《机器学习入门》2020_ 薛涛课件学习笔记及 Chatgpt 答疑内容汇总

YangZeping Chatgpt

2023-08-03

目录

1 引言

在本阶段任务中，需要学习 Elastic Net Regression(弹性网络回归)，这是机器学习的一种建模方法。忆得 2020 年曾上过薛涛老师开设的《机器学习》，

虽是入门课程，但也受益匪浅，故重温复习，此为笔记。此外今时 Chatgpt 已诞生并成为学习的好助手，故其答疑有益内容也汇总收录，作为笔记。

“如果我看得比别人更远些，那是因为我站在巨人的肩膀上。”勉之。

2 机器学习简史 A Brief History of Machine Learning From Mendel to AlphaGo

2.1 什么是模型

数据 Data：对现实/现象的观测 observations in real world
变量 Variable：对某一现象的、可重复的观测 repeatable observations for a specific aim
模型 Model：解释不同变量之间的联系 link different variables
模型的组成：当我们使用机器学习算法构建模型时，通常可以将模型分为两个主要部分：*对应法则 (functional form)* 和*参数 (parameters)*。
1. 对应法则 (functional form)：对应法则是模型的基本形式或结构，它定义了输入特征和目标变量之间的关系。对应法则可以是线性的、非线性的、概率模型等。例如，在线性回归模型中，对应法则是一个线性方程，表示目标变量与输入特征之间的线性关系。对应法则确定了模型的整体形式和基本假设。
2. 参数 (parameters)：参数是模型中的可调整变量，它们决定了对应法则中的具体数值。参数的值可以根据训练数据进行估计或优化，以使模型能够最好地拟合训练数据和泛化到新的未知数据。在线性回归模型中，参数即为每个输入特征对应的**权重值**，用于线性组合特征来预测目标变量。参数的值决定了模型的具体形态，不同的参数值可以导致不同的预测结果。
对应法则和参数之间的关系是：对应法则定义了模型的整体结构和基本形式，而参数则决定了对应法则中的具体数值。通过调整参数的值，我们可以改变对应法则中的权重和偏差，从而改变模型的预测能力和泛化能力。
在训练过程中，我们的目标是通过优化算法来找到最优的参数值，使得模型能够最好地拟合训练数据并具有良好的泛化能力。参数的选择对于模型的性能和准确性至关重要。
机器学习：通过数据学习规则本身

2.2 孟德尔的模型发现：潜变量

潜变量的概念：潜变量 (latent variable) 是指在观察数据中无法直接测量或观察到的变量。它是一种存在于数据背后的概念或构建，无法直接观察到或测量，但对于解释数据关系和模型建立非常重要。

潜变量通常无法被直接观察到，因为它们是抽象的、内在的概念，无法直接量化或测量。相反，我们可以通过观察其相关的指标、表现或观测变量来推断或间接地推测潜变量的存在或价值。

潜变量的例子包括心理学中的人格特质、情绪状态，经济学中的消费者偏好、市场需求，教育研究中的学习动机等。这些潜变量对于理解和解释数据的关系非常重要，因为它们可以帮助我们揭示背后的隐藏模式、结构和机制。

在统计建模和机器学习中，潜变量模型被广泛用于处理潜变量的问题。这些模型通过观测变量之间的关系来推断潜变量，并通过参数估计或优化方法来解释数据的变异性和关联性。常见的潜变量模型包括主成分分析 (PCA)、因子分析 (Factor Analysis)、结构方程模型 (Structural Equation Modeling) 等。

通过使用潜变量模型，我们可以更好地理解数据背后的潜在结构和规律，从而更准确地进行数据分析、预测和决策。

复杂问题，如果引入更多潜变量层，进一步增加参数占有的权重，从而摆脱经验对模型的影响，进而实现由数据直接到规律。

孟德尔的成功：测交实验 (Test Cross) 的成功，说明：预测是验证模型的“金标准”，Predition is the golden standard to judge a model.

A summary of Mendel's Discovery:

1. 应用了模型的思想，统一解释 F0→F1 和 F1→F2 问题。

2. 引入了 Hidden layer(潜变量) 的思想：增加参数空间，捕捉非线性效应 → 深度学习的根本思想之一：摆脱“对应关系”的预设，利用参数模拟任意函数关系。

依赖预测，评估模型好坏‘

2.3 机器学习简史

机器学习算法的分类：机器学习算法可以根据其学习方式和任务类型进行分类。常见的分类包括有监督学习 (supervised learning)、无监督学习 (unsupervised learning) 和强化学习 (reinforcement learning)。

有监督学习 (Supervised Learning)：有监督学习是指在训练过程中，算法接收到带有已知标签或类别的训练样本作为输入，通过学习样本的特征和对应的标签之间的关系，来建立一个预测模型。该模型可以用于预测新的未标记样本的标签或类别。有监督学习的任务包括分类和回归。常见的有监督学习算法有决策树、支持向量机、逻辑回归、神经网络等。

无监督学习 (Unsupervised Learning)：无监督学习是指在训练过程中，算法接收到没有标签或类别信息的训练样本作为输入，通过学习样本之间的关系和结构，来发现数据中的模式、群组或其他隐藏的结构。无监督学习的任务包括聚类、降维和关联规则挖掘等。常见的无监督学习算法有 K 均值聚类、主成分分析 (PCA)、关联规则挖掘等。

强化学习 (Reinforcement Learning)：强化学习是一种通过与环境的交互学习来做出决策的机器学习算法。在强化学习中，智能体通过采取不同的行动来与环境进行互动，并根据环境的反馈 (奖励或惩罚) 来调整其行为策略，以最大化长期累积的奖励。强化学习适用于需要进行决策的问题，如游戏、机器人控制、自动驾驶等。常见的强化学习算法有 Q-learning、深度强化学习等。

机器学习算法的历史：机器学习算法的历史可以追溯到上世纪 50 年代和 60 年代。以下是对机器学习算法历史的简要介绍：

1950 年代-1960 年代：机器学习的起源可以追溯到人工智能 (AI) 的早期阶段。在这个时期，研究人员开始探索如何让机器通过学习和推理来模仿人类的智能。早期的机器学习算法主要基于符号主义方法，如逻辑推理和专家系统。

1960 年代-1970 年代：在这个时期，统计学习理论和概率图模型开始受到关注。研究人员开始使用统计方法来建立模型，通过对数据进行概率建模和参数估计来进行预测和推断。这个时期出现了一些经典的统计学习算法，如线性回归、朴素贝叶斯分类器等。

1980 年代-1990 年代：在这个时期，机器学习算法开始接受更多的关注和应用。支持向量机 (SVM) 等算法在这个时期提出并广泛应用于模式识别和分类问题。此外，决策树、神经网络等算法也在这个时期得到了发展和应用。

2000 年代至今：随着互联网和大数据时代的到来，机器学习进入了一个爆发增长的阶段。在这个时期，由于计算能力的提高和数据的丰富，深度学习算法开始崭露头角。深度学习算法利用多层神经网络模拟人脑的神经元结构，通过大规模数据训练来进行特征提取和模式识别。深度学习在图像识别、自然语言处理等领域取得了重大突破。

此外，近年来，强化学习和迁移学习等新兴的机器学习方法也开始受到广泛关注和研究。

机器学习的发展同时依赖：模型进展，算力增长

机器学习算法的流派 (主义)：在机器学习算法的历史中，出现了不同的主义方法，这些方法在不同的时期受到关注和应用。以下是对机器学习算法历史上的各种主义的简要介绍：

符号主义 (Symbolism)：符号主义是机器学习算法早期的主流方法。符号主义强调通过推理和逻辑推断来模拟人类的智能。在符号主义方法中，使用符号表示和处理知识，并使用规则进行推理和决策。早期的专家系统就是基于符号主义方法开发的。然而，符号主义方法在处理大规模的数据和复杂的问题时遇到了困难，逐渐被其他方法所取代。

统计主义 (Statisticalism)：统计主义方法强调通过统计建模和概率推断来进行机器学习。这种方法利用数据的统计特性来建立模型，并使用统计推断来进行预测和推理。统计主义方法在 20 世纪 60 年代和 70 年代开始受到关注，并成为机器学习算法的重要分支。常见的统计主义算法包括线性回归、朴素贝叶斯、隐马尔可夫模型等。

连接主义 (Connectionism)：连接主义是指通过神经网络模型来进行机器学习的方法。连接主义方法模拟了大脑神经元之间的连接和信息传递过程。神经网络模型由多个节点 (神经元) 组成，节点之间通过连接传递信息和进行计算。连接主义方法在 20 世纪 80 年代和 90 年代开始受到关注，并在图像识别、语音识别等领域取得了重要的突破。常见的连接主义算法包括感知机、多层感知机、卷积神经网络等。

深度主义 (Deep Learning)：深度主义是近年来兴起的一种机器学习方法，也是连接主义的一种延伸。深度主义通过多层神经网络模型 (深度神经网络) 来进行特征提取和模式识别。深度主义利用大规模数据进行训练，通过多层网络的层次化特征提取能力来提高模型的性能。深度主义在图像识别、自然语言处理等领域取得了许多突破，成为现代机器学习的重要方法。

这些主义方法在机器学习算法的发展中各有特点和应用领域。随着技术的

进步和问题的复杂性，不同的主义方法相互融合和发展，为解决现实世界的复杂问题提供了更多的选择。

2.4 机器学习应用案例：Alpha Go

让机器学棋：有监督学习，穷举法列出所有可能及其胜负，对算力挑战机大使用蒙特卡洛搜索树 (Monte Carlo tree search) 方法，划分优势步和劣势步，舍弃后者 (修剪枝叶)，使问题进入计算机算力范围。
蒙特卡洛搜索树 (Monte Carlo tree search，简称 MCTS) 是一种用于决策制定的算法，特别适用于没有完全信息的博弈或规划问题。它是通过模拟随机事件来评估可能的行动，并构建一棵搜索树来指导决策过程。 MCTS 算法包括以下几个关键步骤： 1. 扩展 (Expansion)：从当前状态开始，通过选择未探索的行动来扩展搜索树。扩展的目的是增加树的分支，以便更好地探索可能的行动。 2. 选择 (Selection)：在搜索树中，根据一定的策略选择一个行动。常用的策略是使用上界置信区间 (Upper Confidence Bounds, UCB) 来权衡探索和利用。 3. 模拟 (Simulation)：从选中的行动开始，通过模拟游戏的随机走子来评估该行动的效果。模拟通常会一直进行到游戏结束，或者达到预先设定的模拟次数。 4. 反向传播 (Backpropagation)：将模拟的结果反向传播回搜索树，更新每个节点的统计信息，例如行动的胜利次数和总模拟次数。通过不断地进行扩展、选择、模拟和反向传播，MCTS 算法逐渐优化搜索树，提供更准确的行动评估。最终，根据搜索树中的统计信息，可以选择具有最高胜率或最高期望收益的行动作为决策。蒙特卡洛搜索树广泛应用于各种博弈游戏，如围棋、象棋和扑克等。它的优势在于可以处理大规模的状态空间和隐含信息，同时具有较好的效率和可扩展性。

3 回归模型 Regression Model

3.1 数据视角下的“解方程”

体悟：模型 (变量、对应关系、参数) 在代入一组组实际数据后，就转化为了方程。解方程本质即是求模型参数的过程。

3.2 方程组的矩阵表示

体悟：向量的意思就是，你这个一个位置，可以有很多观测，想象成数据框的同一列有很多行就好。

3.3 数据中的不确定性

现实数据具有不确定性：数据和规律并非完美“拟合”：

模型对规律的近似：适当对模型简化，剥啄主要规律，能够提供计算效率。

数据统计困难：现实中完美变量很难获得，例如每个变量都有对其标准的定义。

测量误差：现实中对变量的观测也并非完美。例如，对时间的测量受限于工具的精度，秒表精度就有限。

在不确定性下，哪个选择最优？

3.4 从解方程到回归模型

现实数据下的解方程：

$$y_{n\times 1} = X_{n\times p}\beta_{p\times 1}$$

一般情况 n»p

困境：一般情况下，一组参数 $\beta_{p\times 1}$ 仅能使得 n 个方程中的 p 个成立，因此方程没有完美的解。

模型的复杂度：参数越多，模型拟合越好

- n 组方程最多可求解 n 个参数
- 一般来说，模型参数个数 P«n
- (N-P) 被称为模型“自由度”
- 当 P=n 时，完美拟合。

解决方法：猜方程的解 → 更优的猜测 → 比较猜测 ($\hat{y}$) 和数据 (y) 的差别

损失函数：平方和损失函数

$$L(\hat{y}, y) = \sum_i (y-\hat{y})_i^2 = (y-\hat{y})'(y-\hat{y}) = ||y-\hat{y}|| = \sum_i (y_i - x_i\hat{\beta})^2$$

寻找最优模型 → 寻找 $\hat{\beta}$ 使得损失函数最小
$argmin_{\beta}||y_i - x_i\hat{\beta}||$

4 最优化理论初步 Introduce optimization

4.1 目标函数的概念

目标函数是指在优化问题中用来衡量目标的函数。在数学建模和优化领域，我们通常希望找到一个最优的解来最大化或最小化目标函数的值。

目标函数通常包含一个或多个变量，我们的目标是通过调整这些变量的值来使目标函数达到最大值或最小值。优化问题的目标可以是各种形式，如最大化利润、最小化成本、最大化效用、最小化距离等，具体取决于问题的性质和目标的定义。

例如，假设我们要最小化一个简单的二次函数目标函数：

$$f(x) = ax^2 + bx + c$$

其中，a、b 和 c 是常数，x 是变量。我们的目标是找到使函数 $f(x)$ 达到最小值的 x 的值。

在优化问题中，目标函数往往与一些约束条件一起使用，以定义一个完整的优化问题。这些约束条件可以是等式约束或不等式约束，用来限制变量的取值范围。

此外常见的目标函数包括：损失函数 (Loss Function), 最大似然函数 (Maximum Likelihood), 最大熵函数 (Maximum Entropy)。寻找 β 值，使得 L 最小：

$$L(\hat{y}, y) = \sum_i (y - \hat{y})_i^2 = (y - \hat{y})'(y - \hat{y}) = ||y - \hat{y}|| = \sum_i (y_i - x_i\hat{\beta})^2$$

优化问题的目标函数设计和求解是优化算法和方法的核心，常见的优化方法包括最优化理论、梯度下降、线性规划、非线性规划等。

4.2 目标函数举例

三种类型的目标函数:

1. 合理的损失函数是指能够准确反映模型预测与真实值之间差异的函数。合理的损失函数通常具有以下特点:
连续性:合理的损失函数在整个定义域上都是连续的,这样可以使优化算法能够顺利进行。
可导性:合理的损失函数在大部分或全部定义域上都是可导的,这样可以使用梯度下降等基于梯度的优化算法进行模型参数的更新。
非负性:合理的损失函数对于预测误差始终是非负的,表示误差的度量是正向的。
一些常见的合理的损失函数包括均方误差 (Mean Squared Error, MSE)、平均绝对误差 (Mean Absolute Error, MAE)、对数损失 (Logarithmic Loss) 等。

2. 无理的损失函数是指在某些情况下不合理或不适用的损失函数。这些损失函数可能具有以下问题:
不连续性:无理的损失函数可能在某些点上不连续,导致优化算法无法进行或不稳定。
不可导性:无理的损失函数可能在某些点上不可导,使得基于梯度的优化算法无法使用。
非实用性:无理的损失函数可能不符合具体问题的特点,无法准确地衡量模型的性能。

3. 低效的损失函数是指计算复杂度较高或优化困难的损失函数。这些损失函数可能需要大量的计算资源或时间来计算,或者需要使用复杂的优化算法才能达到最优解。

最优化的研究内容:
- 设计或应用合理的目标函数 Lasso 回归
岭回归
- 针对特定类型的且标函数,设计更高效的最优化算法
梯度下降 (Gradient Descent)
牛顿法 (Newton's Method)

4.3 回归模型的最优化求解

单参数的平方和损失函数最优化：拆开，求导，令等于零，得到极值点。

多参数的平方和损失函数最优化：同理，只是找到的是空间上的极值点 (就像山岭，故名岭回归)。

梯度下降算法:

固定 β_1，更新定 β_2，求梯度，沿斜率更新 $\beta 2$，最终得到 β_2 的值；
固定 β_2，更新 β_1，同理得到 β_2 的值。

5 模型拟合优度 Model fitness

5.1 拟合优度的概念

拟合优度是用于衡量回归模型对观测数据的拟合程度的指标。它反映了**模型预测值与实际观测值之间的接近程度** (相比于目标函数而言，相当于事后评价 ↓)。

区别：模型拟合优度与损失函数数学形式上是一致的 (均用 MSE 来表示，只是解释不同)；区别在于拟合优度是确定模型后用于评估，损失函数是确定模型前用于参考。

拟合优度的评价方法:

当用均方根误差衡量拟合优度时，会发现难以两全，这两个值也分别代表着准确度和精密度。

模型拟合优度和损失函数的关系：两者本质上都是对模型表现的评估；

损失函数的是为了参数估算：通常根据某一个标准找到表现最好的模型 (一道菜怎样做才最好吃)；
拟合优度评价是为了评价模型表现：存在多个维度，同一个损失函数并不一定使得所有维度都得到最优化 (恰如准确度和精密度难以两全)。

5.2 准确度和精密度

衔接上述拟合优度评估方法，准确度和精密度分别对应拟合偏差和拟合方差

准确度：拟合偏差
在平均意义上，拟合值对观测值的预测较好
$\hat{y} - \bar{\hat{y}} = mean(y - \hat{y}) = mean(e)$

精密度：拟合方差
模型表现的稳定性，即误差的变异程度

最优估计值是精确度和精密度的权衡。

$$MSE = Var + Bias^2$$

准确度和精密度对模型表现的影响:
实际模型的表现介于两者之间
Var 主导该模型

准确度和精密度的权衡 (影响因素)： 数据
模型设计
估算方法 (损失函数设计)

5.3 影响拟合优度的因素——模型 “复杂度”

一言以蔽之，参数越多，模型越复杂，拟合越好（但是会出现过拟合问题，总在寻找平衡，就像人生)。

影响模型拟合优度的因素： 数据质量 (测量误差)
先验假设 (对应关系假设)
模型复杂度 (Complexity)：参数个数，又称模型容量 (Capacity)
- n 组方程最多可求解 n 个参数
- 一般来说，模型参数个数 P«n
- (N-P) 被称为模型 “自由度”
- 当 P=n 时，完美拟合。

6 模型的交叉验证 Cross Validation

6.1 模型的泛化

建模的目的：**拟合不是建模的目的**
模拟规律需要“迁移”
模型要用于预测

模型的泛化 (Generalization)：机器学习的主要挑战是我们的算法必须能再先前未观测到的数据点上表现良好，而不是只在训练数据集上表现良好。我们把模型在先前未观测的数据点上表现良好的能力成为泛化。
泛化体现了模型模拟的规律需要具有一定适用范围。
预测/追算/插值/估计。

泛化误差的评价方法：
模型预测的表现
具体方法与拟合优度的评价方法类似：Bias；Variance；RMSE；Correlation

泛化误差和拟合误差的区别：拟合误差： 观测数据内误差。
使用最优化方法求得最小拟合误差 (受控制)。

泛化误差： 尽利用观测数据无法准确求解。
即便是引入外部数据，也无法准确求解 (因为外部数据也仅仅是总体的一个样本，不能代表所有的总体)。
机器学习和传统统计的区别是关注泛化误差

体会：目标函数，拟合优度，泛化误差这三者被同一部分内容一以贯之：`Bias; Variance; RMSE; Correlation`

泛化误差的估计方法
1. 引入外部数据：
机器学习的通用任务具有标准测试数据集。
例如，图像识别 (图像识别测试数据)
2. 不引入外部数据：
交叉验证
Bootstrap(自助法)

6.2 交叉验证 (Cross-validation)

交叉验证（Cross-validation）是一种常用的模型评估技术，用于估计模型在未见过的数据上的性能表现。它的基本思想是将已有的数据集分成若干个子集，其中一个子集被保留作为测试集，其余子集用于训练模型。然后，计算模型在测试集上的性能指标，如准确率、均方误差等。

利用数据构造泛化场景：
- 训练数据 (Training data)-参与模型拟合
- 测试数据 (Test data)-不参与模型拟合

对于大数据/复杂模型的交叉验证：K-折交叉验证 (K-fold cross-validation)：取子集作为测试数据 - 随机把数据分为 K 折 - 取其一折作为测试数据 - 其余作为训练数据 - 重复上述过程 K 次，使得所有数据点都扮演过测试的角色

交叉验证结果的评估：
- 具体方法与拟合优度的评价方法类似
- 与泛化误差的关系：交叉验证结果 RMSE 可以作为一种估计模型的泛化误差的方法，本质上是对模型泛化误差的估计，两者是一小一大的包含关系。

6.3 过拟合 (overfit) 和欠拟合 (underfit)

拟合误差和泛化误差的比较：
对于同一个模型：
- 交叉验证结果可以和拟合结果类比
- 一般情况下拟合误差的各项指标更优

对于不同模型：
- 拟合误差随模型复杂度而减少
- 泛化误差随模型复杂度的变化并非如此

随模型复杂度的变化的变化规律

- 拟合误差随模型复杂度的增加而降低
- 泛化误差随模型复杂度的增加先降低再升高。
在极值点左侧，模型欠拟合
在极值点右侧，模型过拟合。
(在简单情况下，可以用拟合误差近似替代泛化误差。)

7 维度灾难与模型正则化 Curse of dimensionality and Model regularization

7.1 维度灾难

模型:

$$y \approx \hat{y} \Leftrightarrow y = \hat{y} + e$$

- “ ” 源于不确定性 (e)
- 规律 (或参数) 决定 $\hat{y}$
- 参数不应该参与 e 的模拟
- 复杂模型，信息冗余，参数错误模拟了 e
- 高维度数据往往造成过拟合

7.2 模型正则化 Model regularization

7.2.1 逐步回归 Stepwise regression

逐步回归（Stepwise Regression）是一种用于构建回归模型的变量选择方法。它通过逐步加入或剔除预测变量，来选择对目标变量具有显著影响的变量，从而建立一个较为简洁而有效的回归模型。

逐步回归可以分为前向选择（Forward Selection）、后向剔除（Backward Elimination）和逐步回归（Stepwise Regression）这三种常见的方法。以下是它们的基本概念:
前向选择（Forward Selection）: 从一个空模型开始，逐步添加预测变量，每次选择对目标变量影响最大的一个变量，直到满足一定的停止准则。

后向剔除（Backward Elimination）：从包含所有预测变量的完整模型开始，逐步剔除对目标变量影响最小的一个变量，直到满足一定的停止准则。
逐步回归（Stepwise Regression）：结合了前向选择和后向剔除的方法，可以在每一步中既添加变量，又删除变量，直到满足一定的停止准则。

逐步回归的停止准则可以根据问题的具体情况选择，常见的准则包括：
AIC（赤池信息准则）：通过最小化 AIC 的值来选择合适的模型。
BIC（贝叶斯信息准则）：通过最小化 BIC 的值来选择合适的模型。
p-value（显著性水平）：根据预测变量的 p-value 来选择具有统计显著性的变量。

逐步回归方法可以帮助剔除不相关或冗余的变量，从而简化模型，提高模型的解释能力和泛化能力。然而，需要注意的是，逐步回归是一种启发式方法，其结果可能受到数据集选择和建模过程中的随机性的影响，因此在使用逐步回归方法时，需要谨慎解释和判断结果。

7.2.2 岭回归 Ridge regression

岭回归简介：
- 起眼较早
- 并非严格针对维度灾难
- 针对共线性问题

$$y \sim x_1\beta_1 + x_2\beta_2$$

$$cor(x_1, x_2) \sim -1 or 1$$

- 低维度数据的信息冗余问题

信息冗余 (多重共线性) 造成的问题极端案例：- $y \sim x_1\beta_1 + x_2\beta_2$
- $cor(x_1, x_2)^2 = 1 \Rightarrow x_2 = \gamma x_1$
- 建模 $y \sim x_1\hat{\beta}$

- 对于任意组合 (β_1, β_2) 满足：$\beta_1 + \gamma\beta_2 = \hat{\beta}$ 都是有效的参数求解

→ 最终导致的情况是：由高效的线性损失函数转为低效的环形损失函数，岭的最低处范围变大，不利于找到极值点。
→ 参数多 → 共线性的可能性大 → 损失函数效度变低

解决方案:
- 对于变量中的重复信息，“平均采纳，不偏不倚”
- 施加约束，降低参数自由度 (shrinkage estimation)
- $\beta_1^2 + \beta_2^2 \leq s(\lambda)$

岭回归：岭回归是一种用于解决多重共线性问题的线性回归方法。它通过引入 L2 正则化项来限制回归系数的大小，从而减小共线性对回归结果的影响。
岭回归的目标是最小化以下形式的目标函数：

$$argmin\beta[(Y - X\beta)^T(Y - X\beta) + \lambda\beta^T\beta]$$

$argmin [(Y - X)(Y - X) +]$ 被称作拉格朗日算子；
s 或 等价，被称为超参数或调整参数 (tuning parameter) 其中，Y 是因变量的观测值，X 是自变量的观测值组成的矩阵， 是回归系数向量， 是正则化参数（也称为岭参数或惩罚参数）。
在目标函数中，第一项表示残差平方和，衡量了模型对观测值的拟合程度；第二项是 L2 正则化项，衡量了回归系数的大小。通过调节正则化参数 的值，可以控制正则化项对回归系数的惩罚程度。
岭回归的求解可以使用最小二乘法或其他优化算法，通过最小化目标函数，得到最优的回归系数估计。
值得注意的是，岭回归在引入正则化项的同时，可能会导致估计的回归系数偏向于缩小，即缩小到离零更近的值。这可以降低共线性的影响，但也可能损失一些预测能力。因此，在选择正则化参数 时，需要进行适当的调参和模型评估，以平衡拟合度和泛化能力。

为什么引入惩罚项 $\lambda\beta^T\beta$ 会使得系数 β 值减小：
目标函数中的正则化项 是岭回归中引入的 L2 正则化项，它的存在会导致回归系数 的缩小。

首先，让我们来理解正则化项的作用。正则化项的目的是通过对回归系数施加惩罚，限制其大小，从而降低模型的复杂度。在岭回归中，使用 L2 正则化项，它的形式是 ，即回归系数的平方和。这意味着正则化项会使回归系数的平方和最小化。
当目标函数最小化时，正则化项会对回归系数产生影响，使得回归系数尽量小。这是因为在目标函数中，回归系数 出现了两次，而且是以平方的形式。为了最小化目标函数，最优的 值会尽量使得 的平方和减小。
因此，当 的值较大时，正则化项的影响会更加明显，回归系数 的估计值会更加偏向于缩小，尤其是对那些对目标变量的预测影响较小的自变量。这可以降低共线性的影响，提高模型的稳定性和泛化能力。
需要注意的是，正则化项的惩罚程度由正则化参数 决定。当 的值较小时，正则化项的影响减弱，回归系数的估计值更接近于普通最小二乘法的结果。

岭回归示例

```
code
```

7.2.3 模型的正则化 Regularization

模型正则化的一般形式:

$$argmin_{\beta}/argmax_{\beta}J(\beta|y,X,f)$$
$$subject\ to\ w(\beta) \leq s$$

w( ) 也称之为模型的惩罚项
调整参数 (Tuning parameter): (s)
- 预设 s，则为一个最优化问题
- 利用交叉验证求解 s，则为一系列最优化问题

机器学习中的正则化:
- 通过引入调整参数，使得优化泛化误差成为机器学习中的重要任务。
- 通过对模型进行合理约束，使得线性模型的应用更为灵活，适用范围更广。
- 是处理高维数据中的最常用方法。

常用的正则化方法：- Ridge - Lasso - Elastic Net - Graphic lasso - SCAD(Non-concave)

8 变量选择：Lasso 回归 Variable Selection:Lasso

8.1 Lasso 回归及其基本理论

Lasso 回归（*Least Absolute Shrinkage and Selection Operator Regression*）是一种用于变量选择和稀疏回归的线性回归方法。与岭回归不同，Lasso 回归使用 L1 正则化项来限制回归系数的大小。

Lasso 回归的目标是最小化以下形式的目标函数：

$$argmin\beta[(Y - X\beta)^T(Y - X\beta) + \lambda \parallel \beta \parallel_1]$$

其中，Y 是因变量的观测值，X 是自变量的观测值组成的矩阵， 是回归系数向量， 是回归系数的 L1 范数， 是正则化参数。

在目标函数中，第一项表示残差平方和，衡量了模型对观测值的拟合程度；第二项是 L1 正则化项，衡量了回归系数的绝对值之和。通过调节正则化参数 的值，可以控制正则化项对回归系数的惩罚程度。

Lasso 回归与岭回归相比，具有一个重要的**特点**：它可以将某些回归系数估计为零，从而实现变量选择的功能。这是因为 L1 正则化项具有稀疏性，会倾向于将一些不重要的变量的系数缩小为零，从而筛选出对目标变量具有显著影响的预测变量。

“稀疏性” 假设 (Sparsity)：- 高维数据包含无效信息
- 无效变量对应的系数为 0
- 系数不为 0 的变量，个数有限——稀疏性
- 稀疏估计具有更好的模型 “解读性”
- 丢掉冗余信息对模型影响不大

Lasso 回归的理论基础涉及到最小角回归（Least Angle Regression）和坐标下降算法（Coordinate Descent Algorithm）等优化算法。这些算法可以高

效地求解 Lasso 回归问题，得到最优的回归系数估计。

需要注意的是，Lasso 回归在引入正则化项的同时，可能会导致估计的回归系数偏向于缩小或取零，这可以降低共线性的影响和模型的复杂度，但也可能损失一些预测能力。因此，在选择正则化参数 时，需要进行适当的调参和模型评估，以平衡拟合度和泛化能力。

8.2 Elastic Net 模型介绍

弹性网络（Elastic Net）是一种结合了岭回归和 Lasso 回归的线性回归模型，用于解决高维数据集中存在多重共线性和变量选择的问题。弹性网络的目标函数是通过同时引入 L1 和 L2 正则化项来约束回归系数的大小。它的形式如下：

$$argmin\beta[(Y - X\beta)^T(Y - X\beta) + \lambda_1 \parallel \beta \parallel_1 + \lambda_2\beta^T\beta]$$

其中，Y 是因变量的观测值，X 是自变量的观测值组成的矩阵， 是回归系数向量，λ_1 和 λ_2 是正则化参数。

在目标函数中，第一项表示残差平方和，衡量了模型对观测值的拟合程度；第二项是 L1 正则化项，衡量了回归系数的绝对值之和；第三项是 L2 正则化项，衡量了回归系数的平方和。通过调节正则化参数 λ_1 和 λ_2 的值，可以控制正则化项对回归系数的惩罚程度。

弹性网络综合了 L1 和 L2 正则化的优点。L1 正则化项可以实现变量选择，将某些回归系数估计为零，从而筛选出对目标变量具有显著影响的预测变量；而 L2 正则化项可以降低共线性的影响，提高模型的稳定性。

在实际应用中，弹性网络的正则化参数需要进行适当的调参和模型评估，以平衡拟合度和泛化能力。通常可以使用交叉验证等方法来选择最优的正则化参数。

弹性网络模型在处理高维数据集和存在多重共线性的情况下表现出色，被广泛应用于特征选择和预测建模等领域。

Elastic Net 弹性网络模型的启示：
- 显示数据不严格满足模型假设 - 一些模型可能理论上不美，但有用 - 结合需要，组合不同的正则化项 (一般性的最优化 + 正则化问题求解：基于 matlab 手搓正则化项)

8.3 Lasso 和 Elastic Net 的实现

8.3.1 如何学习一个新的 R package

- **首先看简介**

- 关键词: Vignette(简介) / tutorial(指南)
- http:?/ • **其次看说明**
- Manual
- R help

(看说明，跑例子，然后用自己的数据替换掉例子数据，即可快速完成)

- **最后看原始文献**

8.3.2 建模所需 R package 及步骤

glmnet 说明:

$$argmin_{\beta}(y-x\beta)'(y-x\beta)$$

$$\alpha \parallel \beta \parallel_1 +(1-\alpha) \parallel \beta \parallel_2^2$$

- $\alpha = 1$,Lasso 模型
- $\alpha = 0$,Ridge 岭回归模型
- $\alpha \in (-1, 1)$,Elastic Net 模型

第一步: 按照要求准备数据

x:input matrix, of dimension nobs x nvars; each row is an observation vector. Can be in sparse matrix format (inherit from class "sparseMatrix" as in package Matrix)

y:response variable. Quantitative for family="gaussian", or family="poisson" (non-negative counts). For family="binomial" should be either a factor with two levels, or a two-column matrix of counts or proportions (the second column is treated as the target class; for a factor, the last level in alphabetical order is the target class). For family="multinomial", can be a nc>=2 level factor, or a matrix with nc

columns of counts or proportions. For either “binomial” or “multinomial”, if y is presented as a vector, it will be coerced into a factor. For family=“cox”, preferably a Surv object from the survival package: see Details section for more information. For family=“mgaussian”, y is a matrix of quantitative responses.

第二步：放数据进模型——不报错，有结果

第三步：根据知识调整模型 (参数) alpha 值

lambda 值

y 的数据类型调整为计数类型 (Poisson)

第四步：魔性诊断——交叉验证 • 手动交叉验证
- sample() + for 循环
• 自动交叉验证 (内嵌交叉验证函数)

第五步：模型应用对内应用模型计算指标 (如 metabolic signature)
对外应用模型进行预测。

8.3.3 glmnet() 函数参数介绍

```
# Usage
glmnet(
  x,
  y,
  family = c("gaussian", "binomial", "poisson", "multinomial",
  "cox", "mgaussian"),
  weights = NULL,
  offset = NULL,
  alpha = 1,
  nlambda = 100,
  lambda.min.ratio = ifelse(nobs < nvars, 0.01, 1e-04),
  lambda = NULL,
  standardize = TRUE,
  intercept = TRUE,
```

```
  thresh = 1e-07,
  dfmax = nvars + 1,
  pmax = min(dfmax * 2 + 20, nvars),
  exclude = NULL,
  penalty.factor = rep(1, nvars),
  lower.limits = -Inf,
  upper.limits = Inf,
  maxit = 1e+05,
  type.gaussian = ifelse(nvars < 500, "covariance", "naive"),
  type.logistic = c("Newton", "modified.Newton"),
  standardize.response = FALSE,
  type.multinomial = c("ungrouped", "grouped"),
  relax = FALSE,
  trace.it = 0,
  ...
)
```

R 语言 glmnet 函数的 alpha 参数是什么作用? 和 lambda 参数的区别是? R 语言的 glmnet 函数中的 alpha 参数用于控制弹性网络模型中 L1 和 L2 正则化项的混合比例，从而影响回归模型的特性。

具体来说，alpha 参数的取值范围是从 0 到 1，其中 0 表示纯 L2 正则化（岭回归），1 表示纯 L1 正则化（Lasso 回归），0.5 表示 L1 和 L2 正则化的混合（弹性网络）。

当 alpha 取值为 0 时，模型使用纯 L2 正则化，这意味着强调回归系数的平方和，并且不会产生稀疏性，即所有的回归系数都会保留。

当 alpha 取值为 1 时，模型使用纯 L1 正则化，这意味着强调回归系数的绝对值之和，并且可以实现变量选择，将某些回归系数估计为零，从而筛选出对目标变量具有显著影响的预测变量。

当 alpha 取值为 0.5 时，模型使用 L1 和 L2 正则化的混合（弹性网络），可以在稀疏性和拟合度之间进行权衡。弹性网络综合了 L1 和 L2 正则化的优点，既可以降低共线性的影响，提高模型的稳定性，又可以进行变量选择。

与 lambda 参数不同，alpha 参数主要影响正则化项的类型和混合比例，

lambda 参数用于调节正则化项的惩罚程度。通过调节 alpha 参数的取值，可以在 L1 和 L2 正则化之间进行选择，以获得最佳的回归模型。

R 语言 glmnet 函数的 lambda 参数是什么作用? 在 R 语言的 glmnet 函数中，lambda 参数用于控制正则化项的惩罚程度，从而影响回归模型的稀疏性和拟合度。

具体来说，lambda 参数是一个正则化参数的序列，它控制 L1 和 L2 正则化项在目标函数中的权重。lambda 的取值范围从大到小，表示正则化的强度逐渐增加。

当 lambda 取值较大时，正则化项的惩罚程度较强，模型的稀疏性较高，即可以将一些不重要的变量的系数估计为零，实现变量选择的功能。这有助于减少模型的复杂度和降低过拟合的风险。

当 lambda 取值较小时，正则化项的惩罚程度较弱，模型的拟合度较高。此时，回归系数的估计值更接近于普通最小二乘法的结果，模型可以更好地拟合训练数据。

在 glmnet 函数中，lambda 参数是通过指定 lambda 参数的类型和取值范围来确定的。常见的 lambda 类型包括"lambda.min"、"lambda.1se"和"lambda.sequence"。其中，"lambda.min" 是选择最小的 lambda 值，使得模型的误差达到最小；"lambda.1se" 是选择最大的 lambda 值，使得模型的误差与最小误差之间的差距在一个标准误范围内；"lambda.sequence" 是给定一个 lambda 序列，用于探索不同正则化强度下的模型效果。

通过调节 lambda 参数的取值，可以在稀疏性和拟合度之间进行权衡，得到最优的回归模型。

R 语言 glmnet 函数的 relax 参数是什么作用? 在 R 语言的 glmnet 函数中，relax 参数是用于控制弹性网络模型中 L1 和 L2 正则化项之间的松弛度。

具体来说，relax 参数的取值范围是从 0 到 1，其中 0 表示严格的 L1 和 L2 正则化，1 表示完全松弛的 L1 和 L2 正则化。

当 relax 取值为 0 时，模型使用严格的 L1 和 L2 正则化，这意味着 L1 和 L2 正则化项同时起作用，不会产生松弛。

当 relax 取值为 1 时，模型使用完全松弛的 L1 和 L2 正则化，这意味着 L1 和 L2 正则化项不再同时起作用，只有其中一项起作用。

当 relax 取值介于 0 和 1 之间时，模型使用部分松弛的 L1 和 L2 正则化，正则化项之间的权重会被调整，以产生介于严格和完全松弛之间的效果。

通过调节 relax 参数的取值，可以在 L1 和 L2 正则化之间进行灵活的权衡，调整正则化项之间的相对权重。这有助于进一步控制回归模型的稀疏性和拟合度。

需要注意的是，relax 参数只在使用 glmnet 函数的 elnet 模型（即弹性网络模型）时才有意义，在其他模型中不适用。

R 语言 glmnet 函数的 relax 参数和 alpha 参数的区别是什么? 在 R 语言的 glmnet 函数中，relax 参数和 alpha 参数都是用于控制弹性网络模型中 L1 和 L2 正则化项的混合比例和松弛度，但它们的作用方式略有不同。

1. Alpha 参数：alpha 参数用于控制 L1 和 L2 正则化项的混合比例。它的取值范围是从 0 到 1，其中 0 表示纯 L2 正则化（岭回归），1 表示纯 L1 正则化（Lasso 回归），0.5 表示 L1 和 L2 正则化的混合（弹性网络）。通过调节 alpha 参数的取值，可以在 L1 和 L2 正则化之间进行权衡，从而影响模型的稀疏性和拟合度。

2. Relax 参数：relax 参数用于控制 L1 和 L2 正则化项之间的松弛度。它的取值范围是从 0 到 1，其中 0 表示严格的 L1 和 L2 正则化，1 表示完全松弛的 L1 和 L2 正则化。通过调节 relax 参数的取值，可以调整正则化项之间的相对权重，从而影响模型的稀疏性和拟合度。

从作用方式上来说，alpha 参数主要影响正则化项的类型和混合比例，而 relax 参数主要影响正则化项之间的相对权重和松弛度。它们可以一起使用，以进一步调节弹性网络模型的特性。

R 语言 glmnet 函数的 trace.it 参数的作用是什么? If trace.it=1, then a progress bar is displayed; useful for big models that take a long time to fit.

如果 trace.it=1，则显示进度条；对于需要很长时间来拟合的大模型很有用。