中国科技资源导刊 ISSN 1674-1544
2008年11月 第40卷第6期 30-38
CHINA SCIENCE & TECHNOLOGY RESOURCES REVIEW
ISSN1674-1544 Vol.40 No.6 30-38,Nov.2008

大规模数据挖掘中的数据流管理

[美国] Jon R. Wright Gregg T. Vesonder Tamraparni Dasu
(美国电话电报公司实验室 - 研究所,美国新泽西州弗伦翰公园 07932-0971)
杨庆燕[译]
(清华大学计算机科学与技术系,北京 100084)

摘 要:在企业环境中,管理数据流或者实时数据更新是任何数据挖掘操作的一个主要挑战。无论是数据还是元数据,都要确保数据流的稳定、正确、可验。在这种环境下,实时数据更新很复杂,且数据量大而难懂。管理频繁变化的数据和元数据对企业是巨大的挑战。本文阐述了在管理企业数据的任务中的技术问题,并提出了一种解决方法。这种解决方法可以结合多个领域里的知识,如工程技术和统计学,来理解和标准化企业挖掘的准备工作,以使信息采集和质量管理自动化。

关键词:信息质量;数据挖掘;传送管理;知识工程
中图分类号: G203 **文献标识码**: A **DOI**: 10.3772/j.issn.1674-1544.2008.06.004

Management of Data Streams for Large-scale Data Mining

Jon R. Wright, Gregg T. Vesonder, Tamraparni Dasu
(AT& T Labs - Research, Florham Park, New Jersey 07932-0971, USA)
Translator: Qingyan Yang
(Department of Computer Science and Technology, Tsinghua University, Beijing 100084)

Abstract: In an enterprise setting, a major challenge for any data mining operation is managing data streams or feeds, both data and metadata, to ensure a stable and certifiably accurate flow of data. Data feeds in this environment can be complex, numerous and opaque. The management of frequently changing data and metadata presents a considerable challenge. In this paper, we articulate the technical issues involved in the task of managing enterprise data and propose a multi-disciplinary solution, derived from fields such as knowledge engineering and statistics, to understand, standardize, and automate information acquisition and quality management in preparation for enterprise mining.

Keywords: information quality, data mining, feed management, knowledge engineering

1 引 言

Witten 和 Frank[1], Piatetsky-Shapiro 等[2]和另外一些学者注意到,工业数据挖掘中,在数据挖掘算法应用之前经常需要做大量的工作,这和我们研究真实世界中应用的经历是一致的。在过去的几年中,我们已经能够应用拨号服务数据挖掘工程。作为一个测试平台,这项工程服务于信息质量的想法和实验,对于所有层次的企业都是

第一作者简介:Jon R. Wright (1947-),男,美国电话电报公司实验室高级技术总监,研究方向是电话呼叫数据挖掘。

收稿日期:2008年9月1日。

可见的，并向我们提供了大范围的复杂数据。根据在这项应用中的经验开发了监测方法，从而在某种程度上满足了实际的需求。从根本上说，流入这项工程中的数据量在自动监测工具的要求范围之内，或者至少满足了某些辅助监测系统。所以，我们的想法是，尽管在研究计划中偶尔有些漏操作，但在真实世界的应用中接受了严格的测试。关于应用的某些方面将在后续章节中具体阐述。

我们发现，在现实数据挖掘工程中，90%的工作量花费在数据的采集、准备、管理和其他相关的方面，而分析只占10%的工作量。很多数据挖掘方面的学者却没有意识到这一点，而是将注意力集中在越来越先进的分析和发现算法上，而这些算法恰恰依赖于事先存在的数据库，比如world wide telescope[3]。可以说，大范围的数据采集和管理所涉及的核心实际问题还没有得到应有的关注。

成功管理数据，包括分析之前的准备，可能是任何实际数据挖掘应用成功的关键因素。例如，最有意思的数据来源于支持企业核心事务处理的计算机应用。这些数据采用多种形式——交易记录、应用日志、网络碎片、数据库垃圾等。辅助计算机系统由于它们在事务中的关键作用，几乎总是禁止CPU承担数据挖掘的任务，于是只好把数据传送到更适合数据挖掘的计算环境中。由于现代企业中辅助计算机的增加和核心事务处理的自动化，挖掘中可得到的数据量很大，并在逐渐增长。另外，在企业中，数据挖掘有时是重要事务处理的唯一窗口。元数据由于对解释和理解企业数据有着重要的作用，其数据量也相当大。数据和元数据必然随着企业事务环境的变化而不断扩大。我们原本认为，通过数据仓库可以解决大多数问题，但是现代数据挖掘应用所要求的规模和灵活度使得数据仓库的解决方法缺乏有效和灵活性。

信息质量和信息管理中的许多关键问题以前有所提及[4,5]。特别是，保持高级别的信息质量和数据传送管理密切相关。本文将集中在实现信息质量管理技术所需要的工具和方法上，如在有关企业信息质量管理的出版物中讨论了从数据收集到数据挖掘过程中的技术。我们把整个过程看作大自然中的生命过程，强调在真实环境中数据的增长和变化。

下面讨论工业数据挖掘应用中的就位（mise en place)问题。

在数据挖掘处理中我们用“就位”（mise en place)来表示与采集、准备、管理和解释数据相关的过程。“mise en place”是一个烹饪词汇，意思是放在合适的位置。它描述的是在实际烹饪过程中的量、切、剥等行为。类似地，企业数据必须为数据挖掘做好准备，包括：①收集时间相关的元数据；②为短期的数据和元数据提供和管理长期的存储空间，便于推导出纵向趋势；③监测和提高数据的质量。在许多工业环境中，一个数据挖掘项目的大部分努力都集中在这些“mise en place”行为里。没有这些行为，数据挖掘就不可能有意义[1]。关于数据准备Pyle[6]作了相关论述。

我们的目标是让读者了解这些活动所起的重要作用，通过辅助工具、方法和技术使数据准备标准化，减少在数据准备中的工作，以更好地做好数据挖掘，同时将这些准备过程和数据挖掘放在一个连续、展开的工具集、处理和知识背景中。理解、标准化和处理自动化用到的技术综合了人工智能和统计学知识，包括工程学知识、计划分配、约束管理和基于规则的设计。

下面首先通过讨论过去几年数据挖掘的应用，了解数据流管理中的主要挑战。然后讨论我们开发的检测数据在传送中缺失和毁坏的方法。最后讨论由人工智能衍生出的一些自动化技术。在最后一节中，所有的活动都放在信息质量和信息管理的背景中。我们研究的最终目标是设计和提高技术，减少这些活动的时间和代价，将这些“其中一个”转换成可重复的、自动处理的和抽象的通用技术。

2 拨号服务数据挖掘

我们将从广泛意义上以一个真实世界中企业数据挖掘的应用为例，说明前面讨论的内容，集中讨论数据传输、数据集成和存档中的一系列活动。下面提到的拨号服务数据挖掘将简称为

TDM 应用。我们首先讨论数据传输管理的问题。TDM 应用建立在聚类技术基础上，使用 ATT 实验室开发的商用硬件。这种技术在 Hume and Daniels[7]中有描述。这种聚类技术有一些非常有趣的特征，尤其是在监测和保存大规模数据的完整性方面表现出的能力，提供了对那些数据执行分析和发现算法所需的计算引擎。

数据传输采用图 1 所示的三层架构来管理。

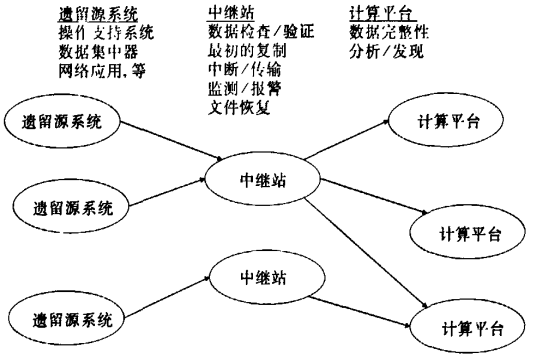

图 1 三层数据流体系架构

图 1 所示的源系统非常不均匀，许多是主机。因为主机运行着许多应用，单个主机可以产生一个或者多个数据流，另外，源系统中的一部分是分布在更大系统集合后面的集中器。在许多情况下，数据按时间表自动产生，但是部分是由数据中心的操作员初始化的程序产生的。例如，源系统有足够的临时存储能力，在删除之前短期保留产生的数据（也许 48 小时）。这是因为如果传输失败或者被毁坏，需要有一段时间来恢复数据。

图 1 所示的计算平台为数据分析提供资源，在它们的辖域中主要保持数据的完整性。另外，许多有趣的活动发生在中继站。一个文件和其他东西首先被一个中继站接收，这个文件通过数据传送专用审查程序，确保这个文件头及其内容相匹配。接着数据被压缩，进行数据最初的复制，然后数据被传送到一个合适的计算平台上。复制是用来保持数据完整性的基本工具[7]。我们希望在中继节点上集中执行两项特定功能；监测到达该节点的文件流并在出现问题时发出信号；在数据丢失之前采取措施恢复数据。

每个数据流都有自己的特征。一些数据流每天 24 小时都在发送固定数量的文件。还有一些数据流发送的文件数量是变化的，依赖于源系统上的活动，如在某个遗留系统进行的交易。这种数据除了反映上升或者下降趋势之外，还反映了具有每天、每周和季节周期。其他数据流可能发送固定数量的文件，但只在工作日发送。也就是说，在周末和节假日里不发送文件。文件可以以一周、一个月或者一个季度为周期来发送，也可以每天发送。

典型地，操作员使用各种专门工具监测数据流，并在检测到异常时采取恰当的措施。监测的频率不统一，可能偶尔在某段时间内的问题未检测到。这项任务因为穿过中继站数据的规模而变得复杂。TDM 应用每月从大约 100 个源中收到超过 60T 的未经压缩的数据。通过应用好的数据压缩方法，我们能够把所需的存储量降到每月大约 6T。一旦数据在中继站被压缩，它将不再以非压缩的形式存在于磁盘上。每月收到的文件数千个。一个遗留系统每月发送大约 8 万个文件，且这些数据的大部分分析需要完整文件的全集。也就是说，数据缺失将使分析无效。

3 在数据流管理中存在的问题及解决方法

3.1 使用统计总体和代理监测器监测异常

(1) 统计过程控制模型。不管流过系统的数据量有多少，TDM 都不会丢失数据。于是，尽早监测到受毁坏的数据以及丢失的文件和其他异常现象是很重要的，因为数据能在 48 小时的时间段内初始化得以恢复。我们的目标之一是使数据流的监测尽可能多地自动化。对于操作员来说，监测或者警戒工作是很困难的[8]，因此要尽可能多地减轻操作员的负担。统计处理控制已经有一个有用的考虑信息质量问题的框架。作为一种最初的警报机制，我们建立一个简单的单变量，通过无参数的控制图表来探测数据流的中断。这种中断要么因为文件丢失，要么因为文件的大小比通常情况的小。小文件表明数据从未传输或者在传输中丢失。

对单个数据流，图 2 和图 3 显示了每小时接收到的文件大小和总文件大小的波动。这个特定的数据流每小时发送固定数量的文件，但是文件的大小是变化的，它包含了反映商业活动变化的主要事务记录。图 2 显示文件的传输时有滞后，在某几个小时内收到的文件数量比期望值低，但在后来的时间里这些缺失的文件能得到补偿。

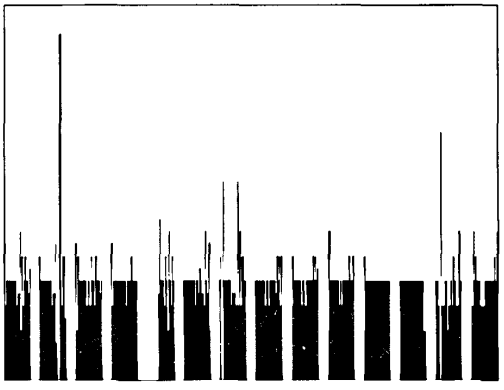

图 2　对于单数据流来说一周内每小时接收文件的数量

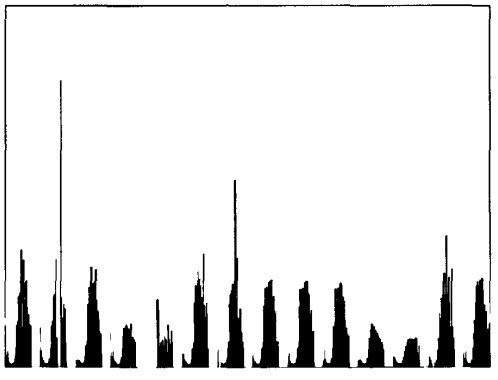

图 3　对于单数据流来说一周内每小时接收文件总量

图 3 清晰地展示了周期为一周内每小时接收文件的大小。大尖峰对应一个大量数据的传输，用于补偿该尖峰之前数据的缺失。

图 4 和图 5 显示了一天里文件的置信度界限。两幅图清楚地说明一天内的活动，它们的高点反映了“繁忙”时段。利用一个稳定时段内基于中值和四分位数间距的矩阵，可以计算出期望值和置信度界限。图 4 显示了一个正常周期，接收到的文件大小在期望值的可接受界限内。

图 5 显示了反常的一天，两个小时的小数据量传输，继而一个累积性传输的尖峰。这通常是因为传输线路的问题。

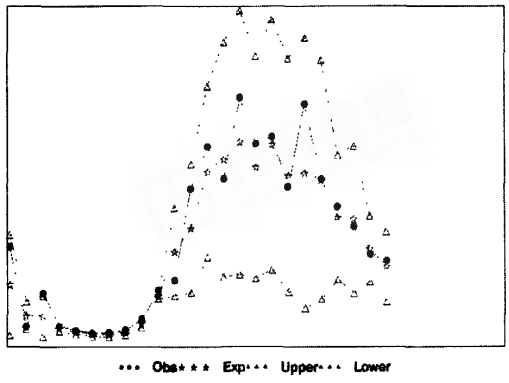

图 4　单天内显示的每小时接收文件的数量控制图
(中值，第 5 和第 95 分位点)

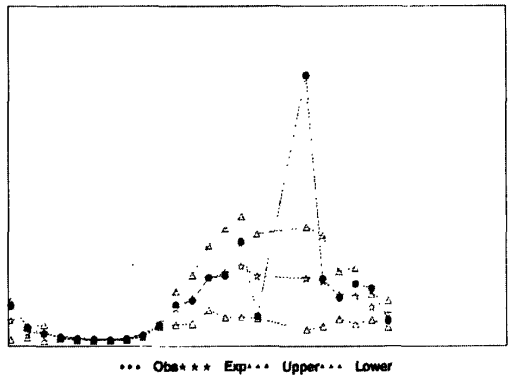

图 5　单天内显示的每小时接收文件的总大小控制图
(中值，第 5 和第 95 分位点)

这两幅图清楚地表明了基于统计处理控制的方式是合理的。使用这个范例，可以从历史数据计算出正常的界限。当某个数据点出界时发出警报。图 5 表明了出界的情况实际上是作为补偿的尖峰。从某种意义上说这是正常的起伏，代表一个内部公司网络，而不是异常状况，因为没有数据丢失。在这个尖峰上报警将是错误的。我们要避免报警错误或者至少提供信息帮助操作员区分可能错误的警报和明显的瞬间干扰。

(2) 统计总体。为了向操作员提供更精准的信息用以区分瞬间干扰和错误警报，我们在称为总体的统计测试集合上进行实验，探测 TDM 应用中的重要数据流[9]。下面是单个数据流总体的

一个采样，由6个非参数的测试组成。这6个测试是：哈姆佩尔界限、基于第5和第95个分位点的界限、基于对数转换的5%截尾均值的界限、对数转换中的3－Sigma界限、基于5%截尾均值的界限和3－Sigma界限。

我们计算每小时接收到的流文件数目和在这一小时中接收到的数据总量在这6项测试中的值。界限值在长度为3个月的数据窗口中计算，结果通过一个简单的图形化配电盘直观地显示出来。

哈姆佩尔界限和截尾均值测试抗干扰能力强，不会受异常值的影响。这是一个重要的特征，因为就像上面提到的，异常点出现在数据流的正常运行中，这时正确发出警报是很重要的。选择基于对数转换的两项测试(测试3和测试4)是因为它们对数据流行为真正发生大的变化很敏锐。对于操作员来说这是危险信号，一旦发出信号就需要立即得到响应。3－Sigma测试的优势在于TDM应用的工程师和操作员对其非常熟悉。3－Sigma测试向应用工程师提供与其他测试比较的已知基准。

(3) 配电盘用户界面。用于样例数据流的配电盘用户界面如图6所示，它基于一周内收集到的真实数据，且这些数据在这周内出现了数据流中数据丢失和毁坏的重大问题。

图中的各行表示6项统计测试的结果。测试中每次计算出一个界外值时，在配电盘上就会出现一个点。X轴表示这周的时间，以小时计算。从图中可以看出，这周早期，大约在第10小时，第5到第95分位点测试的值超出界限。当迅速恢复正常后，大约从第40个小时开始一直在界限之外。大约同一时间，其他一些测试也开始处于界限之外。直到第48个小时，6项测试中有5项都有规律地超出了界限，表明这是操作员需要慎重对待和立即做出反应的条件。

(4) 监测代理。在中继站上部署最好的代理以实现这个警戒系统。这些代理就是一种工具，可以实现我们用来判定警报条件的统计总体和图形测试。这些代理还能将它们的结果传回图形配电盘所在的远程工作站。因为这些代理必须在做重要工作的节点上(也就是中继节点)运行，因

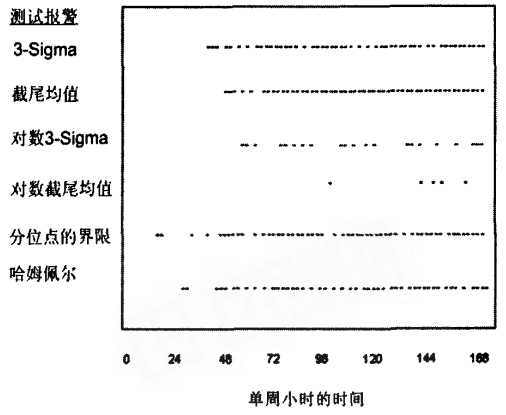

图6 图2和图3中数据显示的配电盘用户界面

此要求这些代理在计算上是轻量级的。例如，占用某节点一部分百分比的CPU去执行探测测试可能没问题，但是一个代理占用20%的CPU就不可接受了。轻量级的要求限制了运行在代理上的统计测试种类。另外，测试的输出必须容易理解，并能让非统计学家接受，从而又进一步限制了测试的选择。在多半情况下，我们为测试总体选择了简单的非参数测试。在文献[9]中有对统计总体的更充分讨论。

3.2 理解系统状态

(1)生命周期里的事件。在TDM应用中的数据复杂度和数据范围使我们难以从整体上理解系统的状态。希望有一些简单的方法告知系统的一些健康状况，以了解数据是否因为系统的某部分工作异常而面临被丢失的危险。通过前面的讨论，总体架构显然是面向文件的，常见的事件与每个文件相关联。例如，所有文件是从源系统传输过来，接收时检查文件，在应用中使用，最后在即时分析不再需要时将文件存档。这些常见的事件就像一个生命周期。通过对有特定事件概要的文件的计数，从整体上描述系统的状态。属于一个特定事件概要的计数提供了数据流系统全部状态的信息。

(2)事件概要。作为实验，将对与一个单数据流关联的一系列文件的概要进行分析。这里我们提到的4个事件是传输失败、再压缩、磁带复制和应用消费。应注意到在现实中有更多的事件发生。首先重要的是，一个文件可以有多个应用

消费。应用之间的相互影响有时能从这些序列中推断出来，从而帮助我们设计一个更有效的过程。

表 1 给出了对于事件频率和在事件的特定频率下呈现的文件数量的符号表示法。

表 1　文件生命周期中的多个事件

	事件频率顺序	文件频率
磁带上	0.010	52388
复制	0.020	16222
复制消耗	0.021	7556
磁带再压缩	0.110	1848
....	....	....
失败再压缩复制消耗	97.121	1

事件频率序列就是在文件的生命周期中每个事件的次数。例如，第一个数据行表示有 52388 个文件传输没有失败，没有再压缩，所有文件一次性写入磁带，并且没有经过分析，也没有在某个应用中被消费。这是一种数据简化的形式，用事件频率统计来代表事件序列。虽然有信息的损失(事件的精确序列和事件间隙大小丢失了)，但是频率统计帮助我们以一种简单的方式，根据事件序列来比较文件。也可以利用点处理方法进行更复杂的分析。实际的表格更大，在表 1 中我们跳过了许多行(大约 90%)，从而可能在表中包含某个奇怪的例子。表 1 中最后一行是经历了 97 次传输失败的文件实例。但这样的文件并不多。进一步检查这个文件的历史发现，这个文件在文件传输软件中出现了故障。但是，软件厂商很快修复了这个故障。

3.3　利用计划自动采取行动

(1) 借用人工智能。监测和警戒在数据流中丢失和毁坏的数据只是其中一个方面。在目前的技术发展水平下，向操作员呈现异常情况后希望操作员采取合适的操作方法加以解决。衡量管理大规模数据的能力的最重要因素可能就是有效地自动控制。我们希望能够在低端和中端任务中实现自动化，在更高层次上，是否涉及人工操作是个开放的选项。

人工智能是试图提高我们对智力本质理解的一门学科，从某种意义上讲，它利用能够对周围环境做出响应的工程自主主体将人类智力区分开来，比如著名的图灵测试[10]。目前，已有多种建立自主主体的方法，一些方法在发展更利于理解的通用智力上有所限制，因此这些方法被摒弃了。然而其中一些方法具有重要的现实意义。按规则编程就是一个例子。按规则编程是基于人工智能团体开发的技术，用于商务处理自动化中[11]。另外，自适应系统中的一些工作[12]开始出现在先进的实验性的系统设计中[13]。也就是说，在系统设计中使用人工智能团体开发的那些已被摒弃的技术，这是有先例的。

(2) 规划系统。自动规划和人工智能规划系统是一项没有得到充分利用的自动化技术。自动化技术在数据流管理中可能变得很重要。Weld[14]描述了规划技术的近期进展，包括基于约束编程和可满足性算法的快速规划方法。然而，在我们看来，即使有些欠缺，但经典的规划方式对许多应用来说足够了。因为我们关心的主要是应用而不是通用智能，能开发和使用一个适用于特定领域的规划方法就完全可以接受了。因此，相对陈旧的方式可能足以奏效。这些陈旧的方式至今还没有经过尝试，也就意味着将错失良机。

按照 Nilsson[15]的说法，一个计划就是运行在状态转换系统上的行为序列。执行这个行为序列之后，系统才能进入所希望得到的状态。每个行为包含执行这次行为先决条件的简单说明和执行结果。一般地，用一种逻辑形式来代表系统状态、行为前提和结果。经典规划模式以大量的假设为基础[16]。其中一个重要的假设是系统没有内在动力。在最简单的层面上，这意味着状态转换系统改变状态的唯一途径是由规划器采取行动。明显地，对于 TDM 应用管理的数据流并不是这样的，但是我们可以这样认为，仍然能够在重要方面实现自动化，支持 TDM 应用。随后我们将再次讨论这个问题。

(3) 文件存档。与数据流管理相关的例子就是文件存档。我们已经讨论过，在应用程序运行时使用的数据是有自然生命周期的。这些数据经过所有应用程序的处理后，进入最后一个阶段，也就是存档。在 TDM 应用中，在磁带机械手的帮

助下，把文件存到磁带上。这个机械手有6个磁带驱动器（或者称为数据传输单元DTEs）和大量的存储槽（或者称为存储单元SEs）。在任何给定的时间里，这些存储槽包含空白磁带和已经写过的磁带。软件命令被用来在驱动器和存储槽之间转移磁带（装入和卸载磁带），并在特定的驱动器中将数据写入磁带。利用这个磁带机械手，将一批文件存入磁带所需的步骤包括：将一个空白磁带从存储槽装入一个空的驱动器；由要存档的文档集构成磁带片段；在磁带里写入磁头以区分磁带的内容；在文件贮存的主系统上创建片段；将片段拷贝到与磁带驱动器连接的主机上；将片段写入磁带；当磁带写满，查证磁带是否被正确写入；卸载驱动器内所有写满的磁带，然后装入合适的存储槽中。

这8个步骤中的每一项都能由一个规划操作完成。于是规划就是在状态空间搜索中应用这些操作，直到原始文件存入磁带中为止。然而，我们本身关心的不是规划器，因为我们总是可以放低要求，设置一个特定领域的规划器。我们关心的是计划执行器，一个执行计划的控制系统。

Nilsson还描述了一种计划表示的简明数据结构，叫做三角形表。三角形表是一个简单计划执行系统的核心。我们将通过一个包括3个操作的简单问题的三角形表来说明这一点。图7显示了问题的初始状态和目标状态。我们有两个操作执行器，一个将磁带从驱动器卸载到存储槽中，一个将存储槽中的磁带转入驱动器中。我们希望通过卸载和装入操作使得vol_0在0号驱动器中，vol_1在1号驱动器中。解决方法不是唯一的，通过图7顶部所示的3个操作就可以达到目标。

上述解决方法的三角形表如表2所示。当然，我们只关心表的对角线下方。列出代表初始状态和组成计划的操作序列。沿着对角线是操作的名称或者类型。考虑代表第一个操作的项，即卸载磁带卷。该项左边一行是初始状态，与该操作的先决条件相对应。换句话说，1号存储单元为空（用Empty se1表示），0号数据传输单元（0号磁带驱动器）包含一个磁带卷（记为Full dte0 vol1）。表示卸载磁带卷操作的项下方的项包含该操作的结果，接下来的操作将依赖这些结果。换句话说，卸载磁带卷操作的一些结果是后续操作的前提。如果在检查第二个操作的项时，第一次装入卷操作，我们可以看到它有两个前提：一是由初始状态满足（在初始状态下的第一列），二是满足第一个操作的结果（出现在卸载卷下方的列中）。最后一行包含目标状态的单元。如果某个操作的结果是目标状态的单元，那么该单元就出现在最后一行。

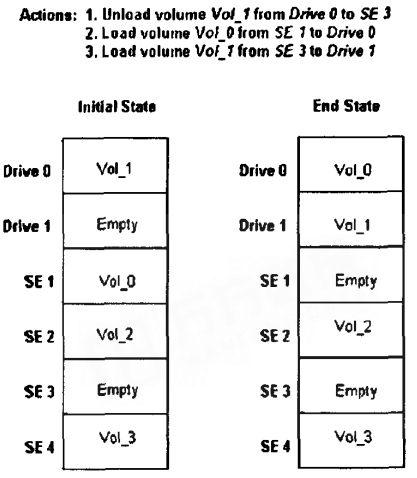

图7 三项行动计划显示磁带机器人的初始与结束状态实例

三角形表由计划的操作序列和初始及目标状态的描述经过简单计算而得。在执行计划时，计划位置在哪？三角形表提供了一种准确的方式来确定执行计划的位置。数据挖掘和数据流相关的任务经常要求从开始到完成的小时数，比如考虑在节点之间传送8万个文件需要多长时间。回答这样的问题，并给出计划执行时的状态报告，估计完成的时间，三角形能够提供有用的依据。另外，三角形表包含关于各个操作之间依赖关系的关键信息。例如，两个装入卷操作是相互独立的，因为这两个操作对应的行和列的交叉项为空。另一方面，这两个操作都依赖于卸载卷操作，因为它们的部分前提出现在卸载卷对应的列中。只要卸载卷操作完成，理论上说，两个装入卷操作就可以并行。如果磁带机械手是单线程，就不能并行了，因为机械手一次只能执行一条命令。并行之所以不能实现是因为系统的限制。因此，三角形表包含必要的重要信息，但要实现操

表 2　三项行动计划的三角形表实例

Start State	Action 1	Action 2	Action 3	
Empty se1 Full dte0 vol1	unload Volume			
Empty dte1	Full se1 vol1	Load Volume		
Full se2 vol0	Empty dte0		Load Volume	
		Full dte1 vol1	Full dte0 vol0	Goal State

作能并行的计划，依靠执行系统还不够。

下面简要讨论怎样使用三角形表来减轻计划没有考虑系统可能产生内在动力的问题。例如，一个用户在计划正在执行时对磁带机械手发出一条命令，要求改变它的状态。或者当在生成计划时机械手的状态是可以变化的，这样当计划开始时，实际的状态已经和生成计划时依据的初始状态不同了。处理这种情况的一种方法是使用最高匹配核心(HMK)的概念。三角形表的第 i 个核心是第 i 列下方及其左边的所有项，包括第 i 行的所有项。将系统的当前状态描述和三角形表相对应，确定在特定时刻哪些项是正确的，哪些项是不正确的。HMK 是系统当前状态中那些所有正确的项中标号最高的核心。

例如，在表 2 三角形表中第 2 个核心有 4 个单元：Empty drive_1、Full se_1 vol_1、Full se_2 vol_0，和 Empty drive_0。如果这 4 个单元在当前状态中都是正确的，第 2 个核心就是一个匹配核心。如果没有更高标号的匹配核心，那么第 2 个核心就是最高匹配核心。

这些项正是那些目前正确且一直保持到计划完成的项，因为计划中的后续操作依赖于它们。根据定义，下一个更高的核心有一些不正确的项，所以必须做一些操作来使它们变成正确的。因此，HMK 要确定计划的下一步合理操作。因为三角形表的核心有重叠，给出状态描述和三角形表，一个简单有效的算法可以找到 HMK。

通过使用 HMK 选择下一步操作，并在每一次操作之后根据传感器的值更新我们的状态描述，我们可以循环运行计划执行器，从而处理某些类型的内在动力。当状态转换为系统的内在动力并使得状态发生变化时，由传感器在当前状态描述中反映。接着，HMK 算法选择一个合适的操作，而不管以前的操作是什么。这可能简单地引导我们回到计划中某个先前的操作。如果没有找到 HMK，必须重新计划，把当前状态描述作为新计划的初始状态。否则，执行被选中的操作。这样进行下去一直到达目标状态。

这样一个系统的真正成功依赖于以下一些因素：由内在动力引起变化的频率、计划产生的速度(在计划生成过程中发生变化的次数)、操作失败的概率和其他一些因素。采用上述方式，对文件存档和文件传输进行了小规模的实验，效果不错。

4　结　论

在这篇文章中，我们提出了实用数据挖掘工程中管理和挖掘不断变化的大规模信息流研究中的一个重要方面。TDM 应用能让我们更清晰地了解实际环境中管理数据的挑战。在实际环境中，数据必须按量从起点传送到终点，且要求没有数据丢失或者保持最小的丢失量。在这种环境下，数据挖掘任务提出了与传统数据挖掘情况完全不同的挑战。数据挖掘中很多正在进行的工作关注的是分析，而没有讨论那些在实际环境中看起来更有争议的方面。“mise en place”是个恰当的比喻。在我们了解到的所有数据挖掘项目中，大部分工作发生在分析之前的准备阶段。

未来工作的一个方面是研究不断变化的数据流以及如何解析和解释这样的数据流。数据挖掘系统本身是对数据变化敏感的下游系统[17]，由一批运行系统采用，并向企业提供这些数据。新产品的引进，已有数据的重新包装或者域的增加和改变都是数据变化的例子。

目前，架构的变化可以通过文件频率和大小的变化来检测。将这些变化和不易于检测到的变化相结合并告知用户，需要利用传统的知识工程技术与专家进行频繁的漫长的交互。Hendler 依靠许多新兴网络标准，比如 W3C 的 OWL，对网络的利用，或者所谓的语义网络获取知识进行精彩

描述。我们计划通过主动挖掘企业网，发现应用中的变化和与数据流相关的数据，从而增强目前开发的变化检测方法。

参考文献

[1] Witten, I. H., E. Frank. Data Mining: Practical Machine Learning Tools (2nd Ed.) [M]. San Francisco: Morgan Kaufmann, 2005.

[2] Piatetsky-Shapiro, Gregory, Ronald Brachman, Tom Khabaza, Willi Kloesgen, Evangelos Simoudis. An Overview of Issues in Developing Industrial Data Mining and Knowledge Discovery Applications [C]. //E. Simoudis, J. Han, and U. Fayyad (Eds.). Proceedings of the Conference on Knowledge Discovery and Data Mining. Menlo Park: AAAI Press, 1996.

[3] Gray, J.. KDD (2003) - Invited Talk, On-Line Science: The Worldwide Telescope as a Prototype for the New Computational Science [C]. //Conference on Knowledge Discovery and Data Mining. Washington, DC, 2003.

[4] Pipino, L., Lee, Y., Wang, R.. Data Quality Assessment [J]. Communications of the ACM, 2002, 45(4): 211-218.

[5] Huang, K. Y., Y. W. Lee, R Y. Wang. Quality Information and Knowledge Management [M]. New York: Prentice-Hall, 1999.

[6] Pyle, D. Data Preparation for Data Mining[M]. San Fransisco: Morgan-Kaufmann, 1999.

[7] Hume, A. G., E. S. Daniels. Ningaui: A Linux Cluster for Business [C]. //Proceedings of the FREENIX Track: 2002 USENIX Annual Technical Conference, Monterey, California, USA, 2002: June 10-15: 195-206.

[8] Davies, D. R., R. Parasuraman. The Psychology of Vigilance [M]. London: Academic Press, 1982.

[9] Wright, J. R, D. Majumder, T. Dasu, G. T. Vesonder. Statistical Ensembles for Managing Complex Data Streams [C]. // K. Matawie (Ed.) Proceedings of the 20th International Workshop on Statistical Modeling. Sydney, 2005: 231-245.

[10] Saygin, A. P., Cicekli, I., Akman, V. Turing Test: 50 Years Later[J]. Minds and Machines, 2000, 10(4): 463-518.

[11] Minsky, Steve. Business Rules Management: New Business Tools for Innovation and Accountability [M]. Business Process Trends, 2004: 7.

[12] Barto, Andrew G., Richard S. Sutton. Reinforcement Learning: An Introduction to Adaptive Computation and Machine Learning[M]. Boston: The MIT Press, 1998.

[13] Joshi, K. R., M. Hiltunen, R. Schlichting, W. H. Sanders, A. Agbaria. Online Model-based Adaptation for Optimizing Performance and Dependability [C]. //Proceedings of the Workshop on Self-Managed Systems. WOSS 2004, October 31-November 1. Newport Beach, CA, 2004.

[14] Weld, D. S. Recent Advances in AI Planning [J]. AI Magazine, 1999, 20(2): 93-123.

[15] Nilsson, Nils J. Principles of Artificial Intelligence[M]. Palo Alto: Tioga Publishing, 1980.

[16] Ghallab, Malik, Dana Nau, Paolo Traverso. Automated Planning: Theory and Practice [M]. San Franciso: Morgan-Kaufmann, 2004.

[17] Evans, E. Domain-Driven Design: Tackling Complexity in the Heart of Software [M]. New York: Addison-Wesley, 2004.