：空间的划分（每一个叶子结点）和每一个叶子结点上的拟合值c。

19.4.3 梯度提升算法

假如不用平方误差损失，换成其他的损失函数时，残差不好计算，这个时候就可以用新的提升算法（算法8.4——梯度提升算法），梯度提升算法依然采用拟合每一个训练数据，由于不是采用平方误差损失，之前用残差拟合下一棵树的方法就不适用了，就替换为下面的算法。

算法8.4（梯度提升算法）

输入：训练数据集$T=\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\}, x\in\mathcal{X}\subseteq \mathrm{R}^n, y\in\mathcal{Y}\subseteq \mathrm{R}$，损失函数$L(y,f(x))$

输出：回归树$\hat{f}(x)$

(1)初始化$f_0(x)=\arg\min\limits_{c}\sum\limits_{i=1}^{N}L(y_i,c)$

(2)对$m=1,2,\cdots,M$ (a)对$i=1,2,\cdots,N$，计算

$$r_{mi}=-\Big[\frac{\partial L(y_i,f(x_i))}{\partial f(x_i)}\Big]_{f(x)=f_{m-1}(x)}$$

(b)对r_{mi}拟合一个回归树，得到第m棵树的叶结点区域$\mathrm{R}_{mj},j=1,2,\cdots,J$

(c)对$j=1,2,\cdots,J$，计算

$$c_{mj}=\arg\min_{c}\sum_{x_i\in \mathrm{R}_{mj}}L(y_i,f_{m-1}(x_i)+c)$$

(d)更新$f_m(x)=f_{m-1}(x)+\sum\limits_{j=1}^{J}c_{mi}I(x\in \mathrm{R}_{mj})$

(3)得到回归树

$$\hat{f}(x)=f_M(x)=\sum_{m=1}^{M}\sum_{j=1}^{J}c_{mj}I(x\in \mathrm{R}_{mj})$$

在算法中，依然是需要一个类似与残差的变量，该值（r_{mi}）是用负梯度计算的，r_{mi}是损失函数关于$f(x)$的负梯度，然后用r_{mi}拟合新的回归树，一般情况下是求解Θ_m，Θ_m包含两个部分：空间上的划分$\mathrm{R}_{mj},j=1,2,\cdots,J$和拟合值$c_{mj}$。

这里就有两个问题：

1. 为什么可以用负梯度近似代替残差？
2. 计算残差时，为什么只能通过残差求解Θ_m的第1部分R_{mj}，第1部分还需要重新计算？

解答： 假设$L(y_i,f(x_i))=[y_i-f(x_i)]^2$，则$-\frac{\partial L(y_i,f(x_i))}{\partial f(x_i)}=-(-2[y_i-f(x_i)])=2[y_i-f(x_i)]=r_{mi}$，推广用到的是一阶泰勒展开的近似，从平方损失的例子中可以看到，用负梯度确实可以代替残差。但是在推导的过程中，可以看到前面有一个系数2，所以负梯度只是残差的代替，并不能等同于残差。这也就是为什么在(c)步中要根据原始的y_i和$f(x)$的形式求解c

由于已经确定了每一个叶子结点区域R_{mj}，对于每一个叶子结点j，只需要考虑该叶子结点，而不需要考虑其他的叶子结点，当求这一个叶子结点（即求分到这个叶子结点的样本点$x_i\in \mathrm{R}_{mj}$）中的样本点x_i，求这些点的经验损失和（R_{mj}上对应样本点的经验风险），真实值为y_i，拟合值为$f_{m-1}(x_i)$再加上第m步中得到的数，这里只关心R_{mj}区域中的叶子结点部分，对于这个叶子结点，新得到的这个数（拟合值）为c，所以整体的拟合值为$f_{m-1}(x_i)+c$。

通过经验风险最小，求得这个叶子结点上那棵新树对应的c值，$f_m(x_i)+c$表示为在R_{mj}区域范围的样本点的拟合值$f_m(x_i)$。所以对于一个新的回归树，已经求了J（第m棵树中叶子结点的个数）个这样的c，更新的形式为$f_m(x)=f_{m-1}(x)+\sum\limits_{j=1}^{J}c_{mj}I(x\in \mathrm{R}_{mj})$，对于每个样本点，只在其中的一个指示函数（indicator function）中等于1，因为只能属于其中的一个R_{mj}，当$I(x\in \mathrm{R}_{mj})=1$时，才累积求和。对于每一棵树$f_m(x)$，都有一个$c_m$，对于每一棵树上，又划分了$J$个区域，所以拟合值为$c_{mj}$。对于 每一个样本点，在每一棵树上，都只有一个指示函数为1，虽然(3)步求和是M,J，但是最后求解时，只有M项的和。

20 第8章-提升方法-AdaBoost训练误差

AdaBoost算法非常受欢迎，主要有3个原因：

1. 分类效果非常好
2. 这个算法非常简单，学习的每一个基本分类器都是很简单的分类器
3. 有比较可靠的理论基础

通过一个例子解释一下算法中一直提到的权值以及样本的分布，假设有一个训练数据集D_1满足某个分布，对其进行抽样得到三个样本点$(x_1, y_1), (x_2, y_2), (x_3, y_3)$，这三个样本点是相互独立的，所以经验概率均为$\frac{1}{3}$，以上这些构成了一个概率分布为$P_1$，通过构建基本分类器，对于每个分类器在训练数据集上有一个分类的效果，对于分类效果差的样本，希望提高在下一轮训练模型时的权值。假设有一个新的数据集D_2，也有三个样本点$(x_1, y_1), (x_2, y_2), (x_3, y_3)$，这些样本点不是独立同分布的，根据$D_1$和在模型上的分类效果，构建一个新的分布$P_2$，这个新的分布对应每个样本点取到的概率就不是相同的值，假设(x_1, y_1)是误分类点，将权值提高到$\frac{2}{3}$，(x_2, y_2)和(x_3, y_3)是分类正确的，权值都为$\frac{1}{6}$。

20.1 定理8.1的证明

定理8.1（AdaBoost的训练误差界）AdaBoost算法最终分类器的训练误差界为

$$\frac{1}{N}\sum_{i=1}^{N} I(G(x_i) \neq y_i) \leqslant \frac{1}{N}\sum_{i} \exp(-y_i f(x_i)) = \prod_{m} Z_m$$

其中，$G(x_i) = \text{sign}(f(x)) = \text{sign}(\quad \alpha_m G_m(x))$

证明：对于正确分类点，$I(G(x_i) \neq y_i) = 0 \leqslant \exp(-y_i f(x_i)) = e^{-1}$，对于误分类点，$I(G(x_i) \neq y_i) = 1 \leqslant \exp(-y_i f(x_i)) = e^{1}$，所以可得$I(G(x_i) \neq y_i) \leqslant \exp(-y_i f(x_i))$，故

$\frac{1}{N}\sum_{i=1}^{N} I(G(x_i) \neq y_i) \leqslant \frac{1}{N}\sum_{i} \exp(-y_i f(x_i))$成立。

现证明$\frac{1}{N}\sum_{i} \exp(-y_i f(x_i)) = \prod_{m} Z_m$：

$$\because Z_m = \sum_{i} w_{mi} \exp(-\alpha_m y_i G_m(x_i))$$

$$w_{m+1,i} = \frac{w_{mi}}{Z_m}\exp(-\alpha_m y_i G_m(x_i))$$

$$\therefore Z_m \cdot w_{m+1,i} = w_{mi} \exp(-\alpha_m y_i G_m(x_i))$$

$$\because \begin{cases} Z_1 \cdot w_{2,i} = \quad w_{1,i} \exp(-\alpha_1 y_i G_1(x_i)) \\ Z_2 \cdot w_{3,i} = \quad w_{2,i} \exp(-\alpha_2 y_i G_2(x_i)) \\ \qquad\qquad \vdots \\ Z_{M-1} \cdot w_{M,i} = \quad w_{M-1,i} \exp(-\alpha_{M-1} y_i G_{M-1}(x_i)) \end{cases}$$

将上述各式都相乘，相同的项可以约去，得到：

$$\prod_{m=1}^{M-1} Z_m w_{M,i} = w_{1,i} \exp(-y_i \sum_{m=1}^{M-1} \alpha_m G_m x_i)$$

对比要证明的等式$\frac{1}{N}\sum_{i} \exp(-y_i f(x_i)) = \prod_{m=1}^{M} Z_m$，其中$f(x_i) = \quad \sum_{m=1}^{M} \alpha_m G_m(x_i)$

左右两边都乘以$\exp(-\alpha_M y_i G_M(x_i))$

$$\because w_{1,i} = \frac{1}{N}$$

$$\therefore \prod_{m=1}^{M-1} Z_m w_{M,i} \cdot \exp(-\alpha_M y_i G_M(x_i)) = \frac{1}{N}\exp(-y_i f(x_i))$$

由于最终得到的是关于整个数据集的，所以需要两边求和。

$$\therefore \prod_{m=1}^{M-1} Z_m \sum_i w_{M,i} \exp(-\alpha_M y_i G_M(x_i)) = \frac{1}{N} \sum_i \exp(-y_i f(x_i))$$

$$\because Z_M = \sum_i w_{M,i} \exp(-\alpha_M y_i G_M(x_i))$$

$$\therefore \prod_{m=1}^{M-1} Z_m \cdot Z_M = \frac{1}{N} \sum_i \exp(-y_i f(x_i))$$

$$\therefore \prod_{m=1}^{M} Z_m = \frac{1}{N} \sum_i \exp(-y_i f(x_i))$$，得证.

定理的直观理解：训练误差可以被Z_m的连乘控制住，如果要训练误差小，就只要最小化每一个Z_m，$Z_m = \sum_i w_{mi} \exp(-\alpha_m y_i G_m(x_i))$，$w_{mi}$是通过上一轮训练的模型得到的，$G_m$是本轮已经训练得到的，所以已知$w_{mi}, G_m$，求解的是$\alpha_m$，通过最小化$Z_m$求解$\alpha_m = \arg\min_{\alpha_m} Z_m$。

$$\frac{\partial Z_m}{\partial \alpha_m} = \sum_i \Big(- w_{mi} y_i G_m(x_i) \exp(-\alpha_m y_i G_m(x_i)) \Big)$$

书中给出的$\alpha_m = \frac{1}{2} \ln \frac{1 - e_m}{e_m}$

20.2 定理8.2的证明

定理8.2（二类分类问题AdaBoost的训练误差界）

$$\prod_{m=1}^{M} Z_m = \prod_{m=1}^{M} [2\sqrt{e_m(1-e_m)}] = \prod_{m=1}^{M} \sqrt{(1-4\gamma_m^2)} \leqslant \exp\Big(-2 \sum_{m=1}^{M} \gamma_m^2 \Big)$$

，其中，$\gamma_m = \frac{1}{2} - e_m$

证明：

$$Z_m = \sum_{i=1}^{N} w_{mi} \exp(-\alpha_m y_i G_m(x_i))$$

$$= \sum_{y_i = G_m(x_i)} w_{mi} e^{-\alpha_m} + \sum_{y_i \neq G_m(x_i)} w_{mi} e^{\alpha_m}$$

$$\because e_m = \sum_{G_m(x_i) \neq y_i} w_{mi}, \alpha_m = \frac{1}{2} \ln \frac{1 - e_m}{e_m}$$

$$\therefore \sum_{y_i = G_m(x_i)} w_{mi} e^{-\alpha_m} + \sum_{y_i \neq G_m(x_i)} w_{mi} e^{\alpha_m} = (1 - e_m) e^{-\alpha_m} + e_m e^{\alpha_m}$$

$$\because e^{-\alpha_m} = \sqrt{\frac{e_m}{1 - e_m}}, e^{\alpha_m} = \sqrt{\frac{1 - e_m}{e_m}}$$

$$\therefore (1 - e_m) e^{-\alpha_m} + e_m e^{\alpha_m} = 2\sqrt{e_m(1 - e_m)}$$

$$\therefore Z_m = 2\sqrt{e_m(1 - e_m)}$$

$$\because \gamma_m = \frac{1}{2} - e_m$$

$$\therefore 2\sqrt{e_m(1 - e_m)} = \sqrt{1 - 4\gamma_m^2}$$

$$\therefore \prod_{m=1}^{M} Z_m = \prod_{m=1}^{M} [2\sqrt{e_m(1 - e_m)}] = \prod_{m=1}^{M} \sqrt{(1 - 4\gamma_m^2)}$$

再考虑$\prod_{m=1}^{M}\sqrt{(1-4\gamma_m^2)} \leqslant \exp\left(-2\sum_{m=1}^{M}\gamma_m^2\right)$

书中介绍的是由e^x和$\sqrt{1-x}$在点$x=0$的泰勒展开式推导，泰勒展开是用一个多项式逼近任意一个函数。

考察$\sqrt{(1-4\gamma^2)} \leqslant \exp(-2\gamma^2)$

$f(x)=\sqrt{1-x}=(1-x)^{\frac{1}{2}}$

$f'(x)=-\frac{1}{2}(1-x)^{-\frac{1}{2}}$

$f''(x)=-\frac{1}{4}(1-x)^{-\frac{3}{2}}$

$f(x)$在$x=0$处的二阶泰勒展开为：

$$\begin{aligned} f(x) &= f(0)+xf'(0)+\frac{1}{2}xf''(0)+\cdots \\ &= 1-\frac{1}{2}x-\frac{1}{8}x^2+\cdots \end{aligned}$$

$\therefore f(4\gamma^2)\approx 1-2\gamma^2-2\gamma^4$

$g(x)=e^x$

$g'(x)=e^x$

$g''(x)=e^x$

$g(x)$在$x=0$处的二阶泰勒展开为：

$$\begin{aligned} g(x) &= g(0)+xg'(0)+\frac{1}{2}x^2g''(0)+\cdots \\ &= 1+x+\frac{1}{2}x^2+\cdots \end{aligned}$$

$\therefore g(-2\gamma^2)\approx 1-2\gamma^2+2\gamma^4$

$$\therefore \begin{matrix} f(4\gamma^2)=\sqrt{(1-4\gamma^2)}\approx 1-2\gamma^2-2\gamma^4 \\ g(-2\gamma^2)=\exp(-2\gamma^2)\approx 1-2\gamma^2+2\gamma^4 \end{matrix}$$

$\because \gamma_m=\frac{1}{2}-e_m$，$\gamma_m$的取值范围是$[0,\frac{1}{2}]$

所以当更高阶次方出现的时候，后面的高阶项趋近于0，那么影响$g(-2\gamma^2)$和$f(4\gamma^2)$关系的只有前面的项，判断$1-2\gamma^2-2\gamma^4$和$1-2\gamma^2+2\gamma^4$的大小，等同于判断$f(4\gamma^2)$和$g(-2\gamma^2)$的大小。

$\because 1-2\gamma^2-2\gamma^4 \leqslant 1-2\gamma^2+2\gamma^4$

$\therefore f(4\gamma^2)\leqslant g(-2\gamma^2)$

$\therefore \prod_{m=1}^{M}\sqrt{(1-4\gamma_m^2)} \leqslant \exp\left(-2\sum_{m=1}^{M}\gamma_m^2\right)$，得证.

21 第8章-提升方法-前向分步算法

本节介绍当使用前向分步算法，求解一个以指数函数为损失函数的加法模型时，得到的结果与AdaBoost算法之间的关系。书中在8.3节已经给出了结论，这两个算法是等价的，现求证该结论。

证明：

21.1 G^*求解

前向分步算法学习的是加法模型$f(x)=\sum_{m=1}^{M}\alpha_m G_m(x)$，其中$G_m(x)$是基本分类器，取值为$\{+1,-1\}$，损失函数为$L(y,f(x))=\exp(-yf(x))$，$y\in\{+1,-1\}$，在该推导过程中，使用的是数学归纳法，已经得到了第$m-1$步的分类器$f_{m-1}(x)=\sum_{j=1}^{m-1}\alpha_j G_j(x)$，求解$f_m(x)=f_{m-1}(x)+\alpha_m G_m(x)$

$$\alpha_m, G_m = \arg\min_{\alpha,G}\sum_{i=1}^{N}\exp\Big\{-y_i\Big[f_{m-1}(x_i)+\alpha G(x_i)\Big]\Big\}$$

令$\bar{w}_{m,i}=\exp\Big(-y_i f_{m-1}(x_i)\Big)$

$\therefore \arg\min_{\alpha,G}\sum_{i=1}^{N}\exp\Big\{-y_i\Big[f_{m-1}(x_i)+\alpha G(x_i)\Big]\Big\}=\arg\min_{\alpha,G}\sum_{i=1}^{N}\bar{w}_{m,i}\exp(-y_i\alpha G(x_i))$，其中$\bar{w}_{m,i}$与$\alpha$和$G$无关，但$\bar{w}_{m,i}$依赖$f_{m-1}(x)$，随着每一轮迭代而发生变化。

因为$y_i G(x_i)$取值为$\{+1,-1\}$，将$\sum_{i=1}^{N}\bar{w}_{m,i}\exp(-y_i\alpha G(x_i))$拆开，可得：

$$\begin{aligned}
&\sum_{i=1}^{N}\bar{w}_{m,i}\exp(-y_i\alpha G(x_i))\\
=&\sum_{i\in M_1}\bar{w}_{m,i}\exp(-\alpha)+\sum_{i\in M_2}\bar{w}_{m,i}\exp(\alpha)\\
=&\sum_{i\in M_1}\bar{w}_{m,i}\exp(-\alpha)+\sum_{i\in M_2}\bar{w}_{m,i}\exp(-\alpha)+\sum_{i\in M_2}\bar{w}_{m,i}\Big[\exp(\alpha)-\exp(-\alpha)\Big]\\
=&\exp(-\alpha)\sum_{i}\bar{w}_{m,i}+\Big[\exp(\alpha)-\exp(-\alpha)\Big]\sum_{i\in M_2}\bar{w}_{m,i}\\
=&\exp(-\alpha)\sum_{i}\bar{w}_{m,i}+\Big[\exp(\alpha)-\exp(-\alpha)\Big]\sum\bar{w}_{m,i}I(y_i\neq G(x_i))
\end{aligned}$$

，其中M_1为正类集合，M_2为负类集合。

$\therefore G_m^*=\arg\min_{G}\quad\bar{w}_{m,i}I(y_i\neq G(x_i))$

21.2 α^*求解

$$\alpha_m=\arg\min_{\alpha}\sum\bar{w}_{m,i}\exp(-\alpha y_i G_m^*(x_i))$$

$$\begin{aligned}
&\because\quad\bar{w}_{m,i}\exp(-\alpha y_i G_m^*(x_i))\\
=&\sum_{i\in M_1}\bar{w}_{m,i}\exp(-\alpha)+\sum_{i\in M_2}\bar{w}_{m,i}\exp(\alpha)\\
=&(e^{\alpha}-e^{-\alpha})\sum_{i=1}^{N}\bar{w}_{m,i}I(y_i\neq G(x_i))+e^{-\alpha}\sum_{i=1}^{N}\bar{w}_{m,i}
\end{aligned}$$

求解上式最小的α，则令上式求导等于0：

$$\therefore(e^{\alpha}+e^{-\alpha})\sum_{i=1}^{N}\bar{w}_{m,i}I(y_i\neq G(x_i))-e^{-\alpha}\sum_{i=1}^{N}\bar{w}_{m,i}=0$$

$$\therefore(e^{2\alpha}+1)\sum_{i=1}^{N}\bar{w}_{m,i}I(y_i\neq G(x_i))=\sum_{i=1}^{N}\bar{w}_{m,i}$$

$$\therefore \alpha_m^* = \frac{1}{2}\ln\frac{\bar{w}_{m,i} - \bar{w}_{m,i} I(y_i \neq G(x_i))}{\bar{w}_{m,i} I(y_i \neq G(x_i))} = \frac{1}{2}\ln\frac{1 - \frac{\bar{w}_{m,i} I(y_i \neq G(x_i)}{\bar{w}_{m,i}}}{\frac{\bar{w}_{m,i} I(y_i \neq G(x_i)}{\bar{w}_{m,i}}}$$

令$e_m = \frac{\bar{w}_{m,i} I(y_i \neq G(x_i)}{\bar{w}_{m,i}}$

则$\alpha_m^* = \frac{1}{2}\ln\frac{1-e_m}{e_m}$

21.3 权重$\bar{w}_{m,i}$的更新

目前G和α都和AdaBoost算法是一致的，现在只差一个权重的更新。

之前令$\bar{w}_{m,i} = \exp\left(-y_i f_{m-1}(x_i)\right)$

$$\begin{aligned}\bar{w}_{m,i} &= \exp\left(-y_i f_{m-1}(x_i)\right) \\ &= \exp\left(-y_i \sum_{j=1}^{m-1} \alpha_j G_j(x_i)\right) \\ &= \prod_j \exp\left(-y_i \alpha_j G_j(x_i)\right)\end{aligned}$$

上述的公式与权值更新只相差一个规范化因子，因为每一轮所乘的规范化因子对每一个样本都是一样的，所以不影响求解结果，该值与AdaBoost算法也是一致的。

现在还差最后一个问题，因为之前讲的是通过数学归纳法推导的，只通过$m-1$步推出了m步，那么第1步是如何推导的呢？第1步的推导过程也可以根据书中的推导出来，唯一的区别就是在最小化损失函数

$\alpha_m, G_m = \arg\min \quad \exp(-y_i f_{m-1}(x_i) + \alpha G(x_i))$ ，在第1步中$f_{m-1}(x_i)$是不存在的，可得

$\alpha_m, G_m = \arg\min \quad \exp(\alpha G(x_i))$ ，剩下的推导过程是一致的，得证.

22 第9章-EM算法及推广-导读

从第9章开始，我们关注的点就和前面的章节不一样了，在第2-8章是分类问题，都属于监督学习，第9章EM算法是非监督学习，本章的推导比较多，在导读部分，重点介绍EM算法的应用、处理问题的特点、与之前算法的区别、EM算法的解决问题流程以及EM算法的性质和简单的变形。

22.1 EM算法的引入

EM算法是用来估计概率分布的，该概率分布的数据是有缺失的，假如数据没有缺失，$X \sim N(\mu, \sigma^2)$，其中μ, σ^2是未知的，用一组数据$(x_1, x_2, \cdots, x_N)$去估计μ, σ^2，估计方法一般为极大似然估计和贝叶斯估计，以上是正常估计密度函数所用的方法。

估计随机变量的密度函数属于无监督学习，但是在EM算法中，观测数据是有缺失的，书中有一个例子“三硬币模型”。

22.1.1 EM算法

22.1.1.1 三硬币模型

随机变量$Z \sim b(1, \pi)$表示观测不到的数据，以概率π取值为1，以概率$1-\pi$取值为0，第1次实验得到z_1，根据该值得到观测值y_1，如果$z_1 = 1$，则$y_1 \sim b(1, p)$，如果$z_1 = 0$，则$y_1 \sim b(1, q)$，已知y_1，并不知道z_1取0或1，只知道$z_1 \sim b(1, \pi)$，π, p, q未知。

根据上面的规则可得到$(z_2, y_2), \cdots, (z_N, y_N)$，令$\theta = (\pi, p, q)$，将$(z, y)$称为完全数据，观测数据$y$称为不完全数据，所以能看到的就是不完全数据，对应完全数据有概率分布$P(y, z) = P(z)P(y|z)$，需要估计的是$P(y, z)$，也就是π, p, q。一般使用的是极大似然估计，需要写出观测值：

$$P(y|\theta) = \sum_z P(y, z|\theta) = \pi p^y (1-p)^{1-y} + (1-\pi) q^y (1-q)^{1-y}$$

所有的观测值Y为

$$\prod_{i=1}^{N} P(y_i|\theta) = \prod_{i=1}^{N} \left[\sum_z P(y_i, z|\theta) \right] = \prod_{i=1}^{N} \left[\pi p^{y_i} (1-p)^{1-y_i} + (1-\pi) q^{y_i} (1-q)^{1-y_i} \right]$$

极大似然估计就是极大化似然函数$\prod_{i=1}^{N} P(y_i|\theta)$，为了简单一点可以取对数，但是后面的形式非常复杂，直接求导等于0非常困难，这就需要EM算法，用迭代的方式求解。

22.1.1.2 EM算法的解释

EM算法并不是直接处理似然函数的，为了简化这个问题，将最大化观测值的似然函数转换为最大化完全数据的似然函数，书中给出了两个角度的解释，第1种角度的解释在9.1节，第2种在9.4节。
完全数据的似然函数：

$$\ln \prod_{i=1}^{N} P(y_i, z_i|\theta)$$

但是现在有一个问题：z_i未知。一旦给出y, z分布的形式，就可以将y, z的联合概率密度写成关于θ的函数，每一个y_i的值是已知的，但对于z_i的只是未知的，要如何处理z_i，就可以利用EM算法中的E步解决：将z_i有关的项，用期望代替$z_i \to E(z)$，对应不同的z，期望$E(z)$都是相同的，求期望的时候需要知道密度函数（随机变量的分布），这个分布由估计量θ, y_i决定。给定y_i, θ求z_i的分布可以用贝叶斯公式，y_i已知，但是θ是需要估计的，所以这个时候采用迭代的方法。
E步： 首先初始化$\theta^{(0)}$，然后进行迭代，第i步记为$\theta^{(i)}$，这样就可以求出期望$E(z_i)$，将期望代入似然函数中。
M步： 最大化似然函数，求解$\theta^{(i+1)} = \arg\max_\theta \ln \prod_{i=1}^{N} P(y_i, E(z)|\theta)$
重复E步，求期望$E(z)$，再继续M步，不断迭代。
总结： 在EM算法中，将极大化观测数据的似然函数转换为极大化完全数据的似然函数，在该过程中，隐变量z_i的值是未知的，采用给定参数的条件下，$E(z)$代替每一个z_i的值，得到用期望代替之后的似然函数，最大化该似然函数，一步一步对θ进行迭代。

22.1.1.3 EM算法

输入：观测变量数据Y，隐变量数据Z，联合分布$P(Y, Z|\theta)$，条件分布$P(Z|Y, \theta)$
输出：模型参数θ
(1)选择参数的初值$\theta^{(0)}$，开始迭代
(2)**E步：**记$\theta^{(i)}$为第i次迭代参数θ的估计值，在第$i+1$次迭代的E步，计算

$$\begin{aligned} Q(\theta, \theta^{(i)}) &= E_Z \left[\ln P(Y, Z|\theta) | Y, \theta^{(i)} \right] \\ &= \sum_Z \ln P(Y, Z|\theta) P(Z|Y, \theta^{(i)}) \end{aligned}$$

这里，$P(Z|Y, \theta^{(i)})$是在给定观测数据Y和当前的参数估计$\theta^{(i)}$下隐变量数据Z的条件概率分布
(3)**M步：**求使得$Q(\theta, \theta^{(i)})$极大化的θ，确定第$i+1$次迭代的参数估计值$\theta^{(i+1)}$

$$\theta^{(i+1)} = \arg\max_\theta Q(\theta, \theta^{(i)})$$

(4)重复第(2)步和第(3)步，直到收敛（收敛条件：$\theta^{(i)}$和$\theta^{(i+1)}$很接近，或者是$Q(\theta^{(i+1)}, \theta^{(i)})$和$Q(\theta^{(i)}, \theta^{(i-1)})$很接近）。
函数$Q(\theta, \theta^{(i)})$是EM算法的核心，称为Q函数。

22.1.2 EM算法的导出

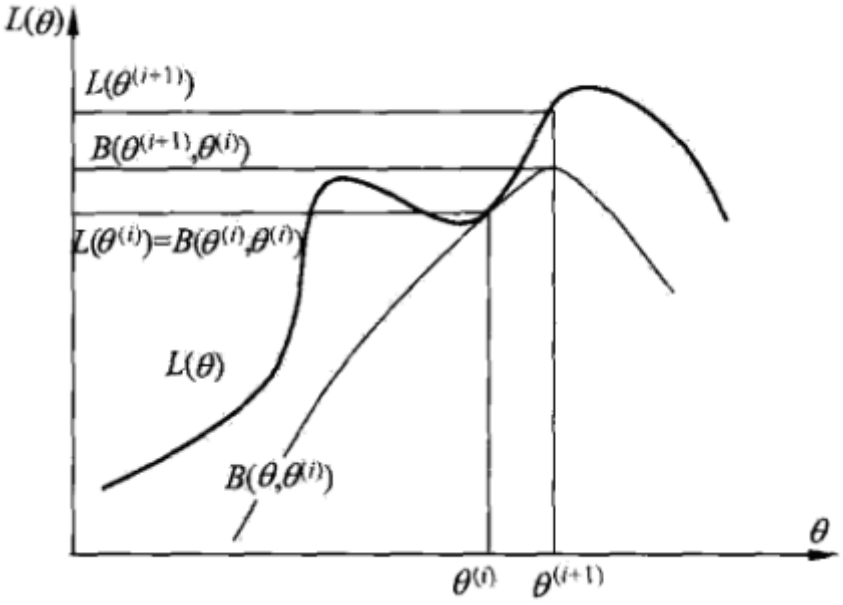

图9-1 EM算法的解释

图9-1中实线为观测数据的似然函数$L(\theta) = P(y|\theta)$，极大似然函数寻找该曲线的最大值，将这个点对应的$\hat{\theta}$作为θ的估计值，$Q(\theta, \theta^{(i)})$函数：当给定$\theta^{(i)}$时，该函数是一个关于θ的函数，图中是用$B(\theta, \theta^{(i)})$来代替的，$B(\theta, \theta^{(i)})$和$Q(\theta, \theta^{(i)})$取极值的点是一样的，$B(\theta, \theta^{(i)})$是$L(\theta)$的下界，求解$\hat{\theta}^{(i)} = \arg\max_{\theta} B(\theta, \theta^{(i)})$，这样就可以求解下一个点$\theta^{(i+1)}$使得函数$B$最大化，求解$\hat{\theta}^{(i+1)}$，不断迭代。

从图中看，$L(\theta)$并不是一个凸函数，如果求极大化，该函数并不是一个凹函数，也就是说，用这种迭代的方法求$L(\theta)$极大值时，当选取的初值不同，可能会收敛到不同的极大值处，所以EM算法得到的结果和初值的选择有关，初值选择不同，可能会得到不同的极大值点。

22.2 EM算法的收敛性

定理9.1 设$P(Y|\theta)$为观测数据的似然函数，$\theta^{(i)}(i = 1, 2, \cdots)$为EM算法得到的参数估计序列，$P(Y|\theta^{(i)})(i = 1, 2, \cdots)$为对应的似然函数序列，则$P(Y|\theta^{(i)})$是单调递增的，即

$$P(Y|\theta^{(i+1)}) \geqslant P(Y|\theta^{(i)})$$

定理9.2 设$L(\theta) = \log P(Y|\theta)$为观测数据的对数似然函数，$\theta^{(i)}(i = 1, 2, \cdots)$为EM算法得到的参数估计序列，$L(\theta^{(i)})(i = 1, 2, \cdots)$为对应的对数似然函数序列。

(1)如果$P(Y|\theta)$有上界，则$L(\theta^{(i)}) = \log P(Y|\theta^{(i)})$收敛到某一值$L^*$

(2)在函数$Q(\theta, \theta')$与$L(\theta)$满足一定条件下，由EM算法得到的参数估计序列$\theta^{(i)}$的收敛值θ^*是$L(\theta)$的稳定点。

本节给出了两个定理：第一个定理，用EM算法求解得到的θ，每次更新$\theta^{(i)}$时，都会使得观测数据的似然函数变大，这个性质可以保证求解最大化的θ值。将最大化观测数据的似然函数转换为最大化完全数据的似然函数，但是通过最大化完全数据的似然函数得到的θ的更新值，会使得观测数据的似然函数越来越大。第二个定理，对EM算法收敛性的一个说明，EM算法会收敛到一个局部的最大值。

22.3 EM算法在高斯混合模型学习中的应用

高斯混合模型是一个非常常见的模型，其密度估计问题也是一个比较复杂的问题，但是可以用EM算法解决。

22.3.1 高斯混合模型

定义9.2（高斯混合模型） 高斯混合模型是指具有如下形式的概率分布模型：

$$P(y|\theta)=\sum_{k=1}^{K}\alpha_k\phi(y|\theta_k)$$

其中，α_k是系数，$\alpha_k\geqslant 0,\sum_{k=1}^{K}\alpha_k=1$，$\phi(y|\theta_k)$是高斯分布密度，$\theta_k=(\mu_k,\sigma_k^2)$，

$$\phi(y|\theta)=\frac{1}{\sqrt{2\pi}\sigma_k}\exp\left(-\frac{(y-\mu_k)^2}{2\sigma_k^2}\right)$$

称为第k个分模型。

首先介绍高斯混合模型，只考虑简单的一维随机变量y，高斯分布就是正态分布，$y\sim N(\mu,\sigma^2)$，给定y的观测值，就可以很容易求出μ和σ^2，但是目前y不是来自高斯分布，而是有一定概率的来自于两个不同的高斯分布$N(\mu_1,\sigma_1^2)$和$N(\mu_2,\sigma_2^2)$，这个就是两个高斯分布的混合，并不知道y来自于哪一个高斯分布，这里涉及到了**隐变量**，这里所举的例子是两个高斯分布的混合，对于一般的形式如定义9.2给出。

θ包含了两个部分，第1部分是取值属于哪个高斯分布，模型中一共有K个高斯分布，另一部分是多项分布，一共有K个取值，只可能取到其中的一个值，对于不同的取值，都对应了一个概率，这个概率的和等于1，取不同的k个概率，对应的就是α_k，也就是以α_1的概率取了第1个高斯分布，以α_2的概率取了第2个高斯分布，以此类推。$\phi(y|\theta_k)$表示第k个高斯分布，其参数对应的μ,σ^2为θ_k，在给定参数θ_k时，$P(y|\theta)$为所求。

三个硬币的例子中，其模型为$P(y|\theta)=\sum_{Z}p(y,z|\theta)=\sum_{Z}p(z|\theta)P(y|z,\theta)$，对应于高斯混合模型中，$p(z|\theta)$是一个多项分布，一共有$K$项，且每一项的概率都为$\alpha_k$，$P(y|z,\theta)$就是确定了某一个$k$的取值之后的高斯分布，把之前的这两个公式代入之后求和，推导之后得到的公式为$P(y|\theta)=\sum_{k=1}^{K}\alpha_k\phi(y|\theta_k)$

22.3.2 高斯混合模型的参数估计

对于高斯混合模型的参数估计，书中给出了详细的推导过程，这里就简单的介绍一下，在EM算法中，极大化完全数据的似然函数$P(y,z|\theta)$，在完全数据中，z是观测不到的，所以在每个E步中都给定当前的θ值和y，得到期望$E(z|y,\theta^{(i)})$，用期望代替z，最后最大化转换之后的似然函数，得到θ的下一个估计值，这就是EM算法在高斯混合模型中的一个流程。

在该模型中，z是一个分类的变量，取值为$1,2,\cdots,K$，书中的推导过程是用了一个向量γ表示z，如果$z=1$，则$\gamma=(1,0,0,\cdots,0)$，如果$z=2$，则$\gamma=(0,1,0,\cdots,0)$，这个相当于One-hot，也就是说z是第i个高斯分布，在γ的第i个分量为1，其他分量都为0，以上是书中对隐变量的处理。

22.4 EM算法的推广

本节从另一个角度解释了EM算法，EM算法可以解释为F函数的极大-极大算法。

算法9.4（GEM算法2）
输入：观测数据，Q函数
输出：模型参数
(1)初始化参数$\theta^{(0)}$，开始迭代
(2)第$i+1$次迭代，第1步：记$\theta^{(i)}$为参数θ的估计值，计算

$$\begin{aligned}Q(\theta,\theta^{(i)})&=E_Z\left[\log P(Y,Z|\theta)|Y,\theta^{(i)}\right]\\&=\sum_{Z}P(Z|Y,\theta^{(i)})\log P(Y,Z|\theta)\end{aligned}$$

(3)第2步：求$\theta^{(i+1)}$使得

$$Q(\theta^{(i+1)},\theta^{(i)})>Q(\theta^{(i)},\theta^{(i)})$$

(4)重复(2)和(3)，直到收敛

算法9.5（GEM算法3）
输入：观测数据，Q函数
输出：模型参数
(1)初始化参数$\theta^{(0)}=(\theta_1^{(0)},\theta_2^{(0)},\cdots,\theta_d^{(0)})$，开始迭代
(2)第$i+1$次迭代，第1步：记$\theta^{(i)}=(\theta_1^{(i)},\theta_2^{(i)},\cdots,\theta_d^{(i)})$为参数$\theta=(\theta_1,\theta_2,\cdots,\theta_d)$的估计值，计算

$$\begin{aligned}Q(\theta,\theta^{(i)})=&E_Z\left[\log P(Y,Z|\theta)|Y,\theta^{(i)}\right]\\=&\sum_Z P(Z|Y,\theta^{(i)})\log P(Y,Z|\theta)\end{aligned}$$

(3)第2步：进行d次条件极大化：
首先，在$\theta_2^{(i)},\theta_3^{(i)},\cdots,\theta_k^{(i)}$保持不变的条件下求使得$Q(\theta,\theta^{(i)})$达到极大的$\theta_1^{(i+1)}$；
然后，在$\theta_1=\theta_1^{(i+1)},\theta_j=\theta_j^{(j)},j=3,4,\cdots,k$的条件下求使$Q(\theta,\theta^{(i)})$达到极大的$\theta^{(i+1)}$；
如此继续，经过d次条件极大化，得到$\theta^{(i+1)}=(\theta_1^{(i+1)},\theta_2^{(i+1)},\cdots,\theta_d^{(i+1)})$使得

$$Q(\theta^{(i+1)},\theta^{(i)})>Q(\theta^{(i)},\theta^{(i)})$$

(4)重复(2)和(3)，直到收敛

算法9.3和EM算法是完全一致的，只不过将Q函数写成了F函数形式。算法9.4中第(2)步和E步是一致的，第(3)步和M步是一致的，在EM算法中的M步是求解使得Q函数最大的θ，在算法9.4中，要求就放宽了一些，求解使得Q函数有提升的θ，这就是EM算法的第1个变形。对于EM算法的第2个变形（算法9.5），E步是不变的，在M步上做了改进，如果θ有很多不同的分量，分别求解使得Q函数最大的每一个分量，当求解第1个分量θ_1时，固定其他分量，以便求解分量θ_1，求解其他分量的方法相同，这样就将求解θ向量的最优解变成求解每一个分量的最优解，很多时候，这样的求解方法能大大简化计算。

23 第9章-EM算法及推广-EM算法的导出

EM算法本质上是估计一个密度函数，在估计密度函数时，通过观测值采用最大化似然函数估计参数θ。EM算法针对的是含有隐变量的密度函数的估计问题，这个时候直接最大化似然函数会比较困难，借鉴的算法思路和第6章改进的迭代尺度法是类似的，通过不等式放缩，将最大化观测数据的对数似然函数转化为另外一个比较好实现的式子，在推导EM算法时，与书上的数学符号保持一致。

首先有一个需要观测的向量θ，观测数据$Y=(y_1,y_2,\cdots,y_N)$，隐变量$Z=(z_1,z_2,\cdots,z_N)$，当求解θ时，似然函数为

$$\begin{aligned}L(\theta)&=\ln P(Y|\theta)\\&=\ln\sum_Z P(Y,Z|\theta)\\&=\ln\left(\sum_Z P(Z|\theta)P(Y|Z,\theta)\right)\end{aligned}$$

假设在第i次迭代后θ的估计值为$\theta^{(i)}$，希望新估计值θ能使$L(\theta)$增加，即$L(\theta)>L(\theta^{(i)})$，则可计算两者的差：

$$L(\theta)-L(\theta^{(i)})=\ln\left(\sum_Z P(Z|\theta)P(Y|Z,\theta)\right)-\ln P(Y|\theta^{(i)})$$

一般来说，对$\ln P_1P_2\cdots P_N$比较好处理，但是如果是$\ln\quad P_1P_2$就不好处理，为了将　求和符号去掉，用Jenson不等式进行缩放处理。

Jenson不等式：

$$f(\sum_i \alpha_i x_i) \geqslant \sum_i \alpha_i f(x_i)$$

其中函数f是凸函数，那么对数函数也是凸函数，$\sum_i \alpha_i = 1$，α_i表示权值，$0 \leqslant \alpha_i \leqslant 1$

对于上述形式，对Z求和，要如何凑出来一个具有Jenson不等式中的α_i呢？很容易想到，关于Z的密度函数，该密度函数取值求和为1，需要构造一个Z的概率分布。

$$\begin{aligned} L(\theta) - L(\theta^{(i)}) =& \ln\left(\sum_Z P(Z|\theta)P(Y|Z,\theta)\right) - \ln P(Y|\theta^{(i)}) \\ =& \ln\left(\sum_Z P(Z|Y,\theta^{(i)})\frac{P(Z|\theta)P(Y|Z,\theta)}{P(Z|Y,\theta^{(i)})}\right) - \ln P(Y|\theta^{(i)}) \end{aligned}$$

利用Jesson不等式，$\ln\left(\sum_Z P(Z|Y,\theta^{(i)})\frac{P(Z|\theta)P(Y|Z,\theta)}{P(Z|Y,\theta^{(i)})}\right) \geqslant \sum_Z P(Z|Y,\theta^{(i)}) \ln \frac{P(Z|\theta)P(Y|Z,\theta)}{P(Z|Y,\theta^{(i)})}$

$$\because \ln P(Y|\theta^{(i)}) = \sum_Z P(Z|Y,\theta^{(i)}) \ln P(Y|\theta^{(i)})$$

$$\begin{aligned} \therefore L(\theta) - L(\theta^{(i)}) &\geqslant \sum_Z P(Z|Y,\theta^{(i)}) \ln \frac{P(Z|\theta)P(Y|Z,\theta)}{P(Z|Y,\theta^{(i)})} - \sum_Z P(Z|Y,\theta^{(i)}) \ln P(Y|\theta^{(i)}) \\ &= \sum_Z P(Z|Y,\theta^{(i)}) \ln \frac{P(Z|\theta)P(Y|Z,\theta)}{P(Z|Y,\theta^{(i)})P(Y|\theta^{(i)})} \end{aligned}$$

令$B(\theta,\theta^{(i)}) = L(\theta^{(i)}) + \sum_Z P(Z|Y,\theta^{(i)}) \ln \frac{P(Z|\theta)P(Y|Z,\theta)}{P(Z|Y,\theta^{(i)})P(Y|\theta^{(i)})}$

$\therefore L(\theta) \geqslant B(\theta,\theta^{(i)})$，也就是说$B(\theta,\theta^{(i)})$是$L(\theta)$的一个下界，要最大化$L(\theta)$，可换成最大化$B(\theta,\theta^{(i)})$，这个和之前的改进的迭代尺度法的思路是一致的。

$$\begin{aligned} \therefore \theta^{(i+1)} =& \arg\max_\theta B(\theta,\theta^{(i)}) \\ =& \arg\max_\theta\left(\sum_Z P(Z|Y,\theta^{(i)}) \ln P(Z|\theta)P(Y|Z,\theta)\right) \\ =& \arg\max_\theta\left(\sum_Z P(Z|Y,\theta^{(i)}) \ln P(Y,Z|\theta)\right) \end{aligned}$$

$$\because Q(\theta,\theta^{(i)}) = \sum_Z \ln P(Y,Z|\theta)P(Z|Y,\theta^{(i)})$$

$$\therefore \theta^{(i+1)} = \arg\max_\theta Q(\theta,\theta^{(i)})$$

等价于EM算法的M步，E步等价于求$\sum_Z P(Z|Y,\theta^{(i)}) \ln P(Y,Z|\theta)$，以上就得到了EM算法，通过不断求解下界的极大化逼近求解对数似然函数极大化。

24 第9章-EM算法及推广-高斯混合模型

本节介绍如何用EM算法求解高斯混合模型中的参数，高斯混合模型是一个非常重要的模型，从理论上讲，可以用高斯混合模型逼近任何一个连续型的分布，这就类似于可以通过泰勒公式（泰勒展开）用一个多项式拟合任何一个函数。

24.1 明确隐变量，写出完全数据的对数似然函数

高斯混合模型的概率分布模型为$P(y|\theta)=\sum_{k=1}^{K}\alpha_k\phi(y|\theta_k)$，其中$\phi(y|\theta_k)$是高斯分布密度函数，是由$\theta_k$决定的，$\alpha_k$表示$y$以$\alpha_k$的概率来自于第$k$个正态分布$\phi(y|\theta_k)$，$\theta=(\alpha_1,\cdots,\alpha_K,\mu_1,\sigma_1^2,\mu_2,\sigma_2^2,\cdots,\mu_K,\sigma_K^2)$，也就是说，要通过观测数据$y$估计$\theta$中的值。

根据EM算法，存在一个隐变量γ，γ表示当前的y来自的高斯分布，对于第一个观测值，有$\gamma_1=(\gamma_{11},\gamma_{12},\cdots,\gamma_{1K})$，其中根据书中的$\gamma_{jk}$的定义：

$$\gamma_{jk}=\begin{cases}1, & \text{第}j\text{个观测来自第}k\text{个分模型}\\ 0, & \text{否则}\end{cases}$$

$$j=1,2,\cdots,N;k=1,2,\cdots,K$$

以上是随机变量的分布，取第1个值的概率为α_1，取第2个值的概率为α_2，……，取第K个值的概率为α_K，一旦知道γ_1的值，就知道从第几个高斯分布中抽取y_1。

$$\begin{aligned}p(\gamma_1,y_1|\theta)&=p(\gamma_1|\theta)\cdot p(y_1|\gamma_1,\theta)\\&=\alpha_1^{\gamma_{11}}\cdot\alpha_2^{\gamma_{12}}\cdots\alpha_K^{\gamma_{1K}}\phi(y_1|\theta_1)^{\gamma_{11}}\phi(y_2|\theta_2)^{\gamma_{12}}\cdots\phi(y_1|\theta_K)^{\gamma_{1K}}\\&=\prod_{k=1}^{K}[\alpha_k\phi(y_1|\theta_k)]^{\gamma_{1k}}\end{aligned}$$

这个是第1个样本点完全数据的密度函数。在极大化似然估计中是极大化似然函数，这需要所有样本点的联合分布，对于所有的样本点，概率密度函数为

$$P(y,\gamma|\theta)=\prod_{j=1}^{N}\prod_{k=1}^{K}\left[\alpha_k\phi(y_j|\theta_k)\right]^{\gamma_{jk}}$$

$\because\prod_{j=1}^{N}\prod_{k=1}^{K}\alpha_k^{\gamma_{jk}}=\prod_{k=1}^{K}\alpha_k^{\sum_{j=1}^{N}\gamma_{jk}}$，$\sum_{j=1}^{N}\gamma_{jk}$表示在N个样本点中，一共有多少个是来自第$k$个高斯分布的，将该数量记为$n_k=\sum_{j=1}^{N}\gamma_{jk}$，$n_1+n_2+\cdots+n_K=N$

$$\therefore\prod_{k=1}^{K}\alpha_k^{\sum_{j=1}^{N}\gamma_{jk}}=\prod_{k=1}^{K}\alpha_k^{n_k}$$

$$\therefore\prod_{j=1}^{N}\prod_{k=1}^{K}\alpha_k^{\gamma_{jk}}=\prod_{k=1}^{K}\alpha_k^{n_k}$$

$$\therefore P(y,\gamma|\theta)=\prod_{k=1}^{K}\alpha_k^{n_k}\prod_{j=1}^{N}\left[\phi(y_i|\theta_k)\right]^{\gamma_{jk}}=\prod_{k=1}^{K}\alpha_k^{n_k}\cdot\prod_{j=1}^{N}\left[\frac{1}{\sqrt{2\pi}\sigma_k}\exp\left(-\frac{(y_j-\mu_k)^2}{2\sigma_k^2}\right)\right]^{\gamma_{jk}}$$

$$\therefore\ln P(y,\gamma|\theta)=\sum_{k=1}^{K}\left\{n_k\ln\alpha_k+\sum_{j=1}^{N}\gamma_{jk}\left[\ln(\frac{1}{\sqrt{2\pi}})-\ln\sigma_k-\frac{1}{2\sigma_k^2}(y_j-\mu_k)^2\right]\right\}$$

24.2 EM算法的E步，确定Q函数

将隐变量都换成期望，隐变量有γ_{jk}和n_k

$\because E(n_k)=E\left(\sum_j\gamma_{jk}\right)=\sum_j E(\gamma_{jk})$，$E(\gamma_{jk}|\theta^{(i)},y)=P(\gamma_{jk}=1|\theta^{(i)},y)$，求解期望时，是根据上一步的$\theta^{(i)}$以及观测数据所有的$y_j$，需要知道$\gamma_{jk}$的分布$P(\gamma_{jk}=1|\theta^{(i)},y)$。

$$\because P(\gamma_{jk}=1|\theta^{(i)},y)=\frac{P(\gamma_{jk}=1,y_j|\theta^{(i)})}{P(y_j|\theta^{(i)})}$$
$$=\frac{P(\gamma_{jk}=1,y_j|\theta^{(i)})}{\sum_{k=1}^{K}P(\gamma_{jk}=1,y_j|\theta^{(i)})}$$
$$=\frac{P(\gamma_{jk}=1|\theta^{(i)})P(y_i|\gamma_{jk}=1,\theta^{(i)})}{\sum_{k=1}^{K}P(y_j|\gamma_{jk}=1,\theta^{(i)})P(\gamma_{jk}=1|\theta^{(i)})}$$

$\because \alpha_k=P(\gamma_{jk}=1|\theta),\phi(y_i|\theta)=P(y_i|\gamma_{jk}=1,\theta)$

$\therefore E(\gamma_{jk}|y,\theta^{(i)})=P(\gamma_{jk}=1|\theta^{(i)},y)=\frac{\alpha_k\phi(y_i|\theta^{(i)})}{\sum_{k=1}^{K}\alpha_k\phi(y_i|\theta^{(i)})}$ ，其中$\theta^{(i)}=(\alpha_k^{(i)},\theta_k^{(i)})$

将γ_{jk}关于给定y和$\theta^{(i)}$的条件下的期望记为$Z_k=E(\gamma_{jk}|y,\theta^{(i)})$，因为各个样本之间是独立同分布的，所以$Z_k$是和$j$无关的。

$$\therefore Q(\theta,\theta^{(i)})=E_Z\left[lnP(y,\gamma|\theta^{(i)})\right]=\sum_{k=1}^{K}\left\{(NZ_k)\ln\alpha_k+Z_k\sum_{j=1}^{N}\left[\ln(\frac{1}{\sqrt{2\pi}})-\ln\sigma_k-\frac{1}{2\sigma_k^2}(y_j-\mu_k)^2\right]\right.$$

24.3 确定EM算法的M步

需要估计的变量有α_k,σ_k,μ_k，然后求偏导等于0：

$$\frac{\partial Q(\theta,\theta^{(i)})}{\partial\mu_k}=0$$
$$\frac{\partial Q(\theta,\theta^{(i)})}{\partial\sigma_k^2}=0$$
$$\begin{cases}\dfrac{\partial Q(\theta,\theta^{(i)})}{\partial\alpha_k}=0\\ \alpha_k=1\end{cases}$$

根据上述公式可以推导出：

$$\mu_k^{(i+1)}=\frac{\sum_{j=1}^{N}\hat{\gamma_{jk}}y_j}{\sum_{j=1}^{N}\hat{\gamma_{jk}}}$$

$$(\sigma_k^2)^{(i+1)}=\frac{\sum_{j=1}^{N}\hat{\gamma_{jk}}(y_i-\mu_k)^2}{\sum_{j=1}^{N}\hat{\gamma_{jk}}}$$

$$\alpha_k^{(i+1)}=\frac{n_k}{N}=\frac{\sum_{j=1}^{N}\hat{\gamma_{jk}}}{N}$$

其中$\hat{\gamma_{jk}} = E\gamma_{jk}, n_k = \sum_{j=1}^{N} E\gamma_{jk}, k = 1, 2, \cdots, K$

25 第10章-隐马尔科夫模型(HMM)-导读

隐马尔可夫模型(HMM)和条件随机场(CRF)都属于概率模型，这两个模型涉及到的随机变量个数都非常多，为了描述随机变量之间的关系（是不是独立的），需要借助一个很有用的工具——概率图模型，概率图模型分为有向图和无向图，隐马尔可夫模型用的就是有向图。

25.1 概率图模型

在有向图中，用∘（圆圈）表示随机变量，可以是一维的，也可以是多维的，既可以是离散随机变量，也可以是连续的，∘叫做结点，图是由结点和边构成的，在有向图中就是有向边，要描述Y受X影响的，就将X和Y连接起来，并用箭头描述从X指向Y的方向。

第9章EM算法中的掷硬币的例子，一共有3个硬币，观察到的只有两个结果，首先掷第1个硬币，以π的概率正面向上，如果第1个硬币正面向上，就去掷第2个硬币，第2个硬币以p的概率出现正面向上，如果第1个硬币反面向上，在第2次的时候，就去掷第3个硬币，第3个硬币以q的概率出现正面向上。

对应到有向图中，X就是第1次掷硬币的结果$\{0, 1\}$，第2次掷硬币的结果也是$\{0, 1\}$，第2次掷硬币概率的参数是依赖第1次掷硬币结果的，第1个随机变量X会对Y产生影响，就可以用概率图来表示。

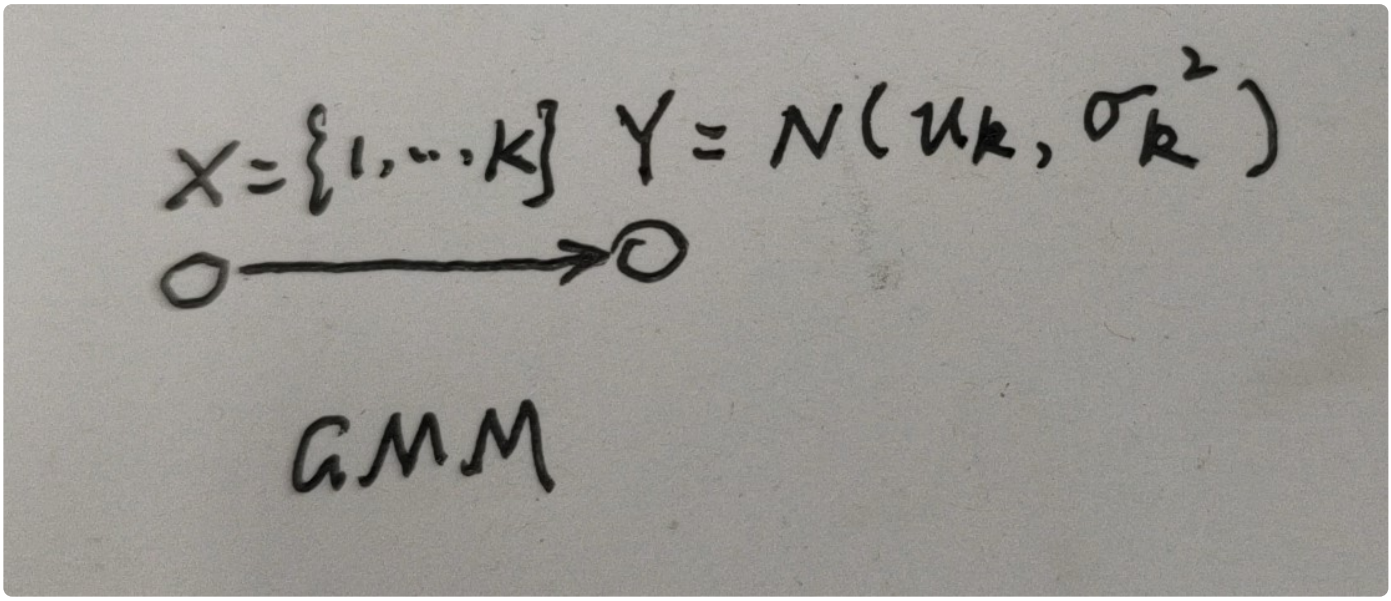

图10-1 高斯混合模型的概率图

对应到EM算法的混合高斯模型，第1个随机变量X是一个离散的分布，它决定了是从第几个高斯模型中抽取变量，在GMM算法中，有$\{1, 2, \cdots, K\}$一共K个高斯分布。随机变量Y也是一个高斯分布$N(\mu, \sigma^2)$，高斯分布的参数取决于第1次随机变量X的取值。以上就是用两个结点和一条边表示高斯混合模型。

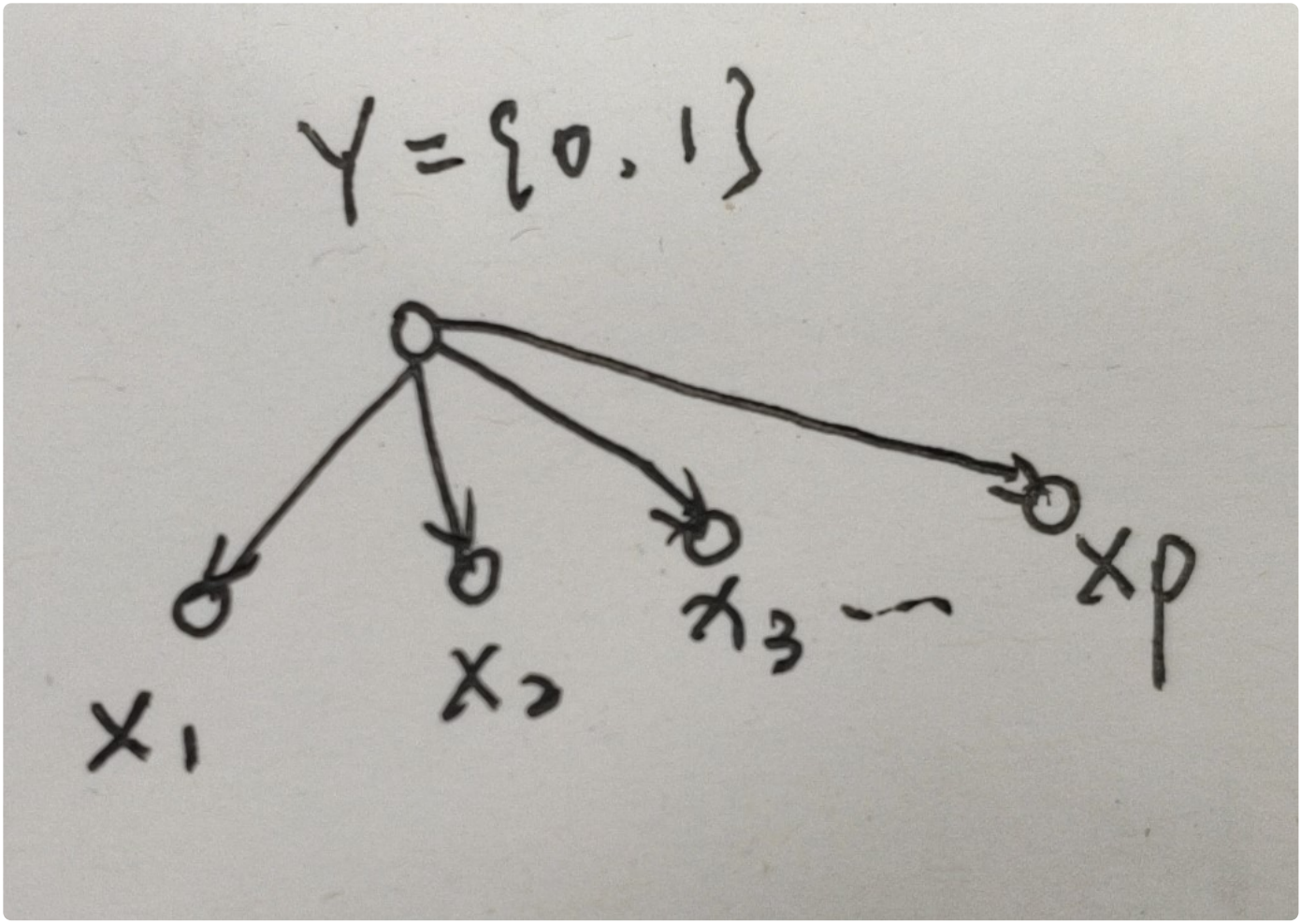

图10-2 朴素贝叶斯模型的概率图

朴素贝叶斯模型的概率图如图10-2所示，X的分布是依赖于类别的，不同的类别，X的分布是不同的，在朴素贝叶斯中，当给定Y的值，X之间是相互独立的，所以只有Y到X的箭头，没有X之间的箭头。

一个箭头可以表示两个随机变量之间的关系，引入条件独立的概念，在概率图模型中，假设有三个随机变量X, Y, Z，之前讲的EM算法是含有隐变量和观测变量的，一般来说，隐变量在图模型中用圆◦表示，如果能观察到一个变量取值的时候，用带阴影的圆•表示。在掷硬币的例子中，第1个结果是观察不到的，用空心圆◦表示，第2个结果是可以观察到的，用带阴影的圆•表示。为什么要强调隐变量和观测变量，圆是空心还是阴影会影响到随机变量的依赖性。

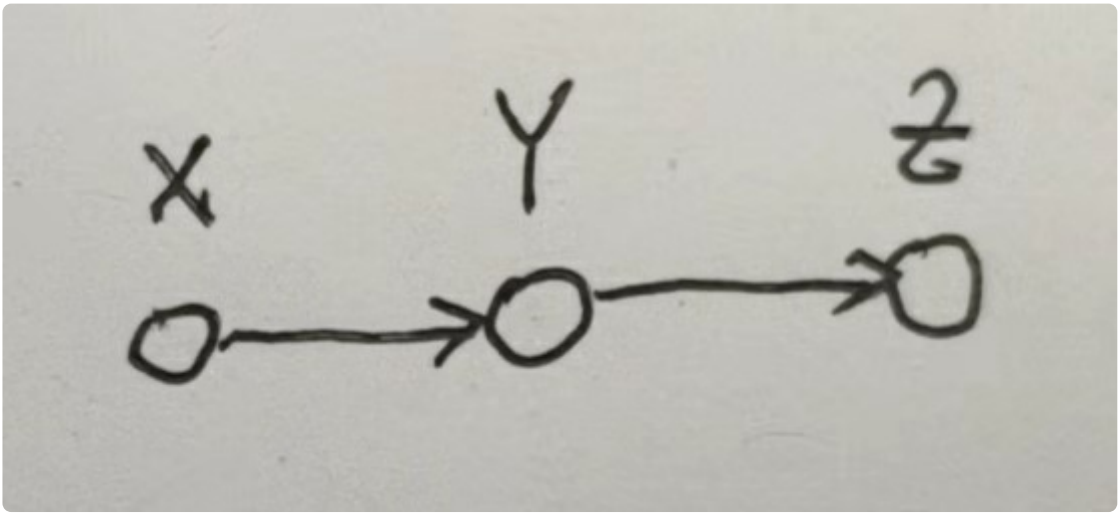

图10-3 随机变量之间的关系(1)

图10-3中，随机变量都是空心圆，这三个随机变量都是观测不到的，可得$P(X, Z) \neq P(X)(Z)$

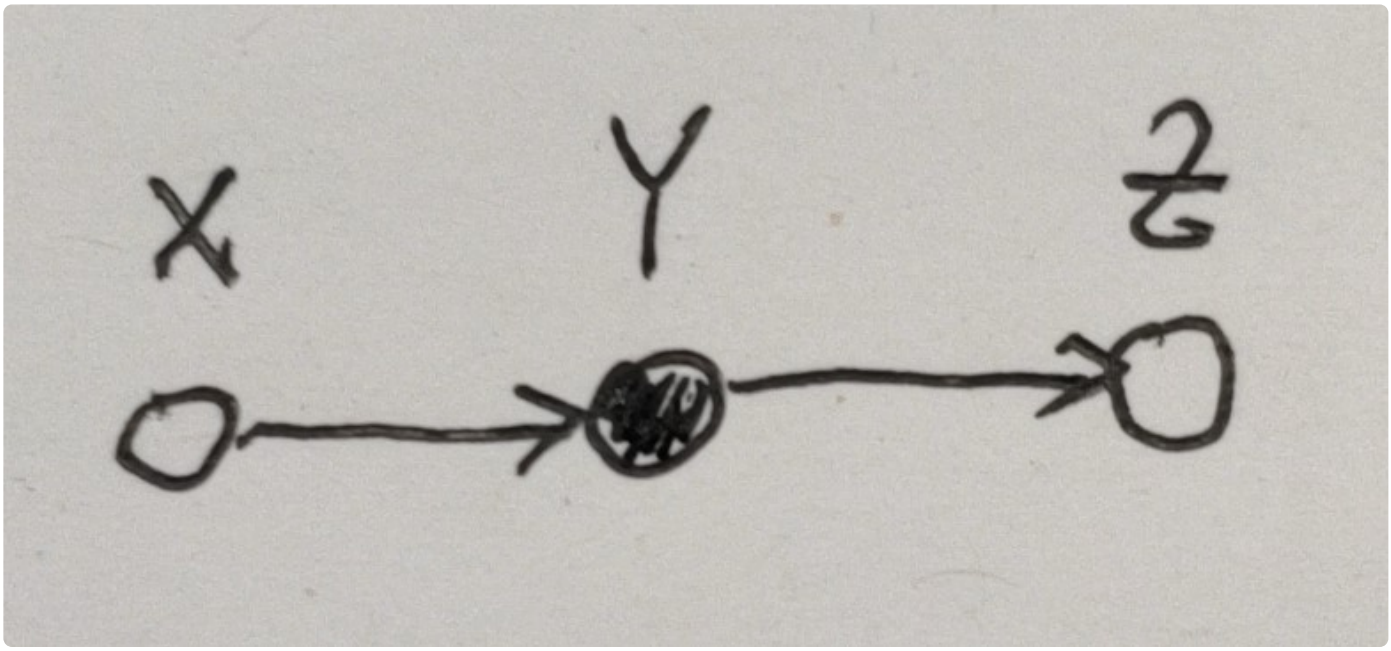

图10-4 随机变量之间的关系(2)

图10-4中，Y是带阴影的圆，随机变量Y是可以观察到的，可得$P(X, Z|Y) = P(X|Y)P(Z|Y)$，从箭头的指向看，信息是从X传到Y，Y传到Z，一旦将Y固定了，信息的流通相当于被Y观察到的值堵住了，所以当观察到Y时，X和Z就是独立的。

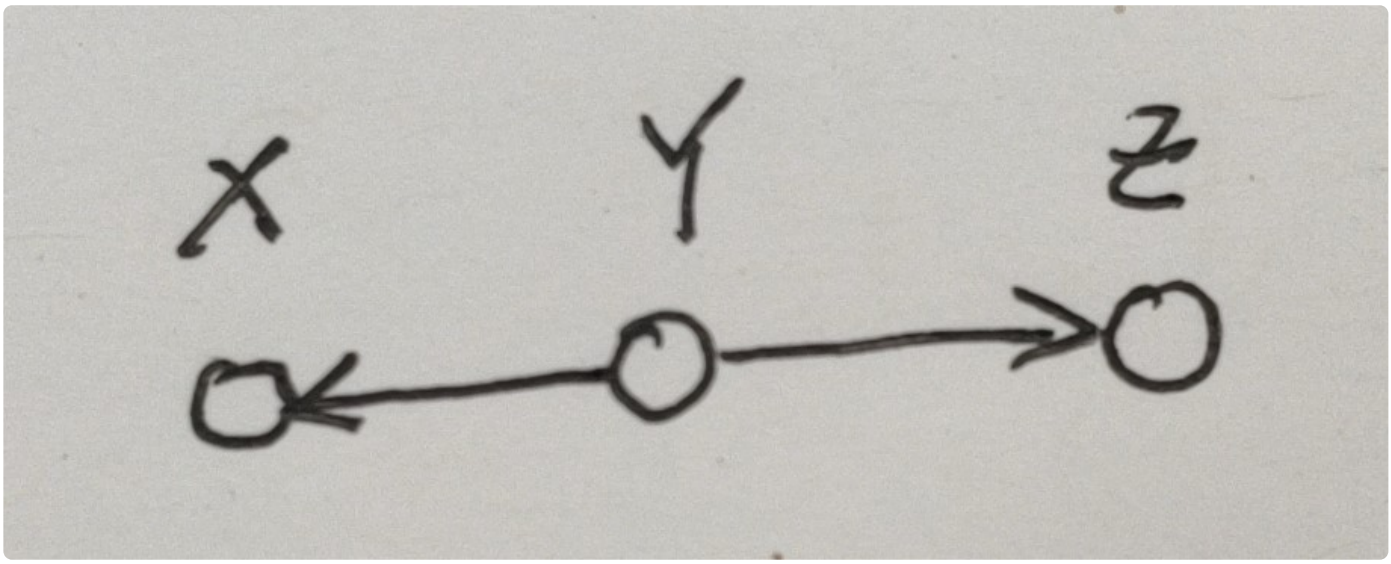

图10-5 随机变量之间的关系(3)

图10-5中，Y指向了两边，这个时候单看X和Z是不独立的，满足$P(X, Z) \neq P(X)P(Z)$，如果给定Y，X和Z是独立的，满足$P(X, Z|Y) = P(X|Y)P(Z|Y)$

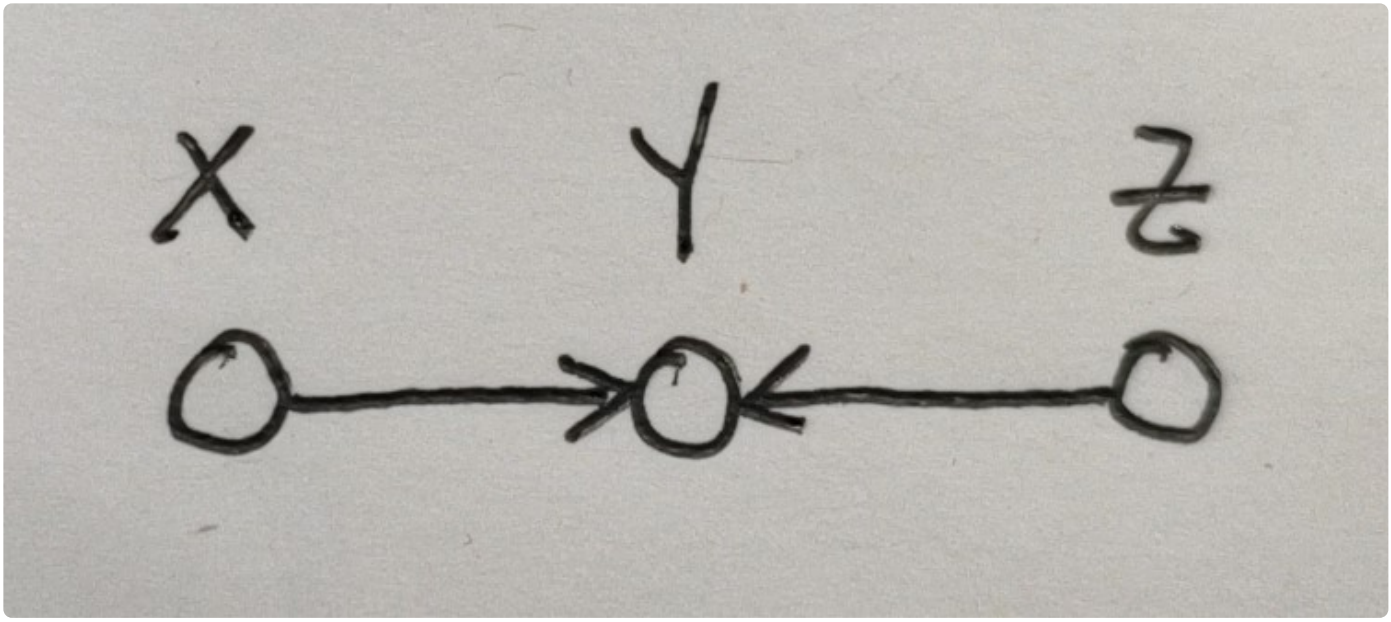

图10-6 随机变量之间的关系(4)

图10-6中，随机变量满足$P(X, Z) = P(X)P(Z), P(X, Z|Y) \neq P(X|Y)P(Z|Y)$

25.2 隐马尔可夫模型的基本概念

> 变量多，用概率图模型（有向图）表示变量间的关系。
> 模型参数及符号：状态集合，状态转移概率矩阵，观测集合，观测概率矩阵 ，初始状态概率向量

25.2.1 隐马尔可夫模型的定义

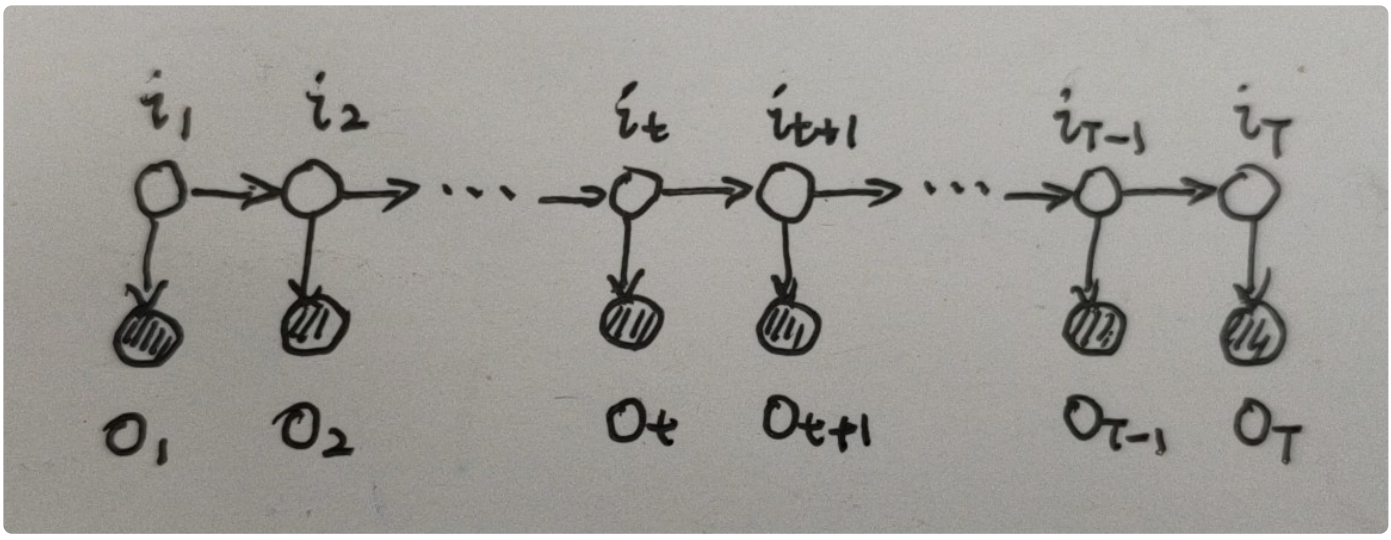

图10-7 隐马尔可夫概率图模型

图10-7是隐马尔可夫模型的概率图模型，如果单看上面一行，是马尔可夫链，因为上面一行都是空心圆（观察不到的隐变量），故称为隐马尔可夫模型。上面这一行观察不到的变量集合称为**状态序列**，下面一行观测到的变量集合称为**观测序列**。状态序列记为$(i_1, i_2, \cdots, i_t, i_{t+1}, i_{T-1}, i_T)$，观测序列记为
$(o_1, o_2, \cdots, o_t, o_{t+1}, \cdots, o_{T-1}, o_T)$

隐马尔可夫模型的特点，已知第1个状态变量i_1，i_1会影响第2个状态变量i_2，同时也会影响观测变量o_1，当得到第2个状态变量i_2，会影响状态变量i_3，也会影响第2个观测变量o_2，依次递推。隐马尔可夫模型从状态变量

看是一个影响一个的，可以理解成时间序列模型，其适用范围为文本分析，比如标注问题。

已知$i_t \in \{q_1, q_2, \cdots, q_N\}, o_t \in \{v_1, v_2, \cdots, v_M\}$，其中N是可能的状态数，M是可能的观测数。状态序列为$I = (i_1, i_2, \cdots, i_T)$，观测序列为$O = (o_1, o_2, \cdots, o_T)$，现考虑状态与观测之间、状态与状态之间的关系。

25.2.1.1 状态与状态之间的关系

$$\begin{array}{cccccc} & i_2 = q_1 & i_2 = q_2 & \cdots & i_2 = q_N \\ i_1 = q_1 & a_{11} & a_{12} & \cdots & a_{1N} \\ i_1 = q_2 & a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \vdots & \cdots & vdots \\ i_1 = q_N & a_{N1} & a_{N2} & \cdots & a_{NN} \end{array}$$

所以$a_{ij} = P(i_2 = q_j | i_1 = q_i)$，$\sum_{i=1}^{N} a_{1i} = 1$

A为状态转移概率矩阵：

$$A_{N\times N} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ & & \vdots & \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix}$$

这里给出的是i_1与i_2之间的关系，在马尔可夫模型假设中，这个关系不随时间t而变化，也就是说i_1和i_2之间的关系可以用这个矩阵表示，i_2和i_3之前的关系也可以用这个矩阵表示。

25.2.1.2 状态与观测之间的关系

$$\begin{array}{cccccc} & o_1 = v_1 & o_1 = v_2 & \cdots & o_1 = v_M \\ i_1 = q_1 & b_1(1) & b_1(2) & \cdots & b_1(M) \\ i_1 = q_2 & b_2(1) & b_2(2) & \cdots & b_2(M) \\ \vdots & \vdots & \vdots & \cdots & vdots \\ i_1 = q_N & b_N(1) & b_N(2) & \cdots & b_N(M) \end{array}$$

，其中$\sum_{i=1}^{M} b_1(i) = 1$

B为观测概率矩阵：

$$B_{N\times M} = \begin{bmatrix} b_1(1) & b_1(2) & \cdots & b_1(M) \\ b_2(1) & b_2(2) & \cdots & b_2(M) \\ & & \vdots & \\ b_N(1) & b_N(2) & \cdots & b_N(M) \end{bmatrix}$$

25.2.1.3 初始状态概率

对于整个模型的参数，就差第1部分——初始状态概率，已知$i_1 \in \{q_1, q_2, \cdots, q_N\}$将每一个的概率都用$\pi_n$表示：

$$\begin{aligned} \pi_1 &= P(i_1 = q_1) \\ \pi_2 &= P(i_1 = q_2) \\ &\vdots \\ \pi_N &= P(i_1 = q_N) \end{aligned}$$

最后，令$\pi = (\pi_1, \pi_2, \cdots, \pi_N)$

在整个隐马尔可夫模型中，参数由初始状态概率向量π，状态转移概率矩阵A和观测概率矩阵B组成，即$\lambda = (\pi, A, B)$，其中π中的参数有N个，A中的参数有$N \times N$个，B中的参数有$N \times M$个，其中自由参数有多少个？π中的自由参数为$N - 1$个（N个参数，1个约束条件），A中的自由参数为$N \times N - N$个，B中的自由参数为$N \times M - M$个。

25.2.2 两个基本假设

1. 齐次马尔可夫性假设：指的是隐变量之间的关系，在任意时刻t，隐变量的取值只与前一时刻状态的取值有关，与其他时刻的状态无关，用公式表示为
$$P(i_t|i_{t-1}, \cdots, i_1) = P(i_t|i_{t-1})$$
2. 观测独立性假设：指的是隐变量和观测变量之间的关系，在任意时刻的观测只依赖于该时刻的马尔可夫链的状态，与其他观测及状态无关，用公式表示为
$$P(o_t|i_T, o_T, i_{T-1}, o_{T-1}, \cdots, i_{t+1}, o_{t+1}, i_t, o_t, \cdots, i_1, o_1) = P(o_t|i_t)$$

这两个基本假设在实际情况中并不是很合理，为什么要做这样的假设呢？其实和朴素贝叶斯是类似的，简化模型中变量的关系。

25.2.3 三个基本问题

1. 概率计算问题：给定模型$\lambda = (A, B, \pi)$和观测序列$O = (o_1, o_2, \cdots, o_T)$，计算在模型$\lambda$下观测序列$O$出现的概率$P(O|\lambda)$
2. 学习问题：已知观测序列$O = (o_1, o_2, \cdots, o_T)$，用极大似然估计的方法估计参数，即估计模型$\lambda = (A, B, \pi)$参数，使得在该模型下观测序列概率$P(O|\lambda)$最大
3. 预测问题：已知观测序列$O = (o_1, o_2, \cdots, o_T)$，求给定观测序列条件概率$P(I|O)$最大的状态序列$I = (i_1, i_2, \cdots, i_T)$，即给定观测序列，求最有可能的对应的状态序列

25.3 概率计算算法

给定模型$\lambda = (A, B, \pi)$和观测序列$O = (o_1, o_2, \cdots, o_T)$，计算在模型$\lambda$下观测序列$O$出现的概率$P(O|\lambda)$，书中一共提到了两个方法，实际上是三种：直接计算法、前向算法、后向算法。

25.3.1 直接计算法

$$\begin{aligned} P(O|\lambda) &= \sum_I P(O|I, \lambda)P(I|\lambda) \\ &= \sum_{i_1, i_2, \cdots, i_T} \pi_{i_1} b_{i_1}(o_1) a_{i_1 i_2} b_{i_2}(o_2) \cdots a_{i_{T-1} i_T} b_{i_T}(o_T) \end{aligned}$$

首先观察求和符号中b有T项，a有$T - 1$项，π有1项，一共有$2T$项，求和是从$i_1, i_2, \cdots, i_T$不同的取值求和，对于i_1取值有N个，i_2取值有N个……i_T取值有N个，所以求和范围有N^T项，要对N^T个范围进行求和，每一个求和范围都需要$O(T)$个计算量，所以计算复杂度为$2T \times N^T = 2TN^T = O(TN^T)$。

25.3.2 前向算法

算法10.2（观测序列概率的前向算法）输入：隐马尔可夫模型λ，观测序列O
输出：观测序列概率$P(O|\lambda)$
(1)初值：$\alpha_1(i) = \pi_i b_i(o_1), \quad i = 1, 2, \cdots, N$
(2)递推，对$t = 1, 2, \cdots, T-1$，

$$\alpha_{t+1}(i) = \left[\sum_{j=1}^{N} \alpha_t(j) a_{ji}\right] b_i(o_t + 1), \quad i = 1, 2, \cdots, N$$

(3)终止

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

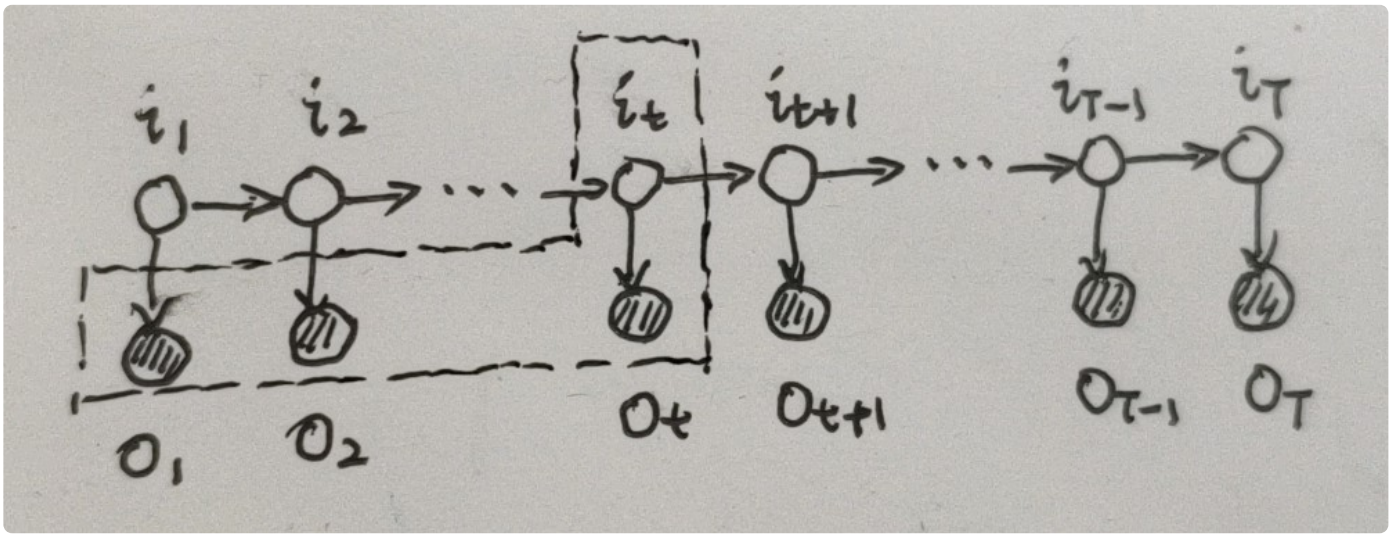

图10-8 前向算法概率图

首先定义了$\alpha_t(i) = P(o_1, o_2, \cdots, o_t, i_t = q_i|\lambda), i = 1, 2, \cdots, N$，如图10-8中表示的就是虚线框中的联合概率分布， 然后通过递推的方式，一步一步得到$\alpha_T(i)$，递推关系为$\alpha_{t+1}(i) = \left[\sum_{j=1}^{N} \alpha_t(j) a_{ji}\right] b_i(o_t + 1)$，当$t = 1$时，可以算出$\alpha_1(1), \alpha_1(2), \cdots, \alpha_1(i), \cdots, \alpha_1(N)$，当$t = 2$时，根据递推可以计算出$\alpha_2(1), \alpha_2(2), \cdots, \alpha_2(i), \cdots, \alpha_2(N)$，同理可以一直递推到时刻$t = T$时，$\alpha_T(1), \alpha_T(2), \cdots, \alpha_T(i), \cdots, \alpha_T(N)$，最后计算$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$。

首先每一时刻都要计算一列的α，一共有T个时刻，当计算每一列的时候，计算每一个α_i都用了前一个时刻N个不同的值，一共计算了N项，也就是t时刻计算了N项，每一个时刻都用了前一个时刻N个值，计算量为N^2，又有T个时刻，整个计算复杂度为TN^2

25.3.3 后向算法

算法10.3（观测序列概率的后向算法）输入：隐马尔可夫模型λ，观测序列O
输出：观测序列概率$P(O|\lambda)$
(1)$\beta_T(i)=1, \quad i=1,2,\cdots,N$
(2)对$t=T-1,T-2,\cdots,1$，

$$\beta_t(i)=\sum_{j=1}^{N}a_{ij}b_j(o_{t+1})\beta_{t+1}(j), \quad i=1,2,\cdots,N$$

(3)

$$P(O|\lambda)=\sum_{i=1}^{N}\pi_i b_i(o_1)\beta_1(i)$$

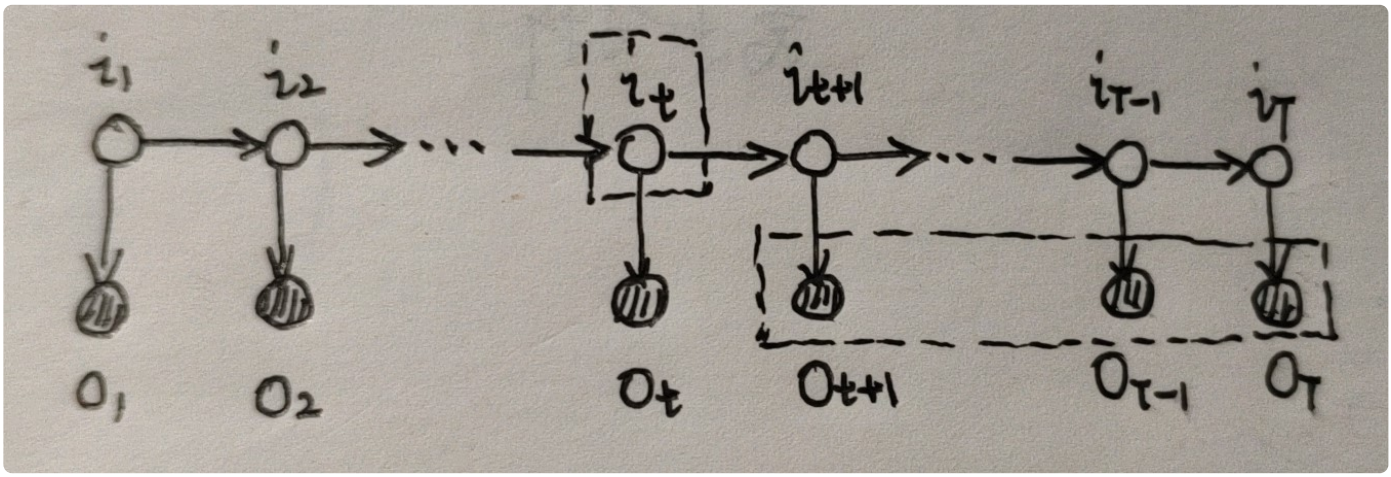

图10-9 后向算法概率图

首先定义$\beta_t(i)=P(o_{t+1},o_{t+2},\cdots,o_T|i_t=q_i,\lambda)$，如图10-9中表示的就是虚线框中的联合概率分布，对于每一个时刻t，β一共有N个值，状态i_t的取值有N个，在后向算法中，首先计算最后一个时刻$t=T$时$(\beta_T(1),\beta_T(2),\cdots,\beta_T(N))$的取值，然后根据后面$N$个值的前一个时刻的值，以此类推，一共有$N$项求和，最后当给定$\lambda$时，计算$P(O|\lambda)$用$\beta_1(i)$表示，此计算的复杂度和前向算法是一样的，计算复杂度为TN^2

25.4 学习算法

本节和10.2节的过程是相反的，已知观测序列$O=(o_1,o_2,\cdots,o_T)$，估计参数λ。和之前讲过的很多模型是类似的，需要用一组训练数据集估计模型，之前可能是估计分类超平面，书中一共提到了两个方法：监督学习方法、Baum-Welch算法，这两种方法的区别在于哪个算法更贴近应用，这两个方法的学习设定是不一样的。

在隐马尔可夫模型中，能观察到的只是观测序列O，但是在监督学习方法中，人为地对观测值进行了标注I，所以状态序列I在该方法中也是已知的，这样不是很贴近现实情况。

25.4.1 监督学习方法

假设已给训练数据包含S个长度相同的观测序列和对应的状态序列$\{(O_1, I_1), (O_2, I_2), \cdots, (O_S, I_S)\}$，那么可以利用极大似然估计法来估计隐马尔可夫模型的参数。具体方法如下：

(1)转移概率a_{ij}的估计

设样本中时刻t处于状态i时刻$t+1$转移到状态j的频数为A_{ij}，那么状态转移概率a_{ij}的估计是

$$\hat{a}_{ij} = \frac{A_{ij}}{\sum_{j=1}^{N} A_{ij}}, \quad i = 1, 2, \cdots, N; j = 1, 2, \cdots, N$$

(2)观测概率$b_j(k)$的估计

设样本中状态为j并观测为k的频数是B_{jk}，那么状态为j观测为k的概率$b_j(k)$的估计是

$$\hat{b}_j(k) = \frac{B_{jk}}{\sum_{k=1}^{M} B_{jk}}, \quad j = 1, 2, \cdots, N; k = 1, 2, \cdots, M$$

(3)初始状态概率π_i的估计$\hat{\pi}_i$为S个样本中初始状态为q_i的频率

观测序列和状态序列都有S个样本，需要估计的参数：

1. 首先初始状态π，如果直接用极大似然估计，就是看这S个样本中在初始状态时的各个取值的频率。
2. 概率转移矩阵A，前一个状态$i_t = q_i$，后一个状态$i_{t+1} = q_j$的概率记作a_{ij}，考察在S个观测序列中，每个观测序列都是有$(i_1, i_2, \cdots, i_{T-1}, i_T)$，一共有$S$个这样的马尔可夫链，总计$S(T-1)$组，在这些序列组中有多少是观测序列为$q_i$，然后在这些组中，又有多少组后一项的状态取值为$q_j$，这样的比例就是$a_{ij}$的估计。
3. 观测概率$b_j(k)$的估计也是类似的。

25.4.2 Baum-Welch算法（EM算法）

监督学习方法中，需要人为地标注每个观测值状态的取值，这样人工的工作量就会非常大，当不进行标注时，相当于模型中有T个隐变量和T个观测变量，当一个模型中既含有隐变量又含有观测变量时，估计模型的参数，很自然地想到用EM算法处理隐变量。Baum-Welch算法本质上就是EM算法，该算法的提出在EM算法之前，属于EM算法的特例。

EM算法已经在第9章介绍过了，需要求解完全数据的联合概率分步$P(O, I|\lambda)$，然后极大化联合概率函数，为了计算简单取对数，$\max \ln P(O, I|\lambda)$，使用迭代的方法更新参数$\lambda = (\pi, A, B)$，首先初始化π, A, B，**E步：** 由于隐变量i的值未知，用给定的初值，在$\ln P(O, I|\lambda)$中所有包含隐变量的项用期望代替，**M步：** 最大化已经替换完的公式，利用拉格朗日乘子法，求解最优解。然后反复迭代，直到估计值收敛。算法10.4给出了EM算法在隐马尔可夫模型中的参数更新的递推式。

算法10.4（Baum-Welch算法）
输入：观测数据$O=(o_1,o_2,\cdots,o_T)$
输出：隐马尔可夫模型
(1)初始化
对$n=0$，选取$a_{ij}^{(0)},b_j(k)^{(0)},\pi_i^{(0)}$，得到模型$\lambda^{(0)}=(A^{(0)},B^{(0)},\pi^{(0)})$
(2)递推，对n=1,2,\cdots,

$$a_{ij}^{(n+1)}=\frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_i(i)}$$

$$b_j(k)^{(n+1)}=\frac{\sum_{t=1,o_t=v_k}^{T}\gamma_t(j)}{\sum_{t=1}^{T}\gamma_t(j)}$$

$$\pi_i^{(n+1)}=\gamma_1(i)$$

其中

$$\gamma_t(i)=\frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)}=\frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N}\alpha_t(j)\beta_t(j)}$$

$$\xi_t(i,j)=\frac{\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}$$

25.5 预测算法

该算法其实就是一个标注问题，已知观测序列和模型参数λ，让计算机标注每一个观测值的状态，书中提出了两个算法：近似算法（得到的结果并不是全局最优解）、维特比算法（用动态规划思想求最优路径）。

25.5.1 近似算法

对每一个观测值进行标注并求解得出状态，实际上要求解的是使得$P(I|O,\lambda)$概率最大的一组状态。
近似算法重新定义了一个函数$\gamma_t(i)$，表示在时刻t处于状态q_i的概率：

$$\gamma_t(i)=P(i_t=q_i|O,\lambda)$$

近似算法的思路就是对于每一个时刻t都进行求解，当前这个时刻出现概率最大的状态$i_t^*=\arg\max_{1\leqslant i\leqslant N}[\gamma_t(i)],t=1,2,\cdots,T$，从而得到状态序列为$I^*=(i_1^*,i_2^*,\cdots,i_T^*)$
这个算法的缺点很明显，求解每一个时刻t的最优状态时，都没有考虑该时刻和前一个时刻下状态之间的联系，有可能 会出现一种情况：相邻的两个状态之间的状态转移概率为0，这种情况说明得到的不是全局的最优解，但这种方法计算比较简单。

25.5.2 维特比算法

当求解最优路径时，可以采用维特比算法。以下用时刻t_1,t_2,t_3阐述这个算法的思想：在时刻t_1时，对应的状态变量为i_1，i_1可能的取值$q_1,q_2,\cdots,q_N$有N个，在时刻t_2时，对应的状态变量为i_2，i_2可能的取值有N个，在时刻t_3时，对应的状态变量为i_3，i_3可能的取值有N个。

首先观察时刻$t=2$时，假设$i_2=q_1$，考察从前一时刻t_1到达q_1的路径的概率，从中选取概率最大的那个路径，假设是从$i_1=q_2$到$i_2=q_1$概率最大，就保留该路径，继续求解$i_2=q_2$时的t_1时刻到达q_2的概率最大的路径，假设是从$i_1=q_N$到$i_2=q_2$概率最大，就保留该路径，依次求取时刻$t=2$概率最大的最优路径。

考察完时刻$t=2$，再考察时刻$t=3$，求解当i_3取各个状态时的最优路径，此时不需要再看$t=2$之前的最优路径了，因为已经求解过了，只需要考察$t=3$的前一时刻的最优路径（即从$t=2$到$t=3$在各个状态的最优路径），已经记录了每一个状态之前的路径以及相应的概率值，用$\delta(i)$表示，$i=1,2,\cdots,N$。

所以当计算$i_3=q_1$的最优路径时，需要用的信息只有$\delta(1),\delta(2),\cdots,\delta(N)$、状态转移概率矩阵$A$，观测值$O$，观测概率矩阵$B$，在时刻$t=3$，一共计算了$N$个之前的最优路径，当计算完成之后，得到了$N$个最优路径以及相应的概率值，依次进行计算，完成后续时刻的最优路径计算，这样就完成了一个递推过程，这就是动态规划，也就是维特比算法的基本思想。

算法10.5（维特比算法）

输入：模型$\lambda=(A,B,\pi)$和观测$O=(o_1,o_2,\cdots,o_T)$

输出：最优路径$I^*=(i_1^*,i_2^*,\cdots,i_T^*)$

(1)初始化

$$\delta_1(i)=\pi_i b_i(o_1),\quad i=1,2,\cdots,N$$

$$\psi_1(i)=0,\quad i=1,2,\cdots,N$$

(2)递推，对$t=2,3,\cdots,T$

$$\delta_t(i)=\max_{1\leqslant j\leqslant N}\left[\delta_{t-1}(j)a_{ji}\right]b_i(o_t),\quad i=1,2,\cdots,N$$

$$\psi_t(i)=\arg\max_{1\leqslant j\leqslant N}\left[\delta_{t-1}(j)a_{ji}\right],\quad i=1,2,\cdots,N$$

(3)终止

$$P^*=\max_{1\leqslant i\leqslant N}\delta_T(i)$$

$$i_T^*=\max_{1\leqslant i\leqslant N}\left[\delta_T(i)\right]$$

(4)最优路径回溯，对$t=T-1,T-2,\cdots,1$

$$i_t^*=\psi_{t+1}(i_{t+1}^*)$$

求得最优路径$I^*=(i_1^*,i_2^*,\cdots,i_T^*)$.

现考察最大化最优路径对应的最大化概率值是什么样的概率，在(2)步的递推，最大化的是$\delta_t(i)$，书中给出了定义$\delta_t(i)=\max\limits_{i_1,i_2,\cdots,i_t}P(i_t=i,i_{t-1},\cdots,i_1,o_t,\cdots,o_1|\lambda)$，只固定了时刻$t$的状态值，然后最大化上述概率，求解使得该概率最大的前面$t-1$个时刻的状态变量的取值，但是预测算法的目标是$\arg\max P(I|O,\lambda)$，这两个概率的最大化是等价的，因为用乘法公式可得$P(I,O)=P(O)P(O|I)$，可知$P(O)$与I无关，所以在I取不同值时，$P(O)$是不变的，所以$\arg\max P(I|O,\lambda)$和$\arg\max P(I,O)$是等价的。

26 第10章-隐马尔科夫模型-前向算法

本节推导两个公式$\alpha_{t+1}(i)=\left[\sum\limits_{j=1}^{N}\alpha_t(j)a_{ji}\right]b_i(o_{t+1})$和$P(O|\lambda)=\sum\limits_{i=1}^{N}\alpha_T(i)$

26.1 式10.17的推导

$\alpha_t(i)$涉及到时刻t的观测值$o_1,o_2,\cdots,o_t$以及状态值q_i，记作$\alpha_t(i)=P(o_1,o_2,\cdots,o_t,i_t=q_i)$，这里忽略$\lambda$的条件概率。

前向算法用来做的就是概率计算，当λ给定时，观测值$o_1,o_2,\cdots,o_t$出现的概率为$P(o_1,o_2,\cdots,o_T)$，这个概率是边际概率，$P(o_1,o_2,\cdots,o_T,i_T=q_i)$是联合概率，可得

$P(o_1,o_2,\cdots,o_T)=\sum\limits_{i=1}^{N}P(o_1,o_2,\cdots,o_T,i_T=q_i)=\sum\limits_{i=1}^{N}\alpha_T(i)$，式10.17可证.

26.2 式10.16的推导

最终的目标就是要求出$\alpha_T(1), \alpha_T(2), \cdots, \alpha_T(N)$，前向算法就是从$(\alpha_1(1), \alpha_1(2), \cdots, \alpha_1(N))$推到$(\alpha_2(1), \alpha_2(2), \cdots, \alpha_2(N))$，一直到$(\alpha_T(1), \alpha_T(2), \cdots, \alpha_T(N))$

为了区分符号，修改为$\alpha_t(j) = P(o_1, o_2, \cdots, o_t, i_t = q_j)$，根据定义

$\alpha_{t+1}(i) = P(o_1, o_2, \cdots, o_t, o_{t+1}, i_{t+1} = q_i)$

根据边际概率和联合概率可得：

$$P(o_1, o_2, \cdots, o_t, o_{t+1}, i_{t+1} = q_i) = \sum_{j=1}^{N} P(o_1, o_2, \cdots, o_t, o_{t+1}, i_{t+1} = q_i, i_t = q_j)$$

将等号右边的式子进行乘法公式拆分：

$$\sum_{j=1}^{N} P(o_1, o_2, \cdots, o_t, o_{t+1}, i_{t+1} = q_i, i_t = q_j) = \sum_{j=1}^{N} P(o_1, o_2, \cdots, o_t, i_t = q_j) P(o_{t+1} | i_{t+1} = q_i) P(i_{t+1} = q_i | i_t$$

根据隐马尔可夫模型：

$$P(o_{t+1} | i_{t+1} = q_i) = b_i(o_{t+1})$$
$$P(i_{t+1} = q_i | i_t = q_j) = a_{ji}$$
$$P(o_1, o_2, \cdots, o_t, i_t = q_j) = \alpha_t(j)$$

$$\therefore \sum_{j=1}^{N} P(o_1, o_2, \cdots, o_t, i_t = q_j) P(o_{t+1} | i_{t+1} = q_i) P(i_{t+1} = q_i | i_t = q_j) = \left[\sum_{j=1}^{N} \alpha_t(j) a_{ji} \right] b_i(o_{t+1})$$

$$\therefore \alpha_{t+1}(i) = \left[\sum_{j=1}^{N} \alpha_t(j) a_{ji} \right] b_i(o_{t+1})$$，式10.16可证.

27 第10章-隐马尔科夫模型-维特比算法

维特比算法是求解给定的观测条件下，使得概率最大的状态变量的序列预测，也就是说，求解已知观测序列$i_1, i_2, \cdots, i_T$的出现的概率最大：$\max P(i_1, i_2, \cdots, i_T | o_1, o_2, \cdots, o_T, \lambda)$等价于

$\max P(i_1, i_2, \cdots, i_T, o_1, o_2, \cdots, o_T)$

$$\because P(i_1, i_2, \cdots, i_T | o_1, o_2, \cdots, o_T, \lambda) = \frac{P(i_1, i_2, \cdots, i_T, o_1, o_2, \cdots, o_T)}{P(o_1, o_2, \cdots, o_T)}$$

因为$P(o_1, o_2, \cdots, o_T)$已知，所以相当于求解$\max P(i_1, i_2, \cdots, i_T, o_1, o_2, \cdots, o_T)$

书中借助了一个新定义的函数，然后用这个函数的递推关系，推导出了维特比算法，这个函数是

$$\delta_t(i) = \max P(i_1, i_2, \cdots, i_{t-1}, i_t = i, o_t, \cdots, o_1)$$

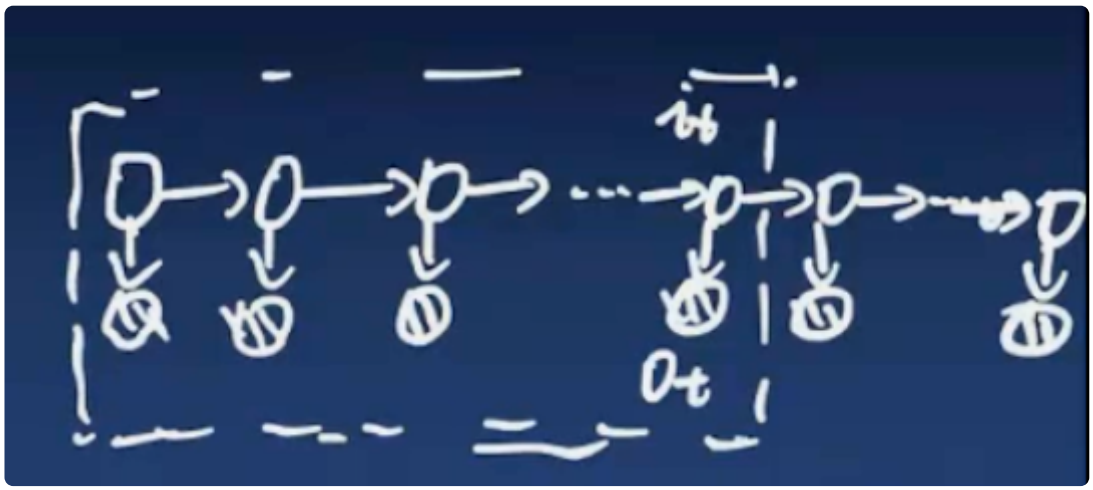

图10-1 HMM模型

函数所表达的就是图10-1中虚线框的部分，当固定i_t状态的取值，寻找前面的最优路径$(i_1, i_2, \cdots, i_{t-1})$使得$\delta_t(i)$最大。

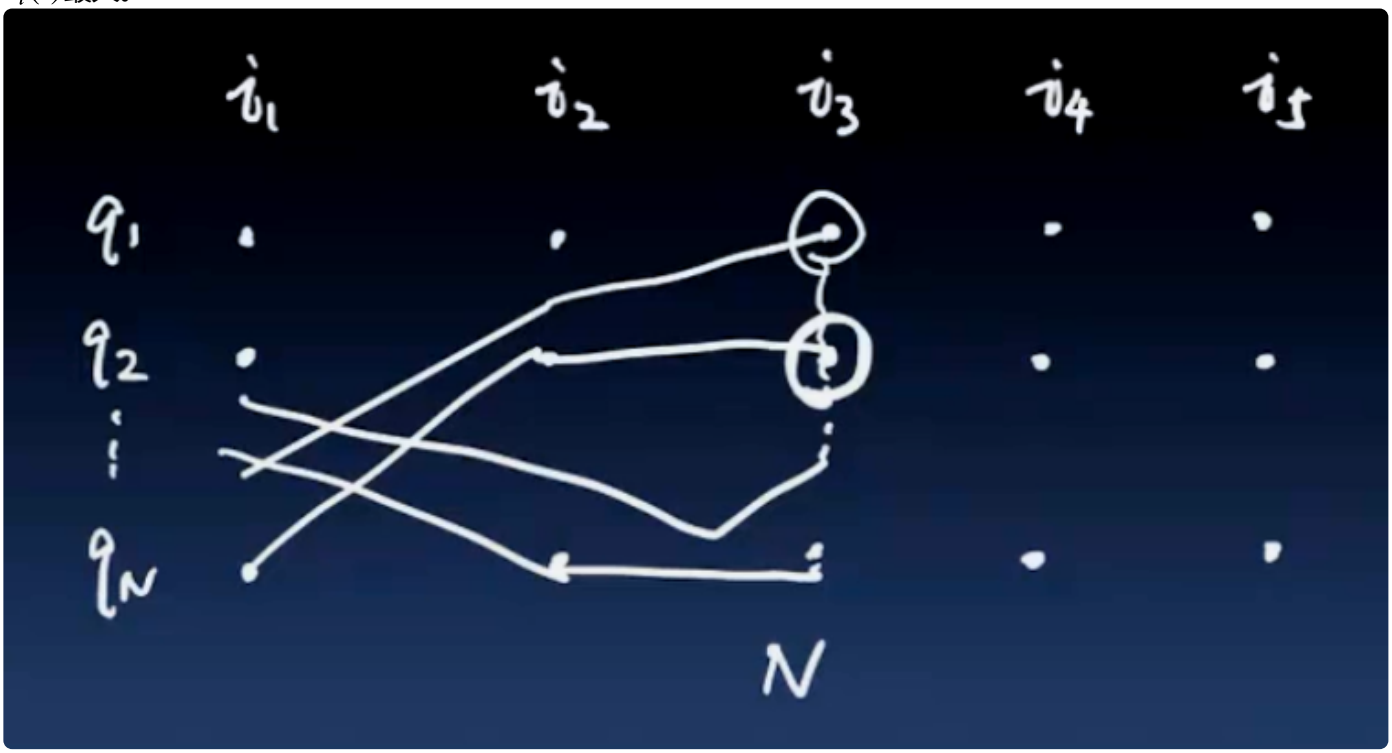

图10-2 维特比算法最优路径

如图10-2所示，假如i_t取值为3，也就是说固定i_3取值（图中的点坐标为(q_2, i_3)），路径为图中所示通过坐标点(q_2, i_3)的折线，变换i_3固定的值（坐标点(q_1, i_3)），找寻前面的路径，使得出现的概率最大，对于每一个不同的取值$q_1, \cdots, q_N$，都有不同的路径，一共有N条路径。

接下来考察$\delta_{t+1}(i)$，在$t+1$的时刻，状态取值为i的最大概率：

$$\delta_{t+1}(i) = \max_{i_1,\cdots,i_t} P(i_1, \cdots, i_t, i_{t+1} = i, o_{t+1}, \cdots, o_1)$$

> Bellman最优性原理：
>
> 假设i_{t+1}的最优路径中，与i_t相连的路径假如已经确定，那么i_t之前的路径也已经确定了，并且不会改变，也就说，在求解i_{t+1}的最优路径时，只需要求解i_{t+1}与i_t之前的最优路径，而不需要再求解i_t之前的最优路径了。

根据Bellman最优性原理:

$$\begin{aligned}\delta_{t+1}(i) &= \max_{i_1,\cdots,i_t} P(i_1, \cdots, i_t, i_{t+1} = i, o_{t+1}, \cdots, o_1) \\ &= \max_j \delta_t(j) P(i_{t+1}|i_t = j) P(o_{t+1}|i_{t+1} = i)\end{aligned}$$

其中$\delta_t(j) = \max P(i_1, i_2, \cdots, i_{t-1}, i_t = j, o_t, \cdots, o_1)$

$\because a_{ji} = P(i_{t+1}|i_t = j), b_i(o_{t+1}) = P(o_{t+1}|i_{t+1} = i)$

$\therefore \delta_{t+1}(i) = \max_j \left[\delta_t(j) a_{ji}\right] b_i(o_{t+1})$

在递推的过程中，上述刚刚介绍了如图10-2中，从i_1到i_2有一个最优路径，对i_3的每一个状态也有一个最优路径，对i_4的每一个状态也有一个最优路径，每考虑一个状态时，最后都得到了N个最优路径，对应i_t下状态的N个取值前面的路径。

当计算i_3的N个最优路径，为什么要将这些路径都保存起来呢？为什么不能直接寻找概率最大的路径？其实是不能的，如果只保留了i_3状态下概率最大的那个路径，当前的最大路径并不能保证，当考虑下一个时刻的最优

路径，该路径是否经过前面状态的概率值最大的路径，因为后面这一步还会带来概率的变化，所以对于i_t的每一个状态，都需要记录之前的最优路径，一直记录到i_T，于是就有了N条最优路径，当计算到i_T时，就可以找到最终概率最大的值对应的最后一个状态，然后再反推前面的路径。

算法10.5（维特比算法）

输入：模型$\lambda = (A, B, \pi)$和观测$O = (o_1, o_2, \cdots, o_T)$

输出：最优路径$I^* = (i_1^*, i_2^*, \cdots, i_T^*)$

(1)初始化

$$\delta_1(i) = \pi_i b_i(o_1), \quad i = 1, 2, \cdots, N$$

$$\psi_1(i) = 0, \quad i = 1, 2, \cdots, N$$

(2)递推，对$t = 2, 3, \cdots, T$

$$\delta_t(i) = \max_{1 \leqslant j \leqslant N} \left[\delta_{t-1}(j) a_{ji}\right] b_i(o_t), \quad i = 1, 2, \cdots, N$$

$$\psi_t(i) = \arg\max_{1 \leqslant j \leqslant N} \left[\delta_{t-1}(j) a_{ji}\right], \quad i = 1, 2, \cdots, N$$

(3)终止

$$P^* = \max_{1 \leqslant i \leqslant N} \delta_T(i)$$

$$i_T^* = \max_{1 \leqslant i \leqslant N} [\delta_T(i)]$$

(4)最优路径回溯，对$t = T-1, T-2, \cdots, 1$

$$i_t^* = \psi_{t+1}(i_{t+1}^*)$$

求得最优路径$I^* = (i_1^*, i_2^*, \cdots, i_T^*)$.

28 第11章-条件随机场-导读

学习本章内容有两个方法：(1)可以对照第10章-隐马尔可夫模型，它们所要解决的问题属于标注问题，第10章是用有向图模型，第11章是用无向图模型来解决标注问题，第10章的第2、3、4节分别解决了隐马尔可夫模型三个问题，在第11章对应的第3、4、5节也分别解决了这三个问题；(2)第11章可以对照第6章学习，因为在条件随机场，使用因子模型表示概率模型时，实际上用的是对数线性模型，第6章的最大熵模型用的额也是对数线性模型，在求解的过程中有相通的地方，第11章也涉及到了改进的迭代尺度法和拟牛顿法。

28.1 概率无向图模型

28.1.1 概率图模型

概率图模型一共分为有向图（贝叶斯网络）和无向图（马尔可夫随机场），有向图主要描述变量间的因果关系，用有向的边来表示因果关系；无向图是不带有箭头的边连接的随机变量。

定义11.1（概率无向图模型）

设有联合概率分布$P(Y)$，由无向图$G = (V, E)$表示，V表示结点集合，E表示边集合，在图G中，结点表示随机变量，边表示随机变量之间的依赖关系。如果联合概率分布$P(Y)$满足成对、局部或全局马尔可夫性，就称此联合概率分布为概率无向图模型或马尔可夫随机场。

28.1.2 马尔可夫性

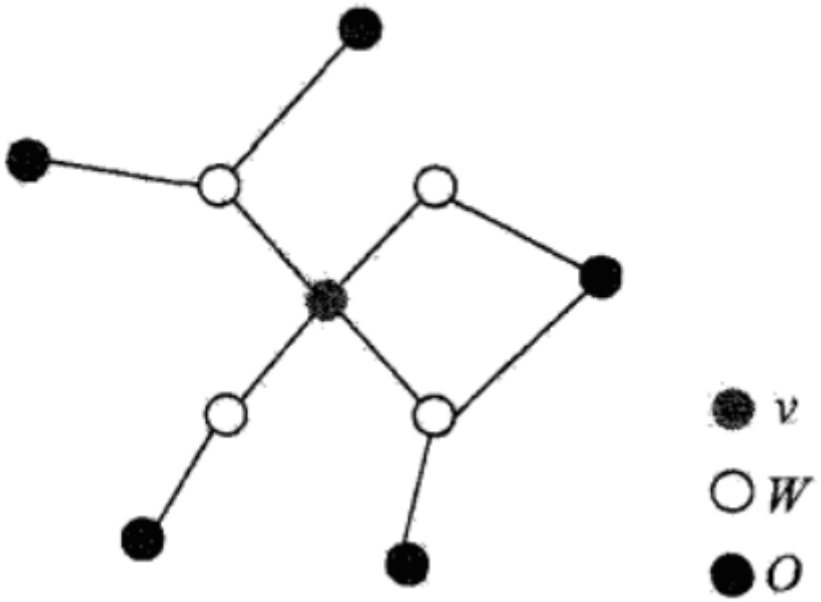

图11-1 局部马尔可夫性

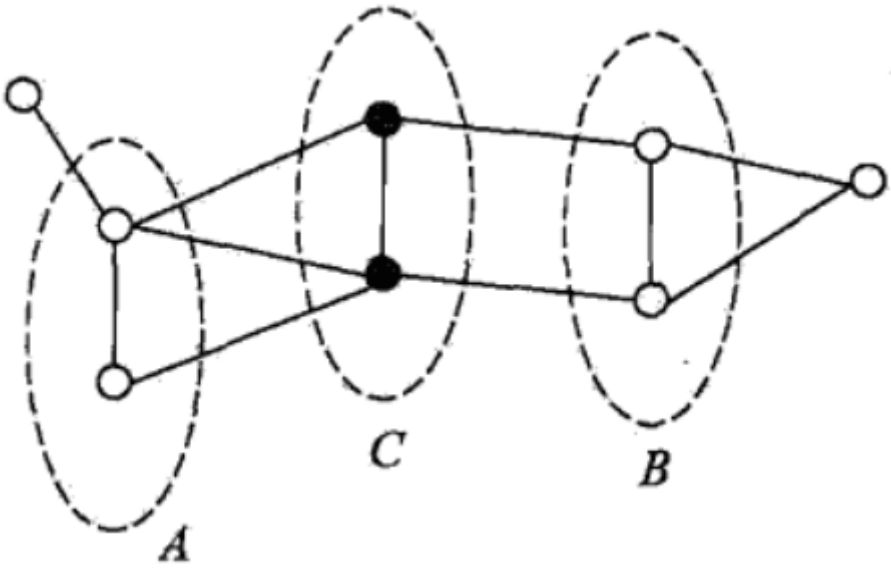

图11-2 全局马尔可夫性

图11-1和图11-2是两个无向图模型，是由圆形的结点和边构成，结点表示随机变量，图中空心结点和带阴影的结点是为了介绍马尔可夫性，而不表示隐变量或可观测变量。在马尔可夫随机场中，用无向边连接两个结点，表示这两个结点的关系，但并不像有向图中，表示依赖（因果）关系。用图模型表示一个概率模型，最重要的就是图模型变量的依赖性或独立性。在无向图中，引入了马尔可夫性描述这种关系，书中介绍了三种：成对马尔可夫性、局部马尔可夫性和全局马尔可夫性。

成对马尔可夫性： 在图11-1中，一共有10个结点（即10个随机变量），任意找两个没有边直接连接的结点，假设有两个随机变量(u, v)没有边相连，剩下的8个随机变量记为O，当给定O时，u和v是独立的，即$P(u, v|O) = P(u|O)P(v|O)$。

局部马尔可夫性： 在无向图11-1中，任意找一个结点v，与v有边相连的所有结点记为W，其余5个结点记为O，当给定W时，v和O是独立的，即$P(v, O|W) = P(v|W)P(O|W)$。

全局马尔可夫性： 在无向图11-2中，一共有8个结点（即有8个随机变量），取中间两个随机变量记为集合C，当将集合C从图中删掉之后，那么剩下的6个结点分成了两个部分，可知左边的3个结点和右边的3个结点没有

任何边将它们相连，当给定C时，A和B是独立的，即$P(A, B|C) = P(A|C)P(B|C)$。

为什么说这三个马尔可夫性是等价的？这里等价的意思为任意一个结点满足成对马尔可夫性等价于任意一个结点满足局部马尔可夫性，也等价于这些结点满足全局马尔可夫性。

28.1.3 概率无向图模型的因子分解

无向图模型提供了一种分析随机变量之间关系的手段，当已知一组随机变量，能很清楚表达随机变量之间关系的方法是联合概率分布P(Y)，根据已知的无向图模型，可以得到联合概率分布$P(Y)$的形式。

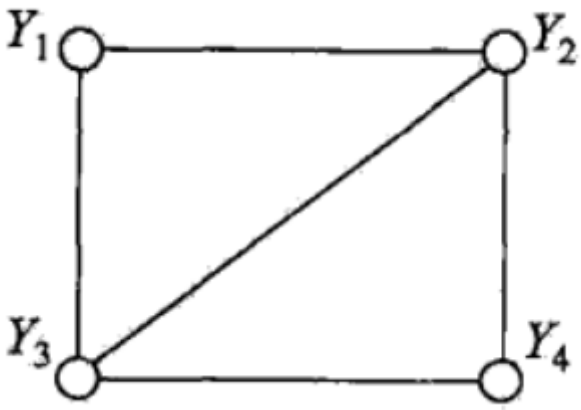

图11-3 无向图的团和最大团

团：在无向图模型中有一些结点（随机变量），这些结点中任意两个结点都有边相连，这些随机变量组成的集合称为团。如图11-3中，Y_1和Y_2有一条边相连，$\{Y_1, Y_2\}$可以称为一个团，同理Y_2和Y_3有一条边相连，$\{Y_2, Y_3\}$也可以称为一个团，不能将$\{Y_1, Y_2, Y_4\}$称为一个团，因为Y_1和Y_4之间是没有边相连的，$\{Y_1, Y_2, Y_3\}$可以组成一个团。

最大团：当给定一个团，在该团中不能再加进任何一个结点使其成为更大的团，比如$\{Y_1, Y_2, Y_3\}$就是一个最大团，

> **定理11.1（Hammersley-Clifford定理）**概率无向图模型的联合概率分布$P(Y)$可以表示为如下形式：
>
> $$P(Y) = \frac{1}{Z}\prod_C \Psi_C(Y_C)$$
>
> $$Z = \sum_Y \prod_C \Psi_C(Y_C)$$
>
> 其中，C是无向图的最大团，Y_C是C的结点对应的随机变量，$\Psi_C(Y_C)$是C上定义的严格正函数，乘积是在无向图所有的最大团上进行的，$\Psi_C(Y_C) = \exp\{-E(Y_C)\}$。$E(Y_C)$称为能量函数。

在图11-3中先寻找最大团，易知$\{Y_1, Y_2, Y_3\}$和$\{Y_2, Y_3, Y_4\}$是最大团，可以写出联合概率分布$P(y_1, y_2, y_3, y_4) = \frac{1}{Z}\Psi_1(y_1, y_2, y_3)\Psi_2(y_2, y_3, y_4)$，这个联合概率分布可以表示为两个因子的乘积，每一个因子都是关于最大团的函数，前面$\frac{1}{Z}$保证概率分布对所有的随机变量求积分等于1，并且$\Psi_1 \geqslant 0, \Psi_2 \geqslant 0$。

28.1.4 总结

本节主要介绍概率无向图模型和因子分解（表示联合概率分布），这里介绍一些扩展内容，当表示一个随机向量各个分量之间的关系，概率$P(Y)$和一组随机变量的关系是一一对应的，即所有的随机变量都能表示成为一个概率分布，所有的概率分布都会一一对应一个随机向量。还介绍了，可以用有向图模型或者无向图模型表示概率分布，第10章的隐马尔可夫模型可以用有向图表示概率分布，第11章条件随机场可以用无向图表示概率分布。

28.2 条件随机场的定义与形式

28.2.1 条件随机场的定义

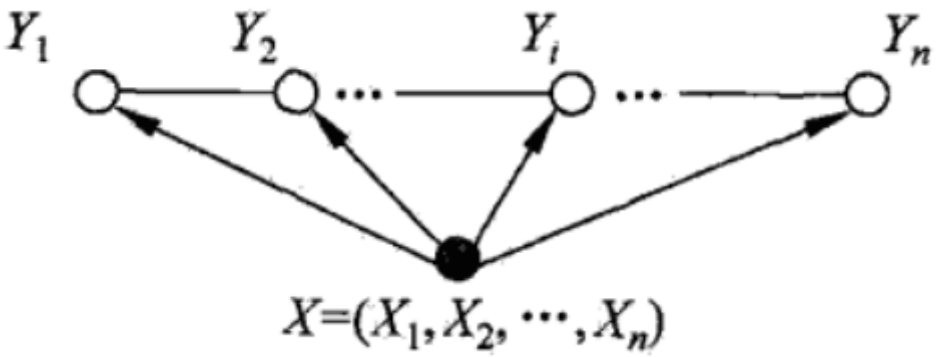

图11-4 线性链条件随机场

如图11-4，如果只考察随机变量$Y=(Y_1, Y_2, \cdots, Y_n)$，这些变量是用无向边连接的，属于无向图（马尔可夫随机场），但现在有另一组随机变量$X=(X_1, X_2, \cdots, X_n)$，对每个随机变量$Y$都产生影响，由于$X$已知，在无向图中就添加了这样一个信息，$X$为条件，$X$和$Y$合起来称为条件随机场，由于$Y$是线性连接的，所以整个模型称为线性链条件随机场。

$$P(Y_v|X, Y_w, w \neq v) = P(Y_v|X, Y_w, w \sim v)$$

其中v表示任意一个结点，$w \neq v$表示v以外的所有结点，$w \sim v$表示与v有边连接的所有结点，上述等式表示给定X，Y，w的条件下，给定其他所有结点v的分布等于给定和它相邻的结点v的分布，其实是局部马尔可夫性。

28.2.2 条件随机场的参数化形式

首先考察条件随机场的最大团个数，根据图11-4，最大团为$\{Y_1, Y_2\}, \{Y_2, Y_3\}, \cdots, \{Y_{n-1}, Y_n\}$，所以条件随机场的因子分解（概率分布函数）是每一个最大团函数的乘积。

书中给出的条件随机场参数化形式如下：

$$P(y|x) = \frac{1}{Z(x)} \exp\left(\sum_{i,k} \lambda_k t_k(y_{i-1}, y_i, x, i) + \sum_{i,l} \mu_l S_l(y_i, x, i) \right)$$

其中$Z(x) = \sum_y \exp\left(\sum_{i,k} \lambda_k t_k(y_{i-1}, y_i, x, i) + \sum_{i,l} \mu_l S_l(y_i, x, i) \right)$，因为是条件随机场，给定了变量$x$，再观察$\sum_{i,k} \lambda_k t_k(y_{i-1}, y_i, x, i)$，最大团的表示是$\lambda_k t_k(y_{i-1}, y_i, x, i)$，可以理解为$\exp(\lambda_k t_k(y_{i-1}, y_i, x, i))$，故最大团函数为$\exp(\lambda_k t_k(y_{i-1}, y_i, x, i))$，还多了一项$\mu_l S_l(y_i, x, i)$，因为在无向概率图中还有一个$X$。

t_k, s_l是两个特征函数，在第6章介绍过特征函数，通常，特征函数t_k, s_l取值为1或0，当满足特征条件时取值为1，否则为0，t_k是关于y_i, y_{i-1}特征函数，s_l是关于y_i特征函数，函数t_k称为转移特征，函数s_l称为状态特征。条件随机场的参数是λ_k, μ_l。

28.2.3 条件随机场的简化形式

$$P(y|x)=\frac{1}{Z(x)}\exp\sum_{k=1}^{K}w_k f_k(y,x)$$

其中

$$Z(x)=\sum_{y}\exp\sum_{k=1}^{K}w_k f_k(y,x)$$

$$f_k(y_{i-1},y_i,x_i)=\begin{cases}t_k(y_k,y_i,x,i), & k=1,2,\cdots,K_1\\ s_l(y_i,x,i), & k=K_1+l;l=1,2,\cdots,K_2\end{cases}$$

$$w_k=\begin{cases}\lambda_k, & k=1,2,\cdots,K_1\\ \mu_l, & k=K_1+l;l=1,2,\cdots,K_2\end{cases}$$

28.2.4 条件随机场的矩阵形式

$$P_w(y|x)=\frac{1}{Z_w(x)}\prod_{i=1}^{n+1}M_i(y_{i-1},y_i|x)$$

其中$Z_w(x)$为规范化因子，是n+1个矩阵的乘积的(start,stop)元素：

$$Z_w(x)+(M_1(x)M_2(x)\cdots,M_{n+1}(x))_{\text{start,stop}}$$

28.3 条件随机场的概率计算问题

条件随机场的概率计算问题等价于第10章的概率计算问题，利用条件随机场的矩阵形式，计算$P(Y=y_i|x)$，和第10章的区别是求解状态概率，而第10章求观测概率，本节采用的算法是前向-后向算法。

28.4 条件随机场的学习算法

在对数线性模型中，参数w就是权重，这个权重包含转移特征、状态特征的权重，和第6章的算法类似，有两种算法：改进的迭代尺度法，拟牛顿法。这两个算法都用在对数线性模型中。

28.4.1 条件随机场的预测算法

在解决标注问题，采用的是维特比算法（动态规划），计算$y^*=\arg\max_{y}P_w(y|x)$

29 第11章-条件随机场-条件随机场的矩阵形式

条件随机场（CRF）的参数化模型：

$$P(y|x)=\frac{1}{Z(x)}\exp\left(\sum_{i,k}\lambda_k t_k(y_{i-1},y_i,x,i)+\sum_{i,l}\mu_l s_l(y_i,x,i)\right)\tag{1}$$

其中包含了参数λ_k,μ_l，特征函数t_k,s_l，特征函数t_k为转移特征，时刻i和$i-1$对应的状态，以及给定的向量X和位置i；s_l为状态特征，只和当前的时刻i的状态有关，以及给定的向量X和位置i。

条件随机场给出的是一个关于y的条件分布，和隐马尔可夫模型不同，在HMM中，联合概率分布为$P(O,I|\lambda)$，在CRF中，观测用随机变量X表示，所求的模型也就是状态y的联合分布，一旦给定特征函数，模型的参数取决于λ_k,μ_l。

条件随机场的矩阵形式：

$$P_w(y|x)=\frac{1}{Z_w(x)}\prod_{i=1}^{n+1}M_i(y_{i-1},y_i|x)\tag{2}$$

其中w为权重，(1)中的$Z(x)$和(2)中的$Z_w(x)$是一致的，区别就在后面一部分。

29.1 推导从参数化形式转化为矩阵形式

以下推导过程中，在特征函数里省略x, i：

$$\exp\left(\sum_{i,k}\lambda_k t_k(y_{i-1},y_i)+\sum_{i,l}\mu_l s_l(y_i)\right)$$

$$=\exp\left\{\sum_i\left[\sum_k\lambda_k t_k(y_{i-1},y_i)+\sum_l\mu_l s_l(y_i)\right]\right\}$$

$$=\prod_i\exp\left(\sum_k\lambda_k t_k(y_{i-1},y_i)+\sum_l\mu_l s_l(y_i)\right)$$

对比(1)，可得

$$M_i(y_{i-1},y_i|x)=\exp\left(\sum_k\lambda_k t_k(y_{i-1},y_i)+\sum_l\mu_l s_l(y_i)\right)$$

书中将$t_k(y_{i-1},y_i)$和$s_l(y_i)$组合成状态函数，将λ_k和μ_l组合成权重向量。为什么说$M_i(x)$是一个矩阵，因为y_{i-1}和y_i都是状态变量，可以通过该公式表示为一个矩阵，得到$M_i(x)=[M_i(y_{i-1},y_i|x)]$

29.2 例11.2讲解

例11.2 给定一个由图11-5所示的线性链条件随机场，观测序列x，状态序列y，$i=1,2,3,n=3$，标记$y_i\in\{1,2\}$，假设$y_0=\text{start}=1, y_4=\text{stop}=1$，各个位置的随机矩阵$M_1(x), M_2(x), M_3(x), M_4(x)$分别是

$$M_1(x)=\begin{bmatrix}a_{01} & a_{02}\\ 0 & 0\end{bmatrix},\quad M_2(x)=\begin{bmatrix}b_{11} & b_{12}\\ b_{21} & b_{22}\end{bmatrix}$$

$$M_3(x)=\begin{bmatrix}c_{11} & c_{12}\\ c_{21} & c_{22}\end{bmatrix},\quad M_4(x)=\begin{bmatrix}1 & 0\\ 1 & 0\end{bmatrix}$$

试求状态序列y以start为起点，stop为终点所有路径的非规范化概率及规范化因子。

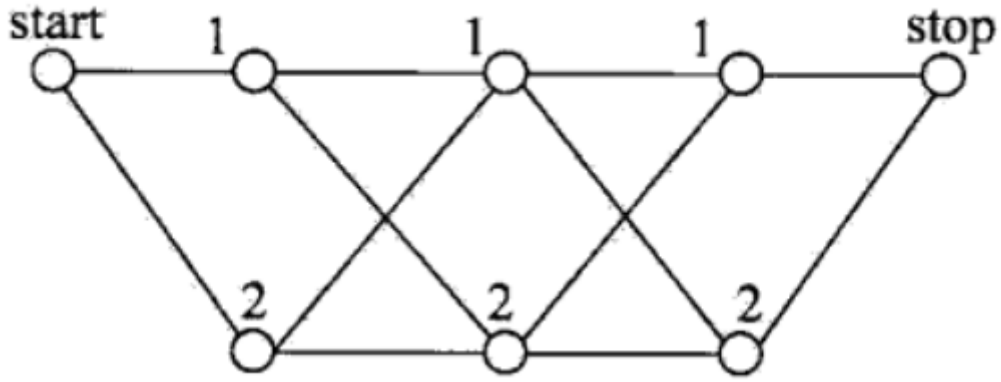

图11-5 状态路径

解答：

题中已知M_1, M_2, M_3, M_4，矩阵中的参数都是已知，已知了矩阵相当于已知条件随机场中的模型形式，以及所有的特征函数和特征向量。

所求的非规范化概率就是4个矩阵相乘，因为M都是矩阵，需要计算是哪个元素相乘，计算非规范化概率不考虑前面的$\frac{1}{Z(x)}$（即规范化因子），M相当于第10章隐马尔可夫模型的状态转移矩阵，初始的概率π相当于M_1的矩阵中，可知$P(y_1=1)\propto a_{01}$，$P(y_1=2)\propto a_{02}$。

为什么是非规范化？因为在第10章隐马尔可夫模型中，每一个状态转移概率矩阵中每一行的和概率等于1，这样就保证了这是一个条件概率分布，但是在矩阵M中，求概率的时候，并没有考虑到前面的规范化因子，所以在矩阵M中每一行的和并不是1。当考虑状态转移关系时，没有考虑这是一个概率，如果求解$P(y_1, y_2, y_3)$联合概率分布，就需要规范化因子，求解局部概率，不考虑规范化。

（1）要计算$y=(1,1,1)$路径的非规范化概率，从矩阵M_1中可得$P(y_0=1, y_1=1)=a_{01}$，从矩阵M_2中可得$P(y_1=1, y_2=1)=b_{11}$，从矩阵M_3中可得$P(y_2=1, y_3=1)=c_{11}$，从矩阵M_4中可得$P(y_3=1, y_4=1)=1$，所以$y=(1,1,1)$路径的非规范化概率为$a_{01}b_{11}c_{11}$，同理可得$y=(1,2,1)$路径的非规范化概率为$a_{01}b_{12}c_{21}$，可得到8个非规范化概率分别是：

$$a_{01}b_{11}c_{11},\quad a_{01}b_{11}c_{12},\quad a_{01}b_{12}c_{21},\quad a_{01}b_{12}c_{22}$$
$$a_{02}b_{21}c_{11},\quad a_{02}b_{21}c_{12},\quad a_{02}b_{22}c_{21},\quad a_{02}b_{22}c_{22}$$

（2）计算规范化因子，因为要保证概率和为1，所以规范化因子$Z(x)$是8个规范化概率的和

$$Z(x)=a_{01}b_{11}c_{11}+a_{01}b_{11}c_{12}+a_{01}b_{12}c_{21}+a_{01}b_{12}c_{22}+a_{02}b_{21}c_{11}+a_{02}b_{21}c_{12}+a_{02}b_{22}c_{21}+a_{02}b_{22}c_{22}$$

29.3 对比矩阵M和状态转移矩阵A

1. 矩阵A是严格的条件概率分布，每一行的和为1，但是M没有要求
2. 在隐马尔可夫模型中，状态转移概率A是不随状态i变化的，在CRF中并没有假设状态是不能变的，参数会更多，模型也会更灵活，所以矩阵表示为M_i

30 第11章-条件随机场-拟牛顿法（附录B）

拟牛顿法也是求解最优化问题的一个方法，针对牛顿法的缺陷提出来的新方法。

30.1 牛顿法

对于一个无约束的最优化问题

$$\min_{x\in \mathrm{R}^n} f(x)$$

假设$f(x)$有二阶连续偏导数，使用迭代的方法求解最优解x^*，若第k步的迭代值为$x^{(k)}$，将$f(x)$在$x^{(k)}$进行二阶泰勒展开：

$$f(x)\doteq f(x^{(k)})+\nabla f''(x^{(k)})(x-x^{(k)})+\frac{1}{2}(x-x^{(k)})^T H(x^{(k)})(x-x^{(k)})$$

其中$H(x^{(k)})$是$f(x)$的海赛矩阵(Hesse matrix)

$$H(x)=\left[\frac{\partial^2 f}{\partial x_i \partial x_j}\right]_{n\times n}$$

在这点$x^{(k)}$的值。 函数$f(x)$有极值，可计算$\nabla f(x)=0$求极小值：

$\nabla f(x)=\nabla f''(x^{(k)})+H_k(x-x^{(k)})=0$

$\therefore x^{(k+1)}=x^{(k)}-H_k^{-1}\nabla f(x^{(k)})$

牛顿法基本思路：

1. 在迭代的过程中，用一个二次函数逼近每一个迭代的点对应的函数值，如果函数有极小值，需要保证该函数是凸函数，对应于一维函数中，其二阶导数要大于0，对应于n维的情况，$H_k(x^{(k)})$是正定矩阵。
2. 在迭代的过程中需要计算H_k^{-1}，对于每一个k，H_k是不一样的，所以计算H_k^{-1}（$n\times n$维矩阵）的计算量是非常大的，这就是牛顿法的一个缺陷。
3. 一般在用迭代的方式求解一个函数极值时，迭代基本的形式都是$x^{(k+1)}=x^{(k)}+\lambda p_k$，在牛顿法中，更新步长$\lambda=1$，更新方向$p_k=-H_k^{-1}\nabla f(x^{(k)})$，如果保证是正定的，就可以保证这是一个向下的方向（可以达到收敛）
4. 除了牛顿法，在梯度下降法中，迭代形式是$x^{(k+1)}=x^{(k)}-\lambda\nabla f(x^{(k)})$，在$\lambda$比较小时，可以保证使得$f(x)$下降的方向，只用到了$f(x)$在点$x^{(k)}$的一阶导，对应到多维中就是梯度。而在牛顿法中，用到了一阶导

和二阶导，因为牛顿法用了二阶导，比梯度下降法更快，即能更快地找到$f(x)$取极小值对应的x^*。

30.2 拟牛顿法

想有一个新方法：（1）同样借助了$f(x)$在点x^k的二阶导，所以比梯度下降法的收敛速度要更快；（2）寻找一个能代替H_k^{-1}的矩阵，使得求逆的过程更加便捷；拟牛顿法就具备上述两点。　　首先考察H_k矩阵，（1）该矩阵是正定的；（2）在牛顿法中，需要求解$\nabla f(x^{k+1}) = \nabla f''(x^{(k)}) + H_k(x - x^{(k)})$，当求解迭代$x^{(k+1)}$时，就是根据$f(x)$在点$x^{(k+1)}$的梯度等于0得到的求解公式。
令$\delta_k = x - x^k, y_k = \nabla f(x^{(k+1)}) - \nabla f(x^{(k)})$
$\therefore H_k\delta_k = y_k$，希望替代矩阵也满足这个关系式。
可得到H_k的两个约束：$H_k\delta_k = y_k, \delta_k = H_k^{-1}y_k$称为拟牛顿条件。

但是不能直接用来求解H_k，因为$\delta_k = x - x^k, y_k = \nabla f(x^{(k+1)}) - \nabla f(x^{(k)})$，首先求解$x^{(k+1)}$的值和对应的$\nabla f(x^{(k+1)})$，这个替代矩阵就是为了计算$x^{(k+1)}$，所以这就是不能直接使用的一个拟牛顿条件，为了让这个条件可以使用，将H_k换成H_{k+1}，即$H_{k+1}\delta_k = y_k, \delta_k = H_{k+1}^{-1}y_k$，也就是说用$x^{(k+1)}$求解$H_{k+1}$，用$H_{k+1}$的替代求解$x^{(k+2)}$

30.3 书上对应的两个算法

DFP算法用矩阵G_k代替H_k^{-1}，这个替代矩阵有一个递推的形式，这个关系是由G_k加上两个秩为1的矩阵构成的，假设$G_{k+1} = G_k + aVV^T + bUU^T$，寻找满足条件的$aV, bU$，就得到了书中DFP算法中给出的$G_k$递推公式：

$$G_{k+1} = G_k + \frac{\delta_k\delta_k^T}{\delta_k^T y_k} - \frac{G_k y_k y_k^T G_k}{y_k^T G_k y_k}$$

从而找到了H_{k+1}^{-1}的替代。　　BFGS算法用矩阵B_k代替H_k，这个替代矩阵同样有一个递推的形式，这个关系同样是由B_k加上两个秩为1的矩阵构成的，书中给出了BFGS算法中的B_k递推公式：

$$B_{k+1} = B_k + \frac{y_k y_k^T}{y_k^T\delta_k} - \frac{B_k\delta_k\delta_k^T B_k}{\delta_k^T B_k\delta_k}$$

其实还有一个问题： 在算法B.3中，依然要计算B_k^{-1}，为什么这个时候求解B_k^{-1}就可行，之前求解H_k^{-1}不可行呢？因为如果求出了B_k^{-1}，那么下一步B_{k+1}^{-1}和B_k^{-1}是满足一个函数关系式的，那么这个函数里面关于B_k^{-1}就不需要再计算了，这个关系式的推导见书上第223也的标注：

> Sherman-Morrison公式：假设A是n阶可逆矩阵，u, v是n维向量，且$A + uv^T$也是可逆矩阵，则
> $$(A + uv^T)^{-1} = -\frac{A^{-1}uv^T A^{-1}}{1 + v^T A^{-1}u}$$

In [ ]: