文章编号: 1000-6788(2010)10-1855-09 中图分类号: TP311 文献标志码: A

基于支持向量机的分布数据挖掘模型 DSVM

琚春华[1,2], 郭飞鹏[3]

(1. 浙江工商大学 计算机与信息工程学院, 杭州 310018; 2. 浙江工商大学 现代商贸研究中心, 杭州 310018;
3. 浙江经贸职业技术学院 信息技术系, 杭州 310018)

摘 要 针对分布环境的数据挖掘要求, 提出了基于支持向量机的分布数据挖掘模型 DSVM. 定义了 DSVM 中特征多叉树的概念, 描述了基于移动 Agent 访问分布数据集来构建特征多叉树的方法, 阐述了通过特征多叉树来反映分布环境各数据集属性总体特征的思想, 并利用该数据结构和支持向量机的特点, 提出了基于壳向量的分布式支持向量机增量算法来修正和完善特征多叉树, 最终实现分布环境下全局的数据挖掘. 实验结果表明, 该模型有效地解决原有分布环境下其他挖掘算法存储开销大、执行效率差、安全性和隐私性低等问题.

关键词 分布数据挖掘; 支持向量机; 特征多叉树; 移动 Agent

Distributed data mining model based on Support Vector Machines

JU Chun-hua[1,2], GUO Fei-peng[3]

(1. Computer and Information Engineering College, Zhejiang Gongshang University, Hangzhou 310018, China;
2. Center for Studies of Modern Business, Zhejiang Gongshang University, Hangzhou 310018, China;
3. Information Technology Department, Zhejiang Economic & Trade Polytechnic, Hangzhou 310018, China)

Abstract The paper presented a distributed data mining model based on Support Vector Machines DSVM. It described the definition of multi-branches tree of Eigen (ET) and the method of building ET based on mobile Agents accessing to distributed datasets. It elaborated the concept by using ET to reflect the characteristic of attribute in the distributed dataset, and then proposed the algorithm of distributed incremental Support Vector Machines based on hull vector (HDIS) using the data structure of ET and the feature of Support Vector Machine. Finally, the ET was modified and improved by HDIS to realize distributed data mining. The experimental results show the DSVM providing high capability and efficiency of distributed business data mining.

Keywords distributed data mining; support vector machine; multi-branches tree of eigen; mobile Agent

1 引言

随着企业网络化信息系统的发展, 企业数据库的集中管理已不能满足实际需求, 受地域空间、时间等的影响, 企业的数据库逐步从集中管理发展到分布管理. 数据库分布于各个门店, 并且随着时间的推移, 其数据也日益增长, 使数据具有分布、异构、海量等特点, 这给数据挖掘提出了严峻的考验[1].

目前, 已有众多的数据挖掘算法, 如神经网络[2]、贝叶斯网络[3]、决策树[4] 等, 用于客户分类、客户流失预测等应用. 但上述挖掘算法在不同程度上具有以下两方面的缺点: 第一, 在处理大规模、高维度、含有非线性关系的数据时效果不理想; 第二, 主要依靠的是经验风险最小化原则, 容易导致泛化能力的下降且模型结构难以确定[5]. 基于结构风险最小化准则的 SVM 算法是少数可以成功解决上述问题的学习算法之一[6]. 另外, 现有的大部分方法在分布环境下挖掘不仅会大量占用存储空间, 增加网络负担, 而且使响应时间变长[7−8].

收稿日期: 2009-09-02
资助项目: 国家自然科学基金 (71071141); 浙江省自然科学基金重点项目 (Z1091224)
作者简介: 琚春华 (1962–), 男, 博士生导师, 教授, 研究方向为人工智能、电子商务等; 郭飞鹏 (1984–), 男, 硕士研究生, 助教, 研究方向为电子商务、数据挖掘.

田大新等[9] 处理大规模数据有着良好的性能, 但未充分考虑数据传输过程中的私有性和安全性. 因此, 研究 SVM 算法的分布式学习机制和挖掘模型具有重要的理论意义和实用价值.

2 相关理论研究

2.1 支持向量机

支持向量机 (SVM) 是基于 VC 理论的机器学习方法. SVM 分类模型的基本思想是构造一个超平面作为决策平面, 使正负模式之间的距离最大. 支持向量机是从线性可分情况下的最优分类面发展而来的, 也是统计学习理论中最实用的部分. 所谓最优分类面就是要求分类面不但能将两类正确分开, 而且使分类间隔最大. 距离最优分类超平面最近的向量称为支持向量[10].

最优分割超平面的计算问题可以描述为一个条件极值问题, Vapnik 通过求解下列 Lagrange 函数的鞍点获得其最优解:

$$L_p = \frac{1}{2}\|w\|^2 + C\sum_i \varepsilon_i - \sum_i \alpha_i \{y_i (\Phi(x_i)\cdot w + b) - 1 + \varepsilon_i\} - \sum_i \mu_i \varepsilon_i \tag{1}$$

其中 w 和 b 分别为属性空间最优超平面的法向量和阈值, ε_i, μ_i 分别为非负的 Lagrange 乘数, C 为非负的误差控制参数, $\Phi(x)$ 为输入空间向属性空间映射的函数.

由 K-K-T 定理可知, 最优解满足以下条件:

$$w = \sum_i \alpha_i y_i \Phi(x_i) \tag{2}$$

$$0 \le \alpha_i \le C, \quad \sum_i \alpha_i y_i = 0 \tag{3}$$

一般情况下, 在式 (2) w 的展开式中, 大多数系数 α_i 为零值, 并不影响分类的结果. 而对 w 的确定有贡献的仅仅是非零值的 α_i 所对应的 x_i, 这就是支持向量 (Support vector, SV)[11]. 因此, SV 集充分描述了整个训练数据集数据的特征, 对 SV 集的划分等价于对整个数据集的分割. 在大多数情况下, 训练集中 SV 的数量只占训练样本集的很少一部分. 因此可以使用 SV 集代替训练样本集进行分类学习, 在不影响分类精度的同时极大地减少训练时间.

2.2 增量学习算法与分布数据挖掘

增量学习是将新增的训练样本作为增量, 对在原训练样本集上训练得到的分类器进行训练从而使新得到的分类器能够对原训练样本集和新增训练样本集均能很好地分类. 与传统的学习方法相比, 增量学习算法可以充分利用历史学习的结果, 从而减少后继训练时间, 无须保存历史数据, 从而减少了存储空间的约束. 目前, 已有学者对基于 SVM 的增量学习算法进行了研究, 主要集中在如何逐步提高增量学习精度和如何提高增量学习计算速度两个方面[12−13].

在 SVM 增量学习中, 对每一类新增训练集而言, 支持向量集合是壳向量集的子集. 如图 1 所示, 实心和空心圆点用 A、B 表示, 代表分属两个不同类别的训练集, 则图中实心圆点侧凸边型边上的样本为训练集 A 的壳向量, 其中 H_2 上的两点为支持向量; 空心圆点侧凸边型边上的样本为训练集 B 的壳向量, 其中 H_1 上的两点为支持向量. H 为 SVM 最优分类超平面, 只有穿过位于凸壳顶点样本的分类面才有可能对两类样本正确分类, 支持向量就在其中产生, 而不可能是位于凸壳内部的样本点[14].

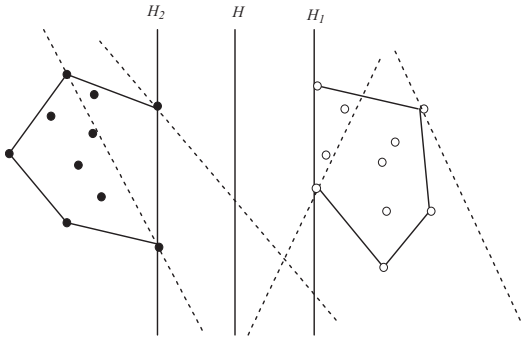

图 1 壳向量与支持向量的关系

对于分布的数据挖掘, 多 Agent 技术目前应用较为广泛[15]. 针对企业分布经营所形成的网络化数据库, 利用 Agent 对分布数据源特征属性的映射和贝叶斯网络学习方法, 建立面向企业经营分析的分布数据挖掘模型, 支持企业进行客户流失预测、价格确定、客户分类等决策活动[16].

3　基于支持向量机的分布数据挖掘模型

DSVM 的主要思想是: 首先对各分站点数据集进行 SVM 局部挖掘; 然后利用特征多叉树构建算法将局部挖掘出的支持向量映射成局部特征多叉树, 通过移动 Agent 将支持向量和壳向量信息装载到下一个站点, 再将新增样本 (前几个站点的壳向量集) 与已有样本 (下一个站点的样本集) 合并后挖掘 (HDIS), 并随着样本集的积累 (各个站点的移动) 逐步提高学习精度; 最终实现分布环境下支持向量机的全局挖掘. 图 2 所示为基于支持向量机的分布数据挖掘模型 DSVM 的体系架构.

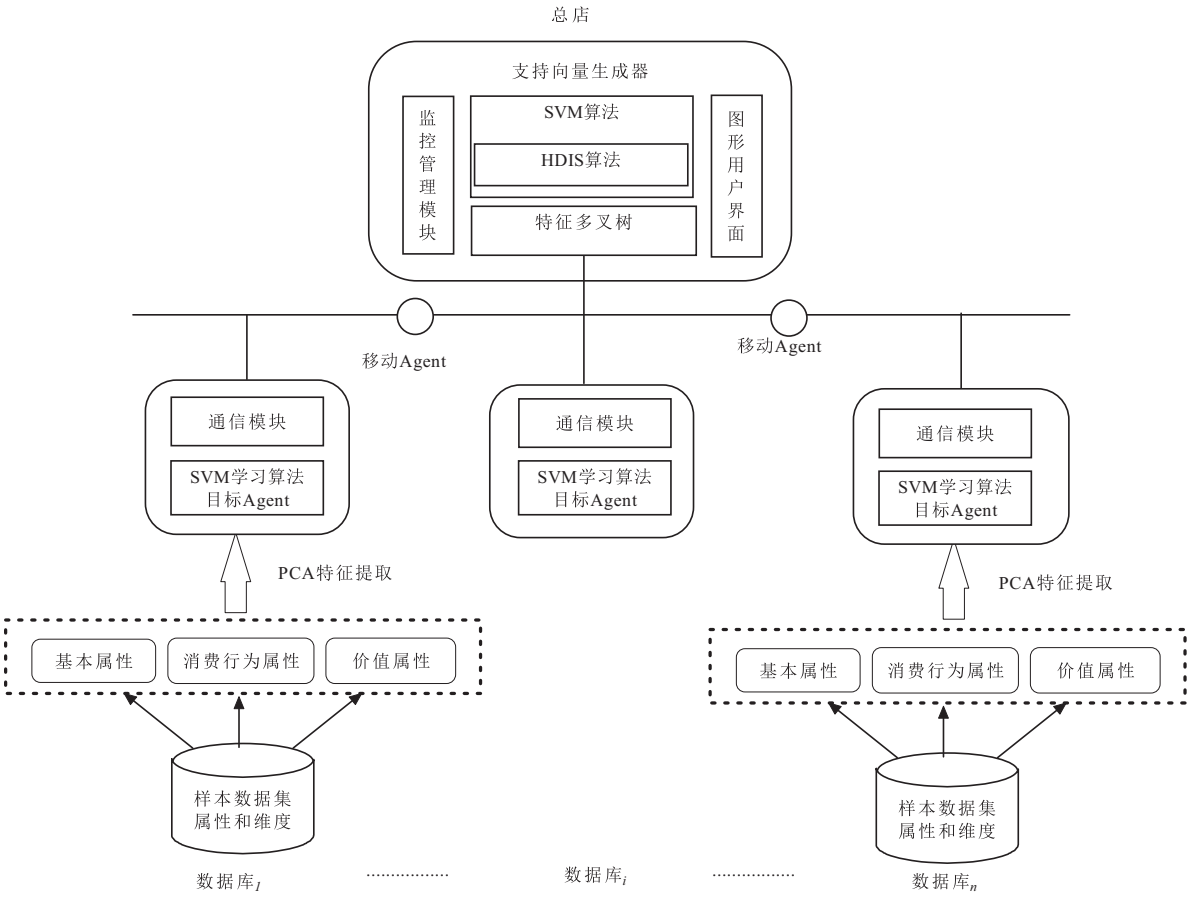

图 2　DSVM 模型体系架构

DSVM 模型挖掘中的两个核心机制: 考虑到企业目前对于数据挖掘决策属性的选择大都采用先验知识, 随着数据量的增大, 指标属性有效性大大降低. 因此, 首先在各个分站点采用多元统计分析方法 —— 主成分分析法 (Principal component analysis, PCA)[17] 从局部数据集中提取出主成分构建特征多叉树 (Multi-branches tree of Eigen, ET), 有效降低训练数据集维度和提高挖掘效率; 围绕 ET 提出一种基于壳向量的分布式支持向量机增量算法 (HDIS).

3.1　特征多叉树

本文在文献 [16] 属性多叉树的概念基础上, 提出了一种新的数据存储结构 —— 特征多叉树. 特征多叉树是一棵带有头表的多叉树, 树中除了叶子节点外每一层的节点对应于数据的一个特征属性, 这些特征是影响决策内容的相关原始属性的线性组合 (PCA 提取的主成分[17]), 如在客户流失预测中, 特征可以是客户持卡类型 (属性: 持卡年限、卡的积分、积累消费金额的线性组合). ET 的边对应于该特征不同的取值, 即该特征有多少种取值, 就有多少条边, 每个节点保存该边对应特征的取值、满足该取值的记录条数和保存下一个具有相同取值的结点的地址 (当下一个节点不存在, 则地址为 null). 叶子节点除有分支节点的信息外, 还多一个字段, 存放根节点到该叶子节点的路径所表示的样本的类别 (如客户流失预测中, 1 代表客户非流失, −1 代表客户流失). 图 3 是进行客户流失预测时的部分特征多叉树的例子, 第二层表示的是主成分持卡类型, 对应三条边, 每条边对应 “金卡”、“银卡”、“普卡” 三种取值, 每个节点保存某一取值及记录条数, 如在 “金卡”

的 2689 条记录中, 交易减少的频率 > 2000 的有 1205 条, 1000 < 交易减少的频率 < 2000 的有 830 条, 交易减少的频率 < 1000 的有 654 条.

特征多叉树的头表是面向节点查找的索引结构, 为 Agent 对 ET 的动态操纵提供支持. 头表中包含四个域: 特征名称 EigenName, 特征取值 EigenValue, 总数 Total 及具有相同特征值的链表头指针 Head. 头表中每一行记录一个特征的一种取值, 总数记录这个特征取该值的样本个数, 用一个链表将树中某个特征所有取值相同的结点链接起来, 并且把头指针放在头表中.

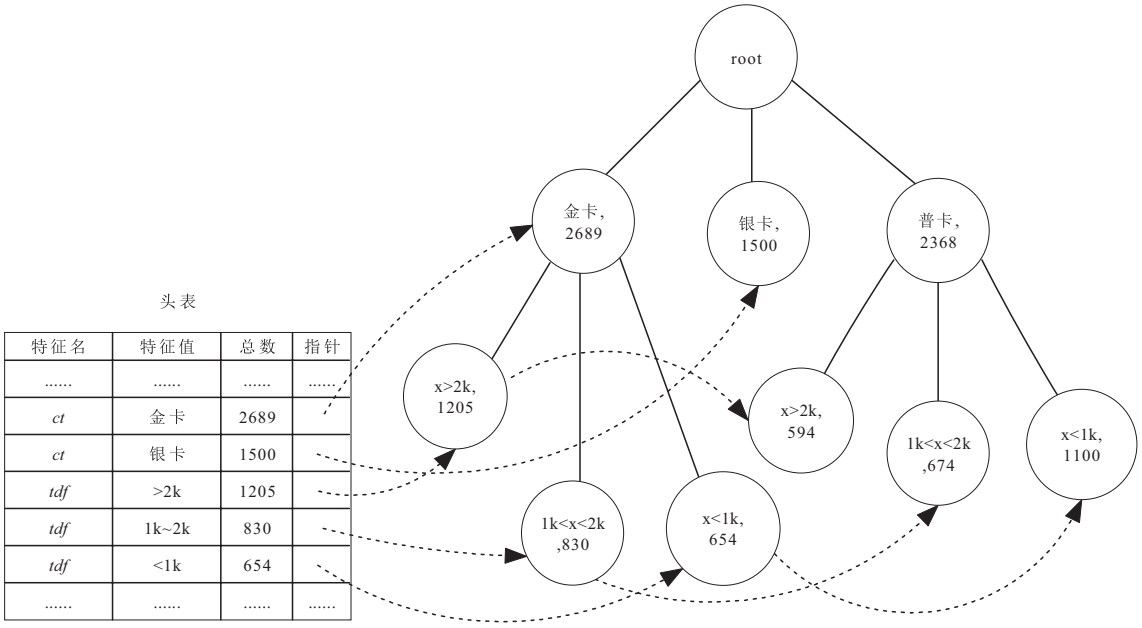

图 3 特征多叉树示例

ET 具有以下两方面优势:

1) 利用 PCA 将分站点的属性值进行主成分提取, 在降低属性维度的同时, 提高了许多含噪声数据集的精确度[18]. 因此, 本文将主成分 (即原来属性的线性组合) 用来构建特征多叉树, 在保留有效信息的同时, 由于树的层数的减少, 对树进行了裁剪, 降低了通信负荷, 从而可提高分站点通信的效率.

2) SVM 分类算法的中心思想就是寻求支持向量, 在两类数据之间形成一个最优分类面, 这个最优分类面要保证分类正确而且分类间隙最大[6]. 为了正确分类, 支持向量一定是位于每一类训练集几何意义上最边缘的点 (支持向量或者壳向量), 而不可能是训练集内部的点[19]. 另一方面, 支持向量数据量的大小又远远比原始数据集小, 所以通过对各个分站点的支持向量遍历进行特征多叉树的构建和修正, 在保证分类精确度的同时, 降低了修正树的时间复杂度.

用 ET 来反映分布数据源中支持向量的属性特征, 避免了分布数据汇总所带来的网络和存储负担. 该 ET 只是记录了分布数据源中支持向量的特征属性信息, 即各特征值和满足给定特征值条件的记录条数, 而无须保存整条记录的信息, 所以节省了网络和存储空间, 有利于运算效率的提高.

3.2 特征多叉树构建算法

当移动 Agent 移动到分布的数据集时, 每访问支持向量集中的一个元组, 可动态在特征多叉树中保存该元组的信息, 增减特征多叉树中结点及对应边的值. 当每个移动 Agent 移动到对应的数据集, 均可修改特征多叉数的边和节点值, 直到各移动 Agent 访问完所有的分布数据库中的支持向量集.

算法 1 特征多叉树构建

输入: 训练样本分布在 N 个子站点中, 分别标注为 $D = \{D_1, D_2, \cdots, D_N\}$; 将各主成分按贡献率从大到小依次排序, $E = \{E_1, E_2, \cdots, E_{n_1}\}$.

输出: 局部支持向量集 (SV_n) 的 ET, $n = \{1, 2, \cdots, N\}$.

方法:

For $k = 1$ to n_1

Begin AgentWrapper

IF ($k == 1$)

THEN

//从 D_n 的 SV_n 中统计 E_1 的所有取值 (n_2) 及取该值的记录条数 $Count$. 然后分别根据取值 EigenValue_k 及条数创建节点 ENode_{1i}, 并将这些节点连接到根节点 root.

For $i = 1$ to n_2

ENode_{1i} [EigenValue_1, $Count$] =CreateTree (E_1, SV_n)

Connection (root, ENode_{1i})

End

ELSE

//分别以上一层主成分的各节点 $\text{ENode}_{(k-1)i}$, 为父节点, 统计 SV_n 中满足父节点以上路径各节点的取值时, E_k 的所有取值 (n_3) 及取该值的记录条数, 分别根据该取值及条数创建节点 ENode_{kj}, 然后将该结点连接到对应的父节点.

For $i = 1$ to n_2

For $j = 1$ to n_3

ENode_{kj} [EigenValue_k, $Count$] =CreateTree (E_k, SV_n)

Connection ($\text{ENode}_{(k-1)i}$, ENode_{kj})

End

End

//把在不同父结点下 (主成分 E_{k-1}) 的子结点 (主成分 E_k) 中具有相同取值的节点指针链接.

Connection2 ()

//将 $Count$ 加入头表中的对应总数 $Total$ 项, 将头节点地址放入头表中具有其特征取值的指针 Head 项.

Insert ()

End

这是当移动 Agent 移动到某个数据集时构建 ET 的操作, 当移动到下一个数据集时, 其修正和完善该 ET 的算法为:

算法 2　特征多叉树修正

输入: 将各主成分按贡献率从大到小依次排序, $E = \{E_1, E_2, \cdots, E_{n_1}\}$; 训练样本分布在 N 个子站点中, 分别标注为 $D = \{D_1, D_2, \cdots, D_N\}$; 反映前 k 个站点 ($D_1 \cdots D_k$) 的支持向量集 (SV_k) 的 ET_k.

输出: 反映 $D_1 \cdots D_{k+1}$ 的 SV_{k+1} 的 ET_{k+1}.

方法:

For $k = 1$ to N

Begin AgentWrapper

//调用 HDIS 算法 (详见下一小节) 生成支持向量集.

SV_{k+1}=HDIS ()

// ET_k 中的所有节点 (n_4).

For $h = 1$ to n_4

//对于某一节点 ENode_{ij} (其中 i 代表某一层主成分, j 代表该层主成分的某个节点), 用 SV_{k+1} 修正和完善 ET_k, 统计支持向量集中满足父节点以上路径各节点取值的记录的条数$count$, 将该条数 $count$ 与该节点原有条数 $Count$ 相加, 并存入该节点.

ENode_{ij} [EigenValue_{n_1}, $Count + count$] =ModifyTree (E_h, SV_{k+1})

Connection2 ()

Insert ()

End

//各个站点移动来修正完善特征多叉树.

NextAgent (ET_{k+1})

End

3.3 基于壳向量的分布式支持向量机增量算法

对于同类数据的集合 S, 定义壳向量集和支持向量.

定义 1 $Hv(S)$ 表示对 S 求其壳向量集 (凸壳) 的算子, 求得的壳向量集用 HV 来表示, 即 $HV = Hv(S)$;

定义 2 令 $Sv(S)$ 表示对 S 求其支持向量集的算子, 求得的支持向量集用 SV 来表示, 即 $SV = Sv(S)$. 设 S_+ 为 S 的同类增量样本集, 且满足 $S \cap S_+ = \Phi$. 则可得到以下命题[20]:

推论 1 $Sv(S) \subset Hv(S)$.

推论 2 $Sv(SV \cup S_+) \neq Sv(S \cup S_+)$.

推论 3 $Sv(HV \cup S_+) = Sv(S \cup S_+)$.

推论 2 表明: 在一般情况下, 一个训练集与其同类增量集的并集的支持向量集, 不等于该训练集的支持向量集与增量集的并集的支持向量集. 因此建立在推论 2 基础上的现有 SVM 增量学习算法不能保证获得正确的分类决策函数. 推论 3 表明, 一个训练集与其同类增量集的并集的支持向量集, 等于该训练集的壳向量集与增量集的并集的支持向量集. 所以本节提出基于壳向量的分布式支持向量机增量算法 HDIS.

根据计算几何理论中的相关定理, 可计算数据集 S 的壳向量集合 (凸壳)$Hv(S)$[21]. 对于企业客户流失预测问题, 分布式 SVM 增量学习问题可以形式化描述为: 存在训练数据集 D_i, 可分为 D_i^+ 和 D_i^- 两类; 下一个站点训练数据集 D_{i+1}, 可分 D_{i+1}^+ 和 D_{i+1}^- 两类, 并且假定保证对于两个数据集有 $D_i \cap D_{i+1} = \Phi$. 则算法的目标是寻找基于样本集合 $D_i \cup D_{i+1}$ 上的 SVM 分类器 ψ 和对应的支持向量集 SV. 具体算法如下:

算法 3 HDIS

输入: 训练样本分布在 N 个子站点中, 分别标注为 $D = \{D_1, D_2, \cdots, D_N\}$; 反映 $D_1 \cdots D_i$ 支持向量集 (SV_i) 的 ET_i.

输出: $D_1 \cdots D_{i+1}$ 支持向量集 SV_{i+1}.

方法:

For $i = 1$ to N

Begin AgentWrapper

//对数据集 D_i 求出壳向量集.

HV_i=$Hv(D_i)$

//遍历 HV_i, 运用特征多叉树算法构建 $HVET_i$.

$HVET_i$=CreateTree (HV_i)

//移到下个站点 D_{i+1}, 将 HV_i 作为新的训练样本集加到 D_{i+1} 中.

NextAgent ($HVET_i$)

//调用 SVM 算法取得支持向量集 SV_{i+1}, 并由此构造新的最优分类器 ψ

$SV_{i+1} = SVM(HV_i \cup D_{i+1})$

End

运用 ET 作为局部支持向量信息的载体, 通过 HDIS 算法修正和完善 ET, 实现各个站点壳向量与支持向量的传递和共享.

4 测试与分析

原型系统 R-DSVM 是在 DSVM 模型和 Toshiba 的 Bee-gent 的基础上, 利用 Bee-gent 相关类和接口来实现分布环境下基于支持向量机的全局挖掘[16]. 本文在实验室局域网环境下测试分布式挖掘算法. 数据来自某连锁商业企业真实的客户数据, 4 台 PC 构成分布式站点.

本节讨论客户流失是个二分类问题, 给定 m 个训练样本 $\{z_j = (x_j, y_j)\}_j^m$, 其中 x_j 为样本输入, $y_j \in \{-1, +1\}$ 为样本输出, -1 表示流失, $+1$ 表示正常. 因为交易数据是非线性数据, 核函数采用高斯核 $k(x, y) = \exp(-||x - y||^2)/\beta$, 其中 $\beta = \frac{1}{m^2}\sum_{i,j=1}^m ||x_i - y_i||^2$. 从数据库中提取与客户流失预测相关的属性, 然后在各自站点进行主成分分析, 提取主成分值, 将这些主成分的取值处理后, 数据格式如下:

$$< \text{label} >< \text{index1} >:< \text{value1} >< \text{index2} >:< \text{value2} > \cdots$$

其中 < label > 是训练数据集的目标值, 对于分类, 它是标识某类的整数 $(-1, 1)$; < index > 是以 1 开始的整数, 可以是不连续的 (代表主成分); < value > 为实数, 也就是主成分值. 图 4 为测试时某个站点处理后的数据集 (每个样本通过 PCA 从 45 维属性降到 7 维主成分). 最终, 利用 Agent 和 ET 采集到的全局支持向量集作为训练数据, 调用 SVM 算法进行全局挖掘.

4.1 分布式支持向量机增量算法之间的对比

使用四个分站点, 每个分站点随机选取 230、275、286、300 个样本. 采用本文提出的挖掘模型. 算法 4 采用局部所有样本集构建 ET; 算法 5 采用局部 SVM 训练得到的 SV 构建 ET; 算法 6 用本文提出的 HDIS 算法修正和完善 ET. 进行分布环境下客户流失预测的测试, 实验结果如表 1 所示.

随着移动 Agent 装载 ET 在各个站点移动, 分布式 SVM 增量学习不断进行. 从表 1 可以看出, 算法 6 与算法 4 相比大大地节约了计算时间, 加快了模型训练的速度, 而分类准确率基本一致; 算法 6 与算法 5 相比时间开销略高, 但是准确率较高, 表明 HDIS 是有效的. 同时, 随着增量学习的不断进行, HDIS 会使部分壳向量转化为非壳向量, 从而实现对训练历史数据进行有选择的遗忘.

4.2 与集中式 SVM 算法之间的对比

分别随机抽取 100、1000 和 10000 条记录作为测试样本均匀分布在 4 个不同站点. 采用数据汇总法与本文提出的方法在支持向量个数、运行时间、分类准确率等方面进行比较. 数据集中法通过 RMI 远程访问, 把分布数据汇总, 然后从汇总数据中直接进行支持向量的提取, 进行全局挖掘; 原型系统 R-DSVM 通过移动 Agent 访问分站点支持向量集来构建 ET, 全局生成综合 ET 后进行挖掘. 结果如表 2.

```
-1  1:0.032 2:1 3:0.224 4:0.056 5:-0.516 6:-1 7:1
-1  1:-0.391 2:-1 3:0.348 4-:0.094 5:-0.567 6:-1 7:1
-1  1:0.583 2:-1 3:0.333 4:0.094 5:-0.311 6:-1 7:-1
+1  1:0.125 2:1 3:1 4:0.415 5:-0.438 6:1 7:1
-1  1:-079 2:-1 3:-0.333 4:-0.547 5:-0.616 6:-1 7:-1
-1  1:0.166 2:1 3:1 4:-0.283 5:-0.630 6:-1 7:-1
+1  1:0.458 2:1 3:1 4:-0.037 5:-0.607 6:-1 7:1
-1  1:0.254 2:1 3:1 4:-0.169 5:-0.337 6:-1 7:1
+1  1:-0.125 2:1 3:0.333 4:-0.132 5:-0.511 6:-1 7:-1
-1  1:-0.083 2:1 3:-1 4:-0415 5:-0.602 6:-1 7:1
+1  1:0.041 2:1 3:-0.357 4:0.849 5:-0.283 6:-1 7:1
+1  1:0.052 2:1 3:1 4:-0.452 5:-0.283 6:-1 7:1
+1  1:-0.041 2:1 3:1 4:-0.667 5:-0.525 6:-1 7:1
+1  1:-0.542 2:1 3:1 4:-0.698 5:-0.812 6:-1 7:1
+1  1:0.203 2:1 3:0.333 4:-0.283 5:-0.552 6:-1 7:1
-1  1:0.523 2:-1 3:0.333 4:-0.660 5:-0.351 6:-1 7:1
-1  1:-0.5 2:1 3:0.333 4:-0.661 5:-0.433 6:-1 7:-1
-1  1:-0.125 2:-1 3:0.333 4 4:-0.509 5:-0.575 6:-1 7:-1
-1  1:0.041 2:-1 3:0.333 4:-0.735 5:-0.356 6:-1 7:1
-1  1:0.458 2:-1 3:-0.325 4:0.056 5:-0.191 6:-1 7:-1
-1  1:-0.083 2:-1 3:0.333 4:-0.320 5:-0.406 6:-1 7:1
-1  1:-0.291 2:-1 3:-0.333 4:-0.792 5:-0.643 6:-1 7:-1
+1  1:0.083 2:1 3:1 4:-0.132 5:-0.584 6:-1 7:-1
-1  1:-0.333 2:1 3:-0.333 4:-0.358 5:-0.168 6:-1 7:1
```

图 4　某分店上部分数据集

表 1　分布式 SVM 增量算法之间的比较

分站点	算法	壳向量	支持向量	时间 (秒)	准确率 (%)
分店 1	4	—	54	47±4	84.80
	5	—	54	47±2	84.80
	6	103	54	47±3	84.80
分店 2	4	—	78	68±4	86.54
	5	—	70	57±3	85.48
	6	159	77	60±3	86.49
分店 3	4	—	98	125±5	88.86
	5	—	88	68±3	86.24
	6	195	96	75±4	88.79
分店 4	4	—	123	236±7	89.58
	5	—	105	80±4	86.57
	6	224	119	95±6	89.39

如表 2 所示, 在数据分布环境下, 集中式 SVM 的时间效率明显低于分布式 SVM 的时间效率. 原因是: 利用数据汇总法时, 需要先将各分布结点上的数据集汇总, 这会占用大量的网络传输时间, 而利用 DSVM 则不同, 它只需要汇总 ET, 该 ET 能概括各分布结点上数据集中支持向量的属性特征值, 利用网络传输 ET 的数据量远远小于传输各分布数据记录的数据量. 另外, 集中 SVM 的准确率要略高于分布式 SVM, 但是相差

不多, 因为在分布的数据集中统计 ET 时, 会存在一定偏差, 导致生成的全局支持向量的个数减少, 构建的分类面的分类准确性有所降低. 但从综合性能来说, 在真实的数据环境下分布式 SVM 要比集中式 SVM 更优越.

表 2 分布式与集中式 SVM 之间的比较

测试记录数 (条)	支持向量机	支持向量	时间 (秒)	准确率 (%)
100	集中式 SVM	68	25±3	87.21
	分布式 SVM	61	16±2	84.48
1000	集中式 SVM	364	282±5	88.53
	分布式 SVM	319	120±4	86.14
10000	集中式 SVM	2624	986±9	90.51
	分布式 SVM	2521	309±5	88.93

4.3 与其他分布式数据挖掘方法之间的对比

为了验证模型的有效性, 与其他分布式方法进行对比, 选用神经网络[9]、决策树 (C4.5, 决策树采用局部所有样本集构建 ET, 通过 Agent 移动到全局站点进行客户流失预测)、贝叶斯 ($p(X|w_i)P(w_i)-p(X|w_j)P(w_j)$ =0 为分类器判别面, 其中 w_i 为分类类型, $p(X|w)$ 为条件概率, $P(w)$=0 为先验概率)[16]. 分别随机抽取 10000 条记录用做测试样本分布在 4 个不同站点.

对不同模型的准确率、命中率、覆盖率和提升系数等参数进行比较, 参数设置见表 3.

模型准确率 =(A+D)/(A+B+C+D); 命中率 =A/(A+C); 覆盖率 =A/(A+B); 提升系数 = 命中率/数据中的客户流失率, 得到的结果如表 4.

表 3 预测计算指标

样本中客户状态	预测流失	预测正常
实际流失	A	B
实际正常	C	D

表 4 各个算法之间的比较

模型算法	准确率	命中率	覆盖率	提升系数
DSVM	0.95	0.90	0.54	8.74
神经网络	0.94	0.93	0.23	8.71
贝叶斯	0.91	0.88	0.51	8.56
决策树	0.85	0.80	0.43	6.34

从上述指标来看, DSVM 模型中使用 HDIS 的预测结果与神经网络、贝叶斯、决策树相比除比神经网络的命中率略低外, 其他指标均具有一定的优势. 其重要原因是神经网络覆盖率较小, 可知该方法一定程度上出现了过拟合现象.

5 结束语

文章提出了基于支持向量机的分布数据挖掘模型, 是以企业分布的商业数据为数据源, 以主成分分析方法和支持向量机为分布式高性能算法设计的基础, 应用移动 Agent 访问分布数据集来构建特征多叉树为中间桥梁, 从分布企业数据库中得到全局知识, 最终实现企业高效、精确的商业决策. 实验表明: 该模型有效地解决原有分布环境下其他挖掘算法存储开销大、执行效率差、安全性和隐私性低等问题. 今后的研究将在模型的动态性、并行运算等方面展开.

参考文献

[1] 王益萍, 琚春华. 基于分布式数据挖掘的连锁商业企业经营决策分析 [J]. 商业研究, 2006, 20(352): 6–10.
Wang Y P, Ju C H. Chain business enterprise decision making based on distributed data mining[J]. Commercial Research, 2006, 20(352): 6–10.

[2] Çelebi D, Bayraktar D. An integrated neural network and data envelopment analysis for supplier evaluation under incomplete information[J]. Expert Systems with Applications, 2008, 35(4): 1698–1710.

[3] 郭明, 郑惠莉, 卢伟. 基于贝叶斯网络的客户流失分析 [J]. 南京邮电大学学报: 自然科学版, 2005, 25(5): 79–83.
Guo M, Zheng H L, Lu W. An analysis of customer loss with Bayesian networks method[J]. Journal of Nanjing University of Posts and Telecommunications: Natural Science Edition, 2005, 25(5): 79–83.

[4] Hung S Y, Yen D C, Wang H Y. Applying data mining to telecom churn management[J]. Expert Systems with Applications, 2006, 31(3): 515–524.

[5] 夏国恩, 金炜东. 基于支持向量机的客户流失预测模型 [J]. 系统工程理论与实践, 2008, 28(1): 71–77.
Xia G E, Jin W D. Model of customer churn prediction on support vector machine[J]. Systems Engineering — Theory & Practice, 2008, 28(1): 71–77.
[6] Vapnik V N. 统计学习理论 [M]. 许建华, 张学工, 译. 北京: 电子工业出版社, 2004.
Vapnik V N. The Nature of Statistical Learning Theory[M]. Beijing: Publishing House of Electronics Industry, 2004.
[7] Wright R N, Yang Z. Privacy-preserving Bayesian network structure computation on distributed heterogeneous data[C]//Proc of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Seattle, USA, 2004: 713–718.
[8] Xun Y, Yan C Z. Privacy-preserving distributed association rule mining via semi-trusted mixer[J]. Data & Knowledge Engineering, 2007, 63(2): 550–567.
[9] 田大新, 刘衍珩, 李宾, 等. 基于 Hebb 规则的分布神经网络学习算法 [J]. 计算机学报, 2007, 30(8): 1379–1388.
Tian D X, Liu Y H, Li B, et al. Distributed neural network learning algorithm based on Hebb rule[J]. Chinese Journal of Computers, 2007, 30(8): 1379–1388.
[10] Chapelle O, Vapnik V. Choosing multiple parameters for support vector machines[J]. Machine Learning, 2002, 46: 131–159.
[11] Syed N, Liu H, Sung K. Incremental learning with support vector machines[C]//Proc of Workshop on Support Vector Machines at the International Joint Conference on Artificial Intelligence, 1999: 272–276.
[12] Ruping S. Incremental learning with support vector machines[C]// Proc IEEE Int Conf on Data Mining, San Jose, CA, USA, November 2001: 641–642.
[13] Hua D, Xiao J S. An incremental learning algorithm for Lagrangian support vector machines[J]. Pattern Recognition Letters, 2009, 30(15): 1384–1391.
[14] 於俊, 周维. 一种基于壳向量的 SVM 快速增量学习算法 [J]. 电子测量与仪器学报, 2006, 20(6): 94–97.
Yu J, Zhou W. A new and fast incremental SVM learning algorithm based on hullvectors[J]. Journal of Electronic Measurement and Instrument, 2006, 20(6): 94–97.
[15] 岳博, 焦李成. 基于 Agent 的分布式数据挖掘系统 [D]. 济南: 山东大学, 2004.
Yue B, Jiao L C. Distributed data mining system based on Agent[D]. Jinan: Shandong University, 2004.
[16] 琚春华, 张捷. 基于贝叶斯网络的分布数据挖掘模型 DDMB 研究 [J]. 情报学报, 2008, 27(5): 52–60.
Ju C H, Zhang J. Distributed data mining model based on Bayesian network[J]. Journal of the China Society for Scientific and Technical Information, 2008, 27(5): 52–60.
[17] Howley T, Madden M G, O'Connell M L, et al. The effect of principal component analysis on machine learning accuracy with high-dimensional spectral data[J]. Knowledge Based Systems, 2006, 19(5): 363–370.
[18] Babaoğlu I, Fındık O, Bayrak M. Effects of principle component analysis on assessment of coronary artery diseases using support vector machine[J]. Expert Systems with Applications, 2009, 37(3): 2182–2185.
[19] Ker-I Ko, Yu F X. On the complexity of convex hulls of subsets of the two-dimensional plane[J]. Electronic Notes in Theoretical Computer Science, 2008, 202(21): 121–135.
[20] 周伟达, 张莉, 焦李成. 支撑矢量机推广能力分析 [J]. 电子学报, 2001, 29(5): 590–594.
Zhou W D, Zhang L, Jiao L C. An analysis of SVMs generalization performance[J]. Acta Electronic Sinica, 2001, 29(5): 590–594.
[21] Osuna E, De Castro O. Convex hull in feature space for support vector machines[J]. Lecture Notes in Computer Science, 2002, 2527/2002: 411–419.