机器学习导论
习题三

171860607, 白晋斌, 810594956@qq.com

2020 年 4 月 23 日

学术诚信

本课程非常重视学术诚信规范，助教老师和助教同学将不遗余力地维护作业中的学术诚信规范的建立。希望所有选课学生能够对此予以重视。[1]

(1) 允许同学之间的相互讨论，但是**署你名字的工作必须由你完成**，不允许直接照搬任何已有的材料，必须独立完成作业的书写过程;

(2) 在完成作业过程中，对他人工作（出版物、互联网资料）中文本的直接照搬（包括原文的直接复制粘贴及语句的简单修改等）都将视为剽窃，剽窃者成绩将被取消。**对于完成作业中有关键作用的公开资料，应予以明显引用**;

(3) 如果发现作业之间高度相似将被判定为互相抄袭行为，**抄袭和被抄袭双方的成绩都将被取消**。因此请主动防止自己的作业被他人抄袭。

作业提交注意事项

(1) 请在 LaTeX 模板中第一页填写个人的学号、姓名、邮箱;

(2) 本次作业需提交该 pdf 文件、问题 4 可直接运行的源码 (.py 文件)、问题 4 的预测结果 (.csv 文件)，将以上三个文件压缩成 zip 文件后上传。注意：pdf、预测结果命名为“学号 _ 姓名”(例如“181221001_ 张三.pdf”)，源码、压缩文件命名为“学号”，例如“181221001.zip”;

(3) 未按照要求提交作业，提交作业格式不正确，**作业命名不规范**，将会被扣除部分作业分数;

(4) 本次作业提交截止时间为**4 月 23 日 23:55:00**。除非有特殊情况（如因病缓交），否则截止时间后不接收作业，本次作业记零分。

[1] 参考尹一通老师高级算法课程中对学术诚信的说明。

1 [20pts] Decision Tree I

(1) **[5pts]** 试分析使用“最小训练误差”作为决策树划分选择的缺陷。

(2) **[5pts]** 树也是一种线性模型，考虑图 (1) 所示回归决策树，X_1, X_2 均在单位区间上取值，t_1, t_2, t_3, t_4 满足 $0 < t_1 < t_3 < 1, 0 < t_2, t_4 < 1$，试绘制出该决策树对于特征空间的划分。假设区域 R_i 上模型的输出值为 c_i，试用线性模型表示该决策树。

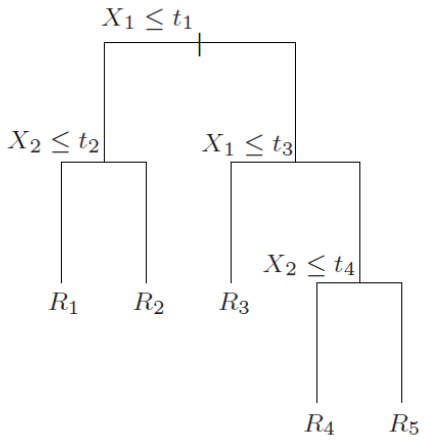

图 1: 回归决策树

(3) **[10pts]** 对于回归树，我们常采用平方误差来表示回归树对于训练数据的预测误差。但是找出平方误差最小化准则下的最优回归树在计算上一般是不可行的，通常我们采用贪心的算法计算切分变量 j 和分离点 s。CART 回归树在每一步求解如下优化问题

$$\min_{j,s}\left[\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_2(j,s)}(y_i-c_2)^2\right]$$

其中 $R_1(j,s)=\{\boldsymbol{x}|x_j\le s\}, R_2(j,s)=\{\boldsymbol{x}|x_j>s\}$。试分析该优化问题表达的含义并给出变量 j, s 的求解思路。

Solution. 此处用于写解答 (中英文均可)

(1) 训练误差是指模型在训练集上的错分样本比例, 使用“最小训练误差”作为决策树划分的选择, 过度关注训练样本的特性, 可能会导致决策树分支过多, 把训练集自身的一些特点当作所有数据都具有的一般性质而导致过拟合, 降低模型泛化能力.

(2) 不妨设 $t_2 < t_4$, 决策树在特征空间的划分如图 (2) 所示.
用线性模型表示该决策树:

$$\begin{aligned}c_i =&\mathbb{I}(X_1-t_1\le 0)\left(\mathbb{I}(X_2-t_2\le 0)c_1+\mathbb{I}(X_2-t_2>0)c_2\right)\\&+\mathbb{I}(X_1-t_3\le 0)\mathbb{I}(X_1-t_2>0)c_3\\&+\mathbb{I}(X_1-t_3>0)(\mathbb{I}(X_2-t_4\le 0)c_4+\mathbb{I}(X_2-t_4>0)c_5)\end{aligned}$$

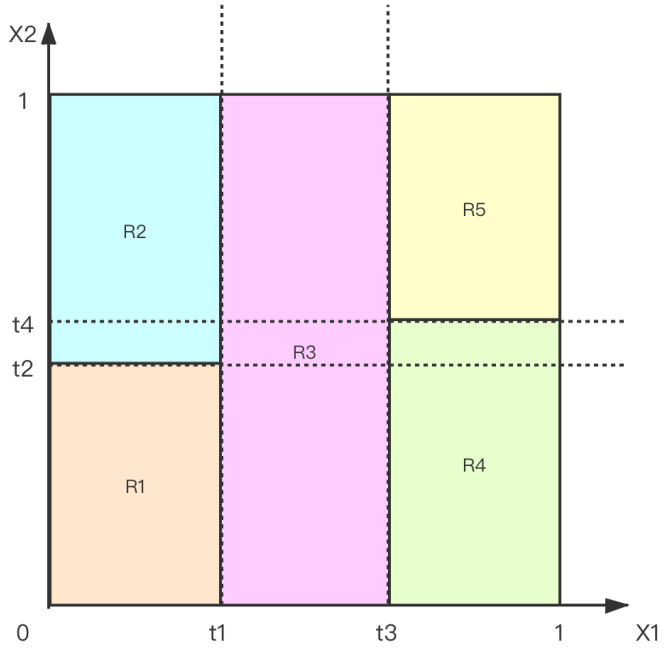

图 2: 决策树在特征空间的划分

亦可被简化为

$$f(x) = \sum_{i=1}^{5} c_i \mathbb{I}(x \in R_i)$$

其中,$\mathbb{I}(\cdot)$ 为指示函数.

(3) 对于回归树模型 $f(x) = \sum_{i=1}^{m} c_i \mathbb{I}(x \in R_i)$, 数据空间被划分为 R_1 R_m 共 m 个空间, 每个空间有固定的输出值 c_m, 因此模型输出与实际误差为 $\sum_{x_i \in R_i} (y_i - f(x_i))^2$, 当 c_m 为相应空间上的所有实际值的均值时, 该平方误差可以达到最优.

当我们选择变量 j 为切分变量, x_j 的取值 s 作为切分点, 则可以得到两个区域:$R_1(j,s) = \{\boldsymbol{x}|x_j \leq s\}, R_2(j,s) = \{\boldsymbol{x}|x_j > s\}$. 当 j 和 s 固定时, 我们要找到两个区域的代表值 c_1,c_2 使得各自区间上的平方差最小, 即得到

$$\min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2 \tag{1}$$

前面已经知道 c_1,c_2 为相应空间上的所有实际值的均值时, 式(1)可以达到最优.

该优化问题想要表达的是求解一组切分变量 j 和分离点 s, 按照该组 (j,s) 对空间进行切分, 可以使切分后的两个空间对应的平方误差最小化.

关于求解变量 (j,s), 我们只需要遍历变量 j, 对固定的切分变量 j 扫描切分点 s, 选择使式(1)达到最小的 (j,s) 对. 即分别计算切分后两个空间的平方误差和, 选择最小的平方误差和对应的切分变量和分离点, 生成两个子空间. 对两个子空间分别递归计算其对应的切分变量 j 和分离点 s, 直至满足递归停止的条件.

其中, 递归停止的条件包括

1. 节点中的样本个数少于预定阈值.
2. 样本集中的 Gini 系数小于预定阈值 (样本种类趋于相同)
3. 没有更多特征.

2 [25pts] Decision Tree II

(1) **[5pts]** 对于不含冲突数据（即特征向量相同但标记不同）的训练集，必存在与训练集一致（即训练误差为 0）的决策树。如果训练集可以包含无穷多个数据，是否一定存在与训练集一致的深度有限的决策树？证明你的结论。（仅考虑单个划分准则仅包含一次属性判断的决策树）

(2) **[5pts]** 考虑如表1所示的人造数据，其中“性别”、“喜欢 ML 作业”是属性，“ML 成绩高”是标签。请画出使用信息增益为划分准则的决策树算法所有可能的结果。（需说明详细计算过程）

表 1: 训练集

编号	性别	喜欢 ML 作业	ML 成绩高
1	男	是	是
2	女	是	是
3	男	否	否
4	男	否	否
5	女	否	是

(3) **[10pts]** 考虑如表2所示的验证集，对上一小问的结果基于该验证集进行预剪枝、后剪枝，剪枝结果是什么？（需给出详细计算过程）

表 2: 验证集

编号	性别	喜欢 ML 作业	ML 成绩高
6	男	是	是
7	女	是	否
8	男	否	否
9	女	否	否

(4) **[5pts]** 比较预剪枝、后剪枝的结果，每种剪枝方法在训练集、验证集上的准确率分别为多少？哪种方法拟合能力较强？

Solution. 此处用于写解答 (中英文均可)

(1) 存在与训练集一致的深度有限的决策树. 证明如下:

不妨设不存在与训练集一致的深度有限的决策树, 即不存在这样一个决策树可以完美划分所有训练集, 即任意决策树总存在某一节点 N, 该节点上有多条训练集的数据无法进行划

分, 即节点 N 有多条数据的特征向量相同但标记不同, 这与已知训练集不含冲突数据相矛盾, 故假设不成立, 存在与训练集一致的深度有限的决策树.

(2) 计算根节点的信息熵

$$Ent(D) = -\sum_{k=1}^{|y|} p_k \log_2 p_k = -(\frac{2}{5}\log_2\frac{2}{5} + \frac{3}{5}\log_2\frac{3}{5}) = 1.12$$

计算当前属性集合 {性别, 喜欢 ML 作业} 中每个属性的信息增益.
先计算属性“性别”, 他有两个不同的取值 {男, 女}, 若使用该属性对 D 进行划分, 则可得到 2 个子集, 分别记为 D^1, D^2, 分别计算其对应的信息熵

$$Ent(D^1) = -(\frac{1}{3}\log_2\frac{1}{3} + \frac{2}{3}\log_2\frac{2}{3}) = 1.35$$

$$Ent(D^2) = -(\frac{2}{2}\log_2\frac{2}{2} + \frac{0}{2}\log_2\frac{0}{2}) = 0$$

于是, 属性“性别”的信息增益为

$$Gain(D, sex) = Ent(D) - \sum_{v=1}^{2}\frac{|D^v|}{|D|}Ent(D^v) = 0.31$$

再计算属性“喜欢 ML 作业”, 他有两个不同的取值 {是, 否}, 若使用该属性对 D 进行划分, 则可得到 2 个子集, 分别记为 D^1, D^2, 分别计算其对应的信息熵

$$Ent(D^1) = -(\frac{2}{2}\log_2\frac{2}{2} + \frac{0}{2}\log_2\frac{0}{2}) = 0$$

$$Ent(D^2) = -(\frac{1}{3}\log_2\frac{1}{3} + \frac{2}{3}\log_2\frac{2}{3}) = 1.35$$

于是, 属性“喜欢 ML 作业”的信息增益为

$$Gain(D, lovingMLhw) = Ent(D) - \sum_{v=1}^{2}\frac{|D^v|}{|D|}Ent(D^v) = 0.31$$

属性“性别”,“喜欢 ML 作业”均取得了最大的信息增益, 因此, 可任选其一作为划分属性, 对每个分支结点再用另一属性作为划分属性即可得到对应的决策树. 所有可能的决策树如图 (3) 所示.

(3) 预剪枝:

对于决策树 1: 在划分前, 所有样例集中在根结点, 若不进行划分, 则该结点被标记为叶结点, 该结点的标签为“是”(ML 成绩高), 用表2的验证集做评估, 验证集精度为 $\frac{1}{4} \times 100\% = 25\%$, 使用“性别”进行划分后,“性别”=“男”得到的根结点包含 1,3,4, 该结点的标签为“否”(ML 成绩高),“性别”=“女”得到的根结点包含 2,5, 该结点的标签为“是”(ML 成绩高), 用表2的验证集做评估, 验证集精度为 $\frac{1}{4} \times 100\% = 25\%$, 验证集精度未上升, 于是, 预剪枝策略将禁止根结点被划分.

对于决策树 2: 在划分前, 所有样例集中在根结点, 若不进行划分, 则该结点被标记为叶结点, 该结点的标签为“是”(ML 成绩高), 用表2的验证集做评估, 验证集精度为 $\frac{1}{4} \times 100\% = 25\%$, 使用“喜欢 ML 作业”进行划分后,“喜欢 ML 作业”=“是”得到的根结点包含 1,2, 该结点的标

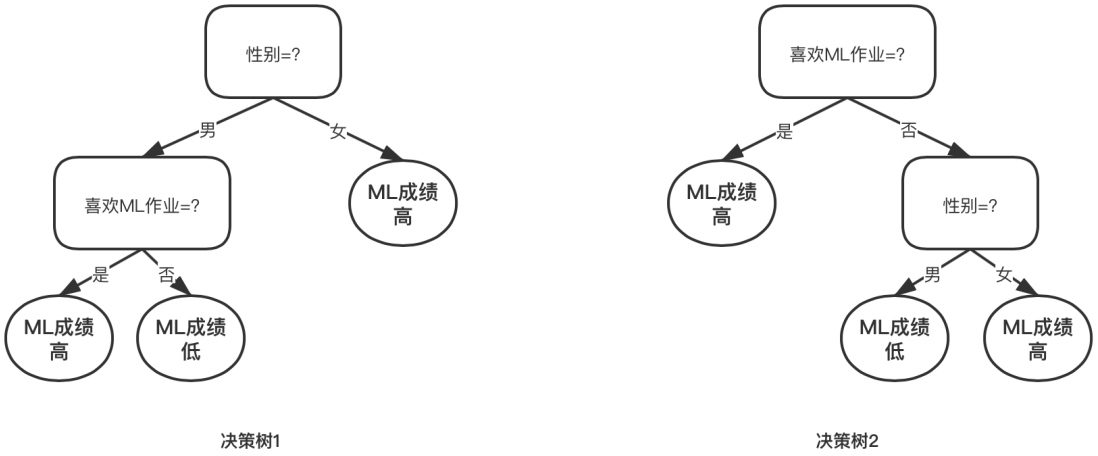

图 3: 所有可能的决策树

签为“是”(ML 成绩高),“喜欢 ML 作业”=“否”得到的根结点包含 3,4,5, 该结点的标签为“否”(ML 成绩高), 用表2的验证集做评估, 验证集精度为 $\frac{3}{4} \times 100\% = 75\%$, 验证集精度上升, 于是, 用属性“喜欢 ML 作业”进行划分得以确定.

接下来考察第二行左边的分支结点, 若不进行划分, 则该结点被标记为叶结点, 该结点的标签为“是”(ML 成绩高), 用表2的验证集做评估, 验证集精度为 $\frac{3}{4} \times 100\% = 75\%$, 使用“性别”进行划分后, 验证集精度为 $\frac{3}{4} \times 100\% = 75\%$, 验证集精度未上升, 于是, 第二行左边的分支结点将禁止被划分.

接下来考察第二行右边的分支结点, 若不进行划分, 则该结点被标记为叶结点, 该结点的标签为“否”(ML 成绩高), 用表2的验证集做评估, 验证集精度为 $\frac{3}{4} \times 100\% = 75\%$, 使用“性别”进行划分后, 验证集精度为 $\frac{2}{4} \times 100\% = 50\%$, 验证集精度下降, 于是, 第二行右边的分支结点将禁止被划分.

预剪枝策略得到的决策树如图 (4) 所示.

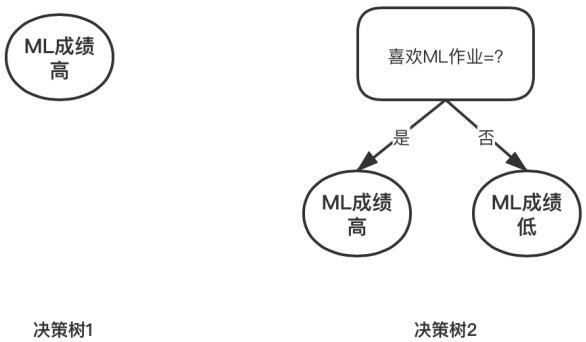

图 4: 预剪枝策略下所有可能的决策树

后剪枝:

对于决策树 1: 验证集精度为 $\frac{1}{4} \times 100\% = 50\%$, 首先考察第二行左边的分支结点, 若将其替换为叶结点, 该叶结点包含训练集 1,3,4, 该结点的标签为“否”(ML 成绩高), 此时决策树的验证集精度为 $\frac{1}{4} \times 100\% = 25\%$, 验证集精度下降, 于是, 第二行左边的分支结点将被保留.
接下来考察第二行右边的分支结点, 若将其替换为叶结点, 该叶结点包含训练集 2,5, 该结点的标签为“是”(ML 成绩高), 此时决策树的验证集精度为 $\frac{2}{4} \times 100\% = 50\%$, 验证集精度不变, 根据奥卡姆剃刀准则, 第二行右边的分支结点将被剪枝.

对于决策树 2: 验证集精度为 $\frac{1}{4} \times 100\% = 50\%$, 首先考察第二行左边的分支结点, 若将其替换为叶结点, 该叶结点包含训练集 1,2, 该结点的标签为“是”(ML 成绩高), 此时决策树的验证集精度为 $\frac{2}{4} \times 100\% = 50\%$, 验证集精度不变, 根据奥卡姆剃刀准则, 剪枝后的模型更好, 于是, 第二行左边的分支结点将被剪枝.
接下来考察第二行右边的分支结点, 若将其替换为叶结点, 该叶结点包含训练集 3,4,5, 该结点的标签为“否”(ML 成绩高), 此时决策树的验证集精度为 $\frac{2}{4} \times 100\% = 75\%$, 验证集精度上升, 于是, 第二行右边的分支结点将被剪枝.
接下来考察根结点, 若将其替换为叶结点, 该叶结点包含训练集 1,2,3,4,5, 该结点的标签为“是”(ML 成绩高), 此时决策树的验证集精度为 $\frac{1}{4} \times 100\% = 25\%$, 验证集精度下降, 于是, 根结点将被保留.

后剪枝策略得到的决策树如图 (5) 所示.

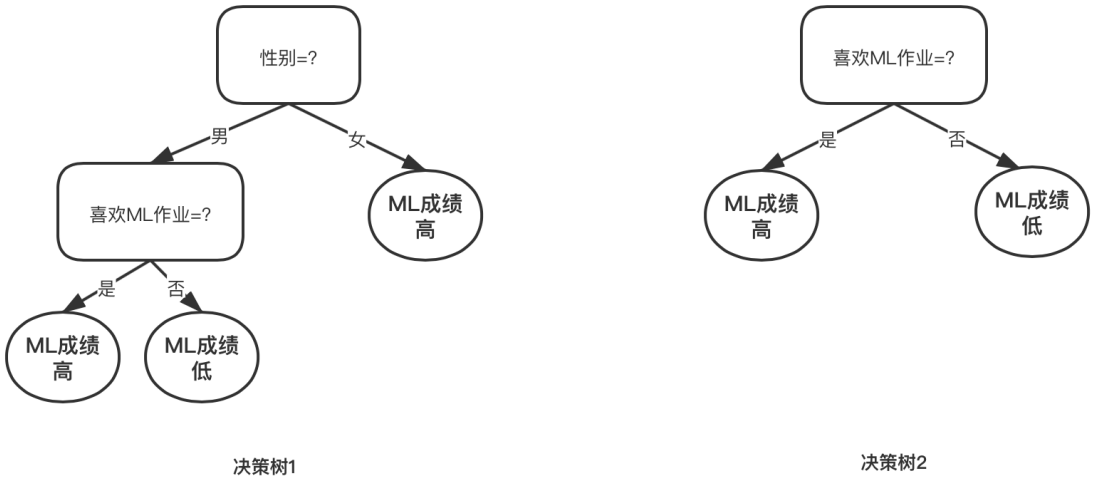

图 5: 后剪枝策略下所有可能的决策树

(4) 预剪枝: 对于决策树 1, 在训练集的准确率为 60%, 在测试集的准确率为 25%, 对于决策树 2, 在训练集的准确率为 80%, 在测试集的准确率为 75%.

后剪枝: 对于决策树 1, 在训练集的准确率为 100%, 在测试集的准确率为 50%, 对于决策树 2, 在训练集的准确率为 80%, 在测试集的准确率为 75%.

对比图 (4) 和图 (5) 可以看出, 后剪枝决策树通常比预剪枝决策树保留了更多的分支, 故而欠拟合风险更小, 泛化性能优于预剪枝决策树. 即后剪枝拟合能力更强.

3 [25pts] SVM with Weighted Penalty

考虑标准的 SVM 优化问题如下 (即课本公式 (6.35)),

$$\begin{aligned} \min_{\mathbf{w},b,\xi_i} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i \\ \text{s.t.} \quad & y_i(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0, i = 1, 2, \cdots, m. \end{aligned} \tag{2}$$

注意到，在(2)中，对于正例和负例，其在目标函数中分类错误或分对但置信度较低的“惩罚”是相同的。在实际场景中，很多时候正例和负例分错或分对但置信度较低的“惩罚”往往是不同的，比如癌症诊断等。

现在，我们希望对负例分类错误 (即 false positive) 或分对但置信度较低的样本施加 $k > 0$ 倍于正例中被分错的或者分对但置信度较低的样本的“惩罚”。对于此类场景下，

(1) **[10pts]** 请给出相应的 SVM 优化问题。

(2) **[15pts]** 请给出相应的对偶问题及 KKT 条件，要求详细的推导步骤。

Solution. 此处用于写解答 (中英文均可)

(1) SVM 优化问题:

$$\begin{aligned} \min_{\mathbf{w},b,\xi_i} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\left[\mathbb{I}\left(y_i > 0\right)\xi_i + \mathbb{I}\left(y_i < 0\right)k\xi_i\right] \\ \text{s.t.} \quad & y_i(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0, i = 1, 2, \cdots, m. \end{aligned} \tag{3}$$

其中,$\mathbb{I}(\cdot)$ 为指示函数.

(2) 对偶问题: 对式(3)使用拉格朗日乘子法可得到其“对偶问题”. 具体来说, 对式(3)的每条约束添加拉格朗日乘子 $\alpha_i \geq 0, \mu_i \geq 0$, 则该问题的拉格朗日函数可写为

$$\begin{aligned} L(\mathbf{w}, b, \alpha, \xi, \mu) =& \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\left[\mathbb{I}\left(y_i > 0\right)\xi_i + \mathbb{I}\left(y_i < 0\right)k\xi_i\right] \\ & + \sum_{i=1}^{m}\alpha_i(1 - \xi_i - y_i(\mathbf{w}^T\mathbf{x}_i + b)) - \sum_{i=1}^{m}\mu_i\xi_i \end{aligned} \tag{4}$$

其中,$\alpha = (\alpha_1; \alpha_2; \cdots; \alpha_m)$. 令 $L(\mathbf{w}, b, \alpha, \xi, \mu)$ 对 $\mathbf{w}, b, \xi$ 的偏导为 0 可得

$$\mathbf{w} = \sum_{i=1}^{m}\alpha_i y_i \mathbf{x}_i$$

$$0 = \sum_{i=1}^{m}\alpha_i y_i$$

$$C\Big[\mathbb{I}\left(y_i>0\right)+\mathbb{I}\left(y_i<0\right)k\Big]=\alpha_i+\mu_i$$

代入可得式(3)的对偶问题

$$\begin{aligned}\max_{\alpha}\quad & \sum_{i=1}^{m}\alpha_i-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j\\ \text{s.t.}\quad & \sum_{i=1}^{m}\alpha_i y_i=0, i=1,2,\cdots,m\\ & 0\le\alpha_i\le C\Big[\mathbb{I}\left(y_i>0\right)+\mathbb{I}\left(y_i<0\right)k\Big], i=1,2,\cdots,m\end{aligned}\tag{5}$$

解出 α_i,μ_i 后, 求出 $\mathbf{w},b$ 即可得到模型

$$f(x)=\mathbf{w}^T\mathbf{x}+b$$

KTT 条件: 从对偶问题(5)解出的 α_i 是式(4)中的拉格朗日乘子, 它恰好对应着训练样本 (x_i,y_i), 注意到式(3)中有不等式约束, 因此上述过程需满足 KKT 条件, 即要求

$$f(x)=\begin{cases}\alpha_i\ge 0,\mu_i\ge 0\\ y_i f(\mathbf{x}_i)-1+\xi_i\ge 0\\ \alpha_i(y_i f(\mathbf{x}_i)-1+\xi_i)=0\\ \xi_i\ge 0,\mu_i\xi_i=0\end{cases}$$

4 [30 pts] 编程题, Linear SVM

请结合编程题指南进行理解

SVM 转化成的对偶问题实际是一个二次规划问题，除了 SMO 算法外，传统二次规划方法也可以用于求解对偶问题。求得最优拉格朗日乘子后，超平面参数 $\mathbf{w},\mathbf{b}$ 可由以下式子得到：

$$\mathbf{w}=\sum_{i=1}^{m}\alpha_i y_i x_i\tag{6}$$

$$\mathbf{b}=\frac{1}{|S|}\sum_{s\in S}(y_s-\sum_{i\in S}\alpha_i y_i x_i^T x_s)\tag{7}$$

请完成以下任务：

(1) **[5pts]** 使用 QP 方法求解训练集上的 SVM 分类对偶问题 (不考虑软间隔情况)。

(2) **[10 pts]** 手动实现 SMO 算法求解上述对偶问题。

(3) **[15 pts]** 对测试数据进行预测，确保预测结果尽可能准确。

Solution. 此处用于写解答 (中英文均可)

(1) 代码位于.py 文件的 qp() 函数. 输入为训练集的特征、训练集标签, 输出为训练集上的 SVM 分类对偶问题的解. 如有需要, 可直接运行 main 函数中对应注释部分代码即可.

(2) 代码位于.py 文件的 smo() 函数. 输入为训练集的特征、训练集标签, 输出为训练集上的 SVM 分类对偶问题的解. 如有需要, 可直接运行 main 函数中对应注释部分代码即可.

(3) 采用留出法, 将数据集按照训练集: 验证集 = 7:3 的比例进行划分, 为了避免切分之后的数据集在特征分布上不够均匀, 我们先将数据打乱, 使数据随机排序, 然后再进行切分. 本小问中主要新增预测函数 predict(), 并组织了从文件读取, 到模型搭建, 再到模型预测, 最后模型评估整个过程的函数, 这些函数已被封装好放于 main() 函数中.

此外, 我们对前两小问的函数进行了扩展, 使其支持线性核、多项式核、高斯核, 并使用不同的核对模型进行训练.

我们还额外考虑了软间隔情况, 新增参数 C, 通过滑动 C 进行测试来得到一个较好的参数.

最终, 经过对参数的遍历测试, 我们发现,SMO 算法性能要略优于 QP 算法, 故以 SMO 算法为代表, 输出测试集标签, 预测结果已被保存为.csv 文件.

相关入口与函数均整理至 main() 函数中, 如有需要, 将代码与数据集置于同一文件夹下, 即可运行.