DeepL
订阅DeepL Pro以翻译大型文件。
欲了解更多信息，请访问www.DeepL.com/pro。

COMP9417 - 机器学习教程。回归 I

在本教程中，我们将探讨线性回归的理论基础。我们首先研究一维的线性回归（单变量线性回归），然后将分析扩展到多变量线性回归。如果你发现自己对本教程中涉及的任何数学知识不确定，我们强烈建议你阅读文本中的相应材料（可在网上免费获取）。

Marc Peter Deisenroth, A. Aldo Faisal和Cheng Soon Ong的《机器学习的数学》。在整个课程中

，我们将把它称为MML书。

问题1. (单变量最小二乘法)

一个*单变量线性回归模型*是一个线性方程$y = w_0 + w_1 x$。学习这样一个模型需要将其拟合到训练数据的样本中，$(x_1, y_1), \ldots, (x_n, y_n)$，以便使损失最小化(通常是Mean
平方误差（MSE）），$L = \frac{1}{n}\sum_{i=1}^{n}(y_i - (w_0 + w_1 x_i))^2$。要找到最佳参数$w_0$和$w_1$，即
为了使这个误差函数最小化，我们需要找到误差*梯度* $\frac{\partial L}{\partial w_0}$和$\frac{\partial L}{\partial w_1}$所以我们需要推导出这些表达式的偏导数，将它们设为零，并求解w_0和w_1。

(a) 推导出单变量线性回归模型的最小二乘估计值（MSE损失函数的最小者）。

解决方案。

首先，我们把单变量线性回归的损失函数$y = w_0 + w_1 x$写为

$$L = L(w_0, w_1) = \frac{1}{n}\sum_{i=1}^{n}(y_i - (w_0 + w_1 x_i))^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - w_0 - w_1 x_i)^2$$

在L的最小值时，相对于w_0，w_1的偏导应该是零。我们将首先取L相对于w的偏导，$_0$。

$$\frac{\partial L}{\partial w_0} = \frac{\partial}{\partial w_0}\frac{1}{n}\sum_{i=1}^{n}(y_i - w_0 - w_1 x_i)^2$$
$$\frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial w_0}(y_i - w_0 - w_1 x_i)^2$$
$$\frac{1}{n}\sum_{i=1}^{n} -2(y_i - w_0 - w_1 x_i)$$
$$= -2\left[\frac{1}{n}\sum_{i=1}^{n} y_i - w_0\frac{1}{n}\sum_{i=1}^{n} 1 - w_1\frac{1}{n}\sum_{i=1}^{n} x_i\right]$$
$$= -2\left[\bar{y} - w_0 - w_1\bar{x}\right]$$

其中，我们引入的符号f是指f的样本平均值，即$\bar{f} = \frac{1}{m}\sum_{j=1}^{m} f_j$
其中m是f的长度。现在，我们将其等同于零，并求解w_0，得到

$$-2\left[\bar{y} - w_0 - w_1\bar{x}\right] = 0 \Rightarrow w_0 = \bar{y} - w_1\bar{x}$$

注意，我们实际上还没有求出w_0，因为我们的表达式取决于w_1，我们也必须对其进行优化。取L关于w的偏导。

$$\frac{\partial L}{\partial w_1} = \frac{\partial}{\partial w_1}\frac{1}{n}\sum_{i=1}^{n}(y_i - w_0 - w_1 x_i)^2$$
$$\frac{1}{n}\sum_{i=1}^{n} -2x_i(y_i - w_0 - w_1 x_i)$$
$$= -2\left[\frac{1}{n}\sum_{i=1}^{n} x_i y_i - w_0\frac{1}{n}\sum_{i=1}^{n} x_i - w_1\frac{1}{n}\sum_{i=1}^{n} x_i^2\right]$$
$$= -2\left[\overline{xy} - w_0\bar{x} - w_1\overline{x^2}\right]$$

现在，我们将其等同于零，并求解w_1，得到

$$-2\left[\overline{xy} - w_0\bar{x} - w_1\overline{x^2}\right] = 0 \Rightarrow w_1 = \frac{\overline{xy} - w_0\bar{x}}{\overline{x^2}}$$

现在我们有一个关于w_0的表达式，即w_1，和一个关于w_1的表达式，即w_0。这些被称为正常方程。为了得到w_0、w_1的显式解，我们可以将w_0*插入*w_1并求解。

$$w_1 = \frac{\overline{xy} - w_0\bar{x}}{\overline{x^2}}$$
$$\frac{\overline{xy} - (\bar{y} - w_1\bar{x})\bar{x}}{\overline{x^2}}$$
$$\frac{\overline{xy} - \bar{x}\bar{y} + w_1\bar{x}^2}{\overline{x^2}}$$

重新排列并求解w_1，得到的是

$$w_1 = \frac{\overline{xy} - \bar{x}\bar{y}}{\overline{x^2} - \bar{x}^2}$$

因此，现在我们有了回归参数w_0和w_1的明确解决方案，因此我们已经完成了。

(b) 证明数据的中心点，即点（x，y）总是在最小二乘回归线上。

解决方案。

最小二乘法的回归线是。

$$\hat{y}(x) = w_0 + w_1 x$$
$$\bar{y} - \frac{\overline{xy} - \bar{x}\bar{y}}{\overline{x^2} - \bar{x}^2}\bar{x} + \frac{\overline{xy} - \bar{x}\bar{y}}{\overline{x^2} - \bar{x}^2} x,$$

其中 "帽子 "符号通常用来表示估计值或预测值，在这种情况下，$\hat{y}(x)$是我们在评价点x处对y的估计值，将$x=\bar{x}$代入，得到。

$$\hat{y}(\bar{x})=\bar{y}。$$

因此，最小二乘法回归线必须通过数据中心点：（$\bar{x}$，$\bar{y}$）。

(c) 为了确保你理解这个过程，试着用 "L2 "正则化版本来解决以下线性回归的损失函数，在这个版本中，我们添加了一个惩罚，惩罚w的大小$_1$。让$\lambda>0$，考虑正则化损失

$$L(w_0, w_1) = \frac{1}{n}\sum_{i=1}^{n}(y_i - (w_0 + w_1 x_i))^2 + \lambda w_1^2$$

解决方案。

对新问题重复上述步骤，可以得到

$$w_0 = \bar{y} - w_1 \bar{x}$$
$$w_1 = \frac{\overline{xy} - \bar{x}\bar{y}}{\overline{x^2} - \bar{x}^2 + \lambda}.$$

问题2. (多元最小二乘法)

在上一个问题中，我们找到了单变量（单一特征）问题的最小二乘法解决方案。现在我们把它推广到有p个特征的情况下。让$x_1, x_2, \ldots, x_n$是n个特征向量（如

对应于n个实例）在$\mathbb{R}^p$，即。

$$x_i = \begin{matrix} x_{i0} \\ x_{i1} \\ \vdots \\ x_{i,p-1} \end{matrix}$$

我们将这些特征向量堆叠成一个矩阵，$X \in \mathbb{R}^{n \times p}$，称为*设计*矩阵。惯例是将特征向量堆叠起来，使$X$的每一行都对应于一个特定的实例，也就是说。

$$X = \begin{matrix} x_1^T \\ x_2^T \\ \vdots \\ x_n^T \end{matrix} = \begin{matrix} x_{10} & x_{11} & \cdots & x_{1,p-1} \\ x_{20} & x_{21} & \cdots & x_{2,p-1} \\ \vdots & \vdots & & \vdots \\ x_{n0} & x_{n1} & \cdots & x_{n,p-1} \end{matrix}$$

其中上标T表示转置操作。请注意，标准的做法是将特征向量的第一个元素取为1，以考虑到偏置项，因此我们将假设$x_{i0} = 1$，对于所有的
$i = 1, \ldots$与前一个问题类似，我们的目标是学习一个权重向量$w \in \mathbb{R}^p$，并进行预测。

$$\hat{y}_i = w^T x_i = w_0 + w_1 x_{i1} + w_2 x_{i2} + \cdots + w_{p-1} x_{i,p-1} \text{。}$$

其中$\hat{y}_i$表示*第*i*个*预测值。为了解决w中的最佳权重，我们可以使用与之前相同的程序并使用MSE。

$$\begin{aligned} L(w_0, w_1, \ldots, w_{p-1}) &= \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \\ &= \frac{1}{n} \sum_{i=1}^{n} (y_i - (w_0 + w_1 x_{i1} + w_2 x_{i2} + \cdots + w_{p-1} x_{i,p-1}))^2 \text{。} \end{aligned}$$

解决这个问题的一个方法是对每一个p权重进行导数，并解决由此产生的方程，但这将是**非常**乏味的。解决这个问题的更有效方法是使用矩阵符号。我们可以把上述损失写成。

$$L(w) = \frac{1}{n} \| y - Xw \|_2^2 \text{。}$$

其中$\| \cdot \|_2$是欧几里得规范。在本问题的其余部分，我们将假设X是一个全阶矩阵，这意味着我们能够计算X的逆$X^T X$。

(a) 证明$L(w)$有一个临界点。

$$\hat{w} = (X^T X)^{-1} X^T y \text{。}$$

注：这里的临界点是指$\left(\frac{dL(w)}{w}\right)(\hat{w}) = 0$，即在临界点评价的梯度为零）。

提示1：如果u是$\mathbb{R}^n$中的一个向量，而v是$\mathbb{R}^n$中的一个固定向量，那么$\frac{\partial v^T u}{\partial u} = v$。

提示2：如果A是一个固定的$n \times n$矩阵，如果$f = z^T A z$，那么$\frac{\partial u^T A u}{\partial u} = Au + A^T u$。

解决方案

。

我们有

$$\|y - Xw\|_2^2 = (y - Xw)^T (y - Xw)$$
$$= y^T y - 2y^T Xw + w^T X^T Xw。$$

为了使上述情况最小化，我们对w进行导数，并设定等于零，如下所示。

$$\frac{d}{dw}(y^T y - 2y^T Xw + w^T X^T Xw) = -2X^T y + 2X^T Xw = 0。 \quad (一组)$$

在上面求解w，可以得到。

$$2X^T Xw = 2X^T y \Rightarrow \hat{w} = (X^T X)^{-1} X^T y$$

(b) $(\frac{dL(w)}{dw})(\hat{w})=0$这个条件对于证明$\hat{w}$是$L$的(全局)最小值是必要的，但并不充分，因为这个点可能是局部最小值或鞍点。证明(a)部分中的临界点确实是L的全局最小者。

提示1：$L(w)$是$w \in \mathbb{R}^p$的函数，因此它的Hessian，H，是二阶偏导的$p \times p$矩阵，也就是说，H的(k, l)个元素是

$$Hkl = \frac{\partial L(w)}{\partial w_k \partial w_l}。$$

我们通常写$H=\nabla^2 L(w)$，其中∇_w是梯度算子，∇^2 表示取梯度两次。请注意，对于多变量函数来说，Hessian起到了二阶导数的作用。

提示2: 如果一个函数的Hessian是正半无限的，那么它就是凸的，这意味着对于任何矢量u。

$$u^T Hu \geq 0。$$

还要注意的是，这个条件意味着对于任何u的选择，乘积项将总是非负的。

提示3：凸函数的任何临界点都是一个全局最小值。

解决方案。

利用第一个提示，我们首先计算L的Hessian。

$$\nabla_w^2 L(w) = \nabla_w (\nabla_w L(w))$$
$$= \nabla_w (-2X^T y + 2X^T Xw)$$
$$= 2X^T X。$$

然后，对于任何矢量$u \in \mathbb{R}^p$，我们有

$$u^T (2X^T X)u = 2(Xu)^T (Xu) = 2\|Xu\|_2^2 \geq 0$$

因为规范总是非负的。因此，L确实是凸的，所以临界点是先前发现的确实是L的全局最小值。

(c) 在接下来的部分，我们将使用上面得出的公式来验证我们在单变量情况下的解决方案。我们假设p=2，所以我们有一个二维的特征向量（一个项为截距，另一个为我们的特征）。写下以下数值。

$$x_i \text{。} \quad y, \quad w, \quad X, \quad X^T X \text{。} \quad (X^T X)^{-1} \text{。} \quad X^T y \text{。}$$

解决方案。

我们有以下结果。

$$x_i = \begin{bmatrix} 1 \\ x_i \\ 1 \end{bmatrix} \in \mathbb{R}^2 \text{。} \qquad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \in \mathbb{R}^n, \qquad w = \begin{bmatrix} 1 \\ w_0 \\ w_1 \end{bmatrix} \in \mathbb{R}^2$$

$$X = \begin{bmatrix} 1 & x_{11} \\ 1 & x_2 \\ \vdots & 1. \\ 1 & x_{n1} \end{bmatrix} \in \mathbb{R}^{n \times 2} \qquad X^T X = \begin{bmatrix} 1 & \bar{x} \\ n & n\bar{x} \\ n\bar{x} & n\bar{x} \\ & 2 \end{bmatrix} \in \mathbb{R}^{2 \times 2}$$

最后，我们有

$$(X^T X)^{-1} = \frac{1}{n(x_2 - \bar{x}^2)} \begin{bmatrix} \bar{x^2} & -\bar{x} \\ -\bar{x} & 1 \end{bmatrix}, \qquad X^T y = \begin{bmatrix} 1 & \bar{y} \\ n\bar{y} & \\ n\bar{x}\bar{y} & \end{bmatrix}$$

(d) 利用上一部分的结果，计算p=2情况下的最小二乘估计。

解决方案。

将（a）和（b）部分的结果放在一起。

$$\hat{w} = (X^T X)^{-1} X^T y = \begin{bmatrix} \bar{y} - \hat{w}_1 \bar{x} \\ \frac{\bar{xy} - \bar{x}\bar{y}}{\bar{x^2} - \bar{x}^2} \end{bmatrix},$$

这与前一个问题中的暴力方法完全相同。

(e) 考虑以下问题：我们有输入$x_1, \ldots, x_5 = 3, 6, 7, 8, 11$，输出 $y_1, \ldots, y_5 = 13, 8, 11, 2, 6$。计算最小二乘法的解决方案，并通过手工和使用python绘制结果。最后，使用sklearn实现来检查你的结果。

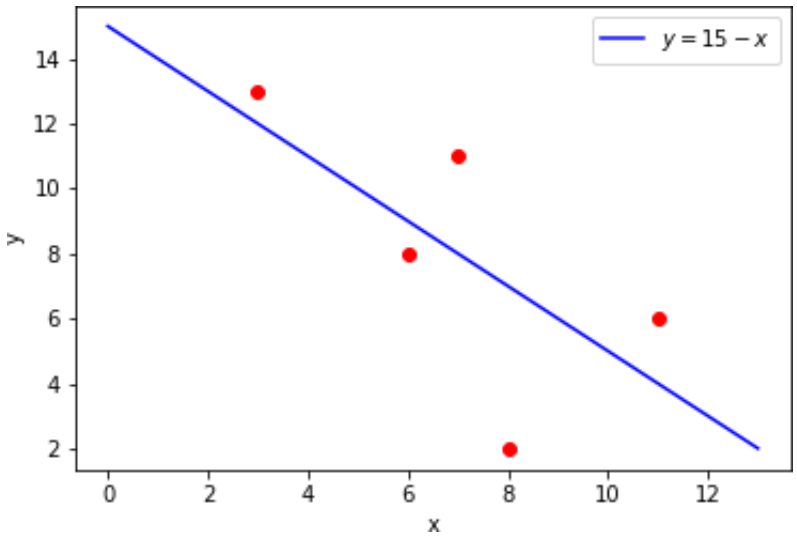
解决方案。

这应该是很简单的，我们可以使用下面的python代码。

```
import matplotlib.pyplot as plt
import numpy as np
```

```
from sklearn import linear_model

y = np.array([13, 8, 11, 2, 6])
x = np.array([3, 6, 7, 8, 11])
n = x.shape[0] 。
X = np.stack((np.ones(n), x), axis=1)


# 计算最小二乘法解决方案 XTX = X.T @ X
XTXinv = np.linalg.inv(XTX)
XTy = X.T @ y
最小二乘法估计 = XTXinv @ XTy


# sklearn比较
model =
linear_model.LinearRegression()
model.fit(x.reshape(-1,1), y)

# 谋划
xx = np.linspace(0,13,1000)
plt.scatter(x, y, color='red')
plt.plot(xx, 15. - xx, color='blue', label='$y = 15 - x$')
plt.legend()
plt.xlabel('x');
plt.ylabel('y')
plt.savefig("LSLine.png")
```

我们最终应该得到以下结果



(f) **高级。**你们中的一些人可能对我们的假设感到担忧，即X是一个全等级矩阵，这是为了确保$X^T X$是一个可逆矩阵而做出的假设。如果$X^T X$不是可逆的，那么我们就无法计算出最小二乘法的解决方案。所以会发生什么

如果X不是全等级的？最小二乘法会失败吗？答案是否定的，我们可以使用一种叫做伪逆的东西，也被称为摩尔-彭罗斯逆。这不在本课程的范围之内，感兴趣的读者可以参考下面的注释，或者MML的第二章。在很高的层次上，伪逆是一个矩阵，其作用类似于不可逆矩阵的逆。在NumPy中，我们可以用'np.linalg.pinv'命令轻松计算伪逆。

(g) 讨论一下*特征图*的概念。你将如何在最小二乘回归的背景下使用特征图？

解决方案。

如果我们处理的是一个回归问题，我们有数据(x_i, y_i)，$i = 1, \ldots n$，其中$x_i \in \mathsf{R}^p$，特征图是任何函数$\varphi : \mathsf{R}^p \to \mathsf{R}^K$，它将数据从原始$p$维空间映射到某个新的$K$维空间。$K$可以是大于或小于
p.例如，如果p=1，那么一个特征图可能看起来像。

$$\varphi : \mathsf{R} \to \mathsf{R}^4 \qquad \varphi(x) = [x, x^2, x^3, \log x]^T .$$

也就是说，我们将每个点映射到一个四维空间，在那里我们采取多项式和对数转换形式。这样做可能会产生一个更好的模型（下周会有更多这方面的内容）。考虑另一个例子，其中$p = 3$，所以$x = [x_1, x_2, x_3]^T$，那么我们可能选择考虑特征图。

$$\varphi : \mathsf{R}^3 \to \mathsf{R}^2 \qquad \varphi(x) = \begin{bmatrix} 1 \\ \frac{1}{2}(x_1 + x_2), x_3^2 \end{bmatrix}^T$$

我们看到，在最初的设定中，我们的最小二乘回归模型是

$$\hat{y}_i = w^T x_i .$$

而现在我们的模型只是

$$\hat{y}_i = w^T \varphi(x_i) .$$

因此，除了我们处理的是一个K维的问题而不是p维的问题外，其实没有什么变化。我们只需构建新的设计矩阵

$$\Psi = \begin{bmatrix} \varphi(x_1)^T \\ \varphi(x_2)^T \\ \vdots \\ \varphi(x_n)^T \end{bmatrix} \in \mathsf{R}^{n \times K}$$

并解决同样的最小化问题。

$$\hat{w} = \arg\min_{w \in R^K} \frac{1}{n} \| y - \Phi w \|^2 .$$

这就产生了

$$\hat{w} = (\Phi^T \Phi)^{-1} \Phi^T y.$$

(h) 平均平方误差（MSE）为

$$\text{MSE}(w) = \frac{1}{n}\|y - Xw\|_2^2。$$

而平方误差之和（SSE）（也被称为残差平方之和（RSS））为

$$\text{SSE}(w) = \|y - Xw\|_2^2。$$

以下陈述是真的还是假的。解释一下原因。

(i) $\arg\min_{w\in\mathbb{R}^p} \text{MSE}(w) = \arg\min_{w\in\mathbb{R}^p} \text{SSE}(w)$

(ii) $\min_{w\in\mathbb{R}^p} \text{MSE}(w) = \min_{w\in\mathbb{R}^p} \text{SSE}(w)$

符号：记得$\min_x g(x)$是$g(x)$的最小值，而$\arg\min_x g(x)$是x的值使$g(x)$最小化。所以$\min_x (x-2)^2 = 0$，但$\arg\min_x (x-2)^2 = 2$。

解决方案。

第一句话是正确的，但第二句话是不正确的。为了理解这一点，让我们考虑一个更简单的例子：将函数$g(x)=(x-2)^2+4$最小化。一个简单的计算表明

$$\arg\min_{x\in R} g(x) = 2。$$

和

$$\arg\min_{x\in R} \frac{1}{n} g(x)=2。$$

换句话说，如果你把函数乘以一个（正）常数，这并不重要，最小化器始终是相同的。然而。

$$\min_{x\in R} g(x) = g(2)=4。$$

和

$$\text{最小值}_{x\in R} \frac{1}{n} g(x) = \frac{1}{n} g(2) = \frac{4}{n}。$$

因此，综上所述，在寻找最小二乘法问题的最小化器时，如果我们考虑MSE或SSE并不重要，我们最终会得到相同的解决方案。但重要的是对目标的最小值的解释。MSE给出了一个模型的平均平方误差的概念，而SSE给出了总平方误差。因此，MSE在某种程度上更有意义，因为你可能有一个更高的SSE，只是因为你有一个更大的样本量n。

问题3（人口与样本参数）。

(a) 群体和样本之间的区别是什么？

解决方案。

人口是我们有兴趣了解的数量，而样本是我们可以接触到的人口的子集。例如，在COVID-19疫苗试验中，人群是地球上的全部人类。理想情况下，我们希望了解一剂疫苗是如何降低整个人口中因COVID-19而死亡的风险的，但现实中我们无法进行这种研究。我们所做的是在规模小得多但希望有代表性的人口样本上进行临床试验，并试图估计/了解人口的情况。

(b) 什么是人口参数？我们怎样才能估计它？

解决方案。

人口参数是一些未知的东西，我们希望用数据来估计。例如，假设我们对澳大利亚所有袋鼠的平均体重感兴趣。捕捉并称量所有袋鼠是不可能的，所以这个数量不可能被准确知道。在处理数据时，建模者要对真实的基本人口做出某些合理的假设，使统计问题变得更加容易解决。

一个合理的假设是，袋鼠的体重是正态分布，平均值为μ，标准差为10（当然我们也可以把标准差当作未知数，但现在我们只关注平均值）。这里的假设让我们想到了权重的袋鼠作为一个随机变量，我们用X表示，我们写$X \sim N(\mu, 10^2)$。现在，假设我们只能接触到100只袋鼠（在当地动物园），我们对每只袋鼠进行称重，并且记录这些数字--我们现在可以获得一个样本，我们把它表示为$X_1, \ldots, X_{100}$。既然我们知道$X_i \sim N(\mu, 10^2)$，我们可以尝试用数据来估计μ。有很多方法来构建人口平均数的估计器，而一个常见的估计器是样本平均数。

$\bar{X} = \frac{1}{100} \sum_{i=1}^{n} X_i$。另一个例子是样本中位数。请注意，在一般情况下，我们用希腊语用字母，如μ和σ来指代种群参数，一般不应使用他们指的是样本统计，如样本平均值或样本标准差。

重要的是，不要过于纠结于人口代表现实世界中的一些实际人口，最好是把人口参数看作是我们想知道但无法测量的东西。一个需要牢记的具体例子是抛硬币的例子。我们知道硬币有两面，头/尾。假设硬币可能是弯曲的，所以我们不知道得到正面的真正概率。在这种情况下，我们可以把p看作是我们的人口参数。我们可以通过多次抛掷硬币来收集数据次，并记录其结果。那么，一个好的统计学假设是：$X \sim \text{Bernoulli}(p)$。

从现在开始，每当你看到语句$X_1, \ldots, X_n \sim F(\theta)$，其中$F$是一些分布，θ是一些参数，你可以简单地假设我们对知道θ感兴趣，但只能够获得样本（数据）$X_1, \ldots, X_n$，我们想用一种聪明的方式来估计θ.