电子商务环境中分布式数据挖掘系统①

Distributed Data Mining in Electronic Commerce Environment

马荣飞 (浙江经济职业技术学院 信息与控制学院 浙江 杭州 310018)

摘 要： 首先将 web 服务和 Agent 最新技术引入了电子商务环境中分布式数据挖掘，提出了电子商务环境下基于 web 服务和移动 Agent 技术的数据挖掘架构 BWADM，最后建立了 BWADM 原型，结合 Web 服务技术，给出了基于 Web 服务的数据挖掘系统逻辑结构，设计并实现了该系统，验证了 BWADM 的合理性和上述算法在效率、精确度等方面的优越性。

关键词： Web 服务 Agent 技术 BWADM

1 引言

电子商务取代了传统的面对面的交互方式，企业和客户通过网络进行交互。由于 Internet/Intranet 的发展，电子商务企业要面对的客户群是巨大而复杂的，市场也变得更大、更复杂，其数据也就变得更加丰富。挖掘电子商务环境中的数据，找出其中有价值的“知识”，企业用户可以根据这些“知识”，追踪市场变化，掌握客户动态，做出正确的针对性的决策。

对电子商务系统进行数据挖掘时，所需要的数据主要是 Web 内容、Web 结构、Web 使用记录、客户的背景信息、交易数据、查询信息等。这些数据具有分布性[1]、异构性、稀疏性、高维性及海量性等特征。但是，目前大部分数据挖掘工具网络功能很弱，主要在单机上运行，不能操作异构数据。数据挖掘算法和模式主要采用集中式，对分布存储的数据需要集中起来，才能进行挖掘，这需要通信网络速度高，并且会导致响应的时间变长、数据的私有性和安全性遭到破坏。当前大部分电子商务挖掘系统的挖掘引擎也是基于封闭体系设计的，难以动态有效地管理和维护多个挖掘算法。由于挖掘结果格式的封闭，许多数据挖掘引擎提供的挖掘结果无法为其他应用系统使用。

本论文提出了电子商务环境下基于 Web 服务和移动 Agent 的数据挖掘架构(BWADM)。BWADM 中，web 服务的组合由 Agent 负责执行，组合服务入口将任务交给 Agent，并等待 Agent 返回结果。BWADM 有效地解决了电子商务环境中数据的分布性、异构性、可扩展性等问题。利用 BWADM，企业可以很容易地根据自己的业务需求定制自己的挖掘算法库，并且可以实现挖掘算法库的动态管理和升级。

2 BWADM原型系统架构设计

BWADM 有 3 类实体(见图 1 所示):组合服务入口、组合服务 Agent(SAgent)和 Web 服务。服务组合客户端的用户通过组合服务入口输入服务组合规范和初始参数。系统将这个组合规范转化为“计划”，并且产生一个 Agent 来实现这个任务，并返回结果。

该架构主要划分为 4 层。底层为数据存储层，包含有不同的信息库。每一个信息库一定对应一个可以对它进行访问的服务组件。在服务组合层，服务组件就是每一个数据源的封装器。这个封装器的作用就是将一个用标准 XML 查询语言翻译成能在经过封装的数据源上执行的操作，抽取和打包查询结果到一个 XML 文档，最后将该文档返回给上层调用。它提供了

① 收稿时间:2008－10-14

一个统一的访问接口以便信息源可以作为相关的数据库来被查询。语义推理层主要将应用表示层传来的服务调用在服务推理逻辑中进行处理。服务推理逻辑会解析请求的合理性，并对照本体知识库中的相关概念定义和规则对服务请求进行标准化和约束设定。经过推理逻辑的处理后，用户的请求将被重构为对信息源的请求。本体知识库为相关领域的信息提供了统一的概念描述，并对概念的表达形式进行统一规范。对照本体库，还可对用户查询范围进行扩大。

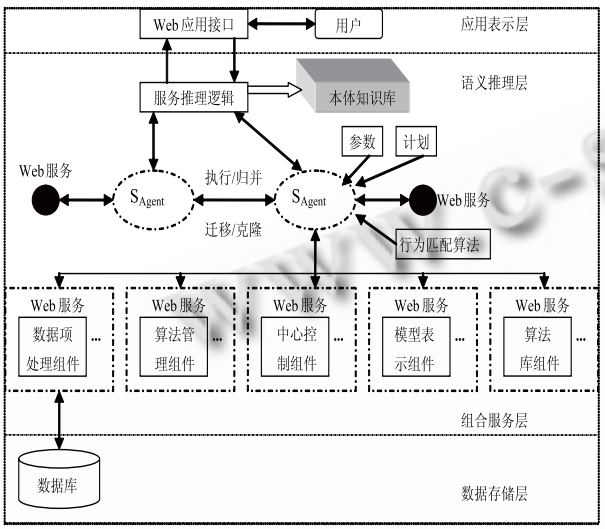

图 1　BWADM 架构图

3　BWADM原型系统的数据流程图

本数据挖掘原型系统的所有挖掘算法都被封装成 Web 服务，根据基于 web 服务的数据挖掘系统的总体架构设计以及对数据挖掘系统关键技术的研究，本原型系统分别实现了数据预处理模块、控制中心模块、算法管理模块、算法库模块和模型表示模块五个组成部分。图 2 是本原型系统的数据流程图。

4　BWADM原型系统模块设计与实现

4.1 开发工具介绍

开发工具采用.NET，它是 Microsoft 的 Web 服务平台，不论操作系统或编程语言有何差别，Web 服务能使应用程序在 Internet 上传输和共享数据，也可以调用其它应用程序的功能，而不考虑其他应用程序是如何生成的。并且在保证应用程序相互独立的同时，Web 服务还使它们能够建立松链接而形成一个合作组来完成某个特定的任务。

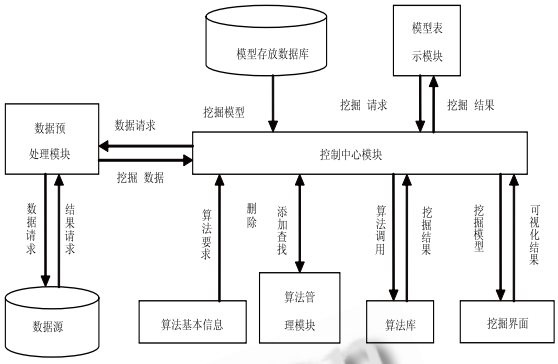

图 2　BWADM 原型数据流程图

4.2 参数文件格式

在挖掘开始时，用户需要在客户端提交数据源信息、数据预处理信息和挖掘任务信息。这些参数信息在挖掘过程中起着非常重要的作用。而且参数文件最终要和挖掘结果模型一起存入挖掘结果数据库中。因此，有必要将用户提交的挖掘参数用 XML 格式的文件存放起来。参数文件共有三种，分别是数据源参数文件，数据预处理参数文件和挖掘任务参数文件。该 XML 文件的文件名用参数文件类型和时间戳共同表示。

4.3 数据预处理

在数据预处理[2]中，数据源中的数据是一次性被取出的。为了在应用程序内部存放取出的数据并对原始数据预处理，本原型系统定义了三个存放和处理原始数据的类:Attribute，Instance，Instances。Attribute 类表示数据表的某个属性相应的元数据和该属性上的操作。Instance 类是用来存放数据表中的某条记录，是一个向量结构。Instances 类则用来存放整个数据源，它由 Attribute 类对象的集合和 Instance 类对象的集合构成，所有的数据预处理都在该类上进行的。

4.4 控制中心模块

控制中心模块是原型系统的调度中心。它首先解析用户提交的数据挖掘请求，收集用户提交的数据源参数、数据预处理参数和挖掘任务参数，然后分别调用数据预处理模块、算法发现子模块以及模型表示模块，将挖掘结果以图形化的形式返回给用户。在本原型系统中使用了一个控制中心类 ControlCenter 来实现控制中心的作用。

ControlCenter 从负责与客户交互以及收集请求参数的 Actionservlet 小服务程序中获得所有的用户

请求参数和请求类型，然后按顺序调度数据预处理模块以及算法发现子模块。如果查找到相应的挖掘算法，则首先与 Web 服务形式的挖掘算法绑定，然后调用该挖掘算法。当成功返回挖掘模型后将挖掘模型表示为 PMML 格式的文档，然后把参数文件和 PMML 格式的挖掘模型存入结果数据库中，并调用模型表示模块解析 PMML 格式的挖掘模型，根据挖掘模型的类型处理成不同的表示方法并返回给用户。

4.5 算法管理模块

算法管理模块包括 UDDI 注册中心、算法发布子模块和算法发现子模块。UDDI 注册中心保存算法提供者发布的所有算法的基本信息。算法发布子模块提供发布界面供算法提供者发布封装成 Web 服务的挖掘算法。算法发现子模块则提供算法发现的 API 函数，供控制中心模块发现需要的算法。

(1) 算法发布子模块

算法发布子模块负责把封装成 Web 服务的挖掘算法发布到 UDDI 注册中心。发布流程是：用户首先获得在 UDDI 注册中心的令牌(如果没有，必须首先在 UDDI 注册中心注册，成为合法用户)，然后分别发布 BusinessEntity、BusinessService、Binding Template 和 tModel。BusinessEntity、BusinessService、BindingTemplate 和 tModel 均为 UDDI 的数据结构。BusinessEntity 表示公司的主要信息，BusinessService 表示公司的服务，BindingTemplate 由指向技术说明的指针和服务的接入 URL 构成，tModel 则提供特定规范或服务的行为描述。

(2) 算法发现子模块

算法发现子模块用于从 UDDI 注册中心获取关于某算法对应 Web 服务的调用信息，以方便控制中心模块调用挖掘算法。下面是到 UDDI 注册中心查找 KNDC 算法的关键代码。

```
Vector findAlgorithmEntity(){
UDDIProxy proxy=new UDDIProxy(); //构造一个 UDDIProxy 对象
try{
proxy.setInquiryURL("http://192.168.10.8/data_mining/inquiry"); //指定查Web服务的UDDI注册中心的入口
}catch(Exception e){
Logger.saveLog(e); }//如果出错，将出错信息存入日志文件
try{
BusinessList bl=proxy.find_business ("KNDC",null,0);//查找算法 KNDC
Vector businessInfo Vector=bl.getBusinessInfos().getBusinessInfoVector();
ReturnbusinessInfoVector; //返回翻 De 算法信息
}catch(Exception e){
Logger.saveLog(e); //如果出错，将出错信息存入日志文件
Return null;}}
```

4.6 Web 服务实现

数据挖掘在整个过程中耗时最长，直接与数据库打交道的部分是放在不同挖掘客户端并行执行的。这可以保证整个系统的挖掘效率，其次由于企业的业务数据并没有在网上传输，无疑也可以保证系统的安全性。而挖掘 Web 服务端则协调不同客户端的挖掘进度，这里有个问题要注意：Web 服务[3]理论上是不能保存状态的，而数据挖掘算法一般较复杂，每次挖掘都会涉及到多轮的交互。每个客户端挖掘的前一轮与后一轮的交互，不同客户端之间不同挖掘轮次的交互都要通过 Web 服务来实现，这就要求 Web 服务端必须有保存状态。要解决这个矛盾需要编程时在 Web 服务端启用 xMLWebServices 方法的会话状态，如下所示：

```
[WebMethod(Enablesession=true)]
Public ArrayList DataMining(ArrayList candItemValue，int step)
{}
```

同时要在 Web 服务启动时定义、初始化 Application 状态值。有了以上两个步骤，每个客户端挖掘的前一轮与后一轮的交互、不同客户端之间不同挖掘轮次的交互便不会存在问题了。

挖掘流程中遇到的第一个 Web 服务是挖掘服务进入模块，这是整个 Web 服务中最简单的一个接口，这个接口的前部分代码是在同步不同客户端的挖掘请求，直到所有的客户都提交了挖掘请求，代码进入下半部分。代码的后半部分在服务器端启动了一个同步线程，之所以开启一个新的线程，是因为主线程有一部分时间是在等待客户端的同步，将发现模块单独提出放在另一个线程空间无疑会提高整个系统的运行效率。

4.7 挖掘客户端实现

挖掘客户端在整个分布式挖掘系统中占有重要的位置，整个挖掘开始后，先在本地进行本地数据挖掘，本地数据挖掘所占用的处理器时间在整个分布式挖掘系统中占的比例很高，这样可以提高整个系统的挖掘效率。

客户端代码中为了引用挖掘 Web 服务，需要首先在编译环境中配置 Web References，然后就可以在客户端代码中使用 Web 服务端命名空间了。由于数据挖掘涉及大量的数据库访问，系统运行所用的时间较长。如果按照常规的设计，当用户在客户端开始一轮挖掘后，整个程序界面会失去响应。所有这一挖掘原型的客户端使用了多线程技术，程序主界面所在的线程空间与数据挖掘所占用的线程空间是隔开的，从而实现了 Web 服务的异步调用。

5 BWADM原型系统的运行实例

5.1 数据源信息和结果数据库信息设定

本实例数据来源于用户使用电子商务平台后产生的服务日志。数据挖掘的第一步是设置数据源信息以及挖掘模型存放的结果数据库信息，具体设置如图 3 所示。需设置的信息包括数据源的 IP、数据源所在数据库的名称、数据库类型、数据库端口号、数据表名称、用户名和密码以及结果数据库 IP、结果数据库名称、结果数据库类型、数据库端口号和访问结果数据库的用户名和密码。这些信息被提交后会保存在 Web 服务器中的参数文件中，以供挖掘过程中随时访问这些信息。如果结果数据库信息没有设定，则挖掘结果并不保存到数据库中。

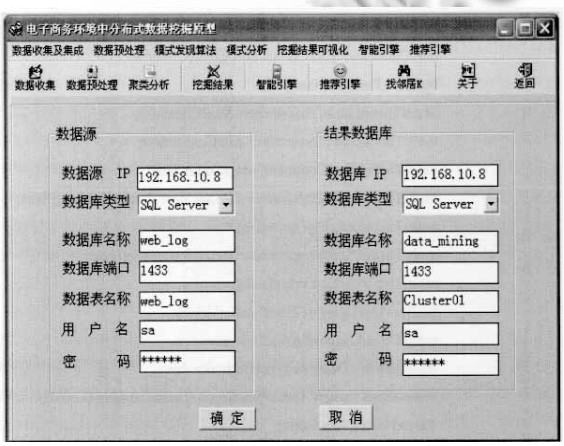

图 3　数据信息设置界面

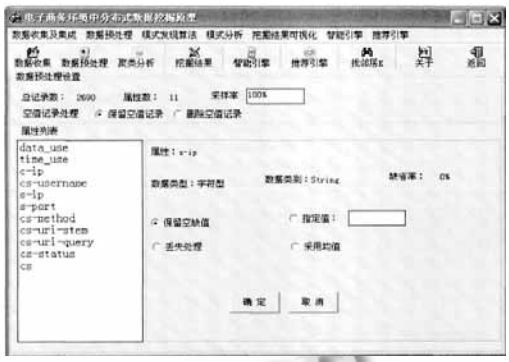

图 4　数据预处理设置界面

5.2 数据预处理设定

数据源信息提交以后，系统自动到数据库中读取相应数据表的信息，并成如图 4 所示的数据预处理信息设定。根据从数据库读取的信息，生成的面包括数据表中总的记录数、属性数、以及属性列表。用户可以在界面上设置整个数据库的采样比例。点击属性列表中某个属性，界面就会显示该属性的基本信息，包括数据类型、数据类别、缺值率和根据数据类别生成的据预处理页面。其中数据类别共有三种:Numeric(数字型)，Nominal(名词型)和 string(字符串型)。数据预处理的设定项包括缺值处理等。缺值处理方法包括保留空缺值、指定值、采用均值和丢失处理。

5.3 挖掘任务设定

本过程主要用于挖掘任务的设定。挖掘任务包括关联规则、聚类分析、分类分析、回归分析、预测分析、异常点检测等。本实例中应用聚类分析挖掘方法对 Web 日志数据进行挖掘。图 5 给出了聚类分析挖掘的参数设定，包括算法选择以及和算法相关的各个参数的设定。

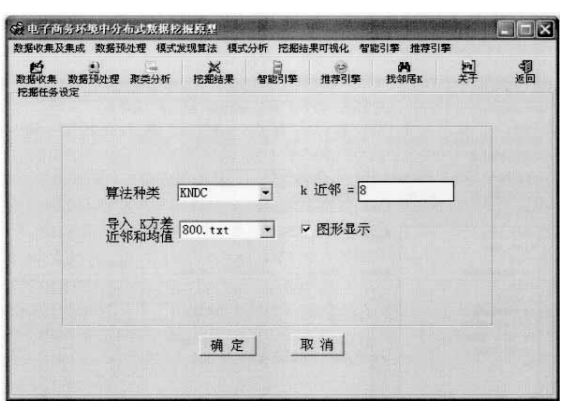

图 5　聚类分析挖掘的参数设定界面

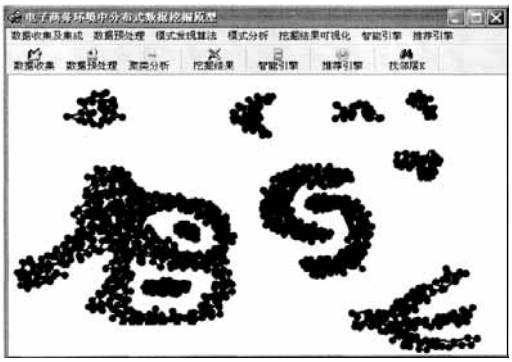

图 6 挖掘结果的连线图显示

5.4 挖掘结果

图 6 是采用聚类分析中的 KNDC 挖掘算法对 Web 日志数据的挖掘部分结果截图。数据转换后的 Web 日志记录经过数据清洗、用户识别、会话识别、路径识别后，从 2690 个用户得到 146 个用户会话，对应的页面总数从 1634 降低到 129 个，从图中可以看出聚类结果为 5 类。根据网站的拓扑结构，分析对应这些页面，发现页面 58，59，60，…，70 是基于电子商务平台的物流企业运作的相关网页，所以如实际的网站拓扑结构那样聚在一类中。当然通过页面聚类可以发现一些网站拓扑结构本身不包含的页面集结果，在实例运行中发现 32，33，34，…，48 这些页面属于电子商务 B2C 模式的网页。其中编号 39 页面不是模拟 B2C 网页，在网站拓扑结构中，39 页面和其余页面没有直接相连，而是在 B2C 实践下的一个实际的电子商店页面。说明访问电子商务 B2C 模式的用户大多会访问此页面，因为通过此页面可以实际自主开电子商店或查询网络上的商品信息。如果在电子商务平台网站主页中添加此 39 页面的超链接，就更激发用户的兴趣，同时对链接到此网站的电子商店的发展起到更好的推动的作用。正如例子所述，对这些页面集的分析可以改进网站拓扑结构，使我们的网站更趋合理。

在电子商务系统中，应用这样的方法可以发现表面上似乎没有联系的商品网页之间具有的密切关系。通过页面聚类，如果发现物品 A 的页面和物品 B 的页面被聚类在一起，就可能揭示物品 A 和物品 B 有一定内在联系或大量用户有同时访问的习惯，就可以在这两个页面上添加相互之间的链接或改变这两个页面的网站结构位置，减少访问两个页面之间的中间页面，提高用户访问商品的效率，从而也提高了商品的销售。

6 结论

本数据挖掘系统采用面向服务的架构，使用了 Web 服务技术，企业可以很容易地根据自己的业务需求定制自己的挖掘算法库，并且可以实现挖掘算法库的动态管理和升级。和以前的数据挖掘系统中算法和其他功能模块紧密藕合的结构相比，本数据挖掘系统具有较高的优越性，具体表现在以下几点。

(1) 算法库的动态管理。用 Web 服务封装的数据挖掘算法都在 UDDI 注册中心注册，使得可以非常容易地实现对挖掘算法的动态管理，包括挖掘算法的查找、添加、删除和更新等。

(2) 平台无关性和语言无关性。系统的设计完全基于 SOAP/HTTP/XML/UDDI 等与平台无关的规范或协议，对挖掘服务的请求可以通过 HTTP 或 SMTP 传输，穿越防火墙。用 Web 服务封装的挖掘算法可以使用任何语言实现，完全脱离了程序设计语言的束缚，给算法设计者最大的自由度。

(3) 算法库的分布式部署。算法库中的算法都是采用 Web 服务的形式封装的，这些挖掘服务不必部署在某一个 Web 服务容器中。根据具体情况可以部署在一台服务器上，也可以分别部署在不同的机器上。可以提高数据挖掘的效率，同时也提高了算法库的安全性。

(4) 按需定制挖掘服务。企业在部署本数据挖掘系统时可以自由地根据企业自身的业务需求量身定做算法库，另外也可以非常容易地添加自己的新算法。

(5) 输出结果采用挖掘模型标准 PMML 表示。可以和其他支持 PMML 的数据挖掘工具实现挖掘模型共享，还可以采用某些 PMML 可视化工具对 PMML 模型可视化。

参考文献

1 Hwang JH, Gu MS, Ryu KH. Context-Based Recommendation Service in Ubiquitous Commerce.Gervasi O. et al.(Eds.):ICCSA 2005, LNCS 3481, 2005:966－976.

2 唐新余,陈海燕,李晓,等.数据清理中几种解决数据冲突的方法.计算机应用研究, 2004,12:209－211.

3 胡春明,怀进鹏,孙海龙.基于 Web 服务的网格体系结构及其支撑环境研究.软件学报, 2004,14(7):1064－1072.