实验五

**function [J, grad] = linearRegCostFunction(X, y, theta, lambda)**

计算含二范数正则项的线性回归的损失函数值和偏导数

代码：

h = X \* theta;

J = (sum((h-y) .^ 2) + lambda \* sum(theta(2:end) .^ 2)) / 2 / m;

grad = (X' \* (h-y) + lambda \* theta) / m;

grad(1) = grad(1) - lambda \* theta(1) / m;

**function [error\_train, error\_val] = ...**

**learningCurve(X, y, Xval, yval, lambda)**

对于某一给定超参数lambda的模型，计算使用不同数量的样本进行训练时后的训练集和验证集误差

代码：

for i = 1:m

[theta] = trainLinearReg(X(1:i,:), y(1:i), lambda);

[error\_train(i) grad] = linearRegCostFunction(X(1:i,:), y(1:i), theta, 0);

[error\_val(i) grad] = linearRegCostFunction(Xval, yval, theta, 0);

endfor

% 对于某一次循环，相当于是用大小为i的训练集训练，因此训练完后计算训练集上的误差，也应当使用这个大小为i的训练集。而验证集上的任何一个样本对于任何一个模型来说都是陌生的，其中的个体不一定符合训练集的统计特征，但总体仍然是与训练集同分布的，因此衡量验证误差时应该对整个验证集进行计算。

% 同时因为此处计算得到的误差不是用于拟合，只是用于衡量与真实值之间的误差，所以此处的误差应仅为mse/rmse，而不需要惩罚项。

**function [lambda\_vec, error\_train, error\_val] = ...**

**validationCurve(X, y, Xval, yval)**

返回一系列不同的lambda下训练的模型的训练和验证误差（以通过选择验证集上的最小误差来选择超参数lambda）

代码：

for i = 1:length(lambda\_vec)

[theta] = trainLinearReg(X, y, lambda\_vec(i));

[error\_train(i) grad] = linearRegCostFunction(X, y, theta, 0);

[error\_val(i) grad] = linearRegCostFunction(Xval, yval, theta, 0);

endfor

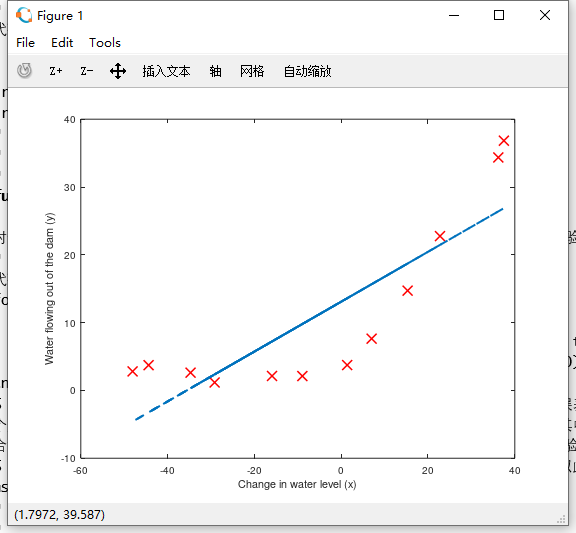
% 同样地，此处的误差不需要计算惩罚项

**Ex5.m运行结果**

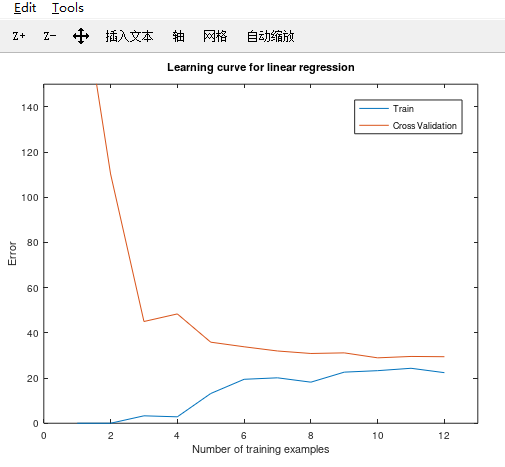
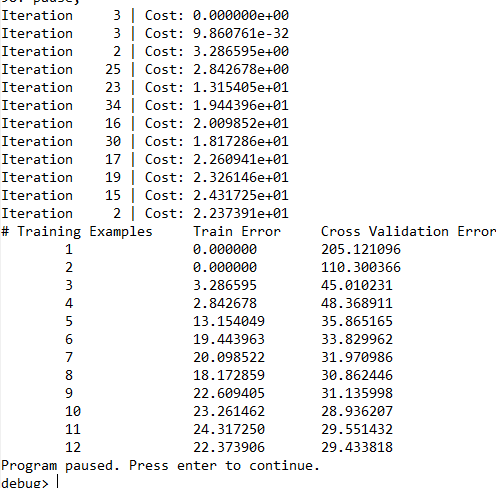
Matlab/Octave小技巧：

warning('off'); % 取消显示命令窗口的运行警告（如除以零警告）

无正则项线性回归结果：

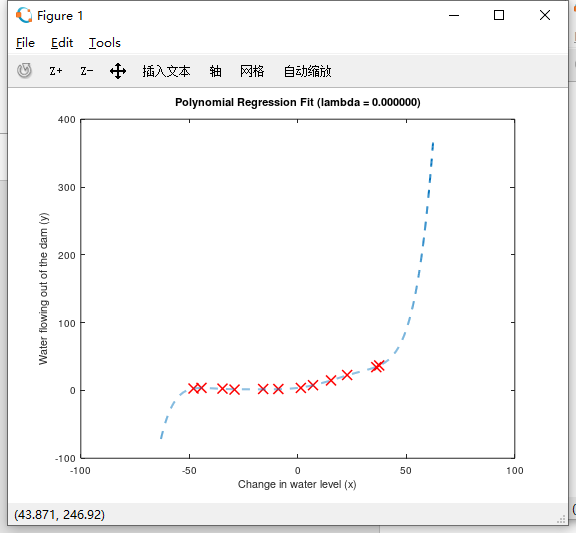


不同样本量（1-12）下的无正则项线性回归结果：

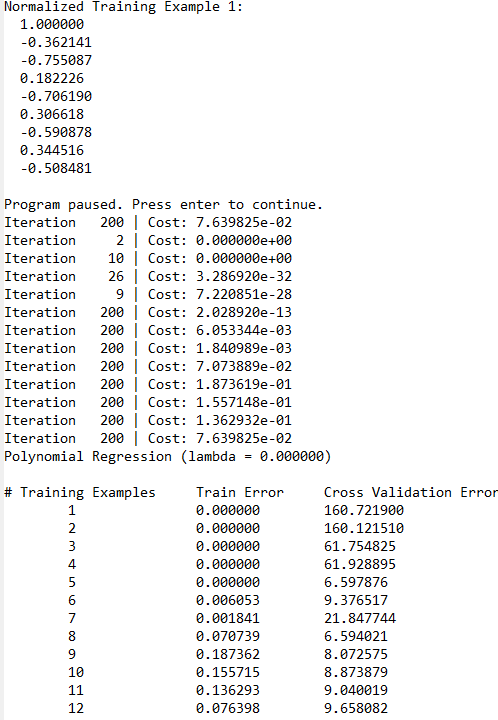
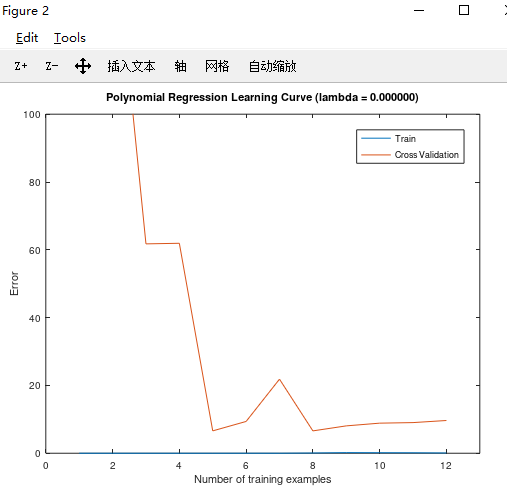


可以看到：对于线性回归，随着样本量增大，训练误差在不断变大且验证误差和训练误差相近，说明线性回归对于此问题来说表现出明显的欠拟合。

无正则化项多项式回归（最高次幂为8）结果：

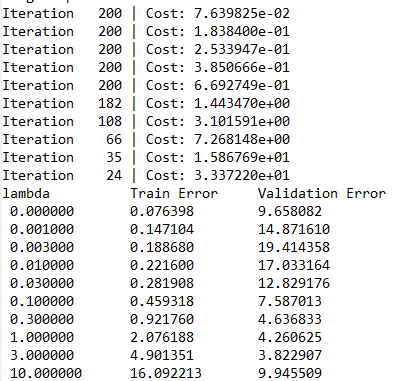
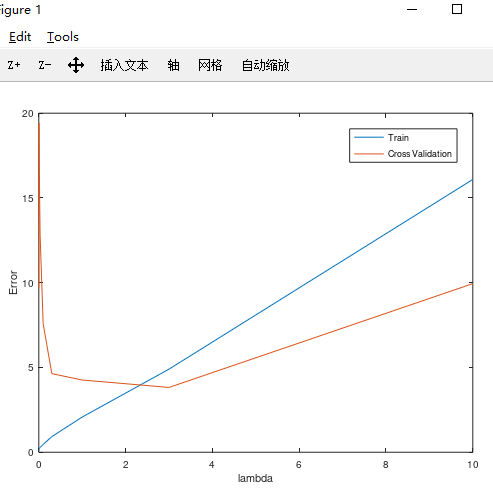


不同样本量（1-12）下的无正则项多项式回归回归结果：

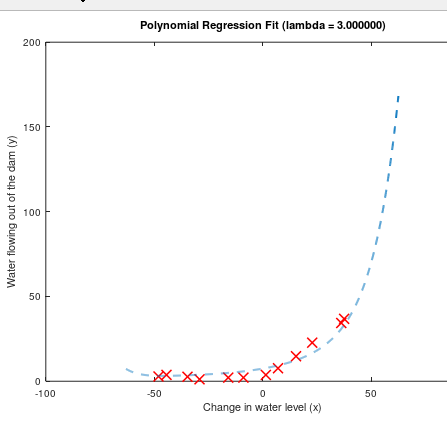
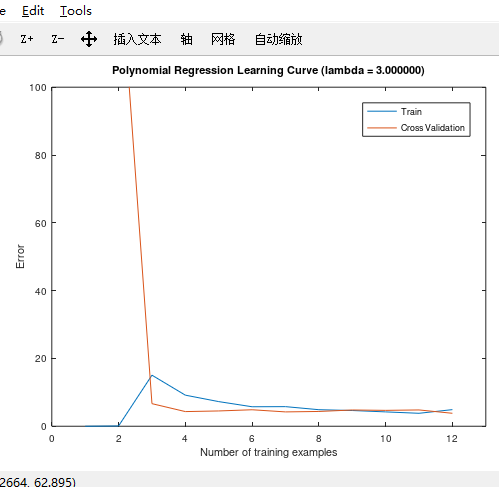
 

可以看到：对于无正则化项的多项式回归，随着样本量增大，训练误差始终保持极低的水平，而验证误差远高于训练误差，说明无正则化的多项式回归对于该问题表现出明显的过拟合。从结果图也可以看出，拟合曲线在样本空间中一旦离开样本点附近，变化便十分陡峭，可以预见其泛化能力不佳亦即过拟合。

最后我们尝试利用validationCurve这个函数在 [0 0.001 0.003 0.01 0.03 0.1 0.3 1 3 10] 这一系列lambda值中选择合适的超参数lambda：

尝试使用lambda=3再次来进行多项式拟合：

从拟合曲线来看，在训练样本之外的变化相较于无正则化时平滑了许多，但是依然不能很好的判断拟合结果。但通过误差曲线来看，当样本量达到一定大小后，训练误差不再随着样本量增大而增加并且验证误差与训练误差也相近，说明模型已经能较好地拟合。

至此便完成了对于某一种模型的一个比较完整的机器学习流程，即通过交叉验证对某一模型寻找合适的超参数进行训练。之后可以再选择其他模型使用同样的流程，然后比较多个模型的测试集上的结果，再进行进一步的模型选择或模型融合。