第八章 分类：基本概念

笔记本： 数据挖掘：概念与技术
创建时间： 2017/12/20 15:44 **更新时间：** 2017/12/27 17:39
作者： Passero

8.1基本概念

什么是分类
e.g.医学研究人员希望分析乳腺癌数据，以此来预测病人应当接受三种具体治疗方案中的哪一种。

分类的一般方法
数据分类是一个两阶段过程，包括学习阶段（构建分类模型）和分类阶段（使用模型预测给定数据的类标号）
图示：

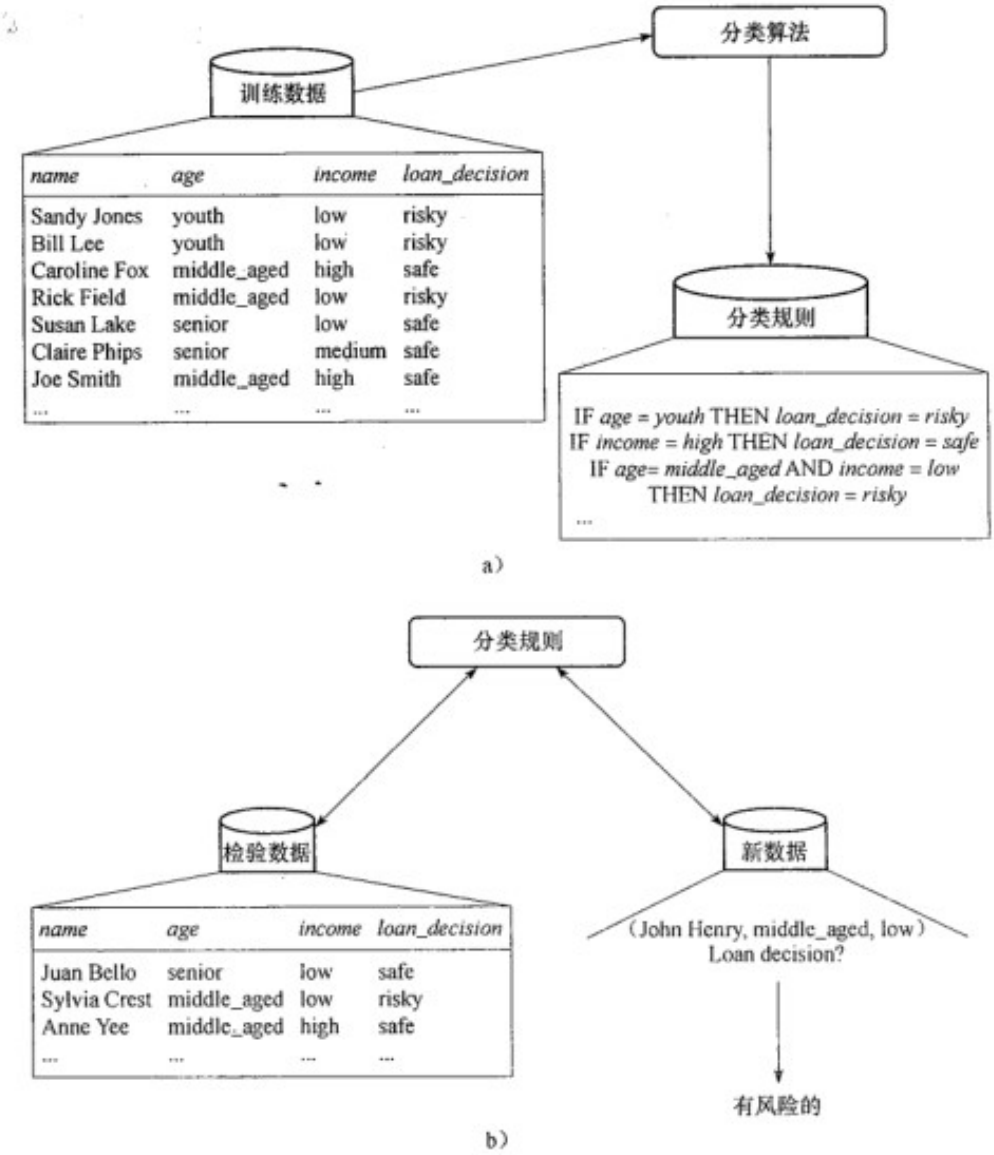

图 8.1 数据分类过程：a）学习：用分类算法分析训练数据，这里，类标号属性是 *loan_decision*，学习的模型或分类器以分类规则形式提供；b）分类：检验数据用于评估分类规则的准确率，如果准确率是可以接受的，则规则用于新的数据元组分类

8.2决策树归纳

决策树归纳是从有类标号的训练元组中学习决策树。决策树是一种类似于流程图的树结构，其中，每个内部结点（非树叶节点）表示在一个属性上的测试，每个分枝代表该测试的一个输出，而每个树叶节点（或终端节点）存放一个类标号。树的最顶层结点是根节点。内部结点用矩形表示，而叶节点用椭圆表示。

e.g.

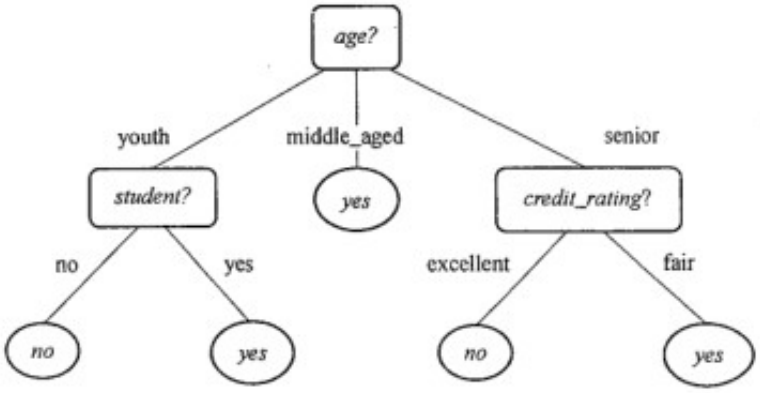

图 8.2 概念 *buys_computer* 的决策树，指出 AllElectronics 的顾客是否可能购买计算机。每个内部（非树叶）结点表示一个属性上的测试，每个树叶结点代表一个类（*buys_computer* = *yes*，或 *buys_computer* = *no*）

决策树归纳

算法：Generate_decision_tree。由数据分区*D*中的训练元组产生决策树。

输入：

- 数据分区*D*，训练元组和它们对应类标号的集合。
- *attribute_list*，候选属性的集合。
- *Attribute_selection_method*，一个确定“最好地”划分数据元组为个体类的分裂准则的过程。这个准则由分裂属性（*splitting_attribute*）和分裂点或划分子集组成。

输出：一棵决策树。

方法：

```
(1)  创建一个结点N;
(2)  if D中的元组都在同一类C中then
(3)    返回N作为叶结点,以类C标记;
(4) if attribut_list为空then
(5)    返回N作为叶结点，标记为D中的多数类;                    //多数表决
(6) 使用Attribute_selection_method(D, attribute_list),找出“最好的”splitting_criterion;
(7) 用splitting_criterion标记结点N;
(8)  if splitting_attribute是离散值的,并且允许多路划分then      //不限于二叉树
(9)     attribute_list←attribute_list-splitting_attribute; // 删除分裂属性
(10) for splitting_criterion的每个输出j
     //划分元组并对每个分区产生子树
(11)    设Dj是D中满足输出j的数据元组的集合;                   // 一个分区
(12)    if Dj为空then
(13)     加一个树叶到结点N,标记为D中的多数类;
(14)    else加一个由Generate_decision_tree(Dj,attribute_list)返回的结点到N;
     endfor
(15) 返回N;
```

e.g.

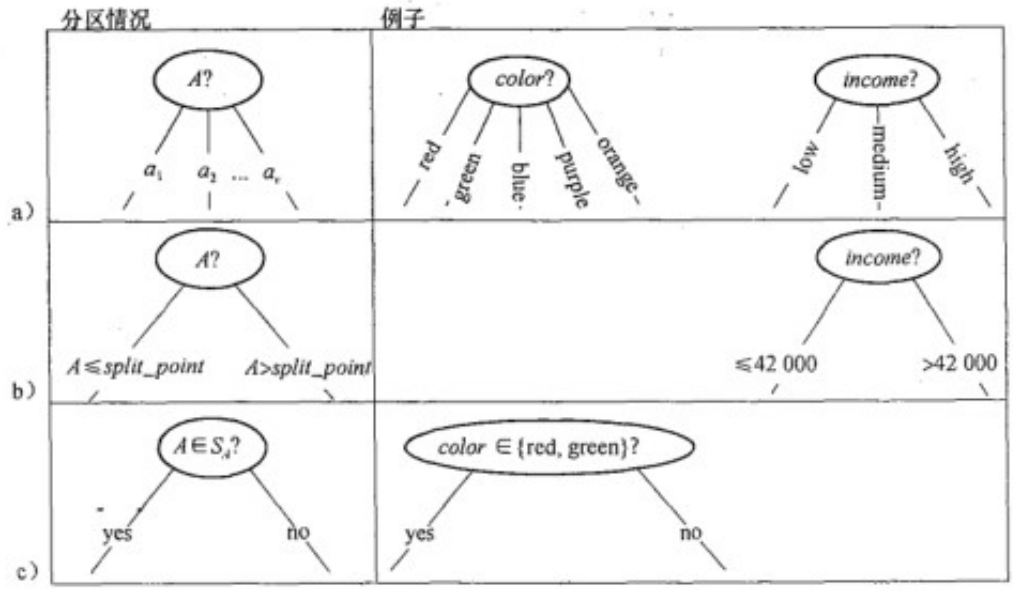

图 8.4 根据分裂准则划分元组的三种可能性，每个都给出了例子。设 A 是分裂属性：a）如果 A 是离散值的，则对 A 的每个已知值产生一个分枝；b）如果 A 是连续值的，则产生两个分枝，分别对应于 $A \leqslant split\_point$ 和 $A > split\_point$；c）如果 A 是离散值的，并且必须产生二叉树，则测试形如 $A \in S_A$，其中 S_A 是 A 的分裂子集

属性选择度量

definition：设数据分区D为标记类元组的训练集。假定类标号属性具有m个不同值，定义了m个不同的类Ci（i = 1, ... , m）。设Ci,D是D中Ci类元组的集合，|D|和|Ci,D|分别是D和Ci,D中元组的个数。

---信息增益（偏向于多值属性）

D中的元组分类所需要的期望信息由下列公式可以求得：

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

其中，p_i 是 D 中任意元组属于类 C_i 的非零概率，并用 $|C_{i,D}|/|D|$ 估计。使用以 2 为底的对数函数是因为信息用二进位编码。$Info(D)$ 是识别 D 中元组的类标号所需要的平均信息量。注意，此时我们所有的信息只是每个类的元组所占的百分比。$Info(D)$ 又称为 D 的熵（entropy）。

用属性A将D划分为v个分区或子集{D1，D2，... , Dv}，其中Dj包含D中的元组，它们的A值为aj。

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

项 $\frac{|D_j|}{|D|}$ 充当第 j 个分区的权重。$Info_A(D)$ 是基于按 A 划分对 D 的元组分类所需要的期望信息。需要的期望信息越小，分区的纯度越高。

信息增益定义为：

$$Gain(A) = Info(D) - Info_A(D)$$

---增益率（倾向于产生不平衡的划分）

分裂信息定义为：

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

增益率定义为：GainRate(A) = Gain(A) / SplitInfoA(D)

---基尼指数（偏向于多值属性，导致相等大小的分区和纯度）

$$Gini(D) = 1 - \sum_{i=1}^{m} p_i^2$$

基尼指数度量数据分区或训练元组集D的不纯度，定义为

其中，p_i 是 D 中元组属于 C_i 类的概率，并用 $|C_{i,D}|/|D|$ 估计。对 m 个类计算和。

---其他属性选择度量

最小描述长度，多元划分，统计卡方检验

树剪枝

---先剪枝：通过提前停止树的构建而对树剪枝。一旦停止，结点就成为树叶。该树叶可以持有子集元组中最频繁的类，或这些元组的概率分布。

---后剪枝：它由完全生长的树剪去子树。通过删除结点的分枝并用树叶替换它而剪掉给定结点上的子树。该树叶的类标号用子树中最频繁的类标记。

复制是树中存在重复的子树。

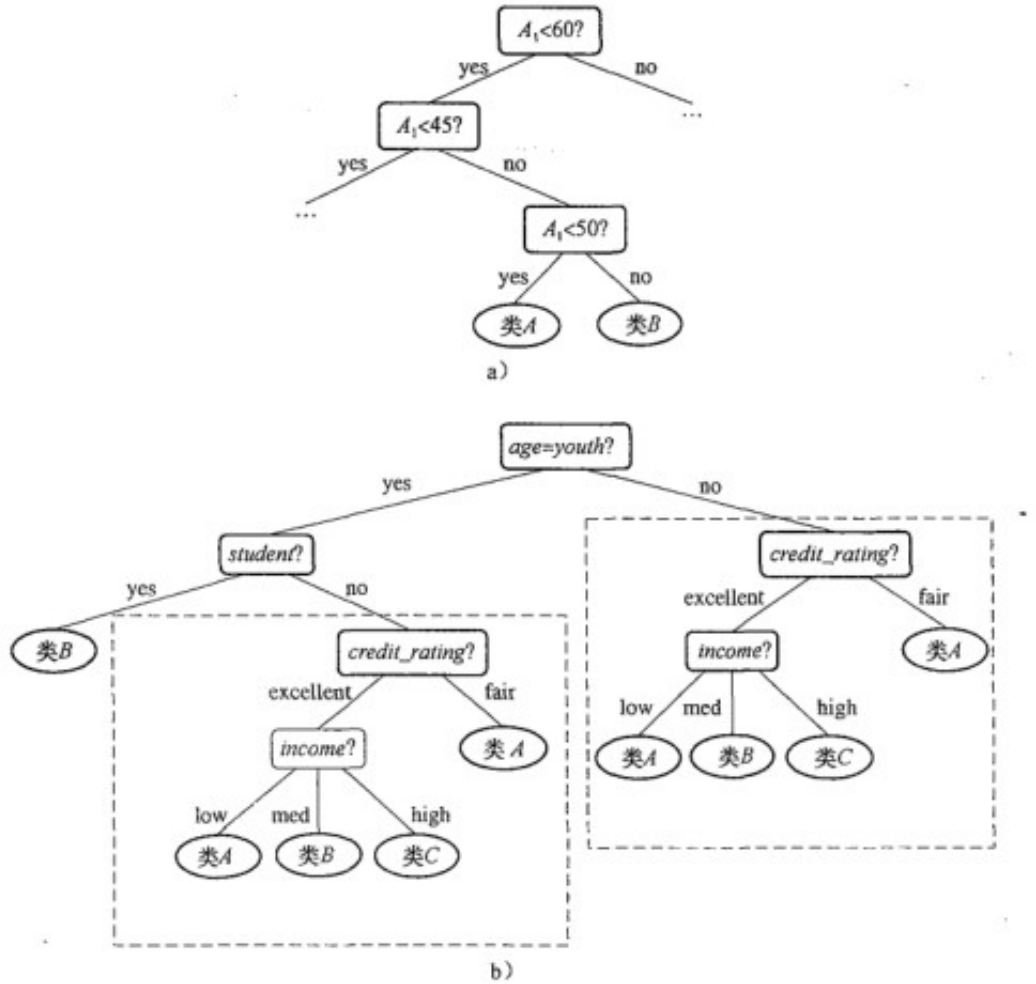

图 8.7　子树的例子：a）重复（其中属性 *age* 沿树的给定分枝重复地测试）；b）复制（树中存在重复的子树，如以结点“*credit_rating*?”开始的子树）

可伸缩性与决策树归纳

AVC-集（其中AVC表示“属性-值，类标号”）

e.g.

age	buys_computer yes	buys_computer no
youth	2	3
middle_aged	4	0
senior	3	2

income	buys_computer yes	buys_computer no
low	3	1
medium	4	2
high	2	2

student	buys_computer yes	buys_computer no
yes	6	1
no	3	4

credit_ratting	buys_computer yes	buys_computer no
fair	6	2
excellent	3	3

图 8.8　存放训练数据的聚集信息的数据结构（例如，描述表 8.1 中数据的 AVC-集）是提高决策树归纳可伸缩性的方法之一

树构造的自助乐观算法（BOAT）

决策树归纳的可视化挖掘

基于感知的分类（PBC-Perception-based Classification）

8.3贝叶斯分类方法

贝叶斯定理

设X是数据元组，在贝叶斯的术语中，X看作“证据”。通常，X用n个属性集的测量值描述。令H为某种假设，如数据元组X属于某个特定类C。对于分类问题，希望确定给定“证据”或观测数据元组X，假设H成立的概率P(H|X)。换言之，给定X的属性描述，找出元组X属于类C 的概率。

$P(H|X)$ 是**后验概率**（posterior probability），或在条件 X 下，H 的后验概率。例如，假设数据元组世界限于分别由属性 *age* 和 *income* 描述的顾客，而 X 是一位 35 岁的顾客，其收入为 4 万美元。令 H 为某种假设，如顾客将购买计算机。则 $P(H|X)$ 反映当我们知道顾客的年龄和收入时，顾客 X 将购买计算机的概率。

相反，$P(H)$ 是**先验概率**（prior probability），或 H 的先验概率。对于我们的例子，它是任意给定顾客将购买计算机的概率，而不管他们的年龄、收入或任何其他信息。后验概率 $P(H|X)$ 比先验概率 $P(H)$ 基于更多的信息（例如顾客的信息）。$P(H)$ 独立于 X。

类似地，$P(X|H)$ 是条件 H 下，X 的后验概率。也就是说，它是已知顾客 X 将购买计算机，该顾客是 35 岁并且收入为4 万美元的概率。

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}$$

贝叶斯定理：

朴素贝叶斯分类

朴素贝叶斯（Naïve Bayesian）分类法或**简单贝叶斯**分类法的工作过程如下：

（1）设 D 是训练元组和它们相关联的类标号的集合。通常，每个元组用一个 n 维属性向量 $\boldsymbol{X}=\{x_1, x_2, \cdots, x_n\}$ 表示，描述由 n 个属性 $A_1, A_2, \cdots, A_n$ 对元组的 n 个测量。

（2）假定有 m 个类 $C_1, C_2, \cdots, C_m$。给定元组 $\boldsymbol{X}$，分类法将预测 $\boldsymbol{X}$ 属于具有最高后验概率的类（在条件 $\boldsymbol{X}$ 下）。也就是说，朴素贝叶斯分类法预测 $\boldsymbol{X}$ 属于类 C_i，当且仅当

$$P(C_i \mid \boldsymbol{X}) > P(C_j \mid \boldsymbol{X}) \qquad 1 \leqslant j \leqslant m, \quad j \neq i$$

这样，最大化 $P(C_i \mid \boldsymbol{X})$。$P(C_i \mid \boldsymbol{X})$ 最大的类 C_i 称为最大后验假设。根据贝叶斯定理（(8.10) 式），

$$P(C_i \mid \boldsymbol{X}) = \frac{P(\boldsymbol{X} \mid C_i)P(C_i)}{P(\boldsymbol{X})} \tag{8.11}$$

（3）由于 $P(\boldsymbol{X})$ 对所有类为常数，所以只需要 $P(\boldsymbol{X} \mid C_i)P(C_i)$ 最大即可。如果类的先验概率未知，则通常假定这些类是等概率的，即 $P(C_1)=P(C_2)=\cdots=P(C_m)$，并据此对 $P(\boldsymbol{X} \mid C_i)$ 最大化。否则，最大化 $P(\boldsymbol{X} \mid C_i)P(C_i)$。注意，类先验概率可以用 $P(C_i)=|C_{i,D}|/|D|$ 估计，其中 $|C_{i,D}|$ 是 D 中 C_i 类的训练元组数。

（4）给定具有许多属性的数据集，计算 $P(\boldsymbol{X} \mid C_i)$ 的开销可能非常大。为了降低计算 $P(\boldsymbol{X} \mid C_i)$ 的开销，可以做**类条件独立**的朴素假定。给定元组的类标号，假定属性值有条件地相互独立（即属性之间不存在依赖关系）。因此，

$$P(\boldsymbol{X} \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i) = P(x_1 \mid C_i)P(x_2 \mid C_i)\cdots P(x_n \mid C_i) \tag{8.12}$$

可以很容易地由训练元组估计概率 $P(x_1 \mid C_i)$，$P(x_2 \mid C_i)$，$\cdots$，$P(x_n \mid C_i)$。注意，x_k 表示元组 $\boldsymbol{X}$ 在属性 A_k 的值。对于每个属性，考察该属性是分类的还是连续值的。例如，为了计算 $P(\boldsymbol{X} \mid C_i)$，考虑如下情况：

（a）如果 A_k 是分类属性，则 $P(x_k \mid C_i)$ 是 D 中属性 A_k 的值为 x_k 的 C_i 类的元组数除以 D 中 C_i 类的元组数 $|C_{i,D}|$。

（b）如果 A_k 是连续值属性，则需要多做一点工作，但是计算很简单。通常，假定连续值属性服从均值为 μ、标准差为 σ 的高斯分布，由下式定义

$$g(x,\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{8.13}$$

因此

$$P(x_k \mid C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i}) \tag{8.14}$$

（5）为了预测 $\boldsymbol{X}$ 的类标号，对每个类 C_i，计算 $P(\boldsymbol{X} \mid C_i)P(C_i)$。该分类法预测输入元组 $\boldsymbol{X}$ 的类为 C_i，当且仅当

$$P(\boldsymbol{X} \mid C_i)P(C_i) > P(\boldsymbol{X} \mid C_j)P(C_j), \qquad 1 \leqslant j \leqslant m, j \neq i \tag{8.15}$$

换言之，被预测的类标号是使 $P(\boldsymbol{X} \mid C_i)P(C_i)$ 最大的类 C_i。

避免零概率值--->拉普拉斯校准/拉普拉斯估计法：可以假定训练数据库D很大，以至于对每个计数加1造成的估计概率的变化可以忽略不计，但可以方便的避免概率值为零。

attention：如果对q个计数都加上1，则必须记住在用于计算概率的对应分母上加上q。

8.4基于规则的分类

使用IF-THEN规则分类

一个IF-THEN规则是一个如下形式的表达式：IF条件THEN结论

e.g. IF age = youth AND student = yes THEN buys_computer = yes
规则的"IF"部分（或左部）成为规则前件或前提。"THEN"部分（或右部）是规则的结论。在规则前件，条件由一个或多个用逻辑连接词AND连接的属性测试。规则的结论包含一个类预测。上例中的规则也可以写作：（age = youth）∧（student = yes）=>（buys_computer = yes）

对于给定的元组，如果规则前件中的条件（即所有的属性测试）都成立，则我们说规则前件被满足（或简单地，规则被满足），并且规则覆盖了该元组。

规则 R 可以用它的覆盖率和准确率来评估。给定类标记的数据集 D 中的一个元组 $\boldsymbol{X}$，设 n_{covers} 为规则 R 覆盖的元组数，$n_{correct}$ 为 R 正确分类的元组数，$|D|$ 是 D 中的元组数。可以将 R 的覆盖率和准确率定义为

$$coverage(R) = \frac{n_{covers}}{|D|} \quad (8.16)$$

$$accuracy(R) = \frac{n_{correct}}{n_{covers}} \quad (8.17)$$

规则的覆盖率是规则覆盖（即其属性值使得规则的前件为真）的元组的百分比，规则的准确率则是可以被规则正确分类的元组所占的百分比。

如果规则被X满足，则称该规则被触发。如果R是唯一满足的规则，则该规则激活，返回X的类预测。
attention：触发并不总意味激活，因为可能有多个规则被满足。当多个规则被触发时，则可能存在一个问题，即它们指定了不同的类。

由决策树提取规则

使用顺序覆盖算法的规则归纳

8.5模型评估与选择

评估分类器性能的度量

保持方法和随机二次抽样

交叉验证

自助法

使用统计显著性检验选择模型

基于成本效益和ROC曲线比较分类器

8.6提高分类准确率的技术

组合分类方法简介

装袋

提升和AdaBoost

随机森林

提高类不平衡数据的分类准确率