数据库 知识发现 KDD 机器学习 (2)

计算机科学 1997Vol. 21 No. 6

KDD:数据库中的知识发现

KDD:Knowledge Discovery in Database

5-9 TP311.13

朱廷劭 高 文

(先进人机通信技术联合实验室 北京 100080)

摘 要 KDD(Knowledge Discovery in Database)is a new research field which comes into being with the development of Database and Artificial Intelligence.It extracts useful information from large Database. In this article, we will introduce the basic concept and process of KDD and the differences among KDD, Machine Learning and Database report. At last, we will give some conclusion of KDD.

关键词 Knowledge Discovery in Database, Data Mining, Database, Machine Learning

一 引言

随着数据库技术的不断发展及数据库管理系统的广泛应用,数据库中存储的数据量急剧增大。但目前数据库系统所能做到的只是对数据库中已有的数据进行存取,人们通过这些数据所获得的信息量仅仅是整个数据库所包含的信息量的一部分,因为目前用于对这些数据进行分析处理的工具却很少,又有局限性。然而,隐藏在这些数据之后的更重要的信息是关于这些数据的整体特征的描述及对其发展趋势的预测,这些信息在决策生成的过程中具有重要的参考价值。

在数据库技术飞速发展的同时,人工智能领域的一个分支——机器学习的研究也取得很大进展。自50年代开始机器学习的研究以来,在不同时期的研究途径和目的也不尽相同,一般大致可分为三个阶段,其研究内容分别为:神经模型和决策理论、概念符号获取及知识加强和论域专用学习。根据人类学习的不同模式人们提出了很多机器学习方法,如:实例学习、观察和发现学习、神经网络和遗传算法等等。其中某些常用且较成熟的算法已被人们运用于实际的应用系统及智能计算机的设计和实现中。

正是由于数据库技术和机器学习技术的发展,也是为了满足人们实际工作中的需要,数据库中的知识发现(KDD)技术逐渐发展起来。KDD也有人称之为数据挖掘(Data Mining),实际两者是有区别的,但一般可以不加区别地使用,在本文我们统一以KDD称之。

二 KDD定义

从开始到现在,人们给KDD下过很多定义。随

够通过ChinaNet与Internet相连。这使我们拥有一个良好的Java环境。同时,由于国际上基于Java的应用尚处于试验阶段,在Internet上有大量的有关Java的免费软件可以下载。这一方面能使我们了解到国外的最新动态,对我们开展对Java的应用研究有借鉴作用;另一方面也使我们能有效地利用别人的成果,既少走弯路,又能在一个较高的水平上开始对Java进行研究和应用。

当然,开展对Java的研究和应用,应该注意的一个问题是,切忌各行其是,大家都在低水平上重复。应该有规划、有目标、有要求;要有组织、有分工、有协作,发挥集团作战的优势,共同把Java的应用和研究搞上去,为我国的网络应用的推广,走出一条捷径。

参考资料

[1] "Moving to JDK 1.1: Using the delegation event model to create custom AWT components", Merlin Hughes, Java World, 1997-05

[2] "Graphic Java: Mastering the AWT", David M. Gergy, Alan L. McClellan, Sunsoft Press, 1997

[3] "JDK1.1 Documentation", Doug Kramer, Sun Microsystems, Inc.

[4] "Web Site Programming with Java", David Harms, et al., McGraw-Hill, 1996

[5] "Java Beans White Paper", Sun Press, 1996

着KDD研究的不断深入，人们对KDD的理解越来越全面，对KDD的定义也不断修改，下面是对KDD比较公认的一个定义：

KDD是从大量数据中提取出可信的、新颖的、有效的并能被人理解的模式的处理过程，这种处理过程是非繁琐的过程。

下面我们对这个定义作详细的解释：

数据：是指一个有关事实F的集合(如学生档案数据库中有关学生基本情况的各条记录)，它是用来描述事物有关方面的信息，一般来说这些数据都是准确无误的。

模式：对于集合F中的数据，我们可以用语言L来描述其中数据的特性。表达式E∈L，E所描述的数据是集合F的一个子集F_E。只有当表达式E比列举所有F_E中元素的描述方法更为简单时，我们才可称之为模式。如："如果成绩在81-90之间，则成绩优良"可称为一个模式，而"如果成绩为81、82、83、84、85、86、87、88、89或90，则成绩优良"就不能称之为一个模式。

处理过程：KDD是一个多步骤的处理过程，包括数据预处理、模式提取、知识评估及过程优化。我们说这个过程是非繁琐的，主要是指这个处理过程的大部分阶段是系统自动进行的而无需人工干涉。

可信：通过KDD从当前数据所发现的模式必须有一定的正确程度，否则KDD就毫无作用。可以通过新增数据来检验模式的正确性，我们用c表示模式E的可信度，c=C(E,F)，其中E∈L，E所描述的数据集合F_E⊂F。

新颖：经过KDD提取出的模式必须是新颖的，至少对系统来说应该如此。模式是否新颖可以通过两个途径来衡量，其一是得到的数据，通过当前得到的数据和以前的数据或期望得到的数据之间的比较来判断该模式的新颖程度；其二是通过其内部所包含的知识，通过对比发现的模式与已有的模式的关系来判断。通常我们可以用一个函数来表示模式的新颖程度N(E,F)，该函数的返回值是逻辑值或是对模式E的新颖程度的一个判断数值。

潜在作用：提取出的模式应该是有意义的，这可以通过某些函数的值来衡量。用u表示模式E的有作用程度，u=U(E,F)。

可被人理解：KDD的一个目标就是将数据库中隐含的模式以容易被人理解的形式表现出来，从而帮助人们更好地了解数据库中所包含的信息。当然一个模式是否容易被人理解，这本身就很难衡量，比较常用的方法是对其简单程度进行衡量。我们假定模式E的简单度(可理解度)s可用函数S(E,F)来衡量。

上面介绍的各种度量函数都只是从不同角度对所发现的模式进行评价，一般为方便起见，往往采用权值来对所发现的模式进行综合评判。在某些KDD系统中，利用函数来求得模式E的权值i=I(E,F,C,N,U,S)；而在其他一些系统中，通过对求得的模式的不同排序来表现模式的权值大小。

由以上叙述，我们可以从KDD角度给知识下个定义：一个模式E，对用户设定的阈值I，如果I(E,F,C,N,U,S)>I，则模式E可称之为知识。

三　KDD的特点

由以上我们可以看出，KDD就是利用机器学习的方法从数据库中提取有价值知识的过程，是数据库技术和机器学习两个学科的交叉学科。数据库技术侧重于对数据存储处理的高效率方法的研究，而机器学习则侧重于设计新的方法从数据中提取知识。KDD利用数据库技术对数据进行前端处理，而利用机器学习方法则从处理后的数据中提取有用的知识。KDD与其他学科也有很强的联系，如统计学、数学和可视化技术等等。

既然KDD和机器学习都是从数据中提取知识，那么两者有什么区别呢？KDD是从现实世界中存在的一些具体数据中提取知识，这些数据在KDD出现之前早已存在；而机器学习所使用的数据是专门为机器学习而特别准备的数据，这些数据在现实世界中也许毫无意义。由于KDD使用的数据来自于实际的数据库，所要处理的数据量可能很大，因此KDD中的学习算法的效率和可扩充性就显得尤为重要；此外，KDD所处理的数据由于来自于现实世界，数据的完整性、一致性和正确性都很难保证，如何将这些数据加工成学习算法可以接收的数据也需要进行深入的研究；再者，KDD可以利用目前数据库技术所取得的研究成果来加快学习过程，提高学习的效率。最后，由于KDD处理的数据来自于实际的数据库，而与这些数据库数据有关的还有其他一些背景知识，这些背景知识的合理运用也会提高学习算法的效率。

在日常的数据库操作中，人们经常使用的是从数据库中抽取数据以生成一定格式的报表，那么KDD与数据库报表工具有什么区别呢？数据库报表制作工具是将数据库中的某些数据抽取出来，经过

一些数学运算，最终以特定的格式呈现给用户，而KDD则是对数据背后隐藏的特征和趋势进行分析，最终给出关于数据的总体特征和发展趋势。报表工具也许能制作出满足下列要求的表格："上学期考试未通过及成绩优秀的学生的有关情况"，但它不能回答下述问题："考试未通过及成绩优秀的学生在某些方面有些什么不同的特征？"而KDD就可以回答这一问题。

四 KDD处理过程

KDD是一个多步骤的处理过程[1]，在处理过程中可能会有很多次的反复，主要包括以下一些处理步骤(见图1)。

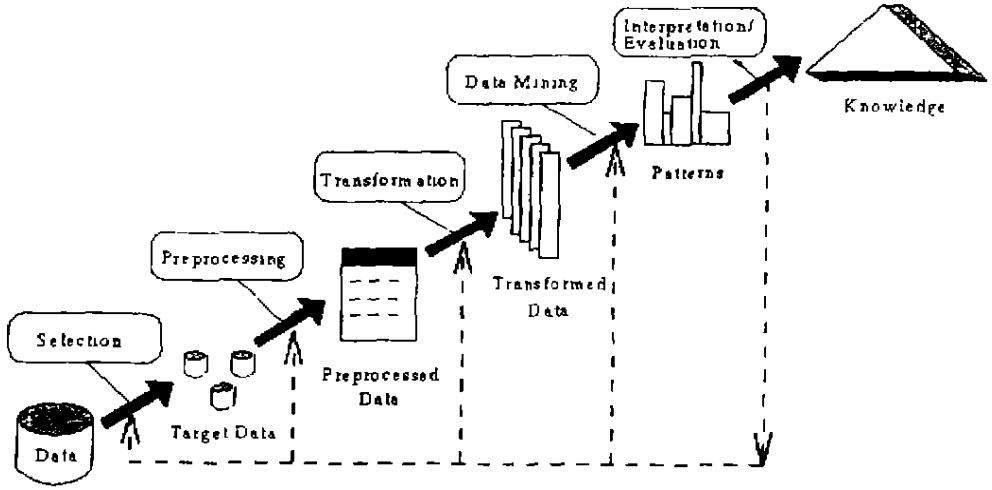


图1 KDD处理过程

1 准备。了解KDD相关领域的有关情况，熟悉有关的背景知识，并弄清楚用户的要求。

2 数据选择。根据用户的要求从数据库中提取与KDD相关的数据，KDD将主要从这些数据中进行知识提取，在此过程中，会利用一些数据库操作对数据进行处理。

3 数据预处理。主要是对阶段2产生的数据进行再加工，检查数据的完整性及数据的一致性，对其中的噪音数据进行处理，对丢失的数据可以利用统计方法进行填补。

4 数据缩减。对经过预处理的数据，根据知识发现的任务对数据进行再处理，主要通过投影或数据库中的其他操作减少数据量。

5 确定KDD的目标。根据用户的要求，确定KDD是发现何种类型的知识，因为对KDD的不同要求会在具体的知识发现过程中采用不同的知识发现算法。

6 确定知识发现算法。根据阶段5所确定的任务，选择合适的知识发现算法，这包括选取合适的模型和参数，并使得知识发现算法与整个KDD的评判标准相一致。

7 数据挖掘(DM)。运用选定的知识发现算法，从数据中提取出用户所需要的知识，这些知识可以用一种特定的方式表示或使用一些常用的表示方式，如产生式规则等等。

8 模式解释。对发现的模式进行解释，在此过程中，为了取得更为有效的知识，可能会返回前面处理步骤中的某些步以反复提取，从而提取出更有效的知识。

9 知识评价。将发现的知识以用户能了解的方式呈现给用户。这期间也包含对知识的一致性的检查，以确信本次发现的知识不与以前发现的知识相抵触。

从上面的介绍可以看出，数据挖掘只是KDD中的一个步骤，它主要是利用某些特定的知识发现算法，在一定的运算效率的限制内，从数据中发现出有关的知识，数据挖掘是KDD中最重要的一步。因此，人们往往不加区别地使用KDD和数据挖掘。

正是由于数据挖掘在KDD中的重要作用，下面我们将介绍数据挖掘的主要目标[4]及其使用的方法[5][6]。

五 数据挖掘的目标及方法

数据挖掘主要是利用各种知识发现算法从数据库数据中发现有关的知识，根据发现的知识的不同种类，可以将数据挖掘的目标分为以下几类：

特征(Characterization)。从与学习任务相关的一组数据中提取出关于这些数据的特征式，这些特征式表达了该数据集的总体特征。例如：通过对某一种疾病的症状的特征提取，得到一组关于该疾病的症状的特征式，利用这些特征式可以识别这种疾病。

区分(Discrimination)。通过对学习数据和对比数据的处理，提取出关于学习数据的主要特征，这些特征可以将学习数据与对比数据区分开来。例如：通过对某种疾病与其他疾病的症状的比较，可以提取出该疾病相对于其他疾病的区分规则，利用这些规则就可以区分出这种疾病。

分类(Classification)。根据数据的不同特征，将其划归为不同的类，这些类是事先利用训练数据建立起来的。例如：利用当前的病例数据可以建立各种疾病的分类规则，对于新的病人，根据其症状及分类规则，可以知道疾病的种类。

关联规则(Association Rule)。是发现数据对象间的相互依赖关系，一个关联规则的形式为：

$$A1 \wedge A2 \cdots \wedge Ai \rightarrow B1 \wedge B2 \cdots \wedge Bj$$

如果 B1，B2，…，Bj 出现，那么 A1，A2，…，Ai 一定出现，这表明数据 A1，A2，…，Ai 和数据 B1，B2，…，Bj 有着某种联系。例如：在对疾病症状的研究过程中，人们也许会发现，某些症状的出现一定会伴随其他一些症状的出现，通过对这种现象的深入研究，也许会找到攻克疾病的方法。

聚类(Clustering)。根据所处理的数据的一些属性，对这批数据进行分类，这种分类是基于当前所处理的数据。经过分类以后的数据，在各类之间其相似程度很小，而在某一类内部，其数据的相似度则很大。分类结束后，每类中的数据由唯一的标志进行标识，类中数据的共同特征也被提取出来用于对该类的特征描述。例如：可以通过对一组新型疾病的聚类，形成每类疾病的特征描述，这样可以对这些疾病进行识别。

预测(Prediction)。通过对数据的分析处理，估计数据库中某些丢失数据的可能值或一个数据集中某种属性值的分布情况。一般是利用数学统计的方法，找出与所要预测的属性相关的属性并根据相似数据的分析估算属性值的分布情况。例如：根据同一单位内其他职工的工资，可以预测某一职工的可能工资。

上面我们介绍了数据挖掘的主要目标，下面我们介绍数据挖掘所使用的主要方法：

数学统计方法：使用这种方法一般是首先建立一个数学模型或统计学模型，然后根据这种模型提取出有关的知识。例如：可由训练数据建立一个 Bayesian 网，然后，根据该网的一些参数及联系权值提取出相关的知识。

机器学习方法：大多数机器学习方法是利用人类的认知模型模仿人类的学习方法从数据中提取知识，由于机器学习经过多年的研究，已取得了一些较满意的成果，因此，在 KDD 中可以利用目前已经比较成熟的机器学习方法。

面向数据库方法：随着数据库技术的发展，其中的一些数据处理方法不断完善并趋于成熟。在 KDD 中，利用现有的一些数据库技术和某些专门针对于数据库的一些启发式方法，可以提取出数据库中的一些特征知识。

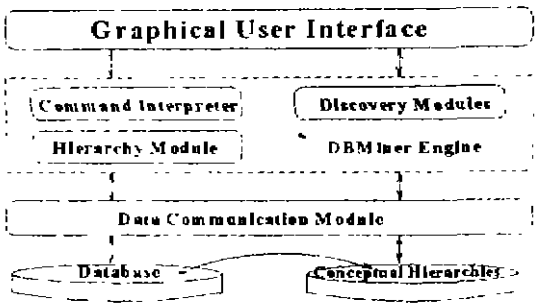
混合方法：上述各种方法各有其优缺点，为提高 KDD 的效率，可将各种方法有机地结合在一起，取长补短，以发现更有价值的知识。例如：机器学习中的推导方法可以和演绎数据结合，前者用于知识的推导，而后者可以验证发现知识的正确性。

其他方法：除了上述方法以外，还有其他一些方法，如数据可视化技术，知识表示技术等等。虽然这些方法并不普遍地应用于 KDD，但它们对数据的一些处理方法也许会对 KDD 有所启发。

六 KDD 系统简介及其 WWW 地址

目前 KDD 的研究已引起各研究机构和公司的关注，一些 KDD 的原型系统相继建立，KDD 的商用软件也已有售。下面我们简单介绍两个 KDD 系统：DBMiner 和 Quest。

DBMiner[7] 是加拿大 Simon Fraser 大学研制的一个原型系统，其结构如图 2 所示。DBMiner 主要由三个模块组成：图形用户界面、DBMiner 引擎和通信模块。图形用户界面主要完成与用户的交互；DBMiner 引擎是该系统的核心模块，所有知识发现的处理均由该模块完成；通信模块主要完成 DBMiner 与数据库服务器之间的数据传输。DBMiner 使用 DMQL(Data Mining Query Language)描述 KDD 的任务，利用 AOI(Attribute-Oriented Induction)和推广树的方法进行知识的获取。



[illegible] DBMiner [illegible]

Quest[2]是由 IBM Almaden 研究中心开发的 KDD 系统，其目标是开发各种数据挖掘方法以更好地用于决策支持。Quest 的系统结构如图 3。

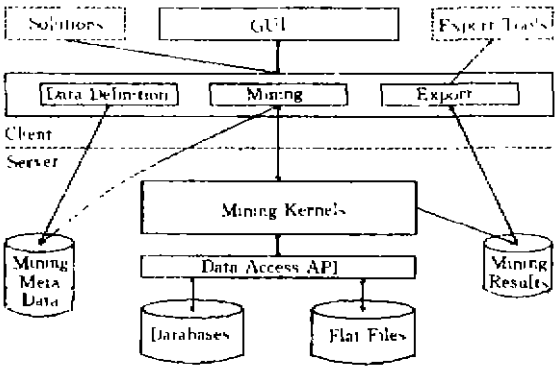


图 3

为方便在 Internet 上查找有关 KDD 的信息，下面我们列出了一些关于 KDD 的比较重要的 WWW 地址，供大家参考：

URL http://info.gte.com/~kdd

http://www.cs.bham.ac.uk/~anp/TheDataMine.html

http://www.ics.uci.edu/AI/ML/Machine-Learning.html

http://www.gmd.de/ml-archive

http://www.cosmic.uga.edu/maincat.html#45

http://www.neuronet.ph.kcl.ac.uk

http://wwwipd.ira.uka.de/~prechelt/FAQ/neural-net-faq.html

结束语　一份最近的 Gartner 报告列举了五项在今后 3 到 5 年内对工业将产生重要影响的关键技术，其中 KDD 和人工智能排名第一，同时这份报告将并行计算机体系结构研究和 KDD 列入今后 5 年内公司应该投资的 10 个新技术领域，由此可见 KDD 研究的重要性。随着数据库技术和人工智能技术的不断发展，相信不久的将来，KDD 会更好地服务于人们。

参考文献

[1] Usama M. Fayyad et al., Eds., Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press, 1996

[2] Rakesh Agrawal et al., The Quest Data Mining System, IBM Almaden Research Center

[3] Jiawei Han et al., DBLearn: A System Prototype for Knowledge Discovery in Relational Databases, School of Computing Science, Simon Fraser Univ., Canada

[4] Yongjian Fu, Discovery of Multiple-level Rules from Large Databases, Ph. D. Thesis, July 1996, Same to [3]

[5] M.-S. Chen, J. Han, and P. S. Yu, Data Mining: An Overview from Database Perspective, IEEE Trans. on Knowledge and Data Eng., 1997

[6] Jiawei Han et al., DBMiner: A System for Mining Knowledge in Large Relational Databases, Proc. 1996 Int'l Conf. on Data Mining and Knowledge Discovery (KDD'96), Portland, Oregon, Aug. 1996

[7] J. Han, Data Mining Techniques, Proc. 1996 ACM-SIGMOD Int'l Conf. on Management of Data (SIGMOD'96), Montreal, Canada, June 1996

（上接第 89 页）

结论　作为一种面向对象的程序设计语言，Java 已经吸引了众多的使用者。我们可以相信，随着 Java 自身的完善和强有力支持工具的涌现，它将成为面向对象程序设计的首选语言。

在本文中，我们阐述了用 Java 实现 OMT 设计的具体方法。文中描述的方法具有一般性，可供具体应用时参考。

参考文献

[1] James Rumbaugh et al., Object-oriented Modeling and Design, Prentice Hall, 1991

[2] Laura Lemay & Charles L. Perkins, Teach Yourself Java In 21 Days, Sams Net Publishing, 1996

[3] Tim O'Reilly, Publishing Models for Internet Commerce, CACM, 39(6), 1996

[4] James Gosling & Henry McGilton, The Java Language Environment: A White Paper, Sun Microsystem, Inc., http://java.sun.com