计算机科学1998 Vol. 25 №. 4

数据库中的时态数据发掘研究*)

Research on Temporal Data Mining in Databases

欧阳为民[1,2]　　蔡庆生[2]

(安徽大学计算中心　合肥230039)[1]　　(中国科学技术大学计算机系　合肥230027)[2]

摘　要　In this paper, we discuss the stages involved in discovery knowledge from dadabases, and put forward several temporal semantics in real world databases and problems faced in mining temporal knowledge.

关键词　Knowledge discovery, Dada mining, Temporal semantics

在实践中，信息系统往往需要处理时态数据。时态数据的出现使得有必要研制在其结构中集成了时间模型的数据库系统。在信息系统中可以发现多种时态数据，例如超市的交易记录有时间标记，病员情况每隔一段时间记录一次，另外还有一些象某人的生日这种类型的其它时态数据。作为处理时态信息的时态数据库已经得到广泛研究[2]。本文试图从KDD角度讨论时态数据发掘的有关问题。首先简要描述数据发掘及其在信息系统中的若干应用，然后描述一种数据发掘系统模型，接着讨论信息系统中的时态语义问题和时态数据发掘的一些可能应用。最后，我们指出在时态数据发掘研究中的若干重要问题。

1　数据发掘及其应用

数据发掘，也称KDD，是指从数据库中提取隐含的先前未知的非平凡的具有潜在应用价值的信息或模式，如知识规则、约束、规律等等[1]。因为KDD是设法在数据库中发现知识，所以数据库中的数据结构、高性能查询、数据操纵和事务处理工具对KDD都有必要加以考虑。由于现实世界数据库本身所固有的特性，加之数据库中的数据并非专门为数据发掘而收集的，KDD工具必须能够处理大规模的海量数据、含噪音和不完备信息。

理想情况是，KDD系统是一个自治的学习Agent，自动地搜索有用的和令人感兴趣的信息，并以适当的形式报告其发现结果。完全自治的目标也许是做不到的，因为究竟是什么令人感兴趣最终得由用户，而不是计算机来决定。事实上，大多数KDD系统或多或少都要在一定程度依靠用户的参与。

数据发掘可以应用于数据库管理系统(DBMS)和更为广泛的信息系统之中。在DBMS中的主要应用包括语义查询优化、完整性约束和不一致性检测。在更广泛的信息系统中，数据发掘可以用作决策支持和分析的工具，可以用于构造知识库，从而可用于专家系统的建造。下面，我们简要讨论这些应用。

(1)决策支持。KDD系统所发现的知识可以作为决策制定的重要依据。决策支持常常会涉及到在用户引导下以某种可解释的方式进行即席数据分析。通过在海量数据中发现知识，KDD工具可为用户提供其先前不知道甚或根本就知道不了的信息。通过允许用户指定待测验的数据和期望发现的规则类型，这种帮助将会更加便利。因此KDD工具可以作为发现可能假设的技术，一旦发现某个令人感兴趣的假设，就可以用较传统的方法去测试验证。

(2)语义查询优化。关于数据库内容的语义知识对查询优化是很有价值的。某一满足数据库当前状态的规则可用于查询变换，以提高性能，即语义查询优化。例如，如果已知计算机系的全部讲师都在教学1楼，那么要求列出在教学2楼的计算机系的讲师这样一个查询就可以通过用于语义查询优化的规则来回答，而不必在数据库中进行代价高昂的搜索。操作的执行次序也是影响查询应答效率的一个重要因素，已知的规则因而也可用于重构查询操作。用于语义查询优化的规则必须与数据库的当前状态一致，这就要求系统开发利用约束时间较长，又要对当前数据状态有效的规则。然而，问题是数据库的更新是经常性的，由于增加新的数据而失效的规则或者被推广，或者被删除，因而规则集的适当逆化过程应该

*)本课题得到国家自然科学基金及安徽省教委科研基金资助。欧阳为民　副教授，在职博士生，主要研究方向：KDD、机器学习，人工智能及其应用。蔡庆生　教授，博士生导师，主要研究方向：机器学习，知识发现，人工智能。

是便利的。因为数据库是动态的，所以刻画数据的规则也可能会发生变化，从而要求领域专家动态地指定这些规则。数据发掘技术可以通过重建用于语义查询优化的规则来克服这一问题。

(3)完整性约束。数据发掘工具可以在数据库中进行完整性约束和不一致性检测。如果对数据库的更新本身是错误的，或者表达了先前未知但却有效的例子，那么就会导致与当前数据不一致。描述数据库中数据的规则可用于检测这种与当前数据不一致的更新。不论是何种情况，执行该事务的用户可作一标记，以指示数据的例外特性。同样，为了检测例外数据，规则也可用于评价数据库中的各部分数据。用于完整性约束的规则既可以是领域专家指定的，也可以是数据发掘工具发现的。显而易见，自动推导完整性约束规则在没有领域专家可用的领域是非常有用的。自动技术还可用于周期性地更新规则集，以反映随时间而变化的数据。

(4)知识库构造。通过数据发掘工具所发现的知识可用于知识库的构造，从而可用于专家系统的建造。从数据库的数据中直接提取知识有几个特别诱人的优势。首先是当没有领域专家可用时，规则可从实例数据中直接导出；其次，即使是有领域专家，数据发掘工具也可在数据库中对领域专家所提出的规则进行验证；第三，数据发掘工具可先从数据库中生成规则，然后由领域专家来进行验证。自动导出规则有利于避免规则定义过程中的人为错误和低效以及主观性。在经过选择的样本集中推导规则在机器学习领域是常见的。数据发掘所追求的是直接从数据库中导出规则，而无需提供专门的训练数据。数据发掘工具对知识库的维护和更新也是有用的，数据库中数据的变化可能会使知识库的规则不再合法。

2 一种数据发掘系统模型

首先，我们描述一种数据发掘系统模型。在实践中，有的系统可能并不具备该模型的全部成分，以该模型作为分析框架有助于突出现有 KDD 系统之间的差异和特点。该模型反映了数据发掘的过程，但却独立于具体的系统结构，我们着眼于数据发掘系统应该做什么，而不是怎样去做。

知识发现可以视为从存在于数据库中的全部规则的空间中选取令人感兴趣的规则之多阶段过程，因而，又是一个不断约简几乎无限大的初始规则空间，直到降至较小的规则子空间的过程。知识库与用户在规则空间约简的每一步都可相互作用。有的数据过滤过程在某些 KDD 系统中也可能并不存在，所以过滤过程应允许规则空间不加简约地通过。每个过滤阶段都可能包含零个或多个过滤程序，这可由用户或系统指定。最终的规则集可被吸收进系统的知识库中，用户可直接与数据库、知识库和已发现的规则集交互。下面我们分别叙述数据发掘过程的几个可能的过滤阶段。。

(1)数据过滤。知识发现的第一步就是选择合适的目标数据集。用户可以通过使用知识模板或数据选择与可视化工具来引导该过程。系统因此可将学习聚焦在与发现目标相关的数据上，并筛选掉不必要的数据。对于连续值属性，一般还要进行离散化处理。另外，由于所涉及的数据卷非常大，考虑到抽样是一种数据初步处理的有效方法，KDD 工具可能会对数据集进行抽样处理。如果需要，在抽样基础上所发现的结果可以在整个数据集上进行验证测试。该阶段的输出是用于数据测试的数据子集和约简了的规则空间。

(2)模式过滤。是知识发现的第二阶段。KDD 系统在该阶段借助于模板或其它类型选择工具来定义待发现规则的类型。这些工具通常是以适当的用户界面向用户提供可用的规则类型和属性值等形式协助用户构造模式。实际上，大多数系统仅能学习有限几个不同类型的规则，所以规则类型可进一步优化以符合系统的限制。我们可以在规则的前件和后件中指示肯定会出现的属性值，或可能包含的最大合取项个数。例如，我们考察下列关于待发现规则的说明：Find all association rules with butter in the consequent。这时，待发现的规则被确定为关联规则，并进一步限制为规则的后件或右边含有属性 butter。模式过滤是通过去除不满足指定模式的规则而约简可能的规则空间。

(3)统计过滤。规则空间在数据发掘的第三阶段是根据统计方法加以进一步过滤。尽管从数据库中发现的很多规则也许满足用户指定的模式，但其中相当一部分规则在统计意义上也许并不重要。因此，统计过滤阶段的目标是删除那些在统计意义上不重要的规则。用户可通过设置适当的统计参数或选择适当的技术来参与该阶段。例如，用户可如下说明待发现的规则：

Find all classification rules having an accuracy of at leaset 85%。

此处，用户指定了统计参数精度的最低阈值。传统的统计技术是评价规则的基础，我们还应该为 KDD 研制专门的相应技术。

(4)语义过滤。是数据发掘过程的最后一个阶段，即设法删除在语义上没有意义的规则。KDD 系统常常会生成大量规则，可是其中相当一部分可能从语义上看也许没有什么意义或不令人感兴趣，甚

至是冗余的。例如，我们有规则：任何男性从不怀孕。该规则尽管满足所要求的模式，并且具有很强的统计支持，但它实在是人人皆知的常识，因而对决策没有特别的价值。不过，也应该注意到这样的规则在语义查询优化或完整性约束中可能是有用的。例如，假定发现某输入数据表明某个男性怀孕了，那么就应该标记该数据，并认为出现了错误。语义过滤主要是用于象决策支持和知识库构造这些强调语义的领域。

通常，用户能够从由数据发掘的四个阶段所生成的规则集中选择令人感兴趣的规则，然后以适当的形式或方法表达用户选中的规则，形成最后的规则集，并将其加入系统的知识库。通过利用某些启发式信息，语义过滤也自动进行，以修剪冗余规则。但是，语义过滤的自动执行要求系统有能力明确定义究竟是什么使得规则令人感兴趣，而这是数据发掘研究中的一个富有挑战性的课题，目前尚未取得实质性进展。

3 时态语义数据

现有时态推理系统的主要不足是其适用范围较窄，大多数系统仅能处理某一种类型的时态数据，仅能学习或发现某一种类型的时态知识。这样的工具难以应用到时态数据类型更为广泛的数据发掘中。在现实世界数据库中存在多种类型的时态数据，这些类型还将因不同的具体应用而进一步多样化。描述现实世界数据库中数据的时态语义对决策支持特别有用。下面，我们指出现实世界数据库中常见的几种典型时态数据。

(1)快照数据(Snapshot Data)。数据的当前值，其中可能包括用户定义的时态值，如日期字段等等。

(2)交易数据(Transaction Data)。具有时间标记的交易记录的集合，其中所有交易都是独立的，如超市中销售交易数据。

(3)交易数据序列(Serial Transaction Data)。具有时间标记的交易记录的集合，其中一个或多个交易与某个特别的实体相关，如超市中的顾客交易序列，病员的医疗检查数据序列。

(4)时间序列数据(Time Series Data)。在某段时间内连续记录的某属性值序列，例如气象、水文数据。

(5)时间片数据(Time Slice Data)。表达在某时间点的模型化实体状态的一个或多个数据集(slice)。这种数据机制在片与片之间也可能是不同的，例如在15年期间的3个时间点所获得的民意调查数据。

(6)时间立方数据(Time Cubic Data)。在数据库或模型化实体历史中的任意给定时间点上的若干数据属性值的实况表达，即既有数据输入数据库的时间，又有其在现实世界中相应的时间值，例如时态数据库。

4 数据库中的时态知识

我们认为可以从具有时态语义的数据库中导出如下几种主要类型的时态知识。

(1)因果关系。在数据库中发现因果关系是时态知识发现中得到最广泛研究的课题之一。某事件引发另一事件的事实要求第一个事件应在第二个事件之前进行或开始。因此，在数据集中探测因果关系是时态数据发掘的固有问题。因果关系的发现在医学领域特别有用。研究人员经常要探寻是什么导致了某一疾病或病员状况。值得注意的是，因果关系要求有被考察是正确的实在证据，因而数据发掘工具主要是起最初因果关系规则的探测作用，所发现的规则随后还要进行充分的测试和检验。

(2)时态关联。某事件在时间上紧随另一事件发生且并不一定就意味着这两者之间存在某种因果关系，但作为一种关联关系却仍然是合法的。在某些情况下，作为第三个先行事件的后继，某两事件可能会自然地同时发生。时态关联还可用于探测行为模式，如在一天特定时间的购物行为关联。

(3)模式、周期和趋势。在随时间变化的单个属性值中探测模式、周期和趋势对企事业单位的决策者可能是有益的。例如一年中不同月的销售周期对于商场进货将是有指导意义的。在连续时间序列的模式探测方面已经做了很多重要工作，而离散时间序列(如交易序列数据和时间立方数据)方面的工作却甚为少见。

(4)时态约束。我们不仅要考虑数据的时态语义，还应考虑规则的时态语义问题。在目前，事实上，规则都是假定永远有效的。在这种情况下，没有任何东西表明规则何时变得合法，又何时被认为非法。同样，目前已知非法的规则也没有说明它在过去或将来是否合法。在现实中，附加上某种时态约束的规则将可以得到更好的描述，也会更有价值。这将允许为了分析目的而显示当前非法但却可能是令人感兴趣的规则，还将允许规则中存在一些空入口项，这些空入口项一直到将来某特定时间才被例示。某些规则可能在一年的前11个月有效，而过12月份的某个时间或到了财政年结束时就无效了。在现实生活中往往存在或希望带有时态约束的规则，发现这种规则的数据发掘技术将是十分有前途的。

5 时态知识发现研究中存在的问题

在时态知识发现的研究中，为适应时态语义的

特殊性，除了应考虑普通数据发掘要解决的问题外，还应考虑如下几个重要问题。

(1)时态关系的表示逻辑。为了对时间进行推理，必然需要描述时态实体之间关系的某种方法。Allen 在文中[2]中提出一种表示时间区间之间关系的分类方法。Freksa 对其作了推广，提出了基于半区间的关系分类方法，所谓半区间是指仅已知一个终点的区间[3]。这一弱化使得在根据不完备知识进行推理方面有了较大的灵活性。另外，近邻函数的采用有利于进行不精确知识推理。度量时态实体的标记不必是绝对的日历时间，时间实体可以相对于另一时间实体进行描述，而不用任何外部的参照点或度量标准。

(2)多种时间模型。在各种情况下，时间可能是连续的、周期的和非线性的，每一种都要求有不同的处理方法来进行知识发现。大多数系统对与特别应用相关的特殊时间模型都有相应的有效解决方法。但目前还没有通用的工具或方法。可以想象，通过对该问题的进一步深入研究，很有可能产生能够处理更复杂时间语义的新方法和新工具。

(3)时间的接近性。某事件或时间区间也许确实出现在另一事件或时间区间之前，但这种关系的重要性却似应取决于两者之间在时间上的是否接近，太远了一般是没有意义的。例如，文革动乱确实发生在1997年利率下调之前，但将这两者关联在一起实在毫无意义，然而，将前两年事件如低通货膨胀率与之相关联却可能是有意义的。因此，事件或时间区间之间的关联只有当它们的发生时间比较接近时才可能有意义。不过，有的类型关联在相关联的事件或时间区间之间要求有较长的时间间隔或延时。确实，某一事件的结果可能必须等相当长的一段时间才会发生，这一事实使该问题更加复杂化了。幸运的是，在数据库中发现时态知识有一定的方便之处，如模式的各成分有一定的顺序，并且常常带有时态标记。Berndt 和 Cliffor 提出动态时间弯曲(dynamic time warping)技术来进行模式与数据的匹配[4]。该技术是沿时间轴对模式进行伸缩变换，以使模式与数据匹配。此时，模式中各成分在时间上的连续性就变得比其发生的实际时间还要重要。该技术可能会为模式搜索中的时间接近性问题提供一种通用方法，因为模式中的各成分/特性的顺序是已知的，即使其具体发生的时间未知。当搜索算法对其所搜索模式相关的时间标度有严格的定义时，模式中事件间的时间接近的程度就会是固定的。例如，Wadt 在文[5]中提出了探测处方药品误用的自动方法，该方法有多条规则用于探测存在于病员情况与处方药品之间的时态关系。药品误用模式是领域专家特别定义的，时间实体之间可接受的时间上的接近程度是固定的。

(4)时间区间的推广。数据推广是许多数据发掘技术的核心。很多现有技术都是利用推广作为发现描述数据的高层概念的途径，典型的如 Han 教授提出的面向属性的归约方法[6]。也有的系统不仅依靠推广，而且还利用多层抽象概念级来进行知识发现[7]。对点值的推广一般可借助于概念层次。连续的和离散的属性都可如此推广。对非标量数据类型，概念层次关系通常可根据领域知识加以构造；而标量数据则可以自动推广，方法是将近邻的值和范围归并为较高层的概念。然而，与基于点的数据不一样，时间区间是由两个终点构成的，因而不是那么容易推广。孤立地推广两个终点，而考虑区间本身，显然不等价于实际的时间区间推广，该问题应该认真研究。

结束语 本文首先讨论了从数据库中发现有用的知识所可能涉及的几个阶段。接着，我们指出了在现实世界数据库的数据中存在的几种时态语义形式，最后提出了从普通数据发掘发展到时态数据发掘所面临的问题。我们认为时态数据发掘研究，第一，应设法扩展现有的数据发掘技术以适应时态语义；第二，再研究专门的时态数据发掘技术；第三，应研究知识发现中的维护问题，以便适应随时间而变化的环境与用户需求。对时态数据发掘而言，是机遇与挑战并存。

致谢 国际 KDD 研究知名学者加拿大 Simon Fraster 大学 Han Jiawei 教授为笔者提供了相关的资料，特此深表感谢。

参 考 文 献

[1]G. Piatetsky-Shapiro, Discovery, analysis, and presentation of strong rules, In Knowledge Discovery in Databases, AAAI/MIT Press, 1991

[2]J. F. Allen, Maintaining knowledg about temporal intervals, CACM, 26(11), 1993

[3]C. Freksa, Temporal reasoning based on semi-intervals, Artificial Intelligence, 54, 1992

[4]D. J. Berndt, J. Clifford, Using dynamic time warping to find pattern in time series, In KDD-94: AAAI Workshop on Knowledge Discovery in Databases, Seattle, Washington, July 1994

[5]T. D. Wade et al., Finding temporal patterns--A set based approach, Artificial Intelligence in Medicine, 6, 1994

[6]Y. Cai et al., Attribute-oriented induction in relational databasea, Same to[1]

[7]J. Han and Y. Fu, Discovery of multiple-level association rules from large databases. In Proc. of the 21th Int. Conf. of VLDB, 1995