# 统计学习方法整理笔记（1-4）

## 1 统计学习概述

统计学习三要素：模型、策略、算法。

### 1.1 模型

模型就是所要学习的条件概率分布或者决策函数

### 1.2 策略

策略即是决定用什么样的准则学习或选择最优的模型。

1. 损失函数（loss function）

统计学习常用的损失函数有以下几种：

（1）0-1 损失函数（0-1 loss function）

$$L(Y,f(X))=\begin{cases}1, & Y\neq f(X)\\ 0, & Y=f(X)\end{cases} \tag{1.5}$$

（2）平方损失函数（quadratic loss function）

$$L(Y,f(X))=(Y-f(X))^2 \tag{1.6}$$

（3）绝对损失函数（absolute loss function）

$$L(Y,f(X))=|Y-f(X)| \tag{1.7}$$

（4）对数损失函数（logarithmic loss function）或对数似然损失函数（log-likelihood loss function）

$$L(Y,P(Y|X))=-\log P(Y|X) \tag{1.8}$$

2. 经验风险最小化和结构风险最小化
    1. empirical risk minimization，ERM：其理论依据是大数定理。但是通常情况下训练数据较少并不满足大数定理的要求，容易发生过拟合现象。
    2. structural risk minimization，SRM：为了防止过拟合现象，SRM增加正则化项，对模型的复杂度进行约束，要求模型复杂度较小。

### 1.3 算法

算法是指学习模型的具体算法，例如BP算法、EM算法等。

### 1.4 模型评估与模型选择

训练误差、测试误差、交叉验证。

生成模型、判别模型。

# 2 感知机模型

## 2.1 模型

$$f(x) = sign(wx + b) \tag{1.1}$$

$$sign(x) = \begin{cases} 1, & \text{if x >= 0} \\ -1, & \text{if x < 0} \end{cases} \tag{1.2}$$

其中所要训练的参数为w和b，感知机模型是一种简单的线性分类模型，属于判别模型。

## 2.2 策略

定义 loss function：

给定训练数据集

$$T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$$

其中，$x_i \in \mathcal{X} = \mathbf{R}^n$，$y_i \in \mathcal{Y} = \{+1, -1\}$，$i = 1, 2, \cdots, N$．感知机 $\text{sign}(w \cdot x + b)$ 学习的损失函数定义为

$$L(w, b) = -\sum_{x_i \in M} y_i (w \cdot x_i + b) \tag{2.4}$$

## 2.3 算法

SGD算法：

输入：训练数据集 $T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$，其中 $x_i \in \mathcal{X} = \mathbf{R}^n$，$y_i \in \mathcal{Y} = \{-1, +1\}$，$i = 1, 2, \cdots, N$；学习率 $\eta\ (0 < \eta \leqslant 1)$；

输出：$w, b$；感知机模型 $f(x) = \text{sign}(w \cdot x + b)$.

（1）选取初值 $w_0, b_0$

（2）在训练集中选取数据 $(x_i, y_i)$

（3）如果 $y_i (w \cdot x_i + b) \leqslant 0$

$$w \leftarrow w + \eta y_i x_i$$

$$b \leftarrow b + \eta y_i$$

（4）转至（2），直至训练集中没有误分类点. ■

3 KNN模型

## 3.1 模型

输入：训练数据集

$$T=\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\}$$

其中，$x_i\in\mathcal{X}\subseteq\mathbf{R}^n$ 为实例的特征向量，$y_i\in\mathcal{Y}=\{c_1,c_2,\cdots,c_K\}$ 为实例的类别，$i=1,2,\cdots,N$；实例特征向量 $x$；

输出：实例 $x$ 所属的类 $y$.

（1）根据给定的距离度量，在训练集 $T$ 中找出与 $x$ 最邻近的 $k$ 个点，涵盖这 $k$ 个点的 $x$ 的邻域记作 $N_k(x)$；

（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 $y$：

$$y=\arg\max_{c_j}\sum_{x_i\in N_k(x)}I(y_i=c_j),\quad i=1,2,\cdots,N;\quad j=1,2,\cdots,K \tag{3.1}$$

式 (3.1) 中，$I$ 为指示函数，即当 $y_i=c_j$ 时 $I$ 为 1，否则 $I$ 为 0. ■

## 3.2 策略

KNN模型是一个只需正向统计的过程，没有待训练参数，也不需要定义 loss function 。但是在统计前要决定策略三要素：距离度量方法、k值选择和分类决策方法。

1. 距离度量方法： 欧式距离、$L_p$距离、曼哈顿距离等。
2. k值选择：k值越小对临近数据点越敏感，模型越复杂，越容易发生过拟合；k值越大，模型越简单，不易发生过拟合，但是模型能力若，预测能力差。
3. 分类决策方法：多数表决，平均值方法等。

## 3.3 算法

最简单的算法就是线性搜索所有数据集，找出K个最近邻。其搜索复杂度为O(n)

优化的算法如**kd树算法**，搜索复杂度为 O(log n)

1. 构造kd树

输入：$k$ 维空间数据集 $T=\{x_1,x_2,\cdots,x_N\}$，

其中 $x_i=(x_i^{(1)},x_i^{(2)},\cdots,x_i^{(k)})^{\mathrm{T}}$，$i=1,2,\cdots,N$；

输出：$kd$ 树.

（1）开始：构造根结点，根结点对应于包含 $T$ 的 $k$ 维空间的超矩形区域.

选择 $x^{(1)}$ 为坐标轴，以 $T$ 中所有实例的 $x^{(1)}$ 坐标的中位数为切分点，将根结点对应的超矩形区域切分为两个子区域. 切分由通过切分点并与坐标轴 $x^{(1)}$ 垂直的超平面实现.

由根结点生成深度为 1 的左、右子结点：左子结点对应坐标 $x^{(1)}$ 小于切分点的子区域，右子结点对应于坐标 $x^{(1)}$ 大于切分点的子区域.

将落在切分超平面上的实例点保存在根结点.

（2）重复：对深度为 $j$ 的结点，选择 $x^{(l)}$ 为切分的坐标轴，$l=j(\mathrm{mod}\,k)+1$，以该结点的区域中所有实例的 $x^{(l)}$ 坐标的中位数为切分点，将该结点对应的超矩形区域切分为两个子区域. 切分由通过切分点并与坐标轴 $x^{(l)}$ 垂直的超平面实现.

由该结点生成深度为 $j+1$ 的左、右子结点：左子结点对应坐标 $x^{(l)}$ 小于切分点的子区域，右子结点对应坐标 $x^{(l)}$ 大于切分点的子区域.

将落在切分超平面上的实例点保存在该结点.

（3）直到两个子区域没有实例存在时停止. 从而形成 *kd* 树的区域划分. ■

2. 利用kd树搜索最近邻

输入：已构造的 *kd* 树；目标点 $x$；

输出：$x$ 的最近邻.

（1）在 *kd* 树中找出包含目标点 $x$ 的叶结点：从根结点出发，递归地向下访问 *kd* 树. 若目标点 $x$ 当前维的坐标小于切分点的坐标，则移动到左子结点，否则移动到右子结点. 直到子结点为叶结点为止.

（2）以此叶结点为“当前最近点”.

（3）递归地向上回退，在每个结点进行以下操作：

（a）如果该结点保存的实例点比当前最近点距离目标点更近，则以该实例点为“当前最近点”.

（b）当前最近点一定存在于该结点一个子结点对应的区域. 检查该子结点的父结点的另一子结点对应的区域是否有更近的点. 具体地，检查另一子结点对应的区域是否与以目标点为球心、以目标点与“当前最近点”间的距离为半径的超球体相交.

如果相交，可能在另一个子结点对应的区域内存在距目标点更近的点，移动到另一个子结点. 接着，递归地进行最近邻搜索；

如果不相交，向上回退.

（4）当回退到根结点时，搜索结束. 最后的“当前最近点”即为 $x$ 的最近邻点. ■

# 4 朴素贝叶斯方法

## 4.1 模型

朴素贝叶斯法通过训练数据来学习联合概率分布P(X,Y)。根据$P(X,Y)=P(X|Y)P(Y)$，所以如下图所示，我们可以通过学习Y的先验概率分布和X的条件概率分布来确定X与Y的联合分布。

$$P(Y=c_k),\quad k=1,2,\cdots,K \tag{4.1}$$

条件概率分布

$$P(X=x|Y=c_k)=P(X^{(1)}=x^{(1)},\cdots,X^{(n)}=x^{(n)}|Y=c_k),\quad k=1,2,\cdots,K \tag{4.2}$$

于是学习到联合概率分布 $P(X,Y)$.

之后根据前面所说的大数定理，训练数据满足这种分布，之后的新数据也当满足这种分布。至于训练数据不足的问题，也可套用结构风险最小的理论。

但是求解P(X|Y)的发杂度非常高，设X为n维向量，$x_j$有$S_j$个不同的取值，Y有K个不同的取值。那么统计P(X|Y)的复杂度为$K\Pi_{j=1}^{n}Sj$.

为降低复杂度，朴素贝叶斯法做了X的每个维度相互独立的假设，那么条件概率分布变为：

$$\begin{aligned}P(X=x|Y=y)&=P(X^1=x^1,\ldots,X^n=x^n|Y=c_k)\\&=\Pi_{j=1}^{n}P(X^j=x^j|Y=c_k)\\&k=1,2,\ldots,K\end{aligned} \tag{4.3}$$

这样使得模型变得简单，可计算。但是其效果就要差一点。

## 4.2 策略

朴素贝叶斯的策略被称为后验概率最大化策略，后面会说明这一策略和经验风险最小化策略是等价的。

1. 后验概率最大化策略

朴素贝叶斯法分类时，对给定的输入 $x$，通过学习到的模型计算后验概率分布 $P(Y=c_k|X=x)$，将后验概率最大的类作为 $x$ 的类输出．后验概率计算根据贝叶斯定理进行：

$$P(Y=c_k|X=x)=\frac{P(X=x|Y=c_k)P(Y=c_k)}{\sum_k P(X=x|Y=c_k)P(Y=c_k)} \tag{4.4}$$

将式 (4.3) 代入式 (4.4) 有

$$P(Y=c_k|X=x)=\frac{P(Y=c_k)\prod_j P(X^{(j)}=x^{(j)}|Y=c_k)}{\sum_k P(Y=c_k)\prod_j P(X^{(j)}=x^{(j)}|Y=c_k)},\quad k=1,2,\cdots,K \tag{4.5}$$

这是朴素贝叶斯法分类的基本公式．于是，朴素贝叶斯分类器可表示为

$$y=f(x)=\arg\max_{c_k}\frac{P(Y=c_k)\prod_j P(X^{(j)}=x^{(j)}|Y=c_k)}{\sum_k P(Y=c_k)\prod_j P(X^{(j)}=x^{(j)}|Y=c_k)} \tag{4.6}$$

注意到，在式 (4.6) 中分母对所有 $c_k$ 都是相同的，所以，

$$y=\arg\max_{c_k}P(Y=c_k)\prod_j P(X^{(j)}=x^{(j)}|Y=c_k) \tag{4.7}$$

2. 后验概率最大等价于经验风险最小

为了证明这个问题，首先定义0-1损失函数如下：

$$L(Y,F(X))=\begin{cases}1, & Y\neq f(X)\\ -1, & Y=f(X)\end{cases}$$

式中f(X)式分类决策函数. 这时，期望风险函数为

$$R_{exp}=E[L(Y,f(X))]$$

然后利用条件期望全期望公式得：

$$R_{exp}(f)=E_x\sum_{k=1}^{K}[L(c_k,f(X))]P(c_k|X)$$

因为P(X=x)是确定的，所以只需对X=x的情况下逐个取最小化，如下：

$$\begin{aligned}f(x)&=argmin_{y\in Y}\sum_{k=1}^{K}L(c_k,y)P(c_kX=x)\\&=argmin_{y\in Y}\sum_{k=1}^{K}P(y\neq c_k|X=x)\\&=argmin(1-P(y=c_k|X=x))\\&=argmax_{y\in Y}P(y=c_k|X=x)\end{aligned}$$

这样经验风险最小化就和前面的后验概率最大化策略目标相同了

## 4.3 算法

**算法 4.1（朴素贝叶斯算法（naïve Bayes algorithm））**

输入：训练数据 $T=\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\}$，其中 $x_i=(x_i^{(1)},x_i^{(2)},\cdots,x_i^{(n)})^{\mathrm{T}}$，$x_i^{(j)}$ 是第 $i$ 个样本的第 $j$ 个特征，$x_i^{(j)}\in\{a_{j1},a_{j2},\cdots,a_{jS_j}\}$，$a_{jl}$ 是第 $j$ 个特征可能取的第 $l$ 个值，$j=1,2,\cdots,n$，$l=1,2,\cdots,S_j$，$y_i\in\{c_1,c_2,\cdots,c_K\}$；实例 $x$；

输出：实例 $x$ 的分类.

（1）计算先验概率及条件概率

$$P(Y=c_k)=\frac{\sum_{i=1}^{N}I(y_i=c_k)}{N},\quad k=1,2,\cdots,K$$

$$P(X^{(j)}=a_{jl}\mid Y=c_k)=\frac{\sum_{i=1}^{N}I(x_i^{(j)}=a_{jl},y_i=c_k)}{\sum_{i=1}^{N}I(y_i=c_k)}$$

$$j=1,2,\cdots,n;\quad l=1,2,\cdots,S_j;\quad k=1,2,\cdots,K$$

（2）对于给定的实例 $x=(x^{(1)},x^{(2)},\cdots,x^{(n)})^{\mathrm{T}}$，计算

$$P(Y=c_k)\prod_{j=1}^{n}P(X^{(j)}=x^{(j)}\mid Y=c_k),\quad k=1,2,\cdots,K$$

（3）确定实例 $x$ 的类

$$y=\arg\max_{c_k}P(Y=c_k)\prod_{j=1}^{n}P(X^{(j)}=x^{(j)}\mid Y=c_k)$$

■

# 5 决策树模型

## 5.1 模型

**定义 5.1（决策树）** 分类决策树模型是一种描述对实例进行分类的树形结构．决策树由结点（node）和有向边（directed edge）组成．结点有两种类型：内部结点（internal node）和叶结点（leaf node）．内部结点表示一个特征或属性，叶结点表示一个类．

用决策树分类，从根结点开始，对实例的某一特征进行测试，根据测试结果，将实例分配到其子结点；这时，每一个子结点对应着该特征的一个取值．如此递归地对实例进行测试并分配，直至达到叶结点．最后将实例分到叶结点的类中．

图 5.1 是一个决策树的示意图．图中圆和方框分别表示内部结点和叶结点．

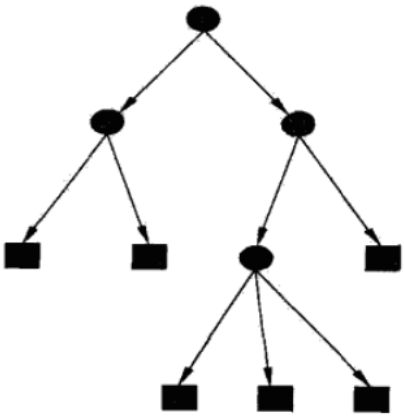

图 5.1　决策树模型

## 5.2 策略

决策树的学习策略是损失函数最小的策略，但是这一策略在学习算法中不会明显的体现出来。具体的学习算法只会要求对于训练数据，分类尽可能正确。这也就蕴含了损失函数最小的思想。

即是损失函数已经最小了，决策树学习还要求树的结构最优，即树的层数要尽量少。

## 5.3 算法

从所有的二叉树中找出结构最优的的树是NP难度的问题。所以具体的算法都是启发式的算法，从根节点开始先找到分类最优的分类特征，然后一次递归地执行下去。

常用的算法有ID3，C4.5 与 CART。这些算法基本都基于信息熵和信息增益的理论。

1. 信息熵