



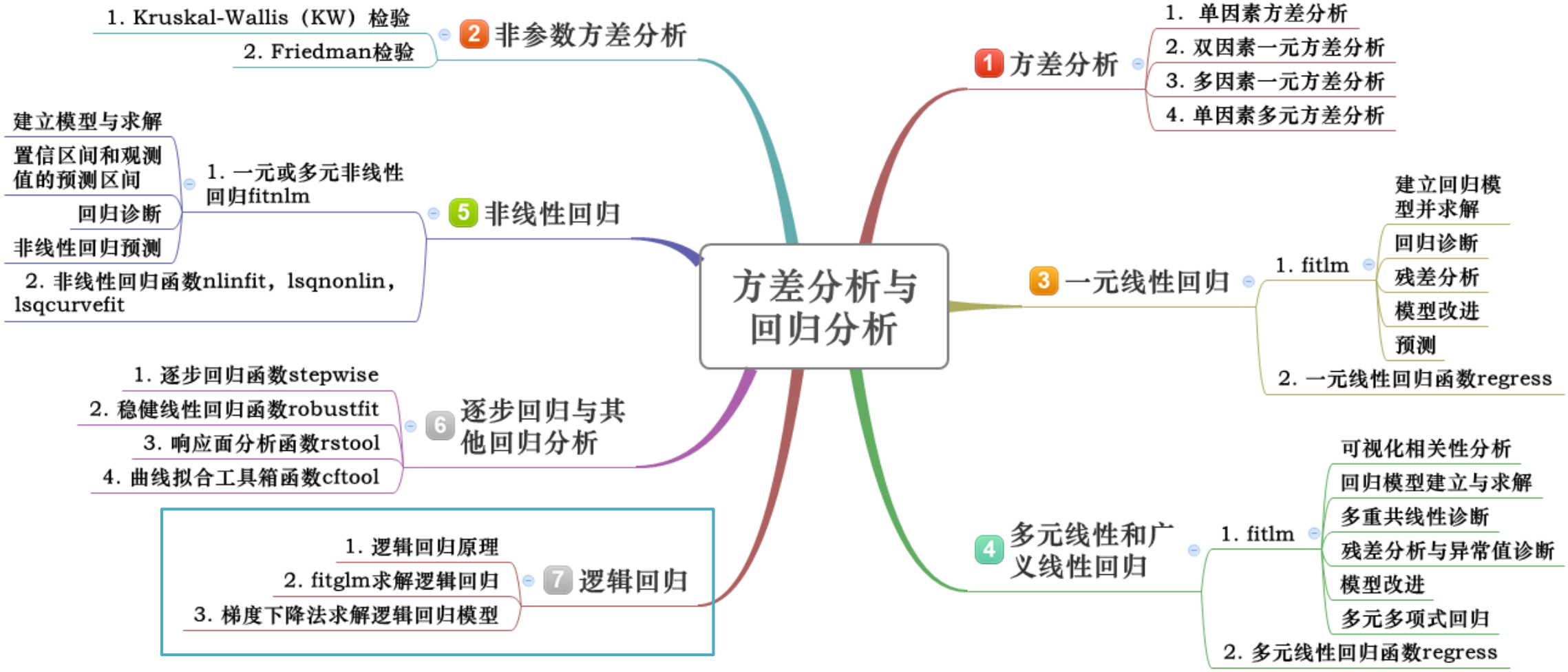
信陽師範学院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

# 第11章 方差分析与回归分析

**讲授人**：牛言涛

**日期**：2020年4月28日

# 第11章 方差分析与回归分析知识点思维导图



# 1. 逻辑回归简介

- 逻辑回归是广义线性回归GLM的一种形式，属于有监督学习。
- 对于某些分类问题，自变量可能是连续的，但是因变量却可能是离散的，例如：根据肿瘤大小判断该肿瘤是否是良性。这种问题不适合用线性回归来解决，虽然可以将连续的因变量值映射到离散的分类上，但是效果和训练复杂度都不尽如人意。
- 因此，逻辑回归（logistic regression）就成为了一个解决分类问题的好方法。

| 无监督学习 | 有监督学习 | |
|---|---|---|
| 聚类算法 | 分类算法 | 回归算法 |
| K均值聚类 | 决策树 | 线性回归 |
| 层次聚类 | 支持向量机 | 非线性回归 |
| 系统聚类 | 贝叶斯 | 逻辑回归 |
| 基于密度的聚类方法(DBSCAN)…… | K近邻算法 | 最小二乘回归 |
| 降维算法 | 逻辑回归 | LOESS局部回归 |
| 主成分分析PCA | 随机森林 | …… |
| 线性判断分析LDA | 关联规则分类 | 神经网络 |
| 异值分解(SVD) | 神经网络…… | 深度学习 |

# 1. 逻辑回归简介

信阳师范学院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

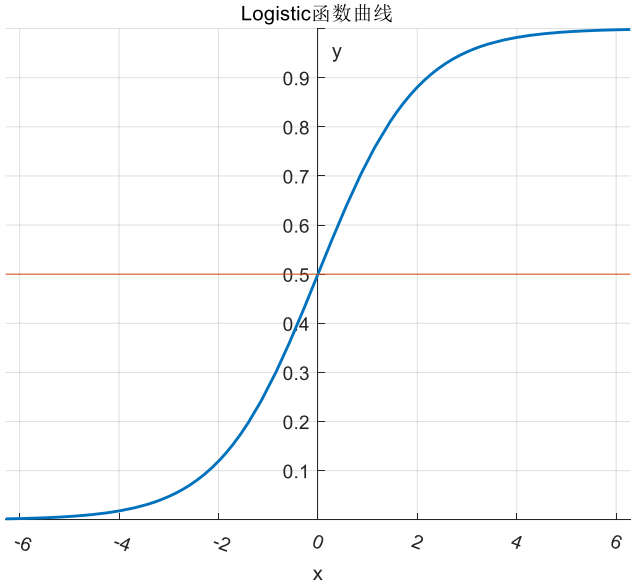
- 所谓逻辑回归，就是通过函数模型将因变量的值控制到0~1之间，然后通过梯度下降法或极大似然估计求出模型的参数，最后使用一个值域在（0，1）的函数进行预测，预测的结果就是分类为1的概率。
- Logistic回归的主要用途：
  - 寻找危险因素：寻找某一疾病的危险因素等；
  - 预测：根据模型，预测在不同的自变量情况下，发生某病或某种情况的概率有多大；
  - 判别：实际上跟预测有些类似，也是根据模型，判断某人属于某病或属于某种情况的概率有多大，也就是看一下这个人有多大的可能性是属于某病。
- Logistic回归主要在流行病学中应用较多，比较常用的情形是探索某疾病的危险因素，根据危险因素预测某疾病发生的概率，等等。例如，想探讨胃癌发生的危险因素，可以选择两组人群，一组是胃癌组，一组是非胃癌组，两组人群肯定有不同的体征和生活方式等。这里的因变量就是是否胃癌，即“是”或“否”，自变量就可以包括很多了，例如年龄、性别、饮食习惯、幽门螺杆菌感染等。自变量既可以是连续的，也可以是分类的。

# 2. 逻辑回归原理

二分类的逻辑回归模型中的因变量只有1或0（如是和否，发生和不发生）两种取值。假设在$p$个独立的自变量$x_1, x_2, \cdots, x_p$作用下，记$y$取1的概率$p = P(y = 1|X)$，取0的概率$1 - p$，取1和取0的概率之比为$p/(1 - p)$，称为事件的优势比（odds），对odds取自然对数即得logistic变换：

$$Logit(p) = \ln\left(\frac{p}{1-p}\right)$$

令 $Logit(p) = \ln\left(\frac{p}{1-p}\right) = z$ ，则 $p = \frac{1}{1+e^{-z}}$，即为Logistic函数(Sigmoid函数)，图像：




当$p$在$(0, 1)$之间变化时，odds取值范围$(0, +\infty)$，$\ln\left(\frac{p}{1-p}\right)$取值范围是$(-\infty, +\infty)$. 由于$p$表征样本属于正类（类别为1）的概率，通常将$p$大于某个阈值（如0.5）的样本预测为“属于正类1”，否则预测结果为“属于负类0”。

# 2. 逻辑回归原理

逻辑回归模型的建立是建立 $\ln\left(\frac{p}{1-p}\right)$ 与自变量的线性回归模型，其模型为：

$$\ln\left(\frac{p}{1-p}\right)=\beta_0+\beta_1 x_1+\cdots+\beta_p x_p+\varepsilon=w^T x$$

因为 $\ln\left(\frac{p}{1-p}\right)$ 的取值范围是$(-\infty,+\infty)$，这样，自变量$x_1,x_2,\cdots,x_p$可在任意取直范围。得到

$$p=P\left(y=1\middle|X\right)=\frac{1}{1+e^{-w^T x}},\qquad 1-p=P\left(y=0\middle|X\right)=1-\frac{1}{1+e^{-w^T x}}=\frac{1}{1+e^{w^T x}}$$

上式表明，逻辑回归的最终目标是根据数据样本确定合适的一组权重$w$，在对原始输入进行加权组合之后，通过关联函数做非线性变换，得到的结果表示样本$x$属于正类的概率。

# 3. 逻辑回归求解函数fitglm

信阳师范学院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

## fitglm为广义线性回归函数，其调用格式：

`mdl = fitglm(tbl)` returns a generalized linear model fit to variables in the table or dataset array `tbl`. By default, `fitglm` takes the last variable as the response variable.

`mdl = fitglm(X,y)` returns a generalized linear model of the responses `y`, fit to the data matrix `X`.

`mdl = fitglm( ___ ,modelspec)` returns a generalized linear model of the type you specify in `modelspec`.

`mdl = fitglm( ___ ,Name,Value)` returns a generalized linear model with additional options specified by one or more `Name,Value` pair arguments.

For example, you can specify which variables are categorical, the distribution of the response variable, and the link function to use.

### Canonical Link Function

The default link function for a generalized linear model is the *canonical link function*.

| Distribution | Canonical Link Function Name | Link Function | Mean (Inverse) Function |
|---|---|---|---|
| 'normal' | 'identity' | $f(\mu) = \mu$ | $\mu = Xb$ |
| 'binomial' | 'logit' | $f(\mu) = \log(\mu/(1-\mu))$ | $\mu = \exp(Xb) / (1 + \exp(Xb))$ |
| 'poisson' | 'log' | $f(\mu) = \log(\mu)$ | $\mu = \exp(Xb)$ |
| 'gamma' | -1 | $f(\mu) = 1/\mu$ | $\mu = 1/(Xb)$ |
| 'inverse gaussian' | -2 | $f(\mu) = 1/\mu^2$ | $\mu = (Xb)^{-1/2}$ |

# 4. 案例分析

例 1：企业到金融商业机构贷款，金融商业机构需要对企业进行评估。例如，Moody公司就是New York的一家专门评估企业的贷款信誉的公司。设：

$$y=\begin{cases}0, \text{企业2年后破产}\\ 1, \text{企业2年后具备还款能力}\end{cases},\quad X_1=\frac{\text{未分配利润}}{\text{总资产}},\quad X_2=\frac{\text{支付利息前的利润}}{\text{总资产}},\quad X_3=\frac{\text{销售额}}{\text{总资产}}$$

建立破产特征变量$y$的回归方程。

```
>> data = xlsread('business_data.xlsx',1,'A2:D67');
>> X = data(:,2:4);
>> Y = data(:,1);
>> GM = fitglm(X,Y,'Distribution','binomial', 'link', 'logit')
```

```
GM =
Generalized linear regression model:
    logit(y) ~ 1 + x1 + x2 + x3
    Distribution = Binomial

Estimated Coefficients:
                   Estimate      SE        tStat       pValue
                   ________    _______    ________    ________
    (Intercept)    -10.153      10.84     -0.93666    0.34893
    x1             0.33125     0.30074     1.1014     0.27071
    x2             0.18088     0.10692     1.6916     0.090714
    x3              5.0875      5.0821      1.001      0.3168

66 observations, 62 error degrees of freedom
Dispersion: 1
Chi^2-statistic vs. constant model: 85.7, p-value = 1.85e-18
```

# 4. 案例分析

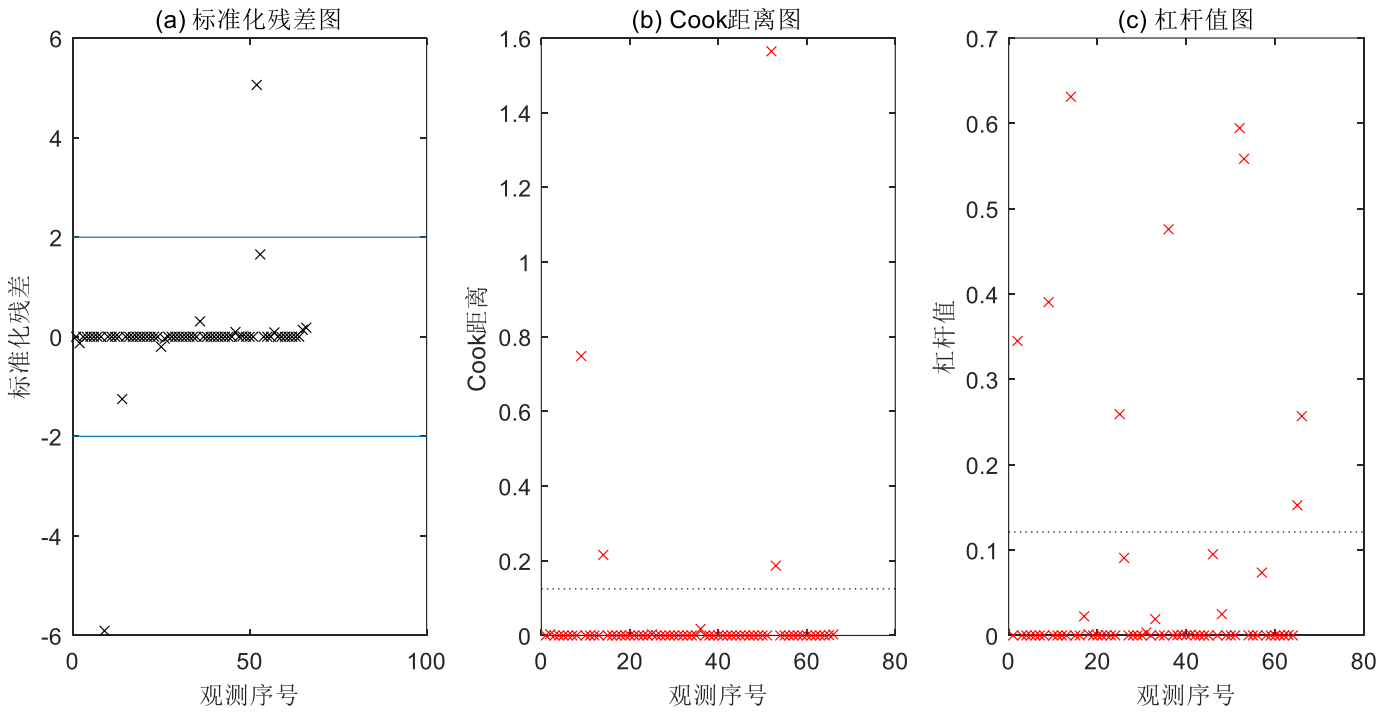


## 对模型进行回归诊断，编写函数

```
function [idout,idinf,idleve] = fitglmplot_outliers(model)
    Res = model.Residuals.Raw;
    Res = (Res - mean(Res))./std(Res);
    idout = find(abs(Res) > 2);  %绝对值大于2
    subplot(1,3,1);
    plot(Res,'kx');
    refline(0,-2);
    refline(0,2);
    title('(a) 标准化残差图')
    xlabel('观测序号');ylabel('标准化残差');
    subplot(1,3,2);
    model.plotDiagnostics('cookd');
    title('(b) Cook距离图')
    xlabel('观测序号');
    ylabel('Cook距离');
    md = model.Diagnostics;
    %强影响点，大于3倍cook距离均值
    idinf = find(md.CooksDistance > 3*mean(md.CooksDistance));
    subplot(1,3,3);
    model.plotDiagnostics('leverage');
    title('(c) 杠杆值图');
    xlabel('观测序号');
    ylabel('杠杆值');
    idleve = find(md.Leverage > 2*(model.NumCoefficients+1)/...
        size(model.Residuals,1));   %  h > 2 (p + 1) / n
end
```

# 4. 案例分析

```
[idout,idinf,idleve] = fitglmplot_outliers(model)
%回代，进行正确率判断：
>> Y1 = predict(GM,X);  %回代预测
>> k = find(Y1>0.5);  %查找大于0.5的行索引
>> class = zeros(length(Y1),1);
>> class(k) = 1;  %预测值大于0.5赋值1
>> correct = find(class == Y);
>> errorind = find(class ~= Y)   %错误样本
errorind =
     9
    52
>> acc = length(correct)/length(Y)
acc =
    0.9697
```




对模型进行回归诊断，识别异常值、高杠杆值和强影响点。对于三个图像，尤其以第9号样本和第52号样本特殊，既是异常值，又是高杠杆值和强影响点，可对此进一步分析，如第9号样本具备还款能力的概率较大，而52号样本正好相反，但数据与我们预期不符。

# 5. 评价指标

信阳师范学院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

- 评价指标：混淆矩阵

| 实际类别 | 预测类别 | | | |
|---|---|---|---|---|
| | | Yes | No | 总计 |
| | Yes | True positives(TP) | False negatives(FN) | P（实际为Yes） |
| | No | False positives(FP) | True negatives(TN) | N（实际为No） |
| | 总计 | P’（被分为Yes） | N’（被分为No） | P+N |

1. 正确率（accuracy）：accuracy = （TP+TN）/(P+N)，即被分对的样本数除以所有的样本数，通常来说，正确率越高，分类器越好；

2. 错误率（error rate)：错误率则与正确率相反，描述被分类器错分的比例，error rate = (FP+FN)/(P+N)，对某一个实例来说，分对与分错是互斥事件，所以accuracy =1 - error rate；

# 5. 评价指标



3. 灵敏度（sensitive）：sensitive = TP/P，表示的是所有正例中被分对的比例，衡量了分类器对正例的识别能力；

4. 特效度（specificity)：specificity = TN/N，表示的是所有负例中被分对的比例，衡量了分类器对负例的识别能力；

5. 精度（precision）：精度是精确性的度量，表示被分为正例的示例中实际为正例的比例，precision=TP/（TP+FP）；

6. 召回率（recall）：召回率是覆盖面的度量，度量有多个正例被分为正例，recall = TP/(TP+FN) = TP/P = sensitive，可以看到召回率与灵敏度是一样的。

7. 其他评价指标

计算速度：分类器训练和预测需要的时间；鲁棒性：处理缺失值和异常值的能力；可扩展性：处理大数据集的能力；可解释性：分类器的预测标准的可理解性，像决策树产生的规则就是很容易理解的，而神经网络的一堆参数就不好理解，只好把它看成一个黑盒子。

# 案例分析

例2：乳腺癌数据一共有569组32维。其中两个分类，良性 benign样本357个、恶性 malignant样本212个。

```
>> load('breast-cancer-data.mat')
>> data(:,1) = [];
>> Y = data(:,1)-1;
>> X = data(:,2:end);
>> GM = fitglm(X,Y,'Distribution','binomial', 'link', 'logit')
GM =
Generalized linear regression model:
    y ~ [Linear formula with 31 terms in 30 predictors]
    Distribution = Binomial
569 observations, 538 error degrees of freedom
Dispersion: 1
Chi^2-statistic vs. constant model: 247, p-value = 6.2e-36
```

```
>> Y1 = predict(GM,X);
>> k = find(Y1>0.5);
>> class = zeros(length(Y1),1);
>> class(k) = 1;
>> correct = find(class == Y);
>> acc = length(correct)/length(Y)
acc =
    0.9877
```

# 11.7 逻辑回归

```
load('breast-cancer-data.mat')
data(:,1) = [];
Y = data(:,1)-1;
X = data(:,2:end);
acc = [];
for i = 1:10
    n = length(Y);
    rp = randperm(n,round(n*0.2));
    Xtest = X(rp,:);
    Ytest = Y(rp);
    Xtrain = X;  Xtrain(rp,:) = [];
    Ytrain = Y;  Ytrain(rp) = [];
    GM = fitglm(Xtrain,Ytrain,'Distribution','binomial', 'link', 'logit');
    preY = predict(GM,Xtest);
    k = find(preY>0.5);
    class = zeros(length(preY),1);
    class(k) = 1;
    correct = find(class == Ytest);
    acclv = length(correct)/length(Ytest);
    acc = [acc,acclv];
end
macc = mean(acc)
```

```
acc = 0.9561    0.9298    0.9386    0.9561    0.9561
0.9386    0.9561    0.9386    0.9737    0.9474
macc =
    0.9491
%十次交叉验证正确率为94.91%
```

# 6. 梯度下降法

信阳师范学院
数学与统计学院

误差的度量就是代价函数。误差越大，代价函数的值就越高。对于监督学习，主要有两种代价函数：

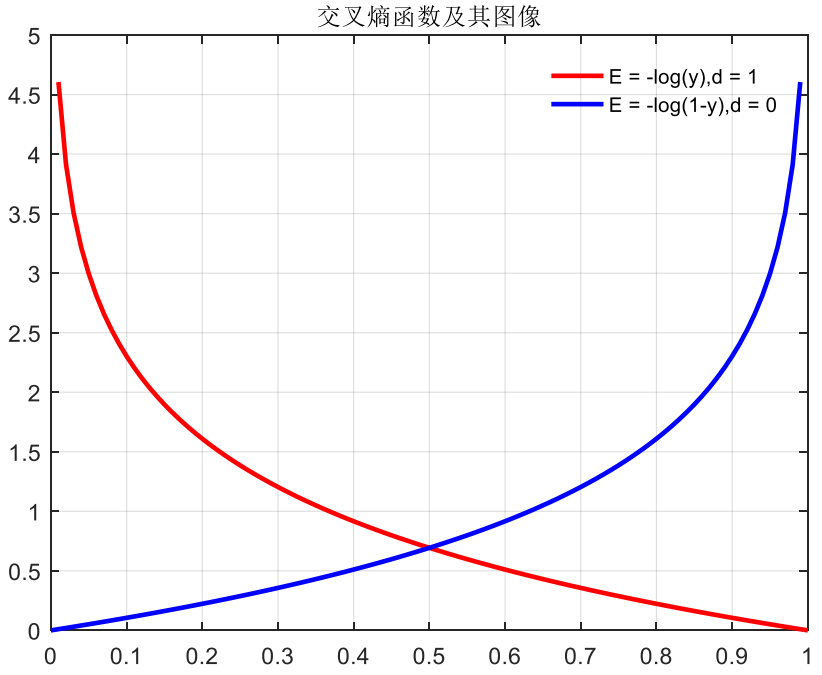
$$J=\sum_{i=1}^{M}\frac{1}{2}\left(d_i-y_i\right)^2 \qquad J=\sum_{i=1}^{M}\left\{-d_i\ln\left(y_i\right)-\left(1-d_i\right)\ln\left(1-y_i\right)\right\}$$

$y_i$是模型预测节点输出，$d_i$是训练数据的实际数值，$M$是训练样本量

交叉熵函数对误差更敏感，且与误差是成正比的，通常认为交叉熵函数导出的学习规则能够得到更好的性能。交叉熵函数与二次函数最主要的不同是，交叉熵函数随着误差的增大而呈几何上升趋势。

$$J=\sum_{i=1}^{M}\left\{-d_i\ln\left(y_i\right)-\left(1-d_i\right)\ln\left(1-y_i\right)\right\}$$

$$E=\begin{cases}-\ln\left(y\right), & d=1\\ -\ln\left(1-y\right), & d=0\end{cases}$$



# 6. 梯度下降法

找到预测函数，一般表示为h函数，用来预测输入数据的判断结果。分类边界为线性边界时，预测函数为 $h_\theta(x)=g(\theta^T x)=\frac{1}{1+e^{-\theta^T x}}$，$h_\theta(x)$函数的值表示结果取1的概率。

找到损失函数，记为$J(\theta)$函数，表示所有训练数据预测值和实际类别的偏差。基于极大似然估计，得到损失函数：

$$Cost(h_\theta(x),y)=\begin{cases}-\log(h_\theta(x)), & y=1\\ -\log(1-h_\theta(x)), & y=0\end{cases}$$

$$J(\theta)=\frac{1}{m}\sum_{i=1}^{m}Cost(h_\theta(x^{(i)}),y^{(i)})=-\frac{1}{m}\left[\sum_{i=1}^{m}y^{(i)}\log(h_\theta(x^{(i)}))+(1-y^{(i)})\log(1-h_\theta(x^{(i)}))\right]$$

找到J(θ)函数的最小值。根据梯度下降的公式，得到迭代公式（α为一常量，1/m一般省略）

$$\theta_j=\theta_j-\alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})x_j^{(i)}\ \Rightarrow\ \theta_j=\theta_j-\alpha\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})x_j^{(i)},\ \ j=0,1,\cdots,n$$

# 6. 梯度下降法

信阳师范学院
数学与统计学院

gradientDes_LR.m

```
function [theta,acc,L,prey] = gradientDes_LR(data,alpha,lamda,threshold,maxiter)
% gradientDes_LR采用批量梯度下降法更新参数，求解逻辑回归系数，使得代价最小化
% data是样本数据，其中最后一列为标准输出值(限定0-1)，只针对二分类
% alpha是学习率，threshold是代价阈值，maxiter最大迭代次数,lamda是正则化系数
% 输出参数：theta逻辑回归最优系数， acc是测试样本正确率，L是每次迭代平均代价

    %% 1、样本数据的处理
    label = data(:,end); %分类标志，对应标准的0或1值，因变量
    n = length(label);  %总的样本量
    %自变量样本，具有多个特征，一列代表一个特征
    Xdata = [data(:,1:end-1),ones(n,1)];
    Xdata = zscore(Xdata);  %中心化

    %%2、选择80%的样本作为训练样本，20%用于测试
    rind = randperm(n); %随机打乱样本序列
    label = label(rind);
    Xdata = Xdata(rind,:);
    %选取前80%作为训练，后20%作为测试
    k = round(0.8*n);
    Xtrain = Xdata(1:k,:);  %训练样本
    Ytrain = label(1:k);  %训练样本的标准输出标签
    Xtest = Xdata(k+1:end,:);  %测试样本
    Ytest = label(k+1:end);  %测试样本的标准输出标签

    %% 3、参数的初始化
    ft_num = size(Xdata,2);  %取特征数，即模型参数个数
    theta = rand(1,ft_num);  %模型参数的初始化，用0-1均匀分布的随机数，行向量
    m1 = size(Xtrain,1);  %训练样本数量
    m2 = size(Xtest,1);  %测试样本数量

    %% 4、批量随机梯度下降法求解模型参数
    L(1) = 1;
    for j = 1:maxiter
        dt = zeros(1,ft_num);  %一行
        loss = 0;
        for i = 1:m1  %每次循环取一个样本用于训练，一行
            x = Xtrain(i,:)'; %一行代表一个样本，转置后成为一列
            y = Ytrain(i);  %真实标准输出
            h = 1/(1+exp(-theta*x)); %函数值表示结果取1的概率
            dt = dt + (h - y)*x';  %一行，增量
            %代价函数，也可采用softmax交叉熵代价函数
            %loss = loss + y*log(h) + (1-y)*log(1-h);
            loss = loss + (h-y)^2;  %平方和损失函数
        end
        %loss = -loss/m1;  %每次迭代后的代价均值，对应交叉熵cost
        loss = loss/m1/2;  %每次迭代后的代价均值，对应平方和cost
        L(j+1) = loss;
        %模型参数的迭代更新，lamda是正则化系数，防止过拟合
        theta = theta - alpha*dt/m1 - lamda*theta/m1;
        %代价函数平均损失小于等于给定阈值，退出迭代
        if abs(L(j+1) - L(j)) <= threshold
            break
        end
    end
end
```

$$\theta_j = \theta_j - \frac{\alpha}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)x_j^{(i)} - \frac{\lambda}{m}\theta_j,\quad j=0,1,\cdots,n$$

# 6. 梯度下降法

```
56        %% 5、对测试样本，进行预测判别，计算正确率
57 —      Pacc = 0;  %正样本，对应1
58 —      Nacc = 0;  %负样本，对应0
59 —      prey = zeros(m2,1);
60 —      for i = 1:m2
61 —          prey(i) = 1/(1+exp(-theta*Xtest(i,:)')); %函数值表示结果取1的概率
62 —          if prey(i) >= 0.5 && Ytest(i) == 1
63 —              Pacc  = Pacc + 1;
64 —          elseif prey(i) < 0.5 && Ytest(i) == 0
65 —              Nacc = Nacc + 1;
66 —          end
67 —      end
68 —      acc = (Pacc + Nacc)/m2;  %正确率
69 —      Ptol = length(find(Ytest == 1));
70 —      Ntol = length(find(Ytest == 0));
71 —      disp('测试样本的混淆矩阵：')
72 —      fuzzMat = [Pacc,Ptol - Pacc; Ntol - Nacc, Nacc]
73
74        %%6、损失cost可视化，可能得不到图像，因为交叉熵函数的log问题。
75 —      plot(L,'r--','LineWidth',1.5) %绘制损失曲线，查看是否收敛
76 —      title('Cost下降曲线');xlabel('训练次数K');ylabel('损失Cost')
77 —      grid on
78 —  end
```

% 交叉熵损失函数测试

>> [theta,acc,L,prey] =

gradientDes_LR(X,0.01,0.2,0.00001,1000);

测试样本的混淆矩阵：

fuzzMat =

8  0

1  4

>> acc

acc =

0.9231

>> data = xlsread('business_data.xlsx',1,'A2:D67');

>> X = [data(:,2:end),data(:,1)];

# 6. 梯度下降法

% 平方和损失函数测试

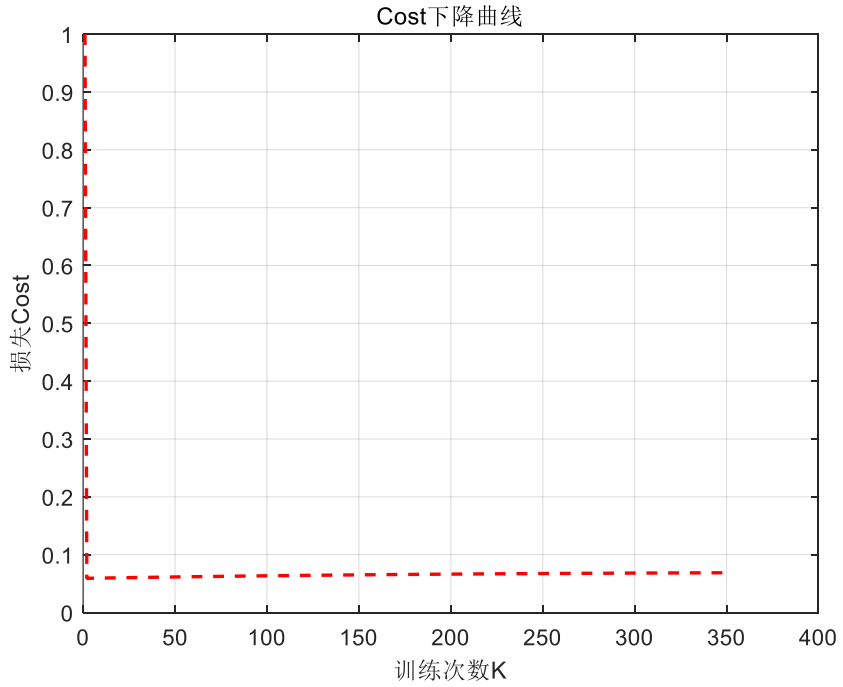
>> [theta,acc,L,prey] = gradientDes_LR(X,0.01,0.2,0.00001,1000);

测试样本的混淆矩阵：

fuzzMat =

8 0

0 5




```
for i = 1:10
    [theta,acc,L,prey] = gradientDes_LR(X,0.01,0.2,0.00001,1000);
    ACC(i) = acc;
end
acc = mean(ACC)
acc = 0.9462  %十次交叉验证，平均正确率
```

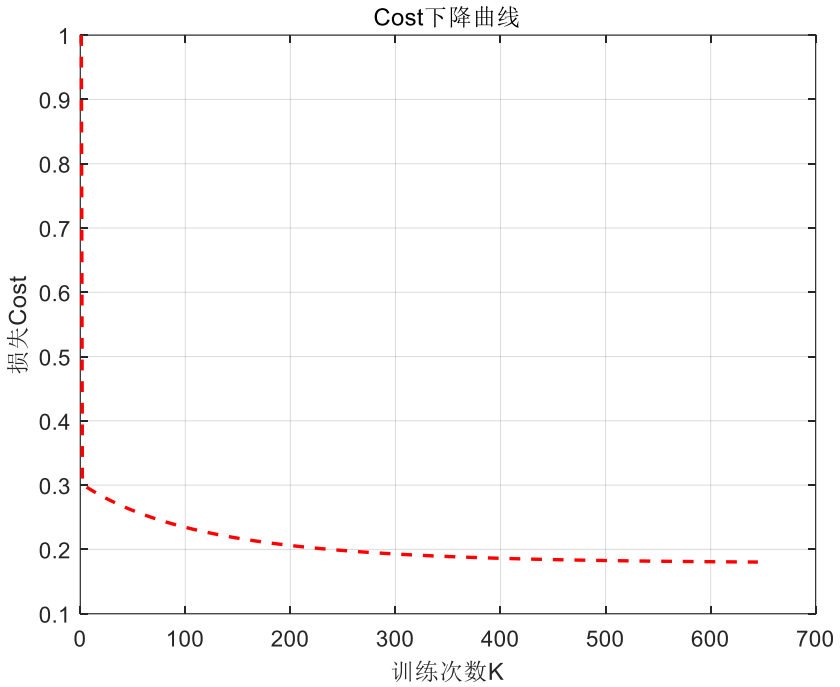
# 6. 梯度下降法

例：鸢尾花案例二分类

```
[iris,class] = xlsread('iris.xlsx');
class = class(2:end,5);
for i = 1:length(class)
    if isequal(class{i},'setosa')
        Y(i) = 0;
    elseif isequal(class{i},'versicolor')
        Y(i) = 1;
    end
end
X = iris(1:100,:);
Y = Y';
data = [X,Y];
[theta,acc,L,prey] = gradientDes_LR(data,0.01,0.2,0.00001,1000);
```

测试样本的混淆矩阵：

```
fuzzMat =
     6     0
     0    14
```




```
>> theta
theta =
    0.3477   -0.3808    0.6797    0.5405    0.1908
>> acc
acc =
     1
```

# 6. 梯度下降法



例：数据集是MostPopular Data Sets（hits since 2007）中的wine数据集，是对在意大利同一地区生产的三种不同品种的酒，做大量分析所得出的数据。第一列为类标志属性，共有三类，分别记为“1”，“2”，“3”。数据包括了三种酒中13种不同成分的数量。13种成分分别为：Alcohol，Malicacid，Ash，Alcalinity of ash，Magnesium，Total phenols，Flavanoids，Nonflavanoid phenols，Proanthocyanins，Color intensity，Hue，OD280/OD315 of diluted wines，Proline。

```
load wine.data
wine = wine(1:130,:);  %取两个类别数据
Y = wine(:,1) - 1;  %数据类别限定为0或1
X = wine(:,2:end);
>> [theta,acc,L,prey] = gradientDes_LR([X,Y],0.01,0.2,0.00001,1000);
theta =
-0.4860   -0.0360   -0.2298    0.4247   -0.1308   -0.1208   -0.2814
0.1805    0.0321   -0.3683    0.1481  -0.1810   -0.5071    0.1201
```

测试样本的混淆矩阵：

```
fuzzMat =
    17     0
     1     8
>> acc
acc =
    0.9615
```

# 6. 梯度下降法

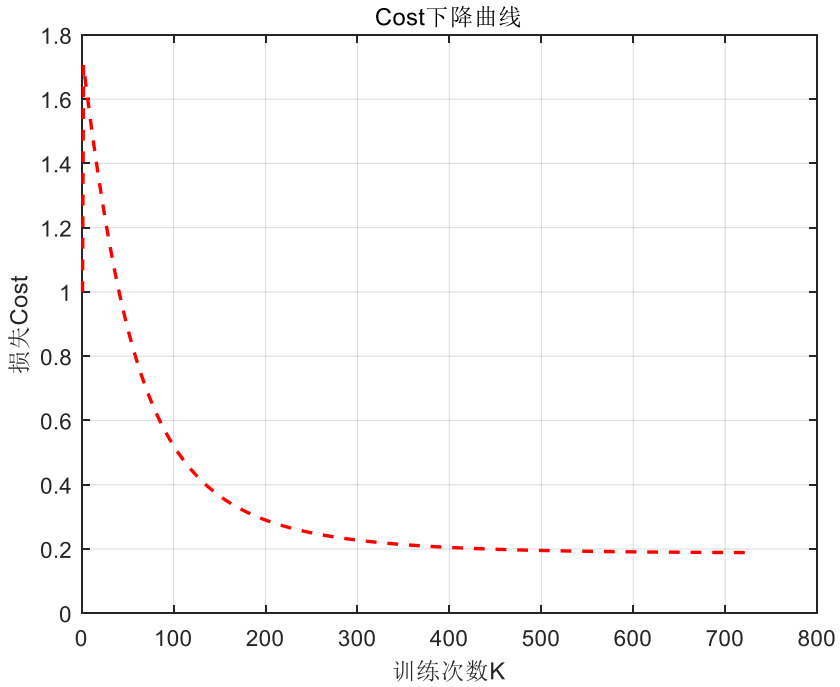
%采用交叉熵损失函数结果

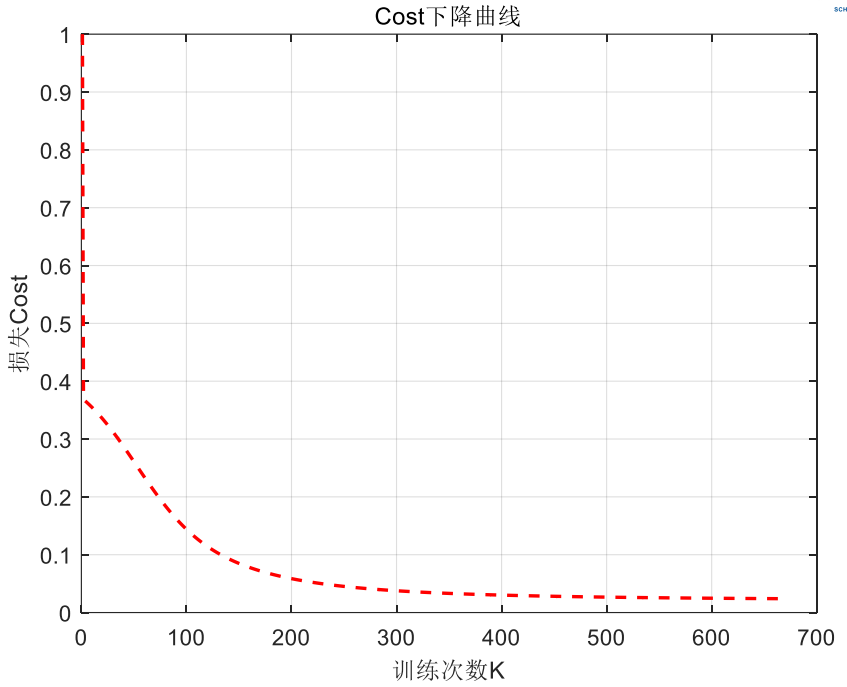
>> [theta,acc,L,prey] = gradientDes_LR([X,Y],0.01,0.2,0.00001,1000);

测试样本的混淆矩阵：

fuzzMat =

| | |
|---|---|
| 17 | 0 |
| 0 | 9 |




上图为平方和损失函数，左图为交叉熵损失函数。交叉熵损失函数的损失向量L可能存在NAN值，得不到图像。这是因为交叉熵函数中log的问题，即预测值h非常小的时候，可能存在计算的NAN。这是可采用二次项函数作为代价函数。


01
IDEA
02
LEARNING
03
CREATIVE
INSPIRATION
EDUCATION
BUSINESS
PHOTO BOOK

# 感谢聆听