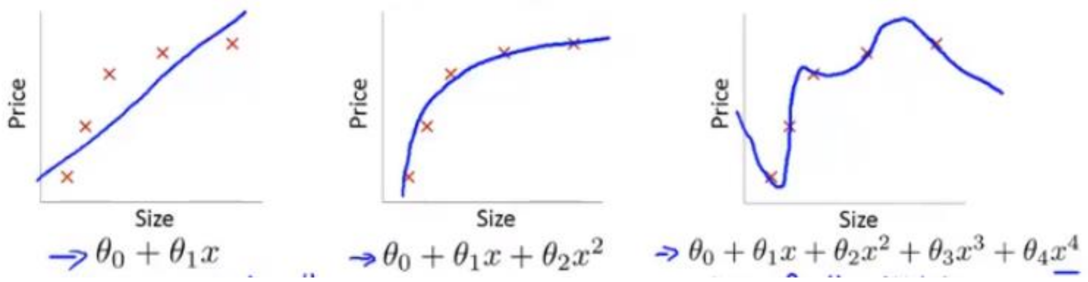
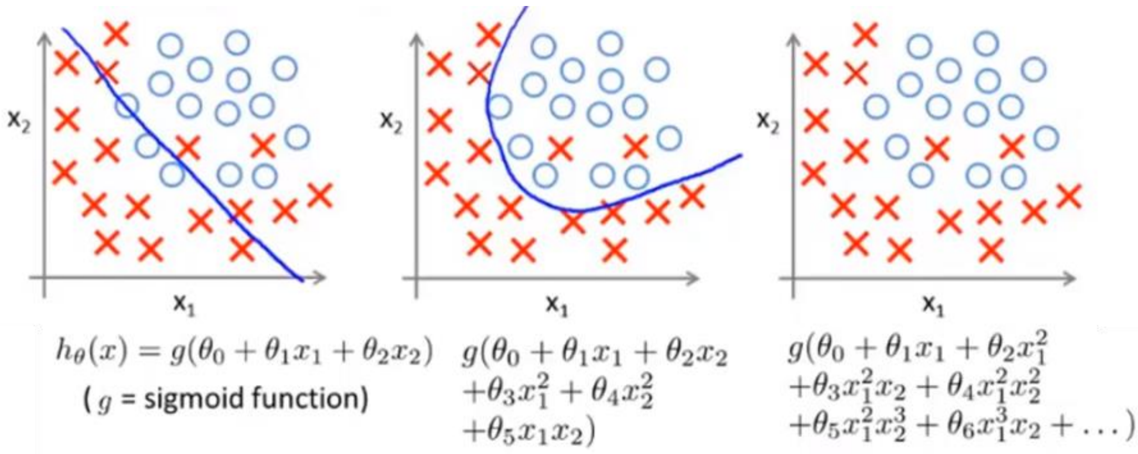
# 1、什么是过拟合

在线性回归分析中，我们针对不同特征值，只考虑了指数为一次的情况，譬如在一元一次回归分析中，画出来的拟合曲线是一条直线，通常情况下，这种拟合出来的效果不好。还以房屋面积预测房价问题，一开始为一次函数，这样效果较差，增加它的次数，拟合效果较好，再增加，发现仅拟合训练集中的数据，它能够最大化拟合，




但泛化性较差（泛化，指拟合没有出现在样本中的数据），这将可能导致对未来数据预测的不准确，以上三种情况，我们分别称作欠拟合，佳拟合，过拟合，一般的，达到中间这种情况，是拟合的较好效果，既能较好拟合训练数据，又能准确预测未来数据。

Logistic 回归同样会出现这样的问题，下面一幅图就能很好的说明问题，



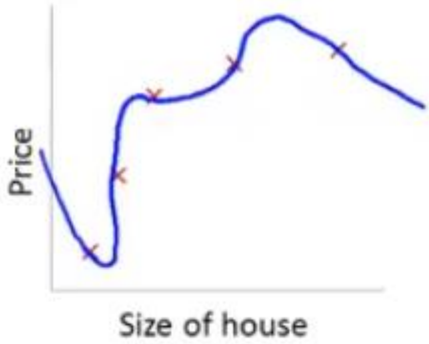

# 2、预防

通常为了拟合训练集中的数据，一开始会选择叫高次的函数，当作出这样一条曲线出来时，我们首先应将它绘制出来，看有没有出现类似上方最右边图形的情况，若有，再考虑降低次数或者直接丢弃某些特征变量。

丢弃特征变量不是一个好的做法，尤其是当所有特征值都对结果有影响时，丢弃与否就显得难以抉择。在模型选择算法中，让程序自主选择丢弃哪些变量，这是对应上述的方法。另一种是正则化，即减小量级或者参数$\theta$的大小。

# 3、正则化

在如下这幅图中，能明显看到它出现了过拟合的情况，丢弃特征值的做法这里不推荐，这里考虑减小高次特征对应的参数，即$\theta_3$ 和$\theta_4$。




$$\theta_0+\theta_1x+\theta_2x^2+\theta_3x^3+\theta_4x^4$$

在计算代价函数的时候，我们考虑给那些具有高次特征值的参数添加一个惩罚项，如下所示：

$$\frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)^2+1000\theta_3^2+1000\theta_4^2$$

这样一来，就好像人为的添加了惩罚$\theta_3$ 和$\theta_4$的因子，系数越大，惩罚度越高，这样最终得到的$\theta_3$ 和$\theta_4$也就越小，高次拟合函数现在看起来就像一个二次函数，既保证了拟合性较好（多特征值都被保留了下来），同时泛化性也较好。

之前我们是知道$x_3$和$x_4$对拟合函数有较大影响的，于是降低了其$\theta$部分。问题是我们无法提前预知哪些参数应该被缩小，为此，我们将不惜代价的减小所有的参数。修改原来的代价函数为如下：

$$J(\theta)=\frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)^2+\lambda\sum_{j=1}^{n}\theta j^2\right]$$

这样一来，我们即对所有的参数都进行了缩小处理，这里的$\lambda$是平衡参数，当$\lambda$较小时，相当于没有作明显的缩小处理，函数还是原来的那个函数，但当$\lambda$变得很大时，缩小处理就会变得很明显，有时候，缩小过头了，我们得出的拟合函数就是一条欠拟合函数，这不是想要的结果。

# 4、运用

## A、梯度下降法

我们使用正则$f(\theta)$化，无非是想让函数拟合度更高，前面计算了代价函数，这只是一个正则化思想的初步理解，实际上在运用过程中，我们所要做的就是在$\theta$更新式里添加上这个正则化的参数项。于是之前我们的$\theta$更新式变成了这样：

$$\left\{\begin{array}{ll}\theta_0=\theta_0-\alpha\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)x_0^{(i)} & \mathrm{I}\\ \theta_j=\theta_j-\alpha\left[\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)x_j^{(i)}+\frac{\lambda}{m}\theta_j\right] & \mathrm{II}\\ \theta_j=\theta_j\left(1-\alpha\frac{\lambda}{m}\right)-\alpha\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)x_j^{(i)} & \mathrm{III}\end{array}\right.$$

这里说明一下，第 2 个、第 3 个式子是一样的，仅是作了一些数学处理，使它看起来和原来的$\theta$更新式看起来差不多。由于$x_1$始终为 1，所以未对它作变换，第 3 个式子$\theta_j(1-\alpha\frac{\lambda}{m})$相比原来只是作了微小缩小化处理，因此总的$\theta$值变小了。

**B、正规方程组**

令偏导等于 0 的思想得到最终的$\theta$值，这是之前学过的内容，现在再引入正则化概念。这是我们之前学过的内容：

$\frac{\partial}{\partial\theta_j}J(\theta)=0$，——>$\theta=(\mathrm{X}'*\mathrm{X})^{-1}*\mathrm{X}'*\mathrm{y}$

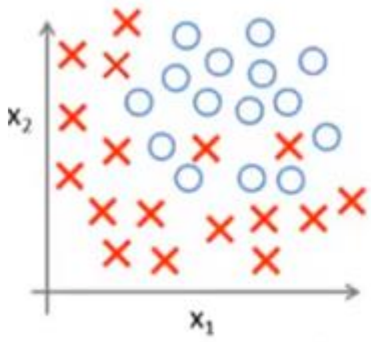
正则化：

$$\theta=\left(X^TX+\lambda\begin{bmatrix}0 & & & \\ & 1 & & \\ & & 1 & \\ & & & \ddots & \\ & & & & 1\end{bmatrix}\right)^{-1}X^Ty$$

可以得到，当$\lambda$>0，括号内矩阵一定可逆，因此也就算出了最佳拟合效果的$\theta$值。

**C、logistic 回归**

在分类学习算法中，同样会出现过拟合的现象，当某些特征值次数较高时，我们




$$g(\theta_0+\theta_1x_1+\theta_2x_1^2 +\theta_3x_1^2x_2+\theta_4x_1^2x_2^2 +\theta_5x_1^2x_2^3+\theta_6x_1^3x_2+\dots)$$

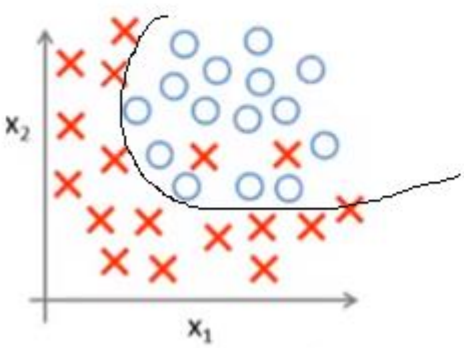
会得到过拟合函数，使得泛化性太差。

因此，所要做的就是在代价函数中添加一个惩罚项，降低次数项参数$\theta$值过高的可能性。同样，在$\theta$值更新中，运用梯度下降算法：

$$\theta_0=\theta_0-\alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})x_0^{(i)}$$

$$\theta_j=\theta_j(1-\alpha\frac{\lambda}{m})-\alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})x_j^{(i)}$$

之后得到的拟合曲线就像这样




$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \dots)$$

**D、高级梯度算法**

[theta, cost] = fminunc(@(t)(costFunction(t, X, y)), initial_theta, options)，在这样一个更新式中，给定一个初始$\theta$值，costFunction 每次迭代返回当前代价值与偏导，因此，将正则化的思想运用在 costFunction 中，是提升 fminunc 高等梯度下降算法准确性与泛化性的必要手段。

function [J, grad] = costFunction(theta, X, y)

m = length(y);

J = 0;

grad = zeros(size(theta));

y_pred = sigmoid(X*theta);

J = -1.0 / m * sum( y.*log(y_pred) + (1-y).*log(1-y_pred) )+ $\frac{1}{2m}\lambda\sum_{j=1}^{n}\theta j^2$

Grad_0 = $\alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})x_0^{(i)}$

Grad_j = $\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})x_j^{(i)}+\frac{\lambda}{m}\theta_j$

end