

大连理工大学

机器学习编程作业

作者：李飞腾

学号：31201003

院系：数学科学学院

日期：2013/4/25

编程作业四道题

以下作业，均为自主完成。部分代码曾以博客等形式流出。

题 1: Gradient descent for Linear Regression

相应函数: GradientDescentLinearReg.m %梯度下降法计算参数theta的函数

测试样例: homework01.m

run homework01

算例: $x=[3 \dots 15]$ $y=y = \text{theta0} + \text{theta1} \cdot x + 0.8 \cdot (\text{rand}(1,m) - \text{rand}(1,m));$

得到规范的X, Y后, 调用

Option.a = a %Learning Rate

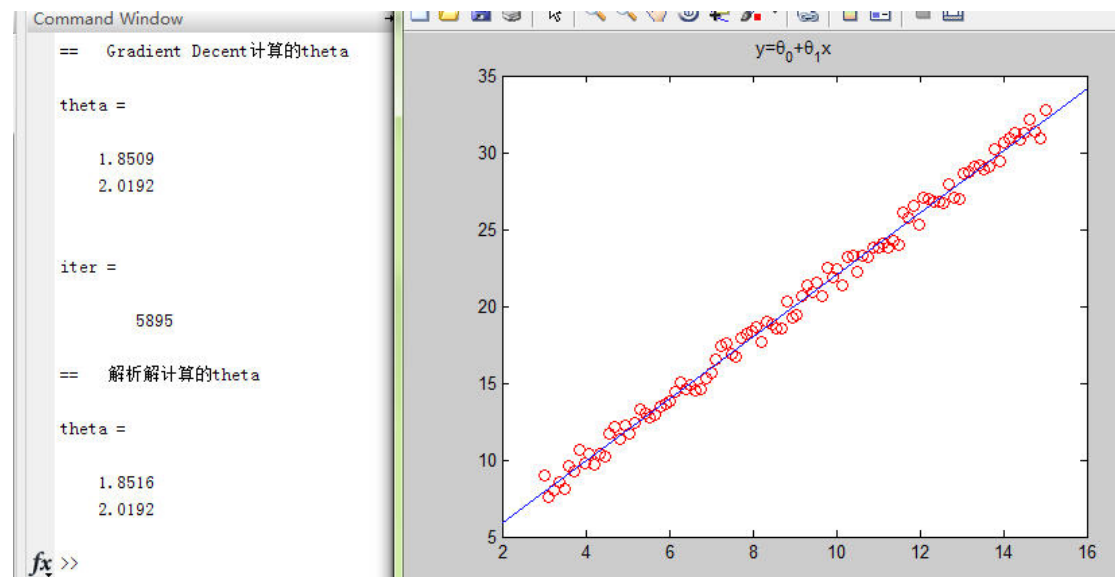
Option.eps = $3e-4$ % 收敛阶设定

% option.lambda = %自由选择, lambda即加入正则项

[theta iter] = GradientDescentLinearReg(X,Y, option)

来计算 theta的值。

得: <仅提供了一维 x 图例; 算法支持任意维 x。>



题 2 overfitting and Regularization

算例: homework02overfit.m homework02Regularization.m

```
x = linspace(1, 10, 20);
```

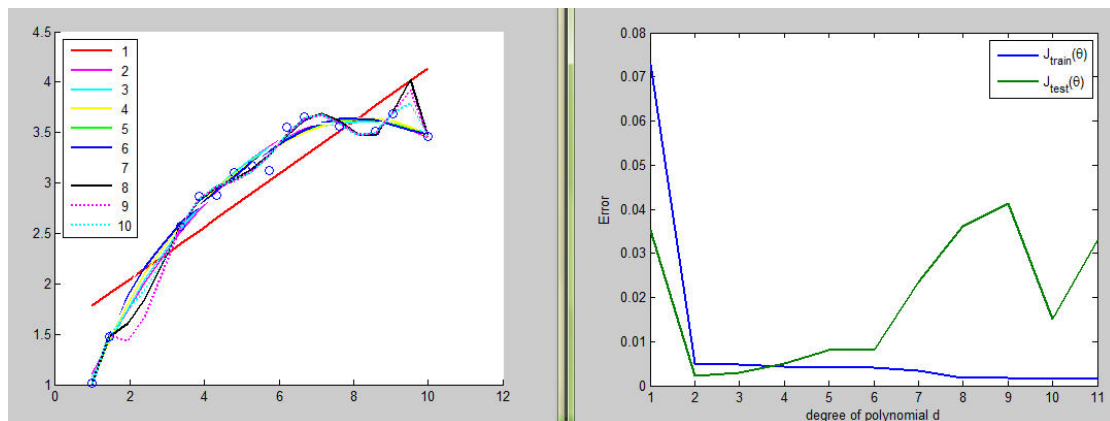
```
y = 0.1*(-1/2*x.^2+8*x+4); %
```

使用多项式 回归 数据点集(xi,yi)

homework02overfit.m 随机选取14个点作为训练点集，剩余6个作为测试点集

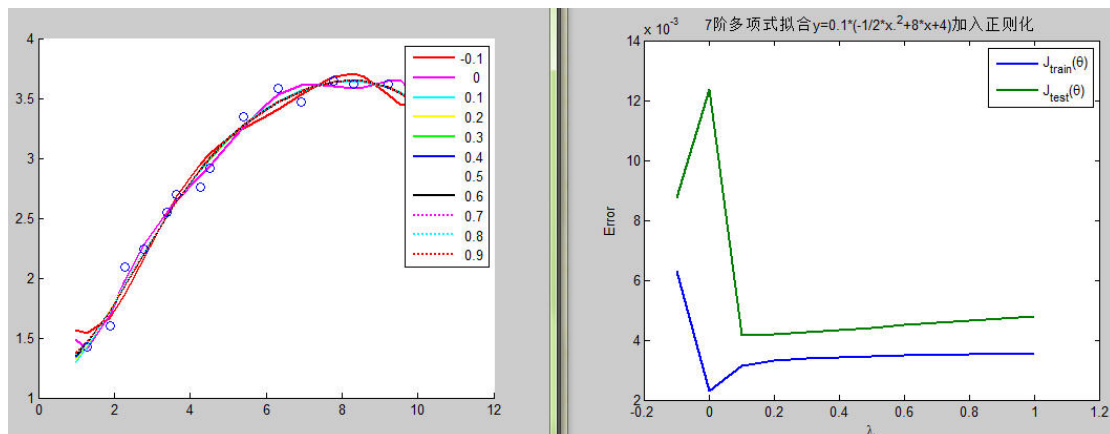
分别用 1 阶、2 阶、...、11 阶多项式来 回归 数据点集

结果如下图：可见在 2 阶 训练误差 和测试误差都较小。阶数增大，训练误差会继续减小、测试误差将增大。跟预期相符。



homework02Regularization.m 使用 7 阶多项式 回归 数据点集 加入正则化项,变化 lambda 的取值。

运行结果如下: 可见 lambda=0 时<品红色线>, 曲线不够平滑。Lambda=0.1 时测试误差最小。



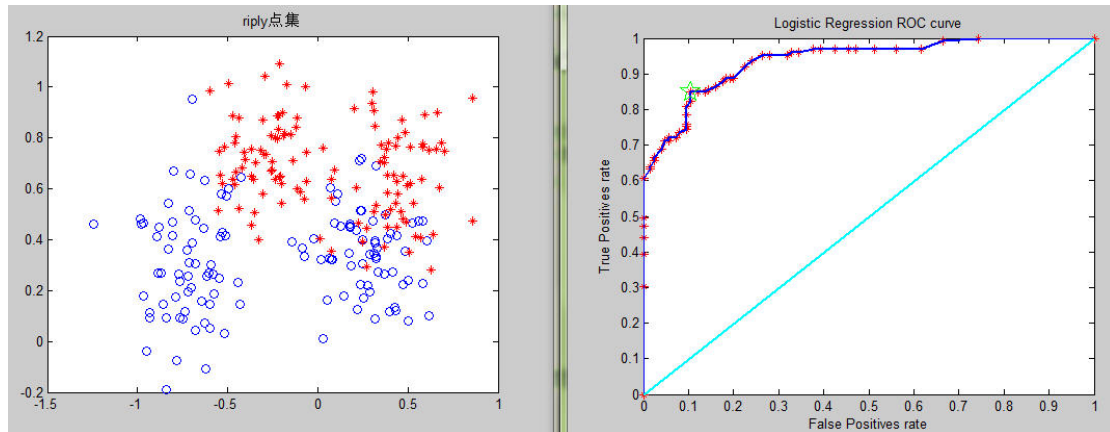
注：由于我换了组数据，所以梯度下降法的参数 a 没有时间调；用了解析解计算 theta。感觉固定步长这事不靠谱，收敛性无法保证。为什么不用精确一维线搜索来确定步长呢？虽然麻烦些，但是收敛能保证。

题 3 Logistic Regression 的 ROC 曲线

相关文件: homework03.m GradientDescentLogisticReg.m plotROC.m

算例: 使用线性不可分数据 riply_trn.mat 2 类别点集 使用 Logistic Regret 计算出概率类别 pro, 然后选择 0: 0.02: 1 的不同 lambda 作为阈值, 来判断类别。最优的 lambda 对应的点用绿色五角星标出。

run homework03.m 可得下图:



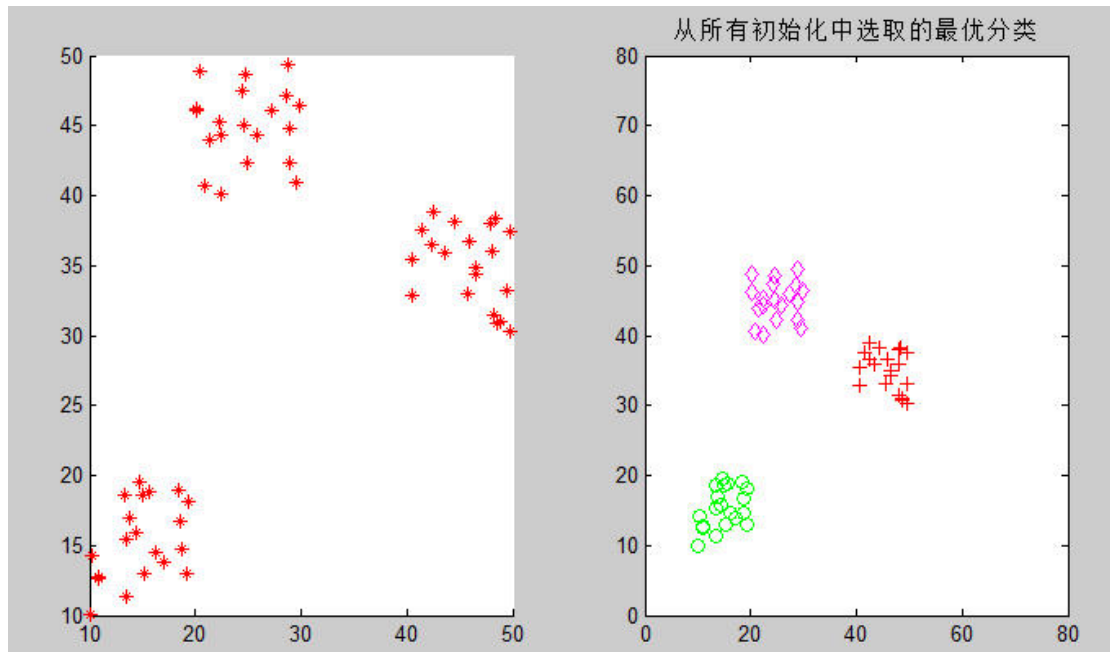
题 4 K-Means 和 EM for GMM

相关函数: KM.m gmmEM.m

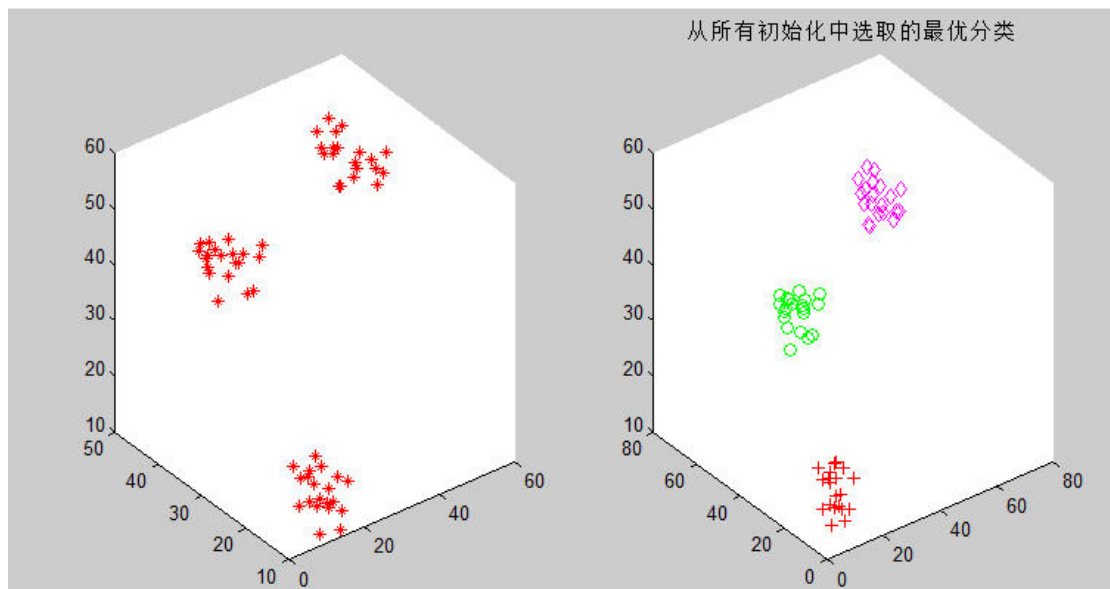
测试样例: homework04GMM.m homework04KM.m

这里 KM 和 gmmEM 都是我以前写的, 直接拿来用了, 符号跟课上用的可能不同。

KM 聚类二维点: KM 使用多次随机选取初始中心来聚类然后选取误差最小的聚类结果



KM 聚类三维点:



EM_for_GMM: 使用了数据 `krkonose.mat` 包含3个不同高斯分布数据集

run homework04GMM 得到输出: 可见用 EM 估计的均值 μ 与真实值很接近

<注意不同的运行尝试, EM_mu 值的顺序可能变化>

```
EM_mu =

    -1.0742    2.0350   -2.9530
    -1.1238    1.9229    3.0589

true_mu =

    -1     2    -3
    -1     2     3
```

协方差部分: true_E VS EM_E

```
val(:, :, 1) =

    1.0000    0.5000
    0.5000    1.0000

val(:, :, 2) =

    1.0000   -0.5000
   -0.5000    1.0000

val(:, :, 3) =

    1.0000    0.2000
    0.2000    1.0000
```

EM_E <2x2x3 double>

```
val(:, :, 1) =

    0.8536    0.3327
    0.3327    0.6747

val(:, :, 2) =

    0.9158   -0.4868
   -0.4868    0.9606

val(:, :, 3) =

    1.0335    0.0380
    0.0380    0.9571
```