基于 Ext-GBDT 集成的类别不平衡信用评分模型

作者	陈启伟，王伟，马迪，毛伟
机构	中国科学院大学；中国科学院计算机网络信息中心；北龙中网(北京科技有限责任公司

发表期刊	《计算机应用研究》
预排期卷	2018 年第 35 卷第 2 期
访问地址	http://www.arocmag.com/article/02-2018-02-031.html
发布日期	2017-03-15 09:18:22
引用格式	陈启伟，王伟，马迪，毛伟. 基于 Ext-GBDT 集成的类别不平衡信用评分模型[J/OL]. [2017-03-15]. http://www.arocmag.com/article/02-2018-02-031.html.

摘要	针对现实信用评分业务中样本类别不平衡和代价敏感问题，以及金融机构更期望以得分的方式直观认识贷款申请人的信用风险的实际需求。提出一种基于 Ext-GBDT 集成的类别不平衡信用评分模型。使用欠采样的方法从“好”客户（大类）中随机采样多份与全部“坏”客户（小类）等量的样本，分别和全部小类构成训练子集；用不同的训练子集及特征采样和参数扰动的方法训练得到多个差异化的 Ext-GBDT 子模型；然后使用简单平均法整合子模型的预测概率；最后将信用概率转换为信用评分。在 UCI 德国信用数据集上，以 AUC 和代价敏感错误率…

关键词	信用评分，类别不平衡，代价敏感，Ext-GBDT，集成学习
中图分类号	TP391
基金项目	

基于 Ext-GBDT 集成的类别不平衡信用评分模型

陈启伟[1,2,3]，王 伟[2,3]，马 迪[3]，毛 伟[2,3]

(1．中国科学院大学，北京 100190；2．中国科学院计算机网络信息中心，北京 100190；3．北龙中网(北京)科技有限责任公司，北京 100190)

摘 要：针对现实信用评分业务中样本类别不平衡和代价敏感问题，以及金融机构更期望以得分的方式直观认识贷款申请人的信用风险的实际需求。提出一种基于 Ext-GBDT 集成的类别不平衡信用评分模型。使用欠采样的方法从“好”客户（大类）中随机采样多份与全部“坏”客户（小类）等量的样本，分别和全部小类构成训练子集；用不同的训练子集及特征采样和参数扰动的方法训练得到多个差异化的 Ext-GBDT 子模型；然后使用简单平均法整合子模型的预测概率；最后将信用概率转换为信用评分。在 UCI 德国信用数据集上，以 AUC 和代价敏感错误率作为评价指标，与决策树、逻辑回归、朴素贝叶斯、支持向量机、随机森林及其集成模型等当前最为常用的信用评分模型进行对比，验证了该模型的有效性。

关键词：信用评分；类别不平衡；代价敏感；Ext-GBDT；集成学习

中图分类号：TP391

Class-imbalance credit scoring using ext-gbdt ensemble

Chen Qiwei[1,2,3], Wang Wei[2,3], Ma Di[3], Mao Wei[2,3]

(1. University of Chinese Academy of Science, Beijing 100190, China; 2. Computer Network Information Center, Chinese Academy of Science, Beijing 100190, China; 3. Knet Co. , Ltd, Beijing 100190, China)

Abstract: In view of class-imbalance and cost-sensitive problem in real credit scoring business, as well as financial institutions prefer to assess credit risk of the loan applicant in an intuitive way, this paper proposed an Ext-GBDT ensemble model for class-imbalance credit score. In this proposed model, firstly, it adopts an under-sampling method to randomly samples several subsets from credible customer (the majority class) and then combines each of them with default one (the minority class) for generating several class-balance training subsets. Secondly, it employs different training subsets as well as feature sampling and parameter disturbance method to train several diverse Ext-GBDT models. After that, it integrates the predicted result of different models by using the simple average method. Finally, it transforms credit probability into credit scoring. In terms of AUC and cost-sensitive error rate, this model against five well-known credit scoring models and their ensemble model on UCI German credit dataset and the research results reveal the validity of the proposed method.

Key Words: credit scoring; class-imbalance; cost-sensitive; Ext-GBDT; ensemble learning

0 引言

随着普惠金融的发展和社会大众消费观念的改变，信贷业务快速发展。而信贷的核心，仍然是如何高效地解决信息不对称问题，从而进行有效的风险管理。在信贷风险管理方面，信用评分技术发挥着重要的作用。如何构建一个可靠的信用评分模型来评估贷款申请人的信用风险，已经成为学术界和商业界重要的研究课题。

信用评分方法的原理是根据贷款申请人的基本信息和过去的表现来建立信用评分模型，并用该模型对具有相同特征的未来申请者的信用进行预测，从而协助放贷机构作决策。良好的信用评分模型可以减少放贷机构的放贷成本，减少不良贷款带来的损失，节省时间而提高放贷效率。信用评分模型的评分效果对放贷机构的利润影响较大，评估的准确率仅仅提高 1% 都能给放贷机构挽回巨大的损失[1]。

在过去的数十年，已经有很多学者使用基于统计和机器学习的方法来构建信用评分模型。常见的方法有：线性判别分析（linear discriminant analysis, LDA）、逻辑回归（logistic regression, LR）、决策树（decision trees, DT）、朴素贝叶斯（naïve Bayes, NB）、神经网络（neural network, NN）、支持向量机

作者简介：陈启伟（1992-），男，广东汕头人，硕士研究生，主要研究方向为数据挖掘、机器学习、信用风险（chenqiwei5@qq.com）；王伟（1977-），男，江苏镇江人，教授级高工，博士，主要研究方向为 DNS 协议、网络安全、数据分析；马迪（1984-），男，安徽六安人，高工，博士，主要研究方向为互联网资源寻址定位技术以及互联网基础资源管理技术；毛伟（1968-），男，四川自贡人，研究员，博士，主要研究方向为下一代互联网、资源寻址与定位

（support vector machine, SVM）、随机森林（random forest, RF）、提升树（boosting trees, BT），部分相关的研究工作如表 1 所示。然而对于这些单模型孰好孰坏，不同文献有不同的意见；为此文献[2]针对该问题展开了研究，在 UCI 公开信用数据集上对比了之前工作中提出的各种模型，实验结果表明在这些单模型中提升树能取得比较好的信用评分效果，而线性判别分析和决策树取得的评分效果较差。

近几年，对信用评分问题的研究集中于采用集成学习的方法（部分相关的研究工作如表 1 所示），这些研究都表明：在信用评分问题上，集成学习模型比单模型更准确。在集成学习方法中，核心的问题就是让基分类器“好而不同”，即尽可能提高基分类器的精度和多样性[9]。当前大多研究使用逻辑回归、支持向量机作为集成学习的基分类器[9,10,11,12]，然而逻辑回归等线性分类器和支持向量机都是“稳定分类器”，即对样本扰动不敏感，难以通过样本扰动构建多样的基分类器。也有少部分研究使用多种分类器组成异构的集成学习模型[12,14]，然而不同类型的分类器输出的预测概率的量纲不同，不能直接求平均，只能通过投票法将信用评分问题当作二类分类问题，而无法获得每个贷款申请人的信用概率。Ext-GBDT(extreme gradient boosting decision trees)是一种增强的提升树模型，具备提升树良好的性能，同时，树模型是一种样本扰动的“不稳定分类器”，很容易通过样本扰动来构建多样的分类器，Ext-GBDT 非常适用作为集成学习的基分类器。

表 1 信用评分技术

分类技术	相关文献
线性判别分析 (LDA)	[2], [3], [4], [6], [7], [11]
逻辑回归 (LR)	[2], [4], [5], [6], [7], [10], [11]
决策树 (DT)	[5], [6], [7], [10], [11]
朴素贝叶斯 (NB)	[12]
神经网络 (NN)	[2], [3], [4], [5], [6], [7], [11]
支持向量机 (SVM)	[2], [7], [8], [10], [11]
随机森林 (RF)	[2], [11]
提升树 (BT)	[2], [11]
逻辑回归集成 (LR Ensemble)	[12]
决策树集成 (DT Ensemble)	[10], [11]
朴素贝叶斯集成 (NB Ensemble)	[11]
神经网络集成 (NN Ensemble)	[9], [10], [11]
支持向量机集成 (SVM Ensemble)	[9], [10], [11]
随机森林集成 (RF Ensemble)	[9]

应用信用评分模型解决的实际问题有两大不容忽视的特点：一是在实际的信贷业务中，很多可能违约的样本在前期的筛选中就被直接拒绝了，导致前期收集到的数据集中可信的“好”客户与违约的“坏”客户的数量不同，“好”客户的数量比“坏”客户多。二是把“坏”客户误分类为“好”客户和把“好”客户误分类为“坏”客户的代价是不同的，误分类“坏”客户的代价更大，应该尽可能避免这样的错误。因此，构建信用评分模型的过程是一个类别不平衡和代价敏感的学习问题。相比于信用评分的大量研究，很少有文献考虑到类别不平衡和代价敏感的问题。采样和代价敏感学习是处理类不平衡问题的常见方法。代价敏感学习要求事先知道错分的代价[15]，而这个代价在实际的信贷业务中是难以评估的，因此代价敏感学习难以应用到信用评分问题中，在实际业务中更多采用采样的方法。文献[2]中使用了简单的“欠采样”和“过采样”方法来处理信用评分中的类别不平衡问题。然而简单的“欠采样”随机丢失了样本，可能丢失一些重要的信息，而简单的“过采样”直接对小类样本进行复制，容易造成过拟合，使用这两种方法来处理信用评分中的类别不平衡问题均不能取得理想的效果。文献[15]使用 SMOTE 算法来产生违约样本，取得了比简单“欠采样”和“过采样”相对更好的效果，然而该算法人为的产生过多的样本，引入了过多的主观因素。文献[16]提出了应对类别不平衡问题的 EasyEnsemble 算法，利用集成学习机制，将大类划分为若干个子集，分别将这些子集和小类构成训练样本来训练多个基分类器，这样每个基分类器都是使用欠采样的方法，但全局却没有丢失重要信息，该方法比 SMOTE 跟简单，效率更高。

很多分类问题中常使用准确率来评价分类器的性能，然而准确率存在不重视小类对分类性能评测的影响的问题[17]。假设有一个训练数据正负样本比为 99:1 的两类分类问题，即使分类器简单的把所有样本都分为正样本，它仍能达到 99%的分类正确率，这显然不太合理。AUC(Area Under the ROC Curve)[18]是一种独立于类别分布的评价指标，比较适合用于评价类别不平衡学习问题[19]。此外，在实际的信贷业务中，将“好”客户误分类为“坏”客户和将“坏”客户误分类为“好”客户这两种情况的代价是不同的，在评价信用评分模型的效果时，还应该使用代价敏感的评价方法。

基于现实信用评分业务中样本类别不平衡和代价敏感的问题，考虑到 Ext-GBDT 作为一种强化的提升树模型，具有较强的学习能力和泛化能力，而树模型的本质让其容易作为集成学习的基分类器构造多样的子模型，即能构造“好而不同”的子模型，从而提高集成模型的分类精度，本文提出了一个基于 Ext-GBDT 集成的类别不平衡信用评分模型。模型先使用“欠采样”的方法从全部可信的“好”客户（大类）中随机地采样多份与全部“坏”客户（小类）等量的样本，分别与全部小类构成训练数据集；然后使用不同的训练子集及特征采样和参数扰动的方法来训练多个差异化的 Ext-GBDT 分类器；接着将每个基分类器得到的预测概率简单平均得到最终的预测概率，并根据阈值将概率转换为分类结果。使用 AUC 和代价敏感错误率作为评价指标，在 UCI 信用评分数据集上测试该模型并与决策树、逻辑回归、朴素贝叶斯、支持向量机、随机森林及其集成模型等当前最为常用的信用评分模型进行对比，结果表现本文提出的模型有更高的 AUC 和更低的代价敏感错误率。

然而，很多信用评分模型仅把信用评分当作一个二类分类问题，只是简单给出了好坏客户的分类；而对于放贷机构，他们更希望的是能以一种更直观的方式认识每一个贷款申请者的信用情况，以指导其更好地制定放贷策略。本文模型输出的是

贷款申请人的信用概率，并进一步使用总体转换法将信用概率转换为一个直观的信用分数，更具有实践价值。最后本文使用金融机构中常用的坏账率[20]和 K-S value (Klmogrov-Smirnov) [21]来评估本文提出的模型在现实信贷业务中的可行性。

1 Extreme Gradient Boosting 和 Ext-GBDT

1.1 Extreme Gradient Boosting

Extreme Gradient Boosting[22]是一种大规模、灵活和分布式的 Gradient Boosting 模型，它是由 Chen Tianqi 基于 Friedman 提出的 Gradient Boosting[23]模型设计的。

已知一个训练数据集 $D=\{(x_1,y_1),(x_2,y_2),\dots,(x_n,y_n)\},x_i\in\mathcal{X}\subseteq\mathbb{R}^m,\mathcal{X}$ 为输入空间，$y_i\in\mathcal{Y}\subseteq\mathcal{R},\mathcal{Y}$为输出空间。则 Extreme Gradient Boosting 可以表示为加法模型：

$$\hat{y}_i=\sum_{k=1}^{K}f_k(x_i)\quad(1)$$

其中：$f_k(x_i)$表示第 k 个子模型，K 为子模型的个数。

为了学习到模型(1)，本文需要最小化以下的正则化目标函数：

$$\mathcal{L}=\sum_i l(\hat{y}_i,y_i)+\sum_k\Omega(f_k)\quad(2)$$

其中，l是损失函数。

对于加法模型的学习，本文采用向前分步算法。首先确定初始提升树$f_0(x)=0$，第 t 步的模型为：

$$\mathcal{L}^{(t)}=\sum_{i=1}^{n}l\left(y_i,\hat{y}_i^{(t-1)}+f_t(x_i)\right)+\Omega(f_t)\quad(3)$$

使用 MSE 损失作为式(3)的损失函数，并进行二阶泰勒展开，得：

$$\mathcal{L}^{(t)}\simeq\sum_{i=1}^{n}\left[l(y_i,\hat{y}_i^{(t-1)})+g_if_t(x_i)+\frac{1}{2}h_if_t^2(x_i)\right]+\Omega(f_t)\quad(4)$$

其中：

$$g_i=\partial_{\hat{y}^{(t-1)}}l(y_i,\hat{y}^{(t-1)})$$
$$h_i=\partial^2_{\hat{y}^{(t-1)}}l(y_i,\hat{y}^{(t-1)})$$

去除式(4)的常数项，化简为：

$$\tilde{\mathcal{L}}^{(t)}=\sum_{i=1}^{n}\left[g_if_t(x_i)+\frac{1}{2}h_if_t^2(x_i)\right]+\Omega(f_t)\quad(5)$$

则最小化式(5)就可以求得第 t 个子模型的参数，用$\tilde{\Theta}_t$表示第 t 个子模型的参数，则：

$$\tilde{\Theta}_t=\underset{\Theta_t}{\operatorname{argmin}}\,\tilde{\mathcal{L}}^{(t)}$$

1.2 Ext-GBDT

在本文中，本文选用 CART 回归树[24]作为子模型的结构，并称其为 Ext-GBDT。

此时，本文学习的目标是对每个子模型，寻找最优的 CART 回归树，使得式（2）最小。

首先本文使用树结构的不纯度来评价树结构的好坏，树结构的不纯度越小越好。树结构的不纯度表示如下：

本文使用如下的正则化函数：

$$\Omega(f_t)=\gamma T+\frac{1}{2}\lambda\|w\|^2$$

由式(5)，第 t 棵子树可以表示为：

$$\tilde{\mathcal{L}}^{(t)}=\sum_{i=1}^{n}\left[g_if_t(x_i)+\frac{1}{2}h_if_t^2(x_i)\right]+\gamma T+\frac{1}{2}\lambda\|w\|^2$$
$$=\sum_{j=1}^{T}\left[\left(\sum_{i\in I_j}g_i\right)w_j+\frac{1}{2}\left(\sum_{i\in I_j}h_i+\lambda\right)w_j^2\right]+\gamma T$$

其中，I_j为树中第 j 个叶子节点的样本集合，T 为树的叶子节点的个数，w_j为第 T 个叶子节点的值。

给定第 t 棵子树的一种结构 q(x),则其不纯度可以表示为：

$$\tilde{\mathcal{L}}^{(t)}(q)=-\frac{1}{2}\sum_{j=1}^{T}\frac{\left(\sum_{i\in I_j}g_i\right)^2}{\sum_{i\in I_j}h_i+\lambda}+\gamma T\quad(6)$$

当然，本文不可能穷举所有的树结构，然后选择其中不纯度最小的那种树结构作为最优的树结构。CART 回归树的学习使用一种贪心的方法，不断寻找当前最优的切分变量和切分值将输入空间切分。因此学习 CART 回归树的核心问题就是如果找到这些切分变量和切分值。这里使用启发式的方法，选择第 j 个特征$x^{(j)}$和它的取值 s，作为切分变量和切分点，并定义两个区间：

$$I_L=\{x|x^{(j)}\le s\}\ \text{和}\ I_R=\{x|x^{(j)}\ge s\}$$

则有$I=I_L\cup I_R$

然后通过求解式(7)：

$$\Theta(s,j)=\underset{s,j}{argmax}\left[\frac{1}{2}\left[\frac{\left(\sum_{i\in I_L}g_i\right)^2}{\sum_{i\in I_L}h_i+\lambda}+\frac{\left(\sum_{i\in I_R}g_i\right)^2}{\sum_{i\in I_R}h_i+\lambda}-\frac{\left(\sum_{i\in I_j}g_i\right)^2}{\sum_{i\in I_j}h_i+\lambda}\right]\right]\quad(7)$$

来寻找最优的切分变量 s 和切分点 j。

为了防止过拟合，若最优的 s 和 j 对应的$\frac{1}{2}\left[\frac{\left(\sum_{i\in I_L}g_i\right)^2}{\sum_{i\in I_L}h_i+\lambda}+\frac{\left(\sum_{i\in I_R}g_i\right)^2}{\sum_{i\in I_R}h_i+\lambda}-\frac{\left(\sum_{i\in I_j}g_i\right)^2}{\sum_{i\in I_j}h_i+\lambda}\right]$值小于$\gamma$，则不对节点进行切分。

使用贪心方法寻找最优的切分变量和切分点的算法描述如下：

算法 1： 贪心法寻找节点的切分变量和切分点

输入：当前节点的样本集 I

特征集 D,特征维度 d

过程：

1:$G\leftarrow\sum_{i\in I}g_i$,$H\leftarrow\sum_{i\in I}h_i$

2:**for** $k=1$ **to** d **do:**

3: $G_L\leftarrow0,H_L\leftarrow0$

4: **for** j **in** $sorted(I,by\ x_{kj})$ **do:**

5: $G_L\leftarrow G_L+g_j,\ H_L\leftarrow H_L+h_j$

6: $G_R\leftarrow G-G_L,\ H_R\leftarrow H-H_L$

7: $score\leftarrow max(score,\frac{G_L^2}{H_L+\lambda}+\frac{G_R^2}{H_R+\lambda}-\frac{G^2}{H+\lambda})$

8: end for

9:end for

输出：最大的 *score* 对应的切分变量 *s* 和点 *j*

2 类别不平衡的 Ext-GBDT 集成模型

本章介绍本文所提出的基于 Ext-GBDT 集成的类别不平衡

信用评分模型的设计。模型主要包括三个部分：a)使用欠采样的方法从大类中随机地采样多份与小类等量的样本，分别与全部小类构成训练数据子集；b)使用不同的训练子集及特征采样和参数扰动的方法来训练多个差异化的 Ext-GBDT 子模型；c)将每个子模型得到的预测概率简单平均得到最终的预测概率。模型的整体结构如图 1 所示。

2.1 产生训练集子集

给定一个大类训练集$\mathbb{P}$和小类训练集$\mathbb{N}$，使用欠采样的方法从大类训练集中随机采样 T 个大类的子集$P_1', P_2', \ldots, P_T'$（采样每个子集时使用不放回采样），使得$|P_i'|=|\mathbb{N}|$，i=1,2,…,T。将 T 份大类的子集分别和全部小类训练集合成 T 份训练子集$D_1, D_2, \ldots, D_T$。这样每个训练子集都是由欠采样产生的，但在全局来看却不会丢失重要信息。

2.2 训练不同的 Ext-GBDT 基分类器

根据文献[25]，创建一个好的 Bagging 集成分类器的一个充分必要条件为：用于组合的基分类器必须是“好而不同”。即基分类器应是尽可能准确并且基分类器之间应尽可能有多样性。对于如何构建多样性的基分类器，常见的做法有以下 4 种：

a) 数据样本扰动。给定初始训练集，从中采样产生不同的训练子集，利用不同的训练子集训练出不同的基分类器。

b) 输入属性扰动。训练数据通常由一组属性描述，从不同的“子空间”提供了观察数据的不同视角。因此，从不同子空间训练出的个体分类器必然有所不同。

c) 基分类器类型扰动。不同的基分类器本身就不同。

d) 算法参数扰动。很多分类算法都有设置参数，不同的参数会产生有差异的分类器。

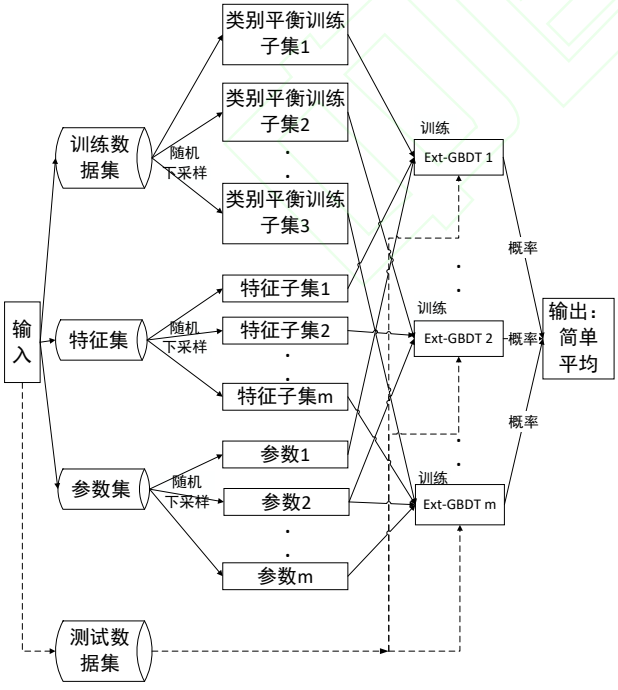

图 1 基于 Ext-GBDT 集成的类别不平衡信用评分模型整体结构

本文选取 Ext-GBDT 作为基分类器，使用样本扰动、输入属性扰动和算法参数扰动相结合的方法来训练多样性的 Ext-GBDT 分类器。Ext-GBDT 是基于提升树模型设计的，而提升树被认为是机器学习性能最好的算法之一[26]，准确率较高。前一步的工作已经产生足够多不同的训练数据，而 Ext-GBDT 是一种树模型，相对于线性分类器和 SVM 等“样本扰动稳定分类器”，树模型是“样本扰动不稳定分类器”，使用样本扰动的方法能让基分类器的有显著的不同。Ext-GBDT 学习算法有较多的参数可以设置，本文通过扰动算法梯度下降的学习率、树的最大深度、提升树迭代次数等参数进行参数扰动。最后再结和属性扰动，多种扰动机制结合，使得基分类器之间的差异更大。

2.3 基分类器集成

根据前面两步工作，可以得到一组不同的 Ext-GBDT 基分类器。接下来的工作就是要使用一个恰当的组合策略将不同的基分类器集成为一个分类器。为了后续将概率转换为信用评分，本文设计的基分类器输出的结果是每个样本属于正类的概率。在二分类问题中，可以通过设置一个阈值，将概率大于阈值的样本分为正类，概率小于阈值的样本分为负类。对于概率的集成策略，常采用的有简单平均法和加权平均法。文献[27,28]对比显示了，加权平均法未必优于简单平均法，而且在基学习器类型相近时宜使用简单平均法。因此，本文使用简单平均法来集成基础学习器。

给定样本$\mathcal{X} = (x_1, x_2, \ldots, x_n)$，$x_i \in \mathbb{R}^m$，集成模型中有 T 个基分类器$\delta_1(x), \delta_2(x), \ldots, \delta_T(x)$，则对于每一个样本$x_i$，其集成输出的结果为：

$$H(x) = \frac{1}{T}\sum_{i=1}^{T} h_i(x)$$

2.4 算法描述

类别不平衡 Ext-GBDT 集成模型的算法描述如下：

算法 2： 类别不平衡的 Ext-GBDT 集成算法

输入：一个小类数据集N，一个大类数据集P，$|N| < |P|$；

基分类器的个数 T；

基分类器学习率扰动的学习率范围 *ETA*，树深度范围 *MAX_DEPTH*，迭代次数范围 *ROUND*，特征采样比例范围 *COL*。

过程：

1：**for** $i=1,2,\ldots,T$ **do**

2：　从P中随机不放回采样一个子集P_i'，$|P_i'| = |N|$；

3：　从 *ETA* 中随机取一个值 *eta*；

4：　从 MAX_DEPTH 中随机取一个值 max_depth；

5：　从 *ROUND* 中随机取一个值 *round*；

6：　从 *COL* 中随机取一个值 *col*；

7：　将$P_i' \cup N$作为训练集，*eta*、*max_depth*、*round*、*col* 作为参数训练一个 Ext-GBDT 分类器$h_i(x)$；

8：end for

输出： $H(x) = \frac{1}{T}\sum_{i=1}^{T} h_i(x)$

2.5 程序实现

使用 Python 编程语言实现的基于 Ext-GBDT 集成的类别不平衡信用评分模型的程序代码及使用说明可以从(https://github.com/cqw5/CreditScoring/tree/master/UnExtGBDTE

nsemble) 获取。

其中训练 Ext-GBDT 子模型、训练类别不平衡的 Ext-GBDT 集成模型和利用模型对新样本的信用概率进行预测的核心程序如下。

程序 1：训练 Ext-GBDT 子模型

```
def ExtGBDT(model_id, train_x, train_y, num_round, eta, max_depth, colsample_bytree):
    """
    :param model_id: int, 子模型 id
    :param train_x: ndarray, 训练数据的 feature
    :param train_y: ndarray, 训练数据的类标签
    :param num_round: 模型迭代次数
    :param eta: double, 学习率
    :param max_depth: int, 树的深度
    :param colsample_bytree: double, 特征采样比
    :return: predict_y: ndarray, 预测结果
    """
    param = {'objective': 'binary:logistic', 'booster': 'gbtree', 'eta': eta, 'max_depth': max_depth, 'eval_metric': 'auc', 'silent': 1, 'min_child_weight': 0.1, 'subsample': 0.7, 'colsample_bytree': colsample_bytree, 'nthread': 4}
    # 训练子模型
    train_X = xgb.DMatrix(train_x, train_y)
    bst = xgb.train(param, train_X, num_round)
    # 将子模型的二进制文件保存
    if not os.path.exists('../model'):
        os.makedirs('../model')
    model_file = '../model/model' + str(model_id)
    pickle.dump(bst, open(model_file, 'w'))
```

程序 2 训练类别不平衡的 Ext-GBDT 集成模型

```
def ExtGBDTEnsembleTrain(sub_clf_num, train_good, train_bad):
    """
    :param sub_clf_num: int, 子模型的个数
    :param train_good: list, 训练数据好客户样本
    :param train_bad: list, 训练数据坏客户样本
    """
    num_train_good = len(train_good)
    num_train_bad = len(train_bad)
    eta_list = [0.01, 0.02, 0.03]  # 学习率 eta 的扰动
    max_depth_list = [5, 6, 7]  # 树深度的扰动
    colsample_bytree_list = [0.7, 0.8]  # 属性采样比的扰动
    round_num_list = range(100, 200) # 子模型迭代次数扰动
    for model_id in range(sub_clf_num): # 训练每一个子模型
    # 从大类好客户中采样与全部坏客户等量的样本
    train_good_sample_id = random.sample(range(num_train_good), num_train_bad)
        train_good_sample = [train_good[i] for i in train_good_sample_id]
        train = train_good_sample + train_bad
        random.shuffle(train)
        train_x, train_y = seprate_xy(train)
        # 参数采样
        round_num = random.choice(round_num_list)
        eta = random.choice(eta_list)
        max_depth = random.choice(max_depth_list)
        colsample_bytree = random.choice(colsample_bytree_list)
        ExtGBDT(model_id, train_x, train_y, round_num, eta, max_depth, colsample_bytree)
```

程序 3 利用模型对新样本的信用概率进行预测

```
def ExtGBDTEnsemblePredict(sub_clf_num, predict_x):
    """
    :param sub_clf_num: int 子模型的个数
    :param predict_x: list, 待预测数据的 feature
    :return: socre: ndarray, 预测结果
    """
    total_score = np.zeros(len(predict_x))  #保存所有子模型结果

    for i in range(sub_clf_num):
        # 读取子模型二进制文件，用子模型进行预测
        predict_X = xgb.DMatrix(predict_x)
        model_file = '../model/model' + str(i)
        bst = pickle.load(open(model_file, 'r'))
        predict_y = bst.predict(predict_X)
        total_score += predict_y
    score = total_score / sub_clf_num  # 所有子模型结果平均
    return score  # 返回新样本的预测结果
```

3 实验分析

为了验证本文提出的基于 Ext-GBDT 集成的类别不平衡信用评分模型的有效性，本章在 UCI 德国信用数据集上比较了本文模型与当前常见信用评分模型的 AUC 和代价敏感错误率。

3.1 实验数据集

本文采用 UCI 提供的机器学习公开数据集中的德国信用数据集对本文方法进行验证，数据的获取方式和数据属性的描述见[29]。

德国信用数据集中，采用官方给出数值化后的“german.data-numeric”文件内的数据，该数据被广泛应用于信用评分模型的验证中。它包含 1000 条贷款申请记录，其中 700 条是可信的“好”客户，300 条是违约的“坏”客户。原始的数据由 19 个属性描述，官方给出的数值化后的文件使用“独热编码(one-hot coding)”将其中标称属性转换为虚拟变量，最终一个贷款申请记录由 24 个属性描述。

3.2 数据预处理

在分类器中，支持向量机等基于距离度量的分类模型，对数据间数量级的差别非常敏感，数据间数量级差别较大会造成其分类的误差较大。为了消除数据间数量级差别对基于距离度量的分类模型的分类结果产生影响，使对比实验更有说服力，实验在进行模型训练前先使用最小-最大规范化方法对数据进行规范化。最小-最大规范化的处理方法如下：

$$x_i = \frac{x_i - x_{min}}{x_{max} - x_{min}}$$

3.3 评价指标

对于类别不平衡问题和代价敏感问题，使用准确率、精确率、召回率、错误率这些来评价模型的性能是不恰当的[11,17]。本文本文使用 AUC(Area Under the ROC Curve)[18]和代价敏感错误率来评价模型的性能。

AUC 是 ROC 曲线下方的面积。由于 AUC 是独立于阈值和错分代价的，因此它非常适合用了类别不平衡和代价敏感问题[19]。AUC 的取值在 0 到 1 之间，取值越大，说明模型的性能越好。

代价敏感错误率的定义依赖于表 2 所示的代价敏感矩阵。其中$cost_{ij}$表示将第 i 类样本预测为第 j 类样本的代价。一般来说，$cost_{ii} = 0$。用$cost_{01}$表示将好客户误判为坏客户的代价，$cost_{10}$表示将坏客户误判为好客户的代价，在信用评分问题中，显然有$cost_{01} < cost_{10}$。用 P 表示好客户集，N 表示坏客户集，设$m = |P \cup N|$，则代价敏感错误率为：

$$E(f;cost) = \frac{1}{m} \times \left(\sum_{x_i \in P} I(f(x_i) \neq y_i) \times cost_{01} + \sum_{x_i \in N} I(f(x_i) \neq y_i) \times cost_{10} \right)$$

设$cost_{10}$和$cost_{01}$的代价比为 n，n 是一个经验值，设$cost_{01} = 1$，则$cost_{10} = n$，(n>1)。代价敏感错误率的计算可以化简为：

$$E(f;cost) = \frac{1}{m} \times \left(\sum_{x_i \in P} I(f(x_i) \neq y_i) + \sum_{x_i \in N} I(f(x_i) \neq y_i) \times n \right)$$

表 2 代价敏感矩阵

真实类别	预测类别	
	好客户	坏客户
好客户	0	$cost_{01}$
坏客户	$cost_{10}$	0

3.4 实验设计

为了验证本文提出的基于 Ext-GBDT 集成的类别不平衡信用评分模型（记为：Un-Ext-GBDT Ensemble），实验采用五次五折交叉验证，使用上文所述的数据集和评价指标，比较了本文提出的模型与当前最为常见的信用评分模型的预测结果。这些模型包括单模型：决策树（DT）、逻辑回归（LR）、朴素贝叶斯（NB）、支持向量机（SVM）、随机森林（RF）；集成模型：决策树集成（DT Ensemble）、逻辑回归集成（LR Ensemble）、朴素贝叶斯集成（NB Ensemble）、支持向量机集成（SVM Ensemble）、随机森林集成（RF Ensemble）。具体如表 3 所示。

经参数调优，本文的 Un-Ext-GBDT Ensemble 模型在训练过程中，使用 40 个子模型，每个子模型学习率扰动范围为 0.01~0.03、树最大深度扰动范围为 5~7、属性采样比扰动范围为 0.7~0.8、迭代次数扰动范围为 100~200。

表 3 基准模型

类型	名称	相关文献
单模型	DT	[5]
	LR	[2]
	NB	[11]
	SVM	[7]
	RF	[11]
集成模型	DT Ensemble	[10]
	LR Ensemble	[12]
	NB Ensemble	[11]
	SVM Ensemble	[10]
	RF Ensemble	[9]

此外，为了实验对比更全面，对比模型还增加本文的 Ext-GBDT 单模型（记为：Ext-GBDT）和 Ext-GBDT 集成模型（记为：Ext-GBDT Ensemble），经参数调优，本文使用如下的参数：

Ext-GBDT：学习率为 0.03，树最大深度为 6，属性采样比为 0.8，迭代次数为 100。

Ext-GBDT Ensemble：使用 40 个子模型，每个子模型使用全部训练集训练，每个子模型学习率扰动范围为 0.01~0.03、树最大深度扰动范围为 5~7、属性采样比扰动范围为 0.7~0.8、迭代次数扰动范围为 100~200。

3.5 实验结果

以 AUC 为评价指标，实验结果如表 4 所示，其中 Fold1、Fold2、Fold3、Fold4、Fold5 分别为 5 次交叉验证的结果，Avg 为这 5 次交叉验证结果的平均。从表 4 中可以看出：

a) 在所有的评估模型中，本文提出的 Un-Ext-GBDT Ensemble 模型具有最高的 AUC 值，预测效果最好。

b) 在所有单模型中，Ext-GBDT 的 AUC 值最高，预测偏差最小。

c) 集成模型的预测效果普遍比单模型好，其主要原因在于集成学习能降低预测结果的方差，提高模型的泛化能力。

以代价敏感错误率为评价指标，实验结果如表 5 所示，其中 Fold1、Fold2、Fold3、Fold4、Fold5 分别为 5 次交叉验证的结果，Avg 为这 5 次交叉验证结果的平均，n 表示将“坏客户误判为好客户”与将“好客户误判为坏客户”的代价比。从表中可以看出，相对于其他模型，Un-Ext-GBDT Ensemble 模型将“好客户误判为坏客户”的概率比较高，将“坏客户误判为好客户”的概率比较低，Un-Ext-GBDT Ensemble 模型更倾向于降低“坏客户误判为好客户”的概率。而在实际的放贷业务中，这两类误判的代价是不同的，将“坏客户误判为好客户”的代价要高于“将好客户误判为坏客户”的代价。表中列“n=5”表示假定代价比为 5 时的代价敏感错误率，列“n=10”表示假定代价比

为 10 时的代价敏感错误率，更直观的反映了 Un-Ext-GBDT Ensemble 模型的效果比其他常见的信用评分模型效果更好，并且随着代价比的提高，Un-Ext-GBDT Ensemble 模型的效果越好。

表 4 不同模型的 AUC 值

模型	AUC					
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Avg
DT	0.7107	0.6250	0.6190	0.6464	0.6310	0.6464
LR	0.7794	0.7342	0.7813	0.8146	0.7715	0.7762
NB	0.7539	0.7175	0.7262	0.7765	0.7661	0.7480
SVM	0.7812	0.7320	0.7750	0.7932	0.7720	0.7707
RF	0.8022	0.6925	0.7447	0.7867	0.7407	0.7534
Ext-GBDT	0.8206	0.7362	0.7905	0.8304	0.7752	0.7906
DT Ensemble	0.8115	0.7342	0.7570	0.8258	0.7367	0.7730
LR Ensemble	0.7896	0.7408	0.7929	0.8272	0.7744	0.7850
NB Ensemble	0.7696	0.7275	0.7300	0.7735	0.7721	0.7545
SVM Ensemble	0.8004	0.7410	0.7889	0.8015	0.7754	0.7814
RF Ensemble	0.8121	0.7376	0.7786	0.8228	0.7605	0.7823
Ext-GBDT Ensemble	0.8250	0.7414	0.7951	0.8299	0.7746	0.7932
Un-Ext-GBDT Ensemble	0.8300	0.7451	0.7990	0.8388	0.7815	**0.8008**

表 5 不同模型的代价敏感错误率

模型	代价敏感错误率							
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Avg	n=5	n=10
DT	0.17+0.15n	0.19+0.165n	0.18+0.18n	0.2+0.155n	0.14+0.15n	0.176+0.16n	0.976	1.776
LR	0.16+0.105n	0.105+0.165n	0.055+0.175n	0.07+0.15n	0.07+0.155n	0.07+0.161n	0.875	1.680
NB	0.145+0.12n	0.16+0.145n	0.155+0.14n	0.18+0.105n	0.18+0.13n	0.164+0.128n	0.804	1.444
SVM	0.06+0.16n	0.115+0.155n	0.06+0.165n	0.085+0.145n	0.075+0.155n	0.079+0.156n	0.859	1.639
RF	0.11+0.13n	0.135+0.18n	0.09+0.165n	0.12+0.14n	0.105+0.14n	0.112+0.151	0.867	1.622
Ext-GBDT	0.06+0.15n	0.09+0.16n	0.035+0.175n	0.095+0.125n	0.055+0.185n	0.067+0.159n	0.862	1.657
DT Ensemble	0.06+0.165n	0.09+0.17n	0.03+0.18n	0.1+0.13n	0.045+0.195n	0.065+0.168n	0.905	1.745
LR Ensemble	0.05+0.155n	0.095+0.15n	0.04+0.185n	0.1+0.135n	0.055+0.165n	0.068+0.158n	0.858	1.648
NB Ensemble	0.155+0.12n	0.15+0.145n	0.16+0.13n	0.175+0.128n	0.145+0.12n	0.157+0.129n	0.802	1.447
SVM Ensemble	0.06+0.15n	0.115+0.16n	0.06+0.16n	0.08+0.15n	0.075+0.16n	0.078+0.156n	0.858	1.638
RF Ensemble	0.065+0.15n	0.075+0.205n	0.04+0.21n	0.055+0.155n	0.045+0.205n	0.056+0.185n	0.981	1.906
Ext-GBDT Ensemble	0.12+0.12n	0.105+0.17n	0.145+0.14n	0.12+0.14n	0.055+0.185n	0.109+0.151n	0.864	1.619
Un-Ext-GBDT Ensemble	0.225+0.07n	0.25+0.07n	0.19+0.09n	0.22+0.04n	0.2+0.075n	**0.217+0.069n**	**0.562**	**0.907**

4 在实际信用评分业务中的可行性分析

4.1 转换信用概率为信用评分

传统的信用评分模型只是给出了好坏客户分类或者好坏客户的概率，但是这不容易理解，在实际的应用中不容易掌握。对于放贷机构，他们更希望的是能以一种更直观的方式认识每一个贷款申请者的信用情况，以指导其更好地制定放贷策略。本文提出的基于 Ext-GBDT 集成的类别不平衡信用评分模型在前面预测的信用概率的基础上，使用总体转换法，将概率值转换为信用评分。总体转换法的公式[21]如下：

$$\text{Score} = \ln\left(\frac{P}{1-P}\right)\times\text{factor} + \text{offset}$$

其中，P 表示申请客户为“好”客户的概率，1-P 表示申请客户为“坏”客户的概率，factor 表示线性变换的系数，通常是一个对数值，offset 表示为调整常数，目的是将评分阈值调整到目标区间。

4.2 坏账率和 K-S 检验

为了验证该模型在实际业务中的可行性，本文随机抽取 UCI 德国信用数据集中 80%的样本作为训练数据训练模型，剩下的 20%的样本作为测试数据，使用上文实验设计中的模型参数，并取 factor=ln60，offset=600，计算测试数据的信用得分。将得分按照由高到底排序，然后划分为 10 个分数段，每个分数段内有 20 个客户，如表 6 所示。观察好坏客户的分布情况可以发现，坏客户集中在 580 以下，随着得分的提高，坏账率迅速

下降。说明该模型比较有效地反映了客户的信用风险特征，通过得分的高低能够区分客户的信用状况。

表 6 测试数据信用分数分布表

分数段	好客户数	坏客户数	总数	好客户累计百分比	坏客户累计百分比	好坏比	坏账率
434 ~ 502	3	17	20	2.14%	28.33%	0.18	85.00%
503 ~ 525	11	9	20	10.00%	43.33%	1.22	45.00%
526 ~ 549	9	11	20	16.43%	61.67%	0.82	55.00%
550 ~ 580	11	9	20	24.29%	76.67%	1.22	45.00%
581 - 604	14	6	20	34.29%	86.67%	2.33	30.00%
605 ~ 624	17	3	20	46.43%	91.67%	5.67	15.00%
625 ~ 651	17	3	20	58.57%	96.67%	5.67	15.00%
652 - 691	19	1	20	72.14%	98.33%	19.00	5.00%
692 ~ 730	19	1	20	85.71%	100.00%	19.00	5.00%
731 - 768	20	0	20	100.00%	100.00%	N/A	0.00%

在实际的金融风控业务中，常用 K-S 值 (klmogrov-smirnov) 来度量风控模型的性能。K-S 值通过“好”客户和“坏”客户的累积百分比函数之间的最大距离来度量模型的区分能力，距离越大，K-S 值越大，则模型对“好”客户和“坏”客户的区分能力越强。K-S 值的区分能力解释如表 7 所示。

表 7　K-S 值区分能力[30]

K-S 值	区分能力
<0.20	无
0.21~0.40	低
0.41~0.50	中
0.51~0.60	高
0.61~0.75	极高
>0.9	太高，可能有问题

图 2 为本文模型在 UCI 德国信用数据集上的 K-S 曲线，计算可得 K-S 值为 0.574，属于区分能力高这一档，可以应用在实际的信贷业务中。

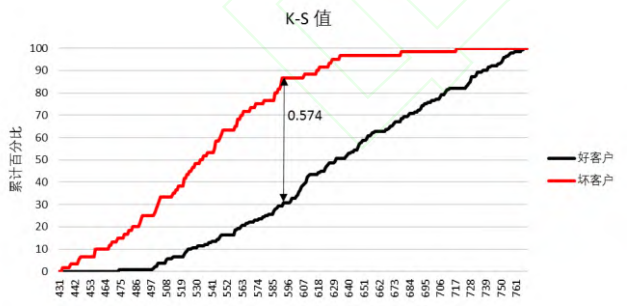

图 2 K-S 曲线

5　结束语

基于实际信贷业务的特点，本文提出了一种基于 Ext-GBDT 集成的类别不平衡信用评分模型。在处理类别不平衡问题方面，模型使用随机“欠采样”的方法产生多个类别平衡的训练数据来训练多个基分类器，并对基分类器进行集成使得全局上样本信息不丢失，从而在不丢失样本信息和改变原始样本分布的情况下学习类别不平衡的信用评分模型。在提升模型性能方面，模型使用样本扰动敏感、特征扰动敏感和参数扰动敏感的强分类器 Ext-GBDT 作为基分类器，并使用样本扰动、特征扰动和参数扰动的方法来学习多样的 Ext-GBDT 基分类器，使得集成模型的基分类器“好而不同”，从而能学习出很好的集成模型。实验在 UCI 德国上比较了本文模型与目前常见的信用评分模型的 AUC 和代价敏感错误率，结果表明，本文提出的模型具有更高的 AUC 和更小的代价敏感错误率。这些提升都能给放贷机构带来更大的收益。最后，结合实际信贷业务的对信用评分模型的实际需求，本文的进一步将模型输出的信用概率转换为直观的信用评分，并用坏账率表和 K-S 值分析该模型在实际业务中的可行性。

参考文献：

[1] Huang C L, Chen Muchen, Wang C J. Credit scoring with a data mining approach based on support vector machines[J]. Expert Systems with Applications, 2007, 33(4): 847-856.

[2] Brown I, Mues C. An experimental comparison of classification algorithms for imbalanced credit scoring data sets[J]. Expert Systems with Applications, 2012, 39(3): 3446-3453.

[3] Altman E I. Financial ratios, Discriminant analysis and the prediction of corporate bankruptcy[J]. Journal of Finance, 1968, 23(4): 589-609

[4] Desai V S, Crook J N, Overstreet G A. A comparison of neural networks and linear scoring models in the credit union environment[J]. European Journal of Operational Research, 1996, 95(1): 24-37

[5] Arminger G, Enache D, Bonne T. Analyzing Credit Risk Data: A Comparison of Logistic Discrimination, Classification Tree Analysis, and Feedforward Networks[J]. Social Science Electronic Publishing, 1997, 12(2): 293-310.

[6] West D. Neural network credit scoring models[J]. Computers & Operations Research, 2000, 27(11–12): 1131-1152.

[7] Baesens B, Vanthienen J. Benchmarking state-of-the-art classification algorithms for credit scoring[J]. Journal of the Operational Research Society, 2003, 54(6): 627-635.

[8] Yang Y. Adaptive credit scoring with kernel learning methods[J]. European

Journal of Operational Research, 2007, 183(3): 1521-1536.

[9] Nanni L, Lumini A. An experimental comparison of ensemble classifiers for bankruptcy prediction and credit scoring[J]. Expert Systems with Applications, 2009, 36(2): 3028-3033.

[10] Calabrese P, Gambassi A. A Hybrid Support Vector Machine Ensemble Model for Credit Scoring[J]. International Journal of Computer Applications, 2011, 17(5): 1-5.

[11] Lessmann S, Baesens B, Seow H V, *et al*. Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research[J]. European Journal of Operational Research, 2015, 247(1): 1-32.

[12] Wang Hong, Xu Qing-Song, Zhou Li-Feng. Large Unbalanced Credit Scoring Using Lasso-Logistic Regression Ensemble[J]. Plos One, 2015, 10(2).

[13] Ala'Raj M, Abbod M F. Classifiers consensus system approach for credit scoring[J]. Knowledge-Based Systems, 2016, 104: 89-105.

[14] Alejo R, García V, Marqués A I, *et al*. Making Accurate Credit Risk Predictions with Cost-Sensitive MLP Neural Networks[M]// Management Intelligent Systems. 2013: 1-8.

[15] Zięba M, Tomczak J M, Gonczarek A. RBM-SMOTE: Restricted Boltzmann Machines for Synthetic Minority Oversampling Technique[M]// Intelligent Information and Database Systems. [S. l.]: Springer International Publishing, 2015: 377-386.

[16] Liu Xuying, Wu Jianxin, Zhou Zhihua. Exploratory undersampling for class-imbalance learning. [J]. IEEE Trans on Systems Man & Cybernetics, Part B, 2009, 39(2): 539-50.

[17] Chawla N V, Japkowicz N, Kotcz A. Editorial: special issue on learning from imbalanced data sets[J]. ACM SIGKDD Explorations Newsletter, 2004, 6(1): 1-6.

[18] Bradley A P. The use of the area under the ROC curve in the evaluation of machine learning algorithms[J]. Pattern Recognition, 1997, 30(7): 1145-1159.

[19] Fawcett T. Introduction to ROC analysis[J]. Pattern Recognition Letters, 2006, 27(8): 861-874.

[20] 宋文力, 王祥, 胡波. 对商业银行贷款坏账率的预测研究[J]. 经济学动态, 2002(7): 40-44.

[21] 朱艳敏. 基于信用评分模型的小微企业贷款的可获得性研究[D]. 苏州: 苏州大学, 2014.

[22] Chen TianQi, Guestrin C. XGBoost: a scalable tree boosting system[C]//Proc of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016.

[23] Friedman J H. Greedy Function Approximation: A Gradient Boosting Machine[J]. Annals of Statistics, 2001, 29(5): 1189-1232.

[24] Breiman L I, Friedman J H, Olshen R A, *et al*. Classification and Regression Trees (CART)[M]// Classification and Regression Trees. [S. l.]: Chapman & Hall/CRC, 1998: 17–23.

[25] Rokach L. Ensemble-based classifiers[J]. Artificial Intelligence Review, 2010, 33(1): 1-39.

[26] Hastie T, Tibshirani R, Friedman J. Boosting and Additive Trees[M]// The Elements of Statistical Learning. 2009: 1071-1080.

[27] Ho T K, Hull J J, Srihari S N. Decision combination in multiple classifier systems[J]. IEEE Trans on Pattern Analysis & Machine Intelligence, 1994, 16(1): 66-75.

[28] Xu L, Krzyzak A, Suen C Y. Methods of combining multiple classifiers and their applications to handwriting recognition[J]. IEEE Trans on Cybernetics, 1992, 22(3): 418-435.

[29] UCI. Statlog(German Credit Data) data set[EB/OL]. https: //archive. ics. uci. edu/ml/machine-learning-databases/statlog/german

[30] 单良, 茆小林. 互联网金融时代消费信贷评分建模与应用[M]. 北京: 电子工业出版社, 2015, P143