学习小组 2 机器学习 第一次作业

王锦宏、吴泽辉、陶少聪、王海东、叶泽林、陈耀顺

（中山大学智能工程学院，广东 深圳，518107）

摘要：为销售某商品，分别向电视(TV)，广播(radio)和报纸(newspaper)三种媒体投放广告。数据集描述了给定三种媒体广告投放预算下的商品销售量(sales)。我们对数据集使用了 10 折交叉验证，训练了一个线性回归模型，预测了商品销售量与三种媒体广告预算之间的关系。

关键词：线性回归 (linear regression)

1 描述 10 折交叉验证对数据集的处理

所谓交叉验证法，就是将一个数据集分为 K 份，然后取其中一份作为测试集，剩余 K-1 份作为训练集。然后，取另一份作为测试集，其余 K-1 份作为训练集。如此循环，直到每一份都做过测试集为止。本题当中采取的是 10 折交叉验证，即取 k=10 时对数据进行处理。每次试验都会得出相应的正确率（或差错率）。10 次的结果的正确率（或差错率）的平均值作为对算法精度的估计。一般还需要进行多次 10 折交叉验证（例如 10 次 10 折交叉验证），并且采取不同的划分数据集的方式，减小因样本划分不同而产生的误差。再求其均值，作为对算法准确性的估计。同时十折交叉验证法作为一种对数据处理结果的评估算法，也可以用于对不同模型拟合结果的一种选择。

另外在本题的数据预处理中，我们将三个输入参量进行缩放，调整为了相同数量级，使得梯度数量级接近，训练结果更加可靠：

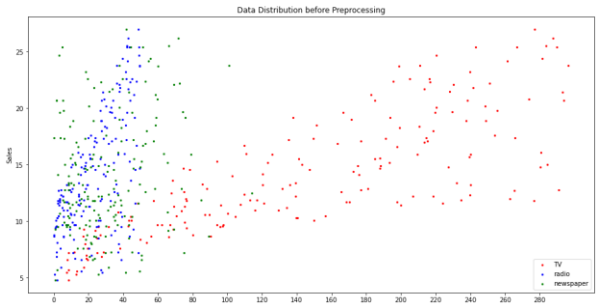


图 1 调整前

图 2 调整后

2 描述所使用的线性模型

2.1 所使用的模型：多元线性回归

对于输入 x 的利用 f(x)来估计输出 y

$$f(x) = \omega^T x + b$$

2.2 参数确定

用最小二乘法来对ω和 b 进行估计，为了方便计算和讨论，将ω和 b 变成一个向量形式：

$$\hat{\omega} = (\omega;\ b)$$

相应的把原来的数据集扩充为x，即：

$$x = \begin{pmatrix} {x_1}^T & 1 \\ {x_2}^T & 1 \\ \vdots & \vdots \end{pmatrix}$$

把标记也写成向量形式$y = (y_1; y_2; \cdots)$，故可得出均方误差为：

$$E_{\hat{\omega}} = (y - x\hat{\omega})^T (y - x\hat{\omega})$$

我们试图让均方误差最小，即：

$$\hat{\omega}^* = \underset{\hat{\omega}}{argmin}(y - x\hat{\omega})^T (y - x\hat{\omega})$$

让$E_{\hat{\omega}}$对$\hat{\omega}$求偏导使之为零：

$$\frac{\partial E_{\hat{\omega}}}{\partial \hat{\omega}} = \frac{\partial (y^T y - \hat{\omega}^T x^T y + x\hat{\omega} y^T + \hat{\omega}^T x^T x\omega)}{\partial \hat{\omega}} = 0$$

从而

$$\hat{\omega}^* = (x^T x)^{-1} x^T y$$

而$x^T x$可能不是满秩矩阵，故采用正则化或梯度下降处理。

3 描述训练模型所使用的算法

3.1 正则化(请见 MATLAB 版本 regularization.m)

$$E_{\hat{\omega}} = (y - x\hat{\omega})^T (y - x\hat{\omega}) + \frac{\lambda}{2}\|\hat{\omega}\|^2$$

$$\hat{\omega}^* = \underset{\hat{\omega}}{argmin}(y - x\hat{\omega})^T (y - x\hat{\omega}) + \frac{\lambda}{2}\|\hat{\omega}\|^2$$

$$\frac{\partial E_{\hat{\omega}}}{\partial \hat{\omega}} = 0$$

$$\hat{\omega}^* = (x^T x + \lambda I)^{-1} x^T y$$

而$x^T x + \lambda I$一定是可逆的，故可以解出解析解。

3.2 梯度下降算法

(1) 取初值$\hat{\omega}^{(0)}$=[0.1;0.1;0.1;0.1]，置 k=0;

(2) 计算$E_{\hat{\omega}}{}^{(k)} = (y - x\hat{\omega}^{(k)})^T (y - x\hat{\omega}^{(k)})$;

(3) 计算梯度$\nabla E(\hat{\omega}^{(k)})$，并且一维搜索确定$\eta_k$得到$\underset{\eta_k \geq 0}{min} E(\hat{\omega}^{(k)} + \eta_k \times (-\nabla E(\hat{\omega}^{(k)}))$

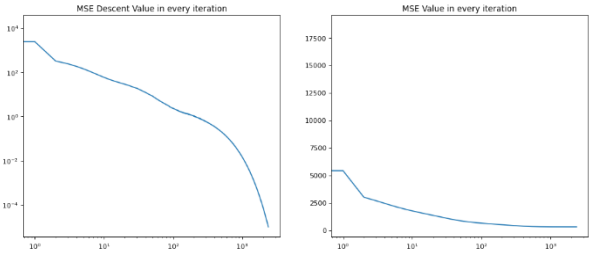
(4) $\hat{\omega}^{(k+1)} = \hat{\omega}^{(k)} + \eta_k \times (-\nabla E(\hat{\omega}^{(k)})$，并计算$E(\hat{\omega}^{(k+1)})$，当$||E(\hat{\omega}^{(k)}) - E(\hat{\omega}^{(k+1)})|| < \varepsilon$时，令$\hat{\omega}^* = \hat{\omega}^{(k)}$，停止迭代

(5) 否则置 k=k+1,返回(2)

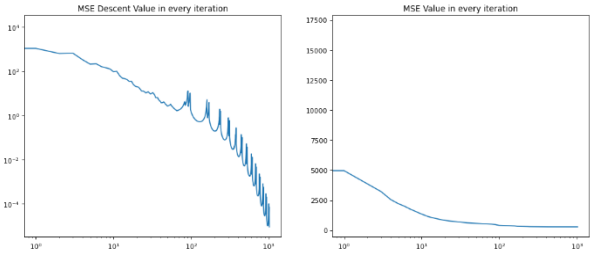
4 分析模型训练结果

4.1 训练过程的总结

我组提交的 T1.ipynb 在进行拟合的时候可以对模型的训练情况进行画图。由于篇幅关系，全部的训练过程图此处不附。取使用源数据训练模型，并用模型验证源数据的情况：



全数据模型在 1465 次迭代后达到收敛，在经过根号处理第一列后收敛次数变为 2390 次。验证自身的均方误差在修改前约为 416.57，修改后为 323.64。同时我小组在研究中发现，一些训练数据集在训练过程中，出现了梯度下降值震荡的情况：



十折训练组在根号处理第一列之前平均 1300 次左右达到收敛，处理后变为约 2100 次，其中也出现了一个 1200 次收敛的特例。

4.2 十折交叉训练结果

训练组	拟合的ω	均方误差（训练）	均方误差（测试）
1	[0.229, 0.188, -0.013, 0.666]	316.63	101.66
2	[0.227, 0.190, -0.014, 0.677]	373.49	44.03
3	[0.219, 0.187, -0.016, 0.718]	374.27	44.29
4	[0.220, 0.193, -0.015, 0.693]	359.37	58.46
5	[0.226, 0.193, -0.017, 0.651]	398.45	19.32
6	[0.223, 0.192, -0.009, 0.651]	377.51	40.41
7	[0.225, 0.189, -0.015, 0.675]	400.46	16.61
8	[0.222, 0.196, -0.023, 0.698]	361.30	57.22
9	[0.229, 0.188, -0.007, 0.644]	384.56	34.27
10	[0.226, 0.190, -0.012, 0.656]	391.32	25.62
全数据	[0.225, 0.191, -0.014, 0.672]	416.57	

4.3 不同数据预处理方式的训练结果

根据对源数据的分析，我们发现参数一['TV']可能存在一个二次情况，为了辅助线性拟合，我们将 TV 列取平方根。均方误差 1 是指用原始数据作线性拟合的结果；均方误差 2 是指对第一列数据作开根处理后再作线性拟合的结果，可以看到这一推理具有正确性，十折训练集的均方误差出现

明显降低：

训练组	均方误差 1	均方误差 2
1	316.625	258.423
2	373.493	299.660
3	374.268	287.851
4	359.365	269.492
5	398.445	305.714
6	377.513	292.402
7	400.462	299.064
8	361.300	296.472
9	384.558	296.364
10	391.319	294.724
全数据	416.573	323.244

可以明显看出，对第一列数据做非线性变换，这里是做根号处理之后，均方误差明显下降。

5 总结模型训练过程中的收获

5.1 基础知识

加深了对 python 的熟悉，如学习了 dataframe 等机器学习常用的数据储存方式，加强了 pandas、numpy、matplotlib 等库的使用；回顾了课本上的数学公式；将算法用代码实现，并用 python 完成矩阵运算，提升了编程能力；任务管理上，训练了 JupyterNotebook 集成环境的使用、利用 Github 完成了小组合作。

5.2 数据处理

学习了数据可视化时，对多变量情况需要进行归一化，所得图片才有对比价值；面对类别样本数不平衡问题，使用阈值移动处理，模型能有更好结果。

5.3 模型调试

学习步长的选取，太大会无法收敛，太小会使训练速度过慢；调试中建议选取较小的学习步长，虽然学习时间较长，但能避免不收敛的情况。

5.4 代码改进：

使用矩阵运算，相比循环运算，能有效的提高效率；尝试了除具体迭代外，其他的判准，尝试了对模型效率的提高。