第5章 朴素贝叶斯分类算法

5.1 基本概念

5.1.1 主观概率

贝叶斯方法是一种研究不确定性的推理方法，不确定性常用贝叶斯概率表示，它是一种主观概率。通常的经典概率代表事件的物理特性，是不随人意识变化的客观存在，而贝叶斯概率则是人的认识，是个人主观的估计，随个人的主观认识的变化而变化。投掷硬币可能出现的正反面两种情形，经典概率代表硬币正面朝上的概率，这是一个客观存在；而贝叶斯概率则指个人相信硬币会正面朝上的程度。同样的例子还有，一个企业家认为“一项新产品在未来市场上销售”的概率是0.8，这里的0.8是根据他多年的经验和当时的一些市场信息综合而成的个人观点。一个投资者认为“购买某种股票能获得高收益”的概率是0.6，这里的0.6是投资者根据自己多年股票生意经验和当时股票行情综合而成的个人信念。

贝叶斯概率是主观的，对它的估计取决于先验知识的正确和后验知识的丰富和准确，因此贝叶斯概率常常可能随个人掌握信息的不同而发生变化。对即将进行的羽毛球单打比赛结果进行预测，不同人对胜负的主观预测不同，如果对两人的情况和各种现场的分析一无所知，就会认为两者的胜负比例为1∶1；如果知道其中一人是奥运会羽毛球单打冠军，而另一人只是某省队新队员，则可能给出的预测是奥运会冠军和省队队员的胜负比例为3∶1；如果进一步知道奥运冠军刚好在前一场比赛中受过伤，则对他们胜负比例的主观预测可能会下调为2∶1。所有的预测推断都是主观的，基于后验知识的一种判断，取决于对各种信息的掌握。

经典概率方法强调客观存在，它认为不确定性是客观存在的。在同样的羽毛球单打比赛预测中，从经典概率的角度看，如果认为胜负比例为1∶1，则意味着在相同条件下，如果两人进行100场比赛，其中一人可能会取得50场的胜利，同时输掉另外50场。主观概率不像经典概率那样强调多次的重复，因此在许多不可能出现重复事件的场合能得到很好的应用。上面提到的企业家对未来产品的预测，投资者对股票是否能取得高收益的预测以及羽毛球比赛胜负的预测中，都不可能进行重复的试验，因此，利用主观概率，按照个人对事件的相信程度而对事件做出推断是一种很合理而易于解释的方法。

5.1.2 贝叶斯定理

1. 基础知识

(1) 已知事件 A 发生的条件下,事件 B 发生的概率,称为事件 B 在事件 A 发生下的条件概率,记为 $P(B|A)$,其中 $P(A)$称为先验概率,$P(B|A)$称为后验概率,计算条件概率的公式如下。

$$P(B \mid A)=\frac{P(A \cap B)}{P(A)} \tag{5-1}$$

条件概率公式通过变形得到如下乘法公式。

$$P(A \cap B)=P(B \mid A)P(A) \tag{5-2}$$

(2) 设 A、B 为两个随机事件,如果有 $P(AB)=P(A)P(B)$成立,则称事件 A 和 B 相互独立。此时有 $P(A|B)=P(A)$,$P(AB)=P(A)P(B)$成立。

设 $A_1,A_2,\cdots,A_n$ 为 n 个随机事件,如果对其中任意 $m(2\leqslant m\leqslant n)$个事件 $A_{k_1},A_{k_2},\cdots,A_{k_m}$,都有

$$P(A_{k_1},A_{k_2},\cdots,A_{k_m})=P(A_{k_1})P(A_{k_2})\cdots P(A_{k_m}) \tag{5-3}$$

成立,则称事件 $A_1,A_2,\cdots,A_n$ 相互独立。

(3) 设 $B_1,B_2,\cdots,B_n$ 为互不相容事件,$P(B_i)>0,i=1,2,\cdots,n$,且 $\bigcup_{i=1}^{n}B_i=\Omega$ 为基本空间,对任意的事件 $A\subset\bigcup_{i=1}^{n}B_i$,计算事件 A 发生的概率的公式为

$$P(A)=\sum_{i=1}^{n}P(B_i)P(A \mid B_i) \tag{5-4}$$

设 $B_1,B_2,\cdots,B_n$ 为互不相容事件,$P(B_i)>0,i=1,2,\cdots,n,P(A)>0$,则在事件 A 发生的条件下,事件 B_i 发生的概率为

$$P(B_i \mid A)=\frac{P(B_iA)}{P(A)}=\frac{P(B_i)P(A \mid B_i)}{\sum_{i=1}^{n}P(B_i)P(A \mid B_i)} \tag{5-5}$$

称该公式为贝叶斯公式。

2. 贝叶斯决策准则

假设 $\Omega=\{C_1,C_2,\cdots,C_m\}$是有 m 个不同类别的集合,特征向量 $\boldsymbol{X}$ 是 d 维向量,$P(X|C_i)$是特征向量 $\boldsymbol{X}$ 在类别 C_i 状态下的条件概率,$P(C_i)$为类别 C_i 的先验概率。根据前面所述的贝叶斯公式,后验概率 $P(C_i|X)$的计算公式为

$$P(C_i \mid X)=\frac{P(X \mid C_i)}{P(X)}P(C_i) \tag{5-6}$$

其中 $P(X)=\sum_{j=1}^{m}P(X|C_j)P(C_j)$。

贝叶斯决策准则为:如果对于任意 $i\neq j$,都有 $P(C_i|X)>P(C_j|X)$成立,则样本模式 $\boldsymbol{X}$ 被判定为类别 C_i。

3. 极大后验假设

根据贝叶斯公式可得到一种计算后验概率的方法:在一定假设的条件下,根据先验概

率和样本数据得到的概率，可以得到后验概率。

令 $P(c)$ 是假设 c 的先验概率，它表示 c 是正确假设的概率，$P(X)$ 表示的是训练样本 X 的先验概率，$P(X|c)$ 表示在假设 c 正确的条件下样本 X 发生或出现的概率，根据贝叶斯公式可以得到后验概率的计算公式为

$$P(c \mid X)=\frac{P(X \mid c)P(c)}{P(X)} \tag{5-7}$$

设 C 为类别集合也就是待选假设集合，在给定未知类别标号样本 X 时，通过计算找到可能性最大的假设 $c \in C$，具有最大可能性的假设或类别被称为极大后验假设(Maximum a Posteriori)，记作 c_{map}：

$$c_{\text{map}}=\arg\max_{c \in C} P(c \mid X)=\arg\max_{c \in C} \frac{P(X \mid c)P(c)}{P(X)} \tag{5-8}$$

由于 $P(X)$ 与假设 c 无关，上式可变为：

$$c_{\text{map}}=\arg\max_{c \in C} P(X \mid c)P(c) \tag{5-9}$$

当没有给定类别概率的情形下，可做一个简单假定：假设 C 中每个假设都有相等的先验概率，也就是对于任意的 $c_i, c_j \in C(i \neq j)$，有 $P(c_i)=P(c_j)$，再做进一步简化，只需计算 $P(X|c)$ 找到使之达到最大的假设。$P(X|c)$ 被称为极大似然假设(Maximum Likelihood)，记为 c_{ml}：

$$c_{\text{ml}}=\arg\max_{c \in C} P(X \mid c) \tag{5-10}$$

5.1.3 朴素贝叶斯分类模型

在贝叶斯分类器的诸多算法中，朴素贝叶斯分类模型是最早出现的，它的算法逻辑简单，构造的朴素贝叶斯分类模型结构也比较简单，运算速度比同类算法快很多，分类所需的时间也比较短，并且大多数情况下分类精度也比较高，因而在实际中得到了广泛的应用。该分类器有一个朴素的假定：以属性的类条件独立性假设为前提，即在给定类别状态条件下，属性之间是相互独立的。朴素贝叶斯分类器的结构示意图如图 5-1 所示。

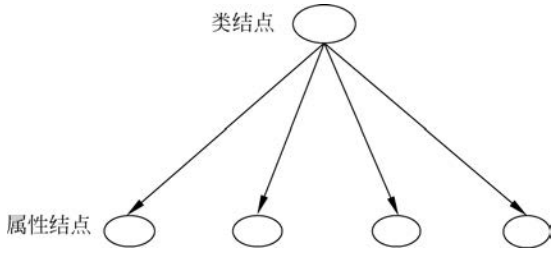

图 5-1 朴素贝叶斯分类器的结构示意图

假设样本空间有 m 个类别 $\{C_1, C_2, \cdots, C_m\}$，数据集有 n 个属性 $A_1, A_2, \cdots, A_n$，给定一个未知类别的样本 $X=(x_1, x_2, \cdots, x_n)$，其中 x_i 表示第 i 个属性的取值，即 $x_i \in A_i$，则可用贝叶斯公式计算样本 $X=(x_1, x_2, \cdots, x_n)$ 属于类别 $C_k(1 \leqslant k \leqslant m)$ 的概率。根据贝叶斯公式，$P(C_k|X)=\dfrac{P(C_k)P(X|C_k)}{P(X)} \propto P(C_k)P(X|C_k)$，即要得到 $P(C_k|X)$ 的值，关键要计算 $P(X|C_k)$ 和 $P(C_k)$。令 $C(X)$ 为 X 所属的类别标签，由贝叶斯分类准则，如果对于任意

$i \neq j$,都有 $P(C_i|X)>P(C_j|X)$成立,则把未知类别的样本 X 指派给类别 C_i,贝叶斯分类器的计算模型如下。

$$C(X)=\arg\ \max P(C_i)P(X \mid C_i) \tag{5-11}$$

由朴素贝叶斯分类器的属性独立性假设,假设各属性 $x_i(i=1,2,\cdots,n)$间相互条件独立,则

$$P(X \mid C_i)=\prod_{k=1}^{n} P(x_k \mid C_i) \tag{5-12}$$

于是式(5-11)被修改为

$$C(X)=\arg\ \max_i P(C_i)\prod_{k=1}^{n} P(x_k \mid C_i) \tag{5-13}$$

$P(C_i)$为先验概率,可通过 $P(C_i)=d_i/d$ 计算得到,其中 d_i 是属于类别 C_i 的训练样本的个数,d 是训练样本的总数。若属性 A_k 是离散的,则概率可由 $P(x_k|C_i)=d_{ik}/d_i$ 计算得到,其中 d_{ik} 是训练样本集合中属于类 C_i 并且属性 A_k 取值为 x_k 的样本个数,d_i 是属于类 C_i 的训练样本的个数。朴素贝叶斯分类的工作过程如下。

(1) 用一个 n 维特征向量 $X=(x_1,x_2,\cdots,x_n)$来表示数据样本,描述样本 X 对 n 个属性 $A_1,A_2,\cdots,A_n$ 的度量。

(2) 假定样本空间有 m 个类别状态 $C_1,C_2,\cdots,C_m$,对于给定的一个未知类别标号的数据样本 X,分类算法将 X 判定为具有最高后验概率的类别,也就是说,朴素贝叶斯分类算法将未知类别的样本 X 分配给类别 C_i,当且仅当对于任意的 j,有 $P(C_i|X)>P(C_j|X)$成立,$1\leqslant j\leqslant m$,$j\neq i$,使 $P(C_i|X)$取得最大值的类别 C_i 被称为最大后验假定。

(3) 由于 $P(X)$不依赖于类别状态,对于所有类别都是常数,则根据贝叶斯定理,最大化 $P(C_i|X)$只需要最大化 $P(X|C_i)P(C_i)$即可。如果类的先验概率未知,则通常假设这些类别的概率是相等的,即 $P(C_1)=P(C_2)=\cdots=P(C_m)$,所以只需要最大化 $P(X|C_i)$即可,否则就要最大化 $P(X|C_i)P(C_i)$。其中可用 S_i/S 对 $P(C_i)$进行估计计算,S_i 是给定类别 C_i 中训练样本的个数,S 是训练样本(实例空间)的总数。

(4) 当实例空间中训练样本的属性较多时,计算 $P(X|C_i)$可能会比较费时,开销较大,此时可以做条件独立性的假定:在给定样本类别标号的条件下,假定属性值是相互条件独立的,属性之间不存在任何依赖关系,则下面等式成立

$$P(X \mid C_i)=\prod_{k=1}^{n} P(x_k \mid C_i)$$

其中概率 $P(x_1|C_i),P(x_2|C_i),\cdots,P(x_n|C_i)$的计算可由样本空间中的训练样本进行估计。实际问题中根据样本属性 A_k 的离散或连续性质,考虑下面两种情形。

- 如果属性 A_k 是连续的,则一般假定它服从正态分布,从而来计算类条件概率。
- 如果属性 A_k 是离散的,则 $P(x_k|C_i)=S_{ik}/S_i$,其中 S_{ik} 是在实例空间中类别为 C_i 的样本中属性 A_k 上取值为 x_k 的训练样本个数,而 S_i 是属于类别 C_i 的训练样本个数。

(5) 对于未知类别的样本 X,对每个类别 C_i 分别计算 $P(X|C_i)P(C_i)$。样本 X 被认为属于类别 C_i,当且仅当 $P(X|C_i)P(C_i)>P(X|C_j)P(C_j)$,$1\leqslant j\leqslant m$,$j\neq i$,也就是说样本 X 被指派到使 $P(X|C_i)P(C_i)$取得最大值的类别 C_i。

朴素贝叶斯分类模型的算法描述如下。

(1) 对训练样本数据集和测试样本数据集进行离散化处理和缺失值处理。

(2) 扫描训练样本数据集，分别统计训练集中类别为 C_i 的样本个数 d_i 和属于类别 C_i 的样本中属性 A_k 上取值为 x_k 的实例样本个数 d_{ik}，构成统计表。

(3) 计算先验概率 $P(C_i)=d_i/d$ 和条件概率 $P(A_k=x_k|C_i)=d_{ik}/d_i$，构成概率表。

(4) 构建分类模型 $C(X)=\arg\max_i P(C_i)P(X|C_i)$。

(5) 扫描待分类的样本数据集，调用已得到的统计表、概率表以及构建好的分类准则，得出分类结果。

5.1.4 朴素贝叶斯分类器实例分析

【例 5.1】 应用朴素贝叶斯分类器来解决这样一个分类问题：根据天气状况来判断某天是否适合于打网球。给定表 5-1 所示的 14 个训练实例，其中每一天由属性 Outlook、Temperature、Humidity 和 Wind 表征，类属性为 Play Tennis。

表 5-1 14 个训练实例

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no

现在有一个测试实例 x：< Outlook = sunny，Temperature = cool，Humidity = high，Wind = strong >，问这一天是否适合于打网球。显然，本题的任务就是要预测此新实例的类属性 Play Tennis 取值 (yes 或 no)，为此，需要构建如图 5-2 所示的朴素贝叶斯网络分类器。

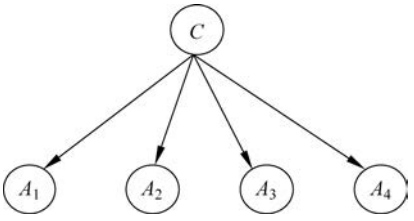

图 5-2 朴素贝叶斯分类器的结构

图中的类结点 C 表示类属性 Play Tennis，其他 4 个结点 A_1、A_2、A_3、A_4 分别代表 4 个属性 Outlook、Temperature、Humidity 和 Wind，类结点 C 是所有属性结点的父亲结点，属性结点和属性结点之间没有任何依赖关系。根据公式，有

$$V(x)=\arg\max_{c\in\{\text{yes,no}\}} P(c)P(\text{sunny}\mid c)P(\text{cool}\mid c)P(\text{high}\mid c)P(\text{strong}\mid c)$$

为了计算 $V(x)$,需要从如表 5-1 所示的 14 个训练实例中估计出概率。

$P(\text{yes})$,$P(\text{sunny}\mid\text{yes})$,$P(\text{cool}\mid\text{yes})$,$P(\text{high}\mid\text{yes})$,$P(\text{strong}\mid\text{yes})$,$P(\text{no})$,

$P(\text{sunny}\mid\text{no})$,$P(\text{cool}\mid\text{no})$,$P(\text{high}\mid\text{no})$,$P(\text{strong}\mid\text{no})$。

具体的计算如下。

$$P(\text{yes})=9/14$$
$$P(\text{sunny}\mid\text{yes})=2/9$$
$$P(\text{cool}\mid\text{yes})=3/9$$
$$P(\text{high}\mid\text{yes})=3/9$$
$$P(\text{strong}\mid\text{yes})=3/9$$
$$P(\text{no})=5/14$$
$$P(\text{sunny}\mid\text{no})=3/5$$
$$P(\text{cool}\mid\text{no})=1/5$$
$$P(\text{high}\mid\text{no})=4/5$$
$$P(\text{strong}\mid\text{no})=3/5$$

所以,有

$$P(\text{yes})P(\text{sunny}\mid\text{yes})P(\text{cool}\mid\text{yes})P(\text{high}\mid\text{yes})P(\text{strong}\mid\text{yes})=0.005\,291$$
$$P(\text{no})P(\text{sunny}\mid\text{no})P(\text{cool}\mid\text{no})P(\text{high}\mid\text{no})P(\text{strong}\mid\text{no})=0.020\,570\,4$$

可见,朴素贝叶斯分类器将此实例分类为 no。

【例 5.2】 应用朴素贝叶斯分类器来解决这样一个分类问题:给出一个商场顾客数据库(训练样本集合),判断某一顾客是否会买计算机。给定表 5-2 所示的 14 个训练实例,其中每个实例由属性 age、income、student、credit rating 表征,样本集合的类别属性为 buy computer,该属性有两个不同取值,即{yes,no},因此就有两个不同的类别($m=2$)。设 C_1 对应 yes 类别,C_2 对应 no 类别。

表 5-2 14 个训练实例

age	income	student	credit rating	buy computer
≤30	high	no	fair	no
≤30	high	no	excellent	no
30～40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31～40	low	yes	excellent	yes
≤30	medium	no	fair	no
≤30	low	yes	fair	yes
>40	medium	yes	fair	yes
≤30	medium	yes	excellent	yes
31～40	medium	no	excellent	yes
31～40	high	yes	fair	yes
>40	medium	no	excellent	no

现在有一个测试实例 x：(age <=30，Income = medium，Student = yes，Credit rating= fair)，问这一顾客是否会买计算机，本题的任务是要判断给定的测试实例是属于 C_1 还是 C_2，根据公式，有：

$$V(x)=\arg\max_{c\in\{\text{yes,no}\}}P(c)P(\text{age}\leqslant 30\mid c)P(\text{medium}\mid c)P(\text{yes}\mid c)P(\text{fair}\mid c)$$

为计算 $V(x)$，先计算每个类的先验概率 $P(C_i)$：

$$P(C_i)\text{：}P(\text{buy computer}=\text{'yes'})=9/14=0.643$$

$$P(\text{buy computer}=\text{'no'})=5/14=0.357$$

为计算 $P(X|C_i)$，$i=1,2$，计算下面的条件概率。

$P(\text{age}=\text{'}\leqslant 30\text{'}|\ \text{buy computer}=\text{'yes'})=2/9=0.222$

$P(\text{age}=\text{'}\leqslant 30\text{'}|\ \text{buy computer}=\text{'no'})=3/5=0.6$

$P(\text{income}=\text{'medium'}|\ \text{buy computer}=\text{'yes'})=4/9=0.444$

$P(\text{income}=\text{'medium'}|\ \text{buy computer}=\text{'no'})=2/5=0.4$

$P(\text{student}=\text{'yes'}|\ \text{buy computer}=\text{'yes'})=6/9=0.667$

$P(\text{student}=\text{'yes'}|\ \text{buy computer}=\text{'no'})=1/5=0.2$

$P(\text{credit rating}=\text{'fair'}|\ \text{buy computer}=\text{'yes'})=6/9=0.667$

$P(\text{credit rating}=\text{'fair'}|\ \text{buy computer}=\text{'no'})=2/5=0.4$

$X=(\text{age}\leqslant 30,\text{income}=\text{medium},\text{student}=\text{yes},\text{credit rating}=\text{fair})$

$P(X\mid C_i)\text{：}P(X\mid\text{buy computer}=\text{'yes'})=0.222\times 0.444\times 0.667\times 0.667=0.044$

$P(X\mid\text{buy computer}=\text{'no'})=0.6\times 0.4\times 0.2\times 0.4=0.019$

$P(X\mid C_i)*P(C_i)\text{：}P(X\mid\text{buy computer}=\text{'yes'})\cdot P(\text{buy computer}=\text{'yes'})=0.028$

$P(X\mid\text{buy computer}=\text{'no'})\cdot P(\text{buy computer}=\text{'no'})=0.007$

因此，对于样本 X，朴素贝叶斯分类预测结果为 buy computer= 'yes'。

5.2 朴素贝叶斯算法的特点及应用

5.2.1 朴素贝叶斯算法的特点

朴素贝叶斯分类算法有诸多优点：逻辑简单、易于实现、分类过程中算法的时间空间开销比较小；算法比较稳定、分类性能对于具有不同数据特点的数据集合来说，其差别不大，即具有比较好的健壮性等优点。

在实际情况中，尽管难以满足朴素贝叶斯模型的属性类条件独立性假定，但它分类预测在大多数情况下仍比较精确。原因有如下几个：要估计的参数比较少，从而加强了估计的稳定性；虽然概率估计是有偏差的，但人们大多关心的不是它的绝对值，而是它的排列次序，因此有偏差的概率估计在某些情况下可能并不要紧；现实中很多时候已经对数据进行了预处理，例如对变量进行了筛选，可能已经去掉了高度相关的量等。除了分类性能很好外，贝叶斯分类模型还具有形式简单、可扩展性强和可理解性好等优点。

朴素贝叶斯分类器的缺点是属性间类条件独立的这个假定，而很多实际问题中这个独立性假设并不成立，如果在属性间存在相关性的实际问题中忽视这一点，会导致分类效果下降。

朴素贝叶斯分类模型虽然在某些不满足独立性假设的情况下其分类效果比较好,但是大量研究表明可以通过各种改进方法来提高朴素贝叶斯分类器的性能。朴素贝叶斯分类器的改进方法主要有两类:一类是弱化属性的类条件独立性假设,在朴素贝叶斯分类器的基础上构建属性间的相关性,如构建相关性度量公式,增加属性间可能存在的依赖关系;另一类是构建新的样本属性集,期望在新的属性集中,属性间存在较好的类条件独立关系。

5.2.2 朴素贝叶斯算法的应用场景

朴素贝叶斯算法应用很广泛,下面举两个实例说明其应用情况。

1. 贝叶斯方法在中医证候和症状描述中的应用

中医证候和症状描述错综复杂,如何较好地对病患所属证候进行鉴别诊断,一直是临床医疗工作者的首要目标,把数据挖掘技术的朴素贝叶斯分类方法应用到中医证候的诊断识别中,是一个较好的尝试。在使用朴素贝叶斯分类方法对中医证候进行分类识别并用遗传算法改进时,经历了以下过程:首先合理抽象鉴别诊断过程并建立数学模型;其次,提出了使用数据挖掘技术中的朴素贝叶斯分类方法对模型求解;第三,考虑到特征数量较大,运用了遗传算法进行特征优化;最后,使用医学上常用的 ROC 曲线评价方法对改进前后的分类识别的效率进行分析比较。

2. 贝叶斯方法在玉米叶部病害图像识别中的应用

在图像分割和特征提取的基础上,利用朴素贝叶斯分类器的统计学习方法,可以实现玉米叶部病斑的分类识别。相关文献表明,贝叶斯分类器具有网络结构简单、易于扩展等特点,对玉米叶部病害的分类识别效果较好,也为其他作物病害图像识别的研究提供了借鉴。

5.3 朴素贝叶斯源代码结果分析

朴素贝叶斯源代码包括 JavaBean.java 和 TestBayes.java 两个文件,相关程序和实验数据可从 github 中下载。

1. JavaBean.java

```
package bayes;
public class JavaBean {
    int age;
    String income;
    String student;
    String credit_rating;
    String buys_computer;
    public JavaBean(int age, String income, String student,
            String credit_rating, String buys_computer) {
        this.age = age;
        this.income = income;
        this.student = student;
        this.credit_rating = credit_rating;
        this.buys_computer = buys_computer;
    }
```

```
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }
    public String getIncome() {
        return income;
    }
    public void setIncome(String income) {
        this.income = income;
    }
    public String getStudent() {
        return student;
    }
    public void setStudent(String student) {
        this.student = student;
    }
    public String getCredit_rating() {
        return credit_rating;
    }
    public void setCredit_rating(String credit_rating) {
        this.credit_rating = credit_rating;
    }
    public String getBuys_computer() {
        return buys_computer;
    }
    public void setBuys_computer(String buys_computer) {
        this.buys_computer = buys_computer;
    }
    @Override
    public String toString() {
        return "JavaBean [age=" + age + ", income=" + income + ", student="
                + student + ", credit_rating=" + credit_rating
                + ", buys_computer=" + buys_computer + "]";
    }
}
```

2. TestBayes.java

```
package bayes;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.util.ArrayList;
public class TestBayes {
    //朴素贝叶斯算法 算法的思想
    public static ArrayList<JavaBean> list = new ArrayList<JavaBean>();
    static int data_length = 0;
    public static void main(String[] args) {
```

```
        //1.读取数据,放入 list 容器中
        File file = new File("C:\\Users\\wang4\\Desktop\\beiyes.txt");
        txt2String(file);
        //数据测试样本
        testData(25, "Medium", "Yes", "Fair");
    }

    //读取样本数据
    public static void txt2String(File file) {
        try {
            BufferedReader br = new BufferedReader(new FileReader(file));
            // 构造一个 BufferedReader 类来读取文件
            String s = null;
            while ((s = br.readLine()) != null) {  //使用 readLine 方法,一次读一行
                data_length++;
                splitt(s);
            }
            br.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    //存入 ArrayList 中
    public static void splitt(String str) {
        String strr = str.trim();
        String[] abc = strr.split("[\\p{Space}]+");
        int age = Integer.parseInt(abc[0]);
        JavaBean bean = new JavaBean(age, abc[1], abc[2], abc[3], abc[4]);
        list.add(bean);
    }

    //训练样本,测试
    public static void testData(int age, String a, String b, String c) {
        //训练样本
        int number_yes = 0;
        int number_no = 0;
        //age 情况个数
        int num_age_yes = 0;
        int num_age_no = 0;
        //income
        int num_income_yes = 0;
        int num_income_no = 0;
        //student
        int num_student_yes = 0;
        int num_student_no = 0;
        //credit
        int num_credit_yes = 0;
        int num_credit_no = 0;

        //遍历 list 获得数据
        for (int i = 0; i < list.size(); i++) {
```

```
        JavaBean bb = list.get(i);
        if (bb.getBuys_computer().equals("Yes")) {      //Yes
            number_yes++;
            if (bb.getIncome().equals(a)) {             //income
                num_income_yes++;
            }
            if (bb.getStudent().equals(b)) {            //student
                num_student_yes++;
            }
            if (bb.getCredit_rating().equals(c)) {      //credit
                num_credit_yes++;
            }
            if (bb.getAge() == age) {                   //age
                num_age_yes++;
            }
        } else {                                        //No
            number_no++;
            if (bb.getIncome().equals(a)) {             //income
                num_income_no++;
            }
            if (bb.getStudent().equals(b)) {            //student
                num_stdent_no++;
            }
            if (bb.getCredit_rating().equals(c)) {      //credit
                num_credit_no++;
            }
            if (bb.getAge() == age) {                   //age
                num_age_no++;
            }

        }
    }
    System.out.println("购买的历史个数:" + number_yes);
    System.out.println("不买的历史个数:" + number_no);
    System.out.println("购买 + age:" + num_age_yes);
    System.out.println("不买 + age:" + num_age_no);
    System.out.println("购买 + income:" + num_income_yes);
    System.out.println("不买 + income:" + num_income_no);
    System.out.println("购买 + student:" + num_student_yes);
    System.out.println("不买 + student:" + num_student_no);
    System.out.println("购买 + credit:" + num_credit_yes);
    System.out.println("不买 + credit:" + num_credit_no);
    //概率判断
    double buy_yes = number_yes * 1.0 / data_length;   //买的概率
    double buy_no = number_no * 1.0 / data_length;     //不买的概率
    System.out.println("训练数据中买的概率:" + buy_yes);
    System.out.println("训练数据中不买的概率:" + buy_no);
    //未知用户的判断
    double nb_buy_yes = (1.0 * num_age_yes / number_yes)
            * (1.0 * num_income_yes / number_yes)
            * (1.0 * num_student_yes / number_yes)
```

```
                * (1.0 * num_credit_yes / number_yes) * buy_yes;
        double nb_buy_no = (1.0 * num_age_no / number_no)
                * (1.0 * num_income_no / number_no)
                * (1.0 * num_student_no / number_no)
                * (1.0 * num_credit_no / number_no) * buy_no;
        System.out.println("新用户买的概率:" + nb_buy_yes);
        System.out.println("新用户不买的概率:" + nb_buy_no);
        if (nb_buy_yes > nb_buy_no) {
            System.out.println("新用户买的概率大");
        } else {
            System.out.println("新用户不买的概率大");
        }
    }
}
```

程序运行界面如图 5-3 所示。

```
购买的历史个数:9
不买的历史个数:5
购买+age:2
不买+age:3
购买+income:4
不买+income:2
购买+stundent:6
不买+student:1
购买+credit:6
不买+credit:2
训练数据中买的概率:0.6428571428571429
训练数据中不买的概率:0.35714285714285715
新用户买的概率:0.028218694885361547
新用户不买的概率:0.006857142857142858
新用户买的概率大
```

图 5-3　朴素贝叶斯程序运行界面

上面的运行结果说明了我们应用朴素贝叶斯分类器来解决这样一个分类问题：25 岁的 student 根据工资水平的购买情况；如图 5-3 所示：年龄为 25 岁、收入中等的情况下，购买的历史个数为 9，不买的历史个数为 5，购买的学生数为 6，不买的学生数是 1，信誉度分别是购买的为 6，不买的为 2，统计出了综合训练数据中买的概率和不买的概率；也预测了一下新用户买的概率和不买的概率，并比较得出：新用户买的概率大。

5.4　基于阿里云数加平台的朴素贝叶斯实例

朴素贝叶斯分类是一种应用基于独立假设的贝叶斯定理的简单概率分类算法，它通过计算某一条数据假设为各个标签时的概率，选择其中概率最大的作为该条数据的标签。该算法具有简单、高效、分类效果稳定的优点；不足之处是：为了简化分类模型，而假定分类数据各个属性间是相互独立的，因此当各个属性之间的关联性不高时，朴素贝叶斯算法是一个很好的选择。总体操作思路与逻辑回归分类算法一样，这里选择网上的某一组数据，其训练数据与测试数据图如图 5-4 和图 5-5 所示，其中，y 为标签列，f0～f7 为非标签列。

用训练数据使用贝叶斯分类算法进行训练，然后对测试数据进行测试，可以得到其分类的准确率等信息，操作流程图如图 5-6 所示，左边为训练数据，右边为测试数据，通过训练数

id	y	f0	f1	f2	f3	f4	f5	f6	f7
1	-1	-0.294118	0.487437	0.180328	-0.292929	-1	0.00149028	-0.53117	-0.0333333
2	+1	-0.882353	-0.145729	0.0819672	-0.414141	-1	-0.207153	-0.766866	-0.666667
3	-1	-0.0588235	0.839196	0.0491803	-1	-1	-0.305514	-0.492741	-0.633333
4	+1	-0.882353	-0.105528	0.0819672	-0.535354	-0.777778	-0.162444	-0.923997	-1
5	-1	-1	0.376884	-0.344262	-0.292929	-0.602837	0.28465	0.887276	-0.6
6	+1	-0.411765	0.165829	0.213115	-1	-1	-0.23696	-0.894962	-0.7
7	-1	-0.647059	-0.21608	-0.180328	-0.353535	-0.791962	-0.0760059	-0.854825	-0.833333
8	+1	0.176471	0.155779	-1	-1	-1	0.052161	-0.952178	-0.733333
9	-1	-0.764706	0.979899	0.147541	-0.0909091	0.283688	-0.0909091	-0.931682	0.0666667
10	-1	-0.0588235	0.256281	0.57377	-1	-1	-1	-0.868488	0.1

图 5-4　训练数据图

id	y	f0	f1	f2	f3	f4	f5	f6	f7
1	+1	-0.882353	0.0854271	0.442623	-0.616162	-1	-0.19225	-0.725021	-0.9
2	+1	-0.294118	-0.0351759	-1	-1	-1	-0.293592	-0.904355	-0.766667
3	+1	-0.882353	0.246231	0.213115	-0.272727	-1	-0.171386	-0.981213	-0.7
4	-1	-0.176471	0.507538	0.278689	-0.414141	-0.702128	0.0491804	-0.475662	0.1
5	-1	-0.529412	0.839196	-1	-1	-1	-0.153502	-0.885568	-0.5
6	+1	-0.882353	0.246231	-0.0163934	-0.353535	-1	0.0670641	-0.627669	-1
7	-1	-0.882353	0.819095	0.278689	-0.151515	-0.307329	0.19225	0.00768574	-0.966667
8	+1	-0.882353	-0.0753769	0.0163934	-0.494949	-0.903073	-0.418778	-0.654996	-0.866667
9	+1	-1	0.527638	0.344262	-0.212121	-0.356974	0.23696	-0.836038	-0.8
10	+1	-0.882353	0.115578	0.0163934	-0.737374	-0.56974	-0.28465	-0.948762	-0.933333

图 5-5　测试数据图

据训练过后的朴素贝叶斯分类器，用来预测测试数据的标签，然后与测试数据的原标签进行对比。多分类评估与混淆矩阵类似，可以得到准确率及相关参数。

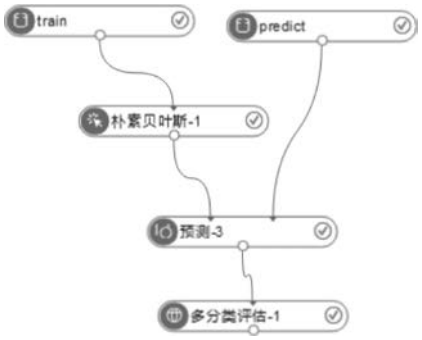

图 5-6　朴素贝叶斯算法流程图

朴素贝叶斯算法的字段设置如图 5-7 所示，其中特征列选择的是训练数据中的 f0～f7 列，排除列用于反选特征列，不可与特征列并存，强制转换列的默认解析规则为：string、

boolean、datetime 类型的列解析为离散类型；double、bigint 类型的列解析为连续类型；若有将 bigint 解析为 categorical 的情况，通过参数 forceCategorical 指定。

图 5-7　朴素贝叶斯字段设置图

预测组件的实验结果如图 5-8 所示，可以看到只有倒数第二条预测的标签有误，其他均正确。

y	prediction_result	prediction_score	prediction_detail
+1	+1	1.1253341740304...	{"+1": 1.125334174030447, "-1": -18.26887905024316}
+1	+1	2.1096017126455...	{"+1": 2.109601712645577, "-1": -22.86762826457826}
+1	+1	1.0119555555558...	{"+1": 1.011955555555831, "-1": -17.12262756256134}
-1	-1	-16.49991472469...	{"+1": -33.18118539397123, "-1": -16.49991472469541}
-1	-1	-61291274633.67...	{"+1": -688535196307.7689, "-1": -61291274633.67934}
+1	+1	-4.842928791659...	{"+1": -4.842928791659737, "-1": -17.8490897549666}
-1	-1	-18.28658192739...	{"+1": -89.31454432308786, "-1": -18.28658192739197}
+1	+1	-2.701748842959...	{"+1": -2.701748842959141, "-1": -17.85795905226513}
+1	-1	-18.21899791703...	{"+1": -20.76204601810873, "-1": -18.21899791703504}
+1	+1	-3.088051422948...	{"+1": -3.088051422948976, "-1": -17.70615005830097}

图 5-8　预测组件的实验结果

多分类评估组件的实验结果如图 5-9 所示，可以得到准确率及其他参数。

模型	TruePositive	TrueNegative	FalsePositive	FalseNegative	Sensitivity	Specificity	Precision	Accuracy	F1	Kappa
+1	6	3	0	1	0.857142...	1	1	0.9	0.9...	0.78...
-1	3	6	1	0	1	0.857142...	0.75	0.9	0.8...	0.78...

图 5-9　实验结果图

5.5　小　　结

贝叶斯方法是一种研究不确定性的推理方法，不确定性常用贝叶斯概率表示，它是一种主观概率。朴素贝叶斯分类算法有诸多优点：逻辑简单、易于实现、分类过程中算法的时间空间开销比较小；算法比较稳定；分类性能对于具有不同数据特点的数据集合来说，其差

别不大，即具有比较好的健壮性等优点。

思 考 题

1. 简述朴素贝叶斯分类的工作过程。

2. 表 5-3 是购买汽车的顾客分类训练样本集。假设顾客的属性集家庭经济状况、信用级别和月收入之间条件独立，则对于某顾客(测试样本)，已知其属性集 $X=<$一般，优秀，12$>$，利用朴素贝叶斯分类器计算这位顾客购买汽车的概率。

表 5-3 购买汽车的顾客训练样本集

序 号	家庭经济状况	信 用 级 别	月收入/千元	购 买 汽 车
1	一般	优秀	10	是
2	好	优秀	12	是
3	一般	优秀	6	是
4	一般	良好	8.5	否
5	一般	良好	9	否
6	一般	优秀	7.5	是
7	好	一般	22	是
8	一般	一般	9.5	否
9	一般	良好	7	是
10	好	良好	12.5	是