同济大学

TONGJI UNIVERSITY

毕业设计(论文)

课题名称	基于决策树的正数据与无标注数据学习
副标题	银行信贷风险的预测
学　院	数学科学学院
专　业	统计学
学　号	1653469
学生姓名	王馨仪
指导教师	吴昊
日　期	2020-06-12

基于决策树的正数据与无标签数据学习

银行信贷风险的预测

摘要

正数据与无标签学习 (PU-Learning) 需要从无标签的数据集和正数据集中学习感兴趣类别的分类器，而没有关于负数据的信息. 它在实际生活中有着很广泛的应用，比如就银行信贷风险预测而言，在之前的预测模型中，一般采用监督学习中的分类器. 但事实上，已知的数据标签中大多是正标签（即信誉好的顾客），极少知道或没有负标签（即信誉不够好的顾客），此时无法使用监督学习中的分类器. 本文基于梯度提升树，考虑正数据与无标签数据学习（PU-Learning）的目标函数，构建一个正数据与无标签数据学习的二分类分类器，来对银行信贷风险进行预测.

关键词: 机器学习，正数据与无标签学习, 梯度提升决策树, 信贷预测，K-L 散度

装 订 线

Positive-unlabeled (PU) Learning Based on Decision Tree

Credit Risk Prediction

ABSTRACT

Positive-unlabeled (PU) learning needs to learn classifiers of interest categories from unlabeled datasets and positive datasets, without extra information about negative data. It has a wide range of applications in real life. For example, in terms of bank credit risk prediction, in previous prediction models, the classifiers in supervised learning are generally used. But in fact, most of the known data labels are positive labels (that is, customers with good reputation), and we rarely know or have no negative labels (that is, customers with bad reputation). At this time, the classifiers in supervised learning cannot be used. Based on the gradient boosting tree, this paper considers the objective function in positive and unlabeled data learning (PU-Learning), and constructs a binary classification classifier to predict bank credit risk.

Keywords: Machine learning, PU Learning, Gradient Boosting Tree, Credit Prediction, K-L Divergence

装 订 线

目录

装 订 线

1 绪论

1.1 信贷风险预测的意义

我国金融体系以银行业为主导，信贷业务作为银行主要收入来源，与之相应的信贷风险成为面临的首要风险. 现实中，我国银行业信贷风险管理体制滞后，信贷风险量化技术有待加强；否则将影响自身的生存发展及其在国际上的竞争力. 而对于银行业而言，贷款利息是其主要收入来源，能否准确评估借贷用户的信用度，则是其发放贷款的重中之重.

1.2 常用预测信贷的分类算法

常用的预测用户信贷能力的分类算法有：

Logistic 回归：Losgistic 回归本质上是采用了 Sigmoid 函数映射的多元线性回归问题，属于有监督学习. Logistic 回归将多元线性回归得到的结果映射到集合 [0,1]，再根据阈值离散化结果.

决策树：决策树算法采用树形结构，使用层层推理来实现最终的分类. 决策树由下面几种元素构成：1. 根节点：包含样本的全集；2. 内部节点：对应特征属性测试；3. 叶节点：代表决策的结果. 预测时，在树的内部节点处用某一属性值进行判断，根据判断结果决定进入哪个分支节点，直到到达叶节点处，得到分类结果. 这是一种基于 if-then-else 规则的有监督学习算法，决策树的这些规则通过训练得到，而不是人工制定的.

决策树和集成学习：通常情况下，仅仅用一棵决策树进行预测，准确率不高且容易出现过拟合的现象. 这时我们考虑用集成方法，用多棵树进行学习，提高预测准确率. 许多决策树算法的变形都利用了这种思想，比如说 Adaboost(Freund, Yoav; Schapire, Robert E, 1997)，GBDT(Friedman, J. H., 1999)，Xgboost(Chen, Tianqi; Guestrin, Carlos, 2016)，LightGBM(Guolin Ke, Qi Meng, et al., 2017) 等等.

本文考虑用梯度提升决策树 (GBDT) 来进行信贷预测. GBDT 的核心就在于，每一棵树学的是之前所有树结论和的残差 (负梯度)，这个残差就是一个加预测值后能得真实值的累加量. 比如 A 的真实年龄是 18 岁，但第一棵树的预测年龄是 12 岁，差了 6 岁，即残差为 6 岁. 那么在第二棵树里我们把 A 的年龄设为 6 岁去学习，如果第二棵树真的能把 A 分到 6 岁的叶子节点，那累加两棵树的结论就是 A 的真实年龄；如果第二棵树的结论是 5 岁，则 A 仍然存在 1 岁的残差，第三棵树里 A 的年龄就变成 1 岁，继续学. 这种集成方式的最大好处在于，每一步的残差计算其实变相地增大了分错样本的权重，而已经分对的样本则都趋向于 0. 这样后面的树就能越来越专注那些前面被分错的样本.

同时，本文在借鉴梯度提升决策树思想的同时，改进了算法的分类函数，使之可以运用在更广泛的问题中.

1.3 有监督学习算法的局限性

在只知道正标签数据、或者正标签数据量远远大于负标签数据量的情况下，该如何预测用户信贷能力. 这个问题在银行系统中十分常见，一般来说，银行的大部分顾客有能力并且也的确支付了贷款，只有极少个别人无力偿还贷款，可以视作“黑用户”.

事实上，我们拿到 15281 条数据中，仅有 2694 条是负数据，及没有偿还贷款的“黑用户”，其余 12587 条数据全都是正数据，及偿还了贷款的“白用户”. 因而，像之前的预测信贷能力的方法，像传统的 Logistic 回归和决策树，两者都是有监督学习，要求提供足够的负数据，这在实际生活中存在一定困难，因而我们考虑，只使用正标签数据和无标签数据的半监督学习，对用户的信贷能力进行预测.

1.4 正数据和无标签学习 (PU-Learning)

在实际生活中，正数据和无标签学习（PU-Learning）是一种很实用的算法. 就像银行信贷风险预测一样，我们面临着从一些正数据和大量未标记数据构建二元分类模型的任务，而没有关于负数据的额外信息.

PU-Learning 在机器学习中也是众所周知的，可用于多种任务，例如多视图学习 (J.T. Zhou, et al., 2012) 和半监督学习 (T. Sakai, M. C. d. Plessis, 2016). 它还可用于数据挖掘中以对数据流 (X. Li, S. Y. Philip, et al.,2009) 或时间序列 (M.N. Nguyen, X.-L. Li, et al., 2011) 进行分类，并检测图形中的事件（例如共现）(J. Silva and R. Willett, 2009).

一般有两种方式来建立 PU 分类器：

1. 两步方法（Two-step Approach). 该方法启发式地从未标注样本里找到可靠的负数据，然后再用监督学习的方法训练二分类器，该方法问题是分类效果严重依赖先验知识，并且该方法假设负数据集和正数据集相互独立，交集为空.

2. 将无标签数据直接作为负样本来训练分类器，然后用这些带有正/负标签的数据来训练分类器，通过不断优化目标函数来减少原始的分类误差. 近几年，(M. Du Plessis, 2014；2015)，提出了两个基于统计学习理论的框架. 这些框架不依赖先验知识，不要遭受原始标签分配错误的问题，并为泛化误差提供了理论的边界. 但这些方法的局限性在于，需要事先知道正数据（包括已标签和未标签的）占总体数据的比例.

在本文中，针对银行信贷风险预测问题，本文采用了 (Hui Chen, et al., 2020) 的目标函数和梯度提升树，构建了一个 PU 分类器，且不需要事先知道正数据（包括已标签和未标签的）占总体数据的比例. 该 PU 分类器对于信贷数据的预测效果良好，近似于使用监督学习目标函数的分类器.

本文的行文思路为：2.1-2.2 介绍了分类器：梯度提升决策树；2.3-2.4 介绍了目标函数，并给出详尽的算法流程；3.1-3.2 将 PU 分类器带入实际数据集，进行信贷风险预测，并给出敏感度分析和与监督学习的效果比较；4 给出了总结与未来工作展望.

1.5 需解决的问题

本文解决的问题如下：

1) 如何在只知道其中一种标签（正标签）的情况下，对数据进行分类. 而这个问题的关键在于选取合适的目标函数.

2) 采用何种分类器，本文考虑决策树的某种变形.

3) 要预测的数据集存在很大的数据不平衡问题，近 90% 的数据为正标签，剩下的为负标签，如何处理这种不平衡.

4) 正标签数据的先验概率分布未知，因而我们无法采用 (M. Du Plessis, 2014；2015) 中提到的无偏估计法.

2 理论部分

从本章开始本文开始介绍具体的算法构思. 首先，本章讲首先介绍梯度提升决策树（分类树），然后介绍本文中使用的正数据与无标签数据学习的目标函数，并且给出具体算法的伪代码.

2.1 梯度提升树

2.1.1 问题引入

在“函数估计”或者“预测标签”类的问题中，往往给定一组随机的“输出变量”或“相应变量” y，以及一组随机的“输入变量”或“解释变量” $\mathbf{x}=\{x_1,\cdots,x_n\}$. 用一组训练集 $\{y_i,\mathbf{x}_i\}_1^N$，其中 $\mathbf{y}=\{y_i\}_1^N$ 已知或部分已知（半监督学习），目的是要获得一个估计的近似函数 $\tilde{F}(\mathbf{x})$ 来最小化目标函数 $L(y,F(x))$ 在联合分布 $(y,\mathbf{x})$ 的数学期望.

$$F^*=\arg\min_F E_{y,x}L(y,F(x))=\arg\min_F E_x[E_y(L(y,F(\mathbf{x}))|\mathbf{x}] \tag{1}$$

这里的目标函数 $L(y,F(x))$ 是自定义的，稍后将给出本文构造的目标函数及其意义.

本文着重考虑二分类问题，即“解释变量”只有两种取值：$\mathbf{y}=\{y_i\}_1^N, y_i\in\{-1,1\}$. 假设我们有一个正数据集 $\mathcal{P}=\{x_1,x_2,\cdots,x_M\}$，即该数据集上 $y_i=1,\forall x_i\in\mathcal{P}$，同时也有一个无标签数据集 $\mathcal{U}=\{x_{M+1},\cdots,x_N\}$，即该数据集上 $y_i,\forall x_i\in\mathcal{P}$ 未知. 我们将 $\mathcal{X}=\mathcal{P}\cup\mathcal{U}$ 作为训练集，目标是找分类器，来预测某个未知集合中元素的分类标签 $\tilde{y_i},\tilde{y_i}\in\{1,-1\}$.

2.1.2 数值优化

假设预测函数 $F(\mathbf{x})$ 以 $\mathbf{P}=\{P_1,P_2,\cdots\}$ 为参数，并且可以写成若干个弱分类器之和，其中 $\mathbf{P}=\{\beta_m,\alpha_m\}_0^M$，第 m 个弱分类器表示为 $h(\mathbf{x};\alpha_m)$，而 β_m 表示第 m 个分类器的权重，α_m 表示第 m 个分类器的参数. 接下来，考虑找到最优的参数 (β_m,α_m) 来最小化目标函数，即优化问题[1]等价于：对于 N 个样本点 $\{\mathbf{x}_i,y_i\}$，找到参数 (β_m,α_m)，$m=0,1,2,\cdots,M$，使得:

$$(\beta_m,\alpha_m)=\arg\min_{\alpha,\beta}\sum_{i=1}^{N}L(y_i,F_{m-1}(\mathbf{x}_i+\beta h(\mathbf{x}_i;\alpha)) \tag{2}$$

弱分类器 $h(\mathbf{x};\alpha)$ 通常是一个比较简单的、参数化的分类函数，$\mathbf{x}$ 是其输入变量，α 是其参数. 像等式[2]这样的展开是许多近似算法的核心，比如说神经网络算法（Rumelhart, Hinton, and Williams 1986），高斯径向基函数核 (Powell, 1987)，MARS (Friedman, 1991)，小波函数 (Donoho, 1993)，支持向量机 (Vapnik, 1995) 等等. 这里我们比较感兴趣的情况是：$h(\mathbf{x};\alpha)$ 是一棵决策树，比如说 $CART^{TM}$ (Breiman, Friedman, Olshen, and Stone, 1983). 对于一棵决策树来说，参数 α 是节点数，节点位置，最大深度等参数.

接下来，在优化问题[2]中，最关键的是：1. 求目标函数的最小值的方法. 2. 选取合适的弱分类器. 之后，本文将详细介绍这两个问题.

2.1.3 梯度下降法

梯度下降法 (Cauchy 1847) 是一种求解局部最小值的方法，核心是将搜索方向设定为负梯度方向，搜索步长与之成比例. 梯度下降法是一种比较简单、快速的优化算法.

注意在梯度提升和梯度下降的分析方式是一致的，只不过把的更新中梯度减号变为加号.

下面考虑用梯度下降法，来求解 2.1.2 中的优化问题.

a) 首先定义初始化弱分类器常数 ρ，其中 $F_0(\mathbf{x})$ 表示初始化的弱分类器，常数 ρ 使得初始预测目标函数（损失函数）达到最小值：

$$F_0(\mathbf{x}) = \arg\min_{\rho} \sum_{i=1}^{N} L(y_i, \rho)$$

b) 在每次迭代中都构造一个弱分类器，并设第 m 次迭代后得到的预测函数为 $F_m(\mathbf{x})$，相应的预测损失函数为 $L(y, F_m(\mathbf{x})$，为使得预测损失函数减少得最快，第 m 个弱分类器 $\beta_m h(\mathbf{x}; \alpha_m)$ 应该建立在第前 $m-1$ 次迭代生成的预测损失函数上，即：

$$-g_m(\mathbf{x}_i) = -\left[\frac{\partial L(y_i, f(\mathbf{x}_i))}{\partial F(\mathbf{x}_i)}\right]_{F(\mathbf{x}_i)=F_{m-1}(\mathbf{x}_i)}, i = 1, \ldots, N \tag{3}$$

基于求得的梯度下降方向，参数 α_m 是使得弱分类器 $h(\mathbf{x}; \alpha_m)$ 沿此方向逼近的参数值，即：

$$\alpha_m = \arg\min_{\alpha} \sum_{i=1}^{N} [-g_m(\mathbf{x}_i) - \beta h(\mathbf{x}_i; \alpha)]^2 \tag{4}$$

β_m 是沿此方向搜索的最优步长，即：

$$\beta_m = \arg\min_{\beta} \sum_{i=1}^{N} L(y_i, F_{m-1}(\mathbf{x}_i) + \beta h(\mathbf{x}_i; \alpha)) \tag{5}$$

c) 每次更新迭代后的预测函数，即 $F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \beta_m h(\mathbf{x}; \alpha_m)$，若相应的预测损失函数满足误差收敛条件或某个参数达到设定的阈值，则终止迭代.

为了避免过拟合现象，通常在每个弱分类器前乘上“学习速率” ν，值域为 $(0, 1]$，ν 值越小，学习越保守，达到同样精度需要的迭代次数 M 越大；ν 值越大，学习越快速，越容易出现过拟合现象：

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \nu \cdot \beta_m h(\mathbf{x}; \alpha_m) \tag{6}$$

2.1.4 回归树

这里我们考虑弱分类器为一个包含 J 个节点的回归树的情况. 当我们的基本分类器是一个包含 J 个节点的回归树时，回归树模型可以表示为：

$$h(x; \{b_j, R_j\}_1^J) = \sum_{b=j}^{J} b_j I(x \in R_j) \tag{8}$$

其中 $\{R_j\}_1^J$ 不相交的区域，它们的集合覆盖了预测值的空间，$\{b_j\}_1^J$ 是叶子节点的值，可以认为是模型 h 的系数. 利用回归树模型，2.1 中步骤 c) 的公式可以被替换为：

$$F_m(x) = F_{m-1}(x) + \rho_m \sum_{j=1}^{J} b_{jm} I(x \in R_{jm}) \tag{9}$$

其中 $\{R_{jm}\}_1^J$ 是第 m 次迭代生成的树所产生的区域. 第 m 次迭代的树用来预测 $\{\tilde{y}_i\}_i^N$。$\{b_{jm}\}$ 可以被表示为

$$b_{jm} = ave_{x_i \in R_{jm}} \tilde{y}_i$$

即用平均值表示该叶子节点拟合的值, 指示函数 $I(\cdot)$ 为 1，如果判断条件成立，否则为 0.

有了下降的方向，我们还需要最好的步长，缩放因子 ρ_m 是步骤 b) 中线性搜索方式的一种解决方案.

从上面可以看出，我们是先求的 b_{jm}，然后在求解 ρ_m，我们能否同时求解呢?

令 $\gamma_{jm} = \rho_m b_{jm}$，2.1 中的公式6 可以被表示为:

$$F_m(x) = F_{m-1}(x) + \sum_{j=1}^{J} \gamma_{jm} I(x \in R_{jm}) \qquad (10) \tag{7}$$

通过优化如下公式来获取最优的系数 γ_{jm}:

$$\{\gamma_{jm}\}_1^J = argmin_{\ \gamma_j\ _1^J} \sum_{i=1}^{N} L\left(y_i, F_{m-1}(x_i) + \sum_{j=1}^{J} \gamma_j I(x \in R_{jm})\right) \qquad (11) \tag{8}$$

由于回归树产生的叶子节点各个区域之间是不相交的，且所有的样本最终都会属于某个叶子节点，所以公式8可以表示为:

$$\gamma_{jm} = argmin_\gamma \sum_{x_i \in R_{jm}} L(y_i, F_{m-1}(x_i) + \gamma) \tag{9}$$

给定当前 $F_{m-1}(x_i)$，γ_{jm} 可以作为叶子节点的值，该值可以看做是基于损失函数 L 的每个叶子节点的最理想的常数更新值，也可以认为 γ_{jm} 是即有下降方向又有下降步长的值.

综上，用回归树作为基本分类器的梯度提升算法流程可以如下表示:

Algorithm 1 GBDT

1: $F_0(x) = c_0$

2: **for** $m = 1$ to M **do**

3: $\tilde{y}_i = -[\frac{\partial L(y, F(x_i))}{\partial F(x_i)}]_{F(x)=F_{m-1}(x)},\ i = 1, \cdots, N;$

4: $\{R_{jm}\}_1^J = J - terminal\ node\ tree(\{\tilde{y}_i, x_i\}_i^N);$

5: $\gamma_{jm} = argmin_\gamma \sum_{x_i \in R_{jm}} L(y_i, F_{m-1}(x_i) + \gamma);$

6: $F_m(x) = F_{m-1}(x) + \sum_{j=1}^{J} \gamma_{jm} I(x \in R_{jm});$

7: **end for**

其中 1 中的初始值 c_0 为 [-1,1] 间的常数；3 是计算残差（利用损失函数的负梯度在当前模型的值作为残差的近似值）；4 是拟合一颗含有 J 个叶子节点的回归树；5 是估计回归树叶子节点的值；6 是更新分类器.

2.2 概率转化函数

由于本文最终要实现的是二分类问题，需要通过某个转化函数，将分类器 $F(x)$ 映射成概率 $p(x) \in [0, 1]$，这种思想类似于 Logistic 回归. 这里使用的概率转化函数是 Logit 函数的某种变形.

$$F(x) = \frac{1}{2} log\left[\frac{\mathbb{P}(y=1|x)}{\mathbb{P}(y=-1|x)}\right]$$

转化成：

$$p_+(x) = \tilde{\mathbb{P}}(y=1|x) = \frac{1}{1+e^{-2F(x)}}$$
$$p_-(x) = \tilde{\mathbb{P}}(y=-1|x) = 1 - p_+(x)$$

该函数来源于 (Jerome H. Friedman, 2001)，与传统 Logit 函数不同在于：这里的分类器 $F(x)$ 是梯度提升树，而前者是多元线性回归函数. 该算法可以根据需求，使用不同的概率转化函数，这需要具体问题具体分析.

2.3 目标函数

在定义了分类器后，我们考虑如何选取合适的目标函数. 在监督学习中，对于一个二分类问题，比较常用的目标函数有：Hinge 损失函数 (Rennie, Jason D. M.; Srebro, Nathan, 2005)，Logistic 回归损失函数，交叉熵等. 而在正标签与无标签学习中，我们可以将这些目标函数加以修改，确保训练集中只有正标签数据和无标签数据.

本节采用 (Hui Chen, et al., 2020) 中的目标函数（损失函数），来实现梯度提升树的算法.

首先，本文构建函数的总体思路是从 Kullback-Leibler 散度（K-L 散度）出发，考虑原始分布和估计分布之间的差异. 考虑正标签数据集 $\mathcal{P}$ 的分布和被分类器（预测函数）$F(x)$ 网络重新构架的分布，用两者之间的 K-L 散度作为目标函数. 而当该目标函数取得最小值的时候，由贝叶斯公式可以知道，$F^*(x) = \mathbb{P}(y=1|\mathbf{x})$.

同时，当正标签数据集 $\mathcal{P}$ 趋近于总体数据集 $\mathcal{X} = \mathcal{P} \cup \mathcal{U}$ 时，本文的目标函数即为原始数据集和估计数据集的 K-L 散度.

2.3.1 Kullback-Leibler 散度 (KLD)

考虑某个观测到的、未知的分布 $p(x)$，假定用一个近似的分布 $q(x)$ 对它进行建模. 如果我们使用 $q(x)$ 来建立一个编码体系，用来把 x 的值传给接收者，那么由于我们使用了 $q(x)$ 而不是真实分布 $p(x)$，平均编码长度比用真实分布 $p(x)$ 进行编码增加的信息量为：

$$D_{KL}(p||q) = \mathbb{E}[log\ p(x) - log\ q(x)] \tag{10}$$

在离散情况下，上述公式10可以表示成：

$$D_{KL}(p||q) = \sum_{i=1}^{N} p(x_i) \cdot log\frac{p(x_i)}{q(x_i)} \tag{11}$$

其中，$p(x), q(x) \in [0,1]$，$\mathbb{E}[\cdot]$ 表示服从分布 $p(x)$ 的数学期望.

下面在此基础上，考虑构建正数据和无标签学习（PU-Learning) 的目标函数.

2.3.2 构建 PU-Learning 目标函数

定理 2.1. 定义目标函数 $L(\cdot)$ 为：

$$L(F(x)) \doteq log\mathbb{E}_f[F(x)] - \mathbb{E}_{f_P}[logF(x)] \tag{12}$$

则有：

$$L(F^*) \leq L(F) \tag{13}$$

其中，$F : \mathbb{R}^d \mapsto [0,1]$ 为分类器；$\mathbb{E}_f[\cdot]$ 和 $\mathbb{E}_{f_P}[\cdot]$ 分别表示服从分布 f 和 f_P 的数学期望；$F^*(x) = \mathbb{P}(y=1|\mathbf{x})$.

由定理2.1可以看出，每一个分类器 $F(x)$ 都是 $F^*(x) = \mathbb{P}(y=1|\mathbf{x})$ 的上界，因而我们可以通过最小化目标函数 $L(\cdot)$ 来估计 $\mathbb{P}(y=1|\mathbf{x})$，并且可以很容易地通过计算数据集 $\mathcal{P}$,$\mathcal{U}$ 的经验平均值来得到目标函数 $L(F(x))$.

需要注意的是，$L(c \cdot F(x)) = L(F(x)), c$，$c$ 是大于 0 的常数，即我们得到的最终结果与最小目标函数是成比例的.

2.3.3 正数据分布的估计

由贝叶斯定理，正数据的分布可以表示为：

$$f_P(x) = \frac{\mathbb{P}(y=1|x) \cdot \mathbb{P}(x)}{\int \mathbb{P}(x) \cdot \mathbb{P}(y=1|x)dx} \tag{14}$$

$$= \frac{F^*(x)}{\mathbb{E}_f[F^*(x)]} \cdot f(x) \tag{15}$$

而每一个分类器 F 则给出了一个 f_P 的近似：

$$f_F(x) = \frac{F(x)}{\mathbb{E}_f[F(x)]} \cdot f(x) \tag{16}$$

同时，f_P 和 f_F 之间的 K-L 散度可以表示成：

$$D_{KL}(f_P||f_F) = L(F) - L(F^*) \tag{17}$$

证明.

$$\begin{aligned} D_{KL}(f_P||f_F) &= \mathbb{E}_{f_P}\left[log\frac{f_P(x)}{f_F(x)}\right] \\ &= \mathbb{E}_{f_P}[logF^*(x)] + \mathbb{E}_{f_P}[logf(x)] - log\mathbb{E}_f[F^*(x)] \\ &- (\mathbb{E}_{f_P}[logF(x)] + \mathbb{E}_{f_P}[logf(x)] - log\mathbb{E}_f[F(x)]) \\ &= -L(F^*) + L(F) \end{aligned}$$

2.4 PU-GBDT 实现算法

这部分将 2.3 中构造的目标函数，带入梯度提升决策树，给出具体的计算流程.

首先，我们的目标函数为

$$L(F(x)) = log(\frac{1}{n_X}\sum_{x\in\mathcal{X}} F(x)) - \frac{1}{n_P}\sum_{x\in\mathcal{P}} logF(x) \tag{18}$$

其中，$\mathcal{X}=\mathcal{U}\cup\mathcal{P}$；$\mathcal{U}$ 和 $\mathcal{P}$ 是用于训练的无标签和正标签数据集；n_X 和 n_P 分别是集合 $\mathcal{X}$ 和 $\mathcal{P}$ 的基数.

接着，我们计算最速下降法中负梯度表达式：

$$\tilde{y}_i=-\left[\frac{\partial L(y,F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)}=\frac{-1}{\sum_{x\in\mathcal{X}}F_{m-1}(x)}+\frac{1}{n_P}\frac{1}{F_{m-1}(x_i)}$$

Algorithm 2 GBDT-PU Learning

1: $F_0(x)=c_0$
2: **for** $m=1$ to M **do**
3: $\quad\tilde{y}_i=\frac{-1}{\sum_{x\in\mathcal{X}}F_{m-1}(x)}+\frac{1}{n_P}\frac{1}{F_{m-1}(x_i)},\ i=1,\cdots,N;$
4: $\quad\{R_{jm}\}_1^J=J-terminal\,node\,tree(\{\tilde{y}_i,x_i\}_i^N);$
5: $\quad\gamma_{jm}=step\cdot ave_{x_i\in R_{jm}}\tilde{y}_i;$
6: $\quad F_m(x)=F_{m-1}(x)+\sum_{j=1}^J\gamma_{jm}I(x\in R_{jm});$
7: **end for**

这里，将算法 GBDT 步骤 5 中的最小化问题改成了 $step\cdot ave_{x_i\in R_{jm}}\tilde{y}_i$，其中 $step$ 为线搜索中的步长，取足够小时一定能保持收敛. 这样简化的原因是：在实际生活中，对于一个 20000×20 的数据集来说，如果在迭代时每一步最小化搜索步长，计算代价过大，取步长为常数，可以大大减少计算量，同时达到类似的收敛效果.

3 实验部分

3.1 信贷风险预测

实验的主要内容分为：1. 就具体某银行用户的数据集进行信贷预测. 2. 灵敏度分析，即给出不同参数下预测准确率，AUC-score. 3. 与相关的正标签与无标签算法进行比较. 4. 与监督学习中的梯度提升决策树算法进行比较.

3.1.1 数据集描述

云从数据集[1] 为一个有 15281 条观测数据，68 个变量的银行信贷数据. 其中，标签为 'label' 的变量为响应变量 $y \in \{-1, 1\}$，12587 条 $y = 1$ 的观测值表示信誉好的 "白客户"，2694 条 $y = -1$ 的观测值表示信誉差的 "黑客户" . 剩余 67 个变量为解释变量，其中包括 "年龄"，"性别"，"学历"，"婚姻"，"居住地" 等等. 本文随机选取 70% 的数据作为训练集，其余 30% 作为测试集，通过最小化训练集上的目标函数，来预测用户是 "白客户" 还是 "黑客户" .

3.1.2 AUC-score

考虑到云从数据集的数据分布不均衡性（正标签数据量远大于负标签数据），在评定一个分类器训练效果是好是坏时，我们不能单纯考虑预测准确率，否则一个全部评定为正标签的分类器也能达到较高的预测准确率. 因而，为了更准确客观地评判分类器预测效果，考虑 AUC-score.

AUC-score 是 ROC 曲线与坐标轴形成的面积，取值范围 $[0, 1]$；AUC-score 越靠近 1，表示预测效果越好，由于 ROC 曲线是一个常用统计量，这里不再赘述.

3.1.3 梯度下降法的收敛性

这部分验证迭代过程中，目标函数是否随迭代次数的增加而减少.

[1] 出于顾客隐私、安全考虑，数据集暂不对外公布. 但可以从https://github.com/xinyiwangjuno/PU-gbdt/tree/master/data中下载 test 数据集，也可自定义数据集进行预测.

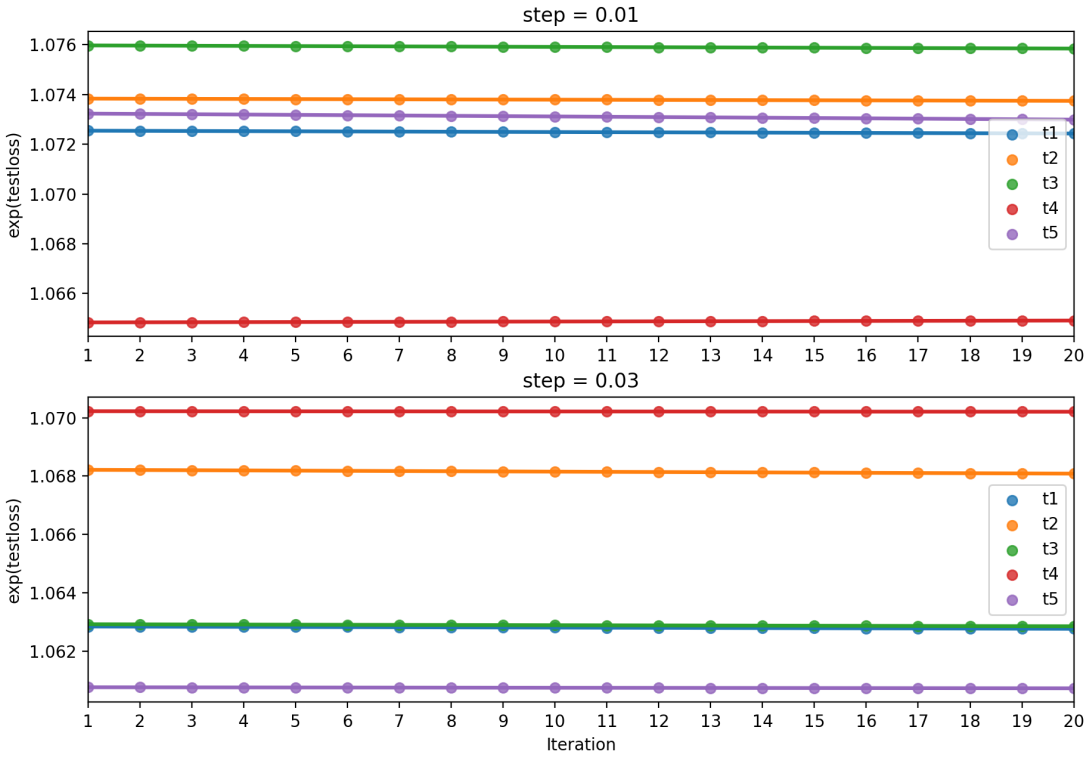

图 3.1: 不同步长下，测试集目标函数的收敛性

可以看出，在迭代过程中，测试集上的目标函数平稳下降，说明该算法是收敛的，但由于搜索步长较小且本身目标函数的取值很小（取了指数后仍在 1 左右徘徊），算法的收敛速度较慢.

3.2 敏感性分析

这部分考虑不同参数下，预测效果的比较. 在原始云从数据集上随机抽取 3000 条观测，70% 作为训练集，剩余 30% 作为测试集.

3.2.1 正标签数据比例

考虑 $p2u_{pro} = \frac{\#\text{ of positive data}}{\#\text{ of unlabeled data}}$ 为 0.1，0.5, 1，5, 10 的情况下，模型的准确率和 AUC-score.

表 3.1: 不同正标签数据比例下的预测结果

$p2u_{pro}$[a]	AUC-score	Train-Loss	Test-Loss	Accuracy(%)
0.1	0.7303	0.0332	0.0661	80.17
0.5	0.7680	0.0338	0.0765	81.25
1	0.7843	0.0269	0.0698	83.58
5	0.8074	0.0175	0.0709	76.22
10	0.8182	0.0135	0.0753	81.68

[a] 该实验中，$max\ iter = 20, sample\ rate = 0.8, learn\ rate = 0.5, max\ depth = 1, split\ points = 2000$

从上表可以看出，随着正标签数据比例的增大，AUC-score 增大，模型预测效果变好. 这是因为我们定义的目标函数正是正标签数据与估计值函数的 K-L 散度，当正标签数据的比例增大时，正标签数据集向全体训练集靠近，从而目标函数向监督学习的目标函数靠近，预测准确率升高.

3.2.2 最大深度 (Max-depth) 与分裂节点数 (Split-Points)

考虑不同决策树最大深度，不同分裂节点数下的计算下的模型预测效果.
设定 $p2u_{pro} = 0.5$, Max-Iter = 20, Sample-rate = 0.8, Learn-Rate = 0.5,
首先，保持 Max-Depth 不变，改变 Split-Points = 1000，预测结果如下表所示：

表 3.2: 不同分裂节点个数下的预测结果

Split-Points	AUC-score	Accuracy(%)
500	0.7826	79.83
1000	0.8573	83.65
2000	0.7796	82.96

然后保持 Split-Points = 1000 不变，改变 Max-Depth，预测结果如下表所示：

表 3.3: 不同最大深度下的预测结果

Max-Depth	AUC-score	Accuracy(%)
1	0.7557	82.40
2	0.7633	78.35
3	0.8210	87.12

由表 3.2 可以看出，本次实验中最优的分裂节点为 1000，预测效率最高，当分裂节点数达到 2000 时，可能存在过拟合的问题，导致预测准确率下降；从表 3.3 看出，本次实验中最优的最大深度值为 3. 本文进一步尝试了最大深度大于 3 的情况，效果不理想且计算时间过长.

3.3 与有监督学习的比较

本节考虑将正标签与无标签学习的梯度提升决策树（PU-GBDT）与监督学习中的梯度提升决策树（GBDT），Xgboost，随机森林（RF）进行比较. 其中，监督学习中的梯度提升决策树用 Negative Binomial Log-likelihood(Jerome H. Friedman, 2001) 作为目标函数，其他监督学习算法用 Logistic 函数作为目标函数.

实验中对四种算法分别进行 10 次随机采样，计算其 AUC-score 和准确率（Accuracy）的均值与方差，结果如下：

表 3.4: 与监督学习算法比较，$p2u_{pro} = 1$

Algorithm	AUC-score	Accuracy(%)
PU-GBDT[a]	0.7814 ± 0.0315	81.48 ± 3.24
GBDT	0.7568 ± 0.0320	82.77 ± 2.45
Xgboost	0.7848 ± 0.0367	84.46 ± 2.74
RF	0.7628 ± 0.0313	83.04 ± 1.702

[a] 该实验中，$p2u_{pro} = 1$，即正数据与无标签数据个数相同.

由上表可以看出，PU-GBDT 的预测效果和监督学习中分类器的效果相差不大，尤其是它的 AUC-score 为 0.7814 ± 0.0315，这对于本文中分布不均衡的数据集来说，是有比较好的现实意义的.

下图给出每次计算过程中，不同算法的准确率：

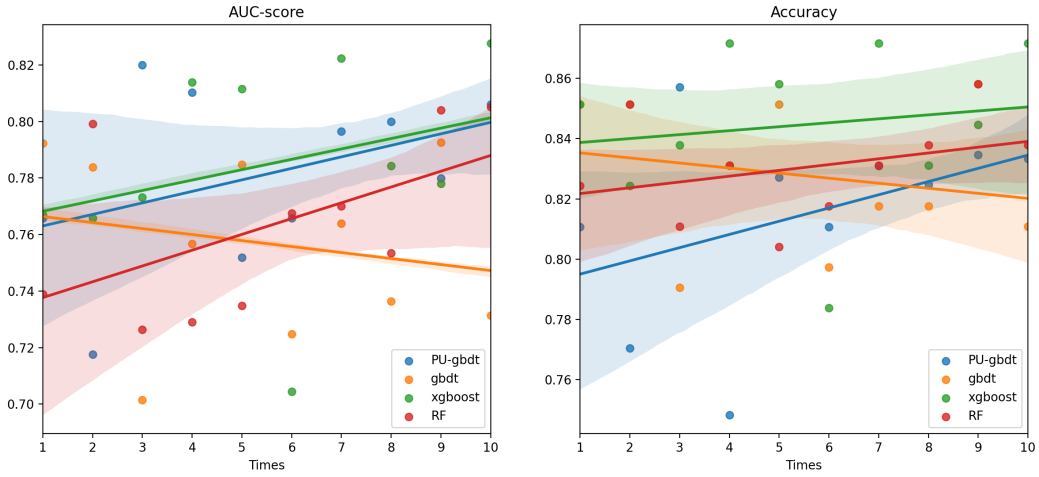

图 3.2: 不同算法中，测试集目标函数的 AUC-score 和正确率

就 AUC-score 而言，算法 Xgboost 的预测效果最好，其次是本文构造的 PU-gbdt，并且 PU-gbdt 的预测效果与有监督算法的差别不大，说明 PU-gbdt 对于预测该信贷数据集的效果还是比较理想的. 就准确率（Accuracy）而言，PU-gbdt 的准确率略逊于 Xgboost 和 Random Forest，但由于我们预测的数据集数据不均衡性较大，准确率（Accuracy）的参考价值没有 AUC-score 大.

4 总结与未来工作展望

本文解决正数据与无标签数据学习（PU-Learning）中的信贷风险问题，从 K-L 散度出发，建立目标函数，再利用梯度提升决策树，构建了一个 PU 分类器. 分类器对于信贷数据的预测效果良好，且逼近有监督学习中的分类器.

未来研究方向包括以下几点：第一，是否可以考虑 Xgboost 作为分类器，本文中的 GBDT 是将目标函数的一阶 Taylor 展开（即梯度）作为残差，而可以借鉴 Xgboost 的想法，对目标函数进行二阶 Taylor 展开，提高残差估计的准确率. 第二，由于数据量过大，实验中耗时过长，可以考虑先对数据集进行降维，再进行预测. 第三，本文对其他 PU 分类器提及较少，可以深化对其他 PU 分类器的学习.

参考文献

[1] Emanuele Sansone, Francesco G. B. De Natale, Senior Member, IEEE and Zhi-Hua Zhou, Fellow, IEEE. Efficient Training for Positive Unlabeled Learning. arXiv:1608.06807v4, 2018.

[2] Hui Chen, Fangqing Liu, Yin Wang, Liyue Zhao, Hao Wu. A Variational Approach for Learning from Positive and Unlabeled Data. arXiv:1906.00642v5, 2020.

[3] C. Elkan and K. Noto. Learning Classifiers from Only Positive and Unlabeled Data. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008:213–220.

[4] M. Du Plessis, G. Niu, and M. Sugiyama. Analysis of Learning from Positive and Unlabeled Data. NIPS, 2014:703–711.

[5] M. Du Plessis, G. Niu, and M. Sugiyama. Convex Formulation for Learning from Positive and Unlabeled Data. ICML, 2015:1386–1394.

[6] J.T.Zhou, S.J.Pan, Q.Mao, and I.W.Tsang. Multi-View Positive and Unlabeled Learning. ACML, 2012:555–570.

[7] T. Sakai, M. C. d. Plessis, G. Niu, and M. Sugiyama. Beyond the Low-Density Separation Principle: A Novel Approach to Semi- Supervised Learning. arXiv preprint arXiv:1605.06955, 2016.

[8] X. Li, S. Y. Philip, B. Liu, and S.-K. Ng. Positive Unlabeled Learning for Data Stream Classification. SDM, vol. 9. SIAM, 2009:257–268.

[9] H. Yu, J. Han, and K. C.-C. Chang. PEBL: Positive Example Based Learning for Web Page Classification Using SVM. International Conference on Knowledge Discovery and Data Mining. ACM, 2002:239–248.

[10] Friedman, Jerome. Multiple Additive Regression Trees with Application in Epidemiology. Statistics in Medicine, 2003, 22 (9):1365–1381.

[11] Friedman, Jerome H. Greedy function approximation: a gradient boosting machine. Annals of Statistics, 2001, 29 (5):1189–1232.

[12] Du Plessis, M., Niu, G., and Sugiyama, M. Convex formulation for learning from positive and unlabeled data. International Conference on Machine Learning, 2015:1386–1394.

[13] Du Plessis, M. C., Niu, G., and Sugiyama, M. Class-prior estimation for learning from positive and unlabeled data. Machine Learning, 2017,106(4):463–492.

[14] Liu, Z., Shi, W., Li, D., and Qin, Q. Partially supervised classification: based on weighted unlabeled samples support vector machine. Data Warehousing and Mining: Concepts, Methodologies, Tools, and Applications. IGI Global, 2008:1216-1230.

[15] G. Blanchard, G. Lee, and C. Scott. Semi-Supervised Novelty Detection. Journal of Machine Learning Research, 2010-11(11):2973–3009.

[16] M. N. Nguyen, X.-L. Li, and S.-K. Ng. Positive Unlabeled Leaning for Time Series Classification. IJCAI, 2011-11:1421–1426.

[17] H. Yu. Single-Class Classification with Mapping Convergence. Machine Learning, 2005, 61(1-3):49–69.

[18] B. Liu, Y. Dai, X. Li, W. S. Lee, and P. S. Yu. Building Text Classifiers Using Positive and Unlabeled Examples. Third IEEE International Conference on Data Mining. IEEE, 2003:179–186.

装 订 线

[19] A. Skabar. Single-class classifier learning using neural networks: An application to the prediction of mineral deposits. Machine Learning and Cybernetics, 2003 International Conference on, vol. 4. IEEE, 2003:2127–2132.

[20] G. Blanchard, G. Lee, and C. Scott. Semi-Supervised Novelty Detection. Journal of Machine Learning Research, 2010-11(11):2973–3009.

[21] Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). The annals of statistics, 2000-28(2):337–407.

[22] Charles Dubout and François Fleuret. Boosting with maximum adaptive sampling. Advances in Neural Information Processing Systems, 2011:1332–1340.

[23] Ron Appel, Thomas J Fuchs, Piotr Dollár, and Pietro Perona. Quickly boosting decision trees-pruning underachieving features early. ICML(3), 2013:594–602.

[24] Sanjay Ranka and V Singh. Clouds. A decision tree classifier for large datasets. Proceedings of the 4th Knowledge Discovery and Data Mining Conference, 1998:2–8.

[25] Ping Li, Christopher JC Burges, Qiang Wu, JC Platt, D Koller, Y Singer, and S Roweis. Mcrank. Learning to rank using multiple classification and gradient boosting. NIPS, 2007-7:845–852.

[26] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016:785-794.

[27] Greg Ridgeway. Generalized boosted models: A guide to the gbm package. Update, 1(1),2007.

谢辞

本论文在吴昊导师的悉心指导下完成的。导师渊博的专业知识、严谨的治学态度，精益求精的工作作风对本人影响深远。

还想感谢帮助我的同学陈晖，师兄何任飞，在做毕设过程中遇到了很多困难，感谢你们的帮助支持。

A PU-gbdt 代码

由于篇幅问题，详见https://github.com/xinyiwangjuno/PU-gbdt