频繁模式挖掘系统的设计与开发

王楠楠,刘慧婷
(安徽大学 计算机科学与技术学院,安徽 合肥 230601)

摘 要: 在日常生活或者相关科学研究中,使用电子设备会产生大量的数据,如何从数据中删除冗余信息,提取或“挖掘”有用信息就成了当前信息科学和技术领域的一个重要的研究方向。频繁模式挖掘作为众多挖掘算法中的一类基本算法,研究主要包括项目集合、项目序列和时间序列等各种数据中的频繁模式挖掘。频繁模式挖掘算法众多,如数据流频繁闭项集挖掘、不确定数据流的最大频繁项集挖掘和不确定数据的频繁模式匹配。该系统设计的目的是将几个课题组开发的挖掘算法进行集成,并利用可视化界面对算法的性能进行直观的比较。通过系统的可视化界面,可将解决同一问题的多个算法的运行结果放入同一张图中,方便用户查看算法的输出结构并进行算法优劣性的比较。

关键词: 频繁模式挖掘; 可视化; 算法集成; 算法比较
中图分类号: TP302 **文献标识码:** A **文章编号:** 1673-629X(2018)02-0150-04
doi: 10.3969/j.issn.1673-629X.2018.02.032

Design and Implementation of Frequent Pattern Mining System

WANG Nan-nan, LIU Hui-ting
(School of Computer Science and Technology, Anhui University, Hefei 230601, China)

Abstract: The electronic device can produce a large amount of data in daily life or relevant scientific research. Therefore, how to remove the redundant information, and extract or “mine” useful information from the data has become an important research direction presently in the field of information science and technology. Frequent pattern mining is basic one of many mining algorithms. The research mainly includes the frequent pattern mining in item sets, item sequence and time series. There are many kinds of frequent pattern mining algorithms like the data flow frequent closure mining, the maximum frequent item set mining of uncertain data streams, and frequent pattern matching of uncertain data. In order to facilitate users to view the results, we design and implement a system to integrate some pattern mining algorithms together. Through the visual interface of the system, we can put the results of several algorithms which solve the same problem into a diagram, in order to make users view the output structure of the algorithms and compare them.

Key words: frequent pattern mining; visualization; algorithm integration; algorithm comparison

0 引 言

频繁模式挖掘是数据挖掘领域的一个重要研究方向[1-3],研究主要包括项目集合、项目序列等各种数据中的频繁模式挖掘[3-4]。频繁模式挖掘得到的结果不但可以用于关联规则、文本分类等其他数据挖掘课题,而且可以应用于实际生活中的很多领域,如市场营销、银行、电信以及网络安全等。文中介绍了利用 QT5.5 开发的频繁模式挖掘系统。该系统根据内部嵌入的算法具有三大功能: 最大频繁项集挖掘、频繁闭项集挖掘和频繁模式匹配。用户可以通过可视化界面来选择所要实现的相应功能。系统设计的目的是方便用户以可视化方式比较“不确定序列模式匹配”、“不确定数据流最大频繁项集挖掘”和“数据流频繁闭模式挖掘”三个模块中所包含的不同算法在时间和空间上的优越性。

系统为一集成工具,为保证算法运行的正确性以及系统的健壮性,系统为半封装系统。用户可以按照系统规定的数据格式导入需要进行处理的数据源,修改算法中相应的参数,选择某个功能中的某一种算法或者某几种算法来执行,然后通过图表的方式比较算法的优劣。

1 相关理论和工具介绍

1.1 开发工具

(1) 软件。

收稿日期: 2017-02-20 **修回日期:** 2017-06-28 **网络出版时间:** 2017-11-15
基金项目: 国家自然科学基金(61202227)
作者简介: 王楠楠(1996-),男,研究方向为机器学习; 刘慧婷,博士,副教授,硕导,研究方向为数据挖掘、机器学习。
网络出版地址: http://kns.cnki.net/kcms/detail/61.1450.tp.20171115.1425.026.html

系统使用 QT 开发而成,版本为 QT Creator3.4.2 base on QT 5.5.0。QT 是 1991 年由奇趣科技开发的一个跨平台 C++图形用户界面应用程序开发框架,是面向对象的框架,使用特殊的代码生成扩展以及一些宏,易于扩展,允许组件编程,而且 QT 具有优良的跨平台特性,在多个操作系统下均可运行[5]。

(2) 开发语言。

C++是在 C 语言的基础上开发的一种面向对象编程语言,应用广泛。C++支持多种编程范式—面向对象编程、泛型编程和过程化编程。其常用于系统开发、引擎开发等领域,是迄今为止最受广大程序员欢迎的编程语言之一,支持类,具有封装、继承、多态等特性。

1.2　系统实现算法

系统主要实现数据流频繁闭项集挖掘、不确定数据流的最大频繁项集挖掘和不确定数据的频繁模式匹配。

数据流频繁闭项集挖掘算法[6-7]基于数据流滑动窗口,利用一个在内存中的数据结构—封闭枚举树(closed enumeration tree,CET),来维护较小的动态候选集。类似于前缀树,树中的每个节点 n_i 代表一个项集 I。CET 只保存频繁闭项集以及频繁闭项集和其余项集之间的边界项集。一个新的事务 T 添加到滑动窗口时,需要遍历 CET 中跟 T 有关的部分。对于每个相关节点 n_I,不仅要更新它的 support 和 tid_sum,有可能还要更新它的节点类型。

不确定数据流最大频繁项集挖掘算法[8-9]根据数据流的特点以及不确定数据的产生原因、表现形式和处理模型,从节约存储空间和减少搜索空间两大切入点入手,基于衰减窗口实现最大频繁项集的挖掘。算法设计了存储不确定数据流概要信息的数据结构,并提出了高效的超集检测方法,同时采用标记树节点的方法避免超集检测,减少了搜索时间。

不确定数据模式匹配算法[10-12]则是采用动态规划策略解决不确定序列中的模式匹配问题。上述几种算法由课题组成员实现,该系统开发的目的是把这些算法集成起来,以可视化的方式把结果展现出来,给用户提供一个友好的频繁模式挖掘工具。

2　系统设计与实现

2.1　系统模型与框架设计

系统结构图如图 1 所示,其结构分为菜单栏、功能栏和状态栏三个部分,系统功能主要体现在功能栏[13]。功能栏包含五个模块: 算法参数修改模块(输入)、功能选择模块、结果输出模块、图表分析模块和外部文件的读写。算法参数修改模块主要是对算法运行过程中需要的关键参数进行修改; 功能选择模块则是由用户选择要执行的算法或结果输出的形式; 结果输出模块是将算法运行结果保存在 txt 文件和 xls 文件中方便用户查看; 图表分析模块将算法运行结果以可视化的形式展现给用户; 外部文件读写则是对 txt 文件的读取。

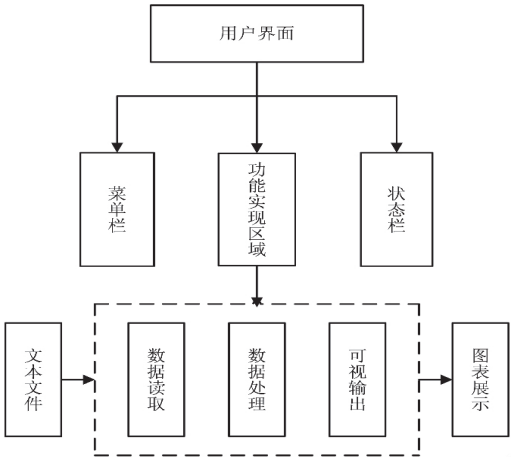


图 1　系统结构

系统具有以下特点:

(1) 整个系统三大功能包含的各个算法虽然共用一个界面,但是算法间的运行是相互独立的。这些算法的输入输出以文本文件为主,用户可以查看输入输出的内容,并可在一定条件下对输入文件进行修改。

(2) 该系统可以实现具有同一功能的不同算法之间的性能比较,所以每次运行系统时的输入数据应为不同的数据。算法运行结果只能保存到系统运行结束时,系统运行结束后,保存结果的文件会自动删除,该系统具备清除缓存文件的功能。

(3) 系统界面友好,简单明了。

2.2　系统功能算法嵌入

系统界面实现使用的语言以 C++为框架,对其进行函数及功能上的扩充。使用的工具是 QT5.0,QT 兼容 C++以及 C 的语法规则。该工具提供 design 模式和代码编写模式,该系统使用的是代码编写模式[14-15]。系统界面如图 2 所示,主要划分为三块: 菜单栏、内容区域和状态栏。

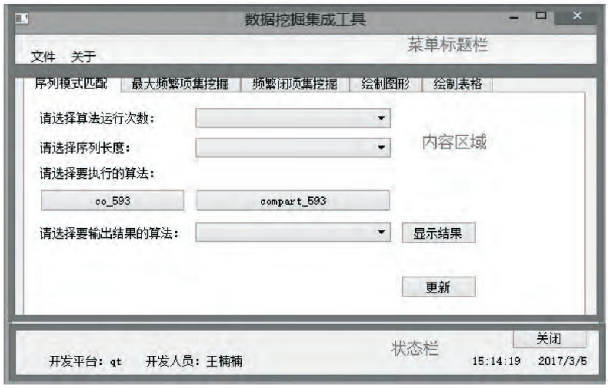


图 2　系统界面

界面中功能选择使用的是 QTableWidget 控件，每个功能页都是一个单独的 QWidget，将每个功能页布局完成后，最后再集成到 QTableWidget 中。前两个 QWidget 为算法功能，算法运行后为图表的显示提供数据。实现该界面的具体流程如图 3 所示。

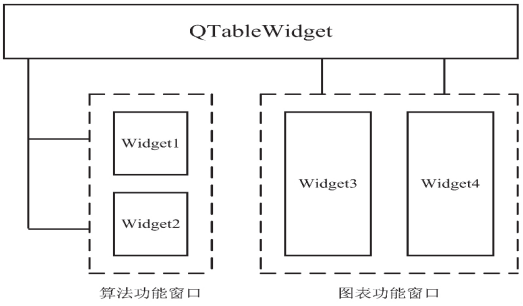


图 3 界面实现流程

点击界面中的按钮实现相应算法的调用使用的是 QT 中的信号槽机制，点击按钮后发出相应的信号，将该信号与要实现的函数槽链接起来，即可实现点击按钮调用算法的功能。

connect(Object, SIGNAL(signal1) , Object2, SLOT (slot))

其中，signal 为对象 Object 的信号; slot 为对象 Object2 的槽。

因为不同算法中包含有相同的函数名和变量名，为避免算法运行出现混乱导致程序出错，每个算法都是一个单独的程序，而用户则是在主界面点击功能按钮来调用这些程序。算法运行成功后，系统会弹出反馈窗口告知用户。

2.3 数据的输入输出处理

算法的输入及输出数据以文本文件的方式存储在文档中，因为 QT 内嵌的字符编码模式与通常的 C++ 编程工具不同，在读取数据时需要将数据进行转换。

为了使系统能够正常运行，对算法参数的输入限制较为严格，用户只能以选择的方式对输入的参数进行更改，具体由 QComboBox 控件实现，代码如下:

```
acombobox = new QComboBox;
acombobox->addItem( tr( "item1") ) ;
acombobox->addItem( tr( "item2") ) ;
acombobox->addItem( tr( "item3") ) ;
```

其中，item1，item2，item3 即为用户可以选择的输入参数。

2.4 图表的实现

系统中的图表主要包含三部分: 表格、柱状图、折线图。其中，表格由类 form 实现，算法在运行过程中会自动将运行时间输出到缓存文件中，当用户选择要查看的算法后，系统会自动读取数据，将其以表格的形式显示出来。

折线图与柱状图则分别通过类 line 和 histogram-view 实现，其界面为上表格下图的形式展现，系统会自动读取默认数据绘图，用户也可以手动选择要绘图的数据。表格中的数据可以修改，但不会改变其原始值。当表格中的数据发生更改时，绘制的图形也会发生变化，其主要由 datachange 函数实现，代码如下:

```
void dataChanged( const QModelIndex &topLeft, const QMod-
elIndex &bottomRight)
{
QAbstractItemView: : dataChanged( topLeft, bottomRight) ;
viewport( ) ->update( ) ;
}
```

3 系统功能演示

3.1 功能及页面

系统界面中多为功能按钮，用户选择要实现的功能，点击功能按钮即可。以最大频繁项集挖掘算法为例，其界面如图 4 所示。

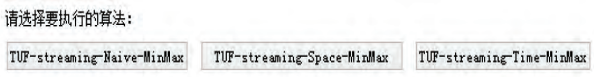
图 4 最大频繁项集挖掘功能演示图

图中，修改参数选项为两组参数，用户点击功能按钮，在下拉菜单中选择要修改的值，两项选择完毕后选择要执行的算法，待反馈窗口出现后，算法即运行成功。如果想查看结果，选择要查看的算法，点击功能按钮即可打开结果保存文件。

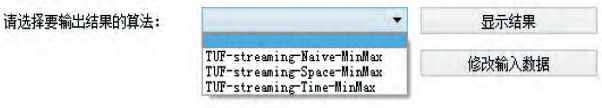
3.2 功能演示

(1) 首先选择输入参数:

(2) 点击要运行算法的功能按钮:

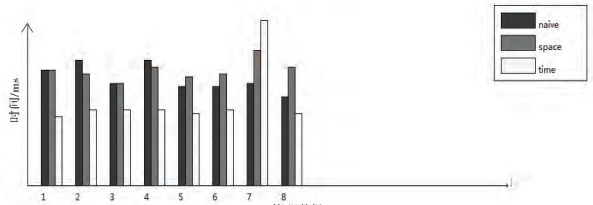


(3) 算法运行成功，选择要查看的内容:



(4) 选择绘制图形功能。该功能为将各个算法运行多次，取平均值进行比较，用户可以选择绘制折线图或柱状图(默认显示图形为该算法运行时间比较，用户可以点击“文件->打开”选择 memory02.txt 文件，来显示算法内存占用比较) ，如图 5 所示。

(a)折线图



(b)柱状图

图 5　算法运行结果

(5) 表格绘制功能。选择相应的算法,即可绘制表格(该功能为对每一个输入数据的运行状况进行比较)。该系统部分算法提供修改输入数据,修改内容格式必须与原内容格式相同,否则算法将运行出错。算法在选择参数时,必须选择确定的值,不可不选,否则算法无法正常运行。表格输出结果如图 6 所示。

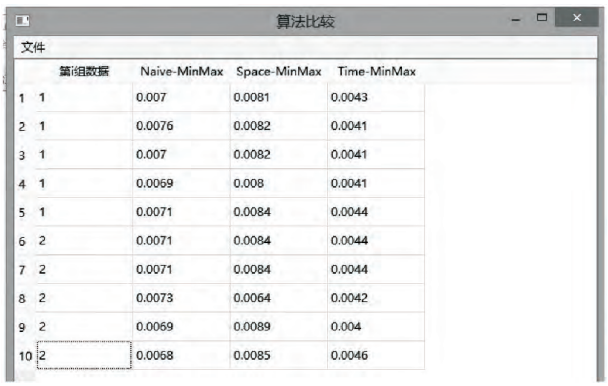


图 6　表格输出结果

4　结束语

文中频繁模式挖掘系统为一算法集成工具,为用户提供一个可视化界面,点击按钮即可实现相应的功能。系统在提供算法功能的同时也提供了绘制图表的功能,图表功能主要是为了实现算法性能的可视化比较,通过多次运行同一算法取平均值,用户可以选择使用折线图与柱状图显示。为了使系统能够良好运行,对算法的输入做了严格的限制,只允许用户修改部分参数。因为系统的主要功能为实现算法性能的可视化比较,所以在运行各个窗口下的算法时,必须全部运行,以防止绘图数据的错乱。

系统运行的算法的输入格式是固定的,当用户更改输入格式时会使系统无法运行。要使系统可以运行更加复杂的算法,需要对集成方法进行改进,使系统的运行与算法的执行分离。这将是下一步的改进方向。

参考文献:

[1] 白川平.多数据流的频繁模式挖掘算法研究[J].宁夏师范学院学报,2014,35(3):86-89.

[2] AGRAWAL R,SRIKANT R.Fast algorithms for mining association rules[C]//Proceedings of the 20th international conference on very large databases.San Francisco,CA,USA: Morgan Kaufmann Publishers Inc.,1994: 487-499.

[3] 战立强.频繁模式挖掘算法研究[D].哈尔滨: 哈尔滨工程大学,2007.

[4] 马海兵.频繁模式挖掘相关技术研究[D].上海: 复旦大学,2005.

[5] 陆文周.QT5 开发及实例[M].北京: 电子工业出版社,2015.

[6] CHI Y,WANG H,YU P S,et al.Moment: maintaining closed frequent itemsets over a stream sliding window[C]//Proceedings of fourth IEEE international conference on data mining.[s.l.]: IEEE,2004: 59-66.

[7] 徐玉生.频繁模式挖掘算法与剪枝策略研究[D].兰州: 兰州大学,2008.

[8] 汤克明.不确定数据流中频繁数据挖掘研究[D].南京: 南京航空航天大学,2012.

[9] 刘慧婷,候明利,赵　鹏,等.不确定数据流最大频繁项集挖掘算法研究[J].计算机工程与应用,2016,52(19):72-77.

[10] LI Yuxuan,BAILEY J,KULIK L,et al.Efficient matching of substrings in uncertain sequences[C]//Proceedings of the 14th SIAM international conference on data mining.Philadelphia,Pennsylvania,USA: [s.n.],2014: 767-775.

[11] ZHAO Zhou,YAN Da,NG W.Mining probabilistically frequent sequential patterns in large uncertain databases[J].IEEE Transactions on Knowledge and Data Engineering,2014,26(5):1171-1184.

[12] GE Tingjian,LI Zheng.Approximate substring matching over uncertain strings[J].Proceedings of the VLDB Endowment,2011,4(11):772-782.

[13] 陈畅伟.基于关联规则的数据挖掘可视化系统的实现[D].武汉: 武汉理工大学,2004.

[14] 戴　巍,霍　亚,马尚昌,等.Qt 下基于组件的嵌入式软件框架设计及实现[J].计算机应用,2016,36: 257-261.

[15] 贺　青,李鹏飞.基于 Qt 的电脑横机上位机的设计[J].微型机与应用,2012,31(19):14-17.