第51卷第5期 南京航空航天大学学报 Vol. 51 No. 5
2019年10月 Journal of Nanjing University of Aeronautics & Astronautics Oct. 2019

DOI:10.16356/j.1005-2615.2019.05.009

基于XGBoost的三分类优惠券预测方法

张微薇[1] 刘 盾[1] 贾修一[2]

(1.西南交通大学经济管理学院,成都,610031;2.南京理工大学计算机科学与工程学院,南京,210094)

摘要: 在O2O营销过程中,优惠券是一种行之有效的营销工具。然而,在不清楚用户是否有消费意愿的情况下,就会产生优惠券滥发的现象。为了提高优惠券的使用率,本文首先将三支决策思想引入到优惠券使用预测问题中,并结合机器学习算法中的集成算法XGBoost对优惠券的使用情况进行模型构建。其次,在三支决策过程中考虑误分类成本和学习成本,使得分类过程更加贴近实际。最后,对阿里巴巴在天池平台提供的用户优惠券真实消费数据进行实验分析。结果表明,使用基于XGBoost的三分类算法可以有效提高分类的精确度。商户不仅可以维持老顾客,还能识别出潜在新客户,从而降低商户的营销成本。

关键词: XGBoost算法;三支决策;误分类成本;学习成本

中图分类号: TP181 **文献标志码:** A **文章编号:** 1005-2615(2019)05-0643-09

Three Classified Coupon Prediction Based on XGBoost Algorithm

ZHANG Weiwei[1], *LIU Dun*[1], *JIA Xiuyi*[2]

(1. School of Economics and Management, Southwest Jiaotong University, Chengdu, 610031, China; 2. School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, 210094, China)

Abstract: In the O2O marketing, coupon is an effective marketing tool. However, when we are not clear whether customers are willing to consume, coupons will be in spamming. In order to improve the utilization rate of coupons, in this paper, firstly the three-way decision-making thought is introduced to forecast the coupons utilization. And combine with the integrated algorithm XGBoost in machine learning, the model for the coupon usage is established. Second, in the process of three-way decision-making, taking into account misclassification cost and learning cost, the classification process should be more close to the actual classification. Finally, the real consumption data of users' coupon provided in Tianchi platform by Alibaba is experimentally analyzed. The results show that the XGBoost-based three-classification algorithm could effectively improve the accuracy of classification, so that merchants could not only maintain the regular customers, but also identify potential new customers, thereby reducing their marketing cost.

Key words: XGBoost algorithm; three-way decisions; misclassification cost; teaching cost

近年来,随着互联网飞速发展,我国的电子商务业务量呈爆炸式增长,企业的营销重心已从线下发展到线上。然而,仍有许多本地的生活服务需要离线完成,线下和线上融合的商业模式逐渐流行起来,这就是与生活息息相关的线上到线下(Online to offline,O2O)模式。O2O是一种多渠道营销的新型电子商务模式,其重点是通过平台进行在线促销,以提高实体店的销售[1]。

随着O2O的不断发展,各个商家试图通过不同的方式引导潜在的消费群体,而优惠券则是一种

基金项目: 国家自然科学基金(61876157,71571148,61773208)资助项目;四川省科技厅应用基础面上基金(2017JY0221)资助项目。

收稿日期: 2019-06-03;**修订日期:** 2019-07-31

通信作者: 刘盾,男,教授,E-mail:newton83@163.com。

引用格式: 张微薇,刘盾,贾修一. 基于XGBoost的三分类优惠券预测方法[J]. 南京航空航天大学学报,2019,51(5):643-651. ZHANG Weiwei, LIU Dun, JIA Xiuyi. Three Classified Coupon Prediction Based on XGBoost Algorithm[J]. Journal of Nanjing University of Aeronautics & Astronautics,2019,51(5):643-651.

行之有效的营销工具。使用优惠券,商家可以提高顾客的忠诚度或获取新顾客,顾客在购买商品或服务时可以获得利益。因此,优惠券被认为是一种双赢的应用[2]。然而,对商家来说,滥发优惠券不仅会增加营销成本,更有可能引起顾客的厌烦心理,损害品牌声誉。因此,识别那些更有可能使用优惠券的顾客,从而进行精准投放,显得极为重要。

根据所采用方法不同,关于优惠券的研究可分为理论和算法两个方面。Zou等提出一个基于位置作为优惠券渠道的模型,来讨论位置是如何影响顾客购买行为[3];Zheng等用优惠券倾向和价值意识作为变量,讨论顾客消费的可能性[4];Kosmopoulou等用改进的Hotelling模型分析优惠券对价格歧视的影响[5]。上述方法都是从理论视角去讨论顾客购买的可能性,但没有进行算法和数据验证。Wu等将优惠券使用概率的预测作为一个二分类问题,利用机器学习方法分析用户的优惠券使用行为,通过朴素贝叶斯算法、KNN算法、Logistic回归、神经网络、决策树以及随机森林等学习算法对实际优惠券使用情况进行预测[6]。结果发现,随机森林模型具有更好的分类性能,对优惠券使用预测准确率最高。

为了提高优惠券的分类精确度和降低营销成本,本文将三支决策思想引入到优惠券使用预测问题中。三支决策是决策粗糙集理论的一种扩展,是一种新兴的处理不确定性问题的粒计算方法[7-11]。目前,三支决策理论已应用到很多领域。Zhou等[12]将三支决策应用在垃圾邮件分类上,提出了一种基于贝叶斯决策理论的三支决策方法;Li等[13]将三支决策应用在人脸识别上;Min等[14-15]将三支决策应用在推荐系统中,提出了基于随机森林的三支推荐和基于回归的三支推荐;Zhang等[16]将三支决策用于自然语言处理,提出了一种三支增强卷积神经网络模型来判断句子的情感倾向。

本文在以往学者将机器学习方法引入优惠券领域的研究基础上,结合人们在决策过程中的不确定性,即当信息不足以让用户做出确切判断时,允许延迟决策的实际情况,提出一种基于极端梯度提升(eXtreme gradient boosting,XGBoost)的三支决策方法。首先,根据集成算法XGBoost计算出每个顾客使用优惠券消费的概率。其次,引入三支决策模型,通过一个代价矩阵来获取划分阈值,并结合概率值和阈值预测顾客的优惠券使用行为。最后,利用天池平台的真实消费数据来验证模型的有效性。

1 相关概念

本节主要回顾集成算法XGBoost和三支决策相关的概念。

1.1 XGBoost算法

集成学习是一项重要的机器学习技术,其基本思路是协同训练多个模型,然后融合多个模型的输出结果作为最终结果。集成学习的方法大致分为三类:Bagging,Boosting和Stacking。Breiman[17]提出了Bagging的概念,即随机采样训练分类器。文献[18-19]对Bagging进行了深入研究,提出了该算法的优缺点。Freund和Shapire[20]提出了另一种增强技术Boosting算法,与Bagging不同,Boosting对每个预测值都赋予一个权重,经过分类器的训练后对权重进行改变。Boosting算法不仅有良好的性能,还具有很强的理论基础和算法特点[21-23]。Stacking (Stacked generalization)是训练不同的个体学习器来得到一个组合学习器的模型[24]。本文中用到的XGBoost算法属于Boosting思想,是在梯度提升树(Gradient boosting decision tree, GBDT)[25]上扩展而来的,它是由Chen和Guestrin[26]提出的一种解决现实分类问题的极差梯度增强树。XGBoost对GBDT的主要改进是对损失函数进行归一化,减小了模型的方差,降低了建模复杂性和模型过度拟合的可能性。此外,传统的GBDT方法只能处理学习中的一阶导数,而XGBoost可以通过泰勒展开处理高阶损失函数。XGBoost方法可以处理稀疏数据,灵活实现分布式并行计算[27]。到目前为止,XGBoost方法已经成为了解决机器学习和数据挖掘问题的一种重要计算工具。

XGBoost是一种专注于梯度提升的机器学习算法[26]。XGBoost的原始模型是GBDT,它将弱学习器模型以迭代的方式组合成强学习器。XGBoost一般用于解决监督学习问题,利用大量的训练数据来预测目标变量。图1给出了XGBoost的算法流程图。XGBoost选择决策树作为它的弱学习器,在每次训练单个弱学习器时,都将上一次分错的类的数据权重提高一点再进行当前单个弱学习器的学习,然后通过加入新的弱学习器,来纠正前面所有弱学习器的残差,最终把多个学习器加权求和,用来进行最终预测。

XGBoost基本模型是决策树,首先定义单颗决策树的输出为

$$f(\boldsymbol{x})=\omega_{q(\boldsymbol{x})},\omega\in\mathbf{R}^{\mathrm{T}},q:\mathbf{R}^{d}\rightarrow\{1,2,\cdots,T\} \quad (1)$$

式中:$\boldsymbol{x}$是输入向量,q表示树的结构,结构函数

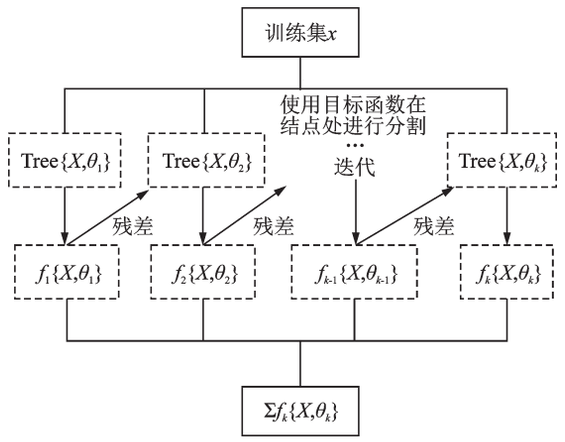

图1　XGBoost的算法流程图

Fig. 1　Flow chart of XGBoost algorithm

$q(\boldsymbol{x})$表示把输入映射到叶子的索引号，ω表示对应于每个索引号的叶子的分数，T是树中叶节点的数量，d为特征维数。

XGBoost算法可以看成是由K颗决策树的集成，则K颗树的集合的输出为

$$y_i=\sum_{k=1}^{K} f_k(x_i) \tag{2}$$

由决策树的模型可知，单颗决策树的复杂度计算公式为

$$\Omega(f_k)=\gamma T+\frac{1}{2}\lambda\|\omega\|^2 \tag{3}$$

类似地，集成树的复杂度可表示为

$$\Omega(f_k)=\gamma T+\frac{1}{2}\lambda\sum_{j=1}^{T}\omega_j^2 \tag{4}$$

式中：T是叶节点的数目，γ是范围在0和1之间的学习速率。γ乘以T等于生成树修剪，防止过度拟合。λ是一个正规化参数，ω是叶子的质量。$\Omega(f_k)$是XGBoost算法的正则项。

此外，XGBoost算法的目标函数在第t步的迭代可以表示为

$$J^{(t)}=\sum_{i=1}^{n} L(y_i,\hat{y}_i)+\sum_{k=1}^{t}\Omega(f_k) \tag{5}$$

式(5)包含两个部分：第一部分代表真实值y_i和预测值$\hat{y}_i$的误差之和，L为误差函数。第二部分代表单颗决策树的复杂度之和。已知：$\hat{y}^t$与$\hat{y}^{t-1}$的函数关系为：$\hat{y}^t=\hat{y}^{(t-1)}+f_t(x_i)$，其中$f_t(x_i)$为第$t$轮需要学习的决策树。因此，式(5)中目标函数可转化为

$$Obj^{(t)}=\sum_{i=1}^{n} L(y_i,\hat{y}_i^t)+\sum_{i=1}^{t}\Omega(f_i)=$$
$$\sum_{i=1}^{n} L(y_i,\hat{y}_i^{(t-1)}+f(x_i))+\sum_{i=1}^{t}\Omega(f_i) \tag{6}$$

总而言之，XGBoost算法是GBDT算法的一种改进，在实际应用中具有一定优势。比如：引入正则化步骤来防止减少过度拟合；并行处理提高了操作速度；允许用户自定义优化目标和评价标准，增加了灵活性；包含处理缺失值的规则；具有特殊的修剪步骤来控制决策树的复杂性。

1.2　三支决策

三支决策是由决策粗糙集理论发展而来的一种新的决策思想，近年来深受各领域学者的关注。相较于二支决策，三支决策不仅符合人的决策行为，而且处理方法更为合理。当提供的信息不足以让用户做出确切判断时，允许延迟决策，等待更多的信息进行下一步决策。因此，三支决策可以规避分类信息不足但却盲目决策造成的风险。

在三支决策理论中，决策者将概念划分为3个区域：正域、负域和边界域。不同的域对应着不同的决策。对于优惠券使用预测问题，决策区域和决策规则关系如表1所示。在决策粗糙集中，x表示顾客，$P(X|[x])$表示某一顾客x的等价类$[x]$属于会使用优惠券这一概念X的先验概率。α和β是划分3个区域的阈值，具体计算将在下一节中阐述。从表中可知，当概率值大于α时，表明顾客x在正域中，预测会使用优惠券；概率值小于β时，表明顾客x在负域中，预测不会使用优惠券；概率值在α和β值之间时，表明顾客x在边界域中，需要更多的信息来进行决策。

在传统决策中，人们往往采用二支决策，即只考虑正域和负域两种情况。然而在信息不充足的情况下，无法做出确切的判定。换言之，此时做出决策的成本极大可能高于暂时不做决策所带来的成本。在传统的二支决策问题中，分类问题会产生误分类成本。误分类成本指的是将对象分到错误的类中所带来的成本。三支决策在分类过程中除了会产生误分类成本以外，还会产生学习成本。学习成本指的是将对象分到边界域后，还需获得更多信息来进行下一步决策的成本。在处理优惠券使用预测问题时，将不能明确被划分区域的顾客放在边界域里延迟决策，等待更多的信息来进行下一步决策。总的来说，基于三支决策的分类问题将会以总成本最小为目来进行分类。

表1　三支决策分类

Tab. 1　Three-way decision classification

数学条件	决策区域	决策规则
$P(X\|[x])\geqslant\alpha$	正域	正类
$\beta\leqslant P(X\|[x])\leqslant\alpha$	边界域	延迟分类
$P(X\|[x])\leqslant\beta$	负域	负类

2 基于XGBoost的三分类优惠券问题

为提高优惠券分类问题的准确率,降低营销成本,本文提出基于XGBoost的三分类算法(TWD-XGBoost)。该算法主要包括以下两步:(1)利用XGBoost算法预测用户领取优惠券之后的使用概率。(2)根据得到的预测使用概率,结合三支决策思想,建立三分类模型,实现优惠券的使用预测。

2.1 基于XGBoost的分类算法

使用XGBoost算法来预测用户领到优惠券之后的使用概率,主要包括两步:(1)构建XGBoost算法模型,并利用训练集训练出相应参数。(2)将训练好的模型用于测试集,计算出用户领用相应优惠券后的使用概率。

XGBoost算法采用CART树作为基学习器,首先定义优惠券预测模型的目标函数,目标函数定义为

$$J(\theta)=L(\theta)+\Omega(\theta) \tag{7}$$

式中:θ为各种公式里的参数。$L(\theta)$是损失函数,通常衡量模型与训练数据的拟合度。常用的损失函数有平方损失和Logistic损失等。$\Omega(\theta)$是一个处罚复杂模型的正规化项。引入正则化的目的在于:通过训练集数据生成的模型能准确地预测新样本,既考虑到模型的简单性,又能保证模型训练误差最小化。模型通过正则化后,不仅很好地能拟合训练集,也能在测试集上表现良好。常用的正则有L1正则和L2正则。

由于优惠券预测是一个分类问题,本文使用Logistic分类器表示损失函数,即

$$L(\theta)=y_i\ln(1+e^{-\hat{y}_i})+(1-y_i)\ln(1+e^{\hat{y}_i}) \tag{8}$$

由式(5)可知,基于XGBoost的优惠券预测模型第t棵树的目标函数。通过对损失函数在$\hat{y}_i^{t-1}$处进行二阶泰勒展开,可以得到

$$\mathrm{Obj}^{(t)}\cong\sum_{i=1}^{n}[L(y_i,\hat{y}_i^{(t-1)})+g_if_t(x_i)+\frac{1}{2}h_if_t^2(x_i)]+\sum_{i=1}^{t}\Omega(f_t) \tag{9}$$

式中:$\Omega(f_t)$是正则化部分,它表示集成树的复杂度,由式(4)给出。g_i是第i个观测值在$t-1$个模型下的一阶偏导数的值,h_i是第i个观测值在$t-1$个模型下的二阶偏导数的值,它们分别表示为

$$\begin{aligned}g_i&=\partial_{\hat{y}^{(t-1)}}L(y_i,\hat{y}^{(t-1)})\\h_i&=\partial^2_{\hat{y}^{(t-1)}}L(y_i,\hat{y}^{(t-1)})\end{aligned} \tag{10}$$

如果忽略式(9)中的常数项,并代入第1.1节式(4)中的正则项,可将目标函数改写为

$$J^{(t)}\approx\sum_{i=1}^{n}[g_iw_{q(x_i)}+\frac{1}{2}(h_iw^2_{q(x_i)})]+\gamma T+\frac{1}{2}\lambda\sum_{j=1}^{T}w_j^2 \tag{11}$$

由于每个数据样本只对应一个叶节点,因此损失函数也可以表示为每个叶节点的损失值之和,即

$$J^{(t)}\approx\sum_{j=1}^{T}[(\sum_{i\in I_j}g_i)w_j+\frac{1}{2}(\sum_{i\in I_j}h_i+\lambda)w_j^2]+\gamma T \tag{12}$$

在式(12)中,G_j和H_j分别被定义为

$$G_j=\sum_{i\in I_j}g_i,H_j=\sum_{i\in I_j}h_i \tag{13}$$

式中I_j表示叶节点j中的所有数据样本,故目标函数的优化可以转换为二次函数的最小值问题,即:在决策树中分割出一定的节点后,可以根据目标函数来评估模型性能的变化。如果该节点分割后决策树模型的性能有所提高,则采用此更改,否则将停止分割。此外,目标函数优化时的正则化可以训练预测分类器防止过拟合。基于上述分析,最终的目标函数可表示为

$$\mathrm{Obj}^{(t)}=\sum_{j=1}^{T}[(\sum_{i\in I_j}g_i)w_j+\frac{1}{2}(\sum_{i\in I_j}h_i+\lambda)w_j^2]+\gamma T=\sum_{j=1}^{T}[G_jw_j+\frac{1}{2}(H_j+\lambda)w_j^2]+\gamma T \tag{14}$$

进一步地,假设XGBoost决策树的结构是固定的,即$q(x)$是固定的,Obj的一阶导数为0,即目标函数达到最佳,则可求得叶子节点j对应的参数值为

$$w_j^*=-\frac{G_j}{H_j+\lambda} \tag{15}$$

式(15)代入到式(14)中,则目标函数可简化为

$$\mathrm{Obj}=-\frac{1}{2}\sum_{i=1}^{T}\frac{G_j^2}{H_i+\lambda}+\gamma T \tag{16}$$

综上所述,基于XGBoost的优惠券预测模型构建分为以下4个步骤:

(1)生成一个决策树;

(2)计算在每个训练样本点处损失函数的一阶导数g_i和二阶导数h_i;

(3)通过贪婪策略生成新的决策树,计算每个叶节点对应的预测值;

(4)将新生成的决策树$f_t(x)$添加到模型中并继续迭代。

用划分好的训练集数据训练好XGBoost模型之后,就可以利用测试集数据用训练好的模型去预测用户在领到优惠券之后的使用概率。

2.2　三分类模型

三分类的核心思想是在分类过程中根据预测概率、分类成本决定预测用户是否会使用优惠券或者延迟决策。

2.2.1　阈值的计算

在贝叶斯决策过程中，状态集合可以表示为$\Omega=\{X,\neg X\}$，X和$\neg X$分别表示1个对象属于X和不属于X。状态X对应的3个分类行为由$R=\{a_P,a_B,a_N\}$表示，a_P表示将用户划分到正域，会使用优惠券；a_N表示将用户划分到负域，不会使用优惠券；a_B表示将用户划分到边界域，延迟决策。这些行为会产生1个3×2的成本矩阵。分类成本主要包括误分类成本和学习成本。如果用户行动属于P，则$\lambda_{PP},\lambda_{BP},\lambda_{NP}$分别表示采取行动$a_P,a_B,a_N$的成本。其中，$\lambda_{NP}$表示将拒绝正确推荐产生的误分类成本，$\lambda_{BP}$是延迟推荐的学习成本；如果用户行动属于$N$，则$\lambda_{PN},\lambda_{BN},\lambda_{NN}$分别表示采取行动$a_P,a_B,a_N$的成本。其中，$\lambda_{PN}$是接受错误推荐的误分类成本，$\lambda_{BN}$是延迟不推荐的学习成本。则采取不同行为所产生的成本矩阵如表2所示。

表2　三支决策的成本函数

Tab. 2　Cost functions for three-way decision

项目	$X(P)$	$\neg X(N)$
a_P	λ_{PP}	λ_{PN}
a_B	λ_{BP}	λ_{BN}
a_N	λ_{NP}	λ_{NN}

因此，采取a_P,a_B,a_N这3种行动下的期望损失可分别表示为

$$\begin{aligned}T_P&=\lambda_{PP}P(X|[x])+\lambda_{PN}P(\neg X|[x])\\T_B&=\lambda_{BP}P(X|[x])+\lambda_{BN}P(\neg X|[x])\\T_N&=\lambda_{NP}P(X|[x])+\lambda_{NN}P(\neg X|[x])\end{aligned}\tag{17}$$

根据贝叶斯决策准则，需要选择期望损失最小的行动集作为最佳行动方案。因此，可以得到3条三支决策规则(P)～(N)

(P) 若 $T_P\leqslant T_B$ 且 $T_P\leqslant T_N$，则判定$x\in\mathrm{POS}(X)$；

(B) 若 $T_B\leqslant T_P$ 且 $T_B\leqslant T_N$，则判定$x\in\mathrm{BND}(X)$；

(N) 若 $T_N\leqslant T_P$ 且 $T_N\leqslant T_B$，则判定$x\in\mathrm{NEG}(X)$。

为了简化规则，考虑到每个对象只属于某一种概念，假设

$$P(X|[x])+P(\neg X|[x])=1\tag{18}$$

此外，在分类过程中，由于误分类成本通常高于延迟决策产生的学习成本，学习成本通常高于正确分类的成本，可以假设

$$\begin{aligned}\lambda_{PP}&\leqslant\lambda_{BP}<\lambda_{NP}\\\lambda_{NN}&\leqslant\lambda_{BN}<\lambda_{PN}\end{aligned}\tag{19}$$

将式(19)代入决策规则(P)～(N)，且考虑阈值大小关系$0\leqslant\beta<\alpha\leqslant1$，可计算得到如下条件

$$\frac{\lambda_{NP}-\lambda_{BP}}{\lambda_{BN}-\lambda_{NN}}>\frac{\lambda_{BN}-\lambda_{NN}}{\lambda_{PN}-\lambda_{BN}}\tag{20}$$

根据式(20)，决策规则(P)～(N)可重写为

(P1)若$P(X|[x])>\alpha$，则$x\in\mathrm{POS}(X)$；

(B1)若$\beta\leqslant P(X|[x])\leqslant\alpha$，则$x\in\mathrm{BND}(X)$；

(N1)若$P(X|[x])\leqslant\beta$，则$x\in\mathrm{NEG}(X)$

结合决策规则(P1)～(N1)，可以计算出阈值α和β的取值为

$$\begin{aligned}\alpha&=\frac{\lambda_{PN}-\lambda_{BN}}{(\lambda_{PN}-\lambda_{BN})+(\lambda_{BP}-\lambda_{PP})}\\\beta&=\frac{\lambda_{BN}-\lambda_{NN}}{(\lambda_{BN}-\lambda_{NN})+(\lambda_{NP}-\lambda_{BP})}\end{aligned}\tag{21}$$

2.2.2　平均成本计算

成本计算在整个分类过程中也十分重要，当给定成本矩阵时，可求得平均成本如下

$$T_C(\alpha,\beta)=T_P(\alpha,\beta)+T_B(\alpha,\beta)+T_N(\alpha,\beta)\tag{22}$$

式中：$T_C(\alpha,\beta)$表示在阈值为α,β时所产生的总成本，$T_P(\alpha,\beta)$，$T_B(\alpha,\beta)$和$T_N(\alpha,\beta)$分别表示阈值为α,β时将用户划分为使用优惠券的成本、延迟分类的成本、将用户划分为不使用优惠券的成本。$T_P(\alpha,\beta)$，$T_B(\alpha,\beta)$，$T_N(\alpha,\beta)$计算除式(17)的计算方式以外，还可以用式(23)进行计算

$$\begin{aligned}T_P(\alpha,\beta)&=\lambda_{PP}n_{PP}+\lambda_{PN}n_{PN}\\T_B(\alpha,\beta)&=\lambda_{BP}n_{BP}+\lambda_{BN}n_{BN}\\T_N(\alpha,\beta)&=\lambda_{NP}n_{NP}+\lambda_{NN}n_{NN}\end{aligned}\tag{23}$$

式中：n_{PP}，n_{BP}和n_{NP}分别表示将实际使用优惠券的用户预测成已使用、延迟分类、未使用的数量；n_{PN}，n_{BN}，n_{NN}分别表示将未用优惠券的用户预测成了已使用、延迟分类和未使用的数量。因此，可以进一步由式(24)计算出平均成本，它等于总成本除以总样本数。

综上所述，在三分类算法中，人们可以根据预测出的用户使用优惠券概率与阈值的大小关系将对象划分到不同的决策域中，并对不同的区域采取不同的决策策略。

$$\mathrm{aver}_{\mathrm{cost}}=\frac{T_C}{n}=\frac{\lambda_{PP}n_{PP}+\lambda_{PN}n_{PN}+\lambda_{BP}n_{BP}+\lambda_{BN}n_{BN}+\lambda_{NP}n_{NP}+\lambda_{NN}n_{NN}}{n_{PP}+n_{PN}+n_{BP}+n_{BN}+n_{NP}+n_{NN}}\tag{24}$$

3 实验结果与分析

为了验证本文所提出算法的有效性，本节选取随机森林、Adaboost，GBDT和XGBoost作为基准算法，进而将本文提出的基于XGBoost的三分类算法（TWD-XGBoost）与上述4种基准算法作对比分析。最后利用真实消费数据对各种算法的实验结果进行比较。

3.1 实验数据

本文使用的数据是来自天池平台的真实消费数据，它包含在线消费数据和离线消费数据。本文主要关注离线情况。离线数据集包含1 754 884条用户行为记录，其中用户539 438条，商家8 414条，优惠券9 738张。每个记录都可以看作是一个向量(用户ID、商户ID、优惠券ID、折扣率、距离、接收日期、日期)。表3是对上述字段的数据特征描述。

表3 数据特征描述

Tab. 3 Data feature description

属性名	描述
User_id	用户ID
Merchant_id	商户ID
Coupon_id	优惠券ID
Discount_rate	优惠率
Distance	用户和商户的距离
Date_received	领取优惠券日期
Date	消费日期

该数据集包含了从2016年1月1日至2016年6月30日之间真实消费行为。为了能够评估本文提出的模型，在处理过程中对数据进行了划分，2016年1月1日到2016年5月31日的数据被当作训练集，共924 931条，2016年6月1日到2016年6月30日的数据被当作测试集，共90 458条。

3.2 评估指标

本文所讨论的评估指标主要选取分类质量和分类成本两个维度。

首先，考虑到优惠券预测问题是1个三分类问题，类似于二分类问题，可用混淆矩阵进行性能评价。真实类别和预测类别分为真正类(TP)、真负类(TN)、假正类(FP)和假负类(FN)。与一般的二分类问题不同，本文采用的数据正负样本不均衡，负样本的数量远大于正样本的数量，所以为了更准确的度量模型预测的效果，结合实际情况，本文采取精确度(Precision)和AUC两个测度来评估模型。

精确度(Precision)描述了判断为真的正例占所有判断为真的样例比重，其计算方法为

$$\mathrm{Precision}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \tag{25}$$

AUC-AUC是ROC曲线的下面积，AUC值是一个概率值，当随机挑选一个正样本以及一个负样本，当前的分类算法根据计算得到的数值将这个正样本排在负样本前面的概率就是AUC值。在评估过程中，这两个值越大，说明模型效果越好。

其次，在计算分类成本时，主要考虑错误分类所产生的误分类成本和延迟决策的学习成本。为了简化模型，假设正确的分类决策不产生误分类成本，即：$\lambda_{PP}=\lambda_{NN}=0$。对于优惠券问题，将实际使用了优惠券的顾客预测为未使用通常比将实际未使用优惠券的顾客预测为使用了成本更高，即：$\lambda_{PN}\leqslant\lambda_{NP}$。根据上述假设，可以计算平均分类成本为

$$\mathrm{aver}_{\mathrm{cost}}=\frac{\lambda_{PN}n_{PN}+\lambda_{BP}n_{BP}+\lambda_{BN}n_{BN}+\lambda_{NP}n_{NP}}{n_{PP}+n_{PN}+n_{BP}+n_{BN}+n_{NP}+n_{NN}} \tag{26}$$

3.3 实验分析与实验结果

为了验证本文提出的TWD-XGBoost的优越性，我们主要从分类质量和分类成本两个方面进行测评。

从分类质量来看，选择了4种集成算法作为基准算法来进行对比分析，其中包括基于Bagging的随机森林算法，基于Boosting的Adaboost，GBDT和XGBoost算法。本文先对4种基准算法在优惠券预测问题上的各个指标进行比较，结果如表4所示。

表4 各集成算法效果比较

Tab. 4 Comparisons of each integration

模型	精确度	召回率	F1	AUC
随机森林	0.721 1	0.844 5	0.778 0	0.907 6
Adaboost	0.716 4	0.738 3	0.727 2	0.856 0
GBDT	0.723 8	0.815	0.794 9	0.925 6
XGBoost	0.734 9	0.885 7	0.803 3	0.928 5

由表4可知，在优惠券预测问题中，上述4种集成算法的精确度、召回率、F1和AUC排序均为：XGBoost＞GBDT＞随机森林＞Adaboost。在比较的这4种算法中，分类效果最好的是XGBoost。

考虑到TWD-XGBoost的分类质量与成本有关，下面研究分别固定λ_{NP}和λ_{PN}时，当其他成本参数变化时，TWD-XGBoost的精确度和AUC值的变化情况，实验结果如表5和表6所示。其中，本文成本参数的选择依据和范围均借鉴于文献[12, 14-15, 28]。

结合表4—6可以看到，TWD-XGBoost算法在表5中任意成本组合下的精确度值均大于表4中4种集成算法的精确度；类似地，TWD-XGBoost算法在任意成本组合下的AUC值均大于表4中4种

表 5　TWD-XGBoost算法的精确度

Tab. 5　Precision of TWD-XGBoost algorithm

精确度	$\lambda_{BP}=\lambda_{BN}$			
$(\lambda_{NP},\lambda_{PN})$	35	30	25	20
(120,80)	0.745 8	0.754 7	0.763 8	0.773 1
(120,90)	0.752 7	0.761 6	0.769 1	0.776 9
(120,100)	0.759 4	0.765 4	0.773 0	0.781 0
(120,110)	0.763 3	0.770 2	0.775 5	0.782 6
(120,120)	0.766 4	0.773 1	0.779 4	0.784 9
(80,80)	0.745 8	0.754 7	0.763 8	0.773 1
(90,80)	0.745 8	0.754 7	0.763 8	0.773 1
(100,80)	0.745 8	0.754 7	0.763 8	0.773 1
(110,80)	0.745 8	0.754 7	0.763 8	0.773 1
(120,80)	0.745 8	0.754 7	0.763 8	0.773 1

表 6　TWD-XGBoost算法的AUC值

Tab. 6　AUC of TWD-XGBoost algorithm

AUC	$\lambda_{BP}=\lambda_{BN}$			
$(\lambda_{NP},\lambda_{PN})$	35	30	25	20
(120,80)	0.951 2	0.955 5	0.959 0	0.962 2
(120,90)	0.951 2	0.955 5	0.958 9	0.962 1
(120,100)	0.951 2	0.955 3	0.958 9	0.962 1
(120,110)	0.950 9	0.955 3	0.958 7	0.962 0
(120,120)	0.950 8	0.955 2	0.958 7	0.961 8
(80,80)	0.935 6	0.943 0	0.949 4	0.955 2
(90,80)	0.941 6	0.947 4	0.952 4	0.957 6
(100,80)	0.946 5	0.950 8	0.955 4	0.959 7
(110,80)	0.949 1	0.953 2	0.957 3	0.961 0
(120,80)	0.951 2	0.955 5	0.959 0	0.962 2

集成算法的AUC值。进一步地，由表5可知，当保持λ_{NP}和λ_{PN}不变时，λ_{BP}越小，精确度越大；当保持λ_{NP}和λ_{BP}不变时，λ_{PN}越大，精确度越大；当保持λ_{PN}和λ_{BP}不变时，λ_{NP}越大，精确度越大；当保持λ_{PN}，λ_{BN}和λ_{BP}不变时，无论λ_{NP}变大还是变小，精确度都保持不变。究其原因，是因为精确度和分到正域的数量有关。在TWD-XGBoost算法中，确定分到正域的数量由阈值α决定，α越大，表明对正域划分越严格，精确度就越高。α由阈值计算公式(21)得出，其大小与λ_{BP}成反比，与λ_{PN}成正比，与λ_{NP}无关。

为了进一步比较本文提出的TWD-XGBoost算法与4种集成算法的分类质量，对上述40组成本组合下的精确度和AUC值求平均，用均值来刻画TWD-XGBoost算法的测试结果，相关实验结果如图2所示。

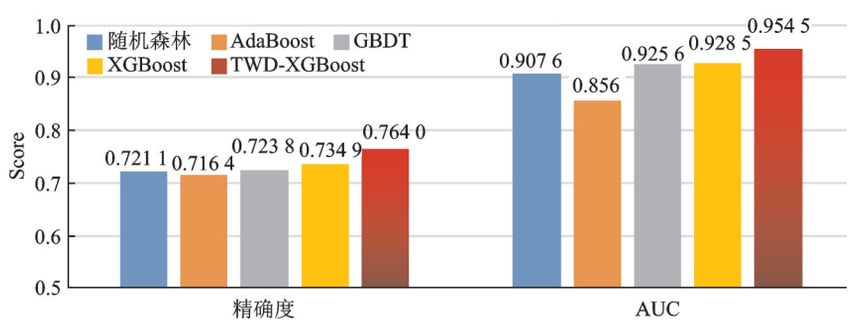

图2　各集成算法的分类效果比较

Fig. 2　Classifying quality comparisons of each integration algorithm

图2表明，相较于其他4种集成算法，TWD-XGBoost算法不仅在精确度和AUC指标上存在一定优势，而且在分类质量上也表现良好。由此可见，与常见的4种集成算法相比，TWD-XGBoost的预测精确度更高，分类质量更好。

为了进一步验证TWD-XGBoost在分类成本，分析了算法改进前后相应的决策成本变化情况。由于在表4的4种集成算法中，只有XGBoost考虑了成本代价，故这里主要选取TWD-XGBoost与XGBoost两种算法作比较。表7给出了当固定$\lambda_{NP}=120$时，两种算法的平均分类成本随着λ_{PN}，λ_{BP}和λ_{BN}变化的计算结果；表8展示了当固定$\lambda_{PN}=80$时，两种算法的平均分类成本随着λ_{NP}，λ_{BP}和λ_{BN}变化的计算结果。

表 7　TWD-XGBoost与XGBoost算法在不同λ取值下的成本比较

Tab. 7　Cost comparison of TWD-XGBoost with XGBoost algorithm under different lambda values

平均分类成本		λ_{PN}				
		120	110	100	90	80
XGBoost算法		4.288 8	4.025 5	3.762 3	3.499 1	3.235 8
$\lambda_{BN}=\lambda_{BP}$	35	3.918 2	3.732 3	3.533 9	3.334 2	3.115 1
	30	3.721 9	3.552 3	3.386 1	3.195 4	3.000 2
	25	3.505 9	3.363 2	3.195 8	3.030 0	2.855 0
	20	3.263 0	3.121 3	2.966 6	2.821 6	2.661 2

由表7可知，当固定λ_{NP}时，无论λ_{PN}，λ_{BP}和λ_{BN}如何变化，TWD-XGBoost平均分类成本要低于XGBoost算法；当固定λ_{PN}时，同样满足上述大小关

表8 TWD-XGBoost与XGBoost算法在不同λ取值下的成本比较

Tab. 8 Cost comparison of TWD-XGBoost with XGBoost algorithm under different lambda values

平均分类成本		λ_{NP}				
		120	110	100	90	80
XGBoost		3.235 8	3.141 7	3.047 5	2.953 4	2.859 2
$\lambda_{BN}=\lambda_{BP}$	35	3.115 1	3.049 1	2.981 6	2.919 3	2.837 6
	30	3.000 2	2.953 1	2.898 9	2.838 7	2.774 0
	25	2.855 0	2.806 4	2.760 2	2.713 2	2.653 5
	20	2.661 2	2.623 5	2.576 3	2.527 1	2.480 8

系。同时,当保持误分类成本λ_{NP}和λ_{PN}不变时,平均分类成本将随着学习成本λ_{BP}的减小而减小;当保持学习成本λ_{BP}和误分类成本λ_{PN}不变时,平均分类成本将随着误分类成本λ_{NP}的减小而减小。

此外,在表7与表8的实验基础上,进一步考虑了不同成本条件下TWD-XGBoost误分类数量与延迟分类数量,计算结果如表9所示。其中,n_{PN}表示用户实际未用但预测为用了的错误数量,$n_B=n_{BP}+n_{BN}$表示延迟分类的数量,n_{NP}表示用户实际用了但预测未用的数量。通过表9可得:当保持λ_{NP}和λ_{PN}不变时,λ_{BP}越小,n_{PN}和n_{NP}越小,n_B越大。这说明当学习成本减小时,延迟分类的数量会增加,误分类数量会减少。类似地,当保持λ_{NP}和λ_{BP}不变时,λ_{PN}越小,n_B就越小,n_{PN}越大;当保持λ_{PN}和λ_{BP}不变时,λ_{NP}越小,n_B就越小,n_{NP}越大。故当误分类成本减小时,延迟分类的数量会减少,误分类数量就会增加。由此可见,在TWD-XGBoost算法中,决策结果会向较小成本的决策区域进行偏移。

综合表7—9的实验结果,可以发现:与XGBoost算法不同,TWD-XGBoost采用三分类方法,通过延迟分类减少了系统的误分类数量,将不确定的部分划分在边界域,极大地降低了误分类成本,从而降低了平均分类成本,使得总成本减小。另外,TWD- XGBoost算法中的三分类方法结果偏向分类成本较小的决策区域,这也符合人们实际的决策行为。

综上所述,通过对分类质量和分类成本的分析,相对4种集成算法,本文提出的TWD-XGBoost能更好地预测用户的使用偏好,在明显提高用户分类质量的同时,也降低了用户分类成本。

表9 不同成本条件下TWD-XGBoost产生的误分类数和延迟分类数

Tab. 9 Number of classification errors and delayed classification generated by TWD-XGBoost under different cost conditions

$(\lambda_{NP},\lambda_{PN})$	$\lambda_{BP}=\lambda_{BN}=35$			$\lambda_{BP}=\lambda_{BN}=30$			$\lambda_{BP}=\lambda_{BN}=25$			$\lambda_{BP}=\lambda_{BN}=20$		
	n_{PN}	n_B	n_{NP}	n_{PN}	n_B	n_{NP}	n_{PN}	n_B	n_{NP}	n_{PN}	n_B	n_{NP}
(120,80)	2 164	1 316	484	2 001	1 877	422	1 834	2 505	373	1 680	3 164	330
(120,90)	2 043	1 566	484	1 879	2 155	422	1 745	2 715	373	1 613	3 318	330
(120,100)	1 924	1 830	484	1 806	2 338	422	1 680	2 867	373	1 553	3 457	330
(120,110)	1 846	2 023	484	1 731	2 504	422	1 630	3 011	373	1 515	3 578	330
(120,120)	1 786	2 181	484	1 680	2 632	422	1 575	3 131	373	1 475	3 691	330
(80,80)	2 164	617	723	2 001	1 286	603	1 834	1 962	505	1 680	2 632	422
(90,80)	2 164	860	629	2 001	1 472	538	1 834	2 124	463	1 680	2 771	390
(100,80)	2 164	1 053	554	2 001	1 631	490	1 834	2 270	422	1 680	2 930	363
(110,80)	2 164	1 189	515	2 001	1 757	455	1 834	2 380	396	1 680	3 052	346
(120,80)	2 164	1 316	484	2 001	1 877	422	1 834	2 505	373	1 680	3 143	330

4 结 论

本文将三支决策引入到XGBoost中,提出了一种基于XGBoost的三分类算法,用于解决优惠券使用预测问题。该算法在传统的XGBoost算法的基础上,引入三支决策“分而治之”和“化繁为简”思想,结合误分类成本、学习成本以及贝叶斯决策将决策空间划分为3个不同的区域,对不同区域上的样本进行相应处理:对正域和负域的样本直接给出预测结果,对边界域中的样本延迟决策。最后,通过天池平台提供的实际数据对XGBoost三分类算法进行算法测评。实验结果表明,相较于本文提到的其他集成算法,本文提出的算法TWD-XGBoost在处理优惠券预测问题时,分类质量和分类成本都有明显改进。在未来的工作中,笔者将继续优化算法,进一步提高优惠券使用预测的精确度和降低分类过程中的平均成本,并将该算法扩展到其他实际推荐问题中去。

参考文献:

[1] WEI P C, TAN C H, SUTANTO J. Leveraging O2O commerce for product promotion: An empirical

investigation in mainland China[J]. IEEE Transactions on Engineering Management, 2014, 61: 623-632.

[2] CHEN Liqun, ESCALANTE B A N, MANULIS M. A privacy-protecting multi-coupon scheme with stronger protection against splitting[C]//Proc of the 11th Int Conf on Financial Cryptography. Scarborough:[s.n.], 2007: 93-108.

[3] ZOU Xiao, HUANG Kewei. Leveraging location-based services for couponing and infomediation[J]. Decision Support Systems, 2015, 78: 93-103.

[4] ZHENG Xiabing, LEE M, CHEUNG C. Examining e-loyalty towards online shopping platforms: The role of coupon proneness and value consciousness[J]. Internet Research, 2017, 27: 243-272.

[5] KOSMOPOULOU G, LIU Qihong, SHUAI J. Customer poaching and coupon trading[J]. Journal of Economics, 2016, 118: 219-238.

[6] WU Jie, ZHANG Yulai, WANG Jianfen. Research on usage prediction methods for O2O coupons[C]// Proceeding of the 25th International Conference on Neural Information Processing. Cambodia:[s.n.], 2018: 175-183.

[7] LIU Dun, LI Tianrui, LIANG Decui. Three-way government decision analysis with decision-theoretic rough sets[J]. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2012, 20: 119-132.

[8] LIU Dun, YAO Yiyu, LI Tianrui. Three-way investment decisions with decision-theoretic rough sets [J]. International Journal of Computational Intelligence Systems, 2011, 4: 66-74.

[9] LI Tongjun, YANG Xiaoping. An axiomatic characterization of probabilistic rough sets[J]. International Jouranl of Approximate Reasoning, 2014, 55(1): 130-141.

[10] WU Weizhi, MI Jusheng, ZHANG Wenxiu. Generalized fuzzy rough sets[J]. Information Sciences, 2003, 151: 263-282.

[11] YANG Xibei, YANG Jingyu, WU Chen, et al. Dominance-based rough set approach and knowledge reductions in incomplete ordered information system [J]. Information Sciences, 2008, 178: 1219-1234.

[12] ZHOU Bing, YAO Yiyu, LUO Jigang. Costsensitive three-way email spam filtering [J]. Journal of Intelligent Information Systems, 2014, 42: 19-45.

[13] LI Huaxiong, ZHANG Libo, HUANG Bing, et al. Sequential three-way decision and granulation for cost-sensitive face recognition[J]. Knowledge- Based Systems, 2016, 91: 241-251.

[14] ZHANG Hengru, MIN Fan. Three-way recommender systems based on random forests[J]. Knowledge-Based Systems, 2016, 91: 275-286.

[15] ZHANG Hengru, MIN Fan, SHI Bing. Regression-based three-way recommendation [J]. Inf-ormation Sciences, 2017, 378: 444-461.

[16] ZHANG Yuebing, ZHANG Zhifen, MIAO Duoqian. Three-way enhanced convolutional neural networks for sentence-level sentiment classification[J]. Information Sciences,2019,477: 55-64.

[17] BREIMAN L. Bagging predictors[J]. Machine Learning, 1996, 24: 123-140.

[18] BUHLMANN P, YU Bin. Analyzing bagging[J]. Annals of Statistics, 2001, 30: 927-961.

[19] ZHU Xingquan, YANG Ying. A lazy bagging approach to classification[J]. Pattern Recognition, 2008, 41: 2980-2992.

[20] FREUND Y, SHAPIRE R E. Experiments with a new boosting algorithm[C]//Proceeding of 13th International Conference on Machine Learning. Bari:[s.n.], 1996: 148-156.

[21] HASTIE T, FRIEDMAN J, TIBSHIRANI R. The elements of statistical learning[J]. Technometrics, 2001, 45: 267-268.

[22] MEIR R R, TSCH G. Advanced lectures on machines learning[M]. Berlin Heidelberg: Spring-Verlag, 2003: 118-183.

[23] 曹莹,苗启广,刘家辰,等. AdaBoost算法研究进展与展望[J]. 自动化学报, 2013, 39(6): 745-758.
CAO Ying, MIAO Qiguang, LIU Jiachen, et al. Advance and prospects of AdaBoost algorithm[J]. Acta Automatica Sinica, 2013,39(6): 745-758.

[24] WOLPERT D H. Stacked generalization[J]. Neural Networks, 1992, 5: 241-259.

[25] FRIEDMAN J H. Greedy function approximation: Agradient boosting machine[J]. The Annals of Statistics, 2001, 29: 1189-1232.

[26] CHEN Tianqi, GUESTRIN C. XGBoost: A scalable tree boosting system[C]//ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.[S.l.]: ACM,2016: 785-794.

[27] WANG Shouxiang, DONG Pengfei, TIAN Yingjie. A novel method of statistical line loss estimation for distribution feeders based on feeder cluster and modified XGBoost[J]. Energies, 2017, 10: 2067-2084.

[28] 叶晓庆, 刘盾, 梁德翠. 基于协同过滤的三支粒推荐算法研究[J]. 计算机科学, 2018,45(1): 90-96.
YE Xiaoqing, LIU Dun, LIANG Decui. Three-way granular recommendation algorithm based on collaborative filtering[J]. Computer Science, 2018, 45(1): 90-96.

(编辑:刘彦东)