非参角度的分类(Classification)

- 贝叶斯分类器(Bayesian Classifier), 错分率(error rate)
- 贝叶斯决策边界(Bayesian Decision Boundary)
- k近邻(KNN)
- 朴素贝叶斯(Naive Bayesian)
- 混淆矩阵(confusion matrix), ROC曲线(roc curve)

回归问题中往往假设响应变量是连续型的, 但在许多情况下, 响应变量是定性的(qualitative), 定性变量往往也被称为属性变量, 分类变量(categorical). 预测一个观测的定性响应值是对观测的**分类(classification)**.

大多数算法并非只是让模型进行一个二分类问题, 还需要预测该实例属于某一类的概率(倾向). 事实上, 当R实现逻辑回归时, 默认输出是对数几率(log-odds)形式的, 必须转化为倾向性分值, 然后才能使用一个滑动的截止值(cut-off)将倾向得分转化为决策, 分类也可以看成是估计倾向性的问题.

1 分类模型

考虑数据:$\{(Y_i, X_i)\}_{i=1}^n$, 其中X_i可以是多维的, 有p个特征, 即

$$X_i = \begin{pmatrix} x_{1i} \\ x_{2i} \\ \vdots \\ x_{pi} \end{pmatrix} \quad Y_i = \begin{Bmatrix} 1 \\ 2 \\ \vdots \\ J \end{Bmatrix}$$

例 1.

$$Y_i = \begin{Bmatrix} \text{猫} \\ \text{狗} \end{Bmatrix}$$

就是一个二分类问题.

我们要利用数据构造分类器

$$f : \begin{pmatrix} x_{1i} \\ x_{2i} \\ \vdots \\ x_{pi} \end{pmatrix} \to Y_i$$

使得$\hat{y} = f(X) = f(x_1, \cdots, x_p)$.

1.1 训练数据错误率(错分率)

(类比MSE) 最常用的衡量估计$\hat{f}$的精度的方法是训练数据错误率(*training error rate*),

$$\frac{1}{n}\sum_{i=1}^{n} I\left(\underbrace{y_i}_{\text{观测值}} \neq \underbrace{\hat{y}_i}_{\text{预测值}}\right)$$

也就是说对训练数据使用估计器$\hat{f}$所造成误差的比例.

1.2 测试数据错误率(错分率)

对于任意新的数据集$\{(X_i', Y_i')\}, i = 1, \cdots, n_0$ 则测试错误率(*test error rate*)

$$AVE(I(y' \neq \hat{y}'))$$

一个好的分类器应当使测试数据错误率越小越好.

用手头上的数据上n个数据点: $i = 1, \cdots, n$划分训练集与测试集, 比如训练$\leftrightarrow$测试可以为80% $\leftrightarrow$ 20%, 70% $\leftrightarrow$ 30%. 即80%的训练集, 20%的验证集, 等.

1.3 贝叶斯分类器(Bayesian Classifier, 非朴素的)

以下X, x_0为向量, 请注意区分.

$$f(x_0) = \underset{j \in \{1, \cdots, J\}}{\arg\min}\, P(Y = j \mid X = x_0)$$

上式表明将x_0分到对应概率最大的"标签".

注记. *Bayesian分类器可以把错误率降到最低.*

可以证明(见李航.统计学习方法.) 贝叶斯分类器将产生最低的测试错误率, 称为Bayes错误率. 事实上, 对$\forall X = x_0$, 错误率将是

$$1 - \max_j P(Y = j \mid X = x_0)$$

因此整体Bayes错分率(Overall Bayes Error)为

$$1 - E\left(\max_j \Pr(Y = j \mid X)\right)$$

例 2 (二分类问题, $J = 2$)**.** *二分类问题中, 只有两个可能的响应值,*

$$f(x_0) = \begin{cases} 1, & if\ P(Y = 1|X = x_0) \geq 0.5 \\ 2, & O.W. \end{cases}$$

这是因为

$$P(Y = 1|X = x_0) + P(Y = 2|X = x_0) = 1$$

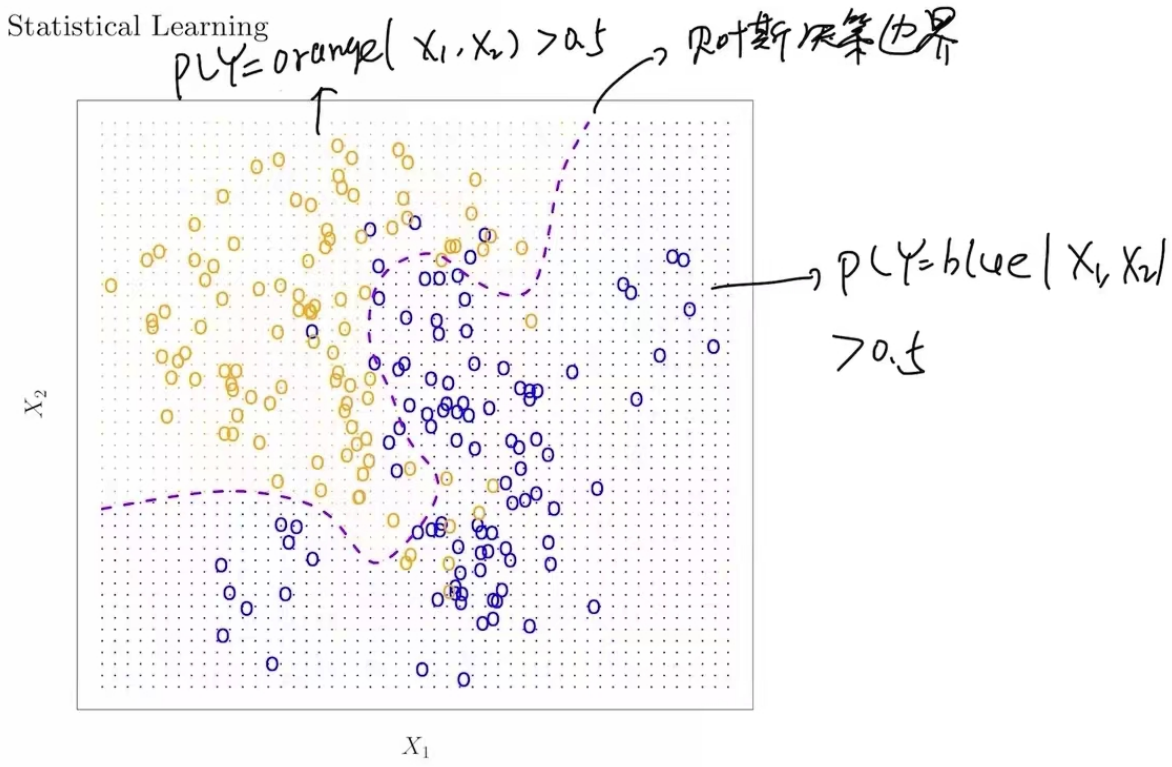
Bayes分类器的缺点: 在现实中往往不存在, 因为需要求条件概率, 是因为需要用到样本的分布,

1.4 Bayes决策边界(Bayesian Decision Boundary)

条件概率为0.5的那些点,

例 3. *若$X = (X_1, X_2)$, $Y = 1, 2$, 则Bayes决策边界为*

$$\Gamma = \{(X_1, X_2) : P(Y = 1|X = (X_1, X_2)) = P(Y = 1|X = (X_1, X_2)) = 0.5\}$$



上述的实验设计可能为,

$$X_1 \sim N(0, 1) \quad X_2|X_1 \sim N(2X_1, 1) \quad Y \sim Bernoulli(logit(X_1, X_2))$$

其中

$$logit(\xi) = \frac{\exp(1^\top \cdot \xi)}{1 + \exp(1^\top \cdot \xi)}$$

用贝叶斯分类器预测定性变量是一个自然的想法, 但对于一个实际的观测数据而言, 很难知道给定X之后Y的条件分布, 因此有时候计算Bayes分类器是不可能的, Bayes分类器相较于其他分类器而言更像是一个黄金标准. 许多方法尝试给定X之后先给出Y的条件分布.

1.5 K近邻方法(KNN)

首先用近邻的方法估计概率:

$$\begin{aligned} P(Y = j|X = x_0) &\approx P(Y = k|X \in x_0\text{的邻域}) \\ &= P(Y = k|X\text{属于}x_0\text{的}K\text{个最邻近的点}) \end{aligned}$$

用$\mathcal{N}_0$表示近邻的K个点的集合, 则可以定义为

$$P(Y=j|X=x_0) \approx P(Y=j|X \in \mathcal{N}_0)$$

因此可以用频率估计概率

$$\widehat{P}(Y=j|X=x_0) = \frac{1}{K}\sum_{i\in\mathcal{N}_0} I(y_i=j)$$

因此显式地写出KNN分类器, 可以写出

$$D(x_0) = \underset{j\in\{1,\cdots,J\}}{\arg\max} \frac{1}{K}\sum_{i\in\mathcal{N}_0} I(y_i=j)$$

可以看出是对Bayes分类器的逼近.

例 4 (二分类问题, $J=2$)**.**

$$D(x_0) = \begin{cases} 1, if\, P(Y=1|X=x_0) > 0.5 \\ 2, O.W. \end{cases}$$

上式中的0.5被称为阈值*(threshold),* 是可以调节的, 只不过取0.5是对*Bayes*分类器的逼近.

这里面K是超参数, K越大越光滑.

1.6 朴素贝叶斯(Naive Bayes)

回顾逆概率定理(Bayes定理)

$$P(Y=k|X=x) = \frac{\pi_k f_k(x)}{\sum_{i=1}^K \pi_i f_i(x)}$$

其中$\pi_k = P(Y=k)$, $f_k(x) = P(X=x|Y=k)$(密度函数, density function). Bayes定理将难以计算的后验(posterior)概率转换为更好计算的先验(prior)概率.

注记 (后验概率(Posterior probability))**.** 在给定预测因子$X=x$的条件下, 出现某一结果$Y=y$的概率. 后验概率不同于结果的先验概率, 先验概率并未考虑预测因子的信息.

对先验Y的分布直接应用频率估计概率:

$$\hat{\pi}_k = \frac{\sum_{i=1}^n I(Y_i=k)}{n}$$

对于p维数据X, 由于维数灾祸不好用频率估计概率, 因此

假设 (条件独立性假设)**.**

$$\begin{aligned} P\left(X=x \mid Y=c_k\right) &= P\left(X^{(1)}=x^{(1)}, \cdots, X^{(n)}=x^{(n)} \mid Y=c_k\right) \\ &= \prod_{j=1}^n P\left(X^{(j)}=x^{(j)} \mid Y=c_k\right) \end{aligned}$$

即不同的特征之间相互独立.

上述的假设说明了为什么该方法是**朴素(naive)**的, 因为我们做了一个简单的假设, 即嘉定X_j独立于其他所有的预测变量 $X_k(k \neq j)$, 这样就可以对 $P(X_j|Y = i)$ 给出估计, 而非直接去估计$P(X_1, \cdots, X_p|Y = i)$.

在*R*语言中,

因此

$$f_k(x) = f_{k_1}(x_1) \cdot \cdots \cdot f_{k_p}(x_p)$$

故密度估计也可以写成

$$\hat{f}_k(x) = \hat{f}_{k_1}(x_1) \cdot \cdots \cdot \hat{f}_{k_p}(x_p)$$

将两个先验估计代入Bayes公式, 有

$$\hat{P}(Y = k|X = x) = \frac{\hat{\pi}_k \hat{f}_k(x)}{\sum_{i=1}^{K} \hat{\pi}_i \hat{f}_i(x)}$$

因此Naive Bayes分类器可以写成

$$D(x_0) = \underset{k \in \{1, \cdots, K\}}{\arg\max} \frac{\hat{\pi}_k \hat{f}_k(x)}{\sum_{i=1}^{K} \hat{\pi}_i \hat{f}_i(x)}$$

它是Bayes分类器的逼近. 而对单独的$\hat{f}_{k_1}(x_1)$密度函数的估计, 有以下几种情况:

- 定量(quantitative):
  - 参数方法: 假定$X_1 \sim N(\mu_1, \sigma_1^2)$
  - 非参数方法: 比如核密度估计.
- 定性(qualitative): 用频率估计概率

1.7 代码实现

1.7.1 数据描述

`Smarket`是股票市场数据, 其数据包含

```
> str(Smarket)
'data.frame':   1250 obs. of  9 variables:
 \$ Year     : num  2001 2001 2001 2001 2001 ...
 \$ Lag1     : num  0.381 0.959 1.032 -0.623 0.614 ...
 \$ Lag2     : num  -0.192 0.381 0.959 1.032 -0.623 ...
 \$ Lag3     : num  -2.624 -0.192 0.381 0.959 1.032 ...
 \$ Lag4     : num  -1.055 -2.624 -0.192 0.381 0.959 ...
 \$ Lag5     : num  5.01 -1.055 -2.624 -0.192 0.381 ...
 \$ Volume   : num  1.19 1.3 1.41 1.28 1.21 ...
 \$ Today    : num  0.959 1.032 -0.623 0.614 0.213 ...
 \$ Direction: Factor w/ 2 levels "Down","Up": 2 2 1 2 2 2 1 2 2 2 ...
```

其中`Lag1`到`Lag5`表示了过去5个交易日每天的投资回报率, 同时`Volume`表示前一日股票成交量, `Today`当日投资回报率, `Direction`这些对应数据在市场走向的方向. 可以排除属性变量`Direction`计算原始数值之间的相关性,

```
> options(digits = 3)
> cor(Smarket[-9])
         Year     Lag1     Lag2     Lag3     Lag4     Lag5  Volume    Today
Year   1.0000  0.02970  0.03060  0.03319  0.03569  0.02979  0.5390  0.03010
Lag1   0.0297  1.00000 -0.02629 -0.01080 -0.00299 -0.00567  0.0409 -0.02616
Lag2   0.0306 -0.02629  1.00000 -0.02590 -0.01085 -0.00356 -0.0434 -0.01025
Lag3   0.0332 -0.01080 -0.02590  1.00000 -0.02405 -0.01881 -0.0418 -0.00245
Lag4   0.0357 -0.00299 -0.01085 -0.02405  1.00000 -0.02708 -0.0484 -0.00690
Lag5   0.0298 -0.00567 -0.00356 -0.01881 -0.02708  1.00000 -0.0220 -0.03486
Volume 0.5390  0.04091 -0.04338 -0.04182 -0.04841 -0.02200  1.0000  0.01459
Today  0.0301 -0.02616 -0.01025 -0.00245 -0.00690 -0.03486  0.0146  1.00000
```

通过相关性矩阵可以看出前几日的投资回报量和当日达投资回报量之间几乎没有关系, 也就是说我们发现的强相关来源于Year, Volumn,

1.7.2 代码

```
nb.fit <- e1071::naiveBayes(Direction~Lag1+Lag2, data = Smarket, subset =train)
nb.fit
mean(Lag1[train][Direction[train] == "Down"])
sd(Lag1[train][Direction[train] == "Down"])
nb.class <- predict(nb.fit, Smarket.2005)
table(nb.class, Smarket.2005$Direction)
mean(nb.class == Smarket.2005$Direction)
nb.preds <- predict(nb.fit, Smarket.2005, type = "raw")
head(nb.preds)
```

最后的结果为

```
> table(nb.class, Smarket.2005$Direction)

nb.class Down  Up
    Down   28  20
    Up     83 121
> mean(nb.class == Smarket.2005$Direction)
[1] 0.591
> nb.preds <- predict(nb.fit, Smarket.2005, type = "raw")
> head(nb.preds)
      Down    Up
[1,] 0.487 0.513
[2,] 0.476 0.524
[3,] 0.465 0.535
[4,] 0.475 0.525
[5,] 0.490 0.510
[6,] 0.491 0.509
```

当然, 用其他包也可以实现相似的功能

```
nb_fit <- klaR::NaiveBayes(Direction~Lag1+Lag2, data = Smarket, subset = train)
```

2 混淆矩阵

错误率和精度虽然常用, 但不能满足所有任务要求. 我们经常会关心“检出的信息中有多少比

例的信息是用户感兴趣的", "用户感兴趣的信息有多少被检索出来了". **"查准率(precision, 预测阳性率)"**和**"查全率(recall, 真阳性率)"**是更为适用此类需求的度量. [1]

对于二分类问题, 可将样例根据其真实类别与学习器预测类别的组合划分为真正例(true positive, 真阳性), 假正例 (false positive, 假阳性), 真反例 (true negative, 真阴性), 假反例 (false negative, 假阴性) 四种情形, 今TP, FP, TN, FN分别表示其对应的样例数, 则显然有$TP + FP + TN + FN$ =样例总数. 分类结果的"混滑矩阵" (confusion matrix) 如表1所示

真实情况	预测结果	
	正例	反例
正例	TP (真正例)	FN (假反例)
反例	FP(假正例)	TN (真反例)

表 1: 混淆矩阵

为简化公式起见, 可以采用如下的记号

$$N = TN + FP$$

$$P = FN + TP$$

$$N^* = TN + FN$$

$$P^* = TP + FP$$

名称	定义	同义词
假阳性率	FP/N	第I类错误, 1−特异性
真阳性率	TP/P	1−第II类错误, 势, 灵敏度, 召回率
预测阳性率	TP/P^*	精确度
预测阴性率	TN/N^*	

表 2: 重要的评价指标

真阳性率和预测阳性率是一对矛盾的度量, 通常只有在一些简单任务中, 才可能使真阳性率和预测阳性率都很高.

3 ROC曲线

很多学习器是为测试样本产生一个实值或概率预测,然后将这个预测值与一个分类闽值 (threshold) 进行比较, 若大于阙值则分为正类,否则为反类. 例如神经网络在一般情形下是对每个测试样本预测出一个[0, 1] 之间的实值然后将这个值与 0.5 进行比较, 大于0.5 则判为正例否则为反例这个实值或概率预测结果的好坏,直接决定了学习器的泛化能力. 实际上,根据这个实值或概率预测结果, 我们可将测试样本进行排序, "最可能"是正例的排在最前面, "最不可能"是正例的排在最后面,

[1]查准率亦称"准确率", 查全率亦称"召回率".

这样, 分类过程就相当于在这个排序中以某个“截断点”(cut point)将样本分为两部分: 前一部分判作正例,后一部分则判作反例.

在不同的应用任务中,我们可根据任务需求来采用不同的截断点, 例如若我们更重视“查准率”, 则可选择排序中靠前的位置进行截断; 若更重视“查全率”, 则可选择靠后的位置进行截断. 因此,排序本身的质量好坏,体现了综合考虑学习器在不同任务下的“期望泛化性能”的好坏, 或者说“一般情况下”泛化性能的好坏. ROC 曲线则是从这个角度出发来研究学习器泛化性能的有力工具.

ROC全称是“受试者工作特征”(Receiver Operating Characteristic)曲线, 曲线相似我们根据学习器的预测结果对样例进行排序,按此顺序逐个把样本作为正例进行预测, 每次计算出两个重要量的值, 分别以它们为横、纵坐标作图,就得到了“ROC 曲线”. ROC 曲线的纵轴是“真正例率”(True Positive Rate,简称TPR),横轴是“假正例率”(False PositiveRate,简称 FPR), 其公式可以写为

$$TPR = \frac{TP}{TP + FN}, \quad FPR = \frac{FP}{TN + FP}$$

分类器的性能表现是通过ROC曲线下的面积(AUC, area under the ROC curve)来表示的, 该曲线能够涵盖所有可能的阈值. 一个理想的ROC曲线会紧贴着左上角, 所以AUC值越大, 分类器越好.

注记. 任何一个建模分类器的AUC至少为0.5.

A 朴素贝叶斯分类器拓展

基本方法

设输入空间 $\mathcal{X} \subseteq \mathbf{R}^n$ 为 n 维向量的集合, 输出空间为类标记集合 $\mathcal{Y} = \{c_1, c_2, \cdots, c_K\}$. 输入为特征向量 $x \in \mathcal{X}$, 输出为类标记 (class label) $y \in \mathcal{Y}$.X 是定义在输入空间 $\mathcal{X}$ 上的随机向量, Y 是定义在输出空间 $\mathcal{Y}$ 上的随机变量. $P(X,Y)$ 是 X 和 Y 的联合概率分布. 训练数据集

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

由 $P(X,Y)$ 独立同分布产生. 朴素贝叶斯法通过训练数据集学习联合概率分布 $P(X,Y)$. 具体地, 学习以下先验概率分布及条件概率分布. 先验概率分布

$$P(Y = c_k), \quad k = 1, 2, \cdots, K$$

条件概率分布

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

于是学习到联合概率分布 $P(X,Y)$. 条件概率分布 $P(X = x \mid Y = c_k)$ 有指数级数量的参数, 其估计实际是不可行的. 事实上, 假设 $x^{(j)}$ 可取值有 S_j 个, $j = 1, 2, \cdots, n$, Y 可取值有 K 个, 那么参数个数为 $K\prod_{j=1}^{n} S_j$.

朴素贝叶斯法对条件概率分布作了条件独立性的假设. 由于这是一个较强的假设, 朴素贝叶斯法也由此得名. 具体地, 条件独立性假设是

$$\begin{aligned} P(X = x \mid Y = c_k) &= P\left(X^{(1)} = x^{(1)}, \cdots, X^{(n)} = x^{(n)} \mid Y = c_k\right) \\ &= \prod_{j=1}^{n} P\left(X^{(j)} = x^{(j)} \mid Y = c_k\right) \end{aligned}$$

朴素贝叶斯法实际上学习到生成数据的机制, 所以属于生成模型. 条件独立假设等于是说用于分类的特征在类确定的条件下都是条件独立的. 这一假设使朴素贝叶斯法变得简单, 但有时会牺牲一定的分类准确率. 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 $P(Y = c_k \mid X = x)$, 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:

$$P(Y = c_k \mid X = x) = \frac{P(X = x \mid Y = c_k) P(Y = c_k)}{\sum_k P(X = x \mid Y = c_k) P(Y = c_k)}$$

将式条件代入上式有

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

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

$$y = f(x) = \arg\max_{c_k} \frac{P(Y = c_k) \prod_j P\left(X^{(j)} = x^{(j)} \mid Y = c_k\right)}{\sum_k P(Y = c_k) \prod_j P\left(X^{(j)} = x^{(j)} \mid Y = c_k\right)}$$

注意到, 在上式中分母对所有 c_k 都是相同的, 所以,

$$y = \arg\max_{c_k} P(Y = c_k) \prod_j P\left(X^{(j)} = x^{(j)} \mid Y = c_k\right)$$

后验概率最大化的含义

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. 假设选择 0-1 损失函数:

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

式中 $f(X)$ 是分类决策函数. 这时, 期望风险函数为

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

期望是对联合分布 $P(X, Y)$ 取的. 由此取条件期望

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

为了使期望风险最小化, 只需对 $X = x$ 逐个极小化, 由此得到:

$$\begin{aligned} f(x) &= \arg\min_{y \in \mathcal{Y}} \sum_{k=1}^{K} L(c_k, y) P(c_k \mid X = x) \\ &= \arg\min_{y \in \mathcal{Y}} \sum_{k=1}^{K} P(y \neq c_k \mid X = x) \\ &= \arg\min_{y \in \mathcal{Y}} (1 - P(y = c_k \mid X = x)) \\ &= \arg\max_{y \in \mathcal{Y}} P(y = c_k \mid X = x) \end{aligned}$$

这样一来, 根据期望风险最小化准则就得到了后验概率最大化准则:

$$f(x) = \arg\max_{c_k} P(c_k \mid X = x)$$

即朴素贝叶斯法所采用的原理.