吃瓜打卡(2023/12/18)-task2

“第 3 章 线性模型” (“pumpkin_book.pdf”, p. 31)

“3.1 基本形式”(“pumpkin_book.pdf”, p. 31)

“当向量 中的元素用分号“;”分隔时表示此向量为列向量，用逗号“,”分隔时表示为行向量。因此，式 (3.2) 中 w = (w1; w2; ...; wd) 和 x = (x1; x2; ...; xd) 均为 d 行 1 列的列向量。” (“pumpkin_book.pdf”, p. 31)

“3.2 线性回归”(“pumpkin_bool.pdf”, p. 31)

“对于存在“序”关系的属性，可通过 连续化将其转化为带有相对大小关系的连续值；对于不存在“序”关系的属性，可根据属性取值将其拆解为 多个属性，” (“pumpkin_book.pdf”, p. 31)

“以上针对样本属性所进行的处理工作便是第 1 章 1.2 基本术语中提到的“特征工程”范畴，完成属性 数值化以后通常还会进行缺失值处理、规范化、降维等一系列处理工作。由于特征工程属于算法实践过程中需要掌握的内容，待学完机器学习算法以后，再进一步学习特征工程相关知识即可，” (“pumpkin_book.pdf”, p. 31)

“符号“arg min”，其中“arg”是“argument” （参 数）的前三个字母，“min”是“minimum”（最小值）的前三个字母，该符号表示求使目标函数达到最小值 的参数取值。” (“pumpkin_book.pdf”, p. 31)

“对比知道，“min”和“arg min”的区别在于，前者输出目标函数的最小值， 而后者输出使得目标函数达到最小值时的参数取值。” (“pumpkin_book.pdf”, p. 31)

“s.t.”是“subject to”的简写，意思是“受约束 于”，即为约束条件。” (“pumpkin_book.pdf”, p. 32)

“最 优化的教材（例如参考文献 [1]）” (“pumpkin_book.pdf”, p. 32)

““西瓜书”在式 (3.5) 左侧给出的凸函数的定义是最优化中的定义，与高等数学中的定义不同，本书也 默认采用此种定义。” (“pumpkin_book.pdf”, p. 32)

“由于一元线性回归可以看作是多元线性回归中元的个数为 1 时的情形，所以此处暂不 给出 E(w,b) 是关于 w 和 b 的凸函数的证明，在推导式 (3.11) 时一并给出，” (“pumpkin_book.pdf”, p. 32)

“闭式解是指可以通过具体的表达式解出待解 参数，例如可根据式 (3.7) 直接解得 w。机器学习算法很少有闭式解，线性回归是一个特例” (“pumpkin_book.pdf”, p. 32)

“如果要想用 Python 来实现上式的话，上式中的求和运算只能用循环来实现。但是如果能将上式向量化， 也就是转换成矩阵（即向量）运算的话，我们就可以利用诸如 NumPy 这种专门加速矩阵运算的类库来进 行编写。” (“pumpkin_book.pdf”, p. 33)

“最小二乘法” (“pumpkin_book.pdf”, p. 34)

“矩阵微分公式可查阅 [2]，矩阵微分原理可查阅 [3]” (“pumpkin_book.pdf”, p. 34)

n **元实值函数**：含 n 个自变量，值域为实数域 $\mathbb{R}$ 的函数称为 n 元实值函数，记为 $f(\boldsymbol{x})$，其中 $\boldsymbol{x}=(x_1;x_2;...;x_n)$ 为 n 维向量。“西瓜书” 和本书中的多元函数未加特殊说明均为实值函数。

(“pumpkin_book.pdf”, p. 35)

“西瓜书”和本书中的多元函数未加特殊说明均为实值函数。” (“pumpkin_book.pdf”, p. 35)

\

凸集：设集合 $D\subset\mathbb{R}^n$ 为 n 维欧式空间中的子集，如果对 D 中任意的 n 维向量 $\boldsymbol{x}\in D$ 和 $\boldsymbol{y}\in D$ 与任意的 $\alpha\in[0,1]$，有

$$\alpha\boldsymbol{x}+(1-\alpha)\boldsymbol{y}\in D$$

则称集合 D 是凸集。凸集的几何意义是：若两个点属于此集合，则这两点连线上的任意一点均属于此集合。常见的凸集有空集 $\varnothing$，整个 n 维欧式空间 $\mathbb{R}^n$。

(“pumpkin_book.pdf”, p. 35)

\

凸函数：设 $D\subset\mathbb{R}^n$ 是非空凸集，f 是定义在 D 上的函数，如果对任意的 $\boldsymbol{x}^1,\boldsymbol{x}^2\in D,\alpha\in(0,1)$，均有

$$f\left(\alpha\boldsymbol{x}^1+(1-\alpha)\boldsymbol{x}^2\right)\leqslant\alpha f(\boldsymbol{x}^1)+(1-\alpha)f(\boldsymbol{x}^2)$$

则称 f 为 D 上的凸函数。若其中的 $\leqslant$ 改为 $<$ 也恒成立，则称 f 为 D 上的严格凸函数。

(“pumpkin_book.pdf”, p. 35)

\

梯度：若 n 元函数 $f(\boldsymbol{x})$ 对 $\boldsymbol{x}=(x_1;x_2;...;x_n)$ 中各分量 x_i 的偏导数 $\frac{\partial f(\boldsymbol{x})}{\partial x_i}(i=1,2,...,n)$ 都存在，则称函数 $f(\boldsymbol{x})$ 在 $\boldsymbol{x}$ 处一阶可导，并称以下列向量

$$\nabla f(\boldsymbol{x})=\frac{\partial f(\boldsymbol{x})}{\partial\boldsymbol{x}}=\begin{bmatrix}\frac{\partial f(\boldsymbol{x})}{\partial x_1}\\\frac{\partial f(\boldsymbol{x})}{\partial x_2}\\\vdots\\\frac{\partial f(\boldsymbol{x})}{\partial x_n}\end{bmatrix}$$

为函数 $f(\boldsymbol{x})$ 在 $\boldsymbol{x}$ 处的一阶导数或梯度，易证梯度指向的方向是函数值增大速度最快的方向。$\nabla f(\boldsymbol{x})$ 也可写成行向量形式

$$\nabla f(\boldsymbol{x})=\frac{\partial f(\boldsymbol{x})}{\partial\boldsymbol{x}^{\mathrm{T}}}=\left[\frac{\partial f(\boldsymbol{x})}{\partial x_1},\frac{\partial f(\boldsymbol{x})}{\partial x_2},\cdots,\frac{\partial f(\boldsymbol{x})}{\partial x_n}\right]$$

我们称列向量形式为“分母布局”，行向量形式为“分子布局”，由于在最优化中习惯采用分母布局，因此“西瓜书”以及本书中也采用分母布局。为了便于区分当前采用何种布局，通常在采用分母布局时偏导符号 ∂ 后接的是 $\boldsymbol{x}$，采用分子布局时后接的是 $\boldsymbol{x}^{\mathrm{T}}$。

(“pumpkin_book.pdf”, p. 35)

Hessian 矩阵：若 n 元函数 $f(\boldsymbol{x})$ 对 $\boldsymbol{x}=(x_1;x_2;...;x_n)$ 中各分量 x_i 的二阶偏导数 $\frac{\partial^2 f(\boldsymbol{x})}{\partial x_i \partial x_j}(i=1,2,...,n;j=1,2,...,n)$ 都存在，则称函数 $f(\boldsymbol{x})$ 在 $\boldsymbol{x}$ 处二阶阶可导，并称以下矩阵

$$\nabla^2 f(\boldsymbol{x})=\frac{\partial^2 f(\boldsymbol{x})}{\partial \boldsymbol{x} \partial \boldsymbol{x}^{\mathrm{T}}}=\begin{bmatrix} \frac{\partial^2 f(\boldsymbol{x})}{\partial x_1^2} & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(\boldsymbol{x})}{\partial x_2 \partial x_1} & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_2^2} & \cdots & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\boldsymbol{x})}{\partial x_n \partial x_1} & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_n^2} \end{bmatrix}$$

为函数 $f(\boldsymbol{x})$ 在 $\boldsymbol{x}$ 处的二阶导数或 Hessian 矩阵。若其中的二阶偏导数均连续，则

$$\frac{\partial^2 f(\boldsymbol{x})}{\partial x_i \partial x_j}=\frac{\partial^2 f(\boldsymbol{x})}{\partial x_j \partial x_i}$$

此时 Hessian 矩阵为对称矩阵。

\
("pumpkin_book.pdf", p. 35)

定理 3.1：设 $D \subset \mathbb{R}^n$ 是非空开凸集，$f(\boldsymbol{x})$ 是定义在 D 上的实值函数，且 $f(\boldsymbol{x})$ 在 D 上二阶连续可微，如果 $f(\boldsymbol{x})$ 的 Hessian 矩阵 $\nabla^2 f(\boldsymbol{x})$ 在 D 上是半正定的，则 $f(\boldsymbol{x})$ 是 D 上的凸函数；如果 $\nabla^2 f(\boldsymbol{x})$ 在 D 上是正定的，则 $f(\boldsymbol{x})$ 是 D 上的严格凸函数。

\
("pumpkin_book.pdf", p. 35)

定理 3.2：若 $f(\boldsymbol{x})$ 是凸函数，且 $f(\boldsymbol{x})$ 一阶连续可微，则 $\boldsymbol{x}^*$ 是全局解的充分必要条件是其梯度等于零向量，即 $\nabla f(\boldsymbol{x}^*)=\mathbf{0}$。

\
("pumpkin_book.pdf", p. 35)

"由于 X 是由样本构成的矩阵，而样本是千变万化的，因此无法保证 XTX 一定是正定矩阵，极易出现 非正定的情形。当 XTX 非正定矩阵时，除了"西瓜书"中所说的引入正则化外，也可用 XTX 的伪逆矩阵 代入式 (3.11) 求解出 ˆ w∗，只是此时并不保证求解得到的 ˆ w∗ 一定是全局最优解。除此之外，也可用下一 节将会讲到的"梯度下降法"求解，同样也不保证求得全局最优解。" ("pumpkin_book.pdf", p. 36)

"3.3 对数几率回归" ("pumpkin_book.pdf", p. 36)

"对数几率回归的一般使用流程如下：首先在训练集上学得模型 y= 1 1 + e−(wTx+b) 然后对于新的测试样本 xi，将其代入模型得到预测结果 yi，接着自行设定阈值 θ，通常设为 θ = 0.5，如 果 yi ⩾ θ 则判 xi 为正例，反之判为反例。" ("pumpkin_book.pdf", p. 36)

"3.3.2 梯度下降法" ("pumpkin_book.pdf", p. 37)

"不同于式 (3.7) 可求得闭式解，式 (3.27) 中的 β 没有闭式解，因此需要借助其他工具进行求解。求解使得式 (3.27) 取到最小值的 β 属于最优化中的"无约束优化问题"，在无约束优化问题中最常用的求解算法有"梯度下降法"和"牛顿法"[1] ，" ("pumpkin_book.pdf", p. 37)

“梯度下降法是一种迭代求解算法，其基本思路如下：先在定义域中随机选取一个点 x0，将其代入函 数 f (x) 并判断此时 f (x0) 是否是最小值，如果不是的话，则找下一个点 x1，且保证 f (x1) f (x0)，然 →→ 配套视频教程：https://www.bilibili.com/video/BV1Mh411e7VU ←” (“pumpkin_book.pdf”, p. 37)

“→→ 欢迎去各大电商平台选购纸质版南瓜书《机器学习公式详解 第 2 版》 ←← 后接着判断 f (x1) 是否是最小值，如果不是的话则重复上述步骤继续迭代寻找 x2、x3、...... 直到找到使 得 f (x) 取到最小值的 x∗。” (“pumpkin_book.pdf”, p. 38)

“3.3.3 牛顿法” (“pumpkin_book.pdf”, p. 38)

“同梯度下降法，牛顿法也是一种迭代求解算法，其基本思路和梯度下降法一致，只是在选取第 t + 1 个点 xt+1 时所采用的策略有所不同，即迭代公式不同。梯度下降法每次选取 xt+1 时，只要求通过泰勒公式在 xt 的邻域内找到一个函数值比其更小的点即可，而牛顿法则期望在此基础之上，xt+1 还必须是 xt 的邻域内的极小值点。” (“pumpkin_book.pdf”, p. 38)

“3.4 线性判别分析” (“pumpkin_book.pdf”, p. 40)

“线性判别分析的一般使用流程如下：首先在训练集上学得模型 y = wTx” (“pumpkin_book.pdf”, p. 40)

“由向量内积的几何意义可知，y 可以看作是 x 在 w 上的投影，因此在训练集上学得的模型能够保证训练 集中的同类样本在 w 上的投影 y 很相近，而异类样本在 w 上的投影 y 很疏远。然后对于新的测试样本 xi，将其代入模型得到它在 w 上的投影 yi，然后判别这个投影 yi 与哪一类投影更近，则将其判为该类。 最后，线性判别分析也是一种降维方法，但不同于第 10 章介绍的无监督降维方法，线性判别分析是 一种监督降维方法，即降维过程中需要用到样本类别标记信息。” (“pumpkin_book.pdf”, p. 41)

“3.5 多分类学习” (“pumpkin_book.pdf”, p. 44)

““海明距离”是指两个码对应位置不相同的个数，” (“pumpkin_book.pdf”, p. 44)

“欧式距离”则是指两个向量之间 的欧氏距离，” (“pumpkin_book.pdf”, p. 44)

“3.6 类别不平衡问题” (“pumpkin_book.pdf”, p. 44)

“对于类别不平衡问题，“西瓜书”2.3.1 节中的“精度”通常无法满足该特殊任务的需求，例如“西瓜 书”在本节第一段的举例：有 998 个反例和 2 个正例，若机器学习算法返回一个永远将新样本预测为反 例的学习器则能达到 99.8% 的精度，显然虚高，因此在类别不平衡时常采用 2.3 节中的查准率、查全率和 F1 来度量学习器的性能。” (“pumpkin_book.pdf”, p. 44)