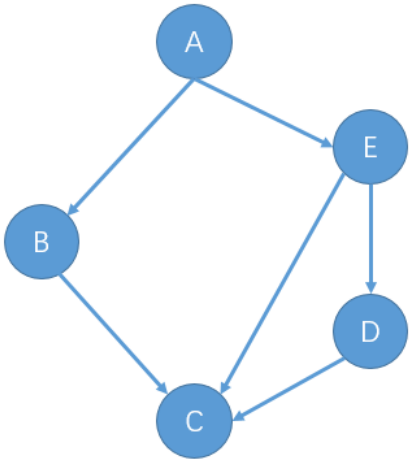
机器学习导论
作业五

151220097，孙旭东，248381185@qq.com

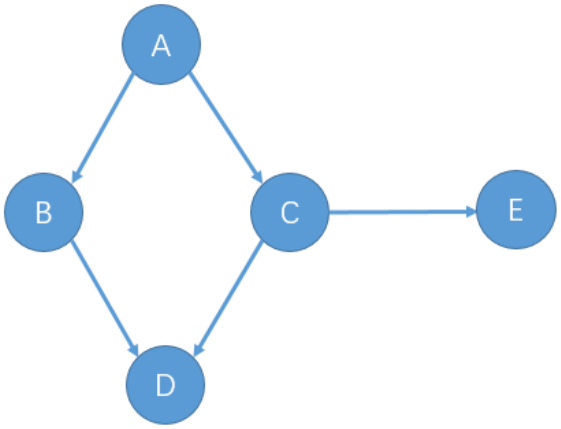
2018 年 6 月 19 日

1 [30pts] Conditional Independence in Bayesian Network

(1) **[5pts]** 请给出图中贝叶斯网结构的联合概率分布的分解表达式。



(2) **[5pts]** 请给出下图中按照道德化方法可以找到的所有条件独立的组合(即哪些变量关于哪些变量或者变量集条件独立)，独立也算做条件独立的一种特例。



(3) **[10pts]** 在这里，首先我们将给出关于“阻塞”的概念，然后我们根据“阻塞”的概念给出条件独立的充要条件。（大家也可以参考这个网站）

> **定义 1** (阻塞). 设X, Y, Z分别是一个有向无环图G里互没有交集的结点集，Z阻塞X中的一结点到Y中一结点的通路P(关于“通路”，在这里只要连通就算一条通路，对路中每条边的方向无任何要求)，当且仅当满足以下条件之一：
>
> 1. P中存在顺序结构$x \to z \to y$或同父结构$x \leftarrow z \to y$，结点z包含在集合Z中；
> 2. P中存在V型结构$x \to z \leftarrow y$，结点z及其孩子结点不包含在集合Z中。

> **定理 1** (条件独立). 设X, Y, Z分别是一个有向无环图G里互没有交集的结点集，如果集合Z阻塞X到Y的任何一条道路，则X和Y在给定Z时条件独立，即$X \perp\!\!\!\perp Y | Z$。

请根据定理1，判断第一问中有哪些条件独立的组合（独立也算条件独立的一种特例）。

(4) **[10pts]** 由以上两问我们可知，道德化方法中的“除去集合z后，x和y分属两个连通分支”并不构成条件独立性的充要条件。如果对道德化方法稍加修改，在连接V型结构父结点前，我们只保留图中X, Y, Z及他们的非孩子结点，之后的步骤则相同。请问你认为用修改后的方法可以保证得到全部的正确的条件独立集合吗？如果可以，请说明理由；如果不能，请给出反例。

Proof. 此处用于写证明(中英文均可)

(1)

$$\begin{aligned} & P(A, B, C, D, E) \\ =& P(A)P(B|A)P(C|B, D, E)P(D|E)P(E|A) \end{aligned} \tag{1.1}$$

(2) 条件独立的组合：

$$\begin{aligned} & E \perp A | C \\ & E \perp B | C \\ & E \perp D | C \\ & A \perp D | B, C \end{aligned} \tag{1.2}$$

还有一些冗余的条件独立组合，也属于合法的条件独立组合：

$$\begin{aligned} &E \perp A|C,B \\ &E \perp A|C,D \\ &E \perp A|C,B,D \\ &E \perp B|C,A \\ &E \perp B|C,D \\ &E \perp B|C,A,D \\ &E \perp D|C,A \\ &E \perp D|C,B \\ &E \perp D|C,A,B \\ &A \perp D|B,C,E \end{aligned} \tag{1.3}$$

(3) 条件独立的组合：

$$\begin{aligned} &A \perp\!\!\!\perp C|B,E \\ &A \perp\!\!\!\perp D|E \\ &B \perp\!\!\!\perp D|A \\ &B \perp\!\!\!\perp D|E \\ &B \perp\!\!\!\perp E|A \end{aligned} \tag{1.4}$$

还有一些冗余的条件独立组合，也属于合法的条件独立组合：

$$\begin{aligned} &A \perp\!\!\!\perp C|B,E,D \\ &A \perp\!\!\!\perp D|B,E \\ &B \perp\!\!\!\perp D|A,E \\ &B \perp\!\!\!\perp E|A,D \end{aligned} \tag{1.5}$$

(4) 修改后的方法可以得到全部的正确的条件独立的集合。令X和Y为单一节点x和y，考虑x，y和z之间的不同情况。

情况一：x和y本来是连通的，z不存在于x和y之间的任何通路，那么根据充要条件，x和y并不会关于z条件独立。根据修改后的算法，道德化之后，删除掉z之后，x和y仍然是连通的。所以得到了相同的结果。

情况二：V型结构，即$x \to z \leftarrow y$，那么根据充要条件，假如有$x \perp\!\!\!\perp y|Z$，z一定不在Z中。如果采用修改后的算法，道德化过程中x和y之间连通，那么删除z并不会阻断x和y在同一分支。所以得到了相同的结果。

情况三：顺序结构或者同父结构，即$x \to z \to y$或者$x \leftarrow z \to y$，那么根据充要条件，存在某个$x \perp\!\!\!\perp y|Z$，使得z在Z中。根据修改后的算法，删除z之后会阻塞$x-z-y$这条通路。如果$Z=\{z\}$，那么修改后的算法仍然可以得到$x \perp\!\!\!\perp y|Z$，如果$Z=\{z,a_1,...a_n\}$，那么z仍然会处于这样的集合Z中，即仍然可以得到$x \perp\!\!\!\perp y|Z$。所以会得到相同的结果。

情况四：x和y本来是不连通的，根据充要条件x和y是独立，使用修改后的算法删除z之后仍

然是不连通的，所以得到相同的结果。

综上可得，可以得到全部的正确的条件独立的集合。

2 [20pts] Naive Bayes Classifier

通过对课本的学习，我们了解了采用“属性条件独立性假设”的朴素贝叶斯分类器。现在我们有如下表所示的一个数据集：

表 1: 数据集

编号	x_1	x_2	x_3	x_4	y
样本1	1	1	1	0	1
样本2	1	1	0	0	0
样本3	0	0	1	1	0
样本4	1	0	1	1	1
样本5	0	0	1	1	1

(1) **[10pts]** 试计算：$\Pr\{y=1|\mathbf{x}=(1,1,0,1)\}$ 与 $\Pr\{y=0|\mathbf{x}=(1,1,0,1)\}$ 的值；

(2) **[10pts]** 使用“拉普拉斯修正”之后，再重新计算上一问中的值。

Solution.

(1)

$$\Pr\{y=1|\mathbf{x}=(1,1,0,1)\}=\frac{\Pr\{\mathbf{x}=(1,1,0,1)|y=1\}\Pr\{y=1\}}{\Pr\{\mathbf{x}=(1,1,0,1)\}} \tag{2.1}$$

根据“属性条件独立性假设”的思想，有：

$$\begin{aligned}&\Pr\{\mathbf{x}=(1,1,0,1)|y=1\}\\=&\Pr\{x_1=1|y=1\}\Pr\{x_2=1|y=1\}\Pr\{x_3=0|y=1\}\Pr\{x_4=1|y=1\}\\=&\frac{2}{3}\times\frac{1}{3}\times\frac{0}{3}\times\frac{2}{3}\\=&0\end{aligned} \tag{2.2}$$

所以可得：

$$\Pr\{y=1|\mathbf{x}=(1,1,0,1)\}=0 \tag{2.3}$$

类似的

$$\Pr\{y=0|\mathbf{x}=(1,1,0,1)\}=\frac{\Pr\{\mathbf{x}=(1,1,0,1)|y=0\}\Pr\{y=0\}}{\Pr\{\mathbf{x}=(1,1,0,1)\}} \tag{2.4}$$

根据“属性条件独立性假设”的思想，有：

$$\begin{aligned}&\Pr\{\mathbf{x}=(1,1,0,1)|y=0\}\\=&\Pr\{x_1=1|y=0\}\Pr\{x_2=1|y=0\}\Pr\{x_3=0|y=0\}\Pr\{x_4=1|y=0\}\\=&\frac{1}{2}\times\frac{1}{2}\times\frac{1}{2}\times\frac{1}{2}\\=&\frac{1}{16}\end{aligned} \tag{2.5}$$

以及

$$\Pr\{y=0\}=\frac{2}{5} \tag{2.6}$$

综上可得

$$\begin{aligned}&\Pr\{y=0|\mathbf{x}=(1,1,0,1)\}\\=&\frac{\Pr\{\mathbf{x}=(1,1,0,1)|y=0\}\Pr\{y=0\}}{\Pr\{\mathbf{x}=(1,1,0,1)\}}\\=&\frac{\frac{1}{40}}{\Pr\{\mathbf{x}=(1,1,0,1)\}}\end{aligned} \tag{2.7}$$

因为$\Pr\{y=1|\mathbf{x}=(1,1,0,1)\}=0$所以可得

$$\Pr\{y=0|\mathbf{x}=(1,1,0,1)\}=1 \tag{2.8}$$

综上可得

$$\begin{aligned}&\Pr\{y=1|\mathbf{x}=(1,1,0,1)\}=0\\&\Pr\{y=0|\mathbf{x}=(1,1,0,1)\}=\frac{\frac{1}{40}}{\Pr\{\mathbf{x}=(1,1,0,1)\}}\end{aligned} \tag{2.9}$$

(2) 根据上题，结合“拉普拉斯修正”，可得：

$$\begin{aligned}&\Pr\{\mathbf{x}=(1,1,0,1)|y=1\}\\=&\Pr\{x_1=1|y=1\}\Pr\{x_2=1|y=1\}\Pr\{x_3=0|y=1\}\Pr\{x_4=1|y=1\}\\=&\frac{3}{5}\times\frac{2}{5}\times\frac{1}{5}\times\frac{3}{5}\\=&\frac{18}{625}\end{aligned} \tag{2.10}$$

还有：

$$\Pr\{y=1\}=\frac{4}{7} \tag{2.11}$$

所以得到：

$$\begin{aligned}&\Pr\{y=1|\mathbf{x}=(1,1,0,1)\}\\=&\frac{\Pr\{\mathbf{x}=(1,1,0,1)|y=1\}\Pr\{y=1\}}{\Pr\{\mathbf{x}=(1,1,0,1)\}}\\=&\frac{\frac{72}{4375}}{\Pr\{\mathbf{x}=(1,1,0,1)\}}\end{aligned} \tag{2.12}$$

类似的，根据”拉普拉斯修正”，可得：

$$\begin{aligned}&\Pr\{\mathbf{x}=(1,1,0,1)|y=0\}\\=&\Pr\{x_1=1|y=0\}\Pr\{x_2=1|y=0\}\Pr\{x_3=0|y=0\}\Pr\{x_4=1|y=0\}\\=&\frac{2}{4}\times\frac{2}{4}\times\frac{2}{4}\times\frac{2}{4}\\=&\frac{1}{16}\end{aligned} \tag{2.13}$$

还有：

$$\Pr\{y=0\}=\frac{3}{7} \tag{2.14}$$

所以得到：

$$\begin{aligned}&\Pr\{y=0|\mathbf{x}=(1,1,0,1)\}\\=&\frac{\Pr\{\mathbf{x}=(1,1,0,1)|y=0\}\Pr\{y=0\}}{\Pr\{\mathbf{x}=(1,1,0,1)\}}\\=&\frac{\frac{3}{112}}{\Pr\{\mathbf{x}=(1,1,0,1)\}}\end{aligned}\tag{2.15}$$

综上可得

$$\begin{aligned}\Pr\{y=1|\mathbf{x}=(1,1,0,1)\}&=\frac{\frac{72}{4375}}{\Pr\{\mathbf{x}=(1,1,0,1)\}}\\\Pr\{y=0|\mathbf{x}=(1,1,0,1)\}&=\frac{\frac{3}{112}}{\Pr\{\mathbf{x}=(1,1,0,1)\}}\end{aligned}\tag{2.16}$$

3 [50pts] Ensemble Methods in Practice

由于出色的性能和良好的鲁棒性，集成学习方法 (Ensemble methods) 成为了极受欢迎的机器学习方法，在各大机器学习比赛中也经常出现集成学习的身影。在本次实验中我们将结合两种经典的集成学习思想：Boosting和Bagging，对集成学习方法进行实践。

本次实验选取UCI数据集Adult，此数据集为一个二分类数据集，具体信息可参照链接，为了方便大家使用数据集，已经提前对数据集稍作处理，并划分为训练集和测试集，大家可通过此链接进行下载。

由于Adult是一个类别不平衡数据集，本次实验选用AUC作为评价分类器性能的评价指标，AUC指标的计算可调用sklearn算法包。

(1) **[5pts]** 本次实验要求使用Python 3或者Matlab编写，要求代码分布于两个文件中，BoostMain.py、RandomForestMain.py (Python) 或 BoostMain.m、RandomForestMain.m (Matlab)，调用这两个文件就能完成一次所实现分类器的训练和测试；

(2) **[35pts]** 本次实验要求编程实现如下功能：

- **[10pts]** 结合教材8.2节中图8.3所示的算法伪代码实现AdaBoost算法，基分类器选用决策树，基分类器可调用sklearn中决策树的实现；
- **[10pts]** 结合教材8.3.2节所述，实现随机森林算法，基分类器仍可调用sklearn中决策树的实现，当然也可以自行手动实现，在实验报告中请给出随机森林的算法伪代码；
- **[10pts]** 结合AdaBoost和随机森林的实现，调查基学习器数量对分类器训练效果的影响 (参数调查)，具体操作如下：分别对AdaBoost和随机森林，给定基分类器数目，在训练数据集上用5折交叉验证得到验证AUC评价。在实验报告中用折线图的形式报告实验结果，折线图横轴为基分类器数目，纵轴为AUC指标，图中有两条线分别对应AdaBoost和随机森林，基分类器数目选取范围请自行决定；
- **[5pts]** 根据参数调查结果，对AdaBoost和随机森林选取最好的基分类器数目，在训练数据集上进行训练，在实验报告中报告在测试集上的AUC指标；

(3) **[10pts]** 在实验报告中，除了报告上述要求报告的内容外还需要展现实验过程，实验报告需要有层次和条理性，能让读者仅通过实验报告便能了解实验的目的，过程和结果。

实验报告.

1. **实验说明** 本次实验环境为：
   OS：Windows10
   编程语言：python 3.6
   实验的输入文件放在/*adult_dataset*/文件夹中。
   运行方法：
   python BoostMain.py
   python RandomForestMain.py

2. **实验目的** 本次实验的主要工作在于实现集成学习的经典算法–AdaBoost算法和随机森林算法。在实现算法之后通过交叉验证的手段来探究基分类器数目和算法性能之间的关系，寻找最优的基分类器数目。在这个过程中，加深对于AdaBoost算法和随机森林算法的理解，同时掌握参数调试的技巧和思想。

3. **实验过程**

a. AdaBoost算法的实现使用python 3.6，借助了sklearn包。

具体的，首先要定义class AdaBoost：

关键的成员变量为α和h。其中α是不同的基分类器对应的权重，h是所有的基分类器组成的列表。

关键的成员函数为fit和$predict$。其中fit函数负责了分类器的训练工作，$predict$则负责了测试样例的预测工作。

fit函数根据参数$Xtrain, yTrain, T$来执行。函数会进行分类器数目T次迭代。在第t次迭代时，根据训练样例$XTrain, yTrain$，样例权重D_t来训练一个新的基分类器h_t，根据预测结果$yPred = h_t(XTrain, D_t)$和$yTrain$来计算误差ϵ：

$$incorrect_i = \begin{cases} 1 & yPred_i = yTrain_i; \\ 0 & yPred_i \neq yTrain_i; \end{cases} \tag{3.1}$$

$$\epsilon = average(incorrect, weights = D_t) \tag{3.2}$$

其中$average$是计算加权平均的函数。得到误差之后就可以得到该分类器的权重α_t，即：

$$\alpha_t = \frac{1}{2} \ln \frac{1-\epsilon}{\epsilon} \tag{3.3}$$

保存h_t和α_t，然后更新样例权重：

$$D_{t+1}(X_i) = \frac{D_t(X_i) \exp(-\alpha_t y_i h(X_i))}{Z_t} \tag{3.4}$$

其中

$$Z_t = \sum_{i=1}^{n} D_t(X_i) \exp(-\alpha_t y_i h(X_i)) \tag{3.5}$$

注意到D_1设置每一项为$\frac{1}{m}$。

$predict$函数接受测试数据$XTest$之后，使用α和h_t来进行预测

$$H(XTest) = sign(\sum_{t=1}^{T} \alpha_t h_t(XTest)) \tag{3.6}$$

在实现基分类器的简单决策树的时候，调用了sklearn包的sklearn.tree.DecisionTreeClassifier。

b. 随机森林算法的实现使用python 3.6，借助了sklearn包。

类似的，也要定义class RandomForest：

关键的成员变量为h，即所有基分类器组成的列表。

Algorithm 1 RandomForestFit

Require:

X：训练集属性

y：训练集类别

T：基分类器数目

k：随机决策树每次划分时随机选择的属性个数

Ensure: h：基分类器列表

for t from 1 to T **do**

    $Xs, ys = Sample(X, y)$

    $f = ExtraTreesFit(Xs, ys, k)$

    $h.append(f)$

end for

关键的成员函数为fit和$predict$。

随机森林算法伪代码如下：

fit函数进行T次迭代，每次迭代对训练集进行$bootstrap$采样，然后用采样到的样本训练

Algorithm 2 RandomForestPredict

Require: X：测试集属性

Ensure: $pred$：预测的类别

$pred = 0$

for t from 1 to T **do**

    $pred = pred + h_t(X)$

end for

$pred = sign(pred)$

一个随机决策树模型并保存。随机决策树模型在每次划分的时候会先随机挑选k个属性，然后在其中寻找最优的划分属性。

$predict$函数用T个随机决策树的模型对测试样例进预测，然后将预测结果相加取符号作为输出的预测结果：

$$H(XTest) = sign(\sum_{t=1}^{T} h_t(XTest)) \tag{3.7}$$

在实现基分类器的随机决策树的时候，调用了sklearn包的sklearn.tree.ExtraTreeClassifier。

c. 对AdaBoost和随机森林算法使用5折交叉验证来寻找最优的基分类器数目。5折交叉验证的做法为：首先将初始训练集随机划分成等分的5份，选择其中的四份作为新的训练集XTr和yTr，剩下一份作为新的验证集XCV和yCV。用XTr和yTr来训练AbaBoost模型或者随机森林模型，训练好模型之后在XCV上进行预测，预测得到的结果$yPred$和yCV计算AUC指标。事先确定基分类器范围，然后在范围内进行上述交叉验证AUC评价，统计所有AUC指标，找到最大的AUC对应的基分类器数目t，作为AdaBoost预测测试集所用的基分类器数目。

注意到划分时为了防止类别不平衡，所以采用了StratifiedSplit，即在划分的子集中各个类别仍然保持和原来一样的比例。

过程伪代码如下：

Algorithm 3 CrossValidationForAuc

Require: $XTrain$：训练集属性

$yTrain$：训练集标签

N：基分类器范围

Ensure: T：最优基分类器数目

$maxAuc = 0, T = 1$

for t from 1 to N **do**

  $Auc = 0$

  for TrIdx, CVIdx in StratifiedSplit(XTrain, yTrain) **do**

    $XTr, yTr = XTrain[TrIdx], yTrain[TrIdx]$

    $XCV, yCV = XTrain[CVIdx], yTrain[CVIdx]$

    $f = model.fit(XTr, yTr)$

    $yPred = f.predict(XCV)$

    $Auc = Auc + ComputeAuc(yCV, yPred)$

  end for

  if $Auc > maxAuc$ **then**

    $T = t$

    $maxAuc = Auc$

  end if

end for

对于AdaBoost和随机森林模型使用交叉验证得到的结果如下：

从图中可以看到对于随机森林来说，基分类器在1到15的区间中，AUC上升较快，当基分

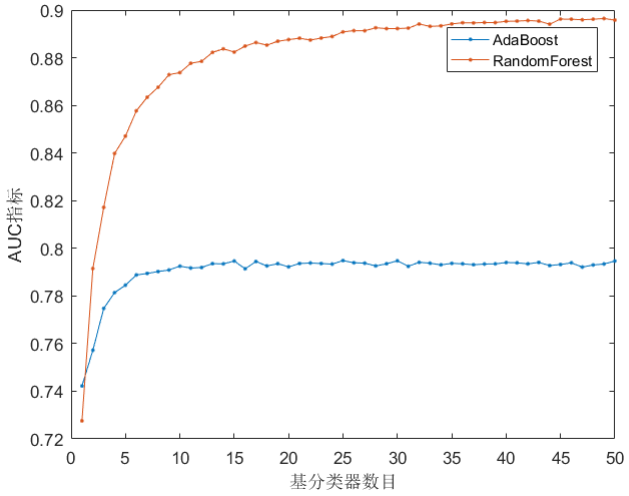


图 1: AdaBoost和随机森林在不同基分类器数目时的AUC指标

类器超过40以后，AUC上升就变得十分缓慢。对于AdaBoost来说，基分类器在1到10的区间中，AUC上升较快，超过10之后，AUC的上升十分缓慢。考虑到基分类器数目大于50之后效果提升微弱，同时基分类器数目过多也会造成训练开销过大，所以控制范围在1到50之间。

d. 选取最优的基分类器数目，即选取在交叉验证时AdaBoost和随机森林取得最好AUC的数目，即AdaBoost选取25个基分类器，随机森林选取49个基分类器。在测试集上测试得到的结果如下：

表 2: AdaBoost和随机森林在测试集上的AUC指标

算法	AUC指标
AdaBoost	0.8288801089639083
随机森林	0.8942615694173404

以0作为阈值，score大于等于0视为正例，反之视为负例，得到AdaBoost和随机森林其他性能如下：

表 3: AdaBoost性能

	precision	recall	f1-score	support
负例	0.88	0.87	0.88	12435
正例	0.6	0.62	0.61	3846

表 4: 随机森林性能

	precision	recall	f1-score	support
负例	0.88	0.92	0.90	12435
正例	0.71	0.61	0.66	3846

4. 实验结果本次实验完成了题目的所有要求：包括AdaBoost和随机森林算法的实现，使用交叉验证方法调整参数，在测试集上完成预测并且得到AUC性能指标。

实验心得：

1.sklearn是非常强大的工具，借助其中的决策树模型可以减轻很多工作量。

2.AdaBoost算法虽然不复杂，但是实现过程中会有一些细节性的问题，比如error的计算要加上权重。

3.AUC的计算使用score效果会很多。

4.使用交叉验证可以方便快捷的进行参数调整。

5.将模型实现成class的形式便于coding。