3.1 线性回归重点摘录与练习解答

（1）线性模型

对于高维数据集，建模时采用线性代数表示法会比较方便。当我们的输入包含 d 个特征时，我们将预测结果 $\hat{y}$（通常使用“尖角” 符号表示 y 的估计值）表示为：

$$\hat{y} = w_1 x_1 + \ldots + w_d x_d + b.$$

将所有特征放到向量 $\mathbf{x} \in \mathbb{R}^d$ 中，并将所有权重放到向量 $\mathbf{w} \in \mathbb{R}^d$ 中，我们可以用点积形式来简洁地表达模型：

$$\hat{y} = \mathbf{w}^\top \mathbf{x} + b.$$

对于 y 也是高维的情况，设特征集合 $\mathbf{X}$，预测值 $\hat{\mathbf{y}} \in \boldsymbol{R}^n$ 可以通过矩阵-向量乘法表示为：

$$\hat{\mathbf{y}} = \mathbf{X}\mathbf{w} + b$$

这个过程中的求和将使用在2.1.3节中有详细介绍的广播机制。给定训练数据特征 $\mathbf{X}$ 和对应的已知标签 $\mathbf{y}$，线性回归的目标是找到一组权重向量 $\mathbf{w}$ 和偏置 b：当给定从 $\mathbf{X}$ 的同分布中取样的新样本特征时，这组权重向量和偏置能够使得新样本预测标签的误差尽可能小。

（2）平方损失函数

当样本 i 的预测值为 $\hat{y}^{(i)}$，其相应的真实标签为 $y^{(i)}$ 时，平方误差可以定义为以下公式：

$$l^{(i)}(\mathbf{w}, b) = \frac{1}{2}\left(\hat{y}^{(i)} - y^{(i)}\right)^2.$$

为了度量模型在整个数据集上的质量，我们需要计算在训练集 n 个样本上的损失均值（也等价于求和）：

$$L(\mathbf{w}, b) = \frac{1}{n}\sum_{i=1}^{n} l^{(i)}(\mathbf{w}, b) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}\left(\mathbf{w}^\top \mathbf{x}^{(i)} + b - y^{(i)}\right)^2.$$

在训练模型时，我们希望寻找一组参数 $(\mathbf{w}^*, b^*)$，这组参数能最小化在所有训练样本上的总损失。如下式：

$$\mathbf{w}^*, b^* = \operatorname*{arg\,min}_{\mathbf{w}, b} L(\mathbf{w}, b).$$

（3）解析解

我们的预测问题是最小化 $\|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2$，将损失关于 $\mathbf{w}$ 的导数设为0，即为 $\dfrac{\partial l}{\partial \mathbf{w}} = 0$ 时，

$$\frac{\partial \|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2}{\partial \mathbf{w}} = \frac{\partial \frac{1}{2}(\mathbf{y} - \mathbf{X}\mathbf{w})^2}{\partial \mathbf{w}} = (\mathbf{y} - \mathbf{X}\mathbf{w}^*)(-\mathbf{X}^\top) = 0$$

得到解析解：

$$\mathbf{w}^* = (\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \mathbf{y}.$$

（4）随机梯度下降

随机度下降最简单的用法是计算损失函数（数据集中所有样本的损失均值）关于模型参数的导数（在这里也可以称为梯度）。但实际中的执行可能会非常慢：因为在每一次更新参数之前，我们必须遍历整个数据集。因此，我们通常会在每次需要计算更新的时候随机抽取一小批样本，这

种变体叫做小批量随机梯度下降（minibatch stochastic gradient descent）。

在每次迭代中，我们首先随机抽样一个小批量 $\mathcal{B}$，它是由固定数量的训练样本组成的。然后，我们计算小批量的平均损失关于模型参数的导数（也可以称为梯度）。最后，我们将梯度乘以一个预先确定的正数 η，并从当前参数的值中减掉。

$$(\boldsymbol{w}, b) \leftarrow (\boldsymbol{w}, b) - \frac{\eta}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \partial_{(\boldsymbol{w}, b)} l^{(i)}(\boldsymbol{w}, b).$$

总结一下可知算法的步骤如下：

1. 初始化模型参数的值，如随机初始化；
2. 从数据集中随机抽取小批量样本且在负梯度的方向上更新参数，并不断迭代这一步骤。对于平方损失和仿射变换。

我们可以明确地写成如下形式:

$$\boldsymbol{w} \leftarrow \boldsymbol{w} - \frac{\eta}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \partial_{\boldsymbol{w}} l^{(i)}(\boldsymbol{w}, b) = \boldsymbol{w} - \frac{\eta}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \boldsymbol{x}^{(i)} \left(\boldsymbol{w}^\top \boldsymbol{x}^{(i)} + b - y^{(i)}\right),$$

$$b \leftarrow b - \frac{\eta}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \partial_b l^{(i)}(\boldsymbol{w}, b) = b - \frac{\eta}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \left(\boldsymbol{w}^\top \boldsymbol{x}^{(i)} + b - y^{(i)}\right).$$

（5）问题解答

1、假设有一些数据$x_1, \ldots, x_n \in \mathbb{R}$。目标是找到一个常数$b$，使得最小化$\sum_i (x_i - b)^2$，找到最优值$b$的解析解，这个问题及其解与正态分布有什么关系?

解：即求

$$b^* = \arg\min \sum_i (x_i - b)^2$$

令

$$\frac{\partial \sum_i (x_i - b)^2}{\partial b} = 0$$

即 $2\sum_i (x_i - b^*) = 0$，故

$$b^* = \frac{1}{n} \sum_i x_i.$$

我们先求取样本关于参数 b 的极大似然估计，令 $x_i = b + \epsilon$，其中$\epsilon \sim \mathcal{N}(0, \sigma^2)$，则似然函数为：

$$L(x \mid b) = (2\pi\sigma^2)^{-\frac{n}{2}} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^n (x_i - b)^2\right)$$

因此对数似然函数为：

$$-l(x \mid b) = -\log L(x \mid b) = \frac{n}{2} \log(2\pi\sigma^2) + \sum_{i=1}^n \frac{1}{2\sigma^2} (x_i - b)^2$$

若求 $\arg\max_b L(x \mid b)$，即求 $\arg\min_b -l(x \mid b)$.根据数理统计写出似然方程可知：

$$b^* = \frac{1}{n} \sum_{i=1}^n x_i.$$

也即求上一问题的解析解。因此，在高斯噪声的假设下，最小化均方误差等价于对线性模型的极大似然估计。

2、推导出使用平方误差的线性回归优化问题的解析解。为了简化问题，可以忽略偏置b（我们可以通过向X添加所有值为1的一列来做到这一点）。

解：这里只给出部分问题的解答，首先对用矩阵和向量表示法写出优化问题（将所有数据视为单个矩阵，将所有目标值视为单个向量）。

$$\hat{\boldsymbol{Y}}_{n,q} = \boldsymbol{X}_{n,d+1}\boldsymbol{w}_{d+1,q}$$

然后是计算损失对 w 的梯度：

$$L = \frac{1}{2}(\boldsymbol{Y} - \hat{\boldsymbol{Y}})^2$$

$$\frac{\partial L}{\partial \boldsymbol{w}} = \frac{\partial \frac{1}{2}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{w})^2}{\partial \boldsymbol{w}} = (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{w})(-\boldsymbol{X}^\top)$$

第三个小问是通过将梯度设为 0、求解矩阵方程来找到解析解。

$$(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{w})(-\boldsymbol{X}^\top) = 0$$

$$-\boldsymbol{X}^\top\boldsymbol{Y} + \boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{w} = 0$$

可得解析解表达式为：

$$\boldsymbol{w}^* = (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top\boldsymbol{Y}$$

解析解可能比使用随机梯度下降（SGD）更好的情况包括：

1. 简单问题：解析解通常适用于简单的问题，其中目标函数和约束条件很容易求导并求解。在这种情况下，直接计算解析解比使用SGD更高效。

2. 小规模数据集：对于小规模的数据集，计算解析解可以很快完成，并且由于数据量较小，解析解的计算开销相对较小。

3. 显式公式要求：某些应用场景可能要求得到显式的公式解析解，例如需要解释、推导或证明的问题。

然而，解析解的方法在以下情况下可能会失效：

1. 复杂问题：对于复杂的问题，目标函数和约束条件可能很难求导或求解，或者求解过程可能非常复杂甚至不存在解析解。在这种情况下，使用SGD等数值优化算法可能更适合。

2. 大规模数据集：对于大规模数据集，计算解析解的计算复杂度可能非常高，甚至无法完成。在这种情况下，SGD通常更具可行性和可扩展性。

3. 随机性和噪声：如果目标函数存在随机性或噪声，并且我们希望在优化过程中考虑到这些因素，那么SGD等迭代方法通常更合适，因为它们可以根据采样的随机梯度进行逐步的调整。