数 据 挖 掘

概 念 与 技 术

——第 四 章——

（加） Jiawei Han
Micheline Kamber 著

http://www.cs.sfu.ca

第四章 数据挖掘元语、语言和系统结构

- 数据挖掘元语：定义数据挖掘任务?
- 一种数据查询语言
- 根据数据查询语言设计图形用户界面
- 数据挖掘系统的结构
- 小结

为什麽要用数据挖掘原语和语言?

- 能不能自动发现数据库中的所有模式? —— 这是不现实的,因为这样的模式太多,但不一定是有兴趣的
- 数据挖掘应该是交互式的过程
  - 用户决定要挖掘的内容
- 用户必须被提供一组原语来与数据挖掘系统通信
- 把这些原语包含到数据挖掘查询语言
  - 使用户交互作用更加灵活
  - 是用户交互界面设计的基础
  - 是数据挖掘产业和实践标准化

什麽定义了一个数据挖掘任务？

- 任务相关数据
- 挖掘知识的类型
- 知识背景
- 兴趣度度量
- 发现模式的可视化

任务相关数据

- 数据库或数据仓库名
- 数据库表和数据仓库立方体
- 数据选择的条件
- 相关属性和维
- 数据分组标准

要挖掘知识的类型

- 概念描述
- 区分
- 关联
- 分类/预测
- 聚类
- 孤立点分析
- 其它的数据挖掘任务

背景知识：概念分层

- 模式分层
  - 如： **street < city < province_or_state < country**
- 集合分组分层
  - 如： **{20-39} = young, {40-59} = middle_aged**
- 操作导出的分层
  - **email 地址: login-name < department < university < country**
- 基于规则的分层
  - **low_profit_margin (X) <= price(X, P1) and cost (X, P2) and (P1 - P2) < $50**

兴趣度度量

- 简洁性

  例如： (关联) 规则长度, (判定树) 树的大小

- 确定性

  例如：置信度: **P(A|B) = n(A and B) / n (B)**,分类的可靠性 或准确性,确定性因子,规则的强度,规则的质量,区分权等

- 实用性

  潜在有用性,**e.g**,关联规则的支持度,描述的噪声阈值

- 新颖性

  不是目前所知道的,令人惊讶的(通常用来删除冗余规则）

发现模式的可视化

- 不同的背景/使用要求不同的形式和表达
  - 如：规则，表，交叉表，饼图或条图等等
- 概念分层 也是很重要的
  - 发现的知识在高概念层表示可能比原始数据概念表示更容易理解
  - 交互式上卷，下钻，转轴，切片和切块 可以帮助用户从不同视觉观察概化数据的知识
- 不同的知识要求不同的表示：关联，分类，聚类等等

第四章 数据挖掘元语、语言和系统结构

- 数据挖掘元语：定义数据挖掘任务?
- 一种数据查询语言
- 根据数据查询语言设计图形用户界面
- 数据挖掘系统的结构
- 小结

一种数据查询语言 (DMQL)

- 为什么**DMQL**很重要
  - **DMQL**能够支持特别的和交互的数据挖掘
  - 提供了一个类似于**SQL**的标准语言
    - 希望能在关系数据库上达到和**SQL**类似的效果
    - 是关系数据库系统发展和演变的基础
    - 促进了信息交换和技术转换,推动了关系数据库技术的商品化和被普遍接受
- 设计
  - 早期**DMQL**是用原语描述设计的

DMQL语法

- 语法说明:
  - 任务相关数据
  - 挖掘的知识类型
  - 概念分层说明
  - 兴趣度量
  - 模式表示和可视化
- 把以上所有的语法聚集就形成了**DMQL**查询

任务相关数据说明的语法

- *use database* <database_name>, 或 *use data warehouse* <data_warehouse_name>
- *from* <*relation*(s)/cube(s)> [*where* condition]
- *in relevance to* <att_or_dim_list>
- *order by* <order_list >
- *group by* <grouping_list>
- *having* <condition>

任务相关数据说明

例**4.9** 本例展示如何用**DMQL**说明例**4.1**描述的任务相关的数据.例**4.1**是挖掘由加拿大顾客在**AllElectronics**经常购买的商品之间的关连规则,涉及顾客的**income**和**age.**此外,用户指出他想将数据按日期分组.这些数据由关系数据库检索.

use database AllElectronics_db

in relevance to I.name,I.price,C.income,C.age

from custom C,item I,purchase P,item_sold S

where I.item_ID=S.item-ID and S.trans_ID=P.trans_ID and P.cust_ID=C. cust_ID and C.country="Canada"

group by P.date

指定挖掘知识类型的语法

- 特征化

  Mine_Knowledge_Specification ::=
  mine characteristics [_as_ pattern_name]
  analyze measure(s)

- 区分

  Mine_Knowledge_Specification ::=
  mine comparison [_as_ pattern_name]
  for target_class _where_ target_condition
  {_versus_ contrast_class_*i* _where_
  contrast_condition_*i*}
  analyze measure(s)

- 关联

  Mine_Knowledge_Specification ::=
  mine associations [as pattern_name]

指定挖掘知识类型的语法

- 分类

Mine_Knowledge_Specification ::=
mine classification **[*as* pattern_name]**
analyze
classifying_attribute_or_dimension

- 预测

Mine_Knowledge_Specification ::=
mine prediction **[*as* pattern_name]**
analyze
prediction_attribute_or_dimension
{*set* {attribute_or_dimension_*i*= value_*i*}}

概念分层说明的语法

- 使用如下语句指出用户所要用的概念分层
  use hierarchy <hierarchy> for <attribute_or_dimension>
- 使 用不同的语法定义不同的概念层
  - 模式分层
    define hierarchy time_hierarchy on date as
    [date,month quarter,year]
  - 集合分组分层
    define hierarchy age_hierarchy for age on customer as
    level1: {*young, middle_aged, senior*} < level0: all
    level2: {20, ..., 39} < level1: *young*
    level2: {40, ..., 59} < level1: *middle_aged*
    level2: {60, ..., 89} < level1: *senior*

概念分层说明的语法

- 操作导出的分层

  define hierarchy age_hierarchy for age on customer as

  {age_category(1), ..., age_category(5)} := cluster(default, age, 5) < all(age)

- 基于规则的分层

  define hierarchy profit_margin_hierarchy on item as

  level_1: low_profit_margin < level_0: all

  if (price - cost)< $50

  level_1: medium-profit_margin < level_0: all

  if ((price - cost) > $50) and ((price - cost) < = $250))

  level_1: high_profit_margin < level_0: all

  if (price - cost) > $250

兴趣度量说明的语法

- 用户可以用如下语句说明兴趣度量和它的阈值:

  with <interest_measure_name> threshold = threshold_value

- 例如:

  with support threshold = 0.05

  with confidence threshold = 0.7

模式表示和可视化说明的语法

- 我们数据挖掘语言需要一种语法, 是的用户可以用一种或多种形式显示发现的模式

  display as <result_form>

- 为方便交互式观察不同概念层,可以用以下的语法定义:

Multilevel_Manipulation ::= *roll up on*
attribute_or_dimension
| *drill down on*
attribute_or_dimension
| *add* attribute_or_dimension
| *drop*
attribute_or_dimension

汇集——一个DMQL查询的例子

use database **AllElectronics_db**

use hierarchy **location_hierarchy for B.address**

mine characteristics as **customerPurchasing**

analyze **count%**

in relevance to **C.age, I.type, I.place_made**

from **customer C, item I, purchases P, items_sold S, works_at W, branch**

where **I.item_ID = S.item_ID and S.trans_ID = P.trans_ID and P.cust_ID = C.cust_ID and P.method_paid = ``AmEx'' and P.empl_ID = W.empl_ID and W.branch_ID = B.branch_ID and B.address = ``Canada'' and I.price >= 100**

with **noise** threshold **= 0.05**

display as **table**

其他数据挖掘语言和数据挖掘语言的标准化

- 关联规则语言说明
  - **MSQL (Imielinski & Virmani'99)**
  - **MineRule (Meo Psaila and Ceri'96)**
  - 使用**Datalog**语法表示查询群 **(Tsur et al'98)**
- 数据挖掘 **OLE DB(Microsoft'2000)**
  - 基于**OLAP**的**OLE, OLE DB, OLE DB**
  - 把数据库管理系统**(DBMS),**数据仓库和数据挖掘集于一体
- **CRISP-DM (CRoss-Industry Standard Process for Data Mining)**
  - 为高效数据挖掘提供平台和处理结构
  - 强调使用数据挖掘技术解决商业问题

第四章 数据挖掘元语、语言和系统结构

- 数据挖掘元语：定义数据挖掘任务?
- 一种数据查询语言
- 根据数据查询语言设计图形用户界面
- 数据挖掘系统的结构
- 小结

根据数据挖掘查询语言设计图形用户界面

- 根据数据挖掘查询语言设计图形用户界面需要考虑哪些方面?
  - 数据收集和数据查询编辑
  - 发现模式的表示
  - 分层结构说明和操作
  - 数据挖掘原语的操作
  - 交互的多层挖掘
  - 其他各种信息

第四章 数据挖掘元语、语言和系统结构

- 数据挖掘元语：定义数据挖掘任务?
- 一种数据查询语言
- 根据数据查询语言设计图形用户界面
- 数据挖掘系统的结构
- 小结

数据挖掘系统的结构

- 将数据挖掘系统和数据库（DB)和/或数据仓库系统（DW)系统耦合
  - 不耦合——单调的文件处理,不推荐
  - 松散耦合
    - 从 DB/DW提取数据
  - 半紧密耦合——提高DM性能
    - 一些基本的数据挖掘原语可以在DB/DW系统实现,例如:排序,索引,聚集,直方图分析,多路连接和一些统计函数的预计算
  - 紧密耦合----一个统一的信息处理环境
    - DM系统被平滑的集成到DB/DM系统,数据挖掘查询根据DB/DW系统的挖掘查询分析,索引,查询处理方法等进行优化

第四章 数据挖掘元语、语言和系统结构

- 数据挖掘元语：定义数据挖掘任务?
- 一种数据查询语言
- 根据数据查询语言设计图形用户界面
- 数据挖掘系统的结构
- 小结

小 结

- 数据挖掘任务说明的五种原语
  - 任务相关数据
  - 被挖掘知识的类型
  - 知识背景
  - 兴趣度度量
  - 用来显示被发现模式的知识表示和可视化技术
- 数据挖掘查询语言
  - 例如用于数据挖掘的**DMQL, MS/OLEDB** 语言
- 数据挖掘系统结构
  - 不耦合,松耦合,半紧密耦合,紧密耦合

References

- E. Baralis and G. Psaila. Designing templates for mining association rules. Journal of Intelligent Information Systems, 9:7-32, 1997.
- Microsoft Corp., OLEDB for Data Mining, version 1.0, http://www.microsoft.com/data/oledb/dm, Aug. 2000.
- J. Han, Y. Fu, W. Wang, K. Koperski, and O. R. Zaiane, “DMQL: A Data Mining Query Language for Relational Databases”, DMKD'96, Montreal, Canada, June 1996.
- T. Imielinski and A. Virmani. MSQL: A query language for database mining. Data Mining and Knowledge Discovery, 3:373-408, 1999.
- M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A.I. Verkamo. Finding interesting rules from large sets of discovered association rules. CIKM’94, Gaithersburg, Maryland, Nov. 1994.
- R. Meo, G. Psaila, and S. Ceri. A new SQL-like operator for mining association rules. VLDB'96, pages 122-133, Bombay, India, Sept. 1996.
- A. Silberschatz and A. Tuzhilin. What makes patterns interesting in knowledge discovery systems. IEEE Trans. on Knowledge and Data Engineering, 8:970-974, Dec. 1996.
- S. Sarawagi, S. Thomas, and R. Agrawal. Integrating association rule mining with relational database systems: Alternatives and implications. SIGMOD'98, Seattle, Washington, June 1998.
- D. Tsur, J. D. Ullman, S. Abitboul, C. Clifton, R. Motwani, and S. Nestorov. Query flocks: A generalization of association-rule mining. SIGMOD'98, Seattle, Washington, June 1998.

http://www.cs.sfu.ca/~han

Thank you !!!