机器学习导论
习题三

171830635, 俞星凯, yuxk@smail.nju.edu.cn

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。

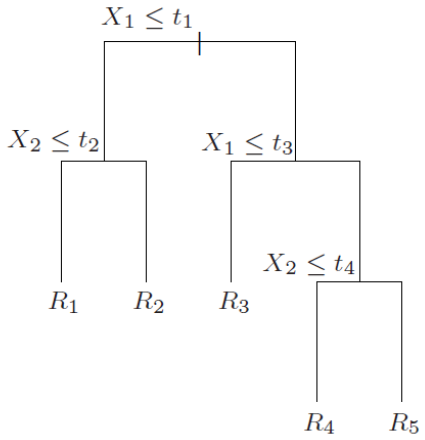

图 1: 回归决策树

(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.

(1) 使用“最小训练误差”作为划分选择，为了尽可能正确分类训练样本，结点划分过程将不断重复，有时会造成决策树分支过多，导致过拟合。

(2)

$$y=\begin{cases} R1, & 0\le X_1\le t_1, 0\le X_2\le t_2 \\ R2, & 0\le X_1\le t_1, t_2< X_2\le 1 \\ R3, & t_1< X_1\le t_3 \\ R4, & t_3< X_1\le 1, 0\le X_2\le t_4 \\ R5, & t_3< X_1\le 1, t_4< X_2\le 1 \end{cases}$$

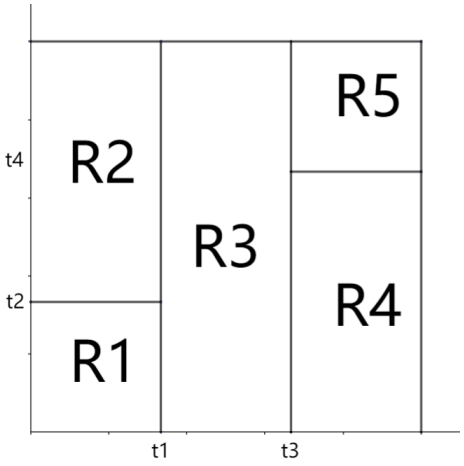

图 2: 特征空间划分

(3) 选择第 j 个维度和分割点 s，根据数据在第 j 个维度上的取值 x_j 与 s 的大小关系，将数据集分成两部分 R_1 和 R_2。对 R_k 中的数据预测 $c_k(k=1,2)$，这样 R_k 中数据的平方误差和 $\sum_{x_i \in R_k(j,s)} (y_i - c_1)^2$，从而我们优化问题

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

容易得到 c_k 是 R_k 中数据标签 y_i 的均值

$$c_k = ave(y_i | x_i \in R_k), \quad k = 1, 2$$

j, s 的优化可以通过下面方法得到

(a) 对于固定的 j，将数据集上不同的 x_j 依次作为切分点 s，计算代价函数，选择使得代价函数最小的 x_j^* 作为切分点 s。

(b) 遍历所有的维度，分别调用上面步骤，选择使得代价函数最小的 j 及其对应的 s。

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）

(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 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

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.

(1) 不一定存在。考虑这样一个数据集：输入 $x \in \mathbb{N}$，输出 $y \in 0, 1$，当 x 是奇数时，$y = 1$，当 x 是偶数时，$y = 0$。对于这个数据集，不存在与包含无穷多个数据的训练集一致的深度有限的决策树。

(2)

$$Gain(D, \text{性别}) = -(\frac{3}{5}\log_2\frac{3}{5}+\frac{2}{5}\log_2\frac{2}{5})+[\frac{3}{5}(\frac{1}{3}\log_2\frac{1}{3}+\frac{2}{3}\log_2\frac{2}{3})+\frac{2}{5}(\frac{1}{2}\log_2\frac{1}{2}+\frac{1}{2}\log_2\frac{1}{2})] = 0.02$$

$$Gain(D, \text{喜欢 ML 作业}) = -(\frac{3}{5}\log_2\frac{3}{5}+\frac{2}{5}\log_2\frac{2}{5})+[\frac{2}{5}(\frac{2}{2}\log_2\frac{2}{2}+\frac{0}{2}\log_2\frac{0}{2})+\frac{3}{5}(\frac{1}{3}\log_2\frac{1}{3}+\frac{2}{3}\log_2\frac{2}{3})] = 0.42$$

因为属性“喜欢 ML 作业”的信息增益大，所以根据喜欢 ML 作业，将数据集 D 划分为 D^1(喜欢) 和 D^2(不喜欢)。D_1 中的样本属于同一类别，无需划分。

$$Gain(D^2, \text{性别}) = -(\frac{1}{3}\log_2\frac{1}{3}+\frac{2}{3}\log_2\frac{2}{3})+[\frac{2}{3}(\frac{0}{2}\log_2\frac{0}{2}+\frac{2}{2}\log_2\frac{2}{2})+\frac{1}{3}(\frac{1}{1}\log_2\frac{1}{1}+\frac{0}{1}\log_2\frac{0}{1})] = 0.92$$

根据性别，将 D^2 划分为 D^3(男生, 不喜欢) 和 D^4(女生, 不喜欢)。
预测 D^1 和 D^4 中的学生成绩高，D^3 中的学生成绩不高。

(3) 预剪枝过程如下：
对 D^1 基于信息增益选择属性“喜欢 ML 作业划分”，划分前训练集中 ML 成绩高的样例多，因此预测成绩高，这在验证集中的精度为 $\frac{1}{4} = 25\%$。划分后对 D^1 预测成绩高，对 D^2 预测成绩不高，验证集精度为 $\frac{3}{4} = 75\%$，所以需要划分。
对 D^2 选择属性“性别”，划分前验证集精度 75%，划分后对 D^3 预测成绩不高，对 D^4 预

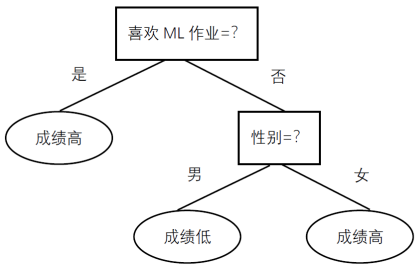

图 3: 决策树

测成绩高，验证集精度为 $\frac{1}{4} = 25\%$，因此禁止划分。

后剪枝过程如下：

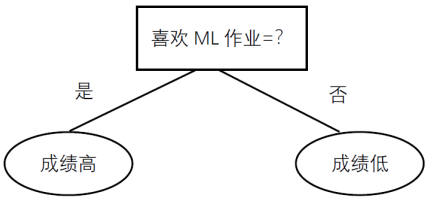

图 4: 预剪枝结果

生成完整决策树，验证集精度为 $\frac{1}{4} = 25\%$。若将性别决策移除，则对于不喜欢 ML 作业的样例预测成绩不高，验证集精度 $\frac{3}{4} = 75\%$，于是决定剪枝。

再考虑将喜欢 ML 作业决策移除，则对于所有样例预测 ML 成绩高，验证集精度 $\frac{1}{4} = 25\%$，决定不剪枝。

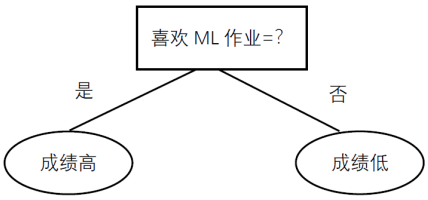

图 5: 后剪枝结果

(4) 预剪枝的训练集精度 80%，验证集精度 75%，后剪枝的训练集精度 80%，验证集精度 75%。在这个例子中拟合能力一样，但是一般来说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))，

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

注意到，在(1)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) 修改 SVM 优化问题

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

(2) 引入拉格朗日乘子得到拉格朗日函数

$$L(w, b, \alpha, \xi, \mu) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}(\frac{k+1}{2} - \frac{k-1}{2}y_i)\xi_i + \sum_{i=1}^{m}\alpha_i(1 - \xi_i - y_i(w^T x + b)) - \sum_{i=1}^{m}\mu_i\xi_i$$

令 $L(w, b, \alpha, \xi, \mu)$ 对 w, b, ξ 的偏导为零可得

$$\begin{aligned} w &= \sum_{i=1}^{m}\alpha_i y_i x_i, \\ 0 &= \sum_{i=1}^{m}y_i x_i, \\ C &= (\alpha_i + \mu_i)/(\frac{k+1}{2} - \frac{k-1}{2}y_i) \end{aligned}$$

代入原问题得到对偶问题

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j x_i^T x_j \\ \text{s.t.} \quad & \sum_{i=1}^{m}\alpha_i y_i = 0, \\ & 0 \leq \alpha_i \leq C(\frac{k+1}{2} - \frac{k-1}{2}y_i), \quad i = 1, 2, \cdots, m. \end{aligned}$$

KKT 条件要求

$$\begin{cases} \alpha_i \geq 0, \quad \mu_i \geq 0, \quad \xi_i \geq 0 \\ y_i f(x_i) - 1 + \xi_i \geq 0 \\ \alpha_i (y_i f(x_i) - 1 + \xi_i) = 0 \\ \mu_i \xi_i = 0 \end{cases}$$

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：

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i x_i \tag{2}$$

$$\mathbf{b} = \frac{1}{|S|} \sum_{s \in S} (y_s - \sum_{i \in S} \alpha_i y_i x_i^T x_s) \tag{3}$$

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.

(1) 使用 python 的 cvxopt 库求解 QP 问题，核心求解过程如下：

```
P = matrix((y @ y.T) * (X @ X.T) + np.diag(np.full(m, 1e-7)))
q = matrix(-np.ones(m))
G = matrix(-np.eye(m))
h = matrix(np.zeros(m))
A = matrix(y.T)
b = matrix(0.0)
solvers.options['show_progress'] = False
sol = solvers.qp(P, q, G, h, A, b)
```

于是可以得到对偶变量 α，进而求出 w 和 b：

```
alpha = np.array(sol['x'])
w = X.T @ (alpha * y)
indice = np.ravel(alpha > 1)
b = np.mean(y[indice] - X[indice] @ w)
```

cvxopt 求出的解在训练集上精度为 81%。

(2) 首先计算距离每个样本最远的异类样本，这里充分利用了数组广播：

```
D = np.sum(np.square(X[np.newaxis,:,:] - X[:,np.newaxis,:]), -1)
D = y @ y.T * D
D = np.argmin(D, 1)
```

然后迭代优化，先计算 KKT 违背程度：

```
w = X.T @ (alpha * y)
indice = np.ravel(alpha > 0.01)
b = np.mean(y[indice] - X[indice] @ w)
err = np.ravel(y * (X @ w + b) - 1)
err[indice] **= 2
err[~indice & (err >= 0)] = 0
err[~indice & (err < 0)] **= 2
```

再确定需要优化的 α_i 和 α_j，根据自己推导的公式更新：

```
i = np.argmax(err)
j = D[i]
c = alpha[i] * y[i] + alpha[j] * y[j] - alpha.T @ y
p = np.sum(alpha * y * X @ X[i].T)
q = np.sum(alpha * y * X @ X[j].T)
r = X[i] @ X[j].T
t = 1 - y[i] * y[j] - y[i] * p + y[i] * q + y[i] * c * r
t /= 2 * r
alpha[i] = max([0, t])
alpha[j] = c * y[j] - y[i] * y[j] * alpha[i]
```

最后得到的结果在训练集上精度为 80% 到 82%。

(3) 尝试使用核技巧，修改训练代码，先求核矩阵：

```
D = np.exp(np.sum(np.square(X[np.newaxis,:,:] - X[:,np.newaxis,:]), -1) \
    / (-2 * sigma ** 2))
P = matrix(y @ y.T * D + np.diag(np.full(m, 1e-7)))
```

同样也要修改测试代码，发现核技巧结果不好。

尝试对某些特征取倒数，最好结果是 82.5%。

```
d = {}
for i in range(1, 6):
  for j in combinations(range(5), i):
    X_temp = X_train.copy()
    X_temp[:, j] = 1 / X_temp[:, j]
    w, b = QP(X_temp, y_train)
    y_pred = np.sign(X_temp @ w + b)
    d[j] = np.mean(y_pred == y_train)
print("the accuracy using inverse of feature:", max(d.items(), key=lambda x: x[1]))
```

类似尝试对某些特征取平方和指数，结果都在 81% 左右。