(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 105930875 A
(43)申请公布日 2016.09.07

(21)申请号 201610292389.0

(22)申请日 2016.05.05

(71)申请人 清华大学
地址 100084 北京市海淀区100084-82信箱

(72)发明人 唐杰 张茜 刘德兵

(74)专利代理机构 北京清亦华知识产权代理事务所(普通合伙) 11201
代理人 张大威

(51)Int.Cl.
G06K 9/62(2006.01)
G06Q 10/04(2012.01)
G06Q 50/20(2012.01)

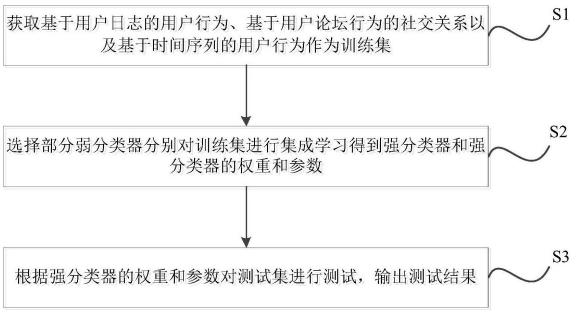
权利要求书2页 说明书11页 附图2页

(54)发明名称

用户退课行为预测方法及装置

(57)摘要

本发明公开了一种用户退课行为预测方法及装置,其中,该方法包括:获取基于用户日志的用户行为、基于用户论坛行为的社交关系以及基于时间序列的用户行为作为训练集;选择部分弱分类器分别对训练集进行集成学习得到强分类器和强分类器的权重和参数;根据强分类器的权重和参数对测试集进行测试,输出测试结果。该方法通过集成学习提高了对用户退课行为预测的准确度。本发明还提出了一种用户退课行为预测装置。



CN 105930875 A

1.一种用户退课行为预测方法，其特征在于，包括以下步骤：

S1，获取基于用户日志的用户行为、基于用户论坛行为的社交关系以及基于时间序列的用户行为作为训练集；

S2，选择部分弱分类器分别对所述训练集进行集成学习得到强分类器和所述强分类器的权重和参数；

S3，根据所述强分类器的权重和参数对测试集进行测试，输出测试结果。

2.如权利要求1所述的用户退课行为预测方法，其特征在于，所述部分弱分类器包括：SVM、Logistic Regression、adaboostM1，、KNN、PosKNN和Attribute WKNN。

3.如权利要求1所述的用户退课行为预测方法，其特征在于，步骤S2，包括：

S21，赋予所述训练集中的每个样本同样大小的初始权重值，权重向量记为D；

S22，选择部分弱分类器分别对所述训练集进行训练，并计算各个弱分类器的分类错误率并选择所述分类错误率最低对应的所述弱分类器存储，其中，所述分类错误率是未被正确分类的样本数目占所有数目的比例；

S23，调整所述每个样本的权重值，并在同一个训练集上再次训练，并找到分类错误率最低对应的所述弱分类器存储并记录训练次数t；

S24，当t小于T时，重复执行所述步骤S22和S23，直到t等于T，固定次数为T次，其中，T为正整数，赋予每一个存储的弱分类器一个权重值α，所述弱分类器的错误率为ε，$\alpha=\frac{1}{2}\ln\left(\frac{1-\varepsilon}{\varepsilon}\right)$，预设阈值为p，将所述权重值α大于p对应的弱分类器加权投票，得到最终的强分类器和所述强分类器的权重和参数，其中，p在0至1之间。

4.如权利要求3所述的用户退课行为预测方法，其特征在于，所述权重值的调整原则为：加大/减小被上次存储的分类器分类错误/正确的样本的权重值，并再次找到分类错误率最低的那个分类器存储起来。

5.如权利要求3所述的用户退课行为预测方法，其特征在于，所述训练集为$(x_1,y_1),\ldots,(x_N,y_N)$，其中，$y_i\in\{1,-1\}$，$x_i$为正确的类别标签，所述训练集的样本的初始分布为$D_1(i)=\frac{1}{N}$，其中，i＝1，...，N，计算弱分类器$h_t:X\rightarrow\{-1,1\}$，对t＝1，...，T，其中，T为循环次数，所述弱分类器在分布D_t上的误差为：$\varepsilon_t=P_{D_t}\left(h_t(x_i)\neq y_i\right)$，计算所述弱分类器的权重：$\alpha_t=\frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$，更新所述训练集的样本的分布：$D_{t+1}(i)=\frac{D_t(i)\exp\left(-\alpha_t y_i h_t(x_i)\right)}{Z_t}$，其中$Z_t$为归一化常数，如果$\alpha_t<P$，更新：$\alpha_t=0$，最后的强分类器为：$H_{final}(x)=sign\left(\sum_{t=1}^{T}\alpha_t h_t(x)\right)$。

6.一种用户退课行为预测装置，其特征在于，包括：

获取模块，用于获取基于用户日志的用户行为、基于用户论坛行为的社交关系以及基于时间序列的用户行为作为训练集；

集成学习模块，用于选择部分弱分类器分别对所述训练集进行集成学习得到强分类器和所述强分类器的权重和参数；

测试模块，用于根据所述强分类器的权重和参数对测试集进行测试，输出测试结果。

7.如权利要求6所述的用户退课行为预测装置，其特征在于，所述部分弱分类器包括：SVM、Logistic Regression、adaboostM1，、KNN、PosKNN和Attribute WKNN。

8.如权利要求6所述的用户退课行为预测装置，其特征在于，所述集成学习模块包括：：

S21，赋予所述训练集中的每个样本同样大小的初始权重值，权重向量记为D；

S22，选择部分弱分类器分别对所述训练集进行训练，并计算弱分类器的分类错误率并选择所述分类错误率最低对应的所述弱分类器存储，其中，所述分类错误率是未被正确分类的样本数目占所有数目的比例；

S23，调整所述每个样本的权重值，并在同一个训练集上再次训练，并找到分类错误率最低对应的所述弱分类器存储并记录训练次数t；

S24，当t小于T时，重复执行所述步骤S22和S23，直到t等于T，固定次数为T次，其中，T为正整数，赋予每一个存储的弱分类器一个权重值α，所述弱分类器的错误率为ε，$\alpha=\frac{1}{2}\ln\left(\frac{1-\varepsilon}{\varepsilon}\right)$，预设阈值为p，将所述权重值α大于p对应的弱分类器加权投票，得到最终的强分类器和所述强分类器的权重和参数，其中，p在0至1之间。

9.如权利要求8所述的用户退课行为预测装置，其特征在于，所述权重值的调整原则为：加大/减小被上次存储的分类器分类错误/正确的样本的权重值，并再次找到错误率最低的那个分类器存储起来。

10.如权利要求8所述的用户退课行为预测装置，其特征在于，所述训练集为$(x_1,y_1),\ldots,(x_N,y_N)$，其中，$y_i\in\{1,-1\}$，$x_i$为正确的类别标签，所述训练集的样本的初始分布为$D_1(i)=\frac{1}{N}$，其中，i＝1,...,N，计算弱分类器$h_t:X\rightarrow\{-1,1\}$，对t＝1,...,T，其中，T为循环次数，所述弱分类器在分布D_t上的误差为：$\varepsilon_t=P_{D_t}\left(h_t(x_i)\neq y_i\right)$，计算所述弱分类器的权重：$\alpha_t=\frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$，更新所述训练集的样本的分布：$D_{t+1}(i)=\frac{D_t(i)\exp\left(-\alpha_t y_i h_t(x_i)\right)}{Z_t}$，其中$Z_t$为归一化常数，如果$\alpha_t<P$，更新：$\alpha_t=0$，最后的强分类器为：$H_{final}(x)=sign\left(\sum_{t=1}^{T}\alpha_t h_t(x)\right)$。

用户退课行为预测方法及装置

技术领域

[0001] 本发明涉及网络信息技术领域,尤其涉及一种用户退课行为预测方法及装置。

背景技术

[0002] 高退课率和低参与度是当前的大规模在线公开课程平台面临的重要问题,对潜在退课用户的行为进行细粒度的分析对设计更好的在线教育平台及设计调整课程至关重要。大规模在线开放课程完整的记录了用户和课程进行交互的过程,这提供给研究者一个细粒度分析用户学习行为的前所未有的机会。

[0003] 相关技术中,一种是使用用户的社会行为来预测用户的退课;一种是按照学习模式对用户进行分类;一种是分析了影响用户参与大规模在线公开课程(Mass Open Online Course,MOOC)的关键因素,并且观察到了用户之间显著的行为和学习模式的差异,并且提出了一个基于动态因子图的模型来预测用户的学习表现和证书获得;还有提出了一个隐表示模型用来抽象学习模式并且预测退课。

[0004] 但是,相关技术的研究对于预测结果的准确度还是不够的,我们也注意到高退课率和低参与度是当前的大规模在线公开课程(MOOC)平台面临的重要问题,对潜在退课用户的行为进行细粒度的分析对设计更好的MOOC的平台至关重要。

发明内容

[0005] 本发明的目的旨在至少在一定程度上解决上述的技术问题之一。

[0006] 为此,本发明的第一个目的在于提出一种用户退课行为预测方法。该方法通过集成学习提高了对用户退课行为预测的准确度。

[0007] 本发明的第二个目的在于提出了一种用户退课行为预测装置。

[0008] 为达上述目的,本发明第一方面实施例的用户退课行为预测方法,包括:S1,获取基于用户日志的用户行为、基于用户论坛行为的社交关系以及

[0009] 基于时间序列的用户行为作为训练集;S2,选择部分弱分类器分别对所述训练集进行集成学习得到强分类器和所述强分类器的权重和参数;S3,根据所述强分类器的权重和参数对测试集进行测试,输出测试结果。

[0010] 本发明实施例的用户退课行为预测方法,首先获取基于用户日志的用户行为、基于用户论坛行为的社交关系以及基于时间序列的用户行为作为训练集,接着选择部分弱分类器分别对训练集进行集成学习得到强分类器和强分类器的权重和参数,最后根据强分类器的权重和参数对测试集进行测试,输出测试结果。该方法通过集成学习提高了对用户退课行为预测的准确度。

[0011] 在一些示例中,所述部分弱分类器包括:SVM(Support Vector Machine,支持向量机)、Logistic Regression(虫口模型)、AdaboostM1,、KNN(k-Nearest Neighbor,K最近邻)、PosKNN和Attribute WKNN。

[0012] 在一些示例中,步骤S2,包括:S21,赋予所述训练集中的每个样本同样大小的初始

权重值，权重向量记为D；S22，选择部分弱分类器分别对所述训练集进行训练，并计算各个弱分类器的分类错误率并选择所述分类错误率最低对应的所述弱分类器存储，其中，所述分类错误率是未被正确分类的样本数目占所有数目的比例；S23，调整所述每个样本的权重值，并在同一个训练集上再次训练，并找到分类错误率最低对应的所述弱分类器存储并记录训练次数t；S24，当t小于T时，重复执行所述步骤S22和S23，直到t等于T，固定次数为T次，其中，T为正整数，赋予每一个存储的弱分类器一个权重值α，所述弱分类器的错误率为ε，$\alpha=\frac{1}{2}\ln\left(\frac{1-\varepsilon}{\varepsilon}\right)$，预设阈值为p，将所述权重值α大于p对应的弱分类器加权投票，得到最终的强分类器和所述强分类器的权重和参数，其中，p在0至1之间。

[0013] 在一些示例中，所述权重值的调整原则为：加大/减小被上次存储的分类器分类错误/正确的样本的权重值，并再次找到错误率最低的那个分类器存储起来。

[0014] 在一些示例中，所述训练集为$(x_1,y_1),\ldots,(x_N,y_N)$，其中，$y_i\in\{1,-1\}$，$x_i$为正确的类别标签，所述训练集的样本的初始分布为$D_1(i)=\frac{1}{N}$，其中，i＝1，...，N，计算弱分类器$h_t$：X→{-1，1}，对t＝1，...，T，其中，T为循环次数，所述分类器在分布D_t上的误差为：$\varepsilon_t=P_{D_t}\left(h_t(x_i)\neq y_i\right)$，计算所述弱分类器的权重：$\alpha_t=\frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$，更新所述训练集的样本的分布：$D_{t+1}(i)=\frac{D_t(i)\exp\left(-\alpha_t y_i h_t(x_i)\right)}{Z_t}$，其中$Z_t$为归一化常数，如果$\alpha_t<P$，更新：$\alpha_t=0$，最后的强分类器为：$H_{final}(x)=sign\left(\sum_{t=1}^{T}\alpha_t h_t(x)\right)$。

[0015] 为达上述目的，本发明第二方面实施例的用户退课行为预测装置，包括：获取模块，用于获取基于用户日志的用户行为、基于用户论坛行为的社交关系以及基于时间序列的用户行为作为训练集；集成学习模块，用于选择部分弱分类器分别对所述训练集进行集成学习得到强分类器和所述强分类器的权重和参数；测试模块，用于根据所述强分类器的权重和参数对测试集进行测试，输出测试结果。

[0016] 本发明实施例的用户退课行为预测装置，首先获取模块获取基于用户日志的用户行为、基于用户论坛行为的社交关系以及基于时间序列的用户行为作为训练集，接着集成学习模块选择部分弱分类器分别对训练集进行集成学习得到强分类器和强分类器的权重和参数，最后测试模块根据强分类器的权重和参数对测试集进行测试，输出测试结果。该装置通过集成学习提高了对用户退课行为预测的准确度。

[0017] 在一些示例中，所述部分弱分类器包括：SVM(Support Vector Machine，支持向量机)、Logistic Regression(虫口模型)、adaboostM1，、KNN(k-Nearest Neighbor，K最近邻)、PosKNN和Attribute WKNN。

[0018] 在一些示例中，所述集成学习模块包括：：S21，赋予所述训练集中的每个样本同样大小的初始权重值，权重向量记为D；S22，选择部分弱分类器分别对所述训练集进行训练，并计算弱分类器的分类错误率并选择所述分类错误率最低对应的所述弱分类器存储，其中，所述分类错误率是未被正确分类的样本数目占所有数目的比例；S23，调整所述每个样

本的权重值，并在同一个训练集上再次训练，并找到分类错误率最低对应的所述弱分类器存储并记录训练次数t；S24，当t小于T时，重复执行所述步骤S22和S23，直到t等于T，固定次数为T次，其中，T为正整数，赋予每一个存储的弱分类器一个权重值α，所述弱分类器的错误率为ε，$\alpha=\frac{1}{2}\ln\left(\frac{1-\varepsilon}{\varepsilon}\right)$，预设阈值为p，将所述权重值α大于p对应的弱分类器加权投票，得到最终的强分类器和所述强分类器的权重和参数，其中，p在0至1之间。

[0019] 在一些示例中，所述权重值的调整原则为：加大/减小被上次存储的分类器分类错误/正确的样本的权重值，并再次找到错误率最低的那个分类器存储起来。

[0020] 在一些示例中，所述训练集为$(x_1,y_1),\ldots,(x_N,y_N)$，其中，$y_i\in\{1,-1\}$，$x_i$为正确的类别标签，所述训练集的样本的初始分布为$D_1(i)=\frac{1}{N}$，其中，i＝1，...，N，计算弱分类器$h_t:X\rightarrow\{-1,1\}$，对t＝1，...，T，其中，T为循环次数，所述分类器在分布D_t上的误差为：$\varepsilon_t=P_{D_t}\left(h_t(x_i)\neq y_i\right)$，计算所述弱分类器的权重：$\alpha_t=\frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$，更新所述训练集的样本的分布：$D_{t+1}(i)=\frac{D_t(i)\exp\left(-\alpha_t y_i h_t(x_i)\right)}{Z_t}$，其中$Z_t$为归一化常数，如果$\alpha_t<P$，更新：$\alpha_t=0$，最后的强分类器为：$H_{final}(x)=sign\left(\sum_{t=1}^{T}\alpha_t h_t(x)\right)$。

[0021] 本发明附加的方面和优点将在下面的描述中部分给出，部分将从下面的描述中变得明显，或通过本发明的实践了解到。

附图说明

[0022] 本发明的上述和/或附加的方面和优点从结合下面附图对实施例的描述中将变得明显和容易理解，其中：

[0023] 图1是根据本发明一个实施例的用户退课行为预测方法的流程图；

[0024] 图2是根据本发明一个实施例的状态机的示意图；

[0025] 图3是根据本发明一个实施例的集成学习方法的流程图；

[0026] 图4是根据本发明一个实施例的用户退课行为预测装置的示意图。

具体实施方式

[0027] 下面详细描述本发明的实施例，所述实施例的示例在附图中示出，其中自始至终相同或类似的标号表示相同或类似的元件或具有相同或类似功能的元件。下面通过参考附图描述的实施例是示例性的，旨在用于解释本发明，而不能理解为对本发明的限制。

[0028] 首先，预测只针对该课程的选课用户。由于课程间的差异性较大，各个课程的训练模型参数和测试是相互独立的。模型中，正例为1，表示用户在预测时间内会继续访问课程；负例为0，表示用户会流失。统计证明，那些在预测时间段内(在本模型中即为预测前一个月)没来访问课程的选课用户，有99％在接下来的10天没来，因此本模型只对在预测前一个月内有行为的选课用户使用算法预测，其余选课用户则默认已流失，即流失率为100％。

[0029] 图1是根据本发明一个实施例的用户退课行为预测方法的流程图。

[0030] 如图1所示，该用户退课行为预测方法可以包括：

[0031] S1，获取基于用户日志的用户行为、基于用户论坛行为的社交关系以及基于时间序列的用户行为作为训练集。

[0032] 需要说明的是，每条特征向量对应于一门课程的选课用户。特征向量主要用到是基于用户日志的用户行为、基于用户论坛行为的社交关系以及基于时间序列的用户行为。

[0033] 可以理解的是，基于用户日志的用户行为可以是从用户日志中解析出来的行为模型。例如，从用户日志中解析出来的行为类型；我们设计了一个基于自动机的用户日志分析算法。如图2所示，状态机由三个状态组成空闲(Idle)、视频(Video)和作业(Assignment)，我们假设用户在任何时刻都处于这三个状态中的一个。举例来说，当用户点击播放课程视频时，他的状态就会转移到视频状态，而当用户点击暂停时，他的状态又会跳回到空闲状态。我们还设置了一个时间阈值，当用户长处于视频或者作业状态时超过这个时间阈值时，他的状态就会回到空闲。此外，我们还从用户的访问log日志中提取出的行为类型主要有以下7种：Video,Page_Close,Problem,Access,Discussion,Wiki,Navigate。在过滤掉一些无效的用户访问行为后，统计用户在预测前一个月内的各种行为类型的次数，按照时间依次罗列。无效访问行为定义：如果在访问记录中，如果用户访问某一页面的持续时间太短，如用户的连续两次访问记录间隔<＝5秒，则认为用户前一次访问无效。

[0034] 可以理解的是，基于用户论坛行为的社交关系可以是用户论坛“朋友”的访问情况。其中，“朋友”的定义是曾经和该用户在论坛里有过交流的用户。例如，统计在预测前10天内访问过该课程的用户“朋友”数。

[0035] 可以理解的是，基于时间序列的用户行为可以是用户访问时间模式和估计。例如，统计用户访问的课程的总天数count以及平均访问时间频率interval，并依照该用户最后访问的时间以及最近的访问时间间隔推算该用户下一次访问的时间，计算该时间和预测时间段的偏差bias。

[0036] S2，选择部分弱分类器分别对训练集进行集成学习得到强分类器和强分类器的权重和参数。

[0037] 在一些示例中，部分弱分类器包括：SVM(Support Vector Machine，支持向量机)、Logistic Regression(虫口模型)、adaboostM1，、KNN(k-Nearest Neighbor，K最近邻)、PosKNN和Attribute WKNN。

[0038] 需要说明的是，SVM、Logistic regression、adaboostM1、KNN这四种模型都是现有的经典模型，在此不再赘述。而PosKNN和Attribute WKNN则是在经典KNN算法的基础上进行了改进的算法。因此接下来重点介绍模型的前提假设、所用特征的选取、所用的改进算法以及boosting集成过程。

[0039] 例如，PosKNN算法原理是：分别选取与训练纪录汉明码距离最近的前k个正例和前k个负例，然后计算这k个正例和k个负例与训练纪录的平均距离，如果正(负)例的平均距离较小，则最终分类结果就为正(负)例。这个算法原理较简单，过程和经典KNN相差无几，在此不再赘述。

[0040] 例如，Attribute WKNN算法我们一直致力于设计一个既准确又快速的分类算法，其中一种很简单但非常准确的分类算法就是上文所述的KNN(k-nearest neighbor

classification)算法。但是KNN算法也有其不足之处，当在对属性较多的训练样本进行分类时，由于假定各个属性权重相同，而现实中却往往不是如此，此时KNN的分类精度就会大大降低，效果不是很理想。因此，为了弥补这一缺陷，本文提出一种基于特征属性加权的KNN改进算法——Attribute WKNN(attribute weight k-nearest neighbor classification)算法。该算法考虑到，由于在实际生活中，不同的特征对于最终的分类结果的贡献是不可能会完全同等的，因而给各个特征赋予了不同大小的权重。这样一来，就能使重要特征的作用得到提升，从而提高了算法的性能。更具体而言，Attribute WKNN算法的基本思想是，为每个特征属性训练出一个合理的权重值，然后根据特征属性的相似性，找到与新的数据属性最为相似的k条训练数据，然后对于新数据的分类决策则是依赖于这k条记录的类别的。大概过程如下：第一步，给每个特征属性一个初始权重值W，并将训练数据集TA均分成为N个小数据集N_K，然后进行N次交叉验证。即对于每一个数据子集N_K(K＝1,…,N)，将其作为测试集TE_K，而其余的N-1个子集则汇集成一个新训练集TA_K。TA_K通过内部循环处理来训练特征属性权重参数W，最后测试TE_K在TA_K上的准确率。第二步，在整个训练集TA上进行一遍内部循环处理训练特征属性权重参数W。

[0041] 为了本领域人员更加清楚Attribute WKNN算法，下面详细说明：对于一条新测试记录X，有：第一距离：可以是欧氏距离，余弦距离等。在本实施例中，对训练集中的一条记录Y，X和Y之间的距离∂定义为：

[0042] $$\partial = \sqrt{\sum_{i=1}^{k}\left[w_i\left(x_i - y_i\right)\right]^2} \qquad (1)$$

[0043] 其中，k为特征的数目，w_i为第i个特征的权重值，x_i和y_i分别是记录X和Y在第i个特征上的取值。第二K邻居：指的是与X距离最为接近的K条记录。第三投票分类法：平均投票：每个邻居的权重相同，并且少数服从多数，哪个类别多就分为哪类。加权投票法：基于相似度的大小，给每个邻居的分类权重值是不一样的。第四Gradient Descent(梯度下降)：该方法的运行原理是：沿着递度下降/上升的方向求解极小/大值。为了更加清晰的阐述其原理，可以将函数看做一座山。而位置就处于山坡的顶端，思考从哪个方向可以最快的下山。方法：确定Learning rate。即向下一步的步伐大小；任意给定一个初始值：θ_0,θ_1；确定好一个方向，并按照上述的步伐大小向下走，然后更新θ_0,θ_1；当下降高度低于了所设置的门槛值时，就终止；核心算法如下所示，其中，convergence是终止条件，α是Learning rate，它决定了下降的步伐大小，太小函数收敛很慢，太大则可能无法找到极值，甚至函数无法收敛。$\frac{\partial}{\partial\theta_j}J(\theta_0,\theta_1)$决定了下降的方向。

[0044]

Repeat until convergence {

$$\theta_j := \theta_j - \alpha\frac{\partial}{\partial\theta_j}J(\theta_0,\theta_1)$$

}

[0045] 其中，特点：类似于贪心算法，只能求得一个局部最优值；随着下降高度对门槛值

的逐渐逼近，下降速度也会逐渐变慢。第五交叉验证，交叉验证(Cross validation)，也称为循环估计。是一种首先将数据集均分成若干个子集的算法。可以设置最开始的那个子集为训练集X，它被当做分析的对象；而其余的子集则成为测试集Y，它用来做验证。我们可以用交叉验证来评判学习方法的泛化能力(generalize)。该方法需要尽可能满足以下条件：训练数据集的比重够大，一般在50％以上；对训练集和测试集的抽样要均匀。交叉验证主要有：

[0046] k-folder cross-validation:将数据集等分为k个子集，对每一个子集X，都有：X当做测试集，而其余子集则成为训练集。这样重复k次交叉验证后，将k次测试集的测试准确率的均值作为最终结果。优点：所有的样本都被验证一次，且曾被用作为训练集以及测试集。10-folder通常被使用。或者是K*2 folder cross-validation是k-folder cross-validation的一个变体，对每一个folder，都平均分成两个集合S0，S1。我们先将S0作训练S1作测试，然后反过来。优点：所有的样本都能被用作训练集和测试集，且测试集和训练集的规模数量足够的大。一般使用k＝10。least-one-out cross-validation(loocv)，假设数据集中有N个样本，那LOOCV也就是N-CV。对每个样本X都有：X作一次测试集，而剩余N-1个则为训练集。优点：几乎所有的样本用来训练模型，这样可以有效的降低泛化误差。实验过程可复制。缺点：当样本总数过多时，由于很高的计算成本，LOOCV在实际操作的效率会大大降低。

[0047] 例如，Attribute WKNN算法数据集如表1所示：

[0048] 表1数据集

[0049]

	feature	class	T	feature
d1	1,2,1	1	t1	0,0,2
d2	0,1,0	0	t2	1,3,0
d3	1,3,2	2		
d4	0,0,1	1		
d5	3,0,1	0		
d6	1,3,0	2		

[0050] 训练数据集为D。共有6条纪录：d1，d2，d3，d4，d5，d6。测试数据集为T。共有2条记录：t1，t2。每条记录均有3个特征属性：f1，f2，f3。其对应的值为v1，v2，v3。初始参数设置：学习比率α＝0.2，“邻居”数k＝1，训练子集总数N＝3，特征属性对应权重w1、w2、w3分别初始化值为0.2、0.5、0.1。算法具体过程如下：交叉验证训练，将训练集D均分为N＝3个子训练集D1(d1，d2)、D2(d3，d4)和D3(d5，d6)对每个子训练集有如下过程(以D2为例)：设置D2为验证集，新训练集D(1+3)＝D1+D3，即(d1，d2，d5，d6)，对于D(1+3)里的每一个d，有(以d2为例)：计算在D(1+3)里除了d2的剩余记录(d1，d5，d6)与d2的距离：

$$\theta(d2,d1)=\sqrt{\sum_{i=1}^{3}\left[w_i\left(v_{2,i}-v_{1,i}\right)\right]^2}=\sqrt{\left[0.2(0-1)\right]^2+\left[0.5(1-2)\right]^2+\left[0.1(0-1)\right]^2}\approx 0.5477$$ (保留四位小数)。

同理可得，θ(d2，d5)≈0.7874，θ(d2，d6)≈1.0198。找到d2的k(k＝1)邻居，由上可知与d2距离最小的是d1，返回k邻居投票值，因为k＝1，所以投票值就为d1的分类值1，由于d2的真实分类值0≠投票值1，所以用梯度下降法修改特征权重：Error＝真实分类值-投票值＝0-1

=-1,对每个特征权重w_i,有:$w_1=w_1+\alpha*Error*V1=0.2+0.2*(-1)*0=0.2$。同理可得$w_2=0.3$,$w_3=0.1$,对验证集D2里的每一个d,有:找到与d距离最小的k邻居并得到投票分类值;计算分类准确率A:$A=\frac{正确被分类数}{T总记录数}*100\%$。在整个训练集上训练对训练集D里的记录d,有:找到与d距离最小的k邻居并得到投票分类值;如果真实分类值≠投票值,则用梯度下降法修改特征权重:Error=真实分类值-投票值对每个特征属性权重w_k,有:$w_k=w_k+\alpha*Error*Vk$。对整个训练集D里的d,有:找到与d距离最小的k邻居并得到投票分类值;计算分类准确率A:$A=\frac{正确被分类数}{T总记录数}*100\%$,测试数据集T对T里的每一个d,有:找到与d距离最小的k邻居并得到投票分类值,计算分类准确率A,$A=\frac{正确被分类数}{T总记录数}*100\%$。

[0051] 其中,如图3所示,在一些示例中,步骤S2,包括:

[0052] S21,赋予训练集中的每个样本同样大小的初始权重值,权重向量记为D。

[0053] S22,选择部分弱分类器分别对训练集进行训练,并计算各个弱分类器的分类错误率并选择分类错误率最低对应的弱分类器存储,其中,分类错误率是未被正确分类的样本数目占所有数目的比例。

[0054] S23,调整每个样本的权重值,并在同一个训练集上再次训练,并找到分类错误率最低对应的弱分类器存储并记录训练次数t。

[0055] S24,当t小于T时,重复执行步骤S22和S23,直到t等于T,固定次数为T次,其中,T为正整数,赋予每一个存储的弱分类器一个权重值α,弱分类器的错误率为ε,$\alpha=\frac{1}{2}\ln\left(\frac{1-\varepsilon}{\varepsilon}\right)$,预设阈值为p,将权重值α大于p对应的弱分类器加权投票,得到最终的强分类器和强分类器的权重和参数,其中,p在0至1之间。

[0056] 在一些示例中,权重值的调整原则为:加大/减小被上次存储的分类器分类错误/正确的样本的权重值,并再次找到分类错误率最低的那个分类器存储起来。

[0057] 更具体地,在一些示例中,训练集为$(x_1,y_1),\ldots,(x_N,y_N)$,其中,$y_i\in\{1,-1\}$,$x_i$为正确的类别标签,训练集的样本的初始分布为$D_1(i)=\frac{1}{N}$,其中,i=1,...,N,计算弱分类器$h_t:X\rightarrow\{-1,1\}$,对t=1,...,T,其中,T为循环次数,分类器在分布D_t上的误差为:$\varepsilon_t=P_{D_t}\left(h_t(x_i)\neq y_i\right)$,计算弱分类器的权重:$\alpha_t=\frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$,更新训练集的样本的分布:$D_{t+1}(i)=\frac{D_t(i)\exp\left(-\alpha_t y_i h_t(x_i)\right)}{Z_t}$,其中$Z_t$为归一化常数,如果$\alpha_t<P$,更新:$\alpha_t=0$,最后的强分类器为:$H_{final}(x)=sign\left(\sum_{t=1}^{T}\alpha_t h_t(x)\right)$。

[0058] 其中,在计算中的代码表示可以如下表示:

[0059]

输入：数据集 $D=\{(x_1,y_1),...,(x_N,y_N)\}$，初始弱分类器 $\Im$，循环次数 T。

输出：所选择的分类器以及与其对应的权重和其他相关参数。

Process:

$D_1(i)=\frac{1}{N}$ //*初始化权重分布

for $t=1,...,T$:

$h_t=\Im(D,D_t)$;//在分布 D_t 上训练，并得到错误率最低的分类器 h_t

$\varepsilon_t=P_{D_t}(h_t(x_i)\neq y_i)$;//计算错误率

$\alpha_t=\frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$;//计算该分类器权重

$D_{t+1}(i)=\frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$;//更新训练样本分布

for $t=1,...,T$:

if ($\alpha_t<P$)//去除错误率没达到阈值的

$\alpha_t=0$;

[0060]

End

[0061] S3，根据强分类器的权重和参数对测试集进行测试，输出测试结果。

[0062] 举例而言，可以选择xuetangX上的财务分析与决策(2014春)课程测试，得到各模型测试结果如表2所示：

[0063] 表2各模型测试结果

[0064]

模型	precision	accuracy	recall	f1
SVM	74.43	72.78	92.39	82.44
Logistic regression	75.34	72.41	89.35	81.75
adaboostM1	76.41	71.05	84.13	80.08
KNN（best k =17）	75.22	68.95	82.17	78.54
Pos KNN（best k =20）	75.32	68.95	81.96	78.5
Attribute WKNN （best k =17）	75.36	72.63	89.78	81.94
集成学习	75.81	74.14	91.96	83.11

[0065] 从表2可以看出，通过集成学习，的确改善了预测结果。

[0066] 本发明实施例的用户退课行为预测方法，首先获取基于用户日志的用户行为、基于用户论坛行为的社交关系以及基于时间序列的用户行为作为训练集，接着选择部分弱分类器分别对训练集进行集成学习得到强分类器和强分类器的权重和参数，最后根据强分类器的权重和参数对测试集进行测试，输出测试结果。该方法通过集成学习提高了对用户退课行为预测的准确度。

[0067] 与上述实施例提供的用户退课行为预测方法相对应，本发明的一种实施例还提供一种用户退课行为预测装置，由于本发明实施例提供的用户退课行为预测装置与上述实施例提供的用户退课行为预测方法具有相同或相似的技术特征，因此在前述用户退课行为预测方法的实施方式也适用于本实施例提供的用户退课行为预测装置，在本实施例中不再详细描述。如图4所示，该用户退课行为预测装置可包括：获取模块10、集成学习模块20和测试模块30。

[0068] 其中，获取模块10用于获取基于用户日志的用户行为、基于用户论坛行为的社交关系以及基于时间序列的用户行为作为训练集。

[0069] 集成学习模块20用于选择部分弱分类器分别对训练集进行集成学习得到强分类器和强分类器的权重和参数。

[0070] 测试模块30用于根据强分类器的权重和参数对测试集进行测试，输出测试结果。

[0071] 在一些示例中，部分弱分类器包括：SVM、Logistic Regression、adaboostM1，、KNN、PosKNN和Attribute WKNN。

[0072] 在一些示例中，集成学习模块20包括：：S21，赋予训练集中的每个样本同样大小的初始权重值，权重向量记为D；S22，选择部分弱分类器分别对训练集进行训练，并计算弱分类器的分类错误率并选择分类错误率最低对应的弱分类器存储，其中，分类错误率是未被正确分类的样本数目占所有数目的比例；S23，调整所述每个样本的权重值，并在同一个训

练集上再次训练，并找到分类错误率最低对应的所述弱分类器存储并记录训练次数t；S24，当t小于T时，重复执行所述步骤S22和S23，直到t等于T，固定次数为T次，其中，T为正整数，赋予每一个存储的弱分类器一个权重值α，所述弱分类器的错误率为ε，$\alpha=\frac{1}{2}\ln\left(\frac{1-\varepsilon}{\varepsilon}\right)$，预设阈值为p，将所述权重值α大于p对应的弱分类器加权投票，得到最终的强分类器和所述强分类器的权重和参数，其中，p在0至1之间。

[0073] 在一些示例中，权重值的调整原则为：加大/减小被上次存储的分类器分类错误/正确的样本的权重值，并再次找到错误率最低的那个分类器存储起来。

[0074] 在一些示例中，训练集为$(x_1,y_1),\ldots,(x_N,y_N)$，其中，$y_i\in\{1,-1\}$，$x_i$为正确的类别标签，训练集的样本的初始分布为$D_1(i)=\frac{1}{N}$，其中，i＝1,...,N，计算弱分类器$h_t:X\rightarrow\{-1,1\}$，对t＝1,...,T，其中，T为循环次数，弱分类器在分布D_t上的误差为：$\varepsilon_t=P_{D_t}\left(h_t(x_i)\neq y_i\right)$，计算弱分类器的权重：$\alpha_t=\frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$，更新训练集的样本的分布：$D_{t+1}(i)=\frac{D_t(i)\exp\left(-\alpha_t y_i h_t(x_i)\right)}{Z_t}$，其中$Z_t$为归一化常数，如果$\alpha_t<P$，更新：$\alpha_t=0$，最后的强分类器为：$H_{final}(x)=sign\left(\sum_{t=1}^{T}\alpha_t h_t(x)\right)$。

[0075] 本发明实施例的用户退课行为预测装置，首先获取模块获取基于用户日志的用户行为、基于用户论坛行为的社交关系以及基于时间序列的用户行为作为训练集，接着集成学习模块选择部分弱分类器分别对训练集进行集成学习得到强分类器和强分类器的权重和参数，最后测试模块根据强分类器的权重和参数对测试集进行测试，输出测试结果。该装置通过集成学习提高了对用户退课行为预测的准确度。

[0076] 在本发明的描述中，需要理解的是，术语“第一”、“第二”仅用于描述目的，而不能理解为指示或暗示相对重要性或者隐含指明所指示的技术特征的数量。由此，限定有“第一”、“第二”的特征可以明示或者隐含地包括至少一个该特征。在本发明的描述中，“多个”的含义是至少两个，例如两个，三个等，除非另有明确具体的限定。

[0077] 在本说明书的描述中，参考术语“一个实施例”、“一些实施例”、“示例”、“具体示例”、或“一些示例”等的描述意指结合该实施例或示例描述的具体特征、结构、材料或者特点包含于本发明的至少一个实施例或示例中。在本说明书中，对上述术语的示意性表述不必须针对的是相同的实施例或示例。而且，描述的具体特征、结构、材料或者特点可以在任一个或多个实施例或示例中以合适的方式结合。此外，在不相互矛盾的情况下，本领域的技术人员可以将本说明书中描述的不同实施例或示例以及不同实施例或示例的特征进行结合和组合。

[0078] 流程图中或在此以其他方式描述的任何过程或方法描述可以被理解为，表示包括一个或更多个用于实现特定逻辑功能或过程的步骤的可执行指令的代码的模块、片段或部分，并且本发明的优选实施方式的范围包括另外的实现，其中可以不按所示出或讨论的顺序，包括根据所涉及的功能按基本同时的方式或按相反的顺序，来执行功能，这应被本发明的实施例所属技术领域的技术人员所理解。

[0079] 本技术领域的普通技术人员可以理解实现上述实施例方法携带的全部或部分步

骤是可以通过程序来指令相关的硬件完成，所述的程序可以存储于一种计算机可读存储介质中，该程序在执行时，包括方法实施例的步骤之一或其组合。

[0080] 尽管上面已经示出和描述了本发明的实施例，可以理解的是，上述实施例是示例性的，不能理解为对本发明的限制，本领域的普通技术人员在本发明的范围内可以对上述实施例进行变化、修改、替换和变型。

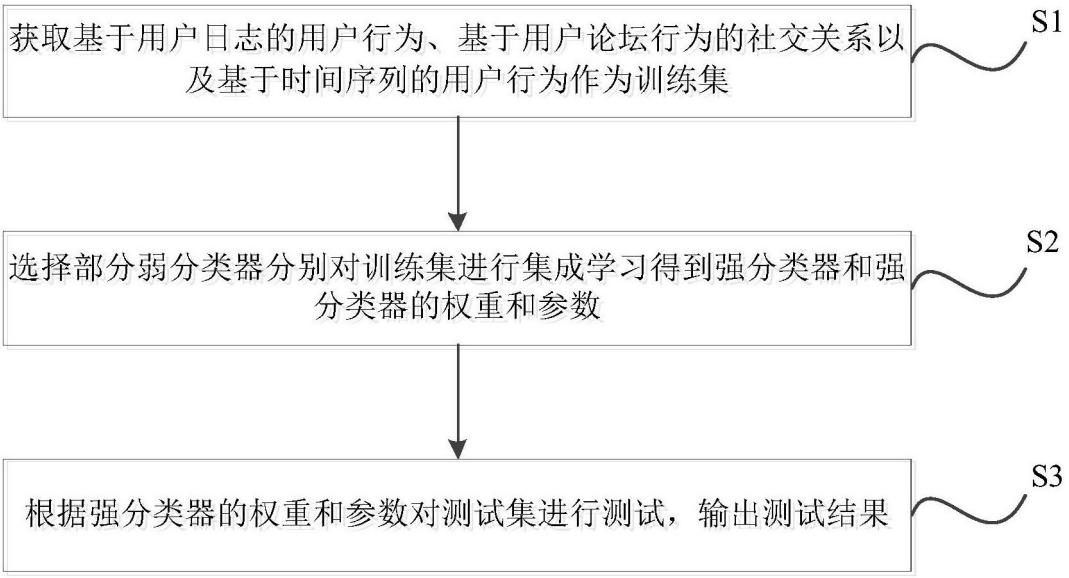

获取基于用户日志的用户行为、基于用户论坛行为的社交关系以及基于时间序列的用户行为作为训练集
S1
选择部分弱分类器分别对训练集进行集成学习得到强分类器和强分类器的权重和参数
S2
根据强分类器的权重和参数对测试集进行测试，输出测试结果
S3

图1

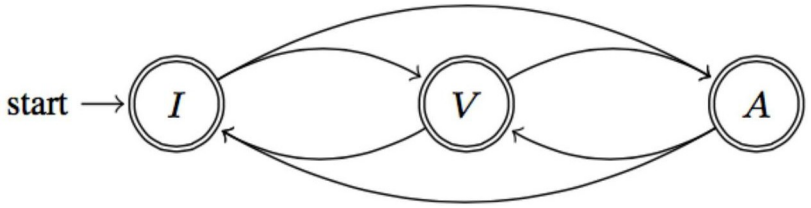

start
I
V
A

图2

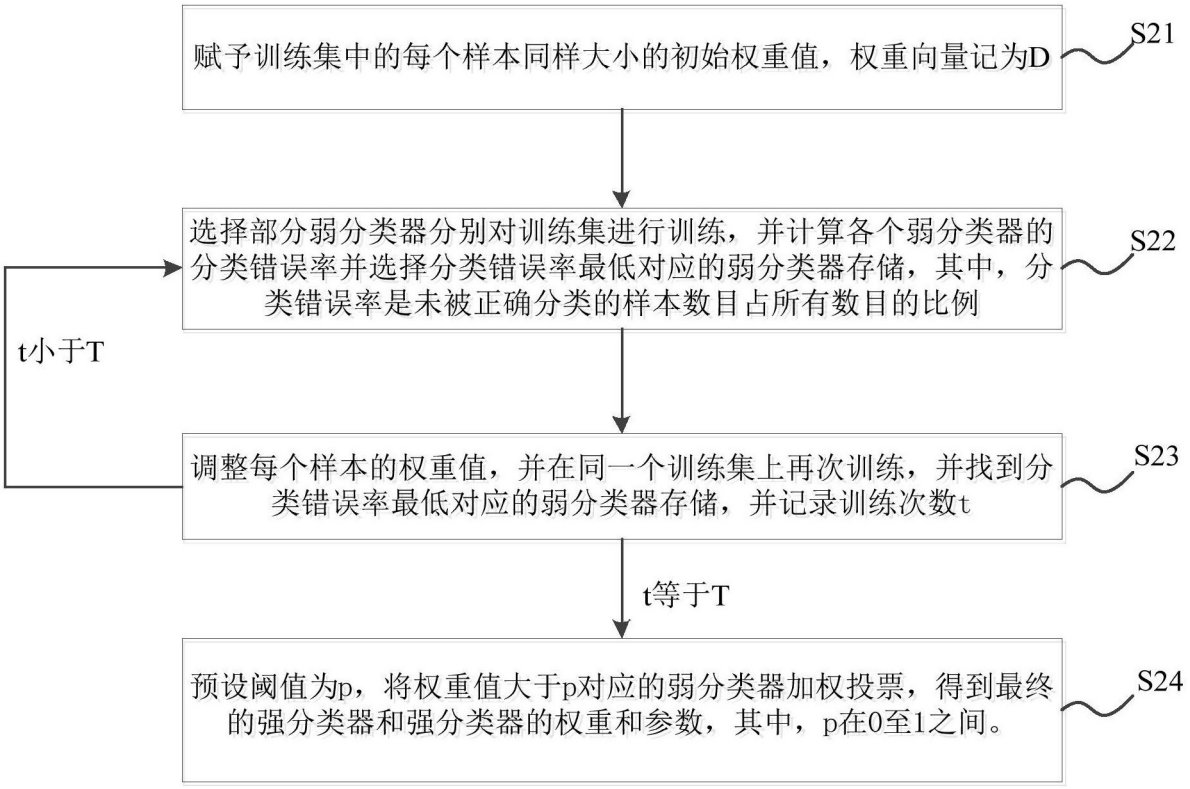

赋予训练集中的每个样本同样大小的初始权重值，权重向量记为D
S21
选择部分弱分类器分别对训练集进行训练，并计算各个弱分类器的分类错误率并选择分类错误率最低对应的弱分类器存储，其中，分类错误率是未被正确分类的样本数目占所有数目的比例
S22
t小于T
调整每个样本的权重值，并在同一个训练集上再次训练，并找到分类错误率最低对应的弱分类器存储，并记录训练次数t
S23
t等于T
预设阈值为p，将权重值大于p对应的弱分类器加权投票，得到最终的强分类器和强分类器的权重和参数，其中，p在0至1之间。
S24

图3


获取模块10
集成学习模块20
测试模块30

图4