11.2.2 非线性回归

1. 非线性拟合

非线性拟合的通用函数是 nlinfit(), 其基本格式为：

[beta,R] = nlinfit(X,Y,modelfun,beta0)

其中，X 为一个或多个自变量的数据，Y 为因变量数据；

modelfun 定义要拟合的含参量非线性函数，包含两个参数：自变量向量和参变量向量，再根据具体表达式写即可；

beta0 为参数初始值。

返回值 beta 为估计的回归系数，R 为残差向量，还返回其它模型诊断信息。这里最大的缺陷是，通常无法事先知道要拟合的含参量非线性函数的形式，一种办法是在 Matlab 曲线拟合工具箱中，探索各种拟合函数形式是否大致符合数据；另一种办法是使用工具软件 1stOpt，能自动搜索最优的拟合函数。

例 11.2 （非线性拟合）

混凝土的抗压强度随养护时间的延长而增加，现将一批混凝土作成 12 个试块，下表记录了养护时间 x（日）及抗压强度 y（kg/cm2）的数据：

表 11-4 混凝土养护时间与抗压强度数据

养护时间 x	2	3	4	5	7	9	12	14	17	21	28	56
抗压强度 y（$+r$）	35	42	47	53	59	65	68	73	76	82	86	99

这里，r 为 0.5 左右的测量误差。已知 x 与 y 之间存在如下的非线性关系：

$$y = a + k_1 e^{mx} + k_2 e^{-mx}$$

其中，a, k_1, k_2, m 为待估计的回归系数。

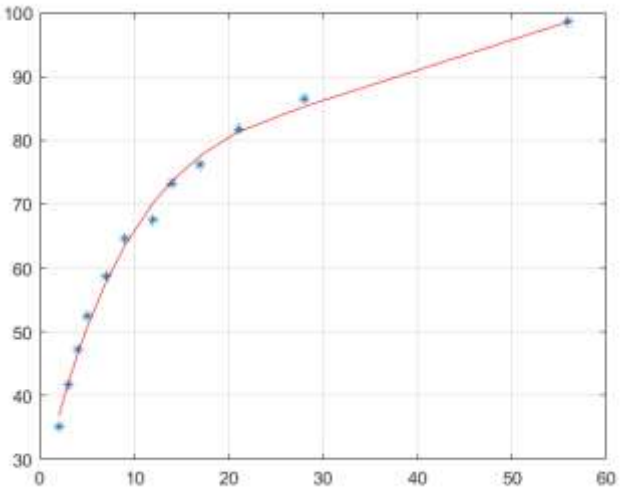
Matlab 代码

```
x = [2 3 4 5 7 9 12 14 17 21 28 56]';
r = rand(12,1) - 0.5;
```

```
y= [35 42 47 53 59 65 68 73 76 82 86 99]' + r;
fun = @(beta,x) beta(1)+beta(2)*exp(beta(4)*x)+beta(3)*exp(-beta(4)*x);
[beta,err] = nlinfit(x, y, fun, rand(1,4));
beta                                                    % 拟合系数估计
[yfit,delta] = nlpredci(fun,x,beta,r,J)
plot(x,y,'*', x, yfit,'r'), grid on
```

运行结果：

```
beta  =  88.0929   0.0303  -63.2757   0.1047
```



2. 插值拟合

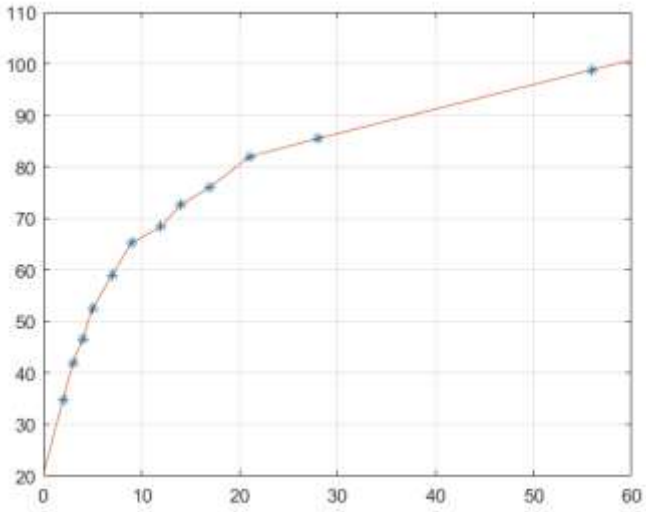
回归拟合的曲线不需要经过各个散点，只需要到各个散点的距离总和最小。插值拟合是经过各个散点，把中间的值按一定规则插补上。

例 11.3 （插值拟合）

仍以例 11.2 的数据为例，用线性插值和三次样条插值来做拟合：

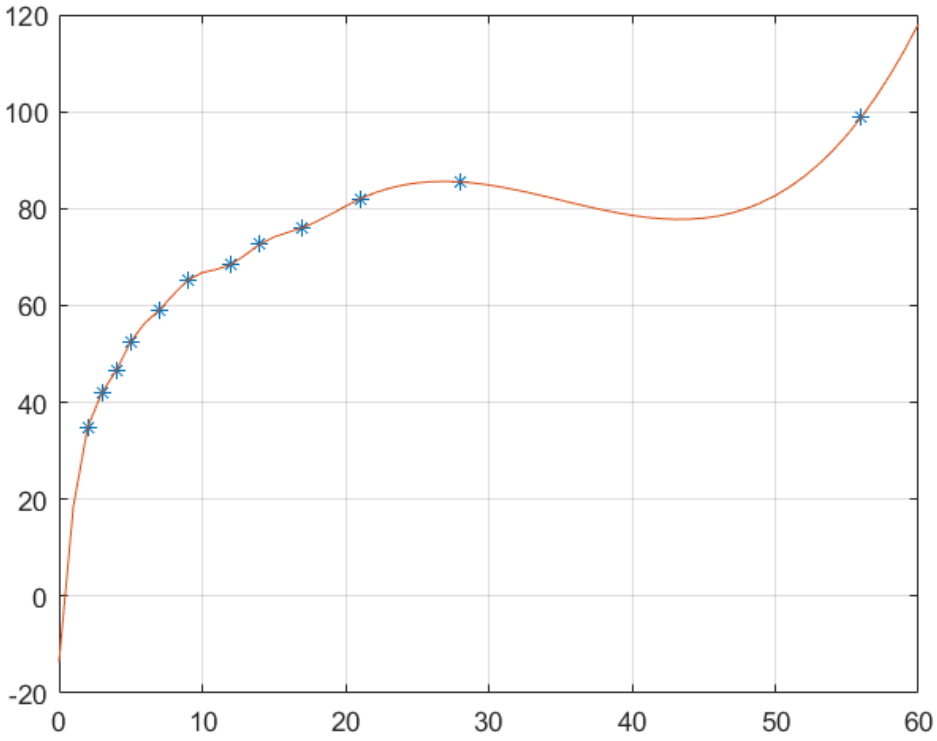
Matlab 代码

```
f1 = fit(x,y,'linearinterp');                    % 线性插值
X = 0:60;
plot(x,y,'*',X,f1(X),'-'), grid on
```



Match 代码

```
f2 = fit(x,y,'cubicinterp');                    % 三次样条插值
X = 0:60;
plot(x,y,'*',X,f2(X),'-'), grid on
```



另外，还有近邻插值 ‘nearestinterp’等。

3. 多项式回归

多项式回归是特殊的非线性函数拟合。

(1) 一元多项式回归

一元多项式是关于 1 个自变量的多项式，其一般形式为：

$$p(x) = p_1 x^n + p_2 x^{n-1} + \cdots + p_n x + p_{n+1} \tag{11-18}$$

一个自变量的回归，又想带有高次项，就可以用一元多项式回归。虽然根据泰勒公式，多项式的次数越高逼近效果越好，但是要注意，多项式回归的次数不能选太高，一般不要超过 3 次，原因是次数过高会有龙格现象，以及过拟合的问题。

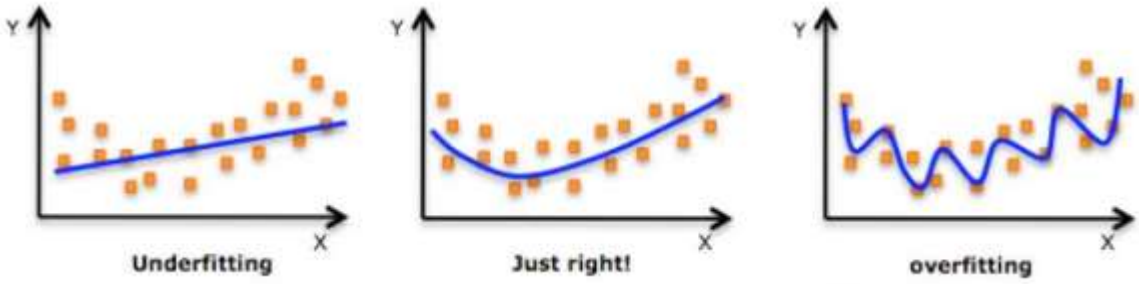


图 11-7 欠拟合、恰好拟合、过拟合

Matlab 提供了 polyfit() 函数实现一元多项式回归，其基本语法为：

```
[p,S,mu] = polyfit(x,y,n)
```

```
[y,delta] = polyval(p,x,S,mu)
```

其中，x 为自变量数据，y 为因变量数据，n 为多项式次数；返回值 p 为拟合多项式的系数向量，与式 (11-5) 对应；S 返回用来估计残差的结构，可用做 polyval() 函数的输入来获取误差估计值；mu 返回 x 的均值和标准差，用于 polyval() 对新数据做中心化和缩放。

例 11.4 （一元多项式回归）

现有我国 1995 年—2014 年总人口数据，如下表所示：

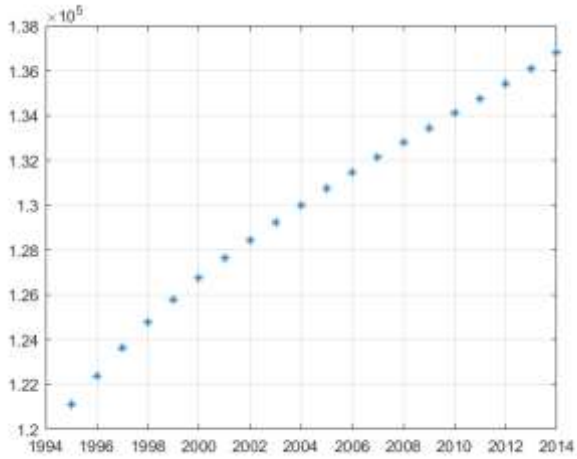
表 11-5 我国 1995 年—2014 年总人口数据

Year	Population
1995	121121
1996	122389
1997	123626
1998	124761
1999	125786
2000	126743
2001	127627
2002	128453
2003	129227
2004	129988
2005	130756
2006	131448
2007	132129
2008	132802
2009	133450
2010	134091
2011	134735
2012	135404
2013	136072
2014	136782

读入数据，绘图探索：

Matlab 代码

```
pop = readtable('我国人口数据.xlsx','PreserveVariableNames',true);
plot(pop.Year,pop.Population,'*'), grid on
```



大体像一个开口向下的抛物线，故尝试用二次多项式进行拟合：

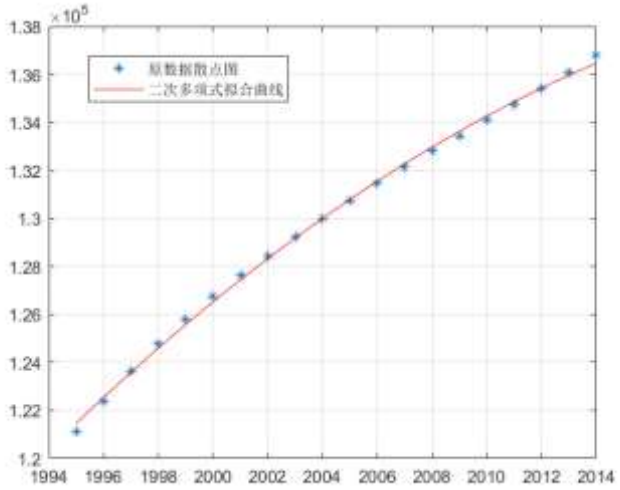
Matlab 代码

```
[P,S] = polyfit(pop.Year,pop.Population,2)                % 二次多项式回归
x1 = 1995:0.5:2014;
y1 = polyval(P,x1,S);
hold on
plot(x1,y1,'-r')
legend('原数据散点图','二次多项式拟合曲线');
polyval(P, 2015:2020)                                     % 预测 2015-2020 年的人口
```

运行结果：

```
ans    =    1.0e+05 *

   1.3694    1.3738    1.3779    1.3816    1.3851    1.3882
```



(2) 多元多项式回归

若因变量与自变量之间的关系不是线性关系，做线性回归效果往往不好。此时一种常用的改进办法，就是用原自变量数据，生成 2 次项甚至 3 次项，相当于用更高阶的泰勒公式去更好地逼近曲线。

比如有两个自变量 x_1, x_2 为例，考虑如下二元二次多项式回归：

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_1 x_2 + \beta_5 x_2^2$$

重新构造新变量：$X_1 = x_1, X_2 = x_2, X_3 = x_1^2, X_4 = x_1x_2, X_5 = x_2^2$，做 5 元线性回归即可实现，或者更简单地，在 fitlm() 函数中使用参数 ‘quadratic’或‘polyijk’ 实现。

关于交互项 x_1x_2 的回归系数 β_4 的解释

如果没有二次项，模型的回归系数可以解释为自变量对因变量的边际效应：

$$\frac{\partial y}{\partial x_1} = \beta_1$$

即，x_1 每增加 1 个单位，会带来 y 增长 β_1 个单位。

有了二次项以后，

$$\frac{\partial y}{\partial x_1} = \beta_1 + 2\beta_3 x_1 + \beta_4 x_2$$

故边际效应不再是常数，而是与 x_1, x_2 的当前值还有关系：线性函数关系。具体到交互项，如果 $\beta_4 > 0$，则边际效应还会随着 x_2 的增加而增大，相当于一种协同效应。

例 11.5 （二元多项式回归模型）

继续以例 11.1 数据为例，只考虑两个自变量：RD_Spend、Marketing_Spend，构建 Profit 的二元多项式回归模型：

Matlab 代码

```
dat = readtable('Predict_Profit.xlsx', 'PreserveVariableNames',true);
dat(:,[2,4]) = [];
lm2 = fitlm(dat,'quadratic')
```

运行结果：

```
lm2  =  线性回归模型:
Profit ~ 1 + RD_Spend*Marketing_Spend + RD_Spend^2 +Marketing_Spend^2
```

估计系数:

	Estimate	SE	tStat	pValue
(Intercept)	5.1451	0.35672	14.423	4.4366e-18
RD_Spend	0.71337	0.11508	6.1987	1.8788e-07
Marketing_Spend	0.037739	0.040871	0.92338	0.36096
RD_Spend:Marketing_Spend	0.0033742	0.0038965	0.86596	0.39132
RD_Spend^2	-0.00046877	0.0091057	-0.051481	0.95918
Marketing_Spend^2	-0.00092411	0.0013386	-0.69037	0.49367

观测值数目: 49，误差自由度: 43
均方根误差: 0.787
R 方: 0.962，调整 R 方 0.958
F 统计量(常量模型): 217，p 值 = 2.33e-29

该回归除了常数项、两个一次项外，还多了交互项 RD_Spend:Marketing_Spend，和两个二次项：RD_Spend^2、Marketing_Spend^2，根据系数估计可以列出该二元多项式回归方程（略）。

当在模型中引入高次项后，肯定会或多或少地提升模型的拟合效果，但同时也带来了副作用：**可能会产生多重共线性**，在实际使用时应当注意。

更需要注意的一点是，这些高次项可能有很多是并不显著的，所以，引入高次项与剔除不显著项是经常要一起来做的，这就需要**逐步回归**。

11.2.3 逐步回归

多元线性回归模型中，并不是所有的自变量都与因变量有显著关系，有时有些自变量的作用可以忽略。这就需要考虑怎样从所有可能有关的自变量中挑选出对因变量有显著影响的部分自变量。

比如，**例 11.1** 的回归结果中，自变量 Administration 就是不显著的。

逐步回归的基本思想是，将变量一个一个地引入或剔出，引入或剔出变量的条件是“偏相关系数”经检验是显著的，同时每引入或剔出一个变量后，对已选入模型的变量要进行逐个检验，将不显著变量剔除或将显著的变量引入，这样保证最后选入的所有自变量都是显著的。

两种极端模型是：最小模型（只有常数项）、最大模型（完全模型）。逐步回归就是从最小模型或最大模型开始，每一步只有一个变量引入或从当前的回归模型中剔除，当没有回归因子能够引入或剔出模型时，该过程停止。

Matlab 提供了 stepwiselm() 函数实现逐步回归，其参数和用法完全同 fitlm().

例 11.6 （逐步回归）

仍以例 11.1 的数据为例，把所有自变量及其二次多项式都考虑进来，通过逐步回归筛选合适的变量，构建回归模型：

Matlab 代码

```
dat = readtable('Predict_Profit.xlsx', 'PreserveVariableNames', true);
lm3 = stepwiselm(dat, 'quadratic')                    % 逐步回归
```

运行结果：

```
lm3 = stepwiselm(dat, 'quadratic')

   1。正在删除  RD_Spend:State, FStat = 0.12765, pValue = 0.88063
   2。正在删除  RD_Spend^2, FStat = 0.073652, pValue = 0.78778
   3。正在删除  Administration:State, FStat = 0.31201, pValue = 0.73405
   4。正在删除  RD_Spend:Administration, FStat = 0.78756, pValue = 0.38073
   5。正在删除  Marketing_Spend^2, FStat = 0.66432, pValue = 0.42026
   6。正在删除  Administration^2, FStat = 1.2151, pValue = 0.27725
   7。正在删除  RD_Spend:Marketing_Spend, FStat = 2.4656, pValue = 0.12444
   8。正在删除  Marketing_Spend:State, FStat = 2.2639, pValue = 0.11711
   9。正在删除  State, FStat = 0.27072, pValue = 0.76415

  10。正在删除  Administration:Marketing_Spend, FStat = 2.7908, pValue =
  0.10191
  11。正在删除  Administration, FStat = 0.26821, pValue = 0.60707

lm3 = 线性回归模型:
                    Profit ~ 1 + RD_Spend + Marketing_Spend

估计系数:
                    Estimate        SE          tStat        pValue
                   ________    _________    _______    _____________

    (Intercept)      4.9785      0.23416      21.261     1.9705e-25
    RD_Spend         0.77538     0.035029     22.136     3.6285e-26
    Marketing_Spend  0.027446    0.013042     2.1043     0.040844

观测值数目: 49，误差自由度: 46
    均方根误差: 0.769
```

R 方: 0.961，调整 R 方 0.959
F 统计量(常量模型): 568，p 值 = 3.74e-33

在 0.05 置信水平下，每个自变量都是显著的，该模型可以作为本案例的最终模型：

Profit = 4.9785+0.7754*RD_Spend+0.0274*Marketing_Spend

其它结果解读、模型诊断等同上节（略）。

11.3 广义线性模型

线性回归，要求因变量是服从正态分布的连续型数据。但实际中，因变量数据经常可能会是类别型、计数型等。

要让线性回归也适用于因变量非正态连续情形，就需要推广到广义线性模型。Logistic 回归、softmax 回归、泊松回归、Probit 回归、二项回归、负二项回归、最大熵模型等都是广义线性模型的特例。

广义线性模型，相当于是复合函数。先做线性回归，再接一个变换：

$$\mathbf{w}^T X + \mathbf{b} = u \sim \text{正态分布}$$
$$\downarrow$$
$$g(u) = y$$

经过变换后到达非正态分布的因变量数据。

一般更习惯反过来写：即对因变量 y 做一个变换，就是正态分布，从而就可以做线性回归：

$$\sigma(y) = \mathbf{w}^T X + \mathbf{b}$$

$\sigma(\cdot)$ 称为连接函数。

表 11-6 常见的连接函数和误差函数

回归模型	变换	连接函数	逆连接函数	误差
线性回归	恒等	$\sigma(y) = y$	$y = x^T\boldsymbol{\theta}$	正态分布
泊松回归	对数	$\sigma(y) = \ln(y)$	$y = \exp(x^T\boldsymbol{\theta})$	泊松分布

Logistic 回归	Logit	$\sigma(y)=\ln\frac{y}{1-y}$	$y=\frac{\exp(x^T\theta)}{1+\exp(x^T\theta)}$	二项分布
Probit 回归	Probit	$\sigma(y)=\Phi^{-1}(y)$	$y=\Phi(x^T\theta)$	Probit 分布
Gamma 回归	逆	$\sigma(y)=\frac{1}{y}$	$y=\frac{1}{x^T\theta}$	Gamma 分布
逆高斯回归	平方逆	$\sigma(y)=\frac{1}{y^2}$	$y=(x^T\theta)^{-1/2}$	逆高斯分布

注：因变量数据只要服从指数族分布：正态分布、伯努利分布、泊松分布、指数分布、Gamma 分布、卡方分布、Beta 分布、狄里克雷分布、Categorical 分布、Wishart 分布、逆 Wishart 分布等，就可以使用对应的广义线性模型。

Matlab 中用 fitglm() 函数实现广义线性回归模型，语法格式、默认规则等与 fitlm() 基本一致：

fitglm(tbl, modelspec, Name, Value)

fitglm(X, y, modelspec, Name, Value)

. 关键的区别是，通过名值对 **'Distribution', '分布名字'** 来设置因变量的分布以选择不同的广义线性模型，常用的分布名字有：

- **'normal'**：正态分布，线性回归
- **'binomial'**：二项分布，Logistic 回归，适合因变量是二分类数据
- **'poisson'**：泊松分布，泊松回归，适合因变量是计数数据
- **'gamma'**：Gamma 分布，Gamma 回归
- **'inverse gaussian'**：逆高斯分布，逆高斯回归

. 名值对 **'link', '连接函数名字'** 可设置连接函数，甚至自定义连接函数，上述分布会自动选择其默认的连接函数。

. 名值对 **'offset'** 设置偏移量，相当于拟合如下模型：

$$\sigma(y)=\text{Offset}+x^T\theta$$

这在泊松回归中很有用，比如 y 是人数服从泊松分布，想考虑人数占比作为因变量，$\ln(y/N)=x^T\boldsymbol{\theta}$ 就等同于 $\ln(y)=\ln(N)+x^T\boldsymbol{\theta}$，该 $\ln(N)$ 就是偏移量。

注：与泊松回归类似的一种回归是负二项回归，同样是针对因变量是计数数据。当个体之间相互独立时，适合用泊松回归；当个体之间存在相关性时，适合用负二项回归。

最后，广义线性回归与线性回归一样，也有回归诊断，也有可以筛选变量的逐步广义线性模型：stepwiseglm(), 用法完全是类似的。

11.3.1 Logistic 回归

Logistic 回归适合因变量是二分类数据，所以名为回归，实际上做的是分类。Logistic 回归也是机器学习中最简单的分类算法，更多的分类算法，还有决策树、随机森林、神经网络、支持向量机等。

例 11.7 （Logistic 回归）

考虑一个实验，参与者看到的表情是在恐惧表情和愤怒表情之间变化的，任务是将每个图像分类为恐惧或愤怒：

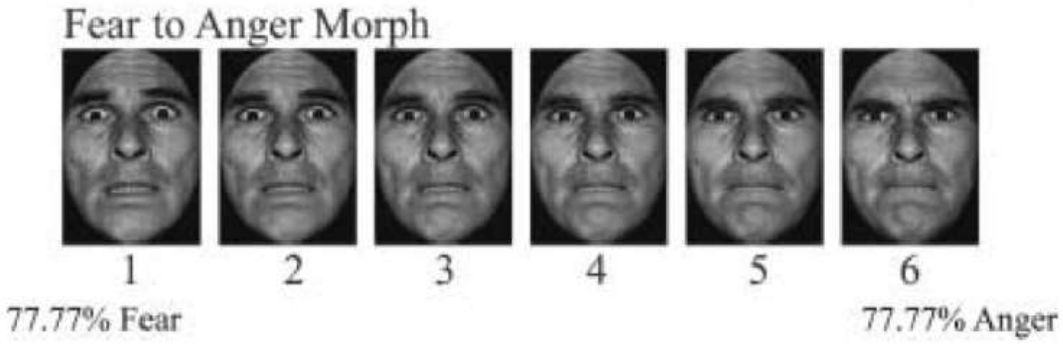


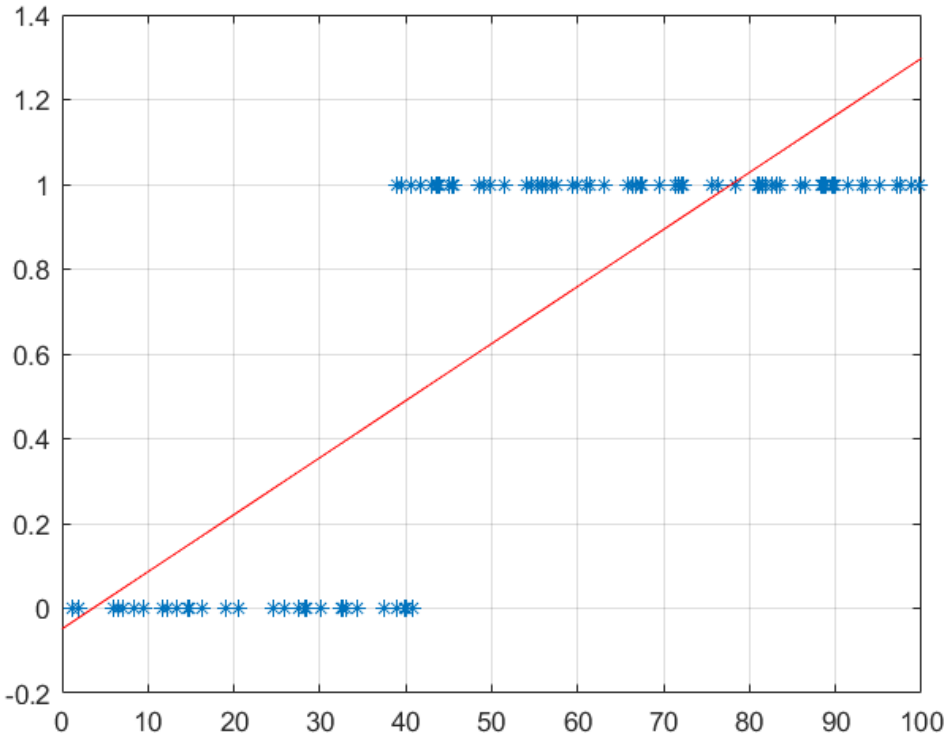
图 11-8 面部表情变化图

自变量是面部表情的量化值，因变量是二分类：将 Anger 编码为“1”，Fear 编码为“0”。先拟合线性回归模型看看：

Matlab 代码

```
dat = dlmread('FearfulAngry.txt');
x = dat(:,1);
y = dat(:,2);
```

```
lm = fitlm(x, y);                          % 拟合线性回归模型
xvals = 0:100;
yhat = predict(lm, xvals');                % 模型在新值上做预测
plot(x,y,'*',xvals, yhat,'r'), grid on
```



这显然是不合适的！“1”和“0”只是类别，不是数值，没有数值的含义。

改成预测 $P\{y=1|x\}$ 介于 0 和 1 之间，连续变化，对称……

找一个可对应它的合适的曲线：$(-\infty, +\infty) \rightarrow (0, 1)$

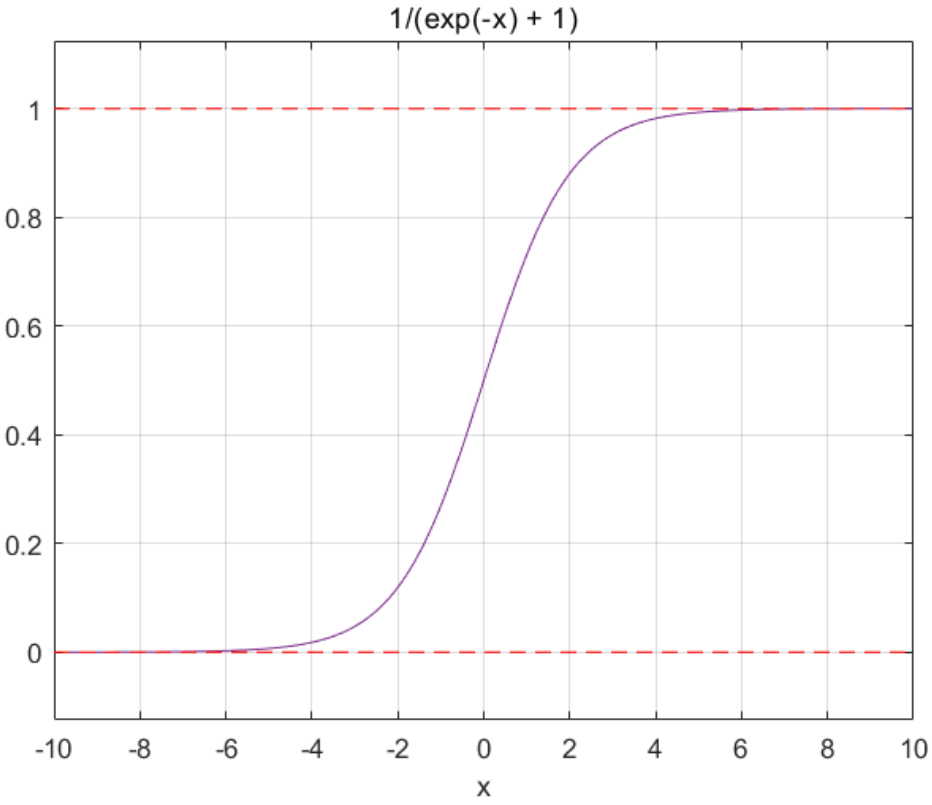
Sigmoid 曲线正合适：

Matlab 代码

```
syms x y

ezplot(1 / (1 + exp(-x)), [-10, 10]), hold on

plot([-10 10], [1,1], 'r--',[-10 10], [0,0], 'r--'), grid on
```



本来线性回归$\theta_0+\theta_1 x$的结果是$(-\infty,+\infty)$，再接一个 Sigmoid 变换，就到$(0,1)$上去了，即

$$P\{\text{"Angry"}\}=\frac{1}{1+e^{\theta_0+\theta_1 x}}$$

反过来写上式就是

$$\text{L}ogit\left(P\left(\text{"}angry\text{"}\right)\right):=\ln\left(\frac{P\left(\text{"}angry\text{"}\right)}{1-P\left(\text{"}angry\text{"}\right)}\right)=\theta_0+\theta_1 x \qquad (11\text{-}19)$$

式 (11-19) 是 Logistic 回归的一般形式。Sigmoid 变换的逆变换就是 Logit 变换，通过接这样一个 Logit 连接函数，整个逻辑就打通了，其它广义线性模型也是同样道理，只是接的连接函数不同而已。

关于 Logistic 回归系数的解释，记

$$\text{Odds}=\frac{P("angry")}{1-P("angry")}=e^{\theta_0+\theta_1 x}=e^{\theta_0}e^{\theta_1 x} \tag{11-20}$$

称为**发生比**。若自变量 x 是连续变量，看它每增加 1 个单位会如何：

$$\frac{\text{Odds}_{x_0+1}}{\text{Odds}_{x_0}}=e^{\theta_1}$$

这表示在其它变量不变的情况下，x 每增加 1 个单位，将会使得关注事件的发生比变化 e^{θ_1} 倍，注意该倍数是相对于原来 x_0 时的发生比而言的。

下面改用 Logistic 回归：

Matlab 代码

```
glm = fitglm(x, y, 'linear', 'Distribution', 'binomial');
glm
```

运行结果：

```
glm    =    广义线性回归模型:

                logit(y) ~ 1 + x1

分布 = Binomial

估计系数:
              Estimate        SE          tStat         pValue
              ________      _______      ________      _________

 (Intercept)  -37.064        19.824       -1.8697       0.061524
     x1        0.9338        0.49619      1.8819        0.059844

101 个观测值，99 个误差自由度
          散度: 1
          卡方统计量(常量模型): 113，p 值 = 2.34e-26
```

根据参数估计值，可以写出回归方程：

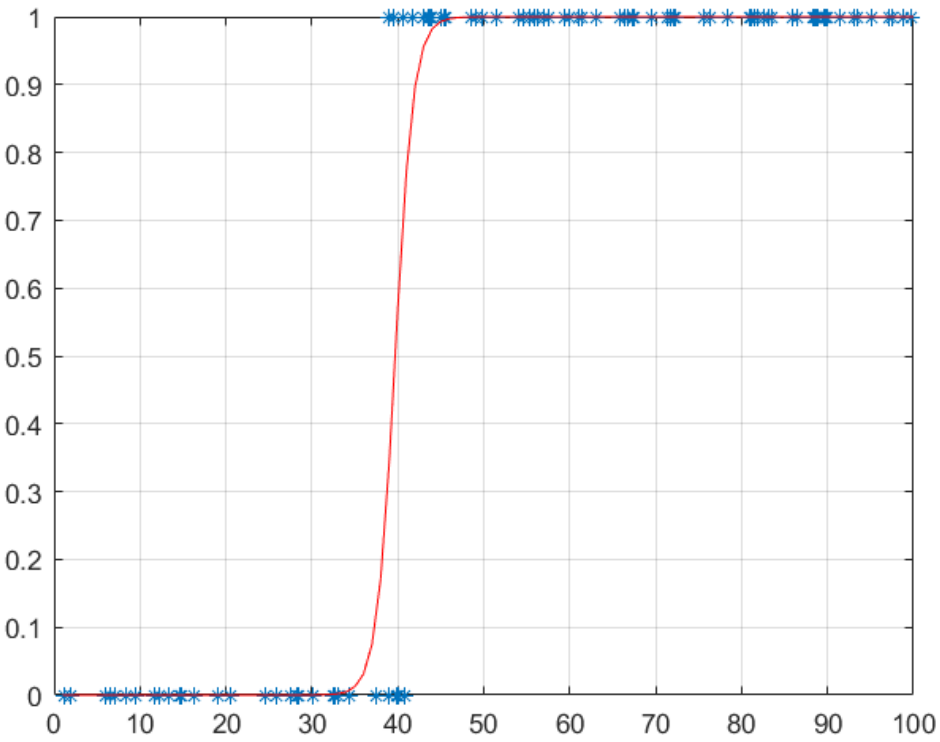
$$P\{"\text{Angry}"\}=\frac{1}{1+e^{-37.064+0.9338x}}$$

用发生比来解释，自变量 x 面部表情值每增加 1，将会使得 "Angry" 的发生比，相对于当前值，变化 $e^{0.9338} = 2.5442$ 倍。

再来看一下，模型的预测效果：

Matlab 代码

```
xvals = 1:100;
yhat2 = predict(glm, xvals');
plot(x, y, '*', xvals, yhat2, 'r'), grid on
```




```
pred = predict(glm, x);                    % 预测概率值
pred(pred >= 0.5) = 1;                     % 以 0.5 为阈值
pred(pred < 0.5) = 0;
mean(pred == y)                            % 预测正确率
```

```
ans =    0.9406
```

预测正确率为 94.06%，非常高！实际上还可以调整阈值（不一定 0.5 是最优的，这就是**调参**），进一步提高正确率。

当然，Logistic 回归也可以做模型检验、回归诊断，可以有更多的自变量，可以是连续的也可以是分类的，还可以有多项式项，也可以用逐步回归 stepwiseglm() 筛选自变量建立最合适的模型。

11.3.2 泊松回归

泊松回归，适合因变量是单位时间或空间上的计数，且大致服从泊松分布，另外还要求：

- 各个样本之间彼此独立
- 因变量数据的均值等于其方差
- $\log(y_i)$ 是自变量 **x** 的线性函数

泊松回归模型的连接函数是 ln(•)，模型可表示为：

$$\ln(y_i) = \theta_0 + \theta_1 x_{i1} + \cdots + \theta_m x_{im} \qquad (11\text{-}21)$$

例 11.8 （泊松回归）

现有美国校园暴力犯罪数据，如下表所示（部分）：

表 11-7 美国校园暴力犯罪数据

enroll1000	type	region	nv	nvrate
5.59	U	SE	30	5.366726
0.54	C	SE	0	0
35.747	U	W	23	0.643411
28.176	C	W	1	0.035491
10.568	U	SW	1	0.094625
3.127	U	SW	0	0
20.675	U	W	7	0.338573
12.548	C	W	0	0
30.063	U	C	19	0.632006

变量 enroll1000 为以千为单位的学生人数，type 为学校类型（学院/大学），region 为学校所在地区，nv 为暴力犯罪人数，nvrate 为暴力犯罪率。

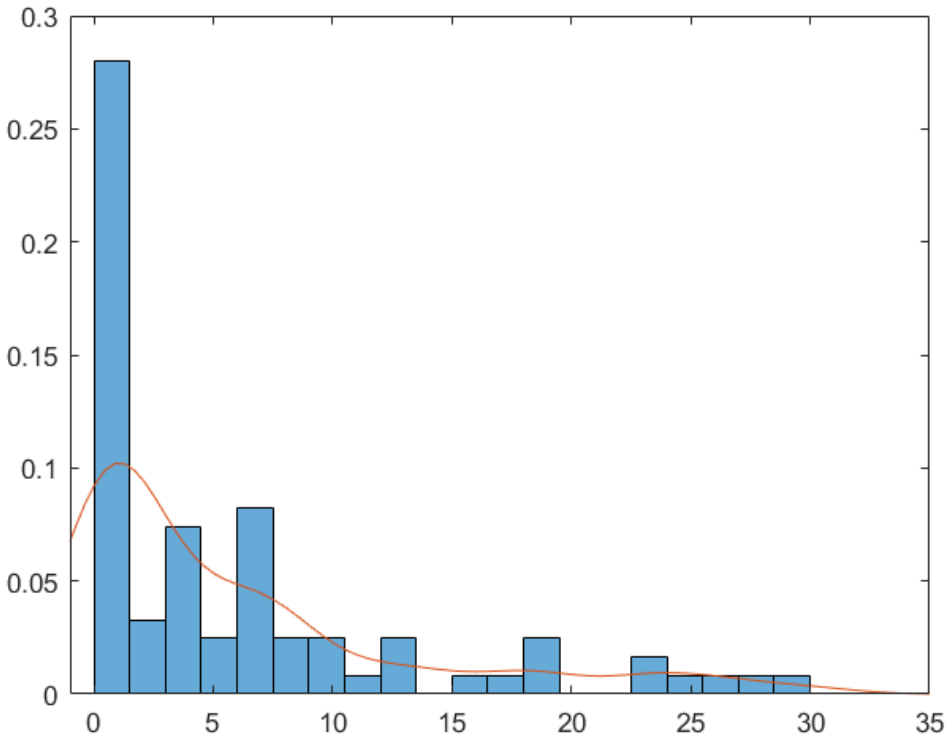
建立泊松回归模型，考察暴力犯罪与学校类型、学校所在地区之间的关系。

先读入数据，绘制直方图和核密度估计图探索因变量 nv：

Matlab 代码

```
vc = readtable('ViolentCrimes.csv', 'PreserveVariableNames', true);
histogram(vc.nv,20,'Normalization','pdf'), hold on
[f,xi] = ksdensity(vc.nv);
```

```
plot(xi,f)
axis([-1 35 0 0.3])
```



建立带偏移量的泊松回归模型，先做了一点准备工作：

- 把自变量 type 和 region 修改为分类变量，并设置水平值的顺序，位于第 1 位的水平值将作为参照组
- 偏移量是 vc.enroll1000 取对数，先计算出来，再用于模型

Matlab 代码

```
vc.type = categorical(vc.type, {'C','U'});
vc.region = categorical(vc.region,{'C','S','W','NE','MW'});
vc.log_enroll = log(vc.enroll1000);
plm = fitglm(vc,'nv~type+region', 'Distribution', 'poisson', 'offset', 'log_enroll')
```

运行结果：

```
plm     =     广义线性回归模型:

                  log(nv) ~ 1 + type + region
  分布 = Poisson
```

估计系数:

	Estimate	SE	tStat	pValue
(Intercept)	-1.5963	0.17115	-9.3267	1.0926e-20
type_U	0.33415	0.13235	2.5247	0.011581
region_S	0.74926	0.14503	5.1662	2.3895e-07
region_W	0.27223	0.18742	1.4525	0.14636
region_NE	0.78081	0.15305	5.1016	3.3687e-07
region_MW	0.099387	0.17752	0.55986	0.57558

81 个观测值，75 个误差自由度
散度: 1
卡方统计量(常量模型): 59.6，p 值 = 1.51e-11

模型结果表明，地区 Northeast、South 与参照组 Central 的暴力犯罪数有显著差异（p 值分别为 2.39e-07、3.37e-07）；

泊松回归的回归系数的解释，与 Logistic 回归的 Odds 的解释是一样的。本例自变量是分类变量，解释的时候是当前组相对于参照组的差异。例如回归系数 0.7708 意味着 Northeast 每千人中的暴力犯罪率是控制学校类型的 Central 地区的 $e^{0.7708} = 2.16$ 倍。

当然，泊松回归也可以做模型检验、回归诊断，可以有更多的自变量，可以是连续的也可以是分类的，还可以有多项式项，也可以用逐步回归 stepwiseglm() 筛选自变量建立最合适的模型。