多项式拟合曲线实验报告

一、实验目标

掌握最小二乘法求解（无惩罚项的损失函数）、掌握加惩罚项（2 范数）的损失函数优化、梯度下降法、共轭梯度法、理解过拟合、克服过拟合的方法(如加惩罚项、增加样本)。

二、实验要求

1. 生成数据，加入噪声；
2. 用高阶多项式函数拟合曲线；
3. 用解析解求解两种 loss 的最优解（无正则项和有正则项）
4. 优化方法求解最优解（梯度下降，共轭梯度）；
5. 用你得到的实验数据，解释过拟合。
6. 用不同数据量，不同超参数，不同的多项式阶数，比较实验效果。
7. 语言不限，可以用 matlab，python。求解解析解时可以利用现成的矩阵求逆。梯度下降，共轭梯度要求自己求梯度，迭代优化自己写。不许用现成的平台，例如 pytorch，tensorflow 的自动微分工具。

三、实验内容

1.生成数据并加入高斯噪声

生成的数据包括训练集和测试集，其中训练集为 10 个点，而测试集为 100 个点，具体方案如下：

1.规定原始函数为$y = \sin(2\pi x)$，取区间$[0,1]$为其一个周期；

2.在区间$[0,1]$上等距离的取 10 个点，在每个点对应的y值上加入一个均值为 0，方差为 0.25 的高斯噪声用来模拟训练集；

3.同样地，在区间$[0,1]$上等距离的取 100 个点，用同样的方法为每个点对应的y值加上高斯噪声作为测试集；

2.为数据增加高次特征

实验的目标是使用多项式来拟合曲线$y = \sin(2\pi x)$，目标函数可以表达为$Y = X * \theta$ 其中Y, X, θ均为向量形式。对于m个数据点，要用n次多项式曲线来拟合，有Y为$m \times 1$的列向量，X为$m \times (n+1)$的矩阵，θ为$(n+1) \times 1$的列向量。

对于矩阵X，其第一列全为 1，其之后的第i列是m个数据点的i次方，其形式大致如下：

$$\begin{bmatrix} 1 & x_1 & \cdots & x_1^n \\ 1 & x_2 & \cdots & x_2^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_m & \cdots & x_m^n \end{bmatrix}$$

因此首先要为数据增加高次特征再进行拟合。

3.实验基本原理

我们用最小二乘法的方式来衡量拟合曲线的优劣程度，最小二乘法的公式如下:

$$J(\theta) = \frac{1}{2m}(X\theta - y)^T(X\theta - y)$$

在斯坦福大学的 CS229 课程中证明了在满足高斯分布的条件下，上式越小，曲线拟合的效果越好，即通常所说的代价函数，这样一来，判断曲线拟合优劣的问题就可以转化为使上式取最小值的问题。

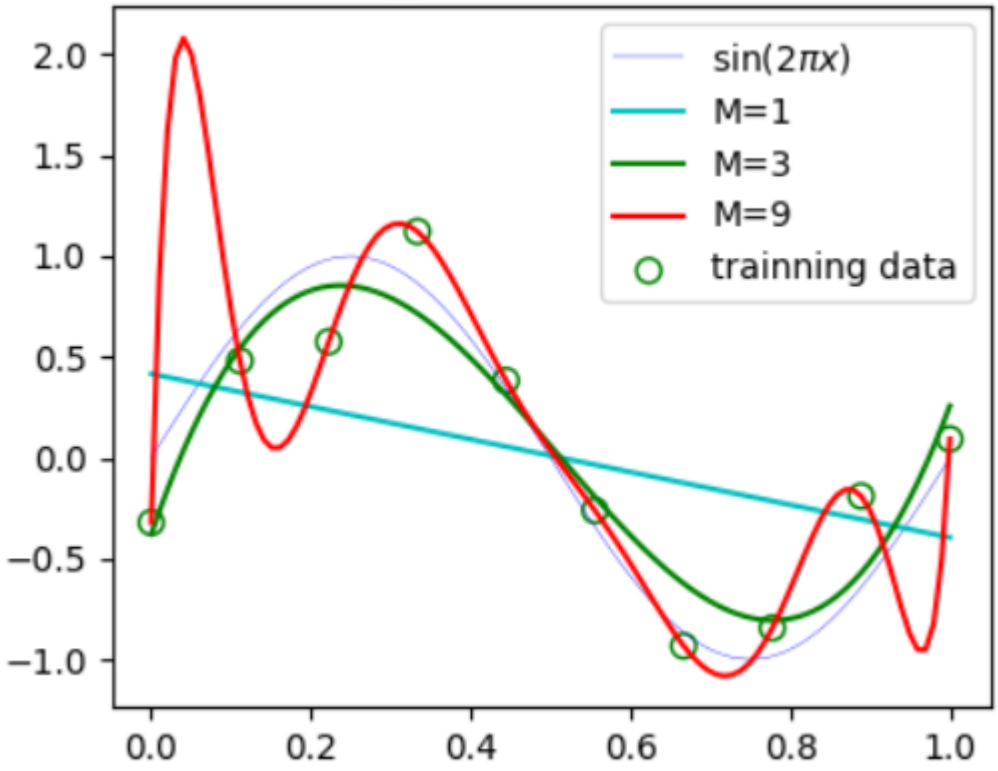
首先想到使代价方程取最小值的方法为求导数为 0，这种方法就是求解析解，另外，在计算方法这门课中，我们了解到了另外两种求函数最小值的方法：梯度下降法和共轭梯度法，这两种方法采用迭代的方式逐渐逼近函数的最小值。

4.无惩罚项的解析解

利用矩阵的迹、矩阵向量求导等知识，对最小二乘法的代价函数进行求导，令其为零，可以得到权重θ的解为：

$$\theta = (X^T X)^{-1} X^T y$$

按照 PRML 书上给出多项式次数$M = 1,3,9$时的函数图像如下：

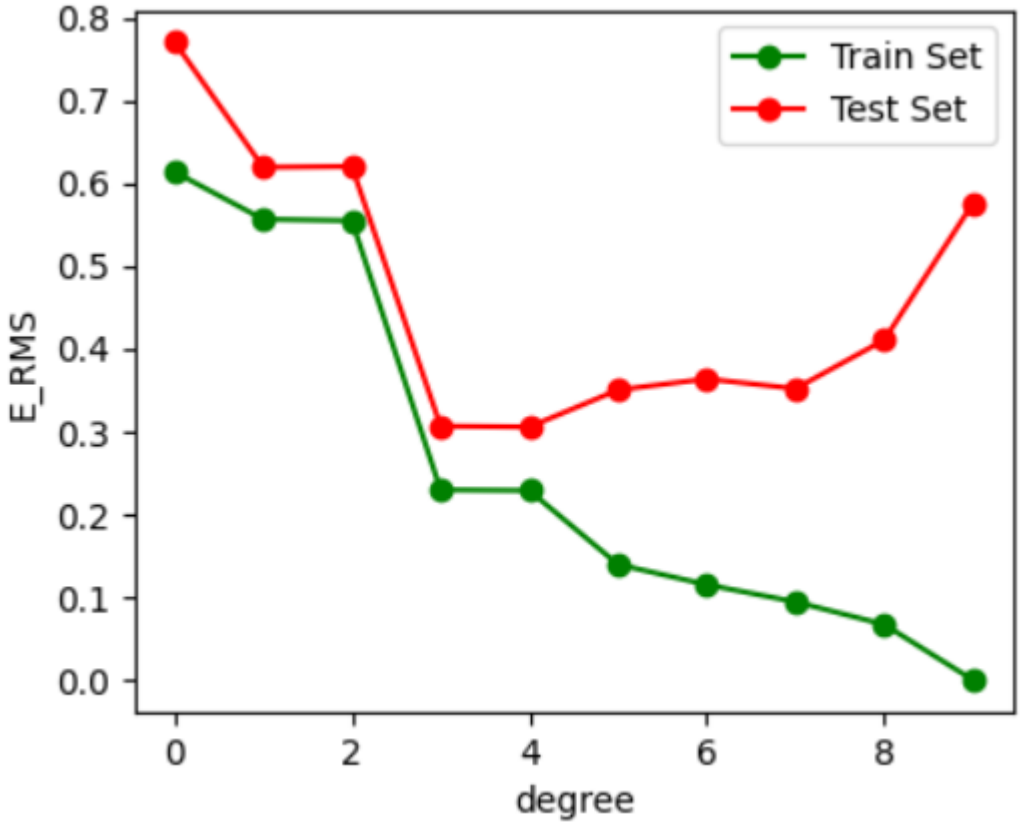


从上图中不难看出，对于无惩罚项的解析解，当多项式次数为 9 时完全经过了所有的样本点，但与原函数$y = sin(\pi x)$之间差距很大，这种情况就是过拟合；当多项式次数为 1 时，图像为一条直线，代价函数很大，同时拟合原曲线的效果也不好；当多项式次数为 3 时，代价函数较小，与原曲线的拟合效果也较好。

图中对于不同阶数的多项式之间的对比是直观和易于理解的，在 PRML 书中给出了另一种科学、定量的方式来描述不同情况下的根据代价函数去判断曲线拟合效果------根均方误差(RMS)。其公式如下：

$$E_{RMS} = \sqrt{2J(\theta)}$$

据此，对于从 0 到 9 阶数的多项式根据训练集$(m = 10)$和测试集$(m = 100)$进行比较，结果见下图：



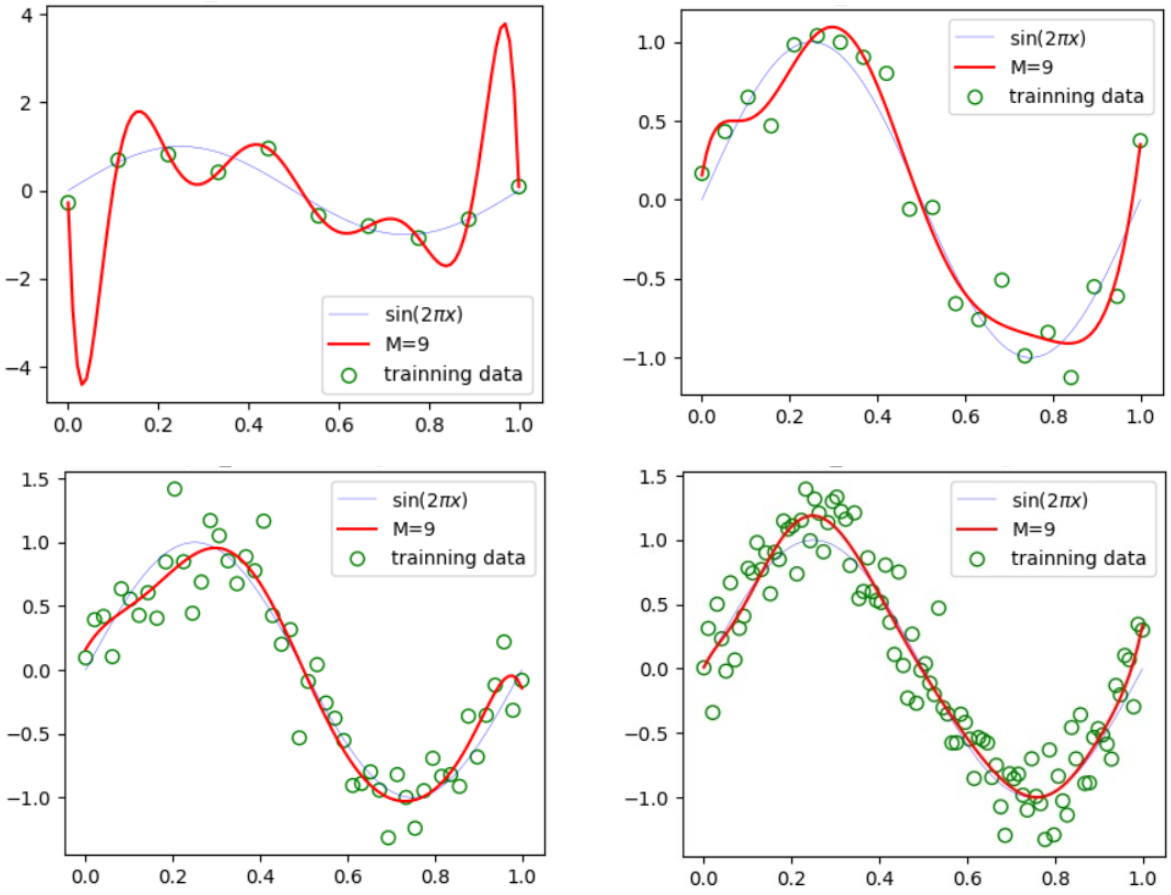
可以看出随着多项式的阶数增加，在训练集上的误差逐渐减小，对于测试集则呈现先降后增的过程，误差增大的过程实际上就是过拟合程度不断增加的过程，而误差在 3 阶或 4 阶处获得最小值，拟合效果较好。

5.克服过拟合的方式------增加样本

从上一节中我们看到了，对于解析解，当拟合的多项式阶数过高时会产生过拟合的现象，体现在图像上就是曲线出现剧烈的振荡，但按照常规的想法，高阶多项式完全可以获得至少与低阶多项式一样的效果（令高阶部分的系数为零），实际上去观察不同阶数的多项式的权重系数θ，我们发现，随着多项式阶数的增加，其系数的绝对值大小也在增加，可以理解为高阶多项式为了更好的与目标点的随机噪声相符进行了过分的调参。按照 PRML 的说法，当数据集的规模增加时，用来拟合数据的模型就越灵活，因此要克服过拟合，可以考虑增加训练集中样本数目，一般的，数据点的数量不应该小于模型的可调节参数的数量。

	M = 1	M = 3	M = 9
θ_0	0.544	-0.247	-0.107
θ_1	-1.157	9.516	-152.085
θ_2		-25.454	3349.517
θ_3		15.879	27807.669
θ_4			120259.954
θ_5			-302430.324
θ_6			458298.721
θ_7			-412337.980
θ_8			202814.080
θ_9			-41994.282

下图显示了当训练集样本数目分别为 10、20、50、100 时 9 阶多项式的拟合效果：



6.克服过拟合的方式------增加惩罚项

大部分情况下，训练集的数据不能随意地增加，而且根据训练集的样本数目去限制参数数目也是不够合理的，于是在 PRML 中引入了另一种克服过拟合的方式，为代价函数增加一个惩罚项：

$$J(\theta)=\frac{1}{2m}(X\theta-y)^T(X\theta-y)+\frac{\lambda}{2}||\theta||^2$$

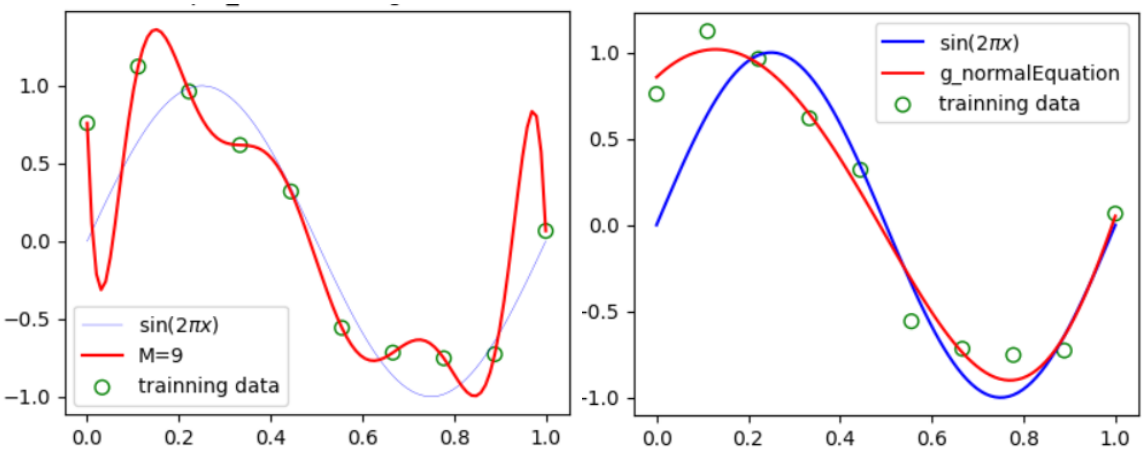
同样地，对其进行求导可以得到权重参数θ的解析解：

$$\theta = (X^T X + \lambda I)^{-1} X^T y$$

其中I为单位阵。不过按照吴恩达在$Coursera$上机器学习这门课中的说法，θ_0属于一个$bias$，对于所有的样本都有统一的影响，因此与过拟合无关，所以单位矩阵I的第一个元素应该设置为 0，即不对θ_0进行惩罚，我在代码中参照了这种写法。

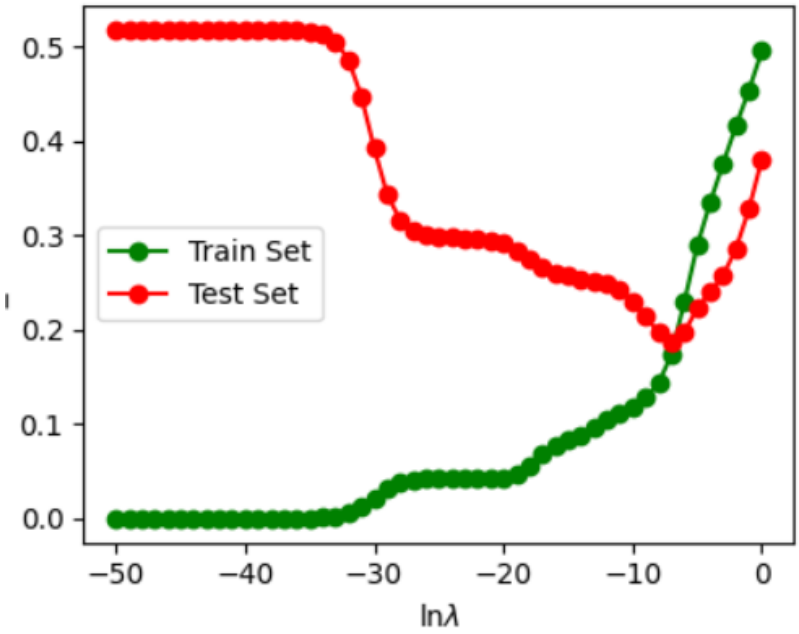
7.带有惩罚项的解析解

根据以上的分析，在惩罚项超参数λ取e^{-8}时，带有惩罚项的解析解拟合曲线图像与无惩罚项的对比如下：



8.超参数λ的选择

从 PRML 中得知，超参数的不同会影响去过拟合化的效果，简而言之，当超参数过大时，去过拟合化程度大，可能引起曲线欠拟合；而超参数过小时，去过拟合化程度不明显，为了选出一个相对较优的超参数，按照 PRML 的指导，对于$\lambda = e^{-50}, e^{-49}, \ldots, 1$的超参数进行比对，根据测试集上的均方误差选取最优，比对结果如下：



多次比对后，发现在$\lambda = e^{-8}$时，测试集上的均方误差较小。

9.梯度下降法

在计算方法中我们学到，对于一个函数求最小值可以在函数上取一个初值点，求该点的负梯度方向，沿着该梯度方向下降，下降步长被称为学习率，通过一定步骤的迭代就可以得到近似解。

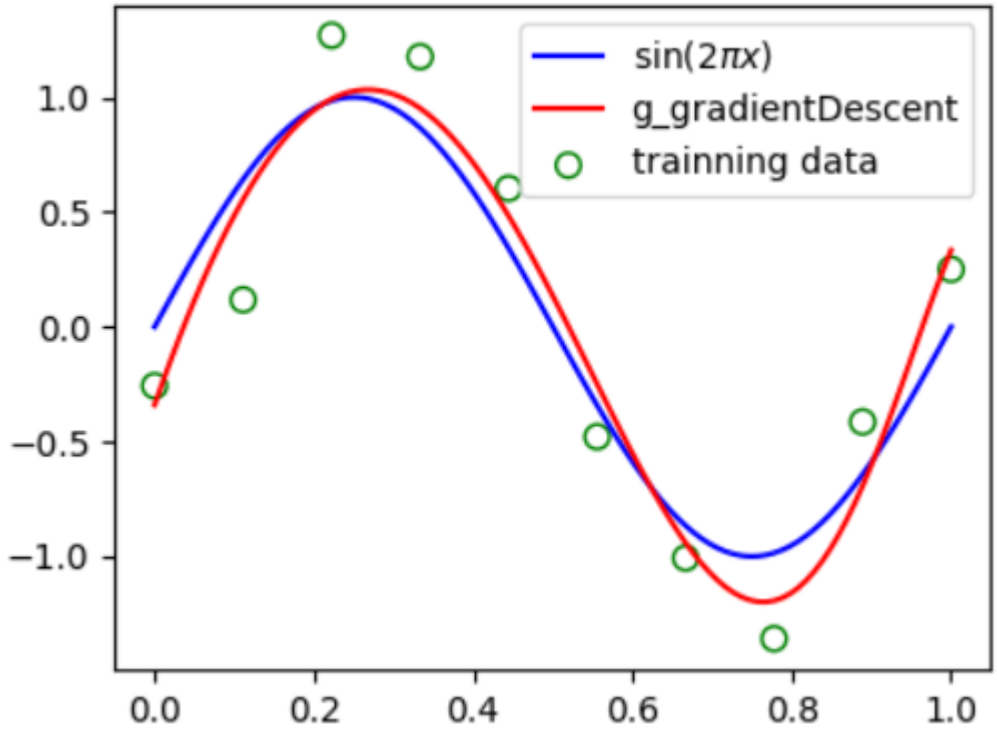
将代价函数对θ求导得到函数的梯度如下：

$$grad_0 = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x_0^{(i)}$$

$$grad_j = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x_j^{(i)} + \frac{\lambda}{m}\theta_j, j = 1,2,3,\dots,n$$

考虑学习率，当学习率过小时，梯度下降速度较慢，而学习率过大时可能造成在最小值附近振荡而不收敛，因此可以采用非定值的学习率进行梯度下降，即初始时定一个较大的学习率，当代价函数不再减小时将学习率减半，这样可以让函数既较快的下降，又不至于振荡，至于函数何时退出迭代，可以比对两次相邻的迭代的代价函数的差值，当其小于一定值时（例如 1e-7）就停止迭代，跳出循环。

超参数沿用上一小节数据即可，当$\lambda = e^{-8}$时，梯度下降法的图像如下：



在控制台输出其迭代次数发现其迭代了 39482 次，多次实验后可得梯度下降的次数基本维持在10^5量级。

10.共轭梯度法

梯度下降法求解时迭代次数很多，用优化的方法（动态减小学习率）也要迭代上万次才能得到近似解，这是由于对函数上某一点进行求导具有局部性质，在该点附近函数也许下降速度最快但总体来看，这个方向未必是函数下降最理想的方向，下面介绍一种可以在有限步内就能获得精确解的一种迭代方法------共轭梯度法。

下面引入共轭的概念，$A \in R^{n\times n}$为对称正定矩阵，$p,l \in R^n$，如果$(p,Al)=0$，则称p与l为$A-$正交或$A-$共轭。观察解析解：

$$\theta = (X^TX+\lambda I)^{-1}X^Ty$$

其可以被化为以下形式：

$$(X^TX+\lambda I)\theta = X^Ty$$

显然$(X^TX+\lambda I)$为对称正定矩阵，因此可以应用共轭梯度法。

对于方程$AX=b$，共轭梯度的算法步骤如下：

1. 任取初值$x^{(0)} \in R^n$；
2. $p_0 = r^{(0)} = b - Ax^{(0)}$
3. 对于$k = 0,1,2,\dots,N$

$$\alpha_k = \frac{(r^{(k)},p_k)}{(Ap_k,p_k)}$$

$$x^{(k+1)} = x^{(k)} + \alpha_k p_k$$

$$r^{(k+1)} = b - Ax^{(k+1)} = r^{(k)} - \alpha_k Ap_k$$

$$\beta_k = -\frac{(r^{(k+1)},Ap_k)}{(p_k,Ap_k)}$$

$$p_{k+1} = r^{(k+1)} + \beta_k p_k$$

超参数$\lambda = e^{-8}$时拟合曲线图像如下：