Apriori-Based 算法改进在电力业务系统功能优化中的应用①

王成现, 胡扬波, 谭 晶

(江苏电力信息技术有限公司, 南京 210024)

摘 要: 提出一种针对电力业务系统功能优化算法. 首先, 从 Web 服务器和客户端采集用户日志数据; 然后, 对用户日志数据进行预处理, 并将事务数据集转换为序列数据库; 最后, 采用改进的 Apriori-based 算法发现紧耦合的功能模块, 进行功能之间的优化组合, 提升业务人员的工作效率. 实验表明该方法在揭示业务功能模块的耦合性方面的有效性.

关键词: 电力业务系统; 数据预处理; 序列模式挖掘; 功能优化

Improved Apriori-Based Algorithm Application to the Electric Power Business System Function Optimization

WANG Cheng-Xian, HU Yang-Bo, TAN Jing

(Jiangsu Electric Power Information Technology Co. Ltd., Nanjing 210024, China)

Abstract: This paper proposes an algorithm for electric power business system function optimization. Firstly we extract log data from the Web server and the client user. Then, we preprocess the user log dataset, and convert transaction dataset into a sequence data. Finally, we use the improved Apriori-based algorithm to find tight coupling function modules, and make optimization combination between the functions to improve the working efficiency of the business. Experiments show that the method is effective in revealing the coupling of the business function module.

Key words: electric power business systems; data preprocessing; sequential pattern mining; function optimization

国家电网公司为深入推进“两个转变”, 实现“一强三优”现代公司的战略目标, 提出了建设坚强智能电网和三集五大体系的重大举措[1], 这就要求信息化建设进一步提升, 再上水平. 促进信息系统集成的一体化平台相关工作作为功能重要的基础性工作, 对推进公司信息化进程和保障公司信息化持续发展有着举足轻重的作用. 2008 年上线的统一框架平台(统一权限和统一工作流), 为所有业务系统提供了统一权限管理平台. 经过几年的推广实施, 统一框架平台已经集成了江苏电力各业务系统的功能, 全面管理江苏电力所有信息化用户, 近年来积累了大量的用户日志信息, 通过挖掘分析用户每日对业务系统和功能菜单使用轨迹形成的时间序列, 挖掘高关联的业务功能模块, 即紧耦合的功能模块, 并进行功能之间的优化组合, 从而进一步完善业务系统, 优化系统服务.

序列是事件的有序列表. 对于时间序列数据, 其序列数据由相等时间间隔记录的数据长序列组成. 时间序列数据包含不同时间点重复测得的数值序列, 可以由多个自然或经济过程产生. 序列模式挖掘关注时间序列模式, 最早是由 Rakesh Agrawal 等人针对购物篮数据分析提出来的[2], 序列模式是一个存在于单个序列或一个序列集中的频繁子序列. 序列模式挖掘就是挖掘相对时间或其他模式中出现频率相对较高的模式, 在 Web 日志挖掘、文本检索、生物基因学、金融股票分析等领域中的应用前景十分广泛.

国内外对序列模式挖掘的研究大体上可分为三类[3,4]: 一类是基于 Apriori 特性的序列模式挖掘算法, 该算法包括排序阶段、大数据项目阶段、转换阶段、

① 收稿时间:2014-06-30;收到修改稿时间:2014-08-08

序列阶段、选最大阶段, 该类算法存在缺少时间限制, 事务的定义过于严格缺少分类层次等问题; 一类是基于投影的序列模式挖掘算法[5], 该算法无需产生候选频繁序列模式, 挖掘对象较小, 但是需要构造投影序列数据库, 增加了挖掘的工作量; 第三类是基于 SPADE 算法的序列模式挖掘[6], 该类算法的最大优点是大大减少了扫描数据库的次数, 但是由于采用广度或深度优先搜索遍历, 需要付出巨大的候选码代价. 针对上面三种序列模式挖掘的不足, 本文采用一种改进的 Apriori-based 方法.

1　系统总体架构

本文从满足电力企业和用户的实际需求出发, 着眼于江苏电力业务系统所积累的大规模用户日志数据, 构建如图 1 所示的系统总体框架, 包括提取事务集、序列模式挖掘和系统应用三个模块. 其中, 提取用户事务集模块是该系统的基础, 主要负责抽取与挖掘目标相关的电力业务系统用户日志数据, 并对其进行预处理(本文主要完成数据插值), 生成供挖掘使用的行为事务集; 序列模式挖掘模块的主要是将用户行为事务集转换为序列数据库, 并利用改进的 Apriori-based 算法从序列数据库中挖掘出频繁序列; 应用模块是分析系统功能模块的耦合性, 实现系统功能模块的优化及完善功能模块设计.

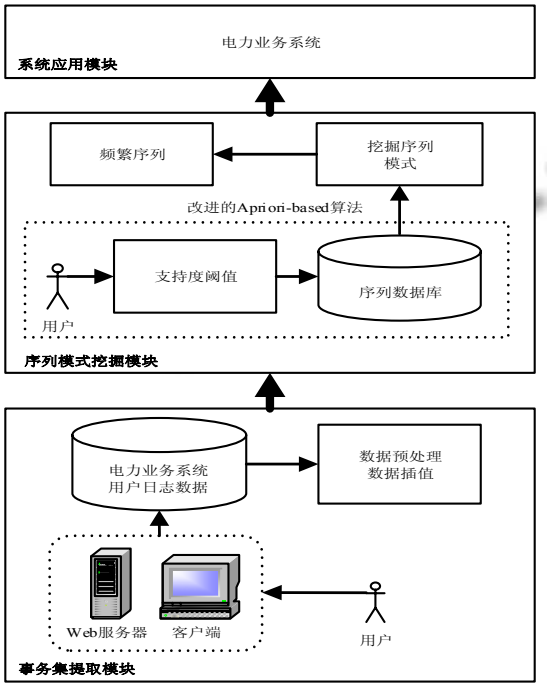

图 1　系统总体框架

2　事务集提取模块

事务集提取模块包含两个阶段, 数据采集和数据预处理. 其中, 数据采集是通过响应方式获取原始数据, 而由于原始数据的不完整和不规则特性, 需通过数据预处理对其进行清洗和处理, 以便适用于之后的工作.

3　用户日志数据数据采集

目前, 数据采集的主要途径有两个:

1) 基于服务器日志的数据采集方法. 服务器日志文件是记录 Web 服务器活动的一种重要工具, 主要通过 Web 服务日志文件中记录的客户端 Http 请求的相关信息, 统计用户访问行为数据. 服务器日志文件提供了详细的客户和服务器的交互活动日志.

2) 基于客户端的数据采集方法. 该方法可利用 Java Script 技术、Java Applet 技术和网页跟踪帧技术直接从客户端获得数据, 能够获得大量的难以从服务器端获得的用户行为数据.

4　数据预处理

电力业务系统数据来源于多个数据源的未被加工、高维、冗余、含有噪音且非均匀分布的复杂数据, 在数据模型、含义、模式、结构和语义上存在不一致性和冲突, 直接使用这些数据进行挖掘分析, 往往会降低挖掘的效率, 甚至产生错误的结果. 因此序列数据预处理研究是数据挖掘研究中的一个重要方面. 常用的预处理技术有数据清洗、数据集成、数据变换和数据插值等[6,7], 本文主要聚焦于数据预处理中的数据插值的方法研究.

电力业务系统中虽然积累了大量的用户日志数据, 但是这些数据是离散的, 有时因为采样点之间过于稀疏, 影响数据挖掘工作的开展, 因此, 需要进行数据插值处理不完整的数据. 由于拉格朗日插值法具有格式整齐规范, 有误差估计公式等优点, 本文采用拉格朗日(Lagrange)插值法[8]做相应的处理, 以适应挖掘分析.

设 $f(x)$在$[a, b]$上具有$(n+1)$阶连续导数, 对于$(n+1)$个结点$(x_j, y_j), j = 0, 1, \ldots, n$, 其中 x_j 互不相同, 满足:

$$f(x_j) = y_j \tag{1}$$

则存在 n 次插值多项式:

$$L_n(x) = \sum_{i=0}^{n} y_i l_i(x) \quad (2)$$

其中 n 次多项式:

$$l_x(x) = \frac{(x - x_0)(x - x_{i-1})(x - x_{i+1})\ldots(x - x_n)}{(x_i - x_0)\ldots(x_i - x_{i-1})\ldots(x_i - x_n)} \quad (3)$$

5 序列模式挖掘算法

5.1 序列模式挖掘相关概念以及挖掘的一般流程

传统的时间序列挖掘方法多是基于概率统计的方法实现, 主要集中在时间序列的建模和预测, 随着 Web 2.0 技术的发展, Web 日志数据大量积累, 如江苏电力公司目前存在大量的业务系统用户行为日志, 因此需要从这些用户行为日志中发现在相同的时间区间或相同的数据集在不同时间区间的相似性和差异性, 从而提取相似序列模式, 挖掘隐藏在大量 Web 日志数据背后的知识. 本节首先定义了序列模式.

1) 序列模式相关概念

在研究序列模式挖掘算法之前, 需要了解序列模式的相关概念[9].

定义 1. 项目(Item): 单个的、最基本的元素, 是不可再分的.

定义 2. 元素(Element): 由不同的项目组成的集合, 单个项目组成的集合可以省略括弧.

定义 3. 序列(Sequence): 由多个不同的元素有序排列组成的, 如序列 S 可以表示为 $S=<s_1, s_2, \ldots, s_l>$, $s_j(1 \le j \le l)$为序列 S 的元素.

定义 4. 序列数据库(Sequence database): 由多条序列组成的序列集合.

定义 5. 序列的支持度(Support): 序列数据库 D 包含序列 S 的数量占总数量的比重.

定义 6. 序列包含: 对于两个序列 $S=<s_1, s_2, \ldots, s_l>$, 和 $T = <t_1, t_2, \ldots, t_m>$, 如果存在整数 $i_1<i_2<\ldots<i_n$, 且 $s_1 \subseteq t_{i1}, s_2 \subseteq t_{i2}, \ldots, s_l \subseteq t_{il}$, 则称 S 包含于 T, 记作 $S \subseteq T$.

定义 7. 序列模式(Sequential Pattern): 在给定一个序列数据库以及最小支持度 min_sup 的条件下, 找出所有满足最小支持度的序列, 每个这样的序列称为一个序列模式.

2) Apriori-based 序列模式挖掘算法

用户的功能菜单点击行为具有时间序列特征, 序列模式挖掘需要根据输入的序列数据集, 将与对象有关的所有时间按时间戳增序排序, 挖掘某个记录与其他对象相关联的一些记录. 现有的 Apriori-based 序列模式挖掘算法指定任意序列数据集 D 和最小支持度阈值 min_sup, 求所有支持数大于阈值的序列(即序列模式), 具体的挖掘流程如下:

步骤 1: 扫描 D, 根据指定的最小支持度阈值 min_sup 挖掘出频繁项集, 并映射为简单、可识别的符号;

步骤 2: 将 D 中的每个序列转换成它所包含的频繁项集的格式, 并用映射符号表示;

步骤 3: 对经过变换处理后的数据采用适合的挖掘算法进行序列模式挖掘;

步骤 4: 对经过序列模式挖掘后产生的一系列序列模式的处理, 删除没有应用价值的序列模式, 归纳、整理、分类序列模式.

该算法具有如下缺点: 1) 用户需要制定序列模式的相邻元素的时间间隔, 如一个序列模式可能发现某个用户在点击了业务功能 A 后一个月以后点击功能 B, 即需要挖掘给定时间内的用户序列模式; 2) Apriori-based 序列模式挖掘需要多次扫描数据集 D, 同时产生的大量的候选数据集, 需要花费大量的时间来挖掘到频繁序列模式. 基于 Apriori 序列模式挖掘算法, 本文提出改进的 Apriori-based 挖掘算法, 如 5.2 节所示.

5.2 改进的 Apriori-based 挖掘算法

电力业务系统中的用户行为集合记为 $I = \{i_1, i_2, \ldots, i_n\}$, I 中的项目 $i_j(1 \le j \le n)$是电力业务系统中用户使用的各种功能模块, n 表示 I 中项目的数量. 设 X 为用户行为事务, 记为<*id, user_id, s, time*>, 其中 *id* 为事务标识编号, *user_id* 为用户标识编号, s 为某个时间用户使用的功能模块的集合, 称为事务集; time 是事务发生时间. 序列 $S = <s_1, s_2, \ldots, s_n>$为用户使用功能模块事务集序列, S 中的所有事务具有相同的用户标识编号 *user_id*, 并且事务按时间顺序排列. 通过对用户浏览行为的事务数据进行转换, 把用户标识相同的记录合并, 将具有相同用户标识的浏览行为事务按时间排序, 得到事务集序列, 继而得到序列数据库 D. 在这个序列数据库上实现序列模式挖掘. 用户对电力业务系统用户行为序列模式挖掘的任务, 就是从用户使用的功能模块事务序列中, 找出用户最感兴趣的频繁事务序列模式.

本文采用一种改进的 Apriori-based 的序列模式挖掘算法[10], 完成挖掘过程, 该算法采用逐层搜索的迭代方法, 具体实现过程如下:

表 1　改进的 Apriori-based 的序列模式挖掘算法

输入: 序列数据库 D 和最小支持数阈值 min_sup

输出: 序列数据库中的序列模式集

改进的 Apriori-based 的挖掘算法:

1　L_1 = {large 1 - sequences}; //扫描序列数据库 D, 得到频繁 1-序列模式集 L_1

2　For (k = 2; $L_{k-1} \neq \emptyset$; k++) do begin //循环迭代, 直到不能找到频繁 k-序列模式为止

3　　C_k = Apriori-new generate(L_{k-1}); //由频繁(k-1)-序列模式生成候选 k-序列模式集 C_k

4　For each sequence s in the database D do

5　　Increment the count of all candidates in C_k that are contained in s; //扫描 D, 计算 C_k 中的各序列模式的支持度

6　End For

7　L_k = Candidates in C_k with minimum support; //生成频繁 k-序列模式集 L_k

8　End For;

9　Return $U_k L_k$; //返回序列模式集

本算法需要多次扫描序列数据库, 在第一次扫描中, 对所有的单个项目(1-序列模式)进行计数. 利用频繁 1-序列模式生成候选频繁 2-序列模式, 进行第二次扫描并求候选频繁 2-序列模式的支持数. 使用频繁 2-序列模式生成候选频繁 3-序列模式, 如此下去, 直到不能找到频繁序列模式为止.

6　实验与结论

为了验证改进的 Apriori-based 序列模式挖掘算法的有效性, 本文使用了江苏电力信息公司的业务系统用户行为数据集, 并以一个月为周期来提取用户行为日志, 该数据集包含了 1233 个用户、3224 个业务系统功能菜单以及 112832 条用户业务系统点击记录. 运行环境为 Windows XP, 主频 2.8GHz, 内存 2G, 硬盘 500G, 程序使用 Java 语言实现. 本文主要完成两组实验, 第一组固定最小支持度 *min_sup*, 不断变化候选频繁项数 k(k≥2), 第二组固定频繁项数, 不断调整 *min_sup*, 并与现有的 Apriori 算法在运行时间上进行比较来验证算法的有效性.

第一组实验选取 *min_sup*=5, 候选频繁项数 k 分别设置为 2、3、4、5 和 6, 实验结果图如图 2 所示. 在图 2 中可以看出在候选频繁项数为 2 时, 两种算法所花费的时间相同, 但当 k>2 时, 改进的算法挖掘候选频繁项时所花费的时间比现有的 Apriori-based 序列模式挖掘算法所花费的时间要少, 表明采用改进的 Apriori-based 能有效提高挖掘业务功能模块的频繁序列挖掘效率.

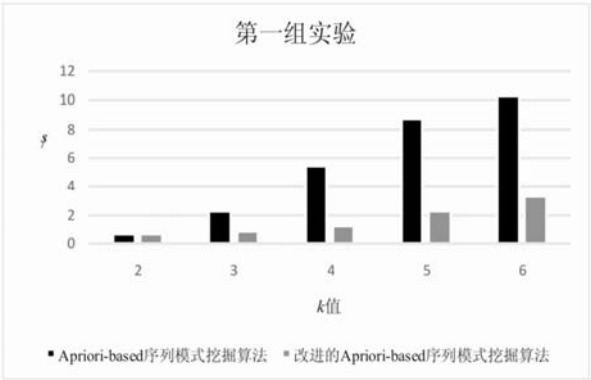

图 2　第一组实验的实验结果图

第二组实验中, 选取了候选项数 k=5, *min_sup* 分布设置为 2、3、4、5 和 6, 实验结果如图 3 所示. 在图 3 中可以看出改进的算法挖掘候选频繁项时所花费的时间比现有的 Apriori-based 序列模式挖掘算法所花费的时间越少.

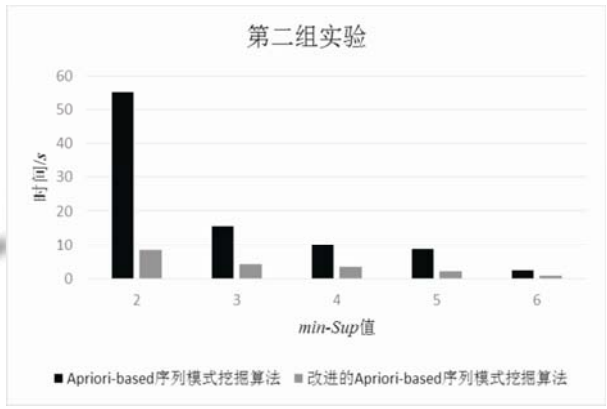

图 3　第二组实验的实验结果图

两组实验结果表明, 改进的 Apriori-based 算法能够在挖掘频繁序列模式时花费的时间要远远小于现有的 Apriori-based 序列模式挖掘算法, 说明改进的算法更能够有效的挖掘业务系统用户行为数据集的频繁序列模式, 即紧耦合的功能模块, 在以后的工作中, 考虑将这些紧耦合的功能模块放入同一个业务系统中, 方便用户操作业务系统, 提升系统服务功能.

参考文献

1 彭林,马骞宙,黄文思,李浩松,廖小云.信息化支撑智能电网建设.电力信息化,2011,9(2):148–151.

2 Srikant AR. Mining sequential patterns. Proc. 95 Int'l Conference Data Engineering. Taibei, Taiwan. March, 1995. 3–1.

3 王虎,丁世飞.序列模式挖掘研究与发展.计算机科学,2009,36(12):14–17.

4 吴孔玲,缪裕青,苏杰,张晓华.序列模式挖掘研究.计算机系统应用,2012,21(6):263–271.

5 Kalpakis K, Gada D, Puttagunta V. Distance measures for effective clustering of ARIMA time-series. Proc. of the 2001 IEEE International Conference on Data Mining. San Jose, U.S.A. 2001. 273–280.

6 Liao TW. Clustering of time series data-a survey. Pattern Recognition Society, Pattern Recognition, 2005, 38(11): 1857–1874.

7 郭长艳.简论数据挖掘中数据预处理技术的功能模块.东南大学学报,2009,15(3):90–92.

8 卓飞豹.多变量时间序列的预处理和聚类研究[学位论文].福州:福建师范大学,2009.

9 Vijayalakshmi S, Mohan V, Sasirekha MS, et al. Extracting sequential access pattern from pre-processed web logs. Proc. of 2011 International Conference on PACC. Coimbatore, India. 2011. 1–6.

10 Ashish P, Aisha P. Graph based approach and clustering of patterns (GACP) for sequential pattern ming. International Journal on Computer Science and Engineering, 2011, 3(4): 1501–1504.