第十周上机课：回归问题中的正则化技术

正则化的问题背景

无论是学习函数还是解偏微分方程，只要涉及统计学习的领域，几乎总会遇到过拟合和欠拟合的问题。
过拟合问题是模型在训练数据上表现很好，但在新数据上表现很差的现象。

统计上会表现为：训练误差小、测试误差大。

模型复杂度可以量化的理解为模型参数的数目，例如：插值问题的多项式次数、神经网络的参数量等。

过拟合一般有三种原因：

1. 数据量太少
    - 图像识别的例子：使用CNN来识别猫狗的图像（分类问题）。训练集仅包含20张黑色猫和20张白色狗的图片。模型最终将学习到：（黑色像素较多的分类为猫、白色像素较多的分类为狗）。训练集误差为0，测试集效果很差。
    - 解决方案：收集更多数据
    - 注意：一般认为较少的数据将会欠拟合、但是当数据量不足与噪声水平抗衡时，模型将会更关注于训练集中的噪声，从而将训练集的噪声模式学习进来。这将会造成训练误差小、而测试误差大的过拟合现象。
2. 模型太复杂
    - 函数拟合例子：使用高次模型拟合低复杂度函数采样点。
    - 解决方案：选择合适复杂度的模型
    - `overfitting.m` 示例。
3. 噪声太多
    - 噪声信号过强，以至于无法识别正常数据，只能学习噪声。
    - 解决方案：清理数据或使用正则化

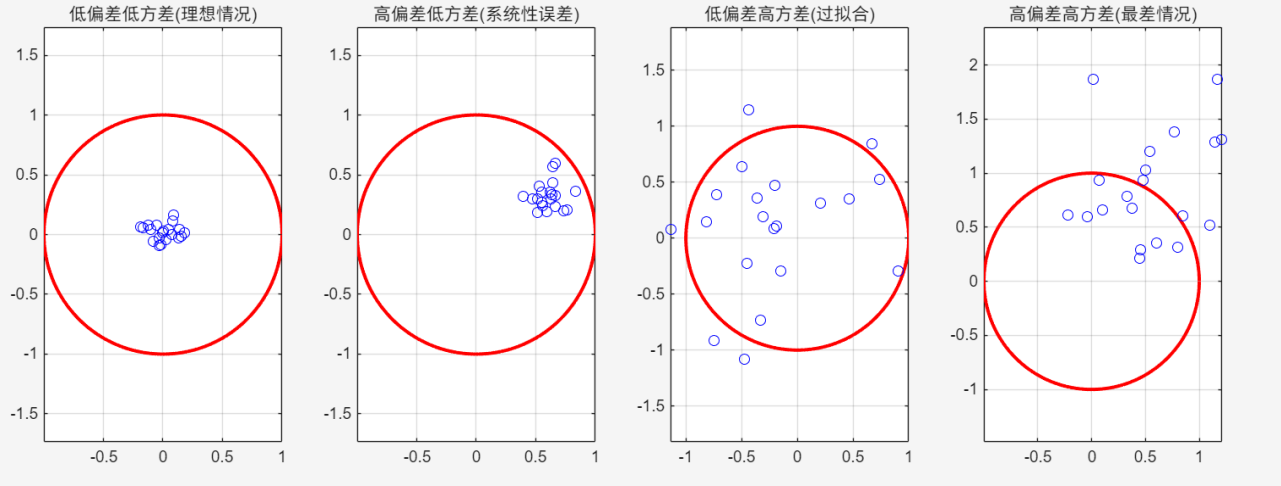
由于过拟合现象的广泛存在，正则化技术被开发出来并广泛用于统计学习领域。今天我们主要关注教材中的两种正则化算法：

- 岭回归：给模型系数加上约束，防止它们变得过大
- **Lasso**回归：不仅约束系数，还能自动选择重要特征

为什么要正则化？——偏差-方差权衡

什么是偏差和方差权衡？

使用 `biasAndVariance.m` 可以可视化偏差和方差：



数学上，泛化误差（类似于测试误差），可以进行偏差方差分解：

$$泛化误差 = 偏差^2 + 方差 + 噪声$$

推导：

设 $f(\mathbf{x};D)$ 表示用训练集 D 训练得到的模型 f 在样本 $\mathbf{x}\in\mathcal{X}$ 上的预测值，则 $f(\mathbf{x};D)$ 是一个随机变量，当训练集 D 固定时， f 在分布 $\mathcal{D}$ 上的泛化误差✦为

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\Big[\big(f(\mathbf{x};D)-y\big)^2\Big].$$

当考虑训练集 $D\sim\mathcal{D}^m$ 的随机性时，在期望意义下， f 的泛化误差为

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\ \mathbb{E}_{D\sim\mathcal{D}^m}\Big[\big(f(\mathbf{x};D)-y\big)^2\Big].$$

我们将对内层的 $\mathbb{E}_{D\sim\mathcal{D}^m}\Big[\big(f(\mathbf{x};D)-y\big)^2\Big]$ 作分解，它表示当 D 变化时的，在期望意义下， f 在任意给定的一个样例 $(\mathbf{x},y)\in\mathcal{X}\times\mathcal{Y}$ 上的泛化误差.

由假设知 $y=F(\mathbf{x})+\epsilon$ ，于是

$$\begin{aligned}\big(f(\mathbf{x};D)-y\big)^2&=\big(f(\mathbf{x};D)-F(\mathbf{x})-\epsilon\big)^2\\&=\big(f(\mathbf{x};D)-F(\mathbf{x})\big)^2+\epsilon^2-2\big(f(\mathbf{x};D)-F(\mathbf{x})\big)\epsilon.\end{aligned}$$

进而有

$$\begin{aligned}\mathbb{E}_{D\sim\mathcal{D}^m}\Big[\big(f(\mathbf{x};D)-y\big)^2\Big]=\mathbb{E}_{D\sim\mathcal{D}^m}\Big[\big(f(\mathbf{x};D)-F(\mathbf{x})\big)^2\Big]+\mathbb{E}_{\epsilon\sim\mathcal{N}(0,\sigma^2)}(\epsilon^2)\\-2\,\mathbb{E}_{D\sim\mathcal{D}^m,\ \epsilon\sim\mathcal{N}(0,\sigma^2)}\Big[\big(f(\mathbf{x};D)-F(\mathbf{x})\big)\epsilon\Big].\end{aligned}$$

对右侧第二项，直接有 $E_{\epsilon\sim\mathcal{N}(0,\sigma^2)}(\epsilon^2)=\sigma^2$. 对右侧第三项，由于噪声独立于一切其他随机变量，故该项为 $\mathbb{E}\Big[f(\mathbf{x};D)-F(\mathbf{x})\Big]\mathbb{E}(\epsilon)=0.$ 因此上式化简为

$$\mathbb{E}_{D\sim\mathcal{D}^m}\Big[\big(f(\mathbf{x};D)-y\big)^2\Big]=\mathbb{E}_{D\sim\mathcal{D}^m}\Big[\big(f(\mathbf{x};D)-F(\mathbf{x})\big)^2\Big]+\sigma^2\qquad(*).$$

又记 $\mu = \mathbb{E}_{D\sim\mathcal{D}^m}\Big[f(\mathbf{x},D)\Big]$，它是常数，进一步对(*)式右侧第一项作分解，同上理有

$$\begin{aligned}\big(f(\mathbf{x};D)-F(\mathbf{x})\big)^2 &= \big(f(\mathbf{x};D)-\mu+\mu-F(\mathbf{x})\big)^2\\ &= \big(f(\mathbf{x};D)-\mu\big)^2+\big(\mu-F(\mathbf{x})\big)^2+2\big(f(\mathbf{x};D)-\mu\big)\big(\mu-F(\mathbf{x})\big).\end{aligned}$$

进而有

$$\begin{aligned}\mathbb{E}_{D\sim\mathcal{D}^m}\Big[\big(f(\mathbf{x};D)-F(\mathbf{x})\big)^2\Big] &= \mathbb{E}_{D\sim\mathcal{D}^m}\Big[\big(f(\mathbf{x};D)-\mu\big)^2\Big]+\mathbb{E}_{D\sim\mathcal{D}^m}\Big[\big(\mu-F(\mathbf{x})\big)^2\Big]\\ &\quad+2\,\mathbb{E}_{D\sim\mathcal{D}^m}\Big[\big(f(\mathbf{x};D)-\mu\big)\big(\mu-F(\mathbf{x})\big)\Big].\end{aligned}$$

注意到 $\mu-F(\mathbf{x})$ 与随机变量 D 无关，故右侧第二项就是常数的期望，并且在第三项中，由 $\mathbb{E}\Big[f(\mathbf{x};D)-\mu\Big]=0$ 知该项为零. 因此上式化简为

$$\mathbb{E}_{D\sim\mathcal{D}^m}\Big[\big(f(\mathbf{x};D)-F(\mathbf{x})\big)^2\Big] = \mathbb{E}_{D\sim\mathcal{D}^m}\Big[\big(f(\mathbf{x};D)-\mu\big)^2\Big]+\big(\mu-F(\mathbf{x})\big)^2.$$

注意到右侧第一项就是随机变量 $f(\mathbf{x};D)$ 的方差，并将 μ 按定义回代到右侧第二项中，然后将上式回代(*)式，得

$$\begin{aligned}&\mathbb{E}_{D\sim\mathcal{D}^m}\Big[\big(f(\mathbf{x};D)-y\big)^2\Big] = \mathrm{Var}_{D\sim\mathcal{D}^m}\Big[f(\mathbf{x};D)\Big]\\ &+\Big(\mathbb{E}_{D\sim\mathcal{D}^m}\Big[f(\mathbf{x};D)\Big]-F(\mathbf{x})\Big)^2+\sigma^2.\end{aligned}$$

这个分解称为**偏差-方差（-噪声）分解**.

为了避免过拟合，需要降低泛化误差，而泛化误差又由偏差、方差和噪声三个量控制。

> 偏差：训练集变化时，学得模型的期望输出与真实标记间的差异；偏差越小，学习算法对训练集的拟合能力越强；
> 方差：训练集变化时，学得模型的变化程度；方差越大，训练集扰动对学得模型的影响越大；
> 噪声：采样所得样本标记与真实标记的差异；噪声越大，问题本身越难。

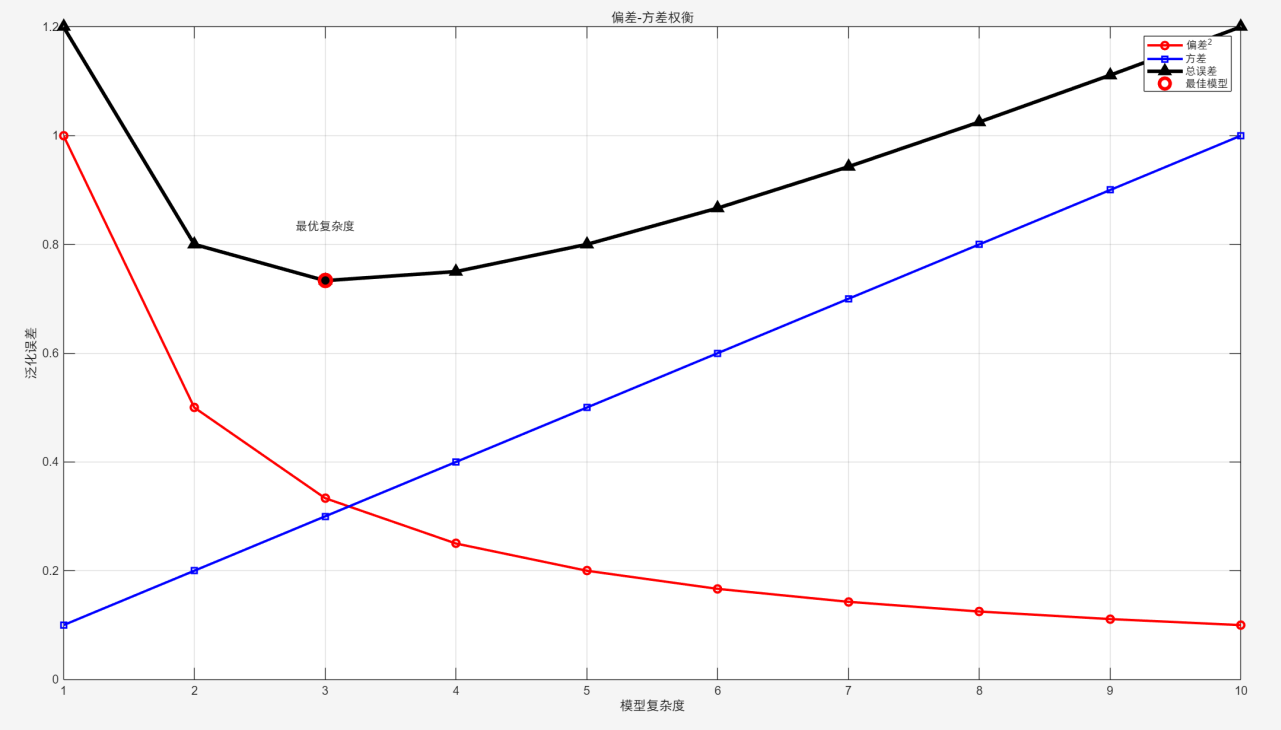
偏差与方差是能够通过控制训练拟合程度来控制的，而噪声则不能人为控制。

一般地，随训练程度增大，偏差减小，方差增大，泛化误差先减小后增大。使用`model_complexity.m`可以验证这一点。

训练不充分时，容易产生欠拟合；训练过度时，容易产生过拟合；

$$泛化误差 = 偏差^2 + 方差 + 噪声$$

使用 `complexity_plot.m` 可以可视化模型复杂程度同泛化误差的关系。



既然偏差和方差不可兼得，寻找最优复杂度就变得格外关键。正则化就是帮助我们找到这个最优点的办法！

岭回归 **Ridge Regression**

岭回归是普通线性回归的L_2正则版本。

原始线性回归的损失函数为：

$$Loss = \|y - X\beta\|_2^2$$
$$\min_{\beta} \|y - X\beta\|_2^2$$

其解析解为：

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

推导：

$$\begin{aligned}\|y-X\beta\|_2^2 &= (y^T-\beta^TX^T)(y-X\beta)\\ &= y^Ty-2y^TX\beta+\beta^TX^TX\beta \quad (\#)\end{aligned}$$

两边同时对β求偏导:

$$\frac{\partial Loss}{\partial \beta}=2X^TX\beta-2X^Ty=0$$

$$\Rightarrow\ \hat{\beta}=(X^TX)^{-1}X^Ty$$

注意到(#)式为β 的二次型，且 X^TX 正定，从而极值点 $\hat{\beta}=(X^TX)^{-1}X^Ty$ 即为最小值点。

岭回归在损失函数中增加了一个权重二范数损失项的平方 ($\lambda>0$)

$$Loss=\|y-X\beta\|_2^2+\lambda\|\beta\|_2^2$$

$$\min_{\beta}\ \|y-X\beta\|_2^2+\lambda\|\beta\|_2^2$$

其解析解为：

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

推导：

$$\begin{aligned}\|y-X\beta\|_2^2+\lambda\|\beta\|_2^2 &= (y^T-\beta^TX^T)(y-X\beta)+\lambda\beta^T\beta\\ &= y^Ty-2y^TX\beta+\beta^T(X^TX+\lambda I)\beta\end{aligned}$$

两边同时对β求偏导:

$$\frac{\partial Loss}{\partial \beta}=2(X^TX+\lambda I)\beta-2X^Ty=0$$

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

当数据特征 X 共线性性较为严重（列线性相关）时，X^TX 几乎不可逆，数值上表现为矩阵 $(X^TX)^{-1}$ 值非常大。

设矩阵 $X=U\Sigma V^*$ 为奇异值分解，其中 Σ 是 $m\times n$ 的半正定对角矩阵。当 X 共线性性较为严重时，Σ 中最小奇异值σ_n近似为0（秩的角度可以理解），从而 $X^TX=V\Sigma^*\Sigma V^*$。对角方阵 $\Sigma^*\Sigma$ 对角线上的最小值σ_n^2将近似0。
其逆：

$$(X^TX)^{-1}=V(\Sigma^*\Sigma)^{-1}V^*$$

将会出现 $(\Sigma^*\Sigma)^{-1}$，有 $1/\sigma_n^2\gg 1$. 这使得 $\hat{\beta}=(X^TX)^{-1}X^Ty$ 数值上非常大。

岭回归增加的 $\lambda>0$ 有限的改善了这一条件。使用 `ridge_reg.m` 可以对比二者的损失函数。

Lasso回归

岭回归的损失函数“告诉”模型：不要将特征的值学习的很大。但是特征虽小，仍不为0。对于噪声数据而言，岭回归几乎总会学习为全部特征都非零。然而真实模型却不是这样的。

为了达到稀疏回归的目的。自动选择重要特征、去掉不相关的特征，得到更简洁、可解释的模型。Lasso回归方法被提出。

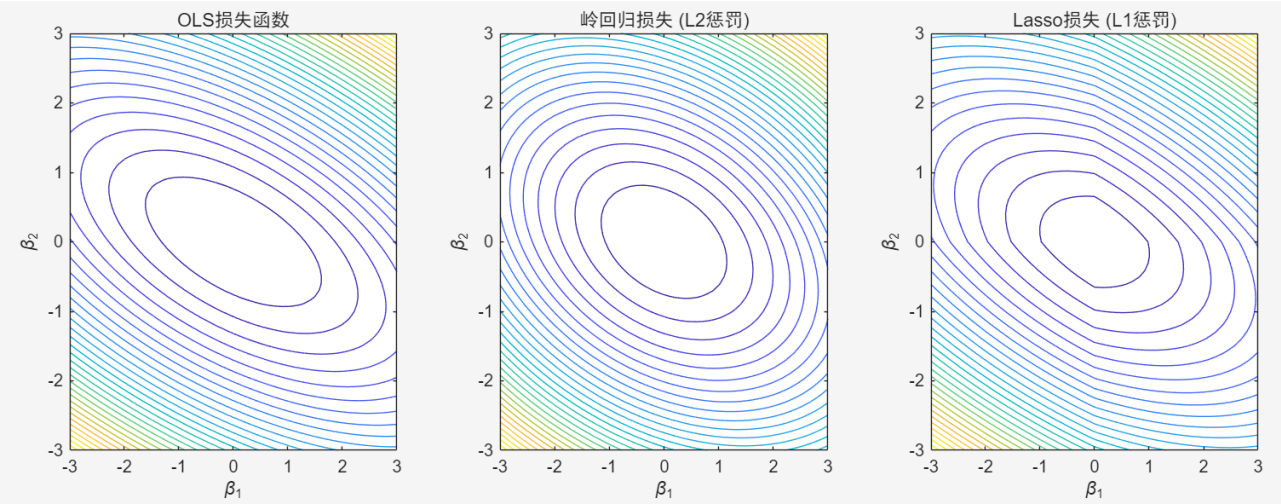
Lasso的核心思想：

告诉模型：可以拟合数据，但请尽量用少的特征 (不需要的特征就保持为0)。

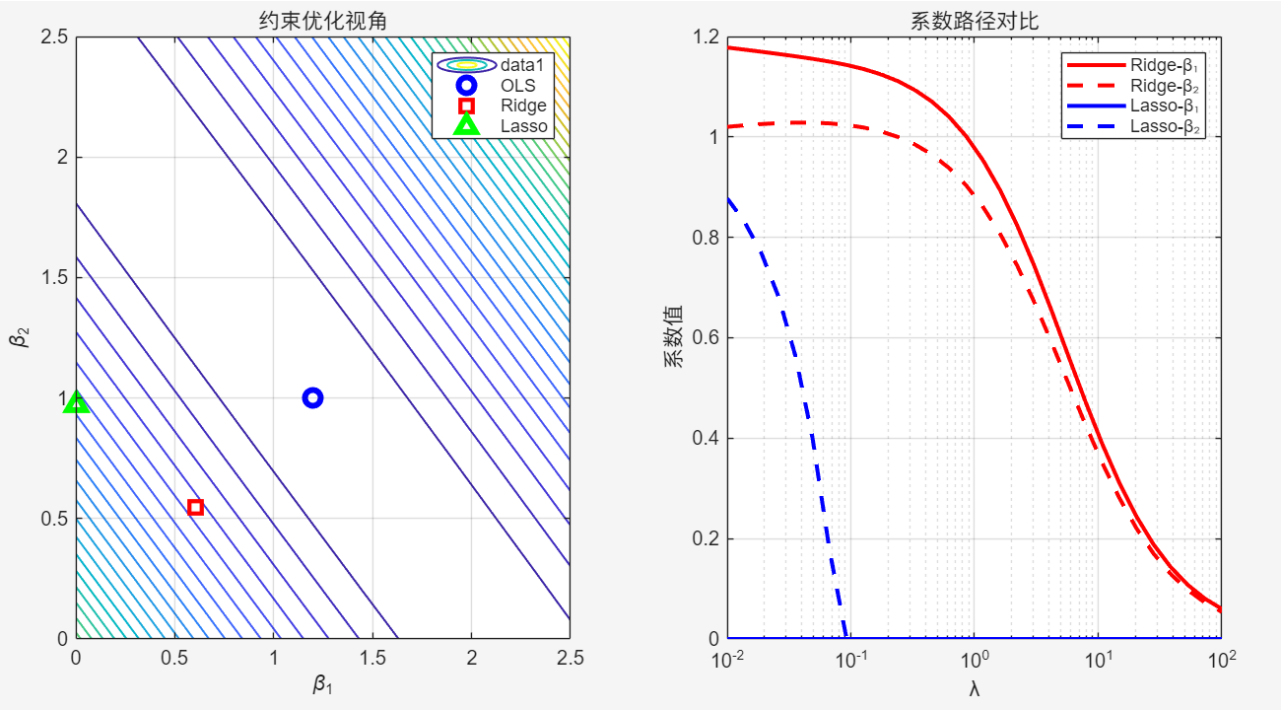
方法：在损失函数中加入对系数绝对值的惩罚。

$$Loss = \|y - X\beta\|_2^2 + \lambda\|\beta\|_1$$

使用 `lasso_reg.m` 对比Lasso和另外两种回归的区别。



Lasso拥有的L1惩罚的菱形约束区域有尖角，最优解更容易落在坐标轴上，使得某些系数正好为0！



图像表明Lasso可以不仅可以收敛到更优的损失函数，同时还让其中的无用特征系数为0。

由于Lasso回归引入的惩罚项是 L_1 的（不可微），因此无解析解。

大作业2

使用MATLAB完成以下任务：

1. 数据加载与预处理：
    - 使用附件中的`diabetes.csv`数据集
    - 将数据分为训练集（70%）和测试集（30%）
    - 对特征进行标准化处理
2. 模型训练与比较：
    - 实现普通线性回归
    - 实现岭回归，在λ=[0.001, 0.01, 0.1, 1, 10, 100]中寻找最优参数
    - 实现**Lasso**回归，在相同λ范围内寻找最优参数
3. 结果分析：
    - 比较三种方法（普通线性回归、岭回归、Lasso回归）在测试集上的MSE
    - 分析各方法选择的特征数量（稀疏性）
    - 绘制系数路径图
    - 讨论哪种方法最适合这个数据集并说明原因

提交：

- pdf/docx 报告，包含必要的可视化内容
- 必要的代码文件
- DDL：11月23日 23：59

数据说明：

AGE	SEX	BMI	BP	S1	S2	S3	S4	S5	S6	Y
59	2	32.1	101	157	93.2	38	4	4.8598	87	151
48	1	21.6	87	183	103.2	70	3	3.8918	69	75
72	2	30.5	93	156	93.6	41	4	4.6728	85	141
24	1	25.3	84	198	131.4	40	5	4.8903	89	206
50	1	23	101	192	125.4	52	4	4.2905	80	135

数据共有11列，最后一列为目标列，前10列为特征列。