

数据挖掘：概念与技术

Jiawei Han and Micheline Kamber著

Monrgan Kaufmann Publishers Inc.

范明 孟小峰等译

机械工业出版社

教材-作者

- <http://www.cs.illinois.edu/homes/hanj/>
- The book will be covered in two courses at CS, UIUC: 伊利诺伊大学, 厄巴纳-尚佩恩(University of Illinois at Urbana-Champaign)
 - CS412: **Introduction to data warehousing and data mining**
Coverage (Chapters 1-7 of This Book)
 - CS512: **Data mining: Principles and algorithms**
(Chapters 8-11 of This Book)



Jiawei Han

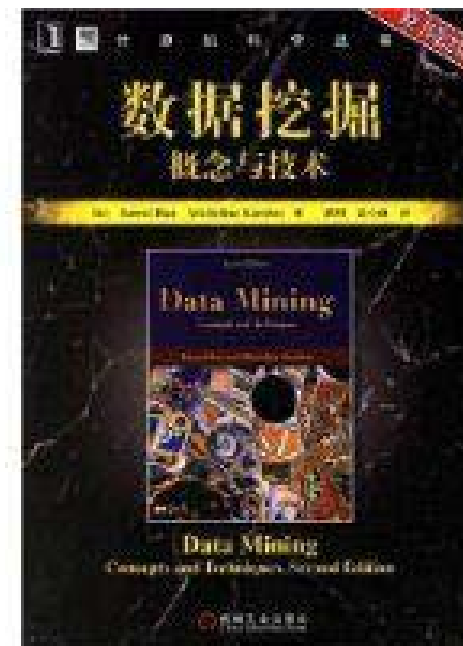
Professor, Department of Computer Science
Univ. of Illinois at Urbana-Champaign
Rm 2132, Siebel Center for Computer Science
201 N. Goodwin Avenue
Urbana, IL 61801, USA
E-mail: hanj[at]cs.uiuc.edu

Ph.D. (1985), Computer Science, Univ. Wisconsin-Madison

Data Mining and Databases

Data Mining Research Group
(Data Mining Group Summary Report: Sp
Database and Information Systems Research
(UIUC Academic Calendar)

Fax: (217) 265-6494
Web: www.cs.uiuc.edu/~hanj





课程信息

- 数据挖掘的（前7章的内容），
 - 第1章 引言
 - 第2章 数据预处理
 - 第3章 数据仓库与OLAP技术概述
 - 第4章 数据立方体计算与数据泛化
 - 第5章 挖掘频繁模式、关联和相关
 - 第6章 分类和预测
 - 第7章 聚类分析
 - 如果有时间（第11章 数据挖掘的应用和发展趋势）
- 导论课程（从数据库角度出发）
- 相关涉及：数据库系统、统计学与机器学习的概念和技术



课时安排与考核

- 课时安排

- 总学时 **48**，讲课学时 **36**，课内上机学时 **12**（课外上机学时 **20**）
起止**01-16**周

- 考核

- 平时成绩+考试成绩



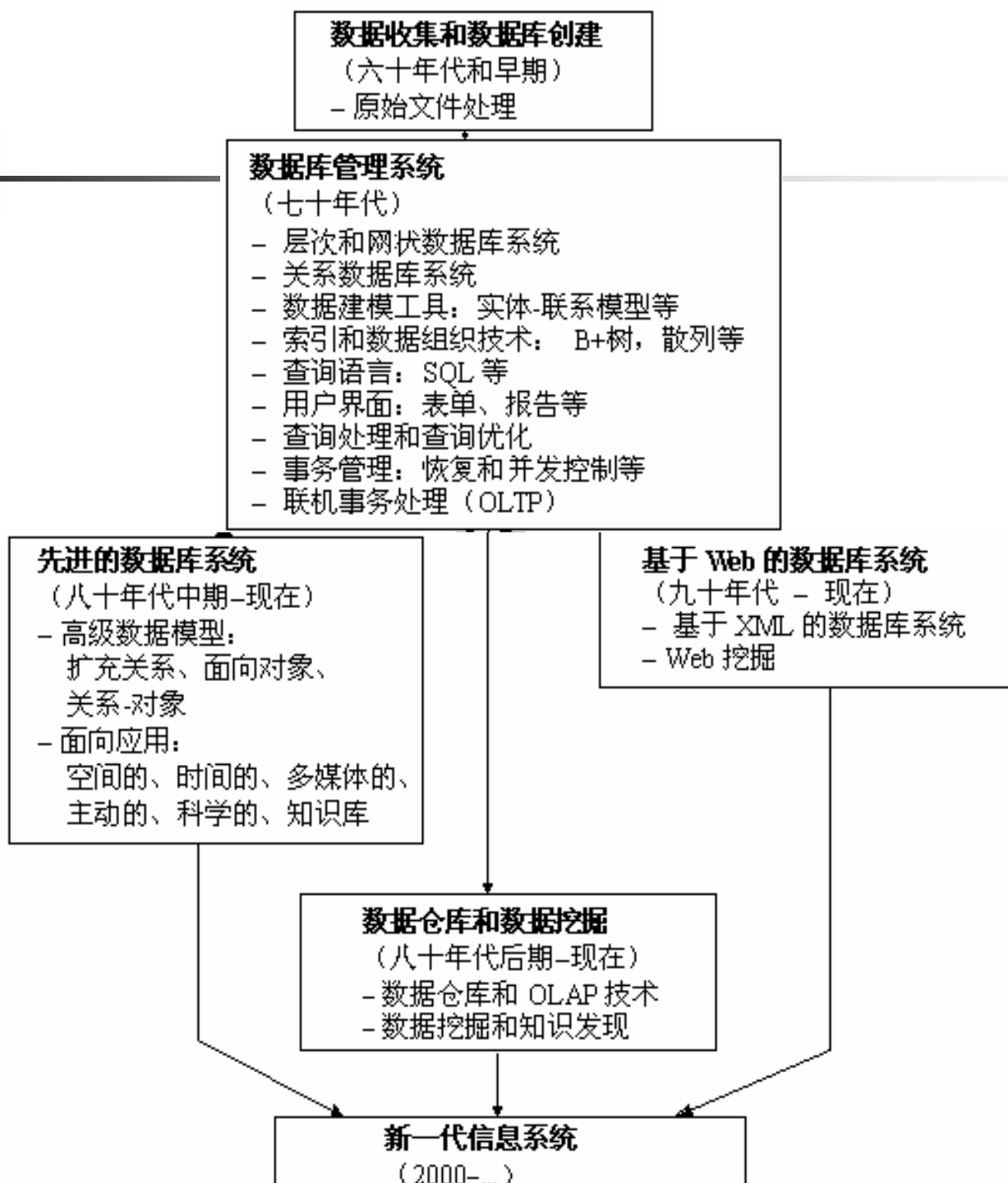
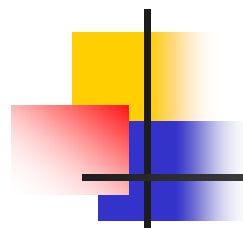
第1章 引论

- 动机：为什么要数据挖掘？
- 什么是数据挖掘？
- 数据挖掘：在什么数据上进行？
- 数据挖掘功能
- 所有的模式都是有趣的吗？
- 数据挖掘系统分类
- 数据挖掘的主要问题



数据处理技术的演进

- **1960s:**
 - 数据收集, 数据库创建, **IMS**层次和网状 **DBMS**
- **1970s:**
 - 关系数据库模型, 关系 **DBMS** 实现
- **1980s:**
 - **RDBMS**, 先进的数据模型 (扩充关系的, **OO**, 演绎的, 等.) 和面向应用的 **DBMS** (空间的, 科学的, 工程的, 等.)
- **1990s—2000s:**
 - 数据挖掘和数据仓库, 多媒体数据库, 和 **Web** 数据库



动机：需要

■ 数据爆炸问题

- 自动的数据收集工具和库, 数据仓库, 和其它信息

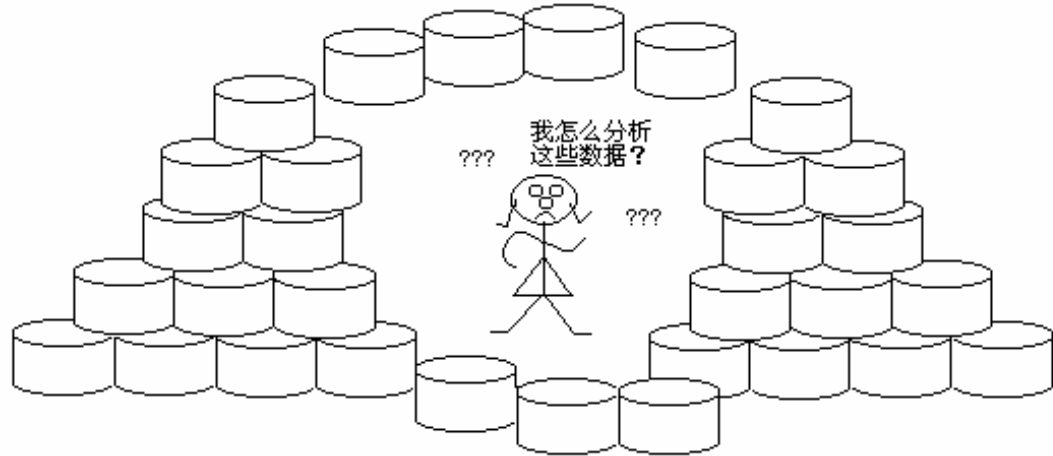
- **Business: Web, e-commerce, transactions, stocks, ...**
- **Science: Remote sensing, bioinformatics, scientific simulation, ...**
- **Society and everyone: news, digital cameras, YouTube**

■ 我们正被数据淹没, 但却缺乏知识

- 数据丰富, 但信息贫乏

■ 解决办法: 数据仓库与数据挖掘

- 数据仓库与联机分析处理(OLAP)
- 从大型数据库的数据中提取有趣的知识(规则, 规律性, 模式, 限制等)





数据挖掘界简史

- **1989 IJCAI Workshop on Knowledge Discovery in Databases (Piatetsky-Shapiro)**
 - Knowledge Discovery in Databases (G. Piatetsky-Shapiro and W. Frawley, 1991)
- **1991-1994 Workshops on Knowledge Discovery in Databases**
 - Advances in Knowledge Discovery and Data Mining (U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, 1996)
- **1995-1998 International Conferences on Knowledge Discovery in Databases and Data Mining (KDD'95-98)**
 - Journal of Data Mining and Knowledge Discovery (1997)
- **1998 ACM SIGKDD, SIGKDD'1999-2001 conferences, and SIGKDD Explorations**
- **More conferences on data mining**
 - PAKDD, PKDD, SIAM-Data Mining, (IEEE) ICDM, etc.
- **ACM Transactions on KDD starting in 2007**



Conferences and Journals on Data Mining

■ KDD Conferences

- ACM SIGKDD Int. Conf. on Knowledge Discovery in Databases and Data Mining (**KDD**)
- SIAM Data Mining Conf. (**SDM**)
- (IEEE) Int. Conf. on Data Mining (**ICDM**)
- Conf. on Principles and practices of Knowledge Discovery and Data Mining (**PKDD**)
- Pacific-Asia Conf. on Knowledge Discovery and Data Mining (**PAKDD**)

■ Other related conferences

- ACM SIGMOD
- VLDB
- (IEEE) ICDE
- WWW, SIGIR
- ICML, CVPR, NIPS

■ Journals

- Data Mining and Knowledge Discovery (DAMI or DMKD)
- IEEE Trans. On Knowledge and Data Eng. (TKDE)
- KDD Explorations
- ACM Trans. on KDD

Where 2 Find References? DBLP, CiteSeer, Google

■ Data mining and KDD (SIGKDD: CDROM)

- Conferences: ACM-SIGKDD, IEEE-ICDM, SIAM-DM, PKDD, PAKDD, etc.
- Journal: Data Mining and Knowledge Discovery, KDD Explorations, ACM TKDD

■ Database systems (SIGMOD: ACM SIGMOD Anthology—CD ROM)

- Conferences: ACM-SIGMOD, ACM-PODS, VLDB, IEEE-ICDE, EDBT, ICDT, DASFAA
- Journals: IEEE-TKDE, ACM-TODS/TOIS, JIIS, J. ACM, VLDB J., Info. Sys., etc.

■ AI & Machine Learning

- Conferences: Machine learning (ML), AAAI, IJCAI, COLT (Learning Theory), CVPR, NIPS, etc.
- Journals: Machine Learning, Artificial Intelligence, Knowledge and Information Systems, IEEE-PAMI, etc.

■ Web and IR

- Conferences: SIGIR, WWW, CIKM, etc.
- Journals: WWW: Internet and Web Information Systems,

■ Statistics

- Conferences: Joint Stat. Meeting, etc.
- Journals: Annals of statistics, etc.

■ Visualization

- Conference proceedings: CHI, ACM-SIGGraph, etc.
- Journals: IEEE Trans. visualization and computer graphics, etc.

什么是数据挖掘?



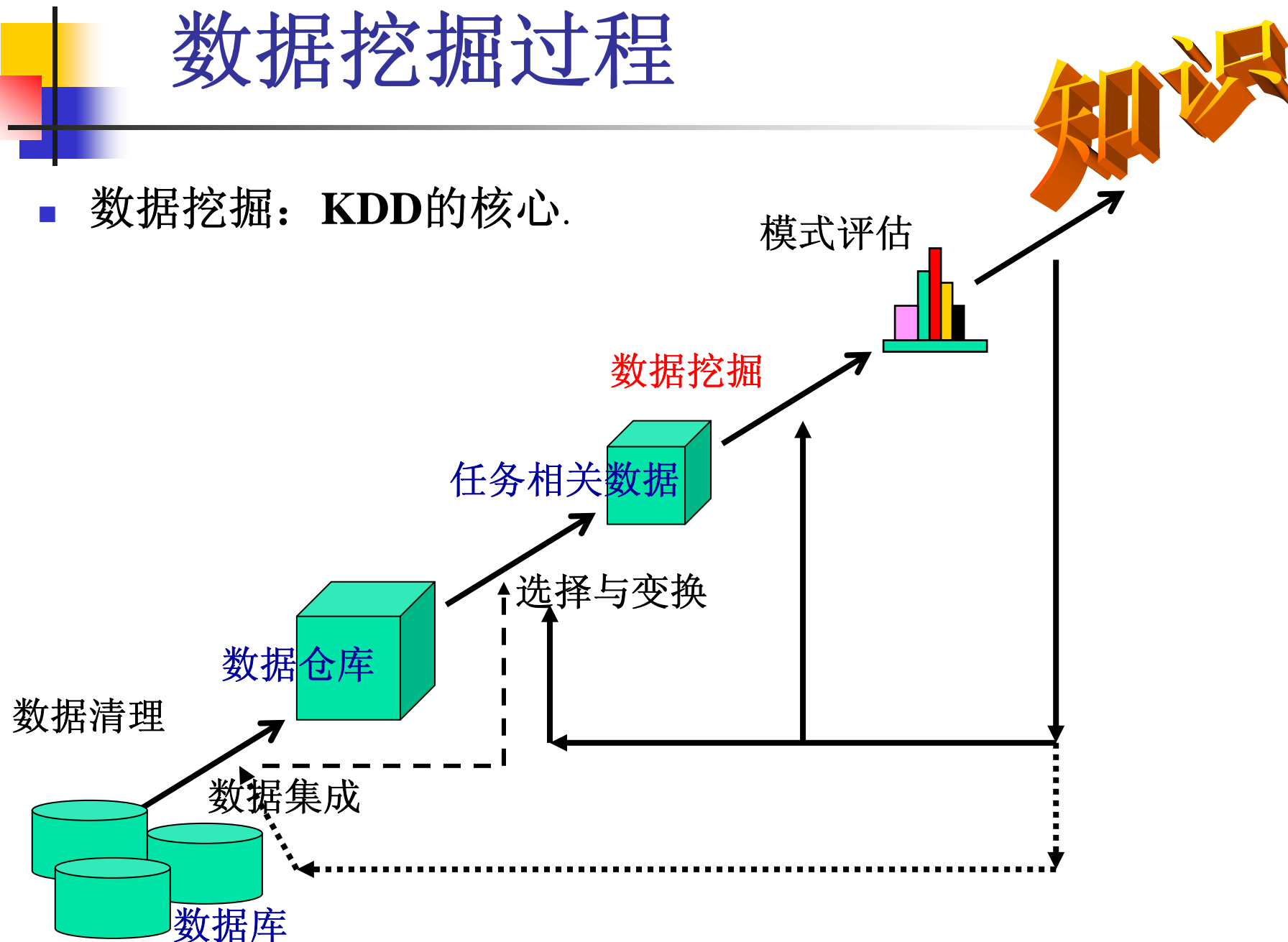
- 数据挖掘 (从数据中挖掘知识):
 - 从大型数据库中提取有趣的 (非平凡的, 蕴涵的, 先前未知的 并且是 潜在有用的) 信息或模式
 - 数据挖掘: 用词不当?
- 其它叫法和“**inside stories**”内幕新闻 :
 - 数据库中知识发现(挖掘) (**Knowledge discovery in databases, KDD**), 知识提取(knowledge extraction), 数据/模式分析(data/pattern analysis), 数据考古(**data archeology**), 数据捕捞(**data dredging**), 信息收获 (**information harvesting**), 商务智能(**business intelligence**), 等.
- 什么不是数据挖掘?
 - (演绎) 查询处理.
 - 专家系统 或小型 机器学习(ML)/统计程序
 - 处理大量数据/ 有效的可伸缩的技术

Why Not Traditional Data Analysis?

- 巨大的的数据 Tremendous amount of data
 - Algorithms must be highly scalable to handle such as tera-bytes of data
- High-dimensionality of data
 - Micro-array may have tens of thousands of dimensions
- High complexity of data
 - Data streams and sensor data
 - Time-series data, temporal data, sequence data
 - Structure data, graphs, social networks and multi-linked data
 - Heterogeneous databases and legacy(遗产) databases
 - Spatial, spatiotemporal, multimedia, text and Web data
 - Software programs, scientific simulations
- New and sophisticated applications

数据挖掘过程

- 数据挖掘：KDD的核心。

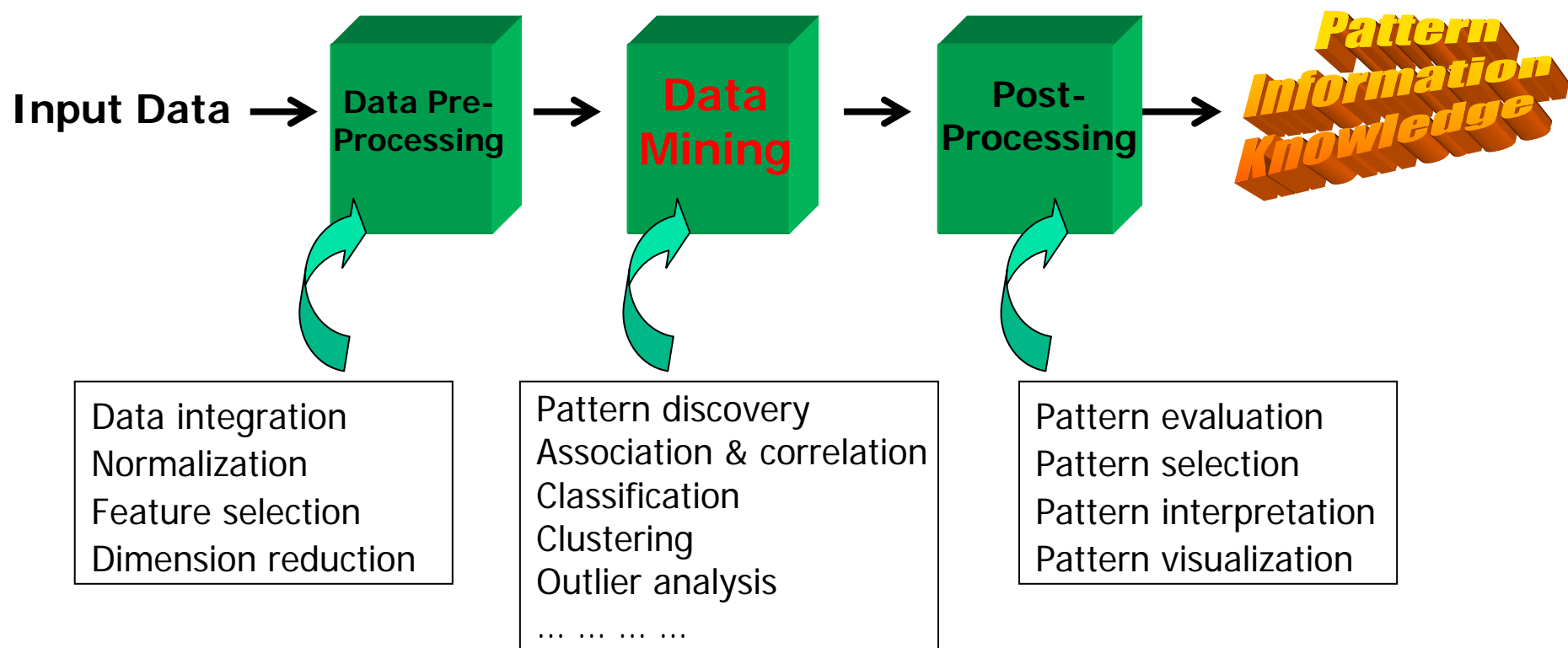




KDD过程的步骤

- 学习应用领域:
 - 相关的先验知识和应用的目标
- 创建目标数据集: 数据选择
- 数据清理和预处理: (可能占全部工作的 **60%!**)
- 数据归约与变换:
 - 发现有用的特征, 维/变量归约, 不变量的表示.
- 选择数据挖掘函数
 - 汇总, 分类, 回归, 关联, 聚类.
- 选择挖掘算法
- 数据挖掘: 搜索有趣的模式
- 模式评估和知识表示
 - 可视化, 变换, 删除冗余模式, 等.
- 发现知识的使用

KDD过程: 机器学习和统计的角度



- This is a view from typical machine learning and statistics communities

典型的数据挖掘系统结构

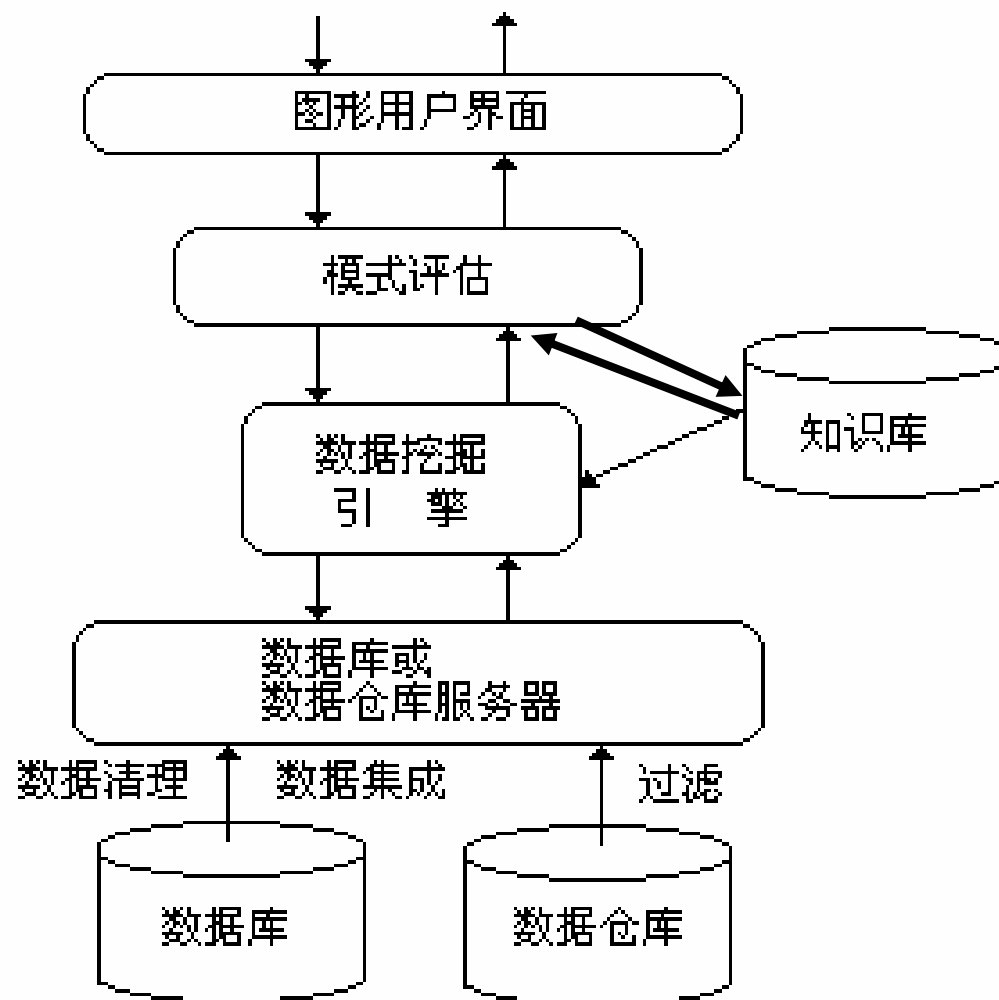


图 1.5：典型的数据挖掘系统结构



为什么要数据挖掘?—可能的应用

- 数据库分析和决策支持
 - 市场分析和管埋
 - 针对销售(**target marketing**), 顾客关系管理, 购物篮分析, 交叉销售(**cross selling**), 市场分割(**market segmentation**)
 - 风险分析与管理
 - 预测, 顾客关系, 改进保险, 质量控制, 竞争能力分析
 - 欺骗检测与管理
- 其它应用
 - 文本挖掘 (新闻组, **email**, 文档资料)
 - 流数据挖掘(**Stream data mining**)
 - **Web挖掘.**
 - 生物信息学/生物 数据分析



市场分析与管理(1)

- 用于分析的数据源在哪?
 - 信用卡交易, 会员卡, 打折优惠卷, 顾客投诉电话, (公共) 生活时尚研究
- 针对销售(**Target marketing**)
 - 找出顾客群, 他们具有相同特征: 兴趣, 收入水平, 消费习惯, 等.
- 确定顾客随时间变化的购买模式
 - 个人帐号到联合帐号的转变: 结婚, 等.
- 交叉销售分析(**Cross-market analysis**)
 - 产品销售之间的关联/相关
 - 基于关联信息的预测



市场分析与管理(2)

- 顾客分类(**Customer profiling**)
 - 数据挖掘能够告诉我们什么样的顾客买什么产品(聚类或分类)
- 识别顾客需求
 - 对不同的顾客识别最好的产品
 - 使用预测发现什么因素影响新顾客
- 提供汇总信息
 - 各种多维汇总报告
 - 统计的汇总信息 (数据的中心趋势和方差)



欺骗检测和管理(1)

- 应用
 - 广泛用于健康照料, 零售, 信用卡服务, 电讯 (电话卡欺骗), 等.
- 方法
 - 使用历史数据建立欺骗行为模型, 使用数据挖掘帮助识别类似的实例
- 例
 - 汽车保险: 检测这样的人, 他/她假造事故骗取保险赔偿
 - 洗钱: 检测可疑的金钱交易 (**US Treasury's Financial Crimes Enforcement Network**)
 - 医疗保险: 检测职业病患者, 医生和介绍人圈



欺骗检测和管理(2)

- 检测不适当的医疗处置

- 澳大利亚健康保险会(Australian Health Insurance Commission)发现许多全面的检查是请求做的,而不是实际需要的(每年节省100万澳元).

- 检测电话欺骗

- 电话呼叫模式: 通话距离, 通话时间, 每天或每周通话次数. 分析偏离期望的模式.
- 英国电讯(British Telecom)识别频繁内部通话的呼叫者的离散群, 特别是移动电话, 超过数百万美元的欺骗.

- 零售

- 分析家估计, 38%的零售业萎缩是由于不忠诚的雇员造成的.



其它应用

- 运动
 - **IBM Advanced Scout**分析NBA的统计数据 (阻挡投篮, 助攻, 和犯规) 获得了对纽约小牛队(**New York Knicks**)和迈阿密热火队(**Miami Heat**) 的竞争优势
- 天文
 - 借助于数据挖掘的帮助,**JPL** 和 **Palomar Observatory** 发现了**22** 颗类星体(**quasars**)
- **Internet Web Surf-Aid**
 - **IBM Surf-Aid** 将数据挖掘算法用于有关交易的页面的**Web**访问日志, 以发现顾客喜爱的页面, 分析**Web** 销售的效果, 改进**Web** 站点的组织, 等.
- **Web**: 页面的分类、聚类、推荐/用户的访问模式



数据挖掘:在什么数据上进行?

- 关系数据库
- 数据仓库
- 事务(交易)数据库
- 先进的数据库和信息存储
 - 面向对象和对象-关系数据库
 - 空间和时间数据
 - 时间序列数据和流数据
 - 文本数据库和多媒体数据库
 - 异种数据库和遗产数据库
 - WWW



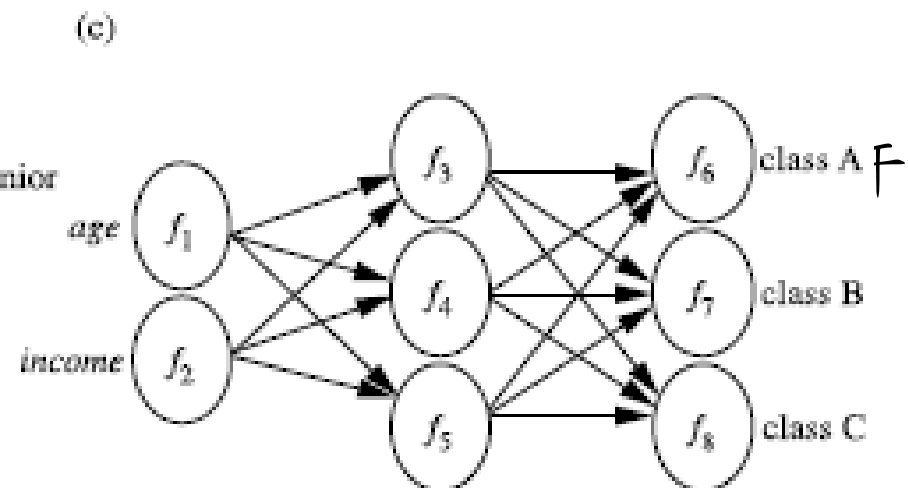
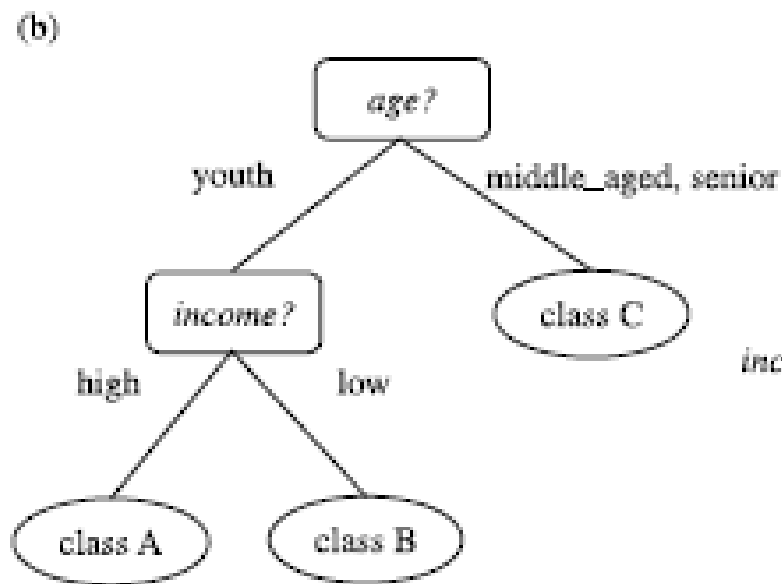
数据挖掘功能(1)

- 概念描述: 特征和区分 Characterization and discrimination
 - 概化, 汇总和比较数据特征, 例如, 干燥和潮湿的地区
- 频繁模式, 关联, 相关 **Frequent patterns, association, correlation vs. causality**
 - **频繁模式**: 数据中频繁出现的模式
 - 多维和单维关联
 - $age(X, "20..29") \wedge income(X, "20..29K") \Rightarrow buys(X, "PC")$
[support = 2%, confidence = 60%]
 - $contains(T, "computer") \Rightarrow contains(T, "software")$
[support = 1%, confidence = 75%]

数据挖掘功能(2)

- (a)
- \angle \angle age(X, "youth") AND income(X, "high") \longrightarrow class(X, "A")
 - age(X, "youth") AND income(X, "low") \longrightarrow class(X, "B")
 - age(X, "middle_aged") \longrightarrow class(X, "C")
 - age(X, "senior") \longrightarrow class(X, "C")

$\frac{1}{7}$



数据挖掘功能(3)

- 聚类分析Unsupervised learning (i.e., Class label is unknown)
 - 类标号(Class label) 未知: 对数据分组, 形成新的类. 例如, 对房屋分类, 找出分布模式
 - 聚类原则: 最大化类内的相似性, 最小化类间的相似性

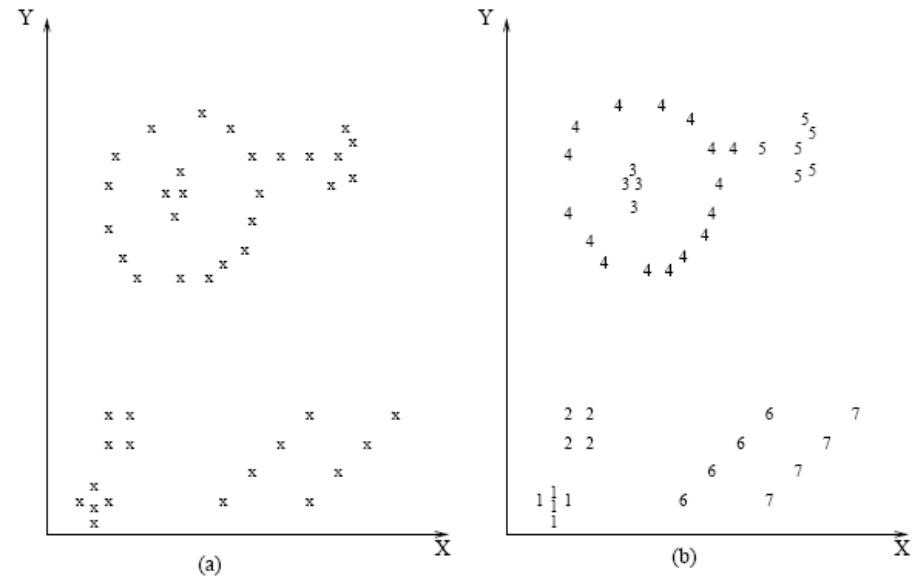
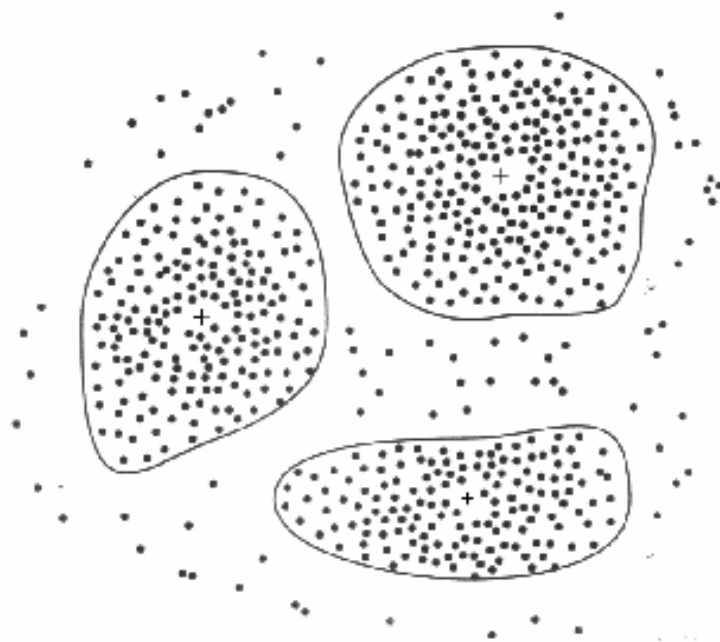


Figure 1. Data clustering.



数据挖掘功能(4)

- 孤立点(Outlier)分析

- 孤立点: 一个数据对象, 它与数据的一般行为不一致
- 孤立点可以被视为例外, 但对于欺骗检测和罕见事件分析, 它是相当有用的

- 趋势和演变分析

- 趋势和偏离: 回归分析
- 序列模式挖掘, 周期性分析
 - **e.g., first buy digital camera, then buy large SD memory cards**
- 基于相似的分析
 - **Approximate and consecutive motifs**



数据挖掘功能(5) -Structure and Network Analysis

- **Graph mining**
 - Finding frequent subgraphs (e.g., chemical compounds), trees (XML), substructures (web fragments)
- **Information network analysis**
 - Social networks: actors (objects, nodes) and relationships (edges)
 - e.g., author networks in CS, terrorist networks
 - Multiple heterogeneous networks
 - A person could be multiple information networks: friends, family, classmates, ...
 - Links carry a lot of semantic information: Link mining
- **Web mining**
 - Web is a big information network: from PageRank to Google
 - Analysis of Web information networks
 - Web community discovery, opinion mining, usage mining, ...

Top-10 Most Popular DM Algorithms:18 Identified Candidates (I)

■ Classification

- #1. C4.5: Quinlan, J. R. C4.5: Programs for Machine Learning. Morgan Kaufmann., 1993.
- #2. CART: L. Breiman, J. Friedman, R. Olshen, and C. Stone. Classification and Regression Trees. Wadsworth, 1984.
- #3. K Nearest Neighbours (kNN): Hastie, T. and Tibshirani, R. 1996. Discriminant Adaptive Nearest Neighbor Classification. TPAMI. 18(6)
- #4. Naive Bayes Hand, D.J., Yu, K., 2001. Idiot's Bayes: Not So Stupid After All? Internat. Statist. Rev. 69, 385-398.

■ Statistical Learning

- #5. SVM: Vapnik, V. N. 1995. The Nature of Statistical Learning Theory. Springer-Verlag.
- #6. EM: McLachlan, G. and Peel, D. (2000). Finite Mixture Models. J. Wiley, New York. Association Analysis
- #7. Apriori: Rakesh Agrawal and Ramakrishnan Srikant. Fast Algorithms for Mining Association Rules. In VLDB '94.
- #8. FP-Tree: Han, J., Pei, J., and Yin, Y. 2000. Mining frequent patterns without candidate generation. In SIGMOD '00.



The 18 Identified Candidates (II)

- **Link Mining**

- **#9. PageRank: Brin, S. and Page, L. 1998. The anatomy of a large-scale hypertextual Web search engine. In WWW-7, 1998.**
- **#10. HITS: Kleinberg, J. M. 1998. Authoritative sources in a hyperlinked environment. SODA, 1998.**

- **Clustering**

- **#11. K-Means: MacQueen, J. B., Some methods for classification and analysis of multivariate observations, in Proc. 5th Berkeley Symp. Mathematical Statistics and Probability, 1967.**
- **#12. BIRCH: Zhang, T., Ramakrishnan, R., and Livny, M. 1996. BIRCH: an efficient data clustering method for very large databases. In SIGMOD '96.**

- **Bagging and Boosting**

- **#13. AdaBoost: Freund, Y. and Schapire, R. E. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. 55, 1 (Aug. 1997), 119-139.**



The 18 Identified Candidates (III)

- **Sequential Patterns**
 - #14. GSP: Srikant, R. and Agrawal, R. 1996. Mining Sequential Patterns: Generalizations and Performance Improvements. 5th International Conference on Extending Database Technology, 1996.
 - #15. PrefixSpan: J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal and M-C. Hsu. PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth. In ICDE '01.
- **Integrated Mining**
 - #16. CBA: Liu, B., Hsu, W. and Ma, Y. M. Integrating classification and association rule mining. KDD-98.
- **Rough Sets**
 - #17. Finding reduct: Zdzislaw Pawlak, Rough Sets: Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers, Norwell, MA, 1992
- **Graph Mining**
 - #18. gSpan: Yan, X. and Han, J. 2002. gSpan: Graph-Based Substructure Pattern Mining. In ICDM '02.



Top-10 Algorithm Finally Selected at ICDM'06

- **#1: C4.5 (61 votes)**
- **#2: K-Means (60 votes)**
- **#3: SVM (58 votes)**
- **#4: Apriori (52 votes)**
- **#5: EM (48 votes)**
- **#6: PageRank (46 votes)**
- **#7: AdaBoost (45 votes)**
- **#7: kNN (45 votes)**
- **#7: Naive Bayes (45 votes)**
- **#10: CART (34 votes)**



挖掘出的所有模式都是有趣的吗？

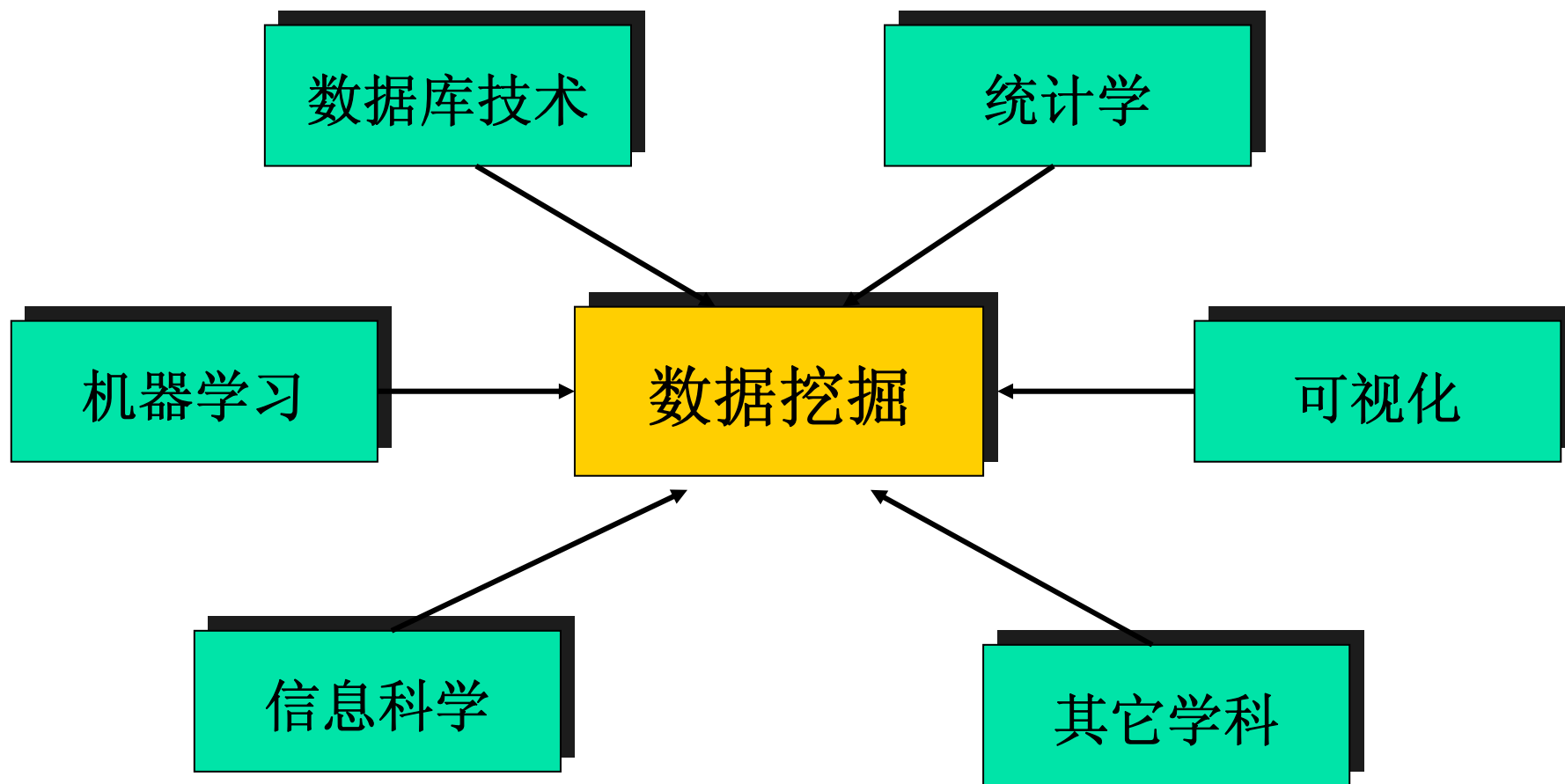
- 一个数据挖掘系统/查询可以挖掘出数以千计的模式,并非所有的模式都是有趣的
 - 建议的方法: 以人为中心, 基于查询的, 聚焦的挖掘
- 兴趣度度量: 一个模式是 **有趣的** 如果它是 易于被人理解的, 在某种程度上在新的或测试数据上是有效的, 潜在有用的, 新颖的, 或验证了用户希望证实的某种假设
- 客观与主观的兴趣度度量:
 - 客观: 基于模式的统计和结构, 例如, 支持度, 置信度, 等.
 - 主观: 基于用户对数据的确信, 例如, 出乎意料, 新颖性, 可行动性(actionability), 等.



能够只发现有趣的模式吗？

- 发现所有有趣的模式：完全性
 - 数据挖掘系统能够发现所有有趣的模式吗？
 - 关联 vs. 分类 vs. 聚类
- 仅搜索有趣的模式：优化
 - 数据挖掘系统能够仅发现有趣的模式吗？
 - 方法
 - 首先找出所有模式, 然后过滤掉不是有趣的那些.
 - 仅产生有趣的模式— 挖掘查询优化

数据挖掘：多学科交叉





数据挖掘分类

- 一般功能
 - 描述式数据挖掘——描述数据的一般性质
 - 预测式数据挖掘——对数据进行推断，做预测
- 不同的角度,不同的分类
 - 待挖掘的数据库类型
 - 待发现的知识类型
 - 所用的技术类型
 - 所适合的应用类型



数据挖掘分类的多维视图

- 待挖掘的数据库
 - 关系的, 事务的, 面向对象的, 对象-关系的, 主动的, 空间的, 时间序列的, 文本的, 多媒体的, 异种的, 遗产的, WWW, 等.
- 所挖掘的知识
 - 特征, 区分, 关联, 分类, 聚类, 趋势, 偏离和孤立点分析, 等.
 - 多/集成的功能, 和多层次上的挖掘
- 所用技术
 - 面向数据库的, 数据仓库 (OLAP), 机器学习, 统计学, 可视化, 神经网络, 等.
- 适合的应用
 - 零售, 电讯, 银行, 欺骗分析, DNA 挖掘, 股票市场分析, Web 挖掘, Web日志分析, 等



OLAP挖掘: 数据挖掘与数据仓库的集成

- 数据挖掘系统, **DBMS**, 数据仓库系统的耦合
 - 不耦合, 松耦合, 半紧密耦合, 紧密耦合
- 联机分析挖掘
 - 挖掘与 **OLAP** 技术的集成
- 交互挖掘多层知识
 - 通过下钻, 上卷, 转轴, 切片, 切块等操作, 在不同的抽象层挖掘知识和模式的必要性.
- 多种挖掘功能的集成
 - 特征分类, 先聚类再关联



数据挖掘查询语言

- 通过数据挖掘查询语言，数据挖掘任务可以通过查询的形式输入到数据挖掘系统中。
- 定义数据挖掘查询语言的优势
 - ①可以让用户透明地使用各种数据挖掘查询语句，而不管它们是怎样实现的；
 - ②可以把数据挖掘平滑地集成到各种应用系统上，而不像当前这样，每一个数据挖掘应用系统都得从头开发设计；
 - ③可以使数据挖掘的研究工作更有继承性，通用良好的查询语言的定义是进一步工作的基础。



数据挖掘原语

用户在进行数据挖掘时,总希望能够通过使用一组数据挖掘原语来与数据挖掘系统通信,以支持有效的和有成果的知识发现。

这组原语包括:

- ①与任务相关的数据;
- ②要挖掘的知识类型;
- ③用于挖掘过程的背景知识:概念分层;
- ④评估模式的兴趣度度量和阈值;
- ⑤可视化发现模式的期望表示。

DMQL 就是基于对这些原语的说明而设计出来的一种有效的数据挖掘查询语言。该语言采用类似于 SQL 的语法,因此它易于和关系查询语言 SQL 集成在一起。



数据挖掘的主要问题(1)

- 挖掘方法和用户交互
 - 在数据库中挖掘不同类型的知识
 - 在多个抽象层的交互式知识挖掘
 - 结合背景知识
 - 数据挖掘语言和启发式数据挖掘
 - 数据挖掘结果的表示和可视化
 - 处理噪音和不完全数据
 - 模式评估: 兴趣度问题
- 性能和可伸缩性(scalability)
 - 数据挖掘算法的性能和可伸缩性
 - 并行, 分布和增量的挖掘方法



数据挖掘的主要问题(2)

- 数据类型的多样性问题
 - 处理关系的和复杂类型的数据
 - 从异种数据库和全球信息系统 (WWW)挖掘信息
- 应用和社会效果问题
 - 发现知识的应用
 - 特定领域的数据挖掘工具
 - 智能查询回答
 - 过程控制和决策制定
 - 发现知识与已有知识的集成: 知识融合问题
 - 数据安全, 完整和私有的保护



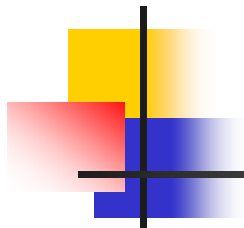
小结

- 数据挖掘: 从大量数据中发现有趣的模式
- 数据库技术的自然进化, 具有巨大需求和广泛应用
- **KDD** 过程包括数据清理, 数据集成, 数据选择, 变换, 数据挖掘, 模式评估, 和知识表示
- 挖掘可以在各种数据存储上进行
- 数据挖掘功能: 特征, 区分, 关联, 分类, 聚类, 孤立点 和趋势分析, 等.
- 数据挖掘系统的分类
- 数据挖掘的主要问题



参考文献

- **U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press, 1996.**
- **J. Han and M. Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann, 2000.**
- **T. Imielinski and H. Mannila. A database perspective on knowledge discovery. Communications of ACM, 39:58-64, 1996.**
- **G. Piatetsky-Shapiro, U. Fayyad, and P. Smith. From data mining to knowledge discovery: An overview. In U.M. Fayyad, et al. (eds.), Advances in Knowledge Discovery and Data Mining, 1-35. AAAI/MIT Press, 1996.**
- **G. Piatetsky-Shapiro and W. J. Frawley. Knowledge Discovery in Databases. AAAI/MIT Press, 1991.**



谢谢大家!





第2章：数据预处理

- 为什么预处理数据？
- 数据清理
- 数据集成
- 数据归约
- 离散化和概念分层产生
- 小结



为什么数据预处理？

- 现实世界中的数据是脏的
 - **不完全**: 缺少属性值, 缺少某些有趣的属性, 或仅包含聚集数据
 - 例, **occupation**=""
 - **噪音**: 包含错误或孤立点
 - 例, **Salary**="-10"
 - **不一致**: 编码或名字存在差异
 - 例, **Age**="42" **Birthday**="03/07/2010"
 - 例, 以前的等级 "1,2,3", 现在的等级 "A, B, C"
 - 例, 重复记录间的差异



数据为什么脏?

- 不完全数据源于
 - 数据收集时未包含
 - 数据收集和数据分析时的不同考虑.
 - 人/硬件/软件问题
- 噪音数据源于
 - 收集
 - 录入
 - 变换
- 不一致数据源于
 - 不同的数据源
 - 违反函数依赖



为什么数据预处理是重要的？

- 没有高质量的数据, 就没有高质量的数据挖掘结果!
 - 高质量的决策必然依赖高质量的数据
 - 例如, 重复或遗漏的数据可能导致不正确或误导的统计.
 - 数据仓库需要高质量数据的一致集成



数据质量：一个多维视角

- 一种广泛接受的多角度：
 - 正确性(**Accuracy**)
 - 完全性(**Completeness**)
 - 一致性(**Consistency**)
 - 合时(**Timeliness**): **timely update?**
 - 可信性(**Believability**)
 - 可解释性(**Interpretability**)
 - 可存取性(**Accessibility**)

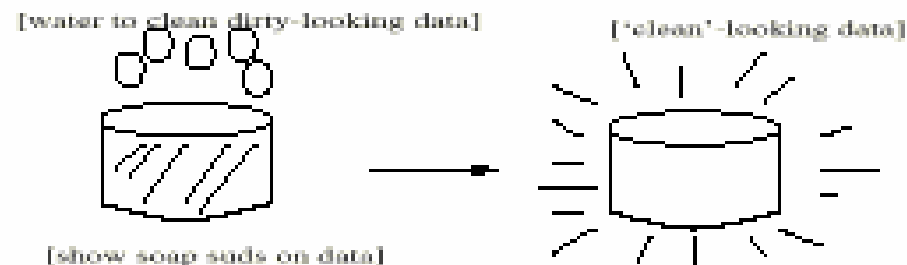


数据预处理的主要任务

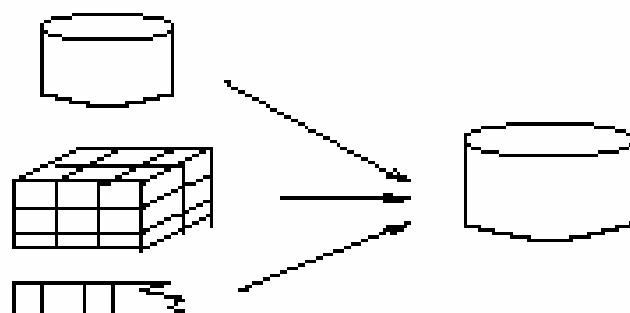
- 数据清理
 - 填充缺失值, 识别/去除离群点, 光滑噪音, 并纠正数据中的不一致
- 数据集成
 - 多个数据库, 数据立方体, 或文件的集成
- 数据变换
 - 规范化和聚集
- 数据归约
 - 得到数据的归约表示, 它小得多, 但产生相同或类似的分析结果: 维度规约、数值规约、数据压缩
- 数据离散化和概念分层

数据预处理的形式

Data Cleaning



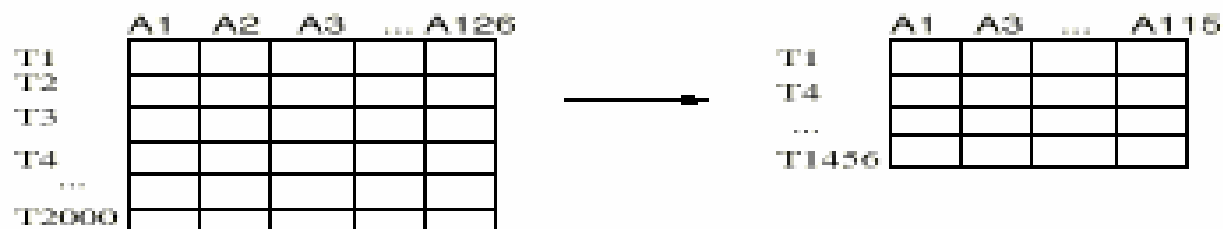
Data Integration



Data Transformation

-2, 32, 100, 59, 48 → -0.02, 0.32, 1.00, 0.59, 0.48

Data Reduction





第2章：数据预处理

- 为什么预处理数据？
- 数据清理
- 数据集成
- 数据归约
- 离散化和概念分层产生
- 小结



数据清理 Data Cleaning

- 现实世界的数据是脏：很多潜在的不正确的数据，比如，仪器故障，人为或计算机错误，许多传输错误
 - **incomplete**: 缺少属性值, 缺少某些有趣的属性, 或仅包含聚集数据
 - e.g., *职业* = “ ” (missing data)
 - **noisy**: 包含错误或孤立点
 - e.g., *Salary* = “-10” (an error)
 - **inconsistent**: 编码或名字存在差异, e.g.,
 - *Age* = “42”, *Birthday* = “03/07/2010”
 - 以前的等级 “1, 2, 3”, 现在等级 “A, B, C”
 - 重复记录间的差异
 - **有意的**(e.g., 变相丢失的数据)
 - Jan. 1 as everyone's birthday?



如何处理缺失数据？

- 忽略元组：缺少类别标签时常用(假定涉及分类—不是很有效，当每个属性的缺失百分比变化大时)
- 手工填写缺失数据：乏味+费时+不可行？
- 自动填充
 - 一个全局常量：e.g., “unknown”, a new class?!
 - 使用属性均值
 - 与目标元组同一类的所有样本的属性均值：更巧妙
 - 最可能的值：基于推理的方法，如贝叶斯公式或决策树



噪音数据Noisy Data

- **Noise:** 被测量的变量的随机误差或方差
- 不正确的属性值可能由于
 - 错误的数据收集工具
 - 数据录入问题 **data entry problems**
 - 数据传输问题 **data transmission problems**
 - 技术限制 **technology limitation**
 - 不一致的命名惯例 **inconsistency in naming convention**
- 其他需要数据清理的问题
 - 重复记录 **duplicate records**
 - 数据不完整 **incomplete data**
 - 不一致的数据 **inconsistent data**



如何处理噪音数据？

- **分箱Binning method:**
 - 排序数据，分布到等频/等宽的箱/桶中
 - 箱均值光滑、箱中位数光滑、箱边界光滑, etc.
- **聚类Clustering**
 - 检测和去除 离群点/孤立点 outliers
- **计算机和人工检查相结合**
 - 人工检查可疑值 (e.g., deal with possible outliers)
- **回归 Regression**
 - 回归函数拟合数据



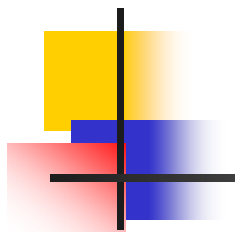
分箱：简单的离散化方法

- **等宽度Equal-width (distance) 剖分:**
 - 分成大小相等的 n 个区间: 均匀网格 **uniform grid**
 - 若 A 和 B 是属性的最低和最高取值, 区间宽度为: $W = (B - A)/N$.
 - 孤立点可能占据重要影响 **may dominate presentation**
 - 倾斜的数据处理不好.
- **等频剖分 (frequency) /等深equi-depth :**
 - 分成 n 个区间, 每一个含近似相同数目的样本
 - **Good data scaling**
 - 类别属性可能会非常棘手.

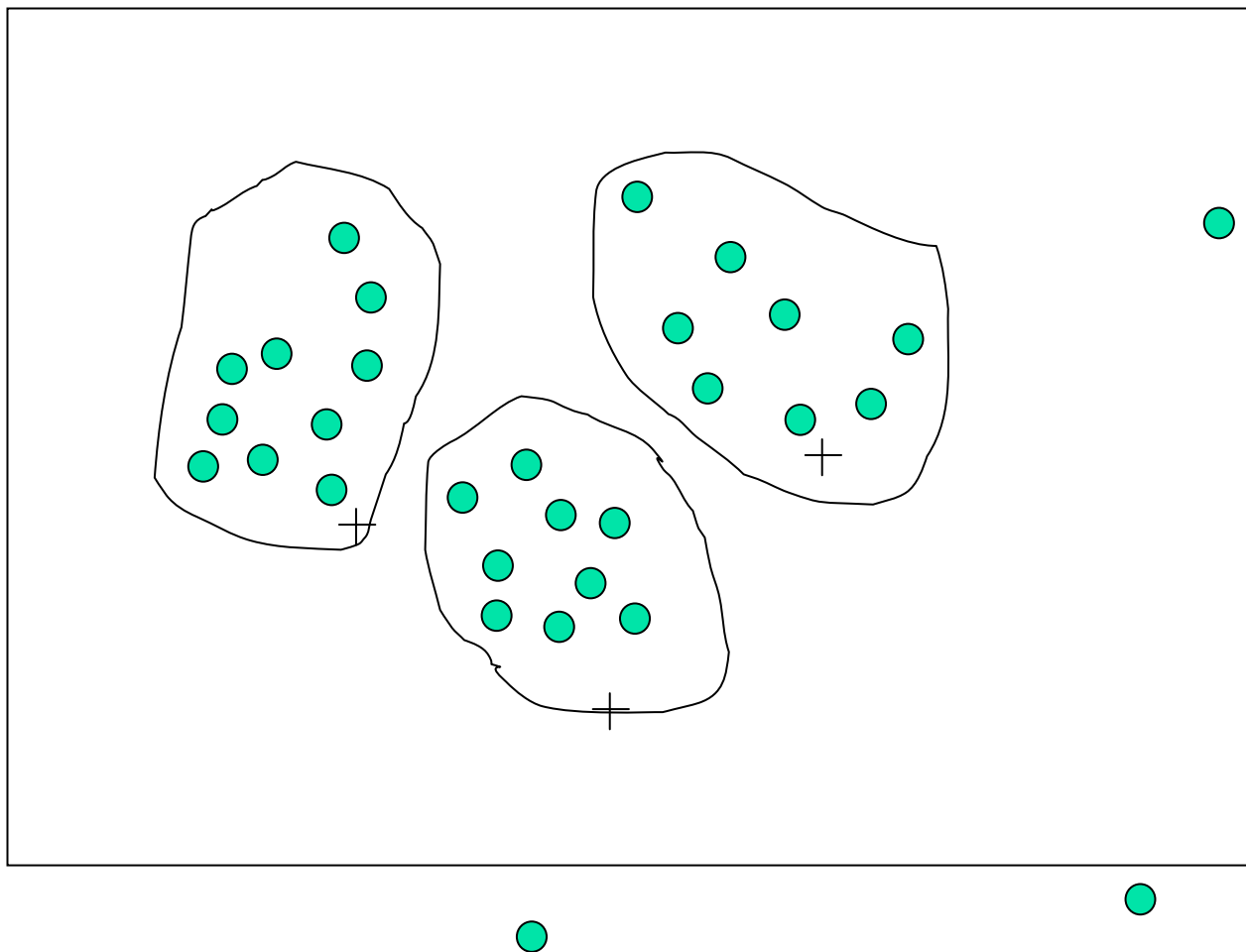


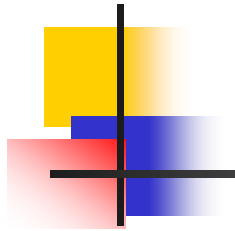
Binning Methods for Data Smoothing

- * Sorted data for price (in dollars): 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34
- * Partition into (equi-depth) bins:
 - **Bin 1:** 4, 8, 9, 15
 - **Bin 2:** 21, 21, 24, 25
 - **Bin 3:** 26, 28, 29, 34
- * Smoothing by bin means:
 - **Bin 1:** 9, 9, 9, 9
 - **Bin 2:** 23, 23, 23, 23
 - **Bin 3:** 29, 29, 29, 29
- * Smoothing by bin boundaries:
 - **Bin 1:** 4, 4, 4, 15
 - **Bin 2:** 21, 21, 25, 25
 - **Bin 3:** 26, 26, 26, 34

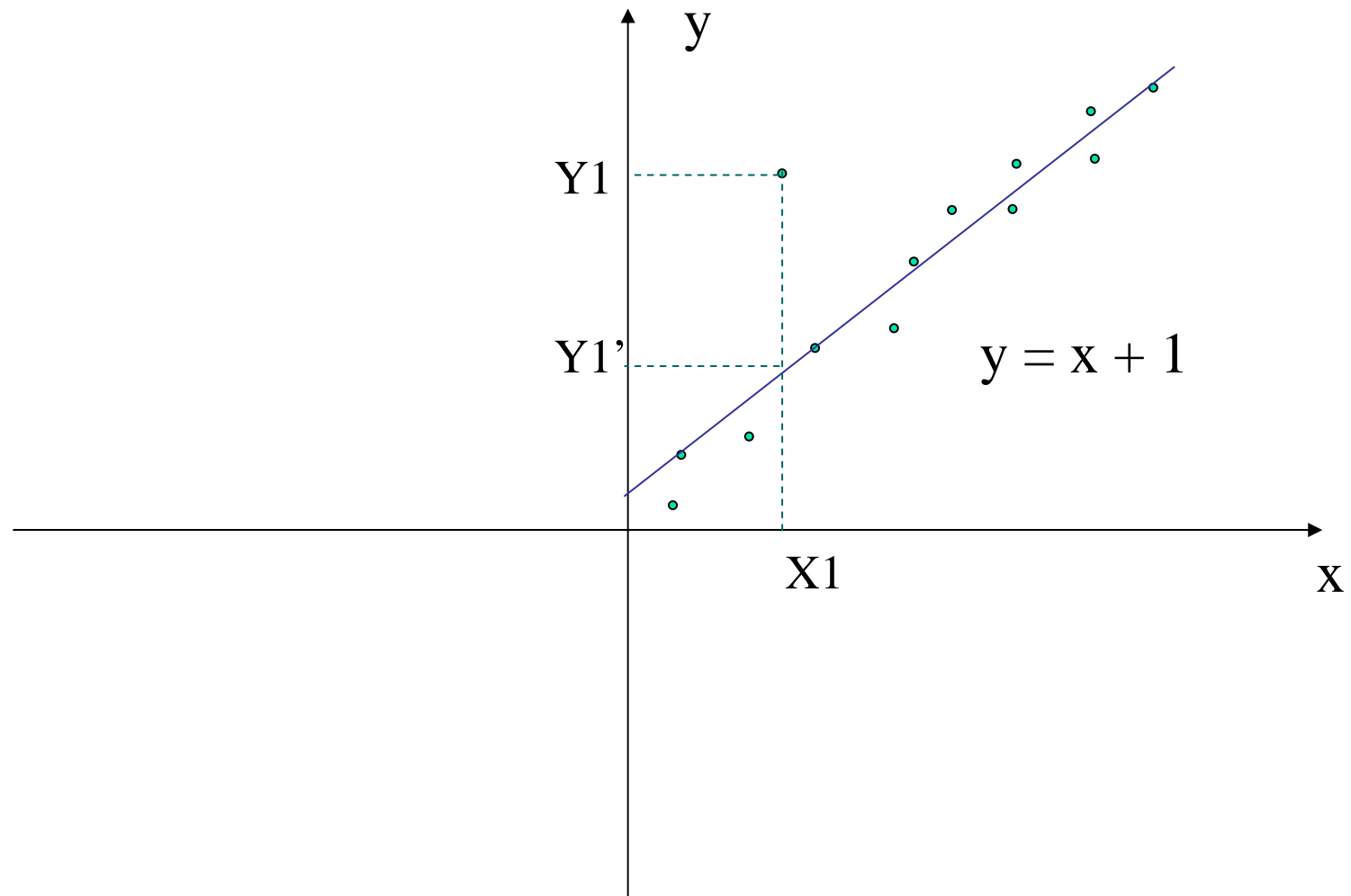


聚类分析





Regression





数据清理作为一个过程

- **数据偏差检测 Data discrepancy detection**
 - 使用元数据(数据性质的知识)(e.g., 领域, 长度范围, 从属, 分布)
 - 检查字段过载 **field overloading**
 - 检查唯一性规则, 连续性规则, 空值规则
 - 使用商业工具
 - 数据清洗 **Data scrubbing**: 使用简单的领域知识(e.g., 邮编, 拼写检查) 检查并纠正错误
 - 数据审计 **Data auditing**: 通过分析数据发现规则和联系发现违规者(孤立点)
- **数据迁移和集成**
 - 数据迁移工具 **Data migration tools**: 允许指定转换
 - 提取/变换/装入工具 **ETL (Extraction/Transformation/Loading) tools**: 允许用户通过图形用户界面指定变换
- **整合两个过程**
 - 两个过程迭代和交互执行(e.g., **Potter's Wheels**)



第2章: 数据预处理

- 为什么预处理数据?
- 数据清理
- 数据集成
- 数据归约
- 离散化和概念分层产生
- 小结



数据集成

■ 数据集成 **Data integration:**

- 合并多个数据源中的数据，存在一个一致的数据存储中
- 涉及3个主要问题：模式集成、冗余数据、冲突数据值

■ 模式集成 **Schema integration**

- 例如., **A.cust-id \equiv B.cust-#**
- 实体识别问题 **Entity identification problem:**
 - 多个数据源的真实世界的实体的识别, e.g., **Bill Clinton = William Clinton**
- 集成不同来源的元数据

■ 冲突数据值的检测 and 解决

- 对真实世界的实体，其不同来源的属性值可能不同
- 原因:不同的表示,不同尺度,公制 vs. 英制



数据集成中冗余数据处理

- 冗余数据 **Redundant data** （集成多个数据库时出现）
 - 目标识别：同一个属性在不同的数据库中有不同的名称
 - 衍生数据：一个属性值可由其他表的属性推导出, e.g., 年收入
- 相关分析 *correlation analysis* / 协方差分析 *covariance analysis*
 - 可用于检测冗余数据
- 小心的集成多个来源的数据可以帮助降低和避免结果数据集中的冗余和不一致，提高数据挖掘的速度和质量



相关分析 (数值数据)

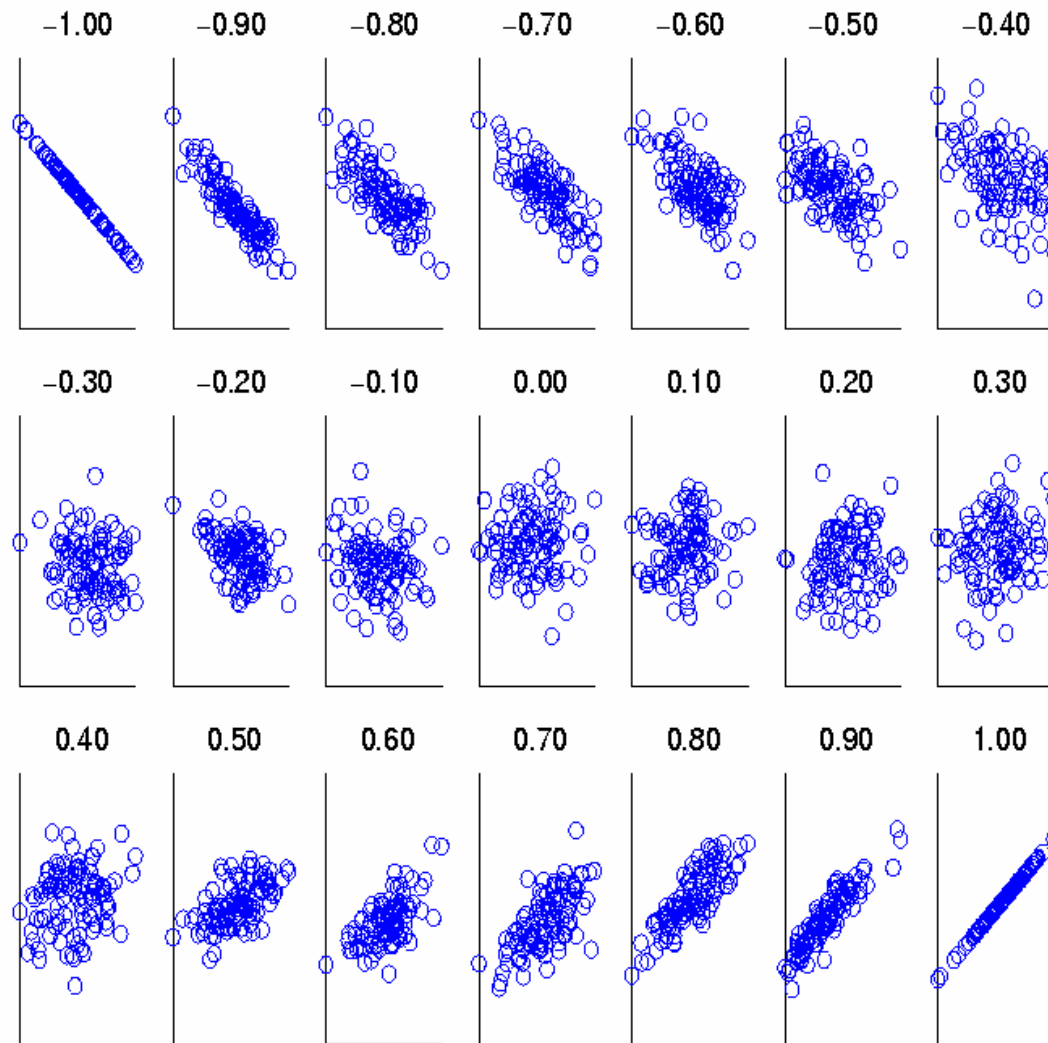
- **Correlation coefficient (also called Pearson's product moment coefficient)**
- 相关系数 (皮尔逊相关系数)

$$r_{A,B} = \frac{\sum_{i=1}^n (a_i - \bar{A})(b_i - \bar{B})}{(n-1)\sigma_A\sigma_B} = \frac{\sum_{i=1}^n (a_i b_i) - n\bar{A}\bar{B}}{(n-1)\sigma_A\sigma_B}$$

n元组个数, \bar{A} 和 \bar{B} 属性A和B上的平均值, σ_A and σ_B 分别为各自标准差, $\Sigma(a_i b_i)$ is the AB叉积 cross-product之和.

- If $r_{A,B} > 0$, A and B 正相关 (A's values increase as B's). 值越大相关程度越高.
- $r_{A,B} = 0$: 不相关; $r_{AB} < 0$: 负相关

相关性的视觉评价



**Scatter plots
showing the
similarity from
-1 to 1.**



相关 (线形关系)

- 相关测量的是对象间的线性关系
- **To compute correlation, we standardize data objects, A and B, and then take their dot product**

$$a'_k = (a_k - \text{mean}(A)) / \text{std}(A)$$

$$b'_k = (b_k - \text{mean}(B)) / \text{std}(B)$$

$$\text{correlation}(A, B) = A' \bullet B'$$



协方差Covariance (Numeric Data)

- **Covariance is similar to correlation**

$$Cov(A, B) = E((A - \bar{A})(B - \bar{B})) = \frac{\sum_{i=1}^n (a_i - \bar{A})(b_i - \bar{B})}{n}$$

Correlation coefficient:

$$r_{A,B} = \frac{Cov(A, B)}{\sigma_A \sigma_B}$$

n元组个数, \bar{A} 和 \bar{B} 属性A和B上的平均值, σ_A and σ_B 分别为各自标准差.

- **正covariance: If $Cov_{A,B} > 0$, 则A 和B 同时倾向于大于期望值.**
- **负covariance: If $Cov_{A,B} < 0$, 则如果 A > 其期望值, B is likely to be smaller than its expected value.**
- **Independence: $Cov_{A,B} = 0$ but the converse is not true:**
 - **Some pairs of random variables may have a covariance of 0 but are not independent. Only under some additional assumptions (e.g., the data follow multivariate normal distributions) does a covariance of 0 imply independence**



Co-Variance: An Example

$$Cov(A, B) = E((A - \bar{A})(B - \bar{B})) = \frac{\sum_{i=1}^n (a_i - \bar{A})(b_i - \bar{B})}{n}$$

- It can be simplified in computation as

$$Cov(A, B) = E(A \cdot B) - \bar{A}\bar{B}$$

- 设两个股票 A 和 B 一周内值如下 (2, 5), (3, 8), (5, 10), (4, 11), (6, 14).
- 问：如果股票是由同行业趋势的影响，它们的价格将一起上升或下降？
 - $E(A) = (2 + 3 + 5 + 4 + 6) / 5 = 20/5 = 4$
 - $E(B) = (5 + 8 + 10 + 11 + 14) / 5 = 48/5 = 9.6$
 - $Cov(A, B) = (2 \times 5 + 3 \times 8 + 5 \times 10 + 4 \times 11 + 6 \times 14) / 5 - 4 \times 9.6 = 4$
- Thus, A and B rise together since $Cov(A, B) > 0$.

相关分析 (名义数据Nominal Data)

■ χ^2 (chi-square) test 开方检验

- σ_{ij} 是 (a_i, b_j) 的观测频度 (实际计数)
- e_{ij} 是 (a_i, b_j) 的期望频度
- N 数据元组的个数

$$\chi^2 = \sum_{i=1}^c \sum_{j=1}^r \frac{(\sigma_{ij} - e_{ij})^2}{e_{ij}}$$

		属 性 A			
		a_1	a_2	\vdots	a_c
B	b_1				
	b_2				
	\vdots				
	b_r				

(A= a_i , B= b_j)

$$e_{ij} = \frac{\text{count}(A = a_i) * \text{count}(B = b_j)}{N}$$

- χ^2 值越大, 相关的可能越大
- 对 χ^2 值贡献最大的项, 其实际值与期望值相差最大的相
- 相关不意味着因果关系

Chi-Square 卡方值计算: 例子

	Play chess	Not play chess	Sum (row)
看小说	250(90)	200(360)	450
不看小说	50(210)	1000(840)	1050
Sum(col.)	300	1200	1500

$$e_{11} = \frac{\text{count(看小说)} * \text{count(下棋)}}{N} = \frac{450 * 300}{1500} = 90$$

- **X² (chi-square)** 计算(括号中的值为期望计值, 由两个类别的分布数据计算得到)

$$\chi^2 = \frac{(250 - 90)^2}{90} + \frac{(50 - 210)^2}{210} + \frac{(200 - 360)^2}{360} + \frac{(1000 - 840)^2}{840} = 507.93$$

- 结果表明like_fiction 和play_chess 关联



数据变换Data Transformation

- 光滑: 去掉噪音, 技术: 分箱、回归、聚类
 - 聚集Aggregation: 汇总, 数据立方体构造
 - 数据泛化Generalization: 概念分层
 - 规范化Normalization: 按比例缩放到一个具体区间
 - 最小-最大规范化
 - z-score 规范化
 - 小数定标规范化
 - 属性Attribute/特征feature 构造
 - 从给定的属性构造新属性
 - 机器学习中称为: 特征构造
- } 数据规约



规范化数据的方法

- 最小-最大规范化 **min-max normalization**

$$v' = \frac{v - \min_A}{\max_A - \min_A} (\text{new_max}_A - \text{new_min}_A) + \text{new_min}_A$$

- 新数据可能“越界”

- **z-score normalization** $v' = \frac{v - \text{均值}_A}{\text{标准差}_A}$

- **normalization by decimal scaling**

- 移动属性A的小数点位置(移动位数依赖于属性A的最大值)

$$v' = \frac{v}{10^j} \quad J \text{ 为使得 } \text{Max}(|v'|) < 1 \text{ 的整数中最小的那个}$$



第2章: 数据预处理

- 为什么预处理数据?
- 数据清理
- 数据集成
- 数据归约
- 离散化和概念分层产生
- 小结

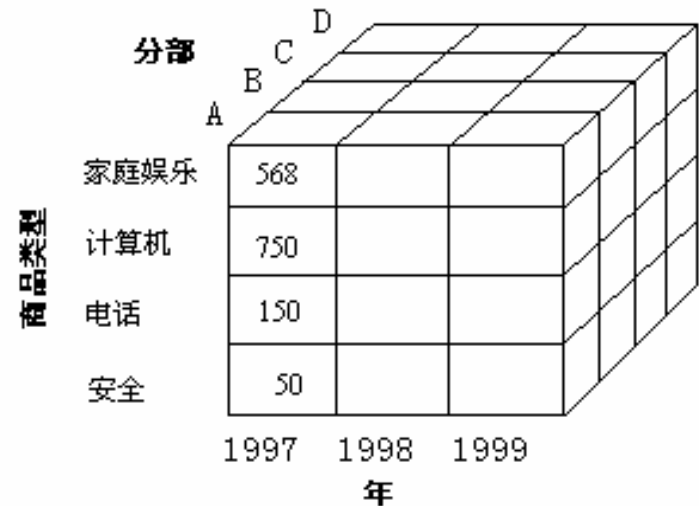


数据规约策略

- 在完整数据上的分析/挖掘耗时太长，以至于不现实
- **Data reduction** 获得数据集的一个规约表示，小很多，接近保持原数据的完整性，使得可得到相同/几乎相同的分析结果
- 数据规约策略如下；
 - 数据立方体聚集：聚集数据立方体结构的数据
 - 维度规约—去除不重要的属性
 - 主成份分析Principal Components Analysis (PCA)
 - 特征子集选择Feature subset selection,
 - 属性产生
 - 数据压缩 Data Compression
 - 基于离散小波变换的数据压缩：图像压缩
 - 数值规约—用某种表示方式替换/估计原数据
 - Regression and Log-Linear Models
 - Histograms, clustering, sampling
 - 离散化和产生概念分层

数据立方体聚集

- 数据立方体存储多维聚集信息
 - 某抽象层上建的数据立方体称为方1
 - 最底层建的方体称为基本方体(base)
 - 最高层的立方体称为 顶点方体(apex cuboid)
- 每个更高层的抽象将减少数据的规模



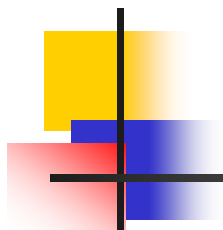
年=1999			
年=1998			
年=1997			
季度	销售额	年	销售额
Q1	\$224,000	1997	\$1,568,000
Q2	\$408,000	1998	\$2,356,000
Q3	\$350,000	1999	\$3,594,000
Q4	\$586,000		

- 使用合适的抽象层上的数据
 - 对数据立方体聚集得到与任务相关的最小立方体

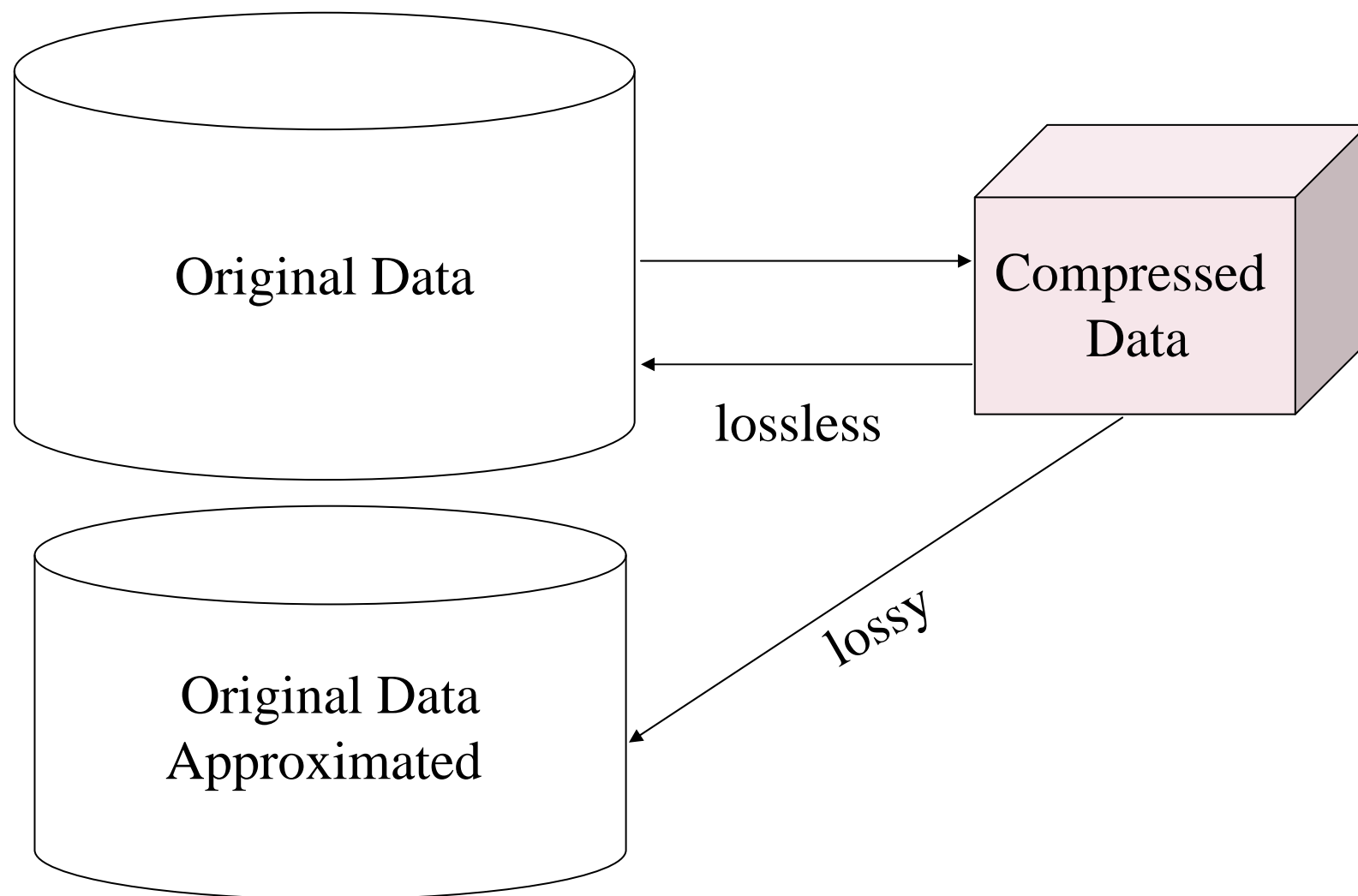


数据压缩 Data Compression

- 字符串压缩
 - 有丰富的理论和调优的算法
 - 典型的是有损压缩；
 - 但只有有限的操作是可行的
- 音频/视频压缩
 - 通常有损压缩，逐步细化
 - 有时小片段的信号可重构，而不需要重建整个信号
- 时间序列不是音频
 - 通常短，随时间缓慢变化
- 维度和数值规约可以被看成是数据压缩的一种形式

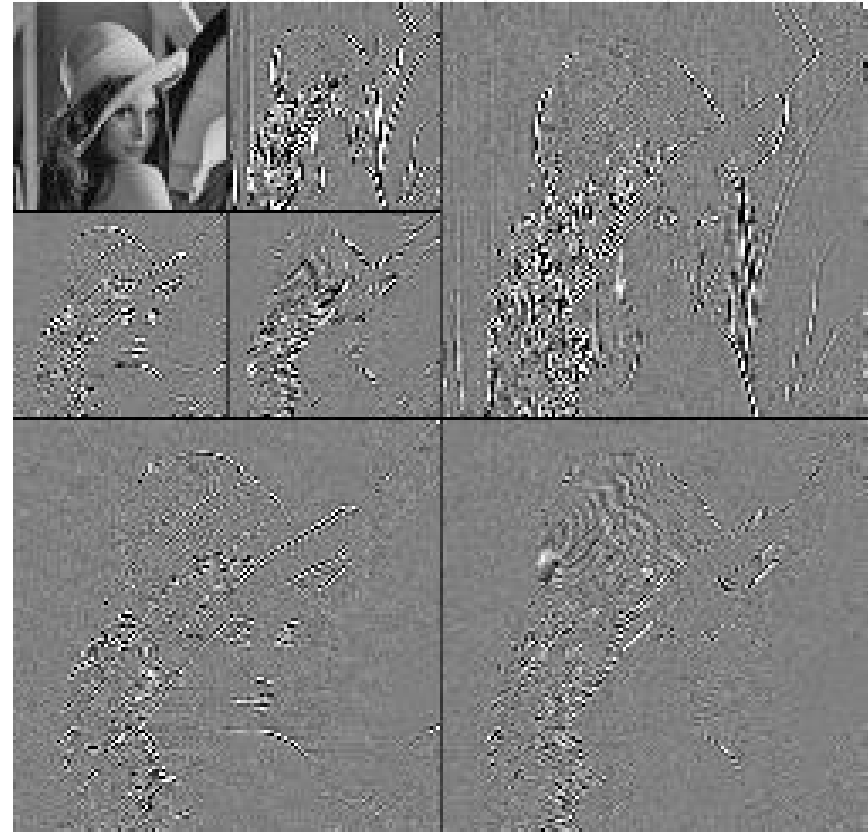
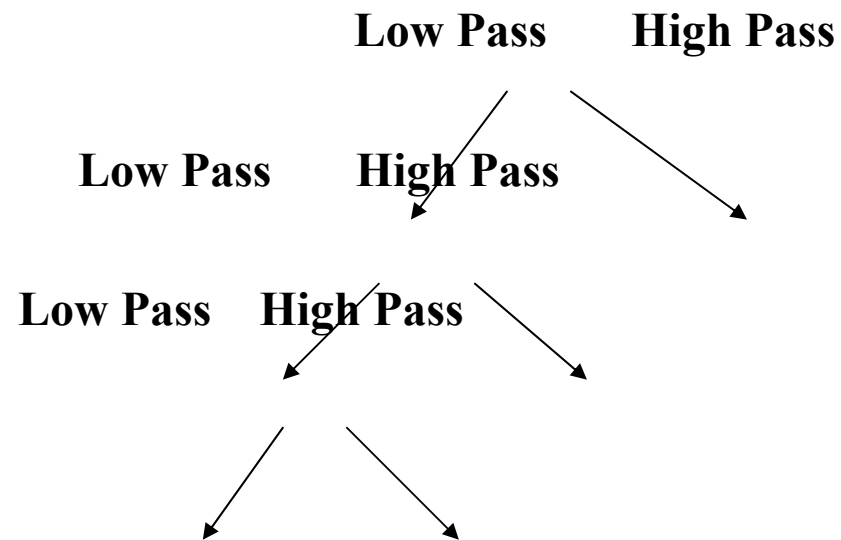


数据压缩



DWT for Image Compression

- Image



- Discrete wavelet transform(DWT):



维度规约-特征选择

- 特征选择 **Feature selection** (i.e., 属性子集选择):
 - 删除不相关/冗余属性, 减少数据集
 - 找出最小属性集, 类别的数据分布尽可能接近 使用全部属性值的原分布
 - 减少了发现的模式数目, 容易理解
- d 个属性, 有 2^d 个可能的属性子集
- 启发式方法 **Heuristic methods** (因为指数级的可能性):
 - 局部最优选择, 期望获得全局最优解
 - 逐步向前选择
 - 逐步向后删除 **step-wise backward elimination**
 - 向前选择和向后删除结合
 - 决策树归纳 **decision-tree induction**



维度规约-启发式特征选择方法

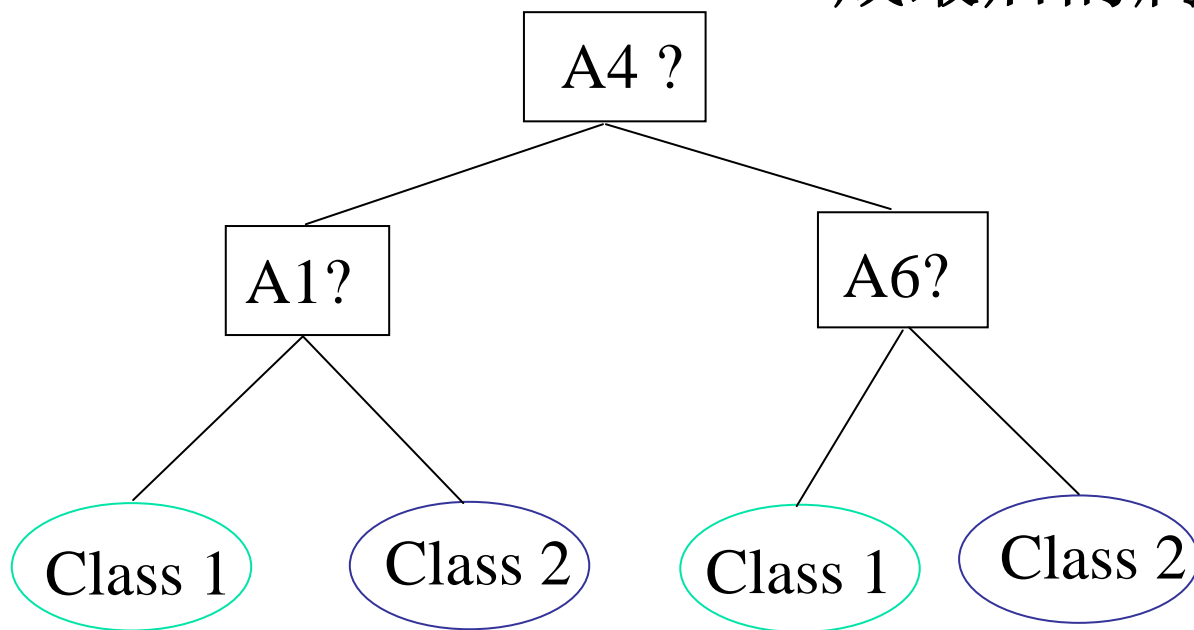
- 特征独立性假设下，最好的单个属性：统计显著性检验方法
 - 两样本t-test，等
- 逐步最好特征选择：
 - 先选出最好的单个属性
 - 下一个最好的特征加进来, ...
- 逐步向后删除：
 - 重复删除最差的特征
- 向前选择和向后删除结合
- 优化分支定界 (**Optimal branch and bound**)
 - 使用特征删除和回溯
- 中止条件各有不同。

维度规约-决策树规约

最初的属性集合:

{A1, A2, A3, A4, A5, A6}

出现在决策树中的属性构成最后的属性子集



-----> 最后的集合: {A1, A4, A6}

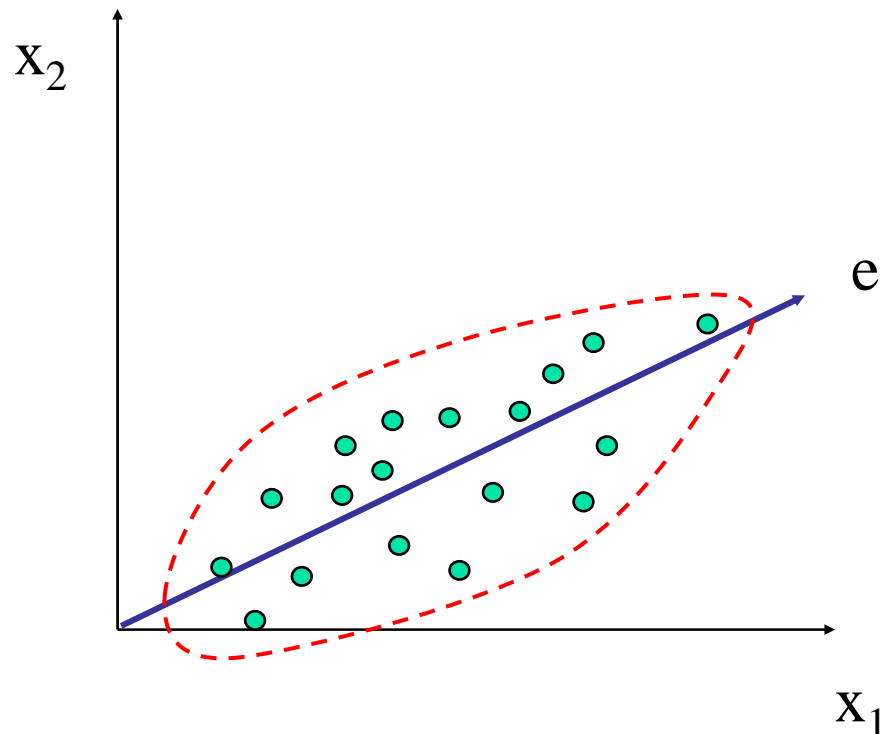


维度规约-属性/特征产生

- **Feature Generation** 产生新的属性，其可以比原始属性更有效地表示数据的重要信息。
- 三个一般方法：
 - 属性提取 **Attribute extraction**
 - 特定领域的
 - 映射数据到新空间
 - E.g., 傅立叶变换, **wavelet transformation**, 流形方法 (**manifold approaches**)
 - 属性构造
 - 组合特征
 - 数据离散化 **Data discretization**

主成分分析 (PCA)

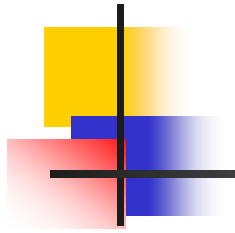
- **principal component analysis, K-L变换**
- 找到一个投影, 其能表示数据的最大变化
- 原始数据投影到一个更小的空间中, 导致维度减少.
 - 发现的协方差矩阵的特征向量, 用这些特征向量定义新的空间



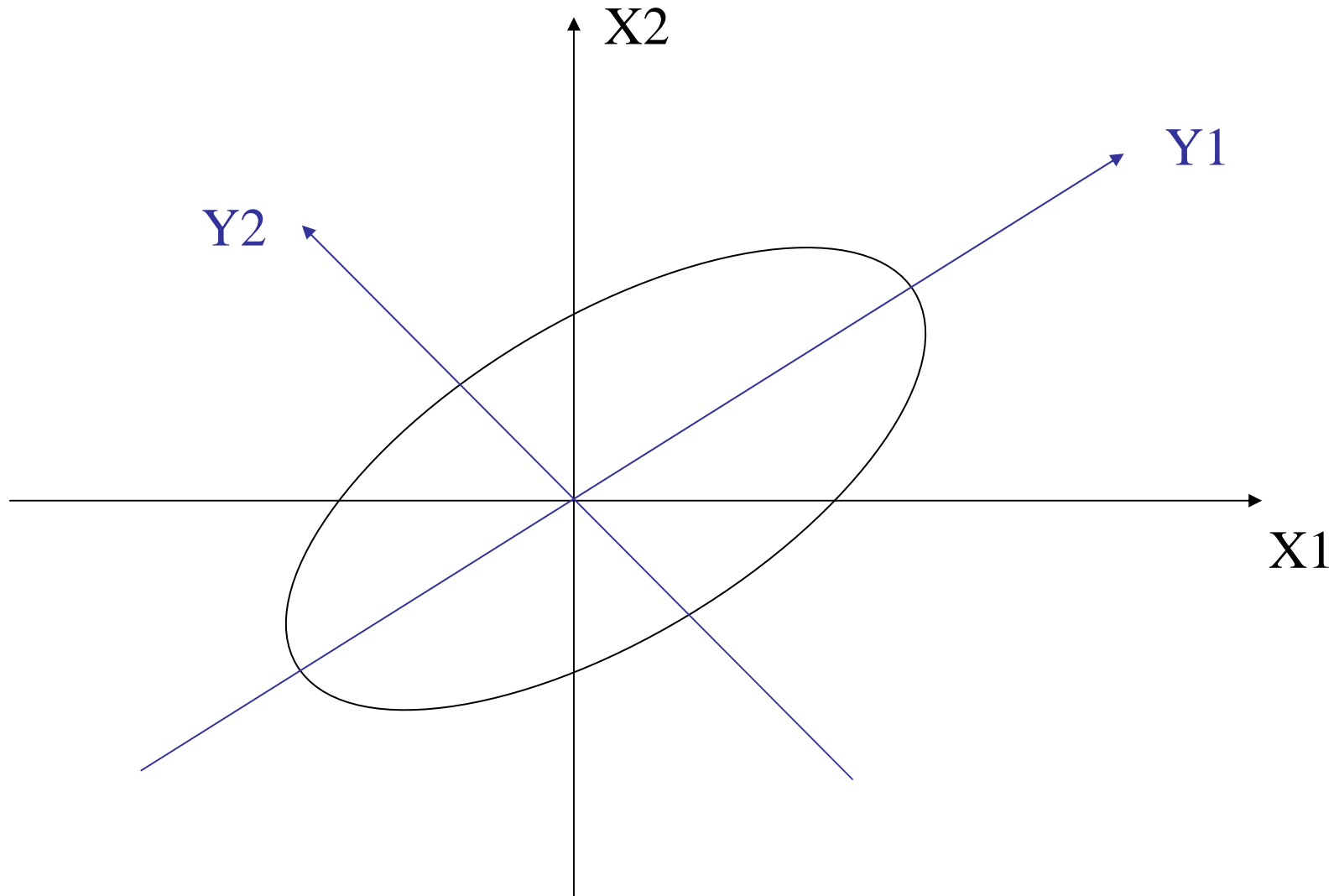


主成分分析 (Steps)

- 给定 p 维空间中的 N 各点, 找到 $k \leq p$ 个正交向量 (*principal components*) 可以很好表示原始数据的
 - 归范化输入数据: 每个属性值位于相同的区间内
 - 计算 k 个标准正交向量, i.e., *principal components*
 - 每个输入的点是这 k 个主成分的线性组合
 - **The principal components are sorted in order of decreasing “significance” or strength**
 - **Since the components are sorted, the size of the data can be reduced by eliminating the *weak components* (i.e., using the strongest principal components, it is possible to reconstruct a good approximation of the original data)**
- **Works for numeric data only**



Principal Component Analysis





数值规约

- 选择替代的、“较小的”数据表示形式
- 参数方法
 - 假设数据适合某个模型，估计模型参数，仅存储的参数，并丢弃数据（孤立点除外）
 - 对数线性模型：
 - 基于一个较小的维组合的子集来估计 离散属性的多维空间中每个点的概率
- 非参数方法
 - 不假定模型
 - **histograms, clustering, sampling**



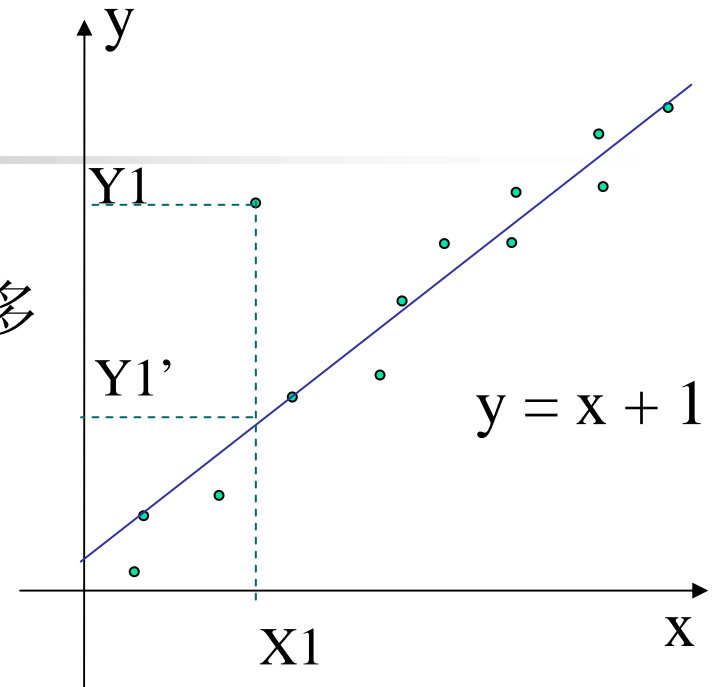
回归和对数线性模型

- 线性回归: 数据拟合到一条直线上
 - 通常使用最小二乘法拟合
- 多元线性回归
 - 允许响应变量 Y 表示为多个预测变量的函数
- 对数线性模型:
 - 近似离散的多维概率分布

回归分析

- 研究因变量/响应变量 Y (**dependent variable/response variable**) 对个或多个自变量/解释变量(*independent variable / explanatory variable*)的相依关系的方法的统称

- 参数需要估计以最好的拟合给定的数据
- 绝大多数情况“最好的拟合”是由最小二乘法(*least squares method*)实现, 其他的方法也有

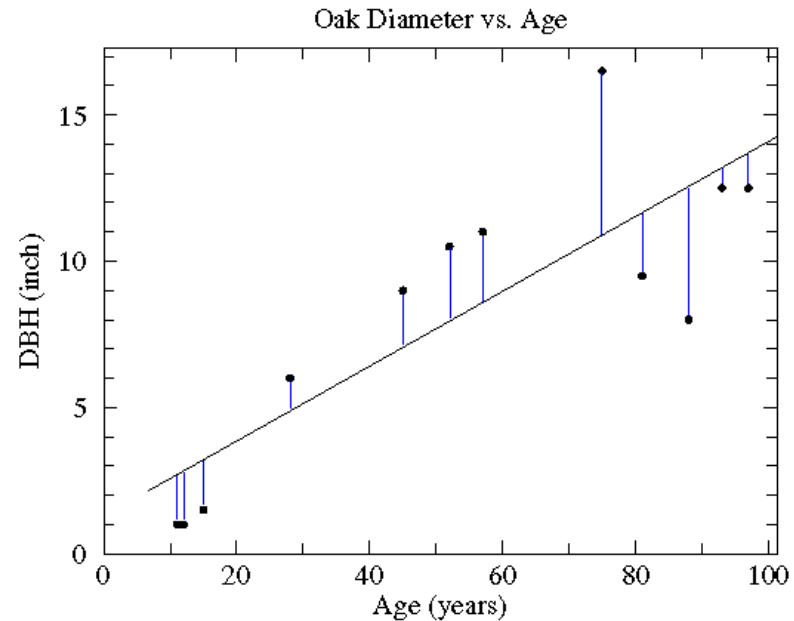
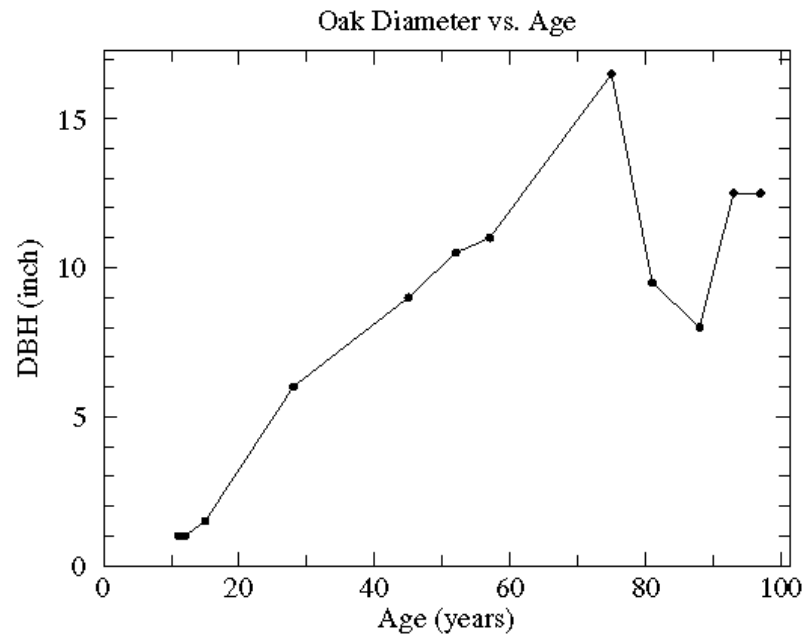


- 用于预测（包括时间序列数据的预测），推断，假设检验和因果关系的建模

线性回归-用于预测

Y: --diameter at breast height(*DBH*) \leftrightarrow X: -- Age

	0	1	2	3	4	5	6	7	8	9	10	11	12
Y	?	1.0	1.0	1.5	6.0	9.0	10.5	11	16.5	9.5	8.0	12.5	12.5
X	34	11	12	15	28	45	52	57	75	81	88	93	97



线性回归(cont.)

- Given x , construct the linear regression model for y against x as:

$$y = \alpha + \beta x + e$$

- Least squares estimation of y given variable x is:

of α and β is $\hat{\alpha} = \bar{y} - \hat{\beta}\bar{x}$ and

$$\hat{\beta} = \frac{s_{xy}}{s_{xx}}, \quad \text{where} \quad s_{xy} = \frac{1}{n-1} \sum_{j=1}^n (x_j - \bar{x})(y_j - \bar{y})$$

is the empirical covariance between x and y ,

$$s_{xx} = \frac{1}{n-1} \sum_{j=1}^n (x_j - \bar{x})^2$$

$$\hat{y} = \bar{y} + \frac{s_{xy}}{s_{yy}} (x - \bar{x}).$$

多元线性回归

- 响应变量: w , 自变量: A_1, A_2, \dots, A_k .
- “5” 样本数目

$$w \quad A_1 \quad A_2 \quad \dots \quad A_k \quad (1)$$

$$\begin{pmatrix} g_1^T \\ g_{s_1}^T \\ \vdots \\ g_{s_k}^T \end{pmatrix} = \begin{pmatrix} \alpha & w^T \\ b & A \end{pmatrix}$$

$$= \begin{pmatrix} \alpha & w_1 & w_2 & w_3 & w_4 & w_5 \\ b_1 & A_{1,1} & A_{1,2} & A_{1,3} & A_{1,4} & A_{1,5} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ b_k & A_{k,1} & A_{k,2} & A_{k,3} & A_{k,4} & A_{k,5} \end{pmatrix}$$

$$\min_{\mathbf{x}} \|A^T \mathbf{x} - \mathbf{w}\|_2.$$

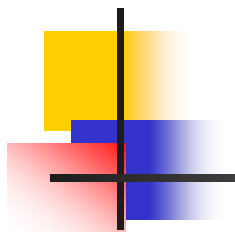
value α is estimated as a linear combination of the expression values of the genes

$$\alpha = \mathbf{b}^T \mathbf{x} = \mathbf{b}^T (A^T)^{\dagger} \mathbf{w},$$

$$w \simeq \mathbf{x}_1 \mathbf{a}_1 + \mathbf{x}_2 \mathbf{a}_2 + \dots + \mathbf{x}_k \mathbf{a}_k,$$

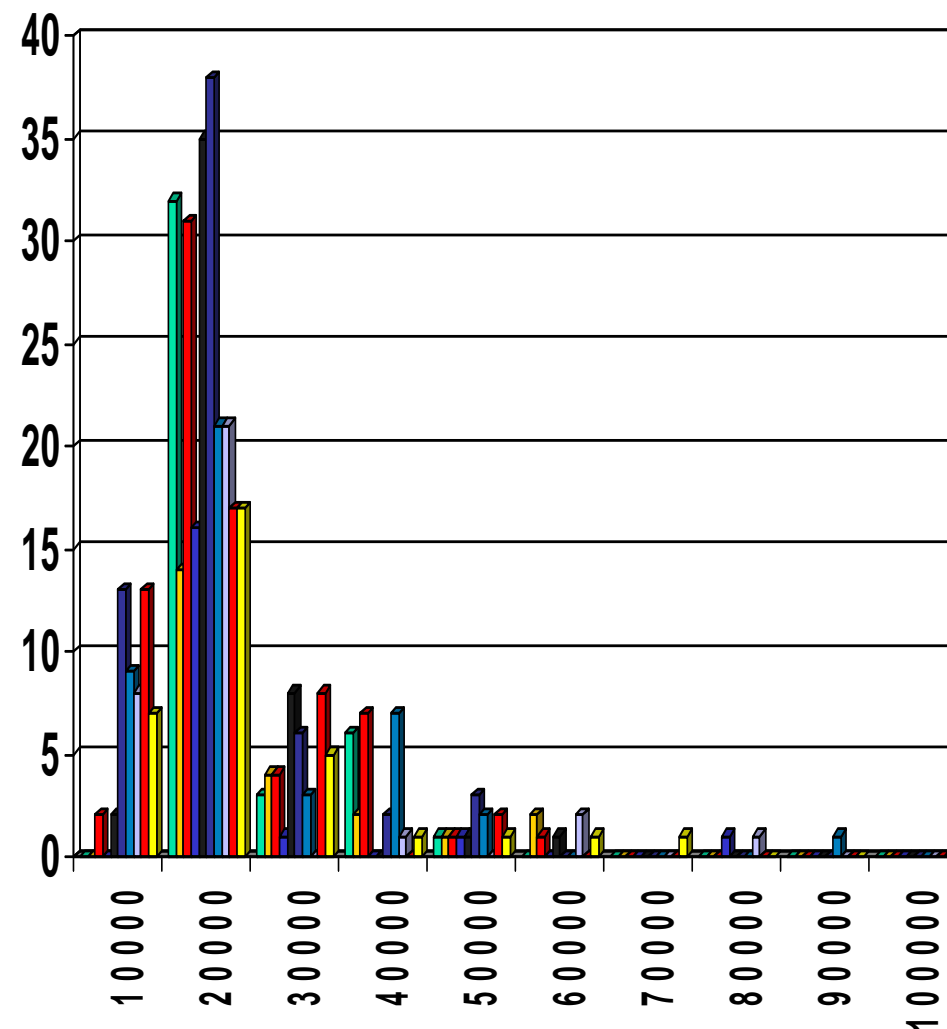
$\mathbf{a}_1, \dots, \mathbf{a}_k$ are the coefficients of the linear combination in the least squares formulation (2). According to the least squares formulation (2), the value α in \mathbf{g}_1 can be estimated by

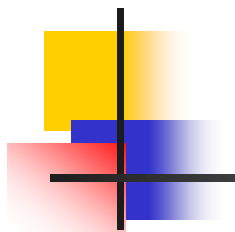
$$\alpha = \mathbf{b}^T \mathbf{x} = b_1 \mathbf{x}_1 + b_2 \mathbf{x}_2 + \dots + b_k \mathbf{x}_k.$$



直方图Histograms

- 把数据划分成不相交的子集或桶
- 一维时可用动态规划优化构建
- 涉及量化问题





聚类Clustering

- 将对象划分成集/簇, 用簇的表示替换实际数据
 - 技术的有效性依赖于数据的质量
- 使用层次聚类, 并多维索引树结构存放
- 非常多的聚类算法和定义

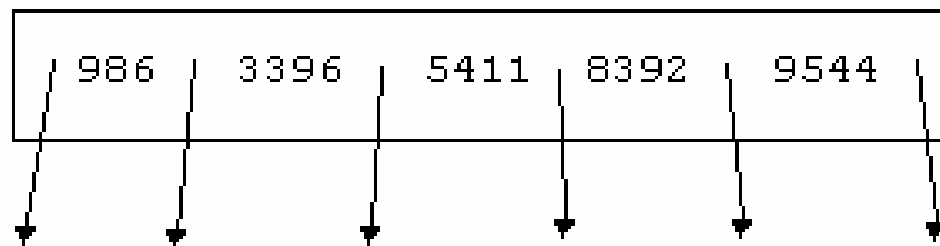


图 3.12 给定数据集的 B+树的根



抽样Sampling

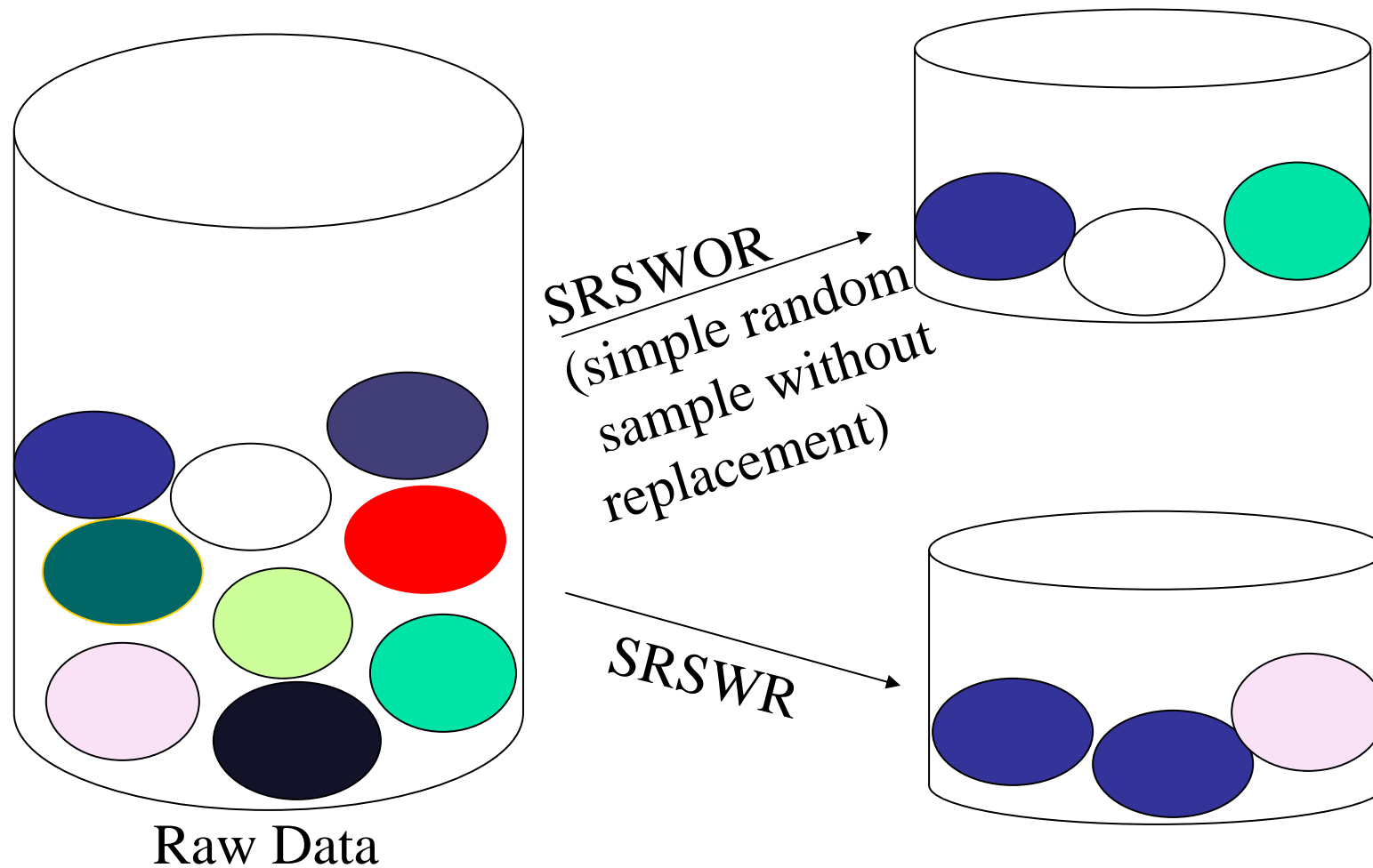
- 抽样: 获得一个小的样本集 s 来表示整个数据集 N
- 允许一个挖掘算法运行复杂度子线性于样本大小
- 关键原则: 选择一个有代表性的数据子集
 - 数据偏斜时简单随机抽样的性能很差
 - 发展适应抽样方法: 分层抽样
- **Note: Sampling may not reduce database I/Os (page at a time)**

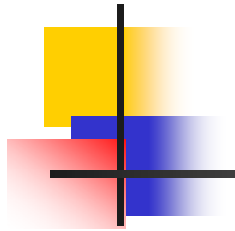


抽样类型 Types of Sampling

- 简单随机抽样 Simple random sampling
 - 相同的概率选择任何特定项目
- 无放回抽样 Sampling without replacement
 - **Once an object is selected, it is removed from the population**
- 放回抽样 Sampling with replacement
 - 一个被抽中的目标不从总体中去除
- 分层抽样 Stratified sampling:
 - 把数据分成不相交部分(层), 然后从每个层抽样(按比例/大约相同比例的数据)
 - 偏斜数据

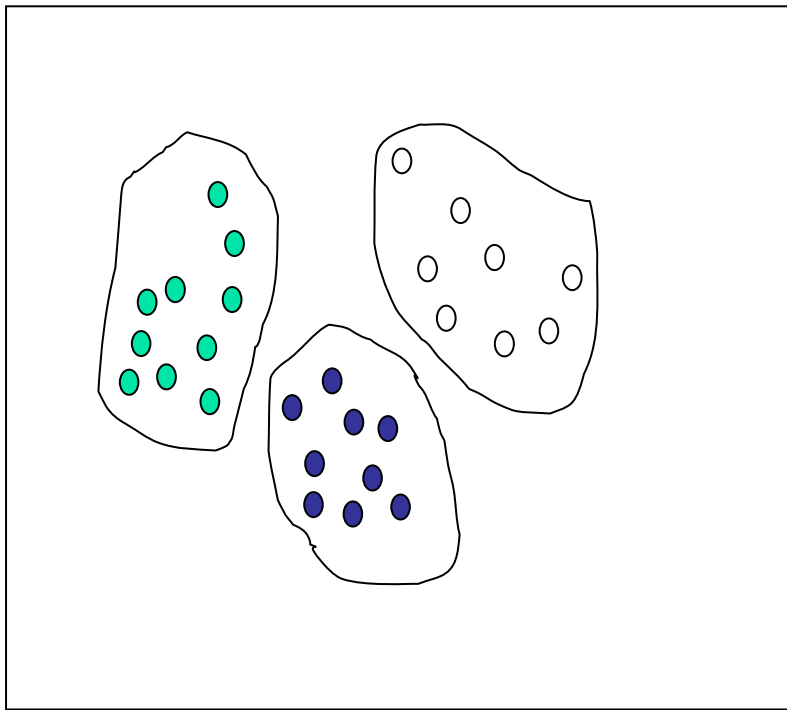
Sampling: With or without Replacement



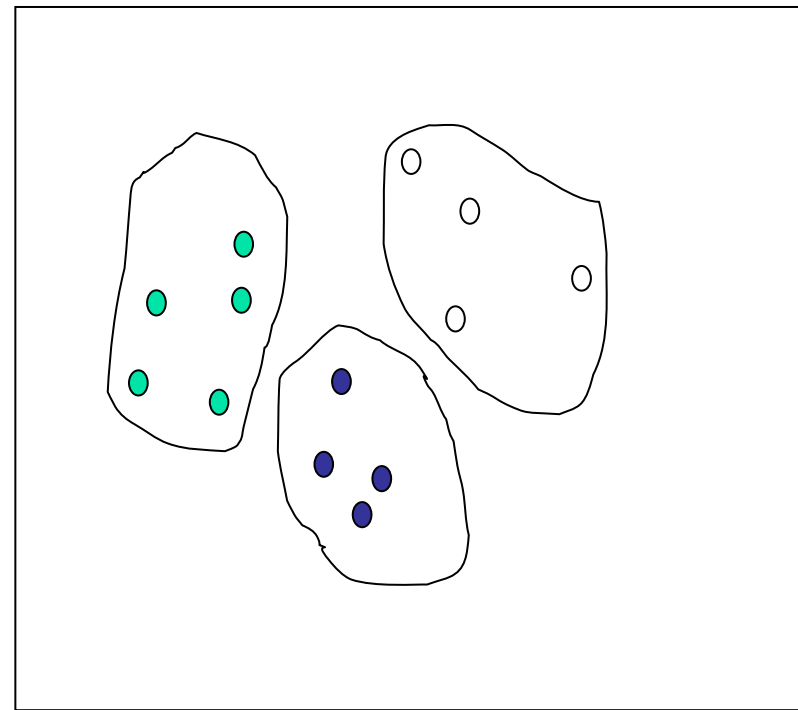


Sampling: Cluster or Stratified Sampling

Raw Data



Cluster/Stratified Sample





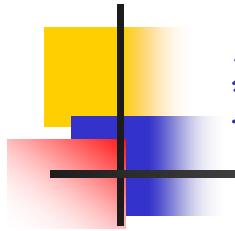
第2章: 数据预处理

- 为什么预处理数据?
- 数据清理
- 数据集成
- 数据归约
- 离散化和概念分层产生
- 小结



离散化 Discretization和概念分成

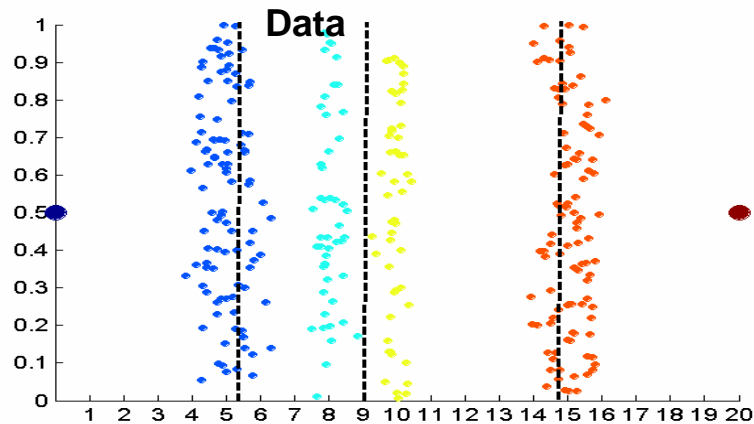
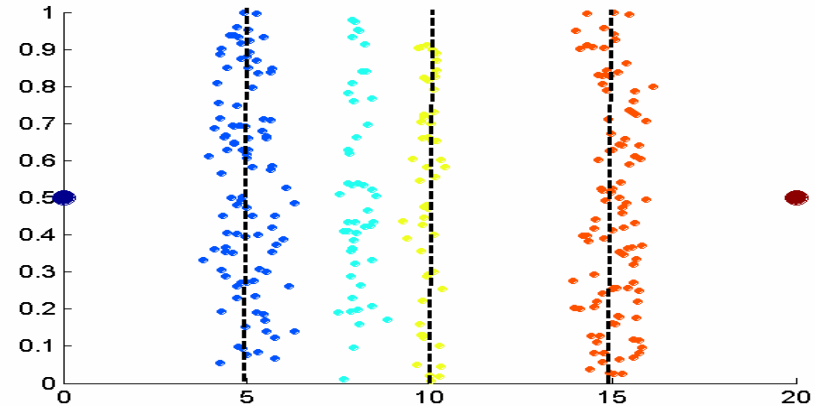
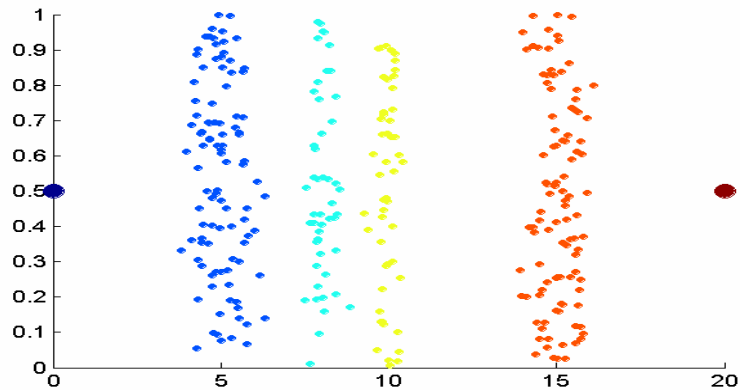
- 三种类型属性：
 - 名义 — values from an unordered set, color, profession
 - 顺序数 — values from an ordered set , e.g., military or academic rank
 - 连续 — real numbers
- 离散化 **Discretization**: 把连续属性的区域分成区间
 - 区间标号可以代替实际数据值
 - 利用离散化减少数据量
 - 有监督 vs. 无监督: 是否使用类的信息
 - 某个属性上可以递归离散化
 - 分裂 **Split (top-down)** vs. 合并 **merge (bottom-up)**
 - 自顶向下: 由一个/几个点开始递归划分整个属性区间
- 递归离散化属性, 产生属性值分层/多分辨率划分: 概念分层



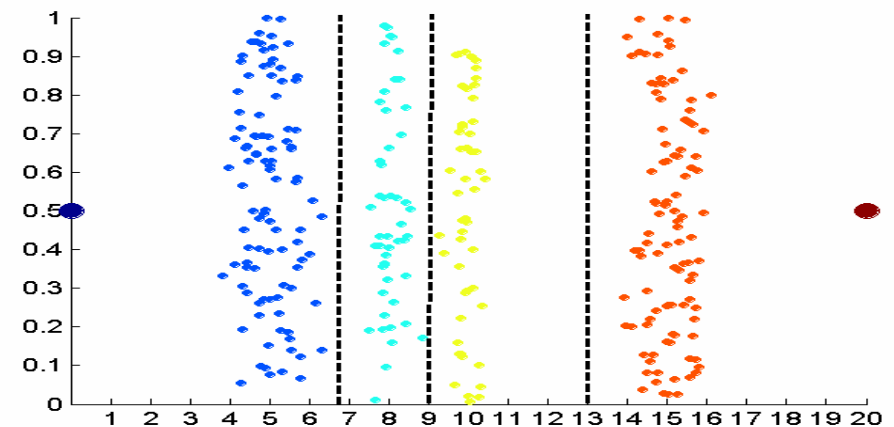
数值数据离散化/概念分层

- 分箱 **Binning**(**Top-down split, unsupervised**)
- 直方图 (**Top-down split, unsupervised**)
- 聚类 (**unsupervised, top-down split or bottom-up merge**)
- 基于 χ^2 分析的区间合并(**unsupervised, bottom-up merge**)
- 基于熵 **Entropy-based discretization**
- 根据自然划分

不用类别(Binning vs. Clustering)



**Equal frequency
(binning)**



**K-means clustering leads to
better results**

基于熵Entropy的离散化

给定一个数据元素的集合 S 。基于熵对 A 离散化的方法如下：

1. A 的每个值可以认为是一个潜在的区间边界；
2. 选择的阈值 T 使其后划分得到的信息增益最大

$$I(S, T) = \frac{|S_1|}{|S|} Ent(S_1) + \frac{|S_2|}{|S|} Ent(S_2)$$

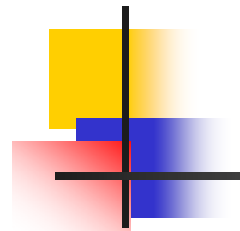
其中， S_1 和 S_2 分别对应于 S 中满足条件 $A < T$ 和 $A \geq T$ 的样本

$$Ent(S_1) = - \sum_{i=1}^m p_i \log_2(p_i)$$

其中， p_i 是类 i 在 S_1 中的概率，

等于 S_1 中类 i 的样本数除以 S_1 中的样本总数。

3. 直到满足某个终止条件 $Ent(S) - I(S, T) > \delta$



Chi-merge离散化

- **Chi-merge: χ^2 -based discretization**
 - **有监督: use class information**
 - **自低向上: find the best neighboring intervals (具有相似的类别分布, i.e., low χ^2 values) to merge**
 - **递归地合并, until a predefined stopping condition**



由自然划分离散化

■ 3-4-5 规则

- 如果最高有效位包含 3, 6, 7 or 9 个不同的值, **partition the range into 3 个等宽区间** (7: 2-3-2分成3个区间)
- 2, 4, or 8 不同的值, 区域分成 4 个等宽区间
- 1, 5, or 10 不同的值, 区域分成5 个等宽区间
- 类似地, 逐层使用此规则



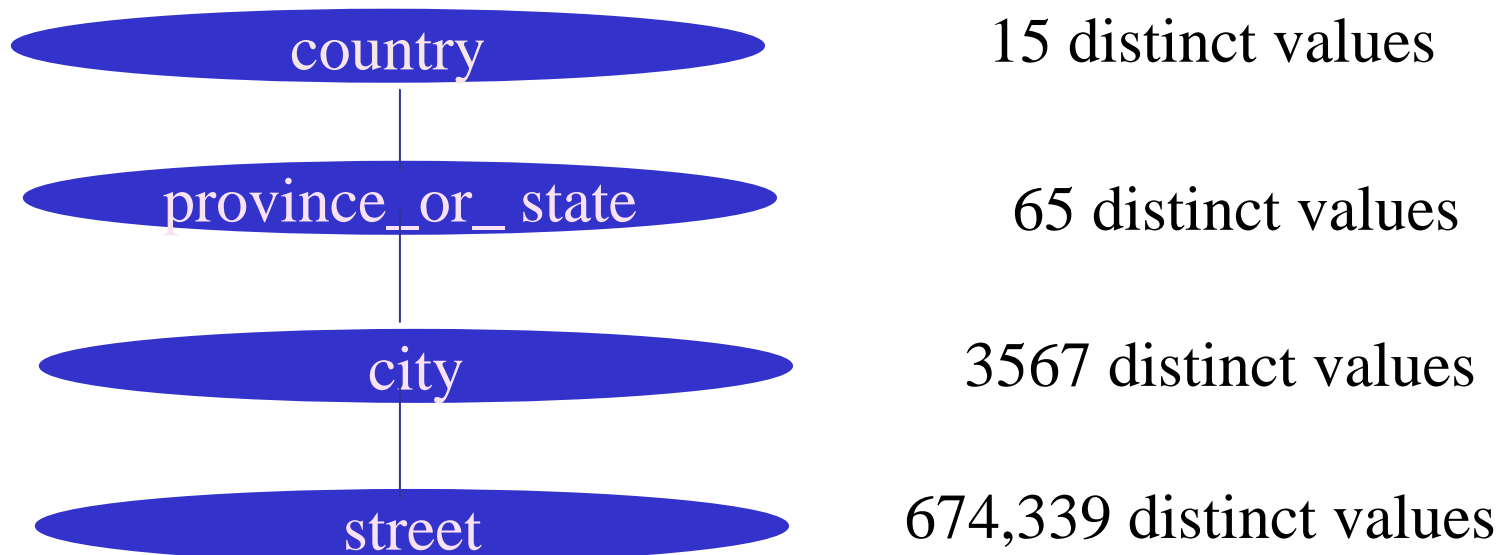
分类数据的概念分层 Categorical Data

- 用户/专家在模式级显式地指定属性的偏序
 - **street < city < state < country**
- 通过显式数据分组说明分层
 - {厄巴纳, 香槟, 芝加哥} < **Illinois**
- 只说明属性集
 - 系统自动产生属性偏序, 根据 每个属性下不同值的数据
 - 启发式规则: 相比低层, 高层概念的属性通常有较少取值
 - **E.g., street < city < state < country**
- 只说明部分属性值



自动产生概念分层

- **Some concept hierarchies can be automatically generated based on the analysis of the number of distinct values per attribute in the given data set**
 - 含不同值最多的属性放在层次的最低层
 - **Note: Exception—weekday, month, quarter, year**





Summary

- **Data preparation is a big issue for both warehousing and mining**
- **Data preparation includes**
 - **Data cleaning and data integration**
 - **Data reduction and feature selection**
 - **Discretization**
- **A lot a methods have been developed but still an active area of research**

Data Reduction, Transformation, Integration

- **Data Quality**
- **Major Tasks in Data Preprocessing**
- **Data Cleaning and Data Integration**
 - **Data Cleaning**
 - i. Missing Data and Misguided Missing Data
 - ii. Noisy Data
 - iii. Data Cleaning as a Process
 - **Data Integration Methods**
- **Data Reduction**
 - **Data Reduction Strategies**
 - **Dimensionality Reduction**
 - i. Principal Component analysis
 - ii. Feature Subset Selection
 - iii. Feature Creation
 - **Numerosity Reduction**
 - i. Parametric Data Reduction: Regression and Log-Linear Models
 - ii. Mapping Data to a New Space: Wavelet Transformation
 - iii. Data Cube aggregation
 - iv. Data Compression
 - v. Histogram analysis
 - vi. Clustering
 - vii. Sampling: Sampling without Replacement, Stratified Sampling
- **Data Transformation and Data Discretization**
 - **Data Transformation: Normalization**
 - **Data Discretization Methods**
 - i. Binning
 - ii. Cluster Analysis
 - iii. Discretization Using Class Labels: Entropy-Based Discretization
 - iv. Discretization Without Using Class Labels: Interval Merge by \hat{A}^2 Analysis
 - **Concept Hierarchy and Its Formation**
 - i. Concept Hierarchy Generation for Numerical Data
 - ii. Concept Hierarchy Generation for Categorical Data
 - iii. Automatic Concept Hierarchy Generation

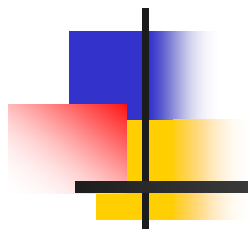


References

- E. Rahm and H. H. Do. Data Cleaning: Problems and Current Approaches. *IEEE Bulletin of the Technical Committee on Data Engineering*. Vol.23, No.4
- D. P. Ballou and G. K. Tayi. Enhancing data quality in data warehouse environments. *Communications of ACM*, 42:73-78, 1999.
- H.V. Jagadish et al., Special Issue on Data Reduction Techniques. *Bulletin of the Technical Committee on Data Engineering*, 20(4), December 1997.
- A. Maydanchik, Challenges of Efficient Data Cleansing (DM Review - Data Quality resource portal)
- D. Pyle. *Data Preparation for Data Mining*. Morgan Kaufmann, 1999.
- D. Quass. A Framework for research in Data Cleaning. (Draft 1999)
- V. Raman and J. Hellerstein. Potters Wheel: An Interactive Framework for Data Cleaning and Transformation, VLDB'2001.
- T. Redman. *Data Quality: Management and Technology*. Bantam Books, New York, 1992.
- Y. Wand and R. Wang. Anchoring data quality dimensions ontological foundations. *Communications of ACM*, 39:86-95, 1996.
- R. Wang, V. Storey, and C. Firth. A framework for analysis of data quality research. *IEEE Trans. Knowledge and Data Engineering*, 7:623-640, 1995.
- <http://www.cs.ucla.edu/classes/spring01/cs240b/notes/data-integration1.pdf>

第3章

数据挖掘的数据仓库与 OLAP技术



第3章: 数据挖掘的数据仓库与OLAP技术

- 什么是数据仓库?
- 多维数据模型
- 数据仓库结构
- 数据仓库实现
- 数据立方体的进一步发展
- 从数据仓库到数据挖掘

什么是数据仓库？

- 有不同的方法定义,但不是严格的.
 - 是一个决策支持数据库,它与组织机构的操作数据库分别维护
 - 数据仓库系统允许将各种应用系统集成在一起,为统一的历史数据分析提供坚实的平台,支持信息处理.
- **W. H. Inmon的定义:** 数据仓库是 面向主题的(subject-oriented), 集成的(integrated), 时变的(time-variant), 和 非易失的(nonvolatile) 数据集合,支持管理决策过程
- 建立数据仓库(Data warehousing):
 - 构造和使用数据仓库的过程

数据仓库—面向主题的

- 围绕重要的主题(如顾客、产品、销售等)组织.
- 关注决策制定者的数据建模与分析,而不是日常的操作和事务处理.
- 数据仓库排除对于决策过程无用的数据,提供特定主题的简明视图.

数据仓库— 集成的

- 通过将多个异种的数据源集成在一起, 而构造
 - 比如, 关系数据库, 一般文件, 联机事务记录
- 使用数据清理和数据集成技术.
 - 确保命名约定, 编码结构, 属性度量等的一致性
 - 例如, 饭店价格: 货币种类, 税, 是否含早餐, 等.
 - 当数据装入数据仓库时, 数据将被转换.

数据仓库— 时变的

- 数据仓库的时间跨度显著地比操作数据库长.
 - 操作数据库数据: 当前值数据.
 - 数据仓库数据: 从历史的角度提供数据 (例如, 过去 **5-10** 年)
- 数据仓库中的每个键结构
 - 显式或隐式地包含时间元素,
 - 但是, 操作数据的键可能包含, 也可能不包含“时间元素”.

数据仓库—非易失的

- 从操作环境转换过来的数据物理地分离存放.
- 数据的更新不在数据仓库环境中出现.
 - 不需要事务处理, 恢复, 和并发控制机制
 - 只需要两种数据存取操作:
 - *数据的初始化装入* 和 *数据访问*.

数据仓库和异种DBMS

- 传统的异种数据库集成:
 - 在异种数据库上建立一个包装程序(wrappers)或中介程序(/mediators)
 - 查询驱动的方法
 - 当查询提交给一个站点时, 使用元数据词典将查询转换成所涉及的异构站点上的相应查询, 查询的结果被集成成为一个全局回答的集合
 - 需要: 复杂的信息过滤, 对资源的竞争
- 数据仓库: 更新驱动的, 高性能
 - 来自异种信息源的数据被预先集成并存储在数据仓库中, 直接用于查询和分析

数据仓库VS.操作数据库

- **OLTP (on-line transaction processing, 联机事务处理)**
 - 传统关系 DBMS 的主要任务
 - 涵盖日常操作: 购买, 库存, 银行, 制造, 工资单, 注册, 记帐, 等.
- **OLAP (on-line analytical processing, 联机分析处理)**
 - 数据仓库系统的主要任务
 - 数据分析和决策制定上提供服务
- **不同的特点 (OLTP vs. OLAP):**
 - 用户和系统的面向性: 顾客 vs. 市场
 - 数据内容: 当前的, 细节的 vs. 历史的, 合并的
 - 数据库设计: **ER** + 应用 vs. 星型 + 主题
 - 视图: 当前的, 局部的 vs. 进化的, 集成的
 - 访问模式: 更新 vs. 只读的, 但是复杂的查询

OLTP vs. OLAP

	OLTP	OLAP
用户	办事员, IT 从业人员	知识工人
功能	日常操作	决策支持
DB 设计	面向应用	面向主题
数据	当前的, 最新的, 细节的, 展平的关系的, 孤立的	历史的, 汇总的, 多维的, 集成的, 加固的
用法	重复	特殊的
访问	读/写 在主键上索引/散列	大量扫描
工作单位	短的, 简单的事务	复杂的查询
访问的记录量	数以十计	数百万
用户数	数千	数百
数据库大小	100MB-GB	100GB-TB
度量	事务吞吐量	查询吞吐量, 响应时间

为什么建立分离的数据仓库？

- 为了两个系统的高性能
 - **DBMS**— 目的是 **OLTP**: 存取方法, 索引, 并发控制, 恢复
 - 数据仓库—目的是 **OLAP**: 复杂的 **OLAP** 查询, 多维视图, 统一.
- 不同的功能和不同的数据:
 - 缺少数据: 决策支持需要历史数据, 通常操作数据库并不维护这些数据
 - 数据统一: 决策支持需要将来自异种数据源的数据统一 (聚集, 汇总)
 - 数据质量: 不同的数据源通常使用不同的数据表示, 编码, 和应当遵循的格式

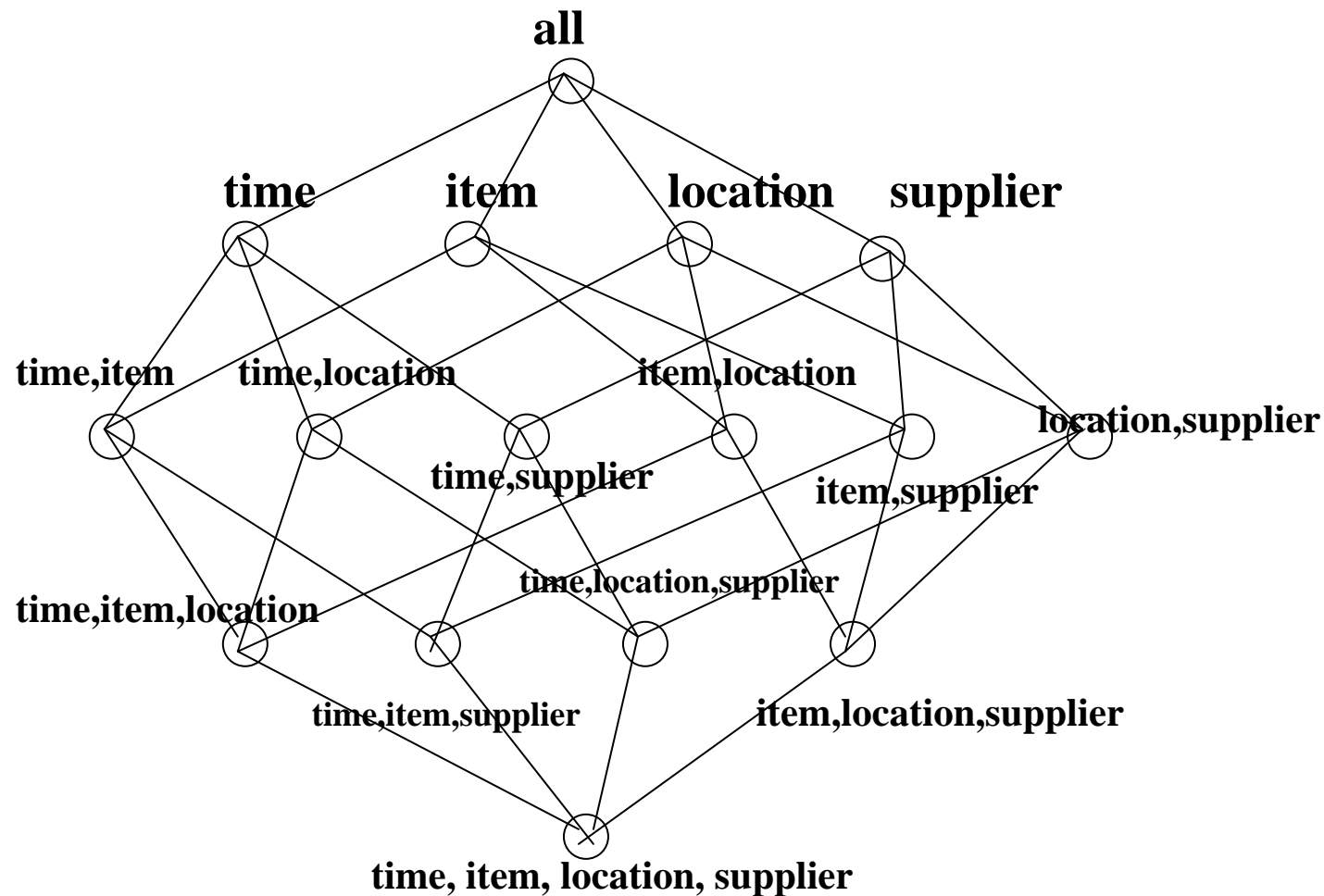
第2章: 数据挖掘的数据仓库与OLAP技术

- 什么是数据仓库?
- 多维数据模型
- 数据仓库结构
- 数据仓库实现
- 从数据仓库到数据挖掘
- 数据立方体的进一步发展

由表和电子数据表到数据方

- 数据仓库基于 多维数据模型，多维数据模型将数据视为数据方(**data cube**)形式
- 数据方(如**sales**) 可以将数据建模, 并允许由多个维进行观察
 - 维表, 如 **item (item_name, brand, type)**, 或 **time(day, week, month, quarter, year)**
 - 事实表包含度量 (如 **dollars_sold**) 和每个相关维表的键
- 在数据仓库的文献中, 一个 **n-D** 基本立方体 称作基本方体 (**base cuboid**). 最顶部的 **0-D**方体存放最高层的汇总, 称作顶点方体(**apex cuboid**). 方体的格形成数据方.

立方体: 方体的格



0-D(顶点) 方体

1-D 方体

2-D方体

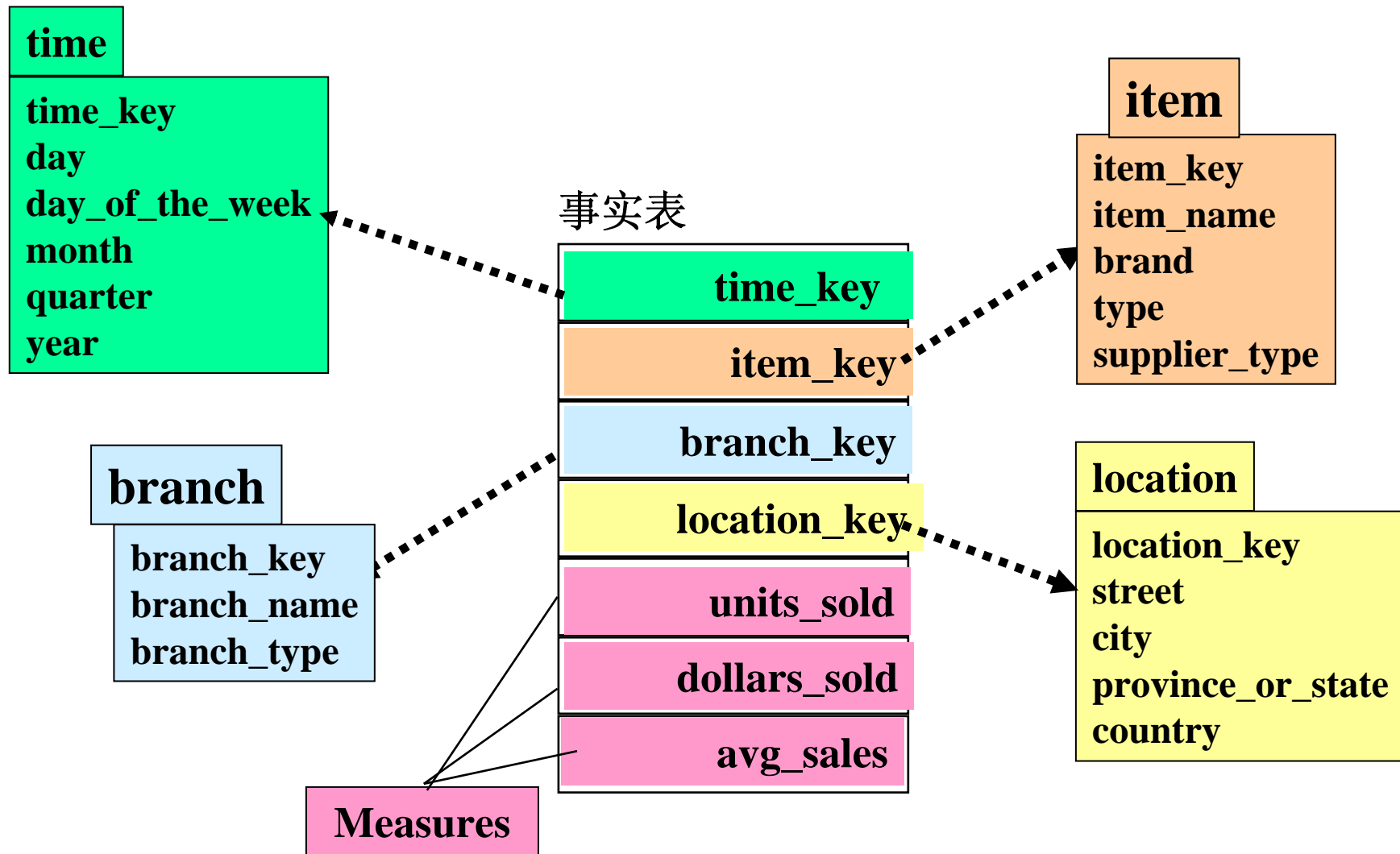
3-D方体

4-D(基本)方体

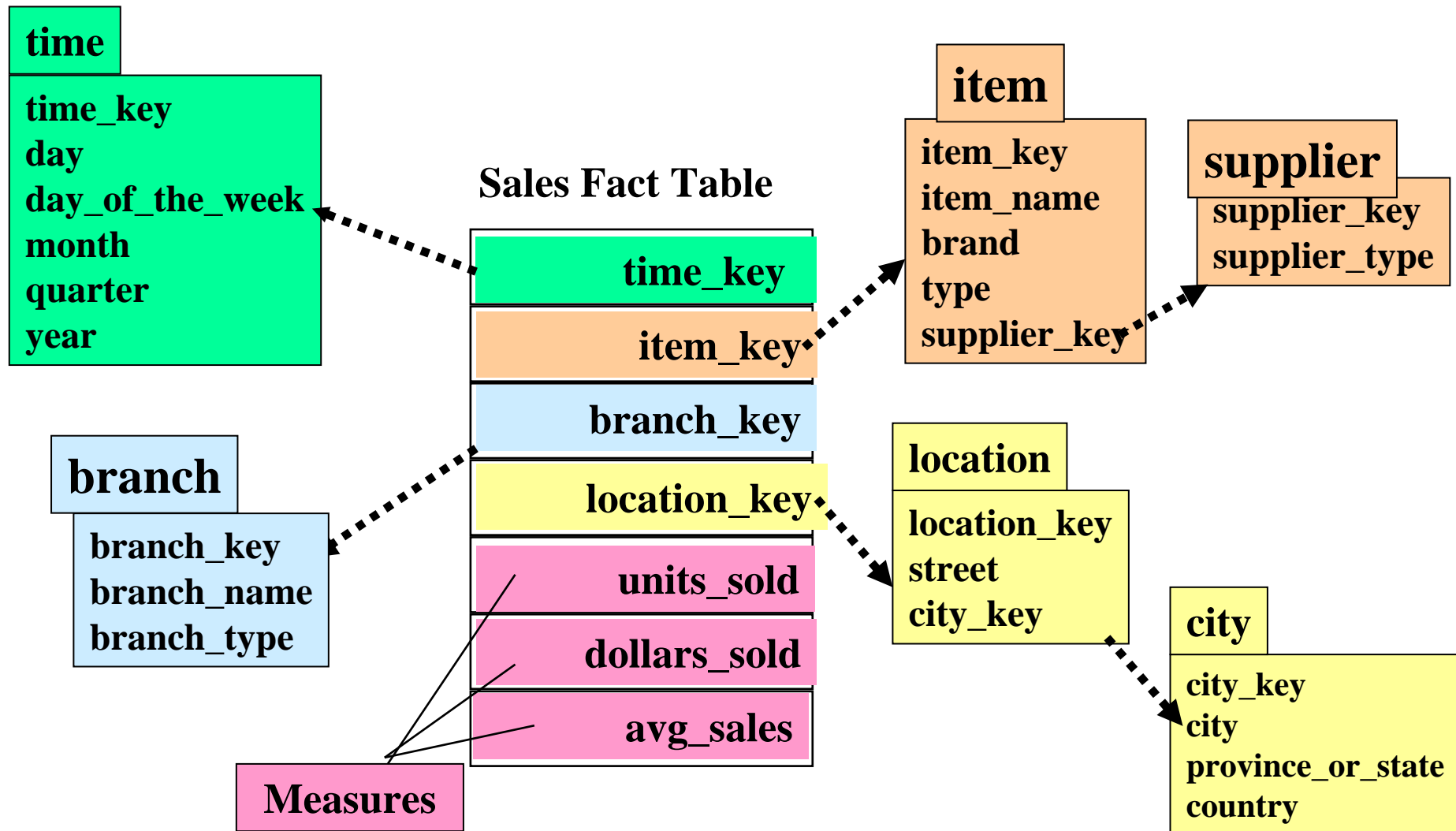
数据仓库的概念建模

- 数据仓库建模: 多维模型, 涉及维和度量
 - 星型模式: 事实表在中央, 连接一组维表
 - 雪花模式: 星型模式的精炼, 其中一些维分层结构被规范化成一组较小的维表, 形成类似于雪花的形状, 减少冗余
 - 事实星座: 多个事实表共享维表, 可以看作星星的集合, 因此称作星系模式, 或事实星座

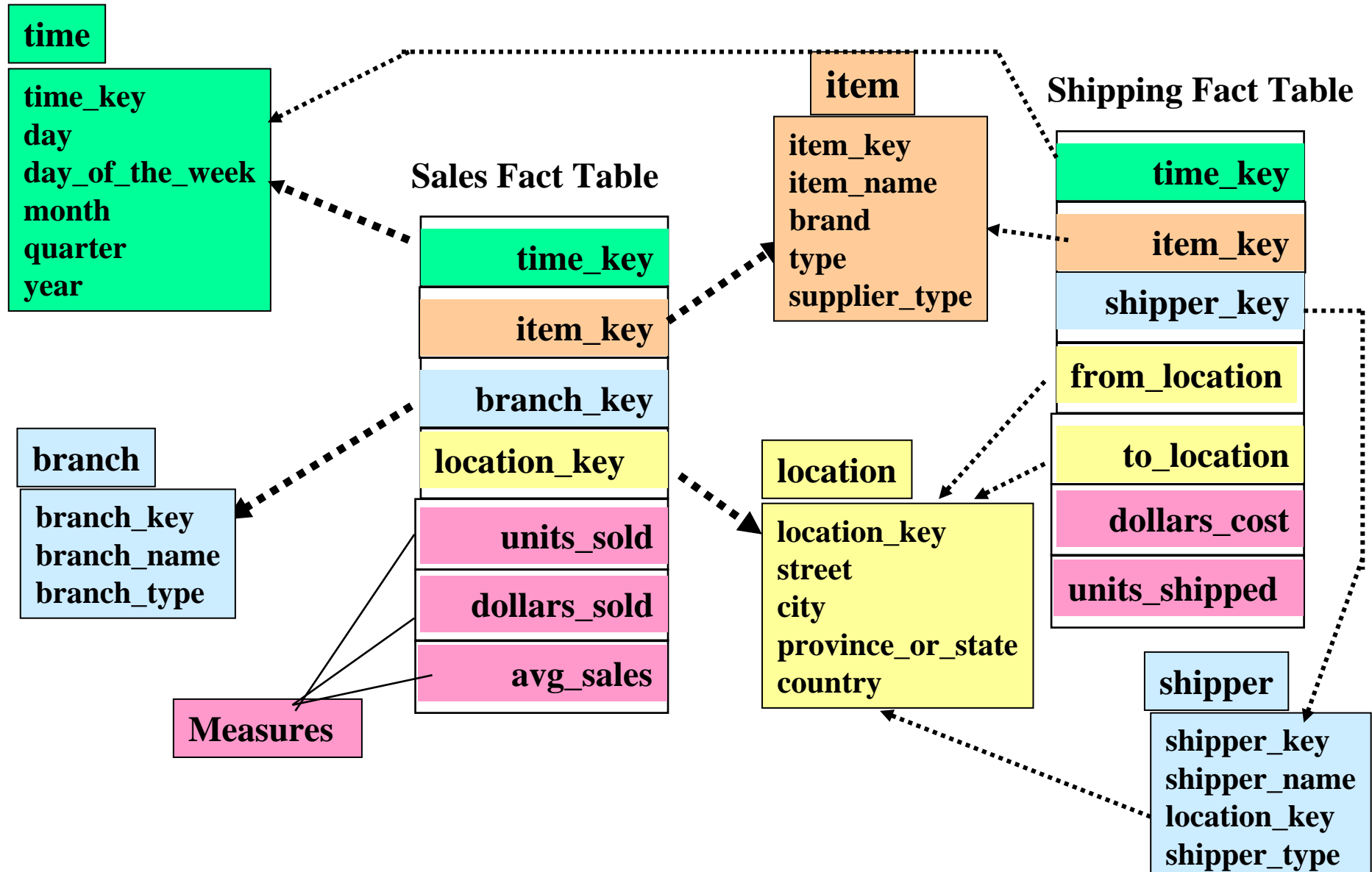
星型模式的例子



雪花模式的例子



事实星座的例子



数据挖掘查询语言 DMQL: 语言原语

- 立方体定义 (事实表)

define cube <cube_name> [<dimension_list>]: <measure_list>

- 维定义 (维表)

define dimension <dimension_name> **as**
(<attribute_or_subdimension_list>)

- 特殊情况 (共享维表)

- 第一次, 如 “cube definition”

- **define dimension** <dimension_name> **as**
<dimension_name_first_time> **in cube** <cube_name_first_time>

用DMQL定义星型模式

```
define cube sales_star [time, item, branch, location]:  
    dollars_sold = sum(sales_in_dollars), avg_sales =  
        avg(sales_in_dollars), units_sold = count(*)  
define dimension time as (time_key, day, day_of_week, month,  
    quarter, year)  
define dimension item as (item_key, item_name, brand, type,  
    supplier_type)  
define dimension branch as (branch_key, branch_name,  
    branch_type)  
define dimension location as (location_key, street, city,  
    province_or_state, country)
```

用DMQL定义雪花模式

define cube sales_snowflake [time, item, branch, location]:

 dollars_sold = sum(sales_in_dollars), avg_sales =
 avg(sales_in_dollars), units_sold = count(*)

define dimension time **as** (time_key, day, day_of_week, month, quarter,
 year)

define dimension item **as** (item_key, item_name, brand, type,
 supplier(supplier_key, supplier_type))

define dimension branch **as** (branch_key, branch_name, branch_type)

define dimension location **as** (location_key, street, city(city_key,
 province_or_state, country))

用DMQL定义事实星座

define cube sales [time, item, branch, location]:

 dollars_sold = sum(sales_in_dollars), avg_sales = avg(sales_in_dollars),
 units_sold = count(*)

define dimension time **as** (time_key, day, day_of_week, month, quarter, year)

define dimension item **as** (item_key, item_name, brand, type, supplier_type)

define dimension branch **as** (branch_key, branch_name, branch_type)

define dimension location **as** (location_key, street, city, province_or_state, country)

define cube shipping [time, item, shipper, from_location, to_location]:

 dollar_cost = sum(cost_in_dollars), unit_shipped = count(*)

define dimension time **as** time **in cube** sales

define dimension item **as** item **in cube** sales

define dimension shipper **as** (shipper_key, shipper_name, location **as** location **in cube** sales, shipper_type)

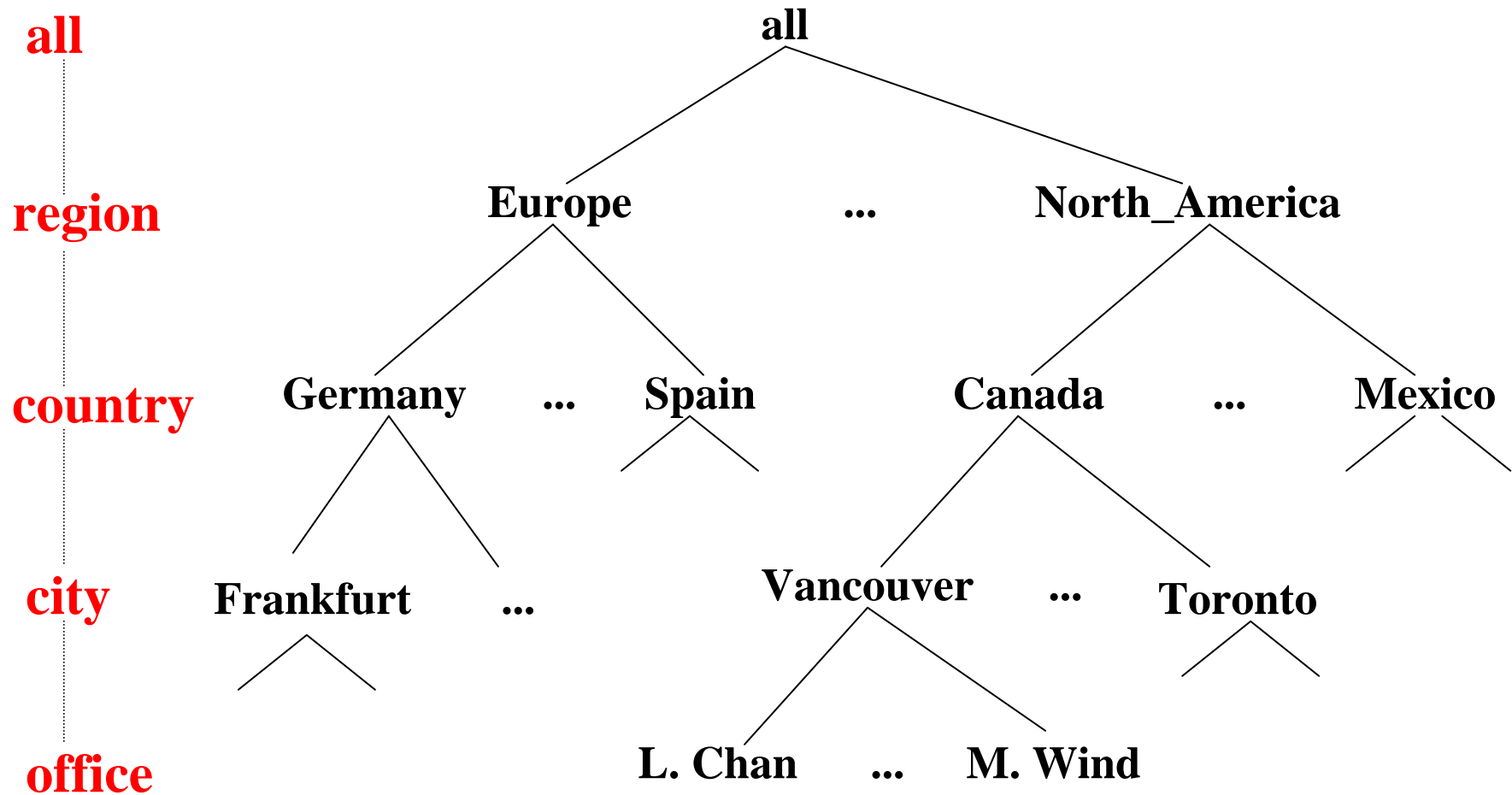
define dimension from_location **as** location **in cube** sales

define dimension to_location **as** location **in cube** sales

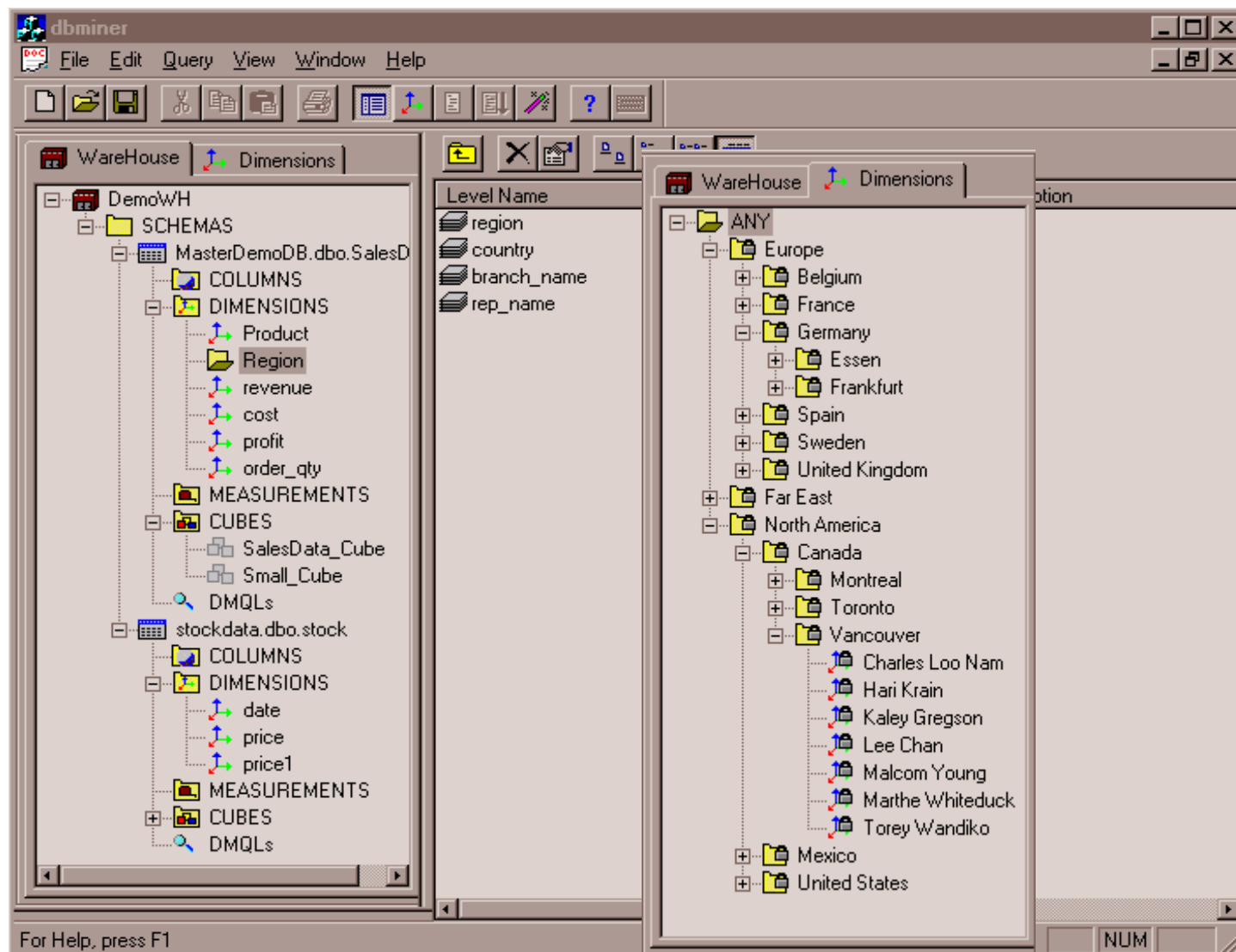
三类度量（数值函数）

- 分布的(distributive): 将数据划分为 n 个集合, 函数在每一部分上的计算得到一个聚集值. 如果将函数用于 n 个聚集值得到的结果, 与将函数用于所有数据得到的结果一样, 则该函数可以用分布方式计算.
 - 例, `count()`, `sum()`, `min()`, `max()`.
- 代数的(algebraic): 如果它能够由一个具有 M (其中, M 是一个整数界)个参数的代数函数计算, 而每个参数都可以用一个分布聚集函数求得.
 - 例, `avg()`, `min_N()`, `standard_deviation()`.
- 整体的(holistic): 如果描述它的子聚集所需的存储没有一个常数界.
 - 例, `median()`, `mode()`, `rank()`.

一个概念分层: 维Location

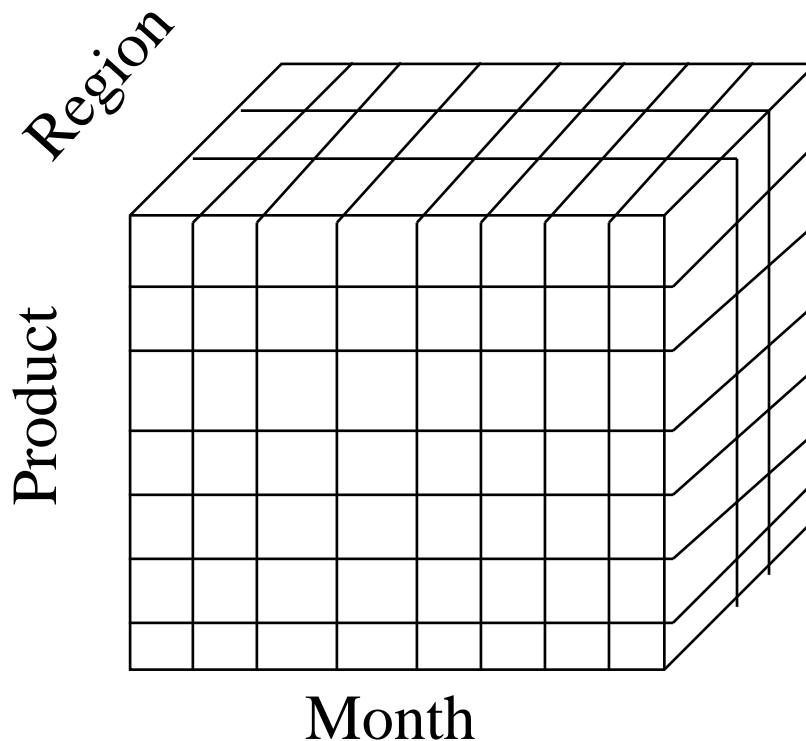


数据仓库和分层结构视图

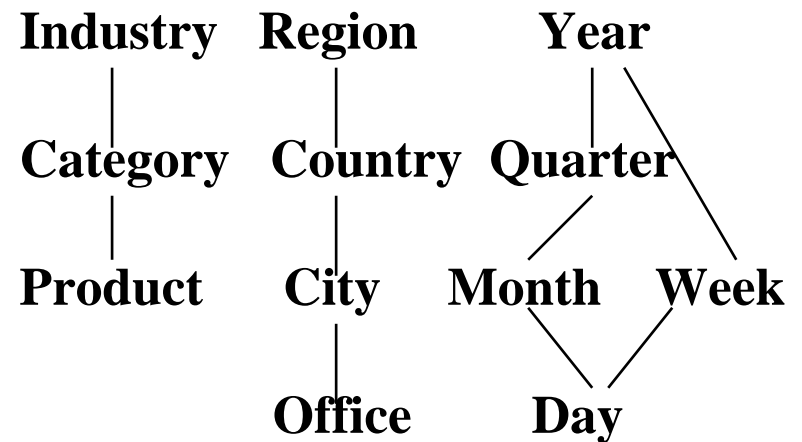


多维数据

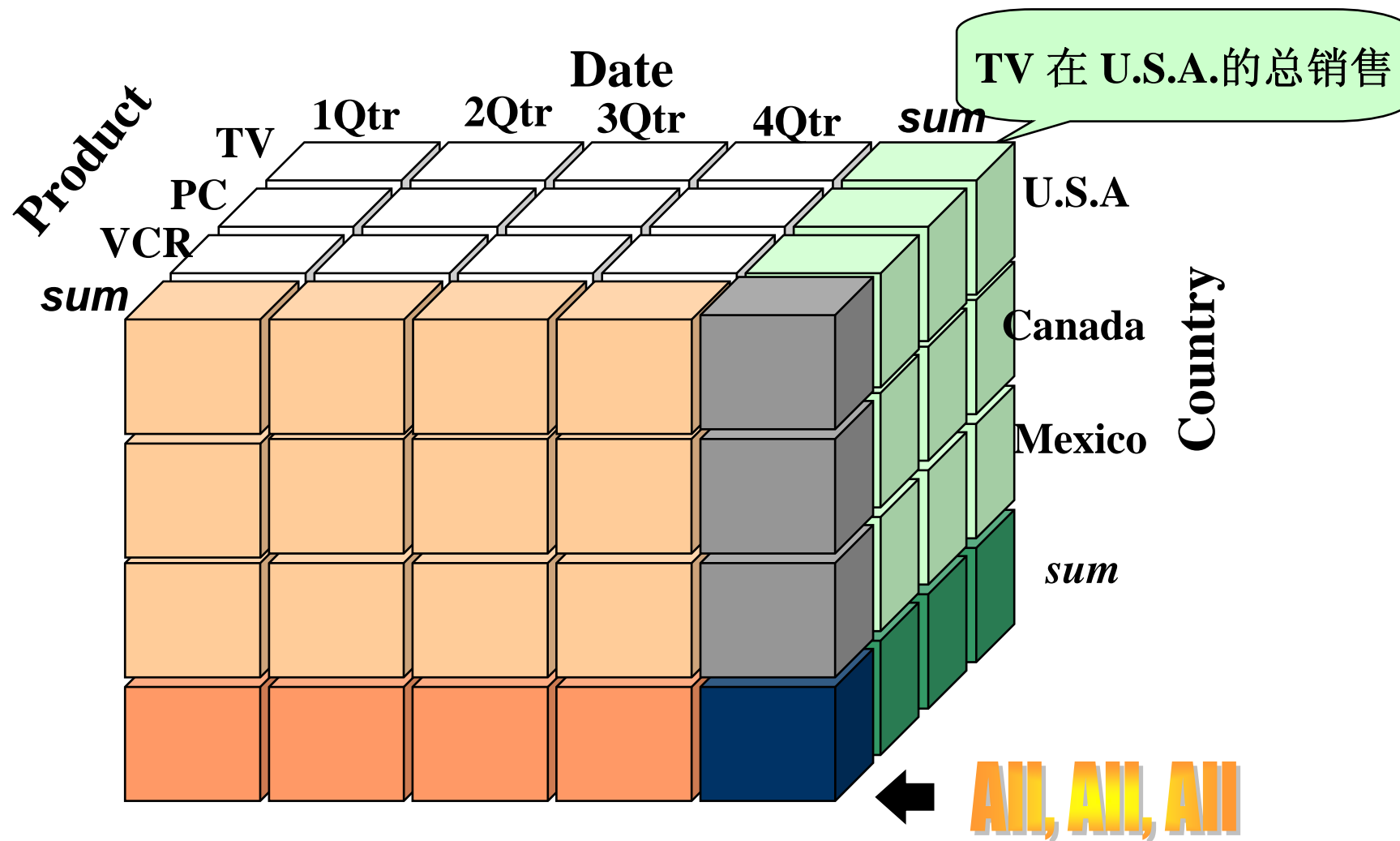
- 多维模型中，数据组织成多维，每维包含由概念分层定义的多个抽象层
- 销售量作为 **product, month, 和 region**的函数



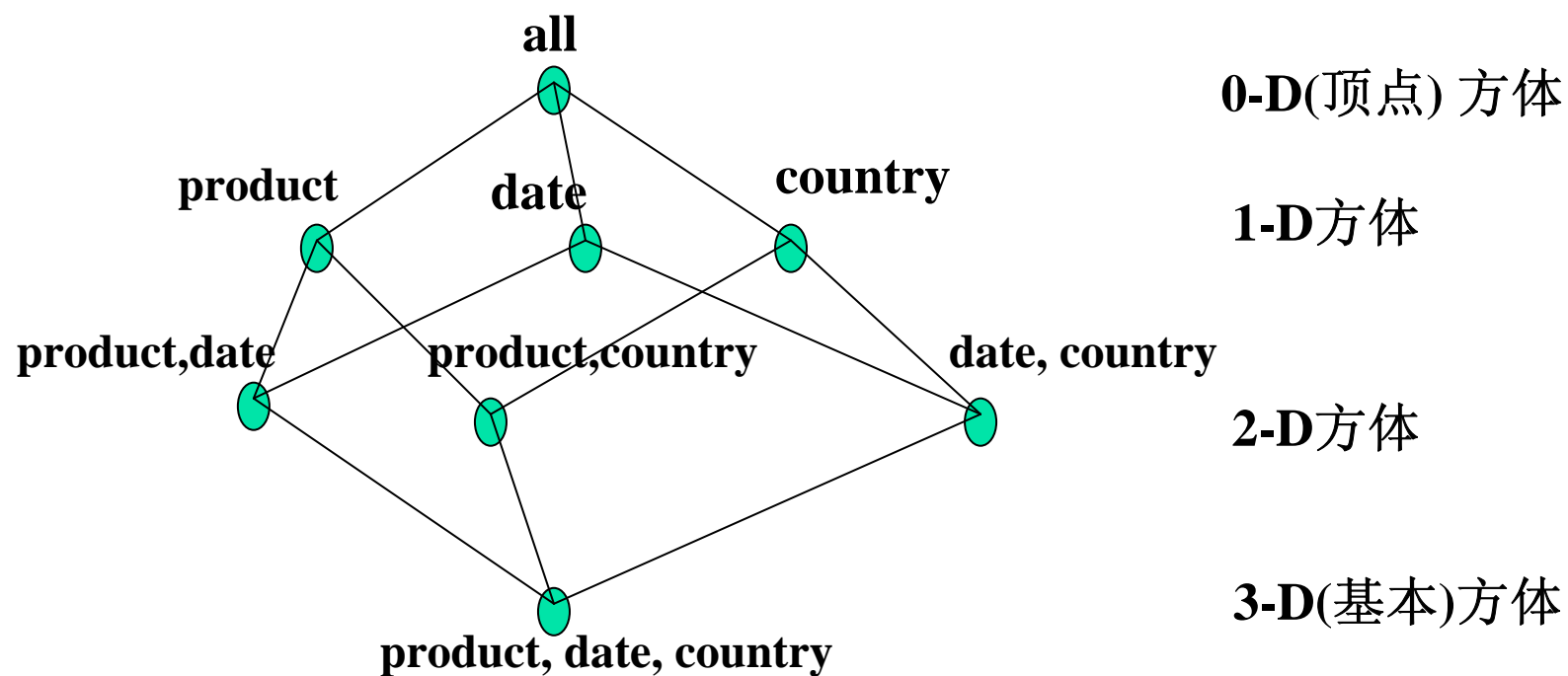
维: **Product, Location, Time**
的分层结构



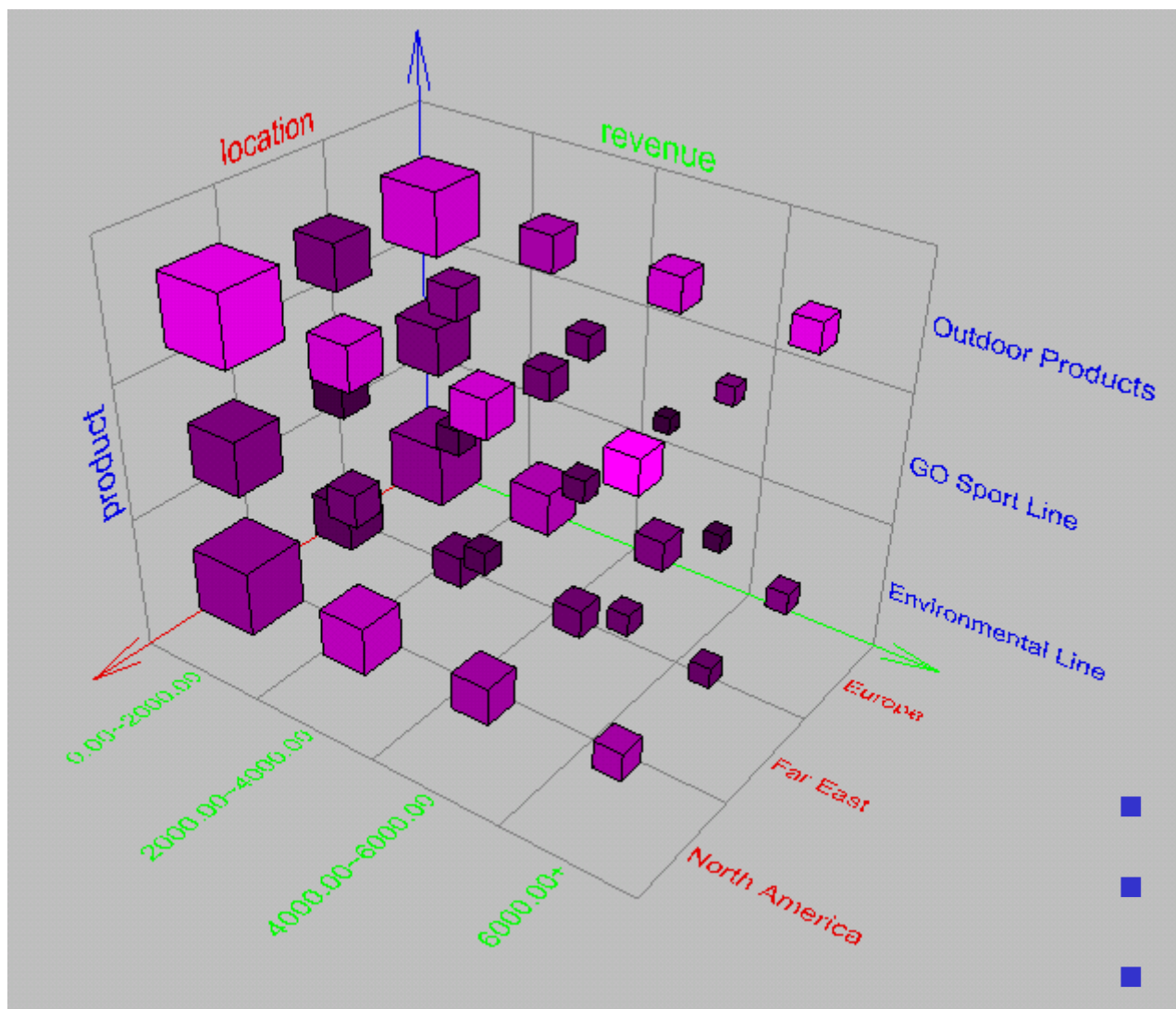
一个数据方的样本



对应于数据方的方体



浏览数据方



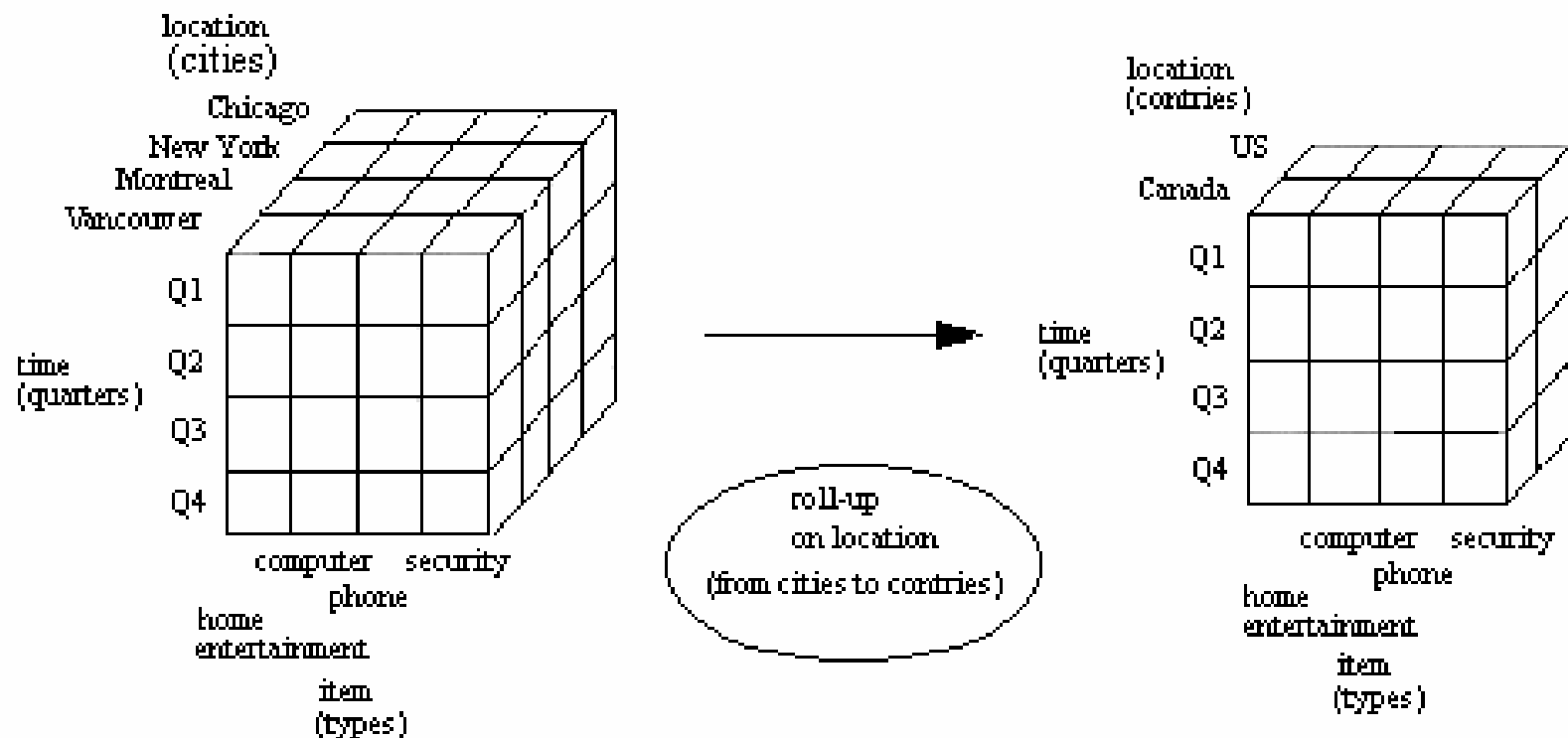
- 可视化
- OLAP 的能力
- 交互式操作

典型的OLAP操作

- 上卷(Roll up)/上钻 (drill-up): 汇总数据
- 下钻(Drill down)/下卷 (roll down): 上卷的逆操作
- 切片(Slice)和切块 :
 - 投影和选择
- 转轴(Pivot)/旋转 (rotate):
 - 调整数据方, 目视操作, 3D 到 2D 平面.
- 其它操作
 - 钻过(drill across): 涉及多个事实表
 - 钻透(drill through): 通过数据方的最底层, 到它背后的关系表 (使用 SQL)

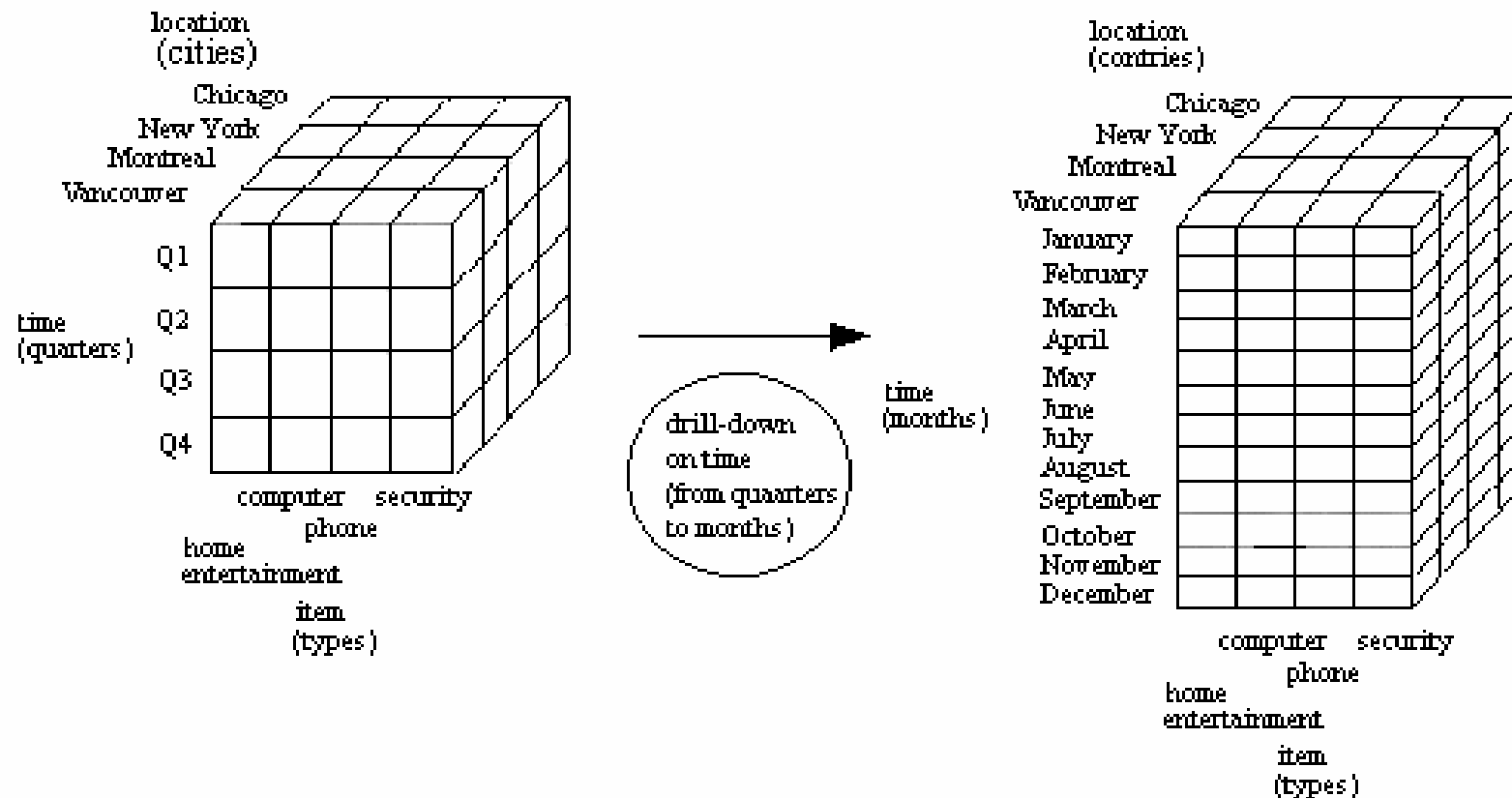
OLAP 操作: 上卷

- 上卷(Roll up)/上钻 (drill-up): 汇总数据
 - 通过沿概念分层攀升或通过维归约
- 在 **location** 上卷(由 **cities** 到 **countries**)



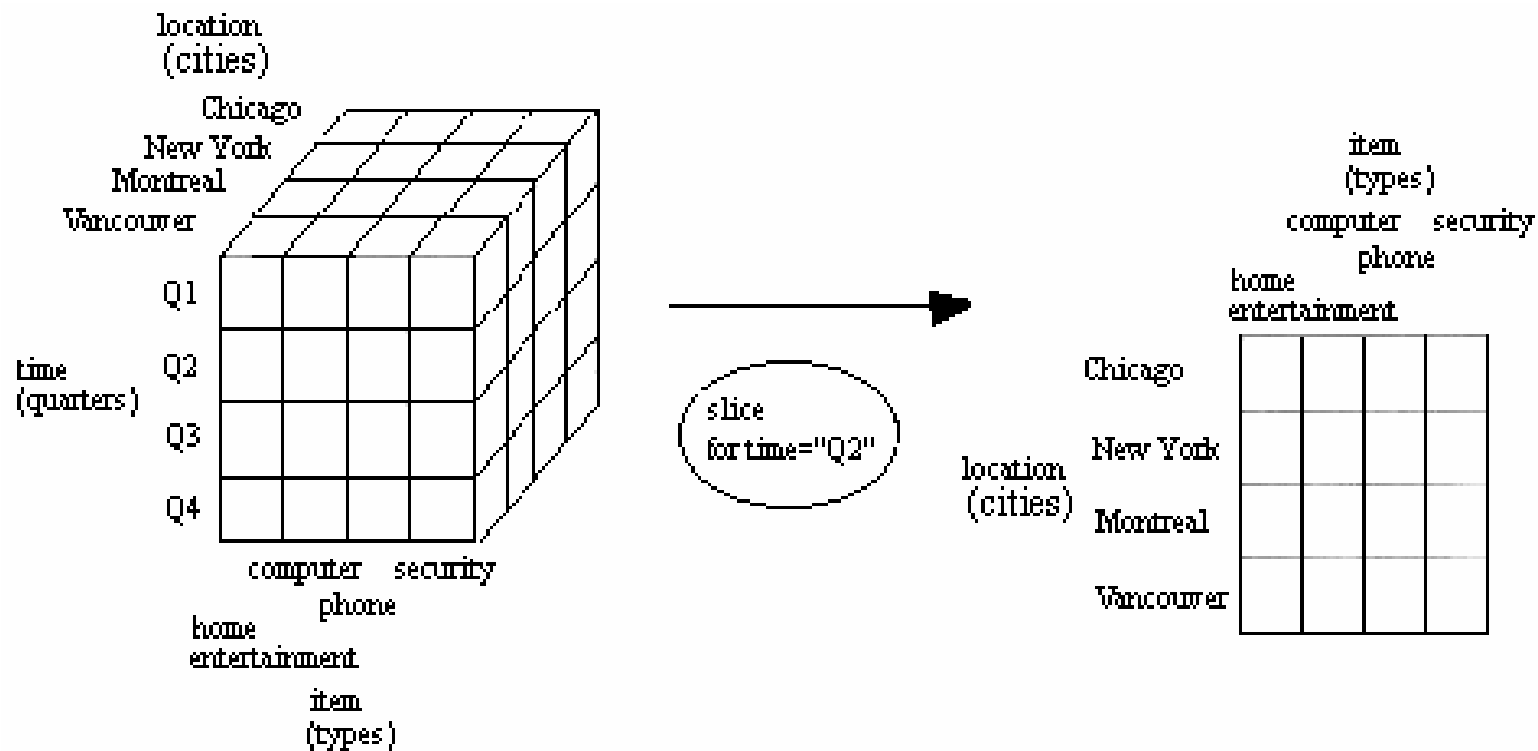
OLAP 操作: 下钻

- 下钻(Drill down)/下卷 (roll down): 上卷的逆操作
 - 由较高层的汇总到较低层的汇总或详细数据, 或者引进新的维
- 在 **time** 下钻 (由 **quarters** 到 **months**)



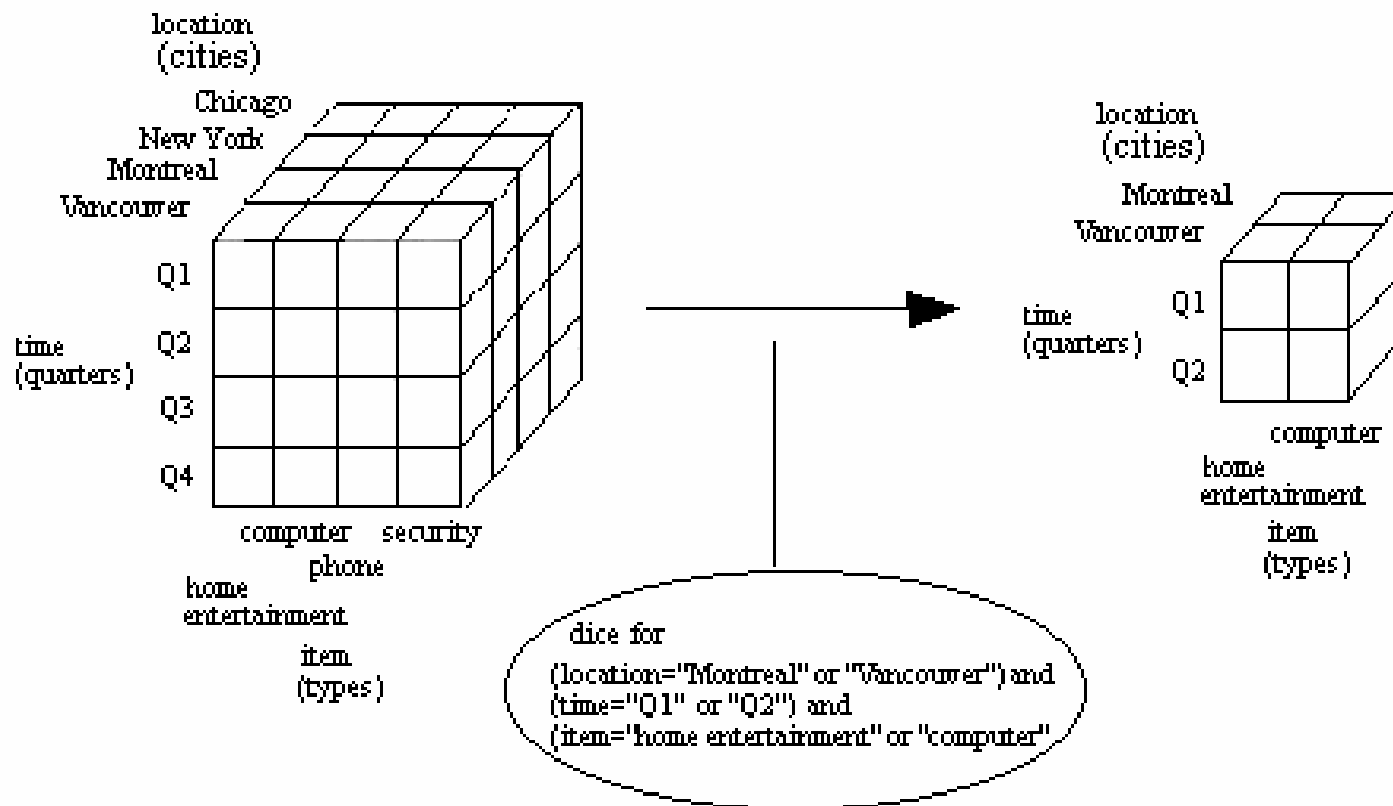
OLAP 操作:切片

- 切片(Slice):
 - 投影和选择, 对一个维进行选择, 导致子方体
- 切片条件: **time="Q2"**



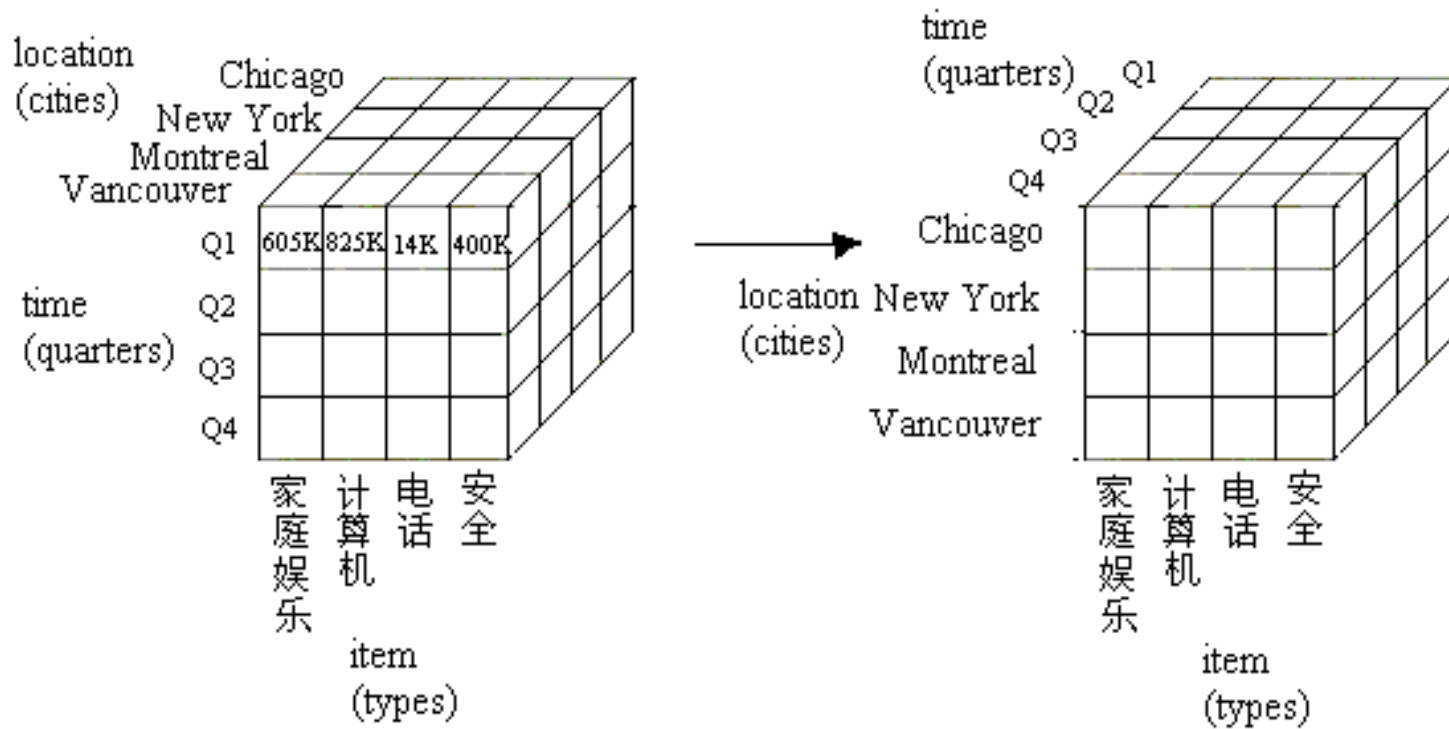
OLAP 操作: 切块

- 切块 : 对两个或多个维执行选择, 导致子方体
- 切块条件: (*location*="Montreal" or "Vancouver") and (*time*="Q1" or "Q2") and (*item*="home entertainment" or "computer")



OLAP 操作: 转轴

- 转轴(Pivot)/旋转 (rotate):
 - 调整数据方, 可视化操作, 提供数据的替代表示.



其他操作

■ 其它操作

- 钻过(drill across): 涉及多个事实表
- 钻透(drill through): 通过数据方的最底层, 到它背后的关系表 (使用 SQL)
- 统计计算
 - 比率、方差; 增长率
- 分析建模, 等

第3章: 数据挖掘的数据仓库与OLAP技术

- 什么是数据仓库?
- 多维数据模型
- 数据仓库结构
- 数据仓库实现
- 从数据仓库到数据挖掘
- 数据立方体的进一步发展

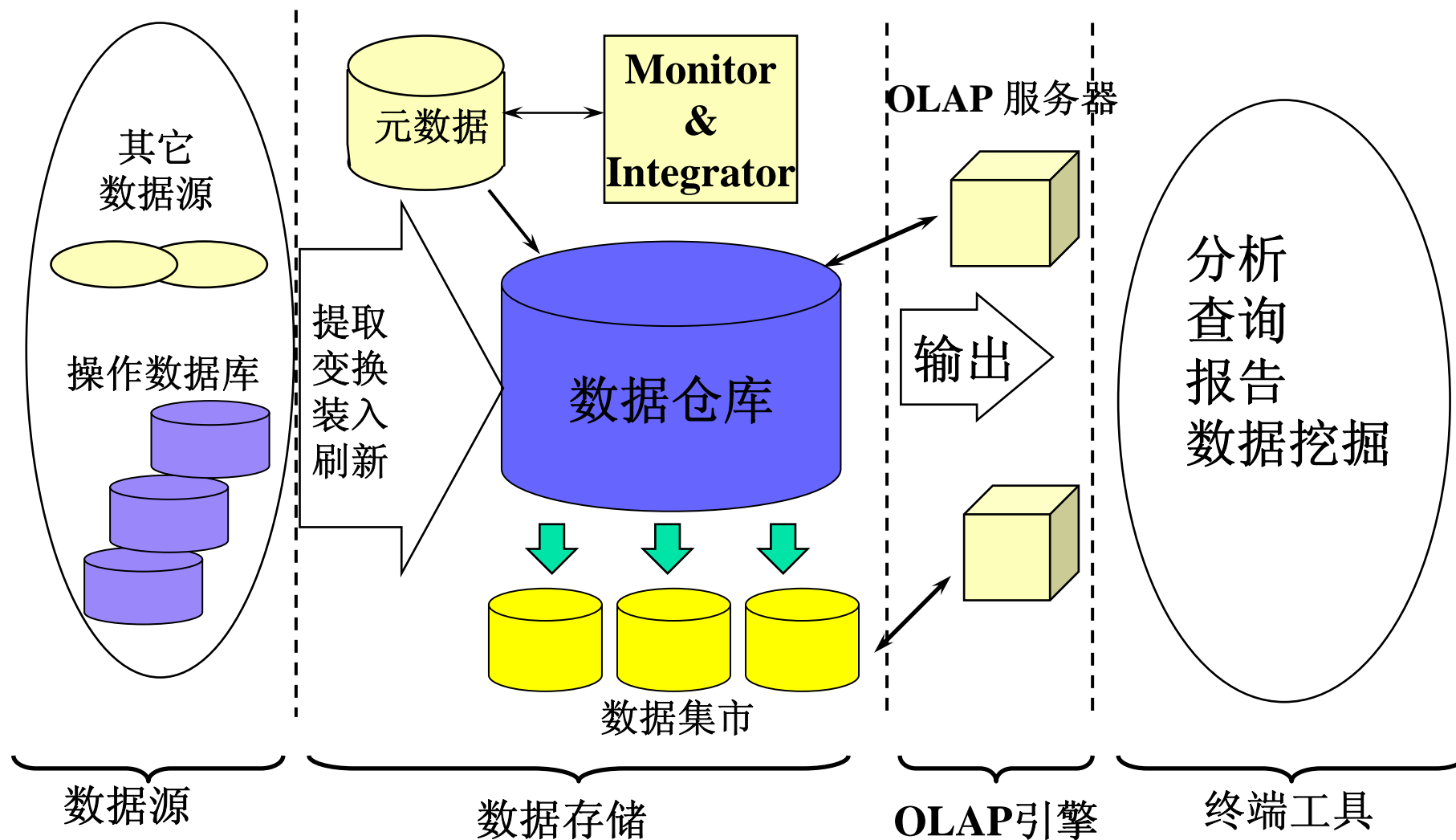
数据仓库设计

- 数据仓库设计中，必须考虑四种视图
 - 自顶向下视图
 - 选择数据仓库所需的有关信息
 - 数据源视图
 - 揭示（操作）数据库系统捕获、存储、和管理的信息
 - 数据仓库视图
 - 由事实表和维表组成
 - 商务查询视图
 - 从最终用户的角度透视数据仓库中的数据

数据仓库设计过程

- 自顶向下, 自底向上方法或二者的结合
 - 自顶向下: 由总体设计和规划开始 (成熟)
 - 自底向上: 由实验和原型开始 (快速)
- 软件工程的观点
 - 瀑布式: 在进行下一步之前, 每一步都进行结构化和系统的分析
 - 螺旋式: 功能渐增的系统的快速产生, 相继版本之间的间隔很短, 快速转向
- 典型的数据仓库设计过程
 - 选取待建模的商务处理, 例如, 订单, 发票, 库存等.
 - 选取商务处理的粒度 (原子层数据), 例如, 单个事务、一天的快照等
 - 选取用于每个事实表记录的维, 如, 时间、商品、顾客、供应商、仓库、事务类型和状态 等
 - 选取将安放在事实表中的度量. 典型的度量是可加的数值量, 如 *dollars_sold* 和 *units_sold*

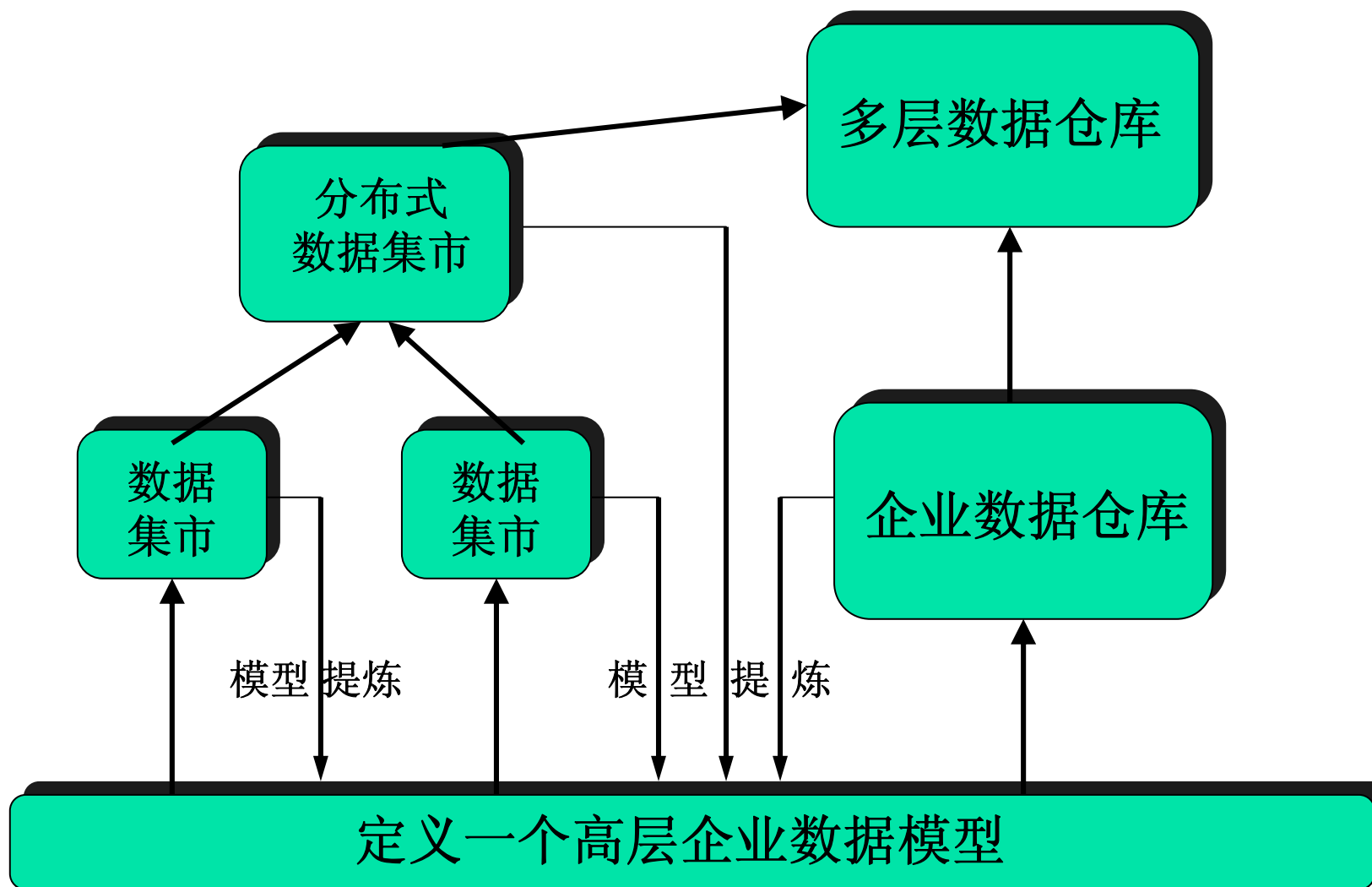
多层结构



三层数据仓库模型

- 企业仓库
 - 搜集了关于主题的所有信息, 跨越整个组织
- 数据集市
 - 数据集市包含企业范围数据的一个子集, 对于特定的用户是有用的. 其范围限于选定的主题, 如销售数据
 - 独立的 vs. 依赖的 (直接来自数据仓库) 数据集市
- 虚拟仓库
 - 操作数据库上视图的集合
 - 只有部分可能的汇总视图被物化

数据仓库开发：一种推荐的方法



OLAP 服务器结构

■ 关系OLAP (ROLAP)

- 使用关系或扩充关系的 **DBMS** 存放和管理仓库数据, 使用**OLAP**中间件支持其它部分
- 包含一个优化的 **DBMS** 后端, 聚集导航逻辑的实现, 以及附加的工具和服务
- 较大的可伸缩性

■ 多维 OLAP (MOLAP)

- 基于数组的多维存储引擎 (稀疏矩阵技术)
- 对预计算的汇总数据快速索引

■ 混合 OLAP (HOLAP)

- 弹性, 底层: 关系的, 高层: 数组.

■ 专门的 **SQL 服务器**

- 对星型/雪花型模式上的**SQL**查询提供特殊的支持

元数据存储

- 元数据是定义数据仓库的数据。有如下类型
 - 描述数据仓库的结构
 - 模式, 视图, 维, 分层结构, 数据源定义, 数据集市的位置和内容
 - 操作元数据
 - 数据血统 (数据变迁历史和转换路径), 数据流通 (主动, 存档, 或净化), 管理信息 (数据仓库使用统计, 错误报告, 审计跟踪)
 - 用于汇总的算法
 - 由操作环境到数据仓库的映射
 - 涉及系统性能的数据
 - 仓库模式, 视图和导出数据定义
 - 商务数据
 - 商务术语和定义, 数据的所有者, 收费政策

数据仓库的后端工具和实用程序

- 数据提取：
 - 由多个异种, 外部数据源收集数据
- 数据清理：
 - 检测数据中的错误, 可能时订正它们
- 数据变换：
 - 将数据由遗产或宿主格式转换成数据仓库格式
- 装载：
 - 排序, 综合, 加固, 计算视图, 检查整体性, 并建立索引和划分
- 刷新
 - 传播由数据源到数据仓库的更新

第2章: 数据挖掘的数据仓库与OLAP技术

- 什么是数据仓库?
- 多维数据模型
- 数据仓库结构
- 数据仓库实现
- 从数据仓库到数据挖掘
- 数据立方体的进一步发展

数据方的有效计算

- 数据方可以视为方体的格

- 最下面的方体是基本方体
- 最上面的 (顶点) 方体只包含一个单元
- 具有L层的n-D数据方包含多少个方体?

- 其中Li是与维i相关联的层数

$$T = \prod_{i=1}^n (L_i + 1)$$

- 数据方的物化(Materialization)

- 物化每一个方体 (全物化), 不物化任何方体(不物化), 或物化某些方体(部分物化)
- 物化方体的选择
 - 基于大小, 共享, 访问频率, 等.

数据方计算

- 用DMQL定义和计算数据方

define cube sales[item, city, year]: sum(sales_in_dollars)

compute cube sales

- 将它变换成类——SQL语句 (用新的操作 **cube by**扩充, 由Gray 等'96 引进)

SELECT item, city, year, SUM (amount)

FROM SALES

CUBE BY item, city, year

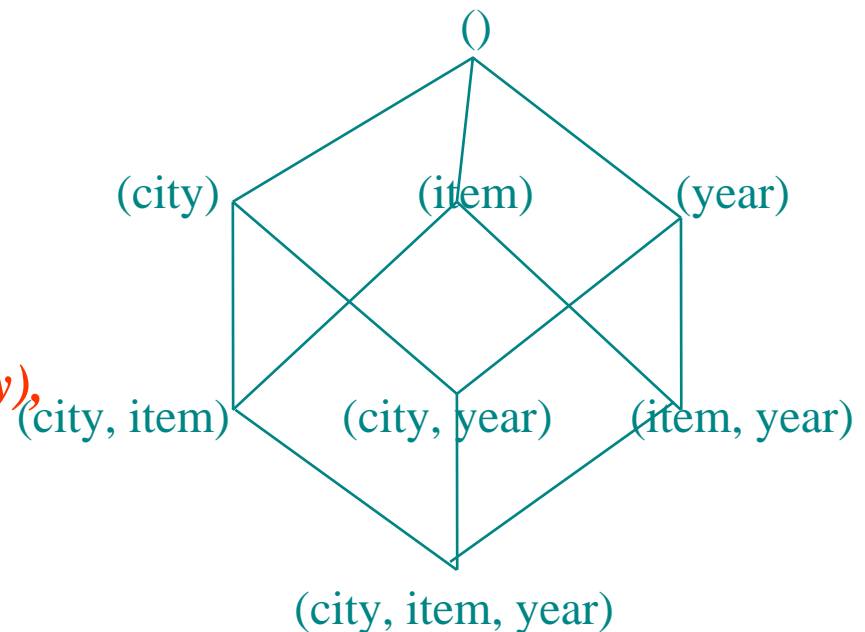
- 需要计算的分组

(city, item, year),

(city,item),(city, year), (item, city),

(city), (item), (year)

()



数据方计算: 基于**ROLAP**的方法(1)

- 有效的方计算方法
 - 基于**ROLAP**的方计算算法 (Agarwal et al'96)
 - 基于数组的方计算算法 (Zhao et al'97)
 - 自底向上的方法 (Beyer & Ramakrishnan'99)
 - 混合的方法 (Han, Pei, Dong & Wang:SIGMOD'01)
- 基于**ROLAP**的方计算算法
 - 排序, 散列, 和分组操作作用于维属性, 以便对相关元组重新排序和分簇
 - 在某些子聚集上分组, 作为“部分分组”
 - 由以前计算的聚集计算新的聚集, 而不必由基本事实表计算

数据方计算: 基于ROLAP的方法(2)

- 取自研究论文
- 基于Hash/排序 的方法 (Agarwal 等. VLDB'96)
 - 最小双亲(Smallest-parent): 由最小的, 先前计算的方体计算方体
 - 存储结果(Cache-results): 存储先前计算的方体, 由它可以计算其它方体, 以减少磁盘I/O
 - 分摊扫描(Amortize-scans): 同时计算尽可能多的方体, 以分摊磁盘的读操作开销
 - 共享排序(Share-sorts): 使用基于排序的方法时, 在多个方体之间共享排序开销
 - 共享划分(Share-partitions): 使用基于hash的方法时, 在多个方体之间共享划分开销

索引|OLAP 数据

- 为了有效的访问，大部分数据仓库系统支持索引结构
- 两种常用的方法对**OLAP**数据进行索引
 - 位图索引 **bitmap indexing**
 - 连接索引 **join indexing**

索引|OLAP 数据: 位图索引

- 在一个特定列上索引
- 列上的每个值是一个位向量: 位操作很快
- 位向量的长度: 基本表的记录数
- 如果数据表中给定行的属性值为 v , 则在位图索引的对应行, 表示该值的位为1, 该行的其它位均为0
- 不适合势(不同值个数)很高的域

基本表

Cust	Region	Type
C1	Asia	Retail
C2	Europe	Dealer
C3	Asia	Dealer
C4	America	Retail
C5	Europe	Dealer

在 Region 上索引

RecID	Asia	Europe	America
1	1	0	0
2	0	1	0
3	1	0	0
4	0	0	1
5	0	1	0

在 Type 上索引

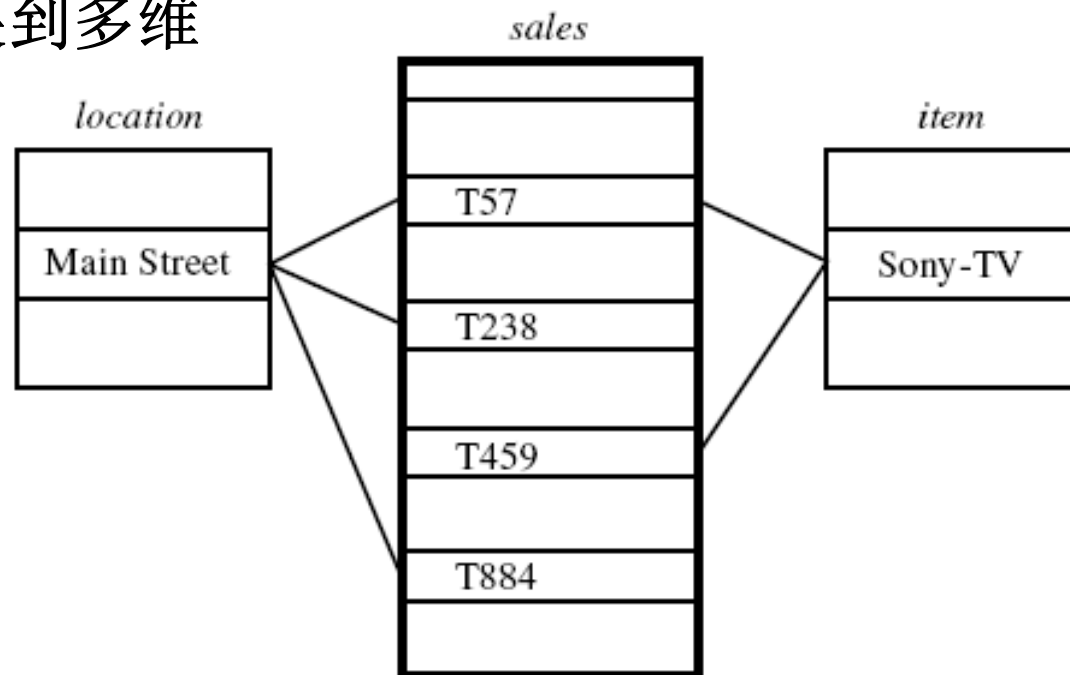
RecID	Retail	Dealer
1	1	0
2	0	1
3	0	1
4	1	0
5	0	1

索引OLAP 数据: 连接索引

- 连接索引: **JI**(**R-id**, **S-id**) , 其中 **R** (**R-id**, ...) $\triangleright \triangleleft$ **S** (**S-id**, ...)
 - 将关系的连接物化在**JI**文件中, 加快了关系连接的速度
- 数据仓库中, 连接索引将星型模式维表的值关联到事实表的行.
 - 例, 事实表*Sales* 和两个维 *city* 和 *product*
 - *city* 上的连接索引对每个不同的城市, 维护一张记录该城市销售的元组的**R**
 - 连接索引可以扩展到多维

Join index table for
item/sales

<i>item</i>	<i>sales_key</i>
...	...
Sony-TV	T57
Sony-TV	T459
...	...



OLAP查询的有效处理

- 物化方体和构造**OLAP**索引结构的目的是加快数据立方体的查询处理速度。
- 查询处理按如下步骤进行：
- 确定哪些操作可以在可用的方体上进行：
 - 将下钻, 上卷等操作变换成对应的SQL和/或**OLAP**操作, 例如, **dice**
= **selection** + **projection**
- 确定相关的操作应当使用哪些物化的方体。

第3章: 数据挖掘的数据仓库与OLAP技术

- 什么是数据仓库?
- 多维数据模型
- 数据仓库结构
- 数据仓库实现
- 从数据仓库到数据挖掘
- 数据立方体的进一步发展

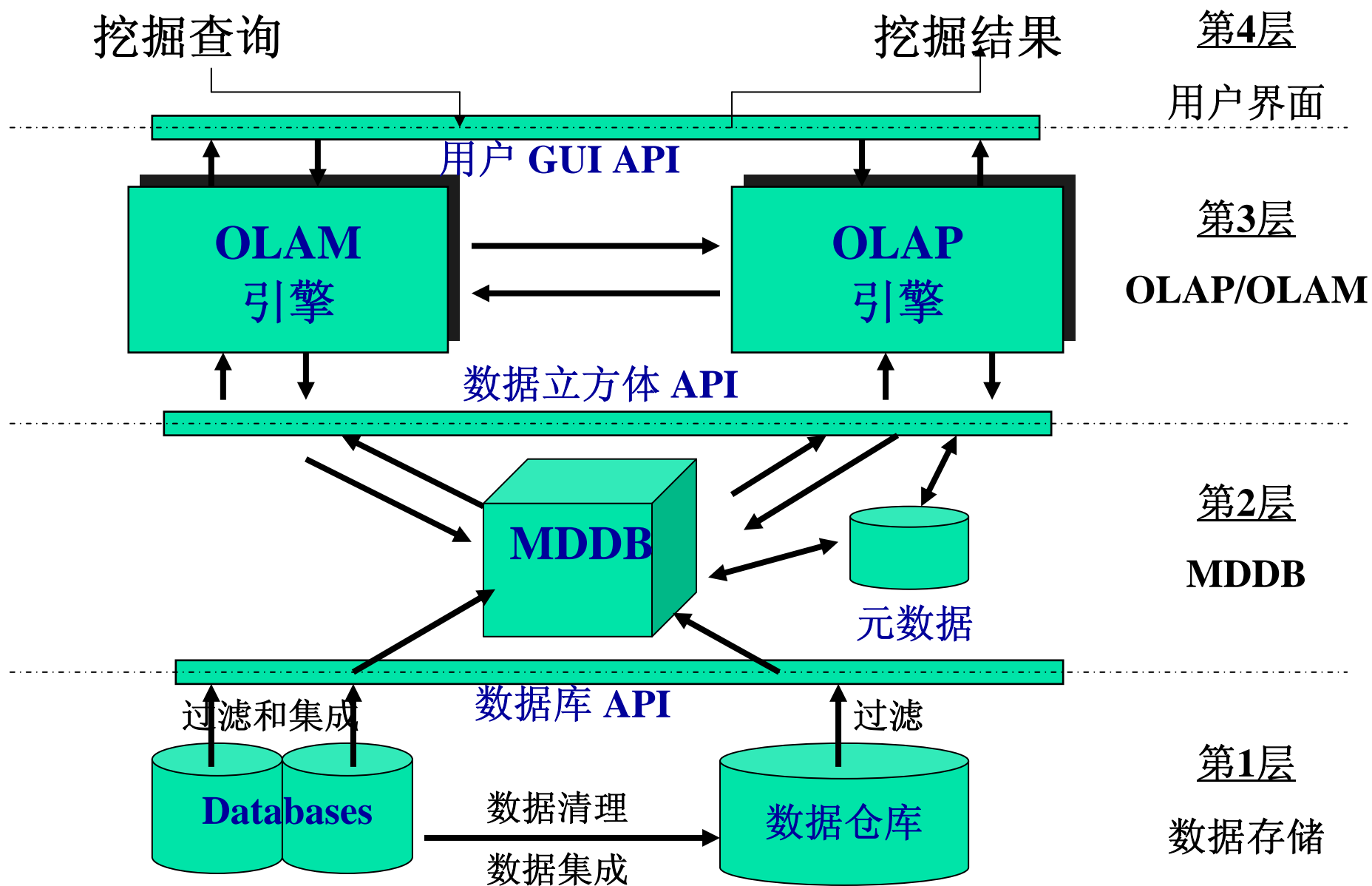
数据仓库使用

- 数据仓库应用的三种类型
 - 信息处理
 - 支持查询, 基本统计分析, 使用交叉表, 表, 图表和图进行报告
 - 分析处理
 - 数据仓库数据的多维分析
 - 支持基本的 **OLAP** 操作, 切片-切块, 上下钻, 转轴
 - 数据挖掘
 - 隐藏模式的知识发现
 - 支持关联, 构造分析模型, 进行分类和预测, 并使用可视化工具提供挖掘结果.
- 三类任务的差别

从联机分析处理到联机分析挖掘

- 为什么要进行联机分析挖掘(OLAM)?
 - 数据仓库中数据的高质量
 - 数据仓库包含集成的, 一致的, 清理过的数据
 - 围绕数据仓库的有价值的信息处理基础设施
 - **ODBC, OLEDB, Web 访问, 服务机制, 报告 和 OLAP 工具**
 - 基于**OLAP**的探测式数据分析
 - 使用上下钻, 切片, 切块, 转轴等进行挖掘.
 - 数据挖掘功能的联机选择
 - 集成多种挖掘功能, 算法和任务, 并进行切换.
- **OLAM**的结构

OLAM 的结构



小结

- 数据仓库
- 数据仓库的 多维数据模型
 - 星型模式, 雪花模式, 事实星座
 - 数据方由维和度量组成
- OLAP 操作: 下钻, 上卷, 切片, 切块 和 转轴
- OLAP 服务器: ROLAP, MOLAP, HOLAP
- 数据方的有效计算
 - 部分 vs. 全部 vs. 不物化
 - 多路数组聚集
 - 位图索引和连接索引的实现

第5章：挖掘关联规则

- 关联规则挖掘
- 事务数据库中(单维布尔)关联规则挖掘的可伸缩算法
- 挖掘各种关联/相关规则
- 基于限制的关联挖掘-
- 顺序模式挖掘
- 小结

关联规则

- 关联规则反映一个事物与其他事物之间的相互依存性和关联性。如果两个或者多个事物之间存在一定的关联关系，那么，其中一个事物就能够通过其他事物预测到。
- 典型的关联规则发现问题是对超市中的货篮数据（**Market Basket**）进行分析。通过分析发现顾客放入货篮中的不同商品之间的关系来分析顾客的购买习惯。

什么是关联规则挖掘

■ 关联规则挖掘

- 首先被Agrawal, Imielinski and Swami在1993年的SIGMOD会议上提出
- 在事务、关系数据库中的项集和对象中发现频繁模式、关联规则、相关性或者因果结构
- **频繁模式**: 数据库中频繁出现的项集

■ 目的: 发现数据中的规律

- 超市数据中的什么产品会一起购买? — 啤酒和尿布
- 在买了一台PC之后下一步会购买?
- 哪种DNA对这种药物敏感?
- 我们如何自动对Web文档进行分类?

频繁模式挖掘的重要性

- 许多重要数据挖掘任务的基础
 - 关联、相关性、因果性
 - 序列模式、空间模式、时间模式、多维
 - 关联分类、聚类分析
- 更加广泛的用处
 - 购物篮分析、交叉销售、直销
 - 点击流分析、DNA序列分析等等

关联规则基本模型

- IBM公司Almaden研究中心的R. Agrawal首先提出关联规则模型，并给出求解算法AIS。随后又出现了SETM和Apriori等算法。其中，Apriori是关联规则模型中的经典算法。
 - 给定一组事务
 - 产生所有的关联规则
 - 满足最小支持度和最小可信度

关联规则基本模型

- 设 $I=\{i_1, \dots, i_m\}$ 为所有项目的集合， D 为事务数据库，事务 T 是一个项目子集（ $T \subseteq I$ ）。每一个事务具有唯一的事务标识 TID 。
- 设 A 是一个由项目构成的集合，称为项集。事务 T 包含项集 A ，当且仅当 $A \subseteq T$ 。
 - 如果项集 A 中包含 k 个项目，则称其为 k 项集。
- 项集 A 在事务数据库 D 中出现的次数占 D 中总事务的百分比叫做项集的支持度。
- 如果项集的支持度超过用户给定的最小支持度阈值，就称该项集是频繁项集（或大项集）。

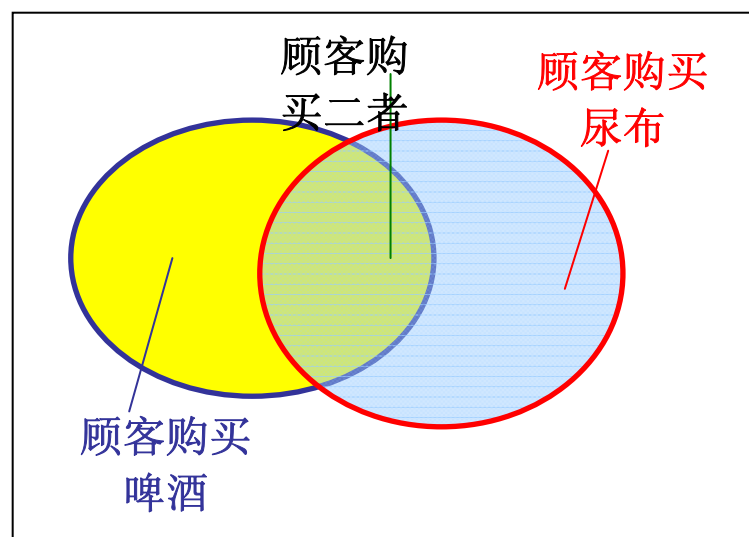
交易ID	购买的商品
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

关联规则基本模型

- 关联规则是形如 $X \Rightarrow Y$ 的逻辑蕴含式，其中 $X \subset I$ ， $Y \subset I$ ，且 $X \cap Y = \emptyset$ 。
- 如果事务数据库 D 中有 $s\%$ 的事务包含 $X \cup Y$ ，则称关联规则 $X \Rightarrow Y$ 的支持度为 $s\%$
 - 实际上，支持度是一个概率值。是一个相对计数。
 - $support(X \Rightarrow Y) = P(X \cup Y)$
- 项集的支持度计数(频率) **support_count**
 - 包含项集的事务数
- 若项集 X 的支持度记为 $support(X)$ ，规则的信任度为 $support(X \cup Y) / support(X)$ 。
 - 是一个条件概率 $P(Y | X)$ 。 $confidence(X \Rightarrow Y) = P(Y | X)$
 - $= support_count(X \cup Y) / support_count(X)$

频繁模式和关联规则

Transaction-id	Items bought
10	A, B, D
20	A, C, D
30	A, D, E
40	B, E, F
50	B, C, D, E, F



- Itemset $X = \{x_1, \dots, x_k\}$
- 找出满足最小支持度和置信度的所规则 $X \rightarrow Y$
 - 支持度, s , 事务包含 $X \cup Y$ 的概率
 - 置信度, c , 事务含 X 也包含 Y 的条件概率.

令 $sup_{min} = 50\%$, $conf_{min} = 50\%$

Freq. Pat.: $\{A:3, B:3, D:4, E:3, AD:3\}$

关联规则 Association rules:

$A \rightarrow D$ (60%, 100%)

$D \rightarrow A$ (60%, 75%)

挖掘关联规则—一个例子

Transaction-id	Items bought
10	A, B, C
20	A, C
30	A, D
40	B, E, F

最小支持度 50%
最小置信度 50%

Frequent pattern	Support
{A}	75%
{B}	50%
{C}	50%
{A, C}	50%

规则 $A \Rightarrow C$:

支持度 = $\text{support}(\{A\} \cup \{C\}) = 50\%$

置信度 = $\text{support}(\{A\} \cup \{C\}) / \text{support}(\{A\}) = 66.6\%$

闭频繁项集 and 极大频繁项集

- 一个长模式包含子模式的数目, e.g., $\{a_1, \dots, a_{100}\}$ contains $\binom{100}{1} + \binom{100}{2} + \dots + \binom{100}{100} = 2^{100} - 1 = 1.27 \times 10^{30}$ sub-patterns!
- 解: Mine *closed patterns* and *max-patterns* instead
- 一个频繁项集 X 是闭的, 如果 X 是频繁的, 且不存在真超项集 *no super-pattern* $Y \supset X$, 有相同的支持度计数
 - (proposed by Pasquier, et al. @ ICDT'99)
- 项集 X 是极大频繁项集 if X is frequent and there exists no frequent super-pattern $Y \supset X$
 - (proposed by Bayardo @ SIGMOD'98)
- 两者有不同, 极大频繁项集定义中对真超集要松一些。

闭频繁项集 and 极大频繁项集

- **Exercise.** $DB = \{ \langle a_1, \dots, a_{100} \rangle, \langle a_1, \dots, a_{50} \rangle \}$
 - $Min_sup = 1.$
- What is the set of **closed itemset**?
 - $\langle a_1, \dots, a_{100} \rangle: 1$
 - $\langle a_1, \dots, a_{50} \rangle: 2$
- What is the set of **max-pattern**?
 - $\langle a_1, \dots, a_{100} \rangle: 1$
- What is the set of **all patterns**?
 - **!!**

关联规则基本模型

- 关联规则就是支持度和信任度分别满足用户给定阈值的规则。
-
- 发现关联规则需要经历如下两个步骤：
 - 找出所有频繁项集。
 - 由频繁项集生成满足最小信任度阈值的规则。

第5章：挖掘关联规则

- 关联规则挖掘
- 事务数据库中(单维布尔)关联规则挖掘的可伸缩算法
- 挖掘各种关联/相关规则
- 基于限制的关联挖掘-
- 顺序模式挖掘
- 小结

Apriori算法的步骤

- **Apriori**算法命名源于算法使用了频繁项集性质的先验（**Prior**）知识。
- **Apriori**算法将发现关联规则的过程分为两个步骤：
 - 通过迭代，检索出事务数据库中的所有频繁项集，即支持度不低于用户设定的阈值的项集；
 - 利用频繁项集构造出满足用户最小信任度的规则。
- 挖掘或识别出所有频繁项集是该算法的核心，占整个计算量的大部分。

频繁项集

- 为了避免计算所有项集的支持度（实际上频繁项集只占很少一部分），Apriori算法引入潜在频繁项集的概念。
- 若潜在频繁 k 项集的集合记为 C_k ，频繁 k 项集的集合记为 L_k ， m 个项目构成的 k 项集的集合为 C_m^k ，则三者之间满足关系 $L_k \subseteq C_k \subseteq C_m^k$ 。
- 构成潜在频繁项集所遵循的原则是“频繁项集的子集必为频繁项集”。

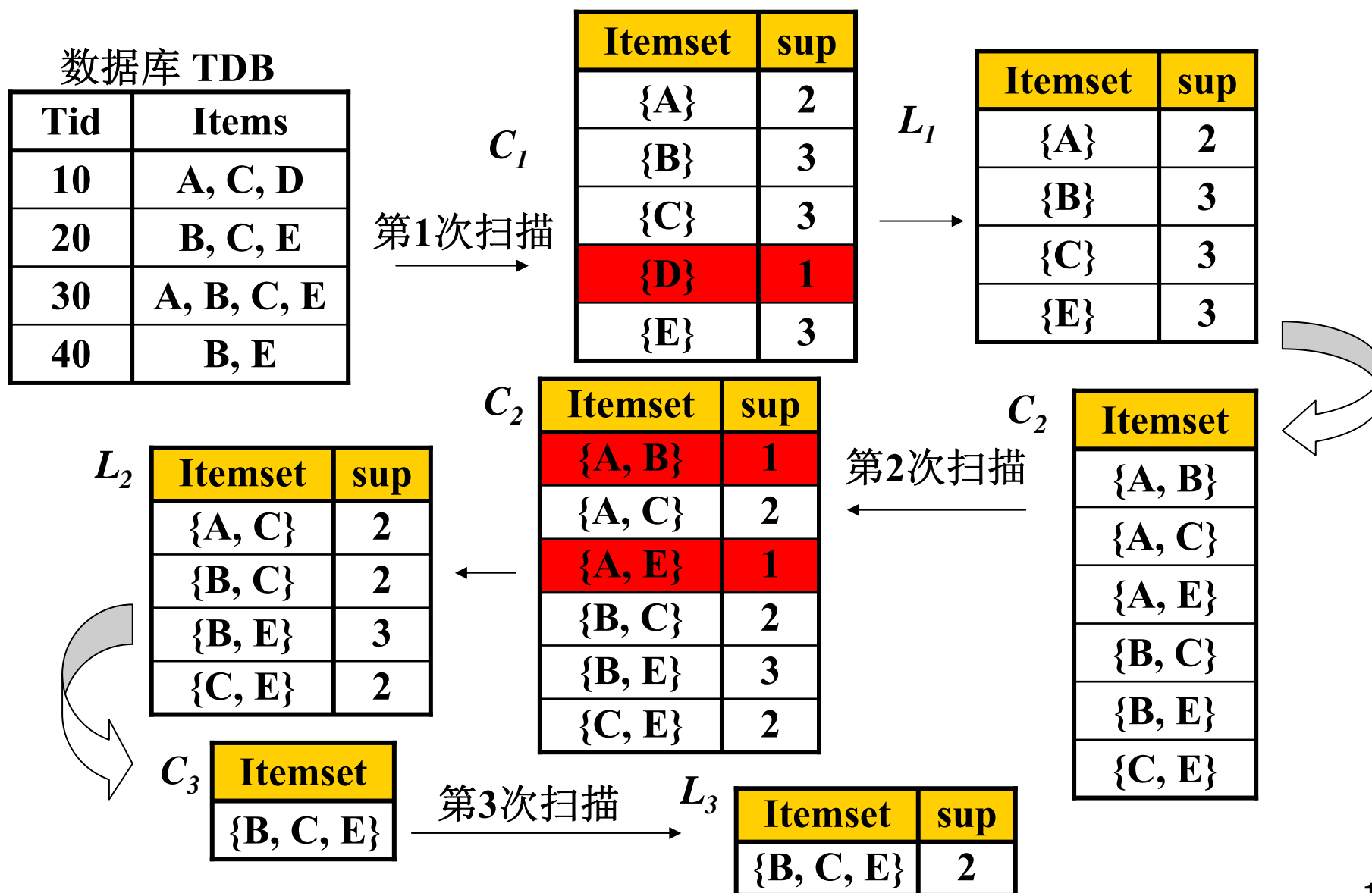
关联规则的性质

- 性质1: 频繁项集的子集必为频繁项集。
- 性质2: 非频繁项集的超集一定是非频繁的。
- Apriori算法运用性质1, 通过已知的频繁项集构成长度更大的项集, 并将其称为潜在频繁项集。
 - 潜在频繁 k 项集的集合 C_k 是指由有可能成为频繁 k 项集的项集组成的集合。
- 以后只需计算潜在频繁项集的支持度, 而不必计算所有不同项集的支持度, 因此在一定程度上减少了计算量。

Apriori: 一种候选产生-测试方法

- 频繁项集的任何子集必须是频繁的
 - 如果 {beer, diaper, nuts} 是频繁的, {beer, diaper}也是
 - 每个包含 {beer, diaper, nuts}的事务 也包含 {beer, diaper}
- Apriori 剪枝原则:
 - 如果一个项集不是频繁的, 将不产生/测试它的超集!
- 方法:
 - 由长度为k的频繁项集产生长度为 (k+1) 的候选项集, 并且
 - 根据 DB测试这些候选
- 性能研究表明了它的有效性和可伸缩性

Apriori 算法 — 一个例子



Apriori算法

- (1) $L_1 = \{\text{频繁1项集}\};$
- (2) **for**($k=2; L_{k-1} \neq \emptyset; k++$) **do begin**
- (3) $C_k = \text{apriori_gen}(L_{k-1});$ //新的潜在频繁项集
- (4) **for all** *transactions* $t \in D$ **do begin**
- (5) $C_t = \text{subset}(C_k, t);$ //找出t中包含的潜在的频繁项
- (6) **for all** *candidates* $c \in C_t$ **do**
- (7) $c.\text{count}++;$
- (8) **end;**
- (9) $L_k = \{c \in C_k \mid c.\text{count} \geq \text{minsup}\}$
- (10) **end;**
- (11) **Answer** = $\bigcup_k L_k$

Apriori的重要细节

- 如何产生候选?
 - 步骤 1: L_k 的自连接
 - 步骤 2: 剪枝
- 候选产生的例子
 - $L_3 = \{abc, abd, acd, ace, bcd\}$
 - 自连接: $L_3 * L_3$
 - $Abcd$: 由 abc 和 abd
 - $Acde$: 由 acd 和 ace
 - 剪枝:
 - $acde$ 被删除, 因为 ade 不在 L_3
 - $C_4 = \{abcd\}$

如何产生候选?

- 假定 L_{k-1} 中的项集已排序(按字典序排序)
- 步骤 1: L_{k-1} 自连接

procedure apriori_gen(L_{k-1} :frequent $(k-1)$ -itemsets)

```
(1)   for each itemset  $l_1 \in L_{k-1}$ 
(2)     for each itemset  $l_2 \in L_{k-1}$ 
(3)       if  $(l_1[1] = l_2[1]) \wedge (l_1[2] = l_2[2]) \wedge \dots \wedge (l_1[k-2] = l_2[k-2]) \wedge (l_1[k-1] < l_2[k-1])$  then {
(4)          $c = l_1 \bowtie l_2$ ; // join step: generate candidates
(5)         if has_infrequent_subset( $c, L_{k-1}$ ) then
(6)           delete  $c$ ; // prune step: remove unfruitful candidate
(7)         else add  $c$  to  $C_k$ ;
(8)       }
(9)   return  $C_k$ ;
```

- Step 2: 剪枝

procedure has_infrequent_subset(c : candidate k -itemset;

L_{k-1} : frequent $(k-1)$ -itemsets); // use prior knowledge

```
(1)   for each  $(k-1)$ -subset  $s$  of  $c$ 
(2)     if  $s \notin L_{k-1}$  then
(3)       return TRUE;
(4)   return FALSE;
```


例子_支持计数=2

AllElectronics 数据库

TID	List of item ID's
T100	I1,I2,I5
T200	I2,I4
T300	I2,I3
T400	I1,I2,I4
T500	I1,I3
T600	I2,I3
T700	I1,I3
T800	I1,I2,I3,I5
T900	I1,I2,I3

C_1

项集	支持度计数
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

比较候选支持度计数
与最小支持度计数



L_1

项集	支持度计数
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

C_2

项集
{I1,I2}
{I1,I3}
{I1,I4}
{I1,I5}
{I2,I3}
{I2,I4}
{I2,I5}
{I3,I4}
{I3,I5}
{I4,I5}

由 L_1 产生
候选 C_2



扫描D, 对每个
候选计数



C_2

项集	支持度计数
{I1,I2}	4
{I1,I3}	4
{I1,I4}	1
{I1,I5}	2
{I2,I3}	4
{I2,I4}	2
{I2,I5}	2
{I3,I4}	0
{I3,I5}	1
{I4,I5}	0

比较候选支持度计数
与最小支持度计数

L_2

项集	支持度计数
{I1,I2}	4
{I1,I3}	4
{I1,I5}	2
{I2,I3}	4
{I2,I4}	2
{I2,I5}	2

比较候选支持度计数
与最小支持度计数

L ₂	
项集	支持度计数
{I1,I2}	4
{I1,I3}	4
{I1,I5}	2
{I2,I3}	4
{I2,I4}	2
{I2,I5}	2

由L₂产生
候选C₃

C ₃	
项集	
{I1,I2,I3}	
{I1,I2,I5}	

扫描D, 对每个
候选计数

C ₃	
项集	支持度计数
{I1,I2,I3}	2
{I1,I2,I5}	2

比较候选支持度计数
与最小支持度计数

L ₃	
项集	支持度计数
{I1,I2,I3}	2
{I1,I2,I5}	2

- 连接： $C_3 = L_2 \bowtie L_2$
 $L_2 = \{\{I1,I2\}, \{I1,I3\}, \{I1,I5\}, \{I2,I3\}, \{I2,I4\}, \{I2,I5\}\}$
 $\{\{I1,I2\}, \{I1,I3\}, \{I1,I5\}, \{I2,I3\}, \{I2,I4\}, \{I2,I5\}\} =$
 $\{\{I1,I2,I3\}, \{I1,I2,I5\}, \{I1,I3,I5\}, \{I2,I3,I4\}, \{I2,I3,I5\}, \{I2,I4,I5\}\}$
- 使用 Apriori 性质剪枝：频繁项集的所有子集必须是频繁的。存在候选项集，其子集不是频繁的吗？
 - {I1,I2,I3}的 2-项子集是{I1,I2}，{I1,I3}和{I2,I3}。{I1,I2,I3}的所有 2-项子集都是 L₂ 的元素。因此，保留{I1,I2,I3}在 C₃ 中。
 - {I1,I2,I5}的 2-项子集是{I1,I2}，{I1,I5}和{I2,I5}。{I1,I2,I5}的所有 2-项子集都是 L₂ 的元素。因此，保留{I1,I2,I5}在 C₃ 中。
 - {I1,I3,I5}的 2-项子集是{I1,I3}，{I1,I5}和{I3,I5}。{I3,I5}不是 L₂ 的元素，因而不是频繁的。这样，由 C₃ 中删除{I1,I3,I5}。

由频繁项集产生关联规则

- 根据公式产生关联规则

$$confidence(A \Rightarrow B) = P(B|A) = \frac{support_count(A \cup B)}{support_count(A)}$$

- 对于每个频繁项集 l ，产生所有的非空子集
- 对于 l 的每个非空子集 s ，如果 $\frac{support_count(l)}{support_count(s)} \geq min_conf$ ，则输出规则“ $s \Rightarrow (l-s)$ ”

频繁模式挖掘的挑战

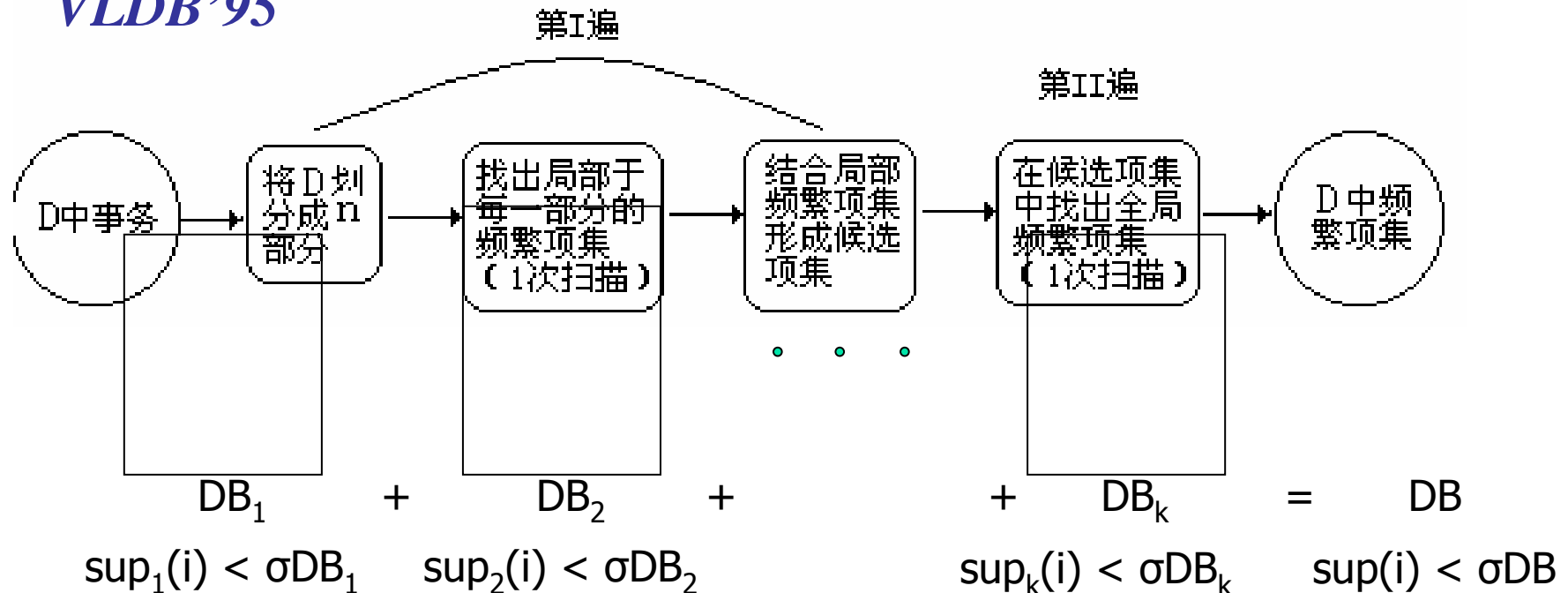
- 挑战
 - 事务数据库的多遍扫描
 - 数量巨大的候选
 - 候选支持度计数繁重的工作量
- 改进 **Apriori**: 基本思想
 - 减少事务数据库的扫描遍数
 - 压缩候选数量
 - 便于候选计数

提高Apriori算法的方法

- **Hash-based itemset counting** (散列项集计数)
- **Transaction reduction** (事务压缩)
- **Partitioning** (划分)
- **Sampling** (采样)

划分: 只扫描数据库两次

- 项集在DB中是频繁的, 它必须至少在DB的一个划分中是频繁的
 - 扫描 1: 划分数数据库, 并找出局部频繁模式 local frequent itemset
 - 扫描 2: 求出全局频繁模式
- A. Savasere, E. Omiecinski, and S. Navathe. **An efficient algorithm for mining association in large databases.** In *VLDB'95*



抽样-频繁模式

- 选取原数据库的一个样本, 使用Apriori 算法在样本中挖掘频繁模式
- 扫描一次数据库, 验证在样本中发现的频繁模式.
- 再次扫描数据库, 找出遗漏的频繁模式
- 牺牲一些精度换取有效性。
- **H. Toivonen. Sampling large databases for association rules.
In *VLDB'96***

DHP: 压缩候选的数量

- 散列项集到对应的桶中，一个其hash桶的计数小于阈值的 k -itemset 不可能是频繁的
- J. Park, M. Chen, and P. Yu. **An effective hash-based algorithm for mining association rules.** In *SIGMOD'95*

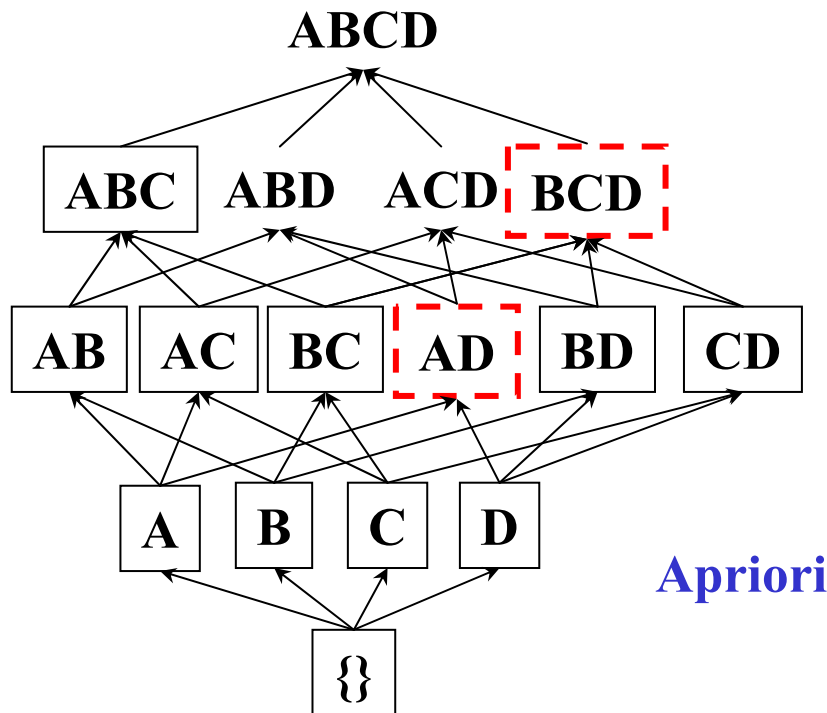
使用散列函数

$$\text{hash}(x, y) = ((\text{order of } x) * 10 + (\text{order of } y)) \bmod 7$$

创建散列表 H_2

H_2							
桶地址	0	1	2	3	4	5	6
桶计数	2	2	4	2	2	4	4
桶内容	{I1,I4} {I3,I5}	{I1,I5} {I1,I5}	{I2,I3} {I2,I3} {I2,I3} {I2,I3}	{I2,I4} {I2,I4}	{I2,I5} {I2,I5}	{I1,I2} {I1,I2} {I1,I2} {I1,I2}	{I1,I3} {I1,I3} {I1,I3} {I1,I3}

DIC (Dynamic itemset counting): 减少扫描次数

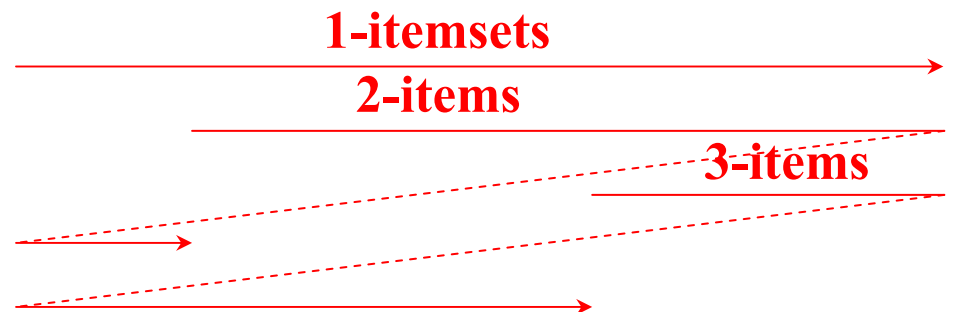


Itemset lattice

S. Brin R. Motwani, J. Ullman, and S. Tsur. **Dynamic itemset counting and implication rules for market basket data.** In *SIGMOD'97*

- 一旦确定 A 和 D 是频繁的, 立即开始 AD 的计数
- 一旦确定 BCD 的两个长度为2的子集是频繁的, 立即开始BCD 的计数

Apriori



DIC

使用垂直数据格式挖掘频繁项集 **Vertical Data Format**

- 使用**tid-list**, 包含**item**的事务的标识的集合;
 - M. Zaki et al. **New algorithms for fast discovery of association rules**. In **KDD'97**
- 扫描一次数据集将水平格式数据转化为垂直格式
- 通过频繁**k**项集的**tid-list**的交集, 计算对应**(k+1)**项集的**tid-list**

<i>itemset</i>	<i>TID_set</i>
I1	{T100, T400, T500, T700, T800, T900}
I2	{T100, T200, T300, T400, T600, T800, T900}
I3	{T300, T500, T600, T700, T800, T900}
I4	{T200, T400}
I5	{T100, T800}

<i>itemset</i>	<i>TID_set</i>
{I1, I2, I3}	{T800, T900}
{I1, I2, I5}	{T100, T800}

The 2-itemsets in vertical data format.

<i>itemset</i>	<i>TID_set</i>
{I1, I2}	{T100, T400, T800, T900}
{I1, I3}	{T500, T700, T800, T900}
{I1, I4}	{T400}
{I1, I5}	{T100, T800}
{I2, I3}	{T300, T600, T800, T900}
{I2, I4}	{T200, T400}
{I2, I5}	{T100, T800}
{I3, I5}	{T800}

频繁模式挖掘的瓶颈

- 多遍数据库扫描是 昂贵的
- 挖掘长模式需要很多遍扫描, 并产生大量候选
 - 挖掘频繁模式 $i_1i_2\dots i_{100}$
 - 扫描次数: 100
 - 候选个数: $\binom{100}{1} + \binom{100}{2} + \dots + \binom{100}{100} = 2^{100} - 1 = 1.27 * 10^{30} !$
- 瓶颈: 候选产生-测试
- 能够避免候选产生吗?

挖掘频繁模式而不产生候选

- 使用局部频繁的项, 由短模式增长产生长模式
 - “abc” 是频繁模式
 - 得到包含 “abc”的所有事务: **DB|abc**
 - “d” 是 **DB|abc** 中的局部频繁项 → **abcd** 是频繁模式

由事务数据库构造FP-树

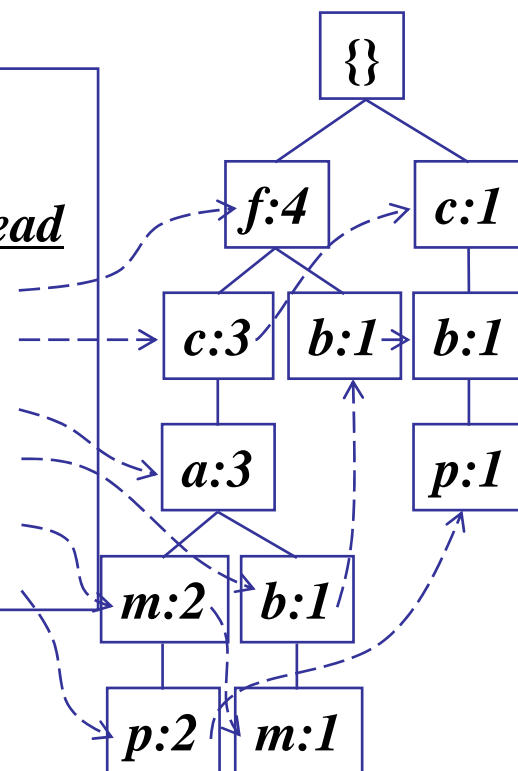
<i>TID</i>	<i>Items bought</i>	<i>(ordered) frequent items</i>
100	{ <i>f, a, c, d, g, i, m, p</i> }	{ <i>f, c, a, m, p</i> }
200	{ <i>a, b, c, f, l, m, o</i> }	{ <i>f, c, a, b, m</i> }
300	{ <i>b, f, h, j, o, w</i> }	{ <i>f, b</i> }
400	{ <i>b, c, k, s, p</i> }	{ <i>c, b, p</i> }
500	{ <i>a, f, c, e, l, p, m, n</i> }	{ <i>f, c, a, m, p</i> }

$\min_support = 3$

1. 扫描 DB 一次, 找出频繁 1-itemset (单个项的模式)
2. 按频率的降序将频繁项排序, 得到 f-list
3. 再次扫描 DB, 构造 FP-树

Header Table		
<i>Item</i>	<i>frequency</i>	<i>head</i>
<i>f</i>	4	
<i>c</i>	4	
<i>a</i>	3	
<i>b</i>	3	
<i>m</i>	3	
<i>p</i>	3	

F-list=f-c-a-b-m-p

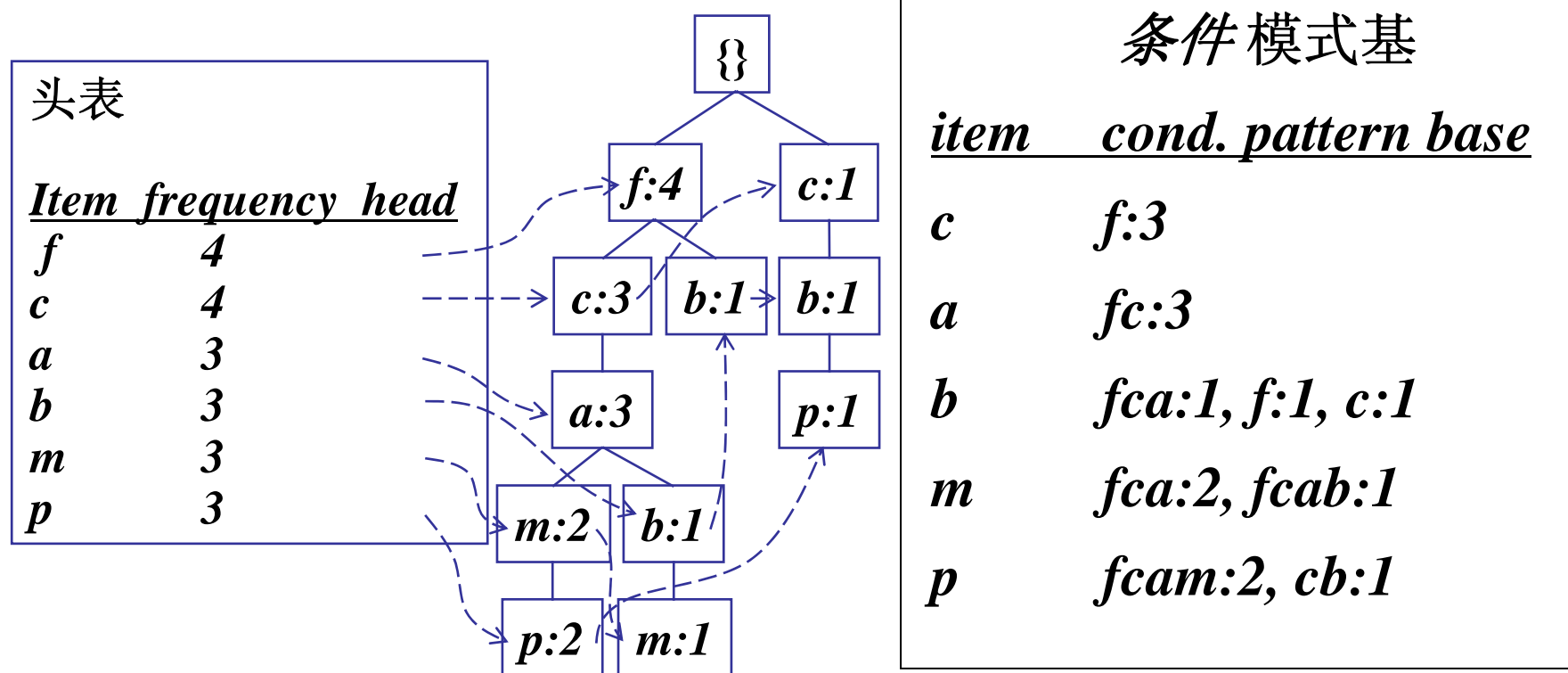


划分模式和数据库

- 可以按照**f-list** 将频繁模式划分成子集
 - **F-list=f-c-a-b-m-p**
 - 包含 **p**的模式
 - 包含 **m** 但不包含 **p**的模式
 - ...
 - 包含 **c** 但不包含 **a, b, m, p**
 - 模式 **f**
- 完全性和非冗余性

从p-条件数据库找出含p的模式

- 从FP-树的频繁项头表开始
- 沿着频繁项 p 的链搜索FP-树
- 收集项 p 的所有变换的前缀路径形成 p 的模式基

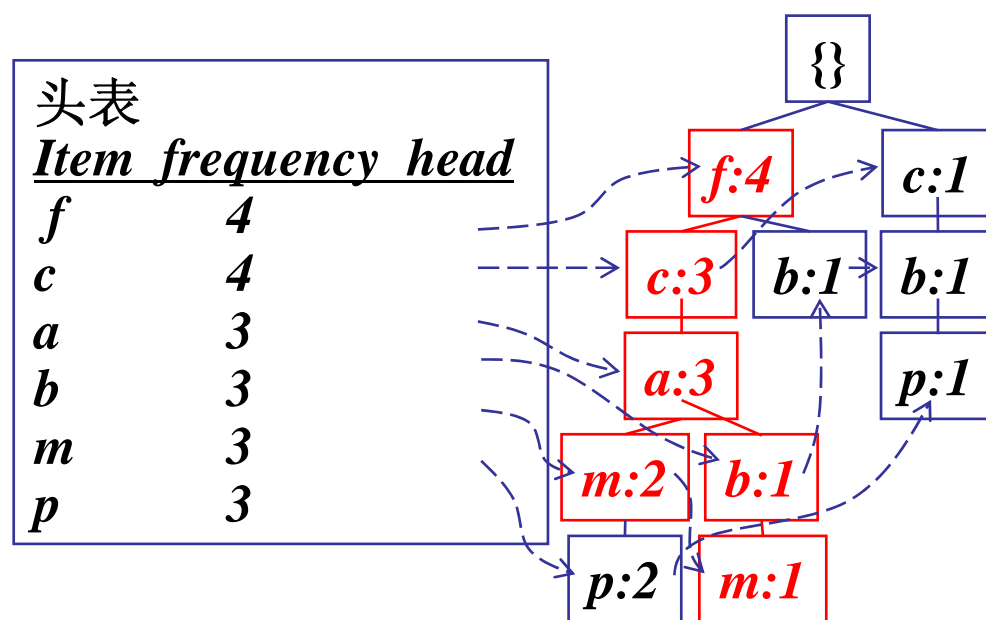


通过建立条件模式库得到频繁集

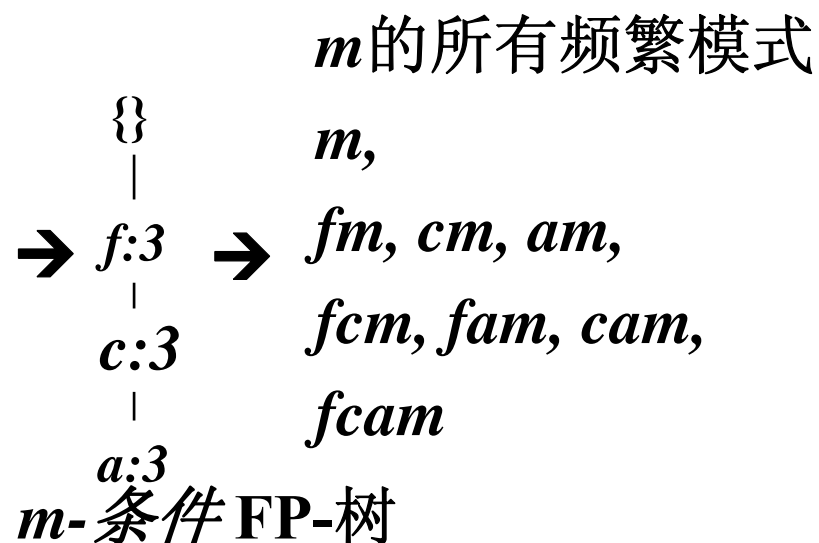
项	条件模式库	条件FP-tree
p	$\{(fcam:2), (cb:1)\}$	$\{(c:3)\} p$
m	$\{(fca:2), (fcab:1)\}$	$\{(f:3, c:3, a:3)\} m$
b	$\{(fca:1), (f:1), (c:1)\}$	Empty
a	$\{(fc:3)\}$	$\{(f:3, c:3)\} a$
c	$\{(f:3)\}$	$\{(f:3)\} c$
f	Empty	Empty

从条件模式基到条件FP-树

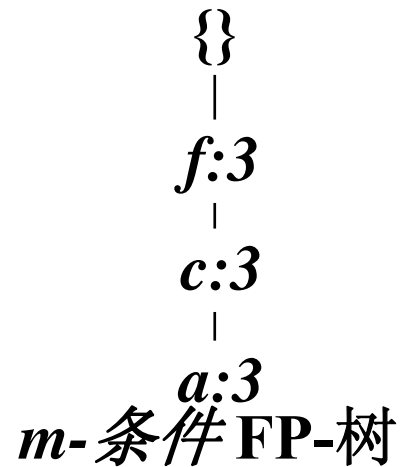
- 对于每个条件模式基
 - 累计条件模式基中每个项的计数
 - 构造模式基中频繁项的FP-树



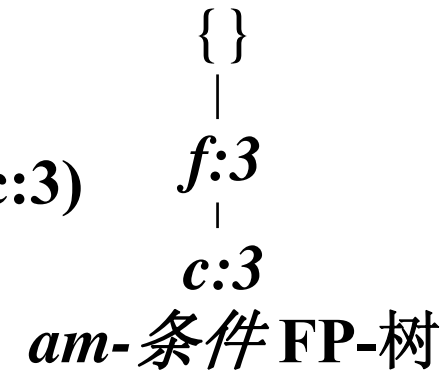
m-条件模式基：
fca:2, fcab:1



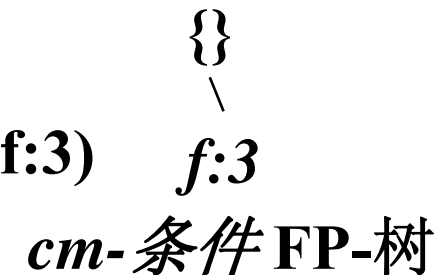
递归: 挖掘每个条件 FP-树



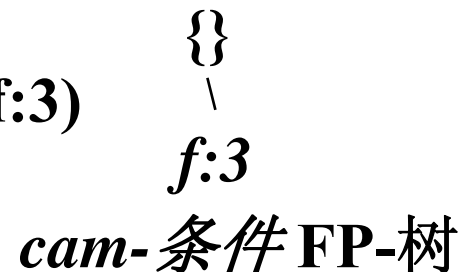
“am”的条件模式基: (fc:3)



“cm”的条件模式基: (f:3)

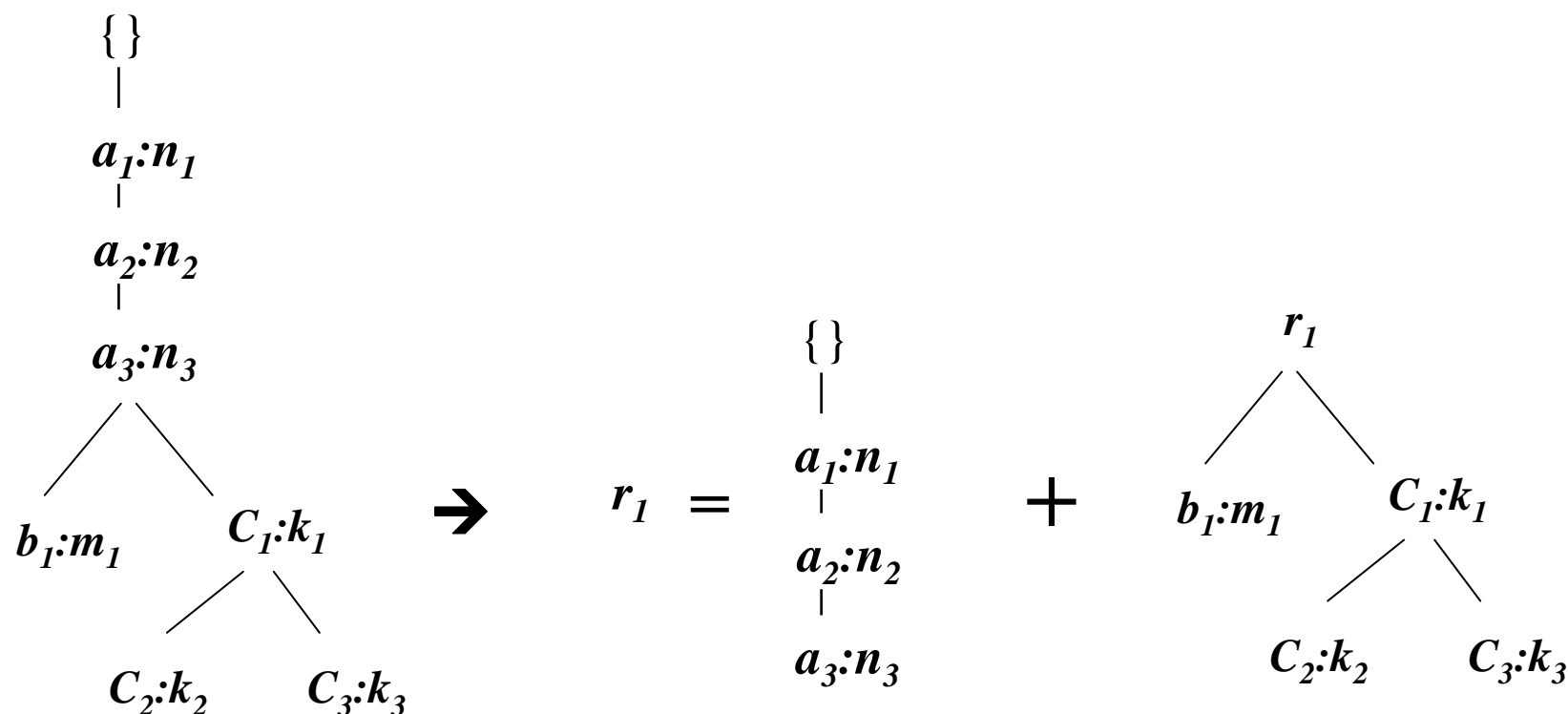


“cam”的条件模式基: (f:3)



特殊情况: FP-树中的单个前缀路径

- 假定 (条件) FP-树 T 具有单个共享的前缀路径 P
- 挖掘可以分解成两步
 - 将单个前缀路径归约成一个结点
 - 连接两部分的挖掘结果



使用FP-树挖掘频繁模式

- 基本思想: 频繁模式增长
 - 通过模式和数据库划分递归地增长频繁模式
- 方法
 - (1)对于每个频繁项, 构造它的条件模式基
 - (2)然后构造它的条件 FP-树
 - (3)在新构造的条件FP-树上重复这一过程
 - 直到结果条件 FP-树为空, 或者它只包含一条路径—单个路径将产生其子路径的所有组合, 每个子路径是一个频繁模式

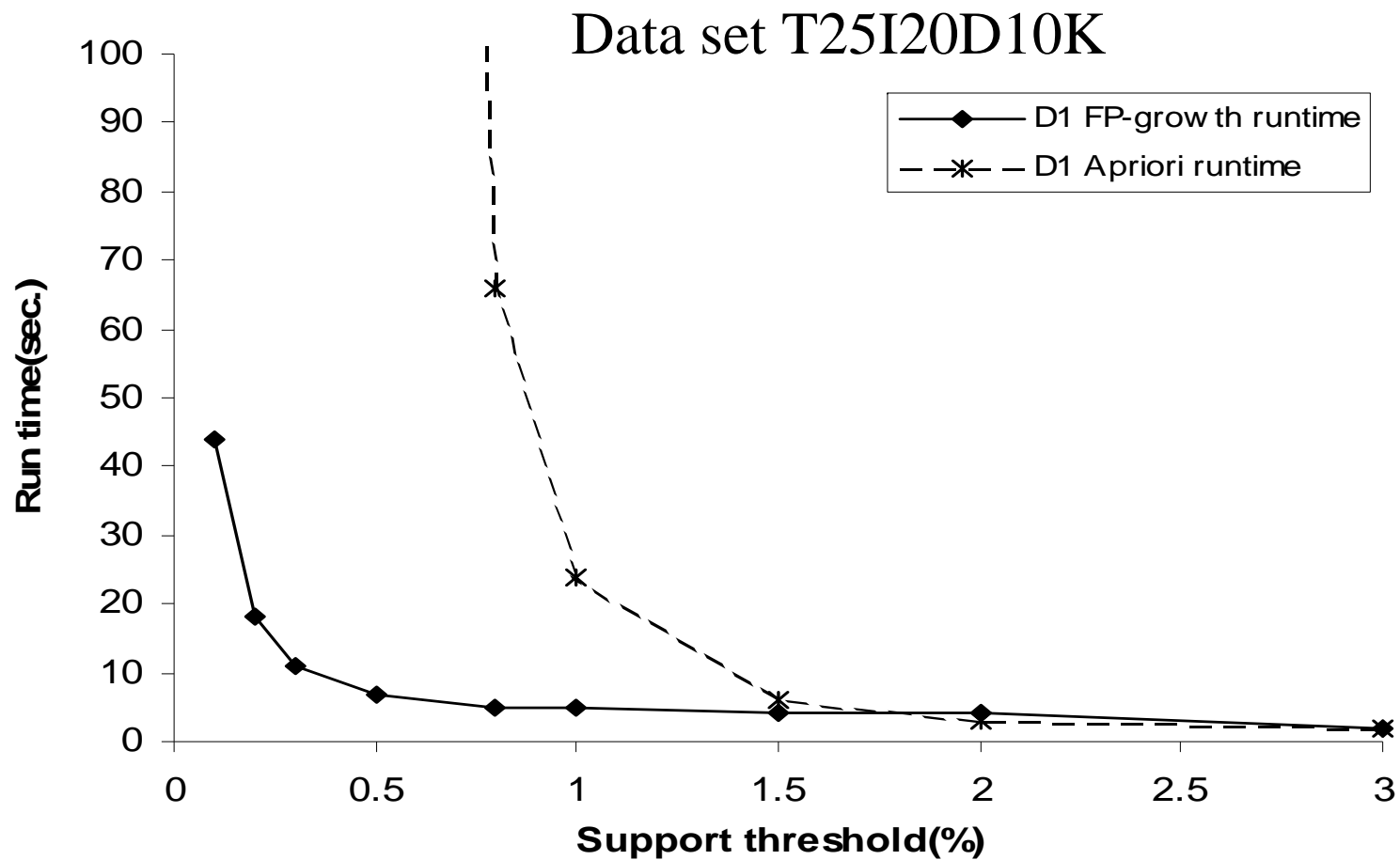
FP-树结构的优点

- 完全性
 - 保留频繁模式挖掘的完整信息
 - 不截断任何事务的长模式
- 压缩性
 - 压缩无关信息—非频繁项被删除
 - 项按频率的降序排列: 越是频繁出现, 越可能被共享
 - 绝对不比原来的数据库大 (不计结点链和计数字段)

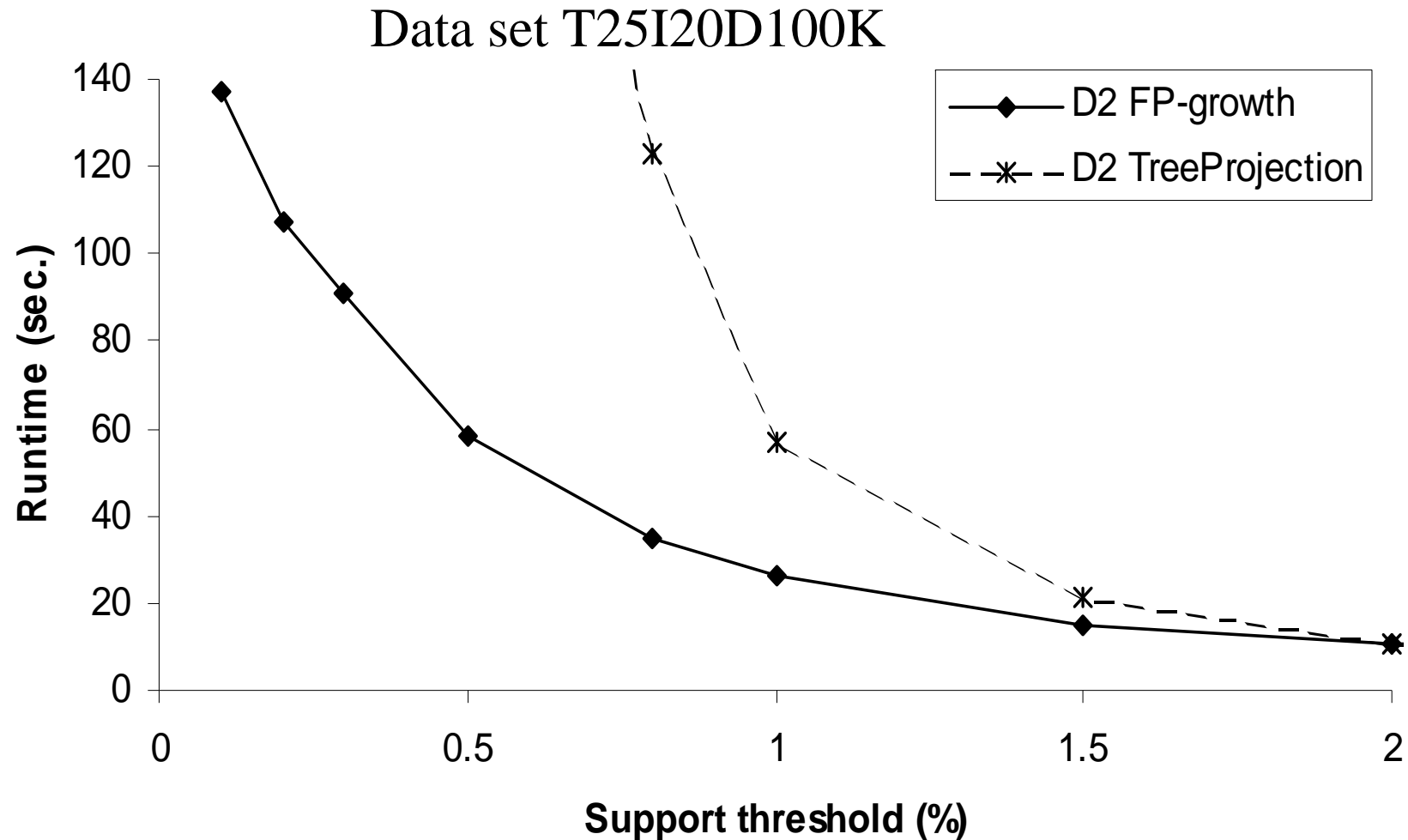
FP-增长的规模化

- FP-树不能放在内存, 怎么办?—数据库投影
- 数据库投影
 - 首先将数据库划分成一组投影 数据库
 - 然后对每个投影数据库构造并挖掘FP-树

FP-增长 vs. Apriori: 随支持度增长的可伸缩性



FP-增长 vs. 树-投影:随支持度增长的可伸缩性



为什么FP-增长是赢家？

- 分治：
 - 根据已经得到的频繁模式划分任务和数据库
 - 导致较小的数据库的聚焦的搜索
- 其它因素
 - 没有候选产生, 没有候选测试
 - 压缩数据库 : **FP-树**结构
 - 不重复地扫描整个数据库
 - 基本操作—局部频繁项计数和建立子**FP-树**, 没有模式搜索和匹配

有关的其他方法

- 挖掘频繁闭项集合和最大模式
 - **CLOSET (DMKD'00)**
- 挖掘序列模式
 - **FreeSpan (KDD'00), PrefixSpan (ICDE'01)**
- 频繁模式的基于限制的挖掘
 - **Convertible constraints (KDD'00, ICDE'01)**
- 计算具有复杂度量的冰山数据方
 - **H-tree and H-cubing algorithm (SIGMOD'01)**

最大模式

- 频繁模式 $\{a_1, \dots, a_{100}\}$ 包含 $(100^1) + (100^2) + \dots + (1^1 0^0 0^0) = 2^{100} - 1 = 1.27 * 10^{30}$ 频繁子模式!
- 最大模式: 频繁模式, 其真超模式都不是频繁的
 - BCDE, ACD 是最大模式
 - BCD 不是最大模式

Min_sup=2

Tid	Items
10	A,B,C,D,E
20	B,C,D,E,
30	A,C,D,F

MaxMiner: 挖掘最大模式

- 扫描1: 找出频繁项

- *A, B, C, D, E*

- 扫描2: 找出以下项集的支持度

- *AB, AC, AD, AE, ABCDE*

- *BC, BD, BE, BCDE*

- *CD, CE, CDE, DE*

Tid	Items
10	<i>A,B,C,D,E</i>
20	<i>B,C,D,E,</i>
30	<i>A,C,D,F</i>

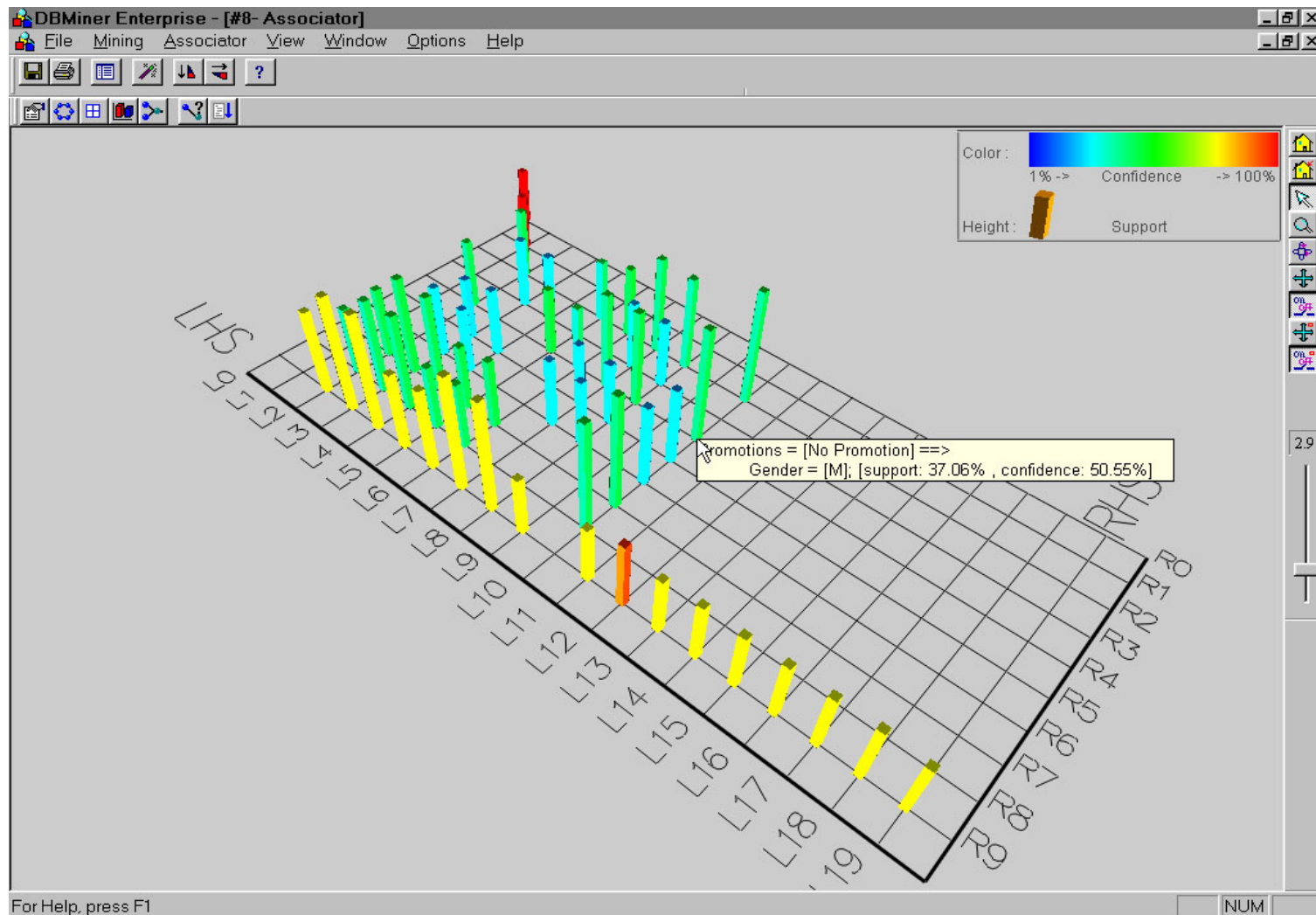
潜在的**最大模式**

- 由于 *BCDE* 是最大模式,

不必在此后的扫描时检查 *BCD, BDE, CDE*

- R. Bayato. **Efficiently mining long patterns from databases.**
In *SIGMOD'98*

关联规则的可视化: Pane Graph



第5章：挖掘关联规则

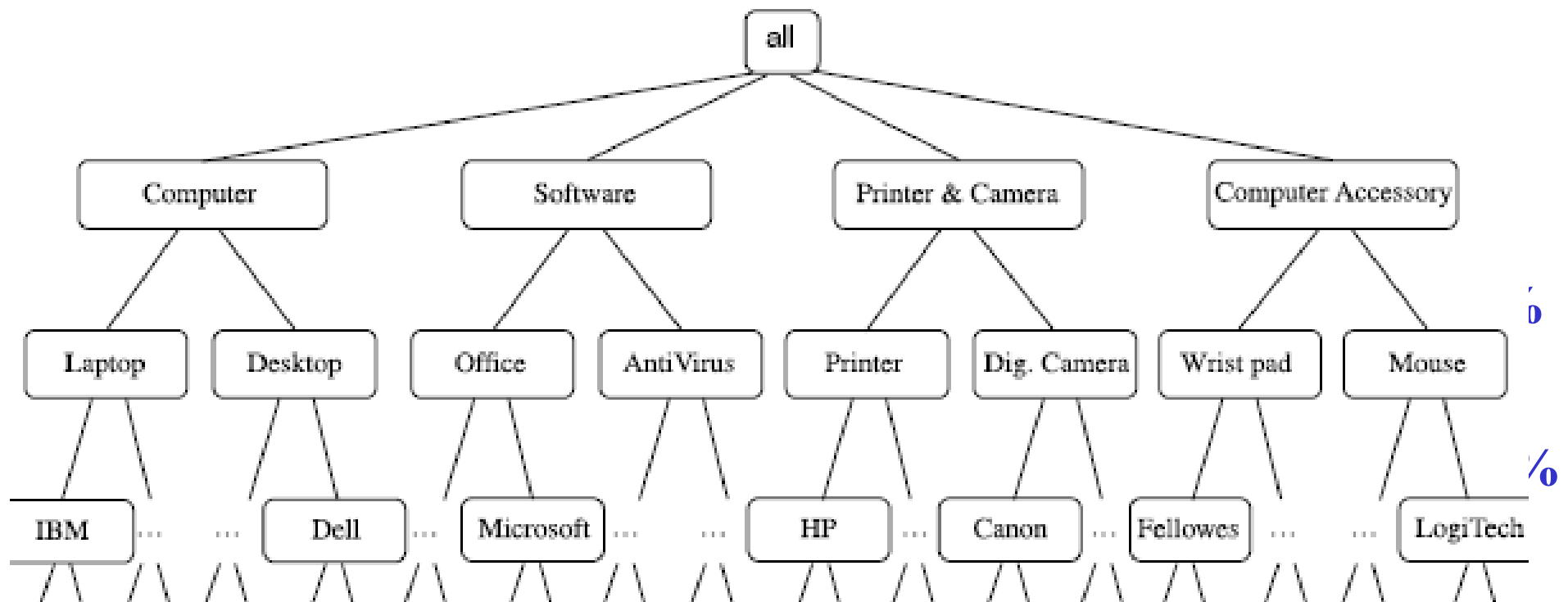
- 关联规则挖掘
- 事务数据库中(单维布尔)关联规则挖掘的可伸缩算法
- 挖掘各种关联/相关规则
- 基于限制的关联挖掘
- 顺序模式挖掘
- 小结

挖掘各种规则或规律性

- 多层关联规则,
- 多维关联规则,
- 量化关联规则,
- 相关性和因果关系, 比率规则, 序列模式, 显露模式, 时间关联, 局部周期性

多层关联规则

- 项常常形成层次结构-概念分层
- 多个抽象层次上挖掘得到的关联规则-多层关联规则
- 灵活的支持度设定：较低层中的项一般具有较低的支持度。



多层关联：冗余过滤

- 由于项之间的“祖先”联系,有些规则可能是多余的.
- 例
 - **milk \Rightarrow wheat bread [support = 8%, confidence = 70%]**
 - **2% milk \Rightarrow wheat bread [support = 2%, confidence = 72%]**
 - **其中2% milk 占milk的1/4**
- 我们可以说第一个规则是第二个规则的祖先.
- 一个规则是冗余的, 如果根据规则的祖先, 其支持度和置信度都接近于“期望”值.

多层挖掘: 逐步深入

- 一种自顶向下, 逐步深入的方法:
 - 首先挖掘最高层的频繁模式:
milk (15%), bread (10%)
 - 然后挖掘它们下层“较弱的”频繁模式:
2% milk (5%), wheat bread (4%)
- 多层之间的不同的最小支持度阈值导致不同的算法:
 - 如果不同层之间采用相同的 *min_support* 则丢弃 t 如果 t' 的任意祖先是非频繁的.
 - 如果在较低层采用递减的 *min_support* 则只考察其祖先为频繁的项集.

多维关联规则

- 单维规则:包括单个谓词（可以多次出现）或单个维
 $\text{buys}(X, \text{"milk"}) \Rightarrow \text{buys}(X, \text{"bread"})$
- 多维规则: 维或谓词 ≥ 2
 - 维间关联规则 (不含重复谓词)
 $\text{age}(X, \text{"19-25"}) \wedge \text{occupation}(X, \text{"student"}) \Rightarrow \text{buys}(X, \text{"coke"})$
 - 混合维关联规则 (含重复谓词)
 $\text{age}(X, \text{"19-25"}) \wedge \text{buys}(X, \text{"popcorn"}) \Rightarrow \text{buys}(X, \text{"coke"})$
- 数据的属性可分为两类
 - 分类属性
 - 有限个不同值, 值之间无序
 - 量化属性
 - 数值的, 值之间隐含次序

挖掘多维关联规则的技术

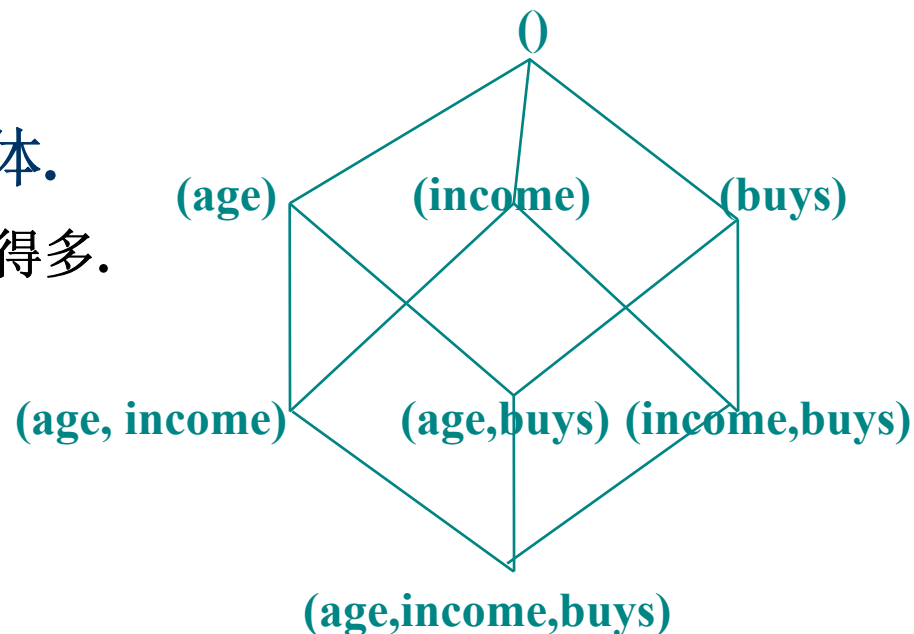
- 搜索频繁 k -谓词集 :包含 k 个合取谓词的集合
 - 例: {age, occupation, buys} 是一个 3-谓词集.
 - 可以按如何处理 age 对技术分类.
 1. 使用量化属性的静态离散化
 - 使用预先定义的概念分层, 对量化属性静态地离散化.
 2. 量化关联规则
 - 根据数据的分布, 将量化属性离散化到“箱”.
 3. 基于距离的关联规则
 - 是一种动态的离散化过程, 它考虑数据点之间的距离.

量化属性的静态离散化

- 使用概念分层, 在挖掘之前离散化.
 - 数值用区间值替换.
- 在关系数据库中, 找出所有的频繁 k -谓词集需要 k 或 $k+1$ 次表扫描.
- 数据立方体非常适合挖掘.
 - n -维方体

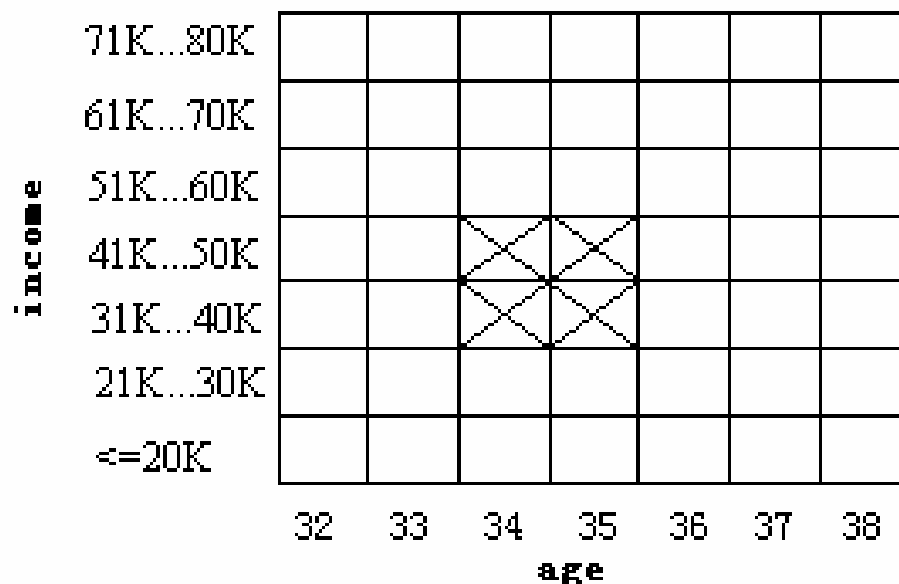
对应于谓词集合的方体.

- 从数据立方体挖掘可以快得多.



量化关联规则

- 数值属性动态地离散化
 - 使挖出的规则的置信度或紧凑性最大化.
- 2-维量化关联规则: $A_{\text{quan1}} \wedge A_{\text{quan2}} \Rightarrow A_{\text{cat}}$ (分类属性)
- ARCS方法: 使用2-D栅格,
 - 1)对属性进行(等宽)分箱
 - 2)找频繁谓词集
 - 3)规则聚类: 对“相邻的”关联规则聚类形成一般关联规则.
- 例:
 $\text{age}(X, "34-35") \wedge$
 $\text{income}(X, "31K - 50K")$
 $\Rightarrow \text{buys}(X, "high\ resolution\ TV")$



挖掘基于距离的关联规则

- 分箱方法不能紧扣区间数据的语义
- 基于距离的划分,更有意义的离散化考虑:
 - 区间内点的密度/数量
 - 区间内点的“紧密性”

Price(\$)	Equi-width (width \$10)	Equi-depth (depth 2)	Distance- based
7	[0,10]	[7,20]	[7,7]
20	[11,20]	[22,50]	[20,22]
22	[21,30]	[51,53]	[50,53]
50	[31,40]		
51	[41,50]		
53	[51,60]		

具有灵活的支持度限制的 多层ML/MD多维关联规则

- 为什么?
 - 现实中项的出现频率差异很大
 - 购物中的钻石, 表, 笔
 - 一致的支持度可能不是一种好的模型
- 灵活的模型
 - 通常, 层越低, 维的组合越多, 长模式越长, 支持度越小
 - 一般规则应当是特指的, 易于理解的
 - 特殊的项或特殊的项群可能被个别地指定, 并具有较高的优先权

兴趣度度量: 相关性(Lift)

- *play basketball* \Rightarrow *eat cereal* [40%, 66.7%] 是误导
 - 吃谷类食品的学生所占的百分比为75%, 比 66.7%还高.
- *play basketball* \Rightarrow *not eat cereal* [20%, 33.3%] 更准确, 其支持度和置信度都较低
- 依赖/相关事件的度量:

$$lift = \frac{P(A \cup B)}{P(A)P(B)}$$

	Basketball	Not basketball	Sum (row)
Cereal谷类	2000	1750	3750
Not cereal	1000	250	1250
Sum(col.)	3000	2000	5000

$$lift(B, C) = \frac{2000 / 5000}{3000 / 5000 * 3750 / 5000} = 0.89$$

$$lift(B, \neg C) = \frac{1000 / 5000}{3000 / 5000 * 1250 / 5000} = 1.33$$

$$all_conf = \frac{sup(X)}{max_item_sup(X)}$$

Which Measures Should Be Used?

- 提升度和 χ^2 不是好的相关度量，对于大的交易数据库
- all-conf or coherence could be good measures (Omiecinski@TKDE'03)
- Over 20 interestingness measures have been proposed (see Tan, Kumar, Sritastava @KDD'02)
- Which are good ones?

symbol	measure	range	formula
ϕ	ϕ -coefficient	-1 ... 1	$\frac{P(A,B) - P(A)P(B)}{\sqrt{P(A)P(B)(1-P(A))(1-P(B))}}$
Q	Yule's Q	-1 ... 1	$\frac{P(A,B)P(\bar{A},\bar{B}) - P(A,\bar{B})P(\bar{A},B)}{P(A,B)P(\bar{A},\bar{B}) + P(A,\bar{B})P(\bar{A},B)}$
Y	Yule's Y	-1 ... 1	$\frac{\sqrt{P(A,B)P(\bar{A},\bar{B})} - \sqrt{P(A,\bar{B})P(\bar{A},B)}}{\sqrt{P(A,B)P(\bar{A},\bar{B})} + \sqrt{P(A,\bar{B})P(\bar{A},B)}}$
k	Cohen's	-1 ... 1	$\frac{P(A,B) + P(\bar{A},\bar{B}) - P(A)P(B) - P(\bar{A})P(\bar{B})}{1 - P(A)P(B) - P(\bar{A})P(\bar{B})}$
PS	Piatetsky-Shapiro's	-0.25 ... 0.25	$P(A,B) - P(A)P(B)$
F	Certainty factor	-1 ... 1	$\max\left(\frac{P(B A) - P(B)}{1 - P(B)}, \frac{P(A B) - P(A)}{1 - P(A)}\right)$
AV	added value	-0.5 ... 1	$\max(P(B A) - P(B), P(A B) - P(A))$
K	Klosgen's Q	-0.33 ... 0.38	$\sqrt{P(A,B) \max(P(B A) - P(B), P(A B) - P(A))}$
g	Goodman-kruskal's	0 ... 1	$\frac{\sum_j \max_k P(A_j, B_k) + \sum_k \max_j P(A_j, B_k) - \max_j P(A_j) - \max_k P(B_k)}{2 - \max_j P(A_j) - \max_k P(B_k)}$
M	Mutual Information	0 ... 1	$\frac{\sum_i \sum_j P(A_i, B_j) \log \frac{P(A_i, B_j)}{P(A_i)P(B_j)}}{\min(-\sum_i P(A_i) \log P(A_i), -\sum_i P(B_i) \log P(B_i))}$
J	J-Measure	0 ... 1	$\max(P(A, B) \log\left(\frac{P(B A)}{P(B)}\right) + P(\bar{A}\bar{B}) \log\left(\frac{P(\bar{B} \bar{A})}{P(\bar{B})}\right))$
G	Gini index	0 ... 1	$P(A, B) \log\left(\frac{P(A B)}{P(A)}\right) + P(\bar{A}\bar{B}) \log\left(\frac{P(\bar{A} \bar{B})}{P(\bar{A})}\right)$
s	support	0 ... 1	$\max(P(A)[P(B A)^2 + P(\bar{B} \bar{A})^2] + P(\bar{A})[P(B \bar{A})^2 + P(\bar{B} \bar{A})^2] - P(B)^2 - P(\bar{B})^2,$
c	confidence	0 ... 1	$P(B)[P(A B)^2 + P(\bar{A} \bar{B})^2] + P(\bar{B})[P(A \bar{B})^2 + P(\bar{A} \bar{B})^2] - P(A)^2 - P(\bar{A})^2)$
L	Laplace	0 ... 1	$P(A, B)$
IS	Cosine	0 ... 1	$\max(P(B A), P(A B))$
γ	coherence(Jaccard)	0 ... 1	$\max\left(\frac{NP(A,B)+1}{NP(A)+2}, \frac{NP(A,B)+1}{NP(B)+2}\right)$
α	all_confidence	0 ... 1	$\frac{P(A,B)}{\sqrt{P(A)P(B)}}$
o	odds ratio	0 ... ∞	$\frac{P(A,B)}{P(A)P(B)}$
V	Conviction	0.5 ... ∞	$\frac{P(A,B)}{P(A)+P(B)-P(A,B)}$
λ	lift	0 ... ∞	$\frac{P(A,B)}{P(A)P(B)}$
S	Collective strength	0 ... ∞	$\frac{\max(P(A), P(B))}{P(A,B)}$
χ^2	χ^2	0 ... ∞	$\frac{P(A,B)P(\bar{A},\bar{B})}{P(\bar{A},B)P(A,\bar{B})} \times \frac{1 - P(A)P(B) - P(\bar{A})P(\bar{B})}{1 - P(A,B) - P(\bar{A}\bar{B})}$

第5章：挖掘关联规则

- 关联规则挖掘
- 事务数据库中(单维布尔)关联规则挖掘的可伸缩算法
- 挖掘各种关联/相关规则
- 基于限制的关联挖掘
- 顺序模式挖掘
- 频繁模式挖掘的应用/扩展
- 小结

基于约束的数据挖掘

- 自动地找出数据库中的所有模式? — 不现实!
 - 模式可能太多, 并不聚焦!
- 数据挖掘应当是一个交互的过程
 - 用户使用数据挖掘查询语言 (或图形用户界面) 指导需要挖掘什么
- 基于约束的挖掘
 - 用户灵活性: 提供挖掘的约束
 - 系统优化: 考察限制, 寻找有效的挖掘—基于约束的挖掘

数据挖掘的约束

- 知识类型约束:
 - 分类, 关联, 等.
- 数据约束 (指定任务相关的数据集) — 使用类 SQL 查询
 - 找出 **Vancouver** 2000年12月份一起销售的产品对
- 维/层约束-指定数据属性/概念分层结构的层次
 - 关于 **region, price, brand, customer category**
- 兴趣度约束
 - 强规则 : $\text{min_support} \geq 3\%$, $\text{min_confidence} \geq 60\%$
- 规则 (或模式) 约束-指定规则形式
 - 小额销售 (价格 $< \$10$) 触发大额销售 ($\text{sum} > \200)

元规则制导挖掘 Meta-Rule Guided Mining

- 元规则是带有部分约束谓词和常量的规则

$$P_1(X, Y) \wedge P_2(X, W) \Rightarrow \text{buys}(X, \text{"iPad"})$$

- 一个导致的规则

$$\text{age}(X, \text{"15-25"}) \wedge \text{profession}(X, \text{"student"}) \Rightarrow \text{buys}(X, \text{"iPad"})$$

- 通常情况, 元规则如下形式的规则模板

$$P_1 \wedge P_2 \wedge \dots \wedge P_l \Rightarrow Q_1 \wedge Q_2 \wedge \dots \wedge Q_r$$

- 挖掘过程

- 找出所有的频繁 $(l+r)$ 谓词集 (基于最小支持度阈值)
 - 比须保留 l 子集的支持度/计数 (计算规则的置信度)
- (挖掘过程中) 尽可能推进约束(见约束推进技术)
- 尽可能地应用置信度, 相关和其他度量

规则约束-剪枝搜索空间

- 规则约束的分类
 - 反单调性**Anti-monotonic**
 - 单调性**Monotonic**
 - 简洁性 **Succinct:**
 - 可转变的**Convertible:**
 - 不可转变的

规则约束-反单调性

- 反单调性
 - 当项集 S 违反规则约束时, 它的任何超集合也违反约束
 - $sum(S.Price) \leq v$ 是反单调的
 - $sum(S.Price) \geq v$ 不是反单调的
- 例. $C: range(S.profit) \leq 15$ 是反单调的
 - 项集 ab 违反约束 C
 - ab 的每个超集也违反约束 C

TDB (min_sup=2)

TID	Transaction
10	a, b, c, d, f
20	b, c, d, f, g, h
30	a, c, d, e, f
40	c, e, f, g

Item	Profit
a	40
b	0
c	-20
d	10
e	-30
f	30
g	20
h	-10

规则约束-单调性

- 单调性
 - 当项集 S 满足约束时, 它的任何超集合也满足约束
 - $sum(S.Price) \geq v$ 是单调的
 - $min(S.Price) \leq v$ 是单调的
- 例. $C: range(S.profit) \geq 15$
 - 项集 ab 满足 C
 - ab 的每个超集合也满足 C

TDB (min_sup=2)

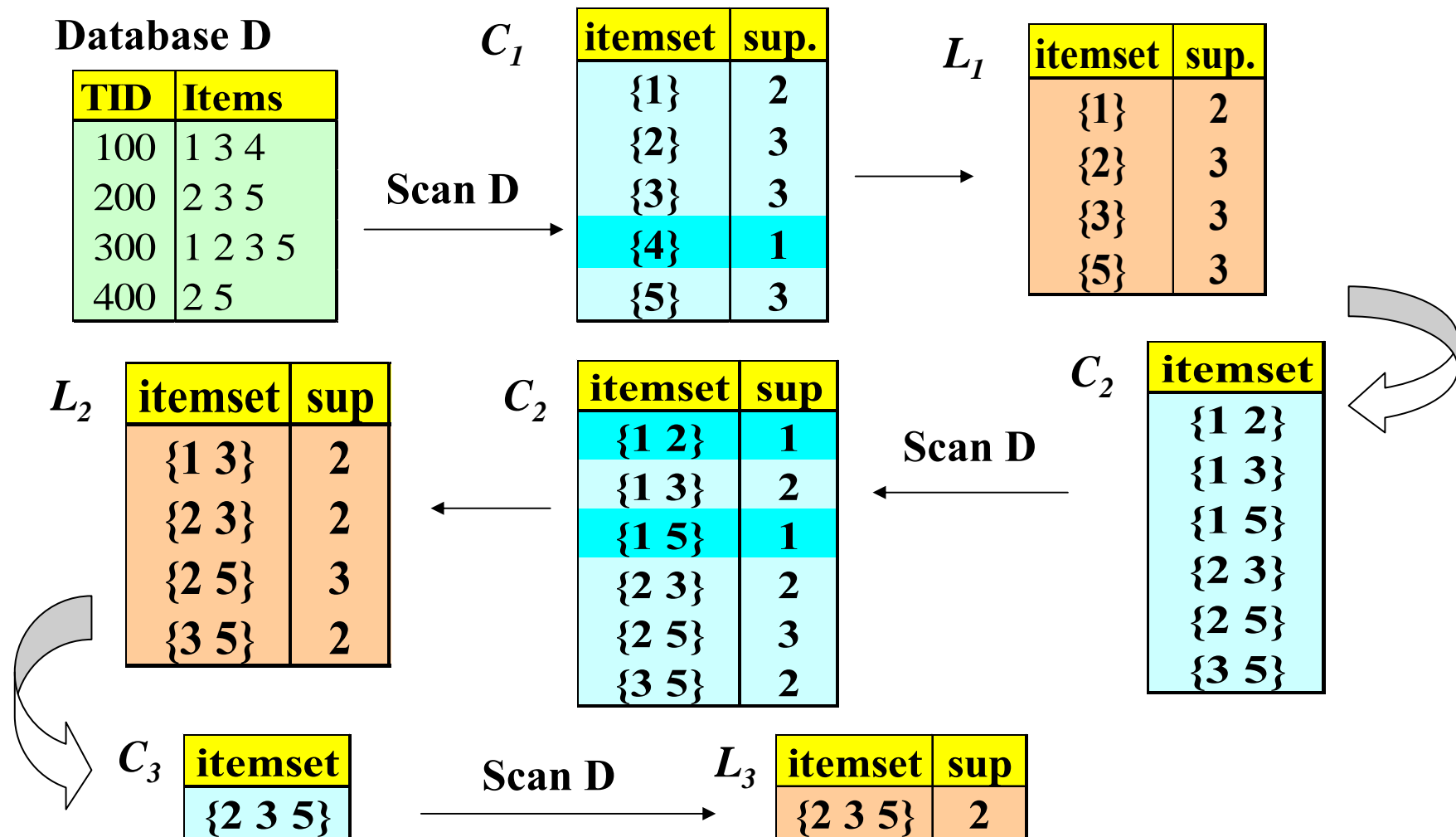
TID	Transaction
10	a, b, c, d, f
20	b, c, d, f, g, h
30	a, c, d, e, f
40	c, e, f, g

Item	Profit
a	40
b	0
c	-20
d	10
e	-30
f	30
g	20
h	-10

简洁性

- 简洁性:
 - 给定满足约束 C 的项的集合 A_I , 则满足 C 的任意集合 S 都基于 A_I , 即, S 包含一个属于 A_I 的子集
 - 思想: 不查看事务数据库, 项集 S 是否满足约束 C 可以根据选取的项确定
 - $\min(S.Price) \leq v$ 是简洁的
 - $\sum(S.Price) \geq v$ 不是简洁的
- 优化: 如果 C 是简洁的, C 是预计数可推进的(pre-counting pushable)

Apriori 算法 — 一个例子



朴素算法: Apriori + 约束

Database D

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D

C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

L_2

itemset	sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

C_2

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

Scan D

C_2

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

C_3

itemset
{2 3 5}

Scan D

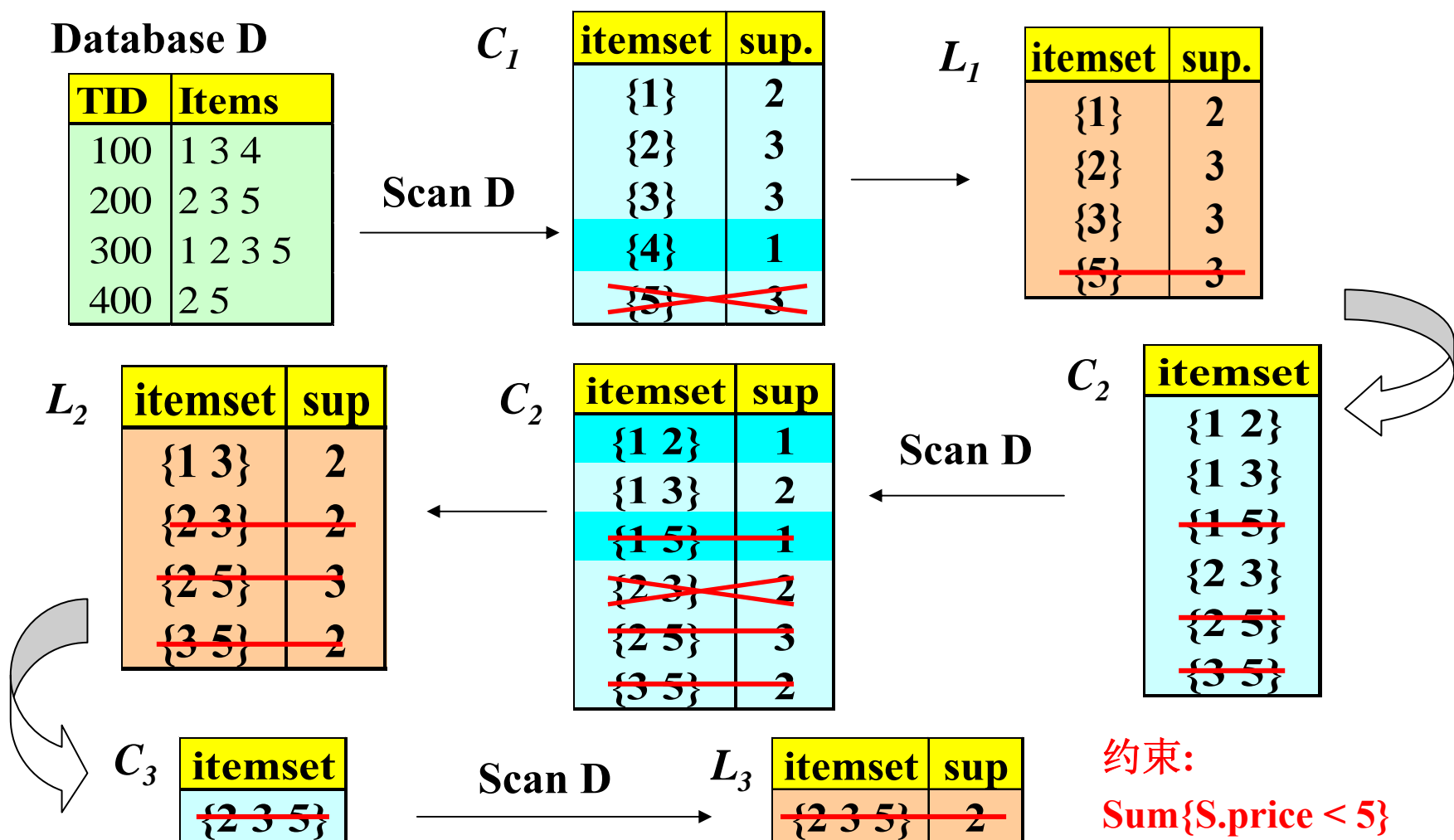
L_3

itemset	sup
{2 3 5}	2

约束:

Sum{S.price < 5}

受约束的Apriori 算法: 推进反单调约束



转换“强硬的”约束

- 通过将项适当地排序, 将强硬的约束转换成反单调的或单调的
- 例 C: $\text{avg}(S.\text{profit}) \geq 25$
 - 将项按profit值的递减序排序
 - $\langle a, f, g, d, b, h, c, e \rangle$
 - 如果项集 afb 违反 C
 - $afbh, afb^*$ 也违反 C
 - 约束C成为 反单调的!

TDB (min_sup=2)

TID	Transaction
10	a, b, c, d, f
20	b, c, d, f, g, h
30	a, c, d, e, f
40	c, e, f, g

Item	Profit
a	40
b	0
c	-20
d	10
e	-30
f	30
g	20
h	-10

可转变的约束

- 设 R 项集的项以特定次序安排,
- 可转变反单调
 - 如果项集 S 违反约束 C , 每个关于 R 以 S 为前缀的项集也违反约束 C
 - 例. $avg(S) \geq v$, 如果项值递减序排列
- 可转变单调
 - 如果项集 S 满足约束 C , 每个关于 R 以 S 为前缀的项集也满足约束 C .
 - 例. $avg(S) \geq v$, 如果项值递增序排列

强可转变约束

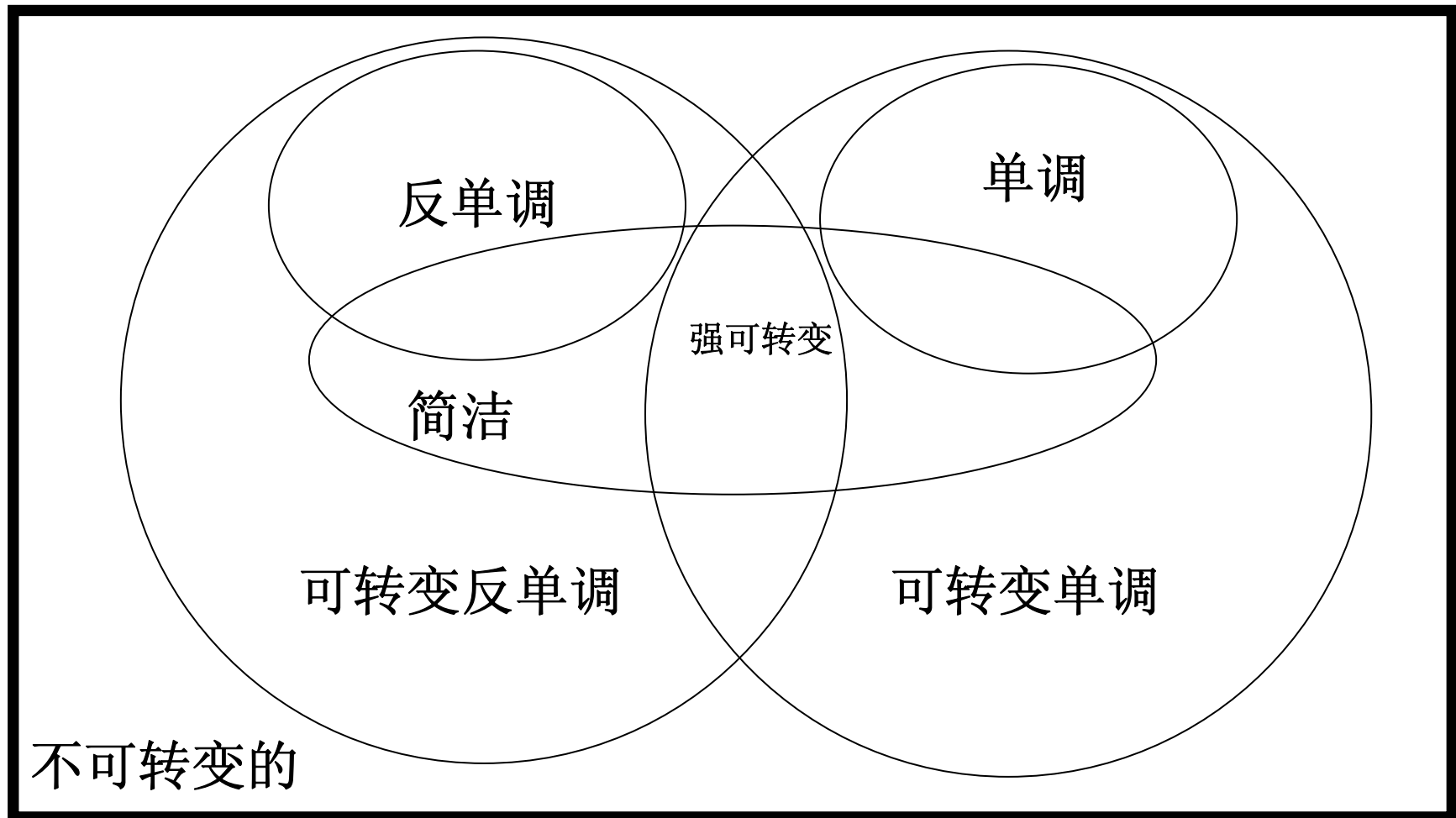
- $\text{avg}(X) \geq 25$ 关于项值的递减序 R :
 $\langle a, f, g, d, b, h, c, e \rangle$ 是可转变反单调的
 - 如果项集 af 违反约束 C , 每个以 af 为前缀的项集也违反 C , 如 afd
- $\text{avg}(X) \geq 25$ 关于项值的递增序 R^{-1} :
 $\langle e, c, h, b, d, g, f, a \rangle$ 是可转变单调的
 - 如果项集 d 满足约束 C , df 和 dfa 也满足, 它们具有前缀 d
- 这样, $\text{avg}(X) \geq 25$ 是 强可转变的

Item	Profit
a	40
b	0
c	-20
d	10
e	-30
f	30
g	20
h	-10

约束的性质汇总

约束	反单调	单调	简洁
$v \in S$	no	yes	yes
$S \supseteq V$	no	yes	yes
$S \subseteq V$	yes	no	yes
$\min(S) \leq v$	no	yes	yes
$\min(S) \geq v$	yes	no	yes
$\max(S) \leq v$	yes	no	yes
$\max(S) \geq v$	no	yes	yes
$\text{count}(S) \leq v$	yes	no	weakly
$\text{count}(S) \geq v$	no	yes	weakly
$\text{sum}(S) \leq v \ (a \in S, a \geq 0)$	yes	no	no
$\text{sum}(S) \geq v \ (a \in S, a \geq 0)$	no	yes	no
$\text{range}(S) \leq v$	yes	no	no
$\text{range}(S) \geq v$	no	yes	no
$\text{avg}(S) \theta v, \theta \in \{=, \leq, \geq\}$	convertible	convertible	no
$\text{support}(S) \geq \xi$	yes	no	no
$\text{support}(S) \leq \xi$	no	yes	no

约束的分类



Apriori 能够处理可转变的约束吗?

- 可转变的, 但既不是单调, 反单调, 也不是简洁的约束不能推进到 Apriori 挖掘算法的挖掘过程中
 - 在逐级的框架下, 不能做直接基于该约束的剪枝
 - 项集 df 违反 约束 $C: \text{avg}(X) \geq 25$
 - 由于 adf 满足 C , Apriori 需要 df 来组装 adf , 因此不能将 df 剪去
- 但是, 在模式增长框架下该约束可以推进到挖掘过程中!

Item	Value
a	40
b	0
c	-20
d	10
e	-30
f	30
g	20
h	-10

具有可转变约束的挖掘

- **C: $\text{avg}(X) \geq 25$, $\text{min_sup}=2$**
- 以值的递减序 **R**:
 $\langle a, f, g, d, b, h, c, e \rangle$
列出事务中的每一个项
 - 关于**R**, **C**是可转变反单调的
- 扫描 **TDB** 一次
 - 删除非频繁项
 - 项 **h** 被删除
 - 项 **a** 和 **f** 是好的, ...
- 基于投影的挖掘
 - 利用项投影的适当次序
 - 许多强硬的约束可以转变成(反)单调的

TDB ($\text{min_sup}=2$)

tem	Profit
a	40
f	30
g	20
d	10
b	0
h	-10
c	-20
e	-30

TID	Transaction
10	a, f, d, b, c
20	f, g, d, b, c
30	a, f, d, c, e
40	f, g, h, c, e

讨论—处理多个约束

- 不同的约束需要不同的, 甚至相互冲突的项序
- 如果存在序 R , 使得约束 C_1 和 C_2 关于 R 是可转变的, 则两个可转变的约束之间不存在冲突
- 如果项序存在冲突
 - 试图先满足一个约束
 - 然后使用另一约束的序, 在相应的投影数据库中挖掘频繁项集

文献: 频繁模式挖掘方法

- **R. Agarwal, C. Aggarwal, and V. V. V. Prasad. A tree projection algorithm for generation of frequent itemsets. Journal of Parallel and Distributed Computing, 2000.**
- **R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. SIGMOD'93, 207-216, Washington, D.C.**
- **R. Agrawal and R. Srikant. Fast algorithms for mining association rules. VLDB'94 487-499, Santiago, Chile.**
- **J. Han, J. Pei, and Y. Yin: “Mining frequent patterns without candidate generation”. In Proc. ACM-SIGMOD'2000, pp. 1-12, Dallas, TX, May 2000.**
- **H. Mannila, H. Toivonen, and A. I. Verkamo. Efficient algorithms for discovering association rules. KDD'94, 181-192, Seattle, WA, July 1994.**

文献: 频繁模式挖掘方法

- **A. Savasere, E. Omiecinski, and S. Navathe. An efficient algorithm for mining association rules in large databases. VLDB'95, 432-443, Zurich, Switzerland.**
- **C. Silverstein, S. Brin, R. Motwani, and J. Ullman. Scalable techniques for mining causal structures. VLDB'98, 594-605, New York, NY.**
- **R. Srikant and R. Agrawal. Mining generalized association rules. VLDB'95, 407-419, Zurich, Switzerland, Sept. 1995.**
- **R. Srikant and R. Agrawal. Mining quantitative association rules in large relational tables. SIGMOD'96, 1-12, Montreal, Canada.**
- **H. Toivonen. Sampling large databases for association rules. VLDB'96, 134-145, Bombay, India, Sept. 1996.**
- **M.J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. New algorithms for fast discovery of association rules. KDD'97. August 1997.**

文献: 频繁模式挖掘 (性能改进)

- **S. Brin, R. Motwani, J. D. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket analysis. SIGMOD'97, Tucson, Arizona, May 1997.**
- **D.W. Cheung, J. Han, V. Ng, and C.Y. Wong. Maintenance of discovered association rules in large databases: An incremental updating technique. ICDE'96, New Orleans, LA.**
- **T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama. Data mining using two-dimensional optimized association rules: Scheme, algorithms, and visualization. SIGMOD'96, Montreal, Canada.**
- **E.-H. Han, G. Karypis, and V. Kumar. Scalable parallel data mining for association rules. SIGMOD'97, Tucson, Arizona.**
- **J.S. Park, M.S. Chen, and P.S. Yu. An effective hash-based algorithm for mining association rules. SIGMOD'95, San Jose, CA, May 1995.**

文献: 频繁模式挖掘 (性能改进)

- **G. Piatetsky-Shapiro. Discovery, analysis, and presentation of strong rules. In G. Piatetsky-Shapiro and W. J. Frawley, Knowledge Discovery in Databases,. AAAI/MIT Press, 1991.**
- **J.S. Park, M.S. Chen, and P.S. Yu. An effective hash-based algorithm for mining association rules. SIGMOD'95, San Jose, CA.**
- **S. Sarawagi, S. Thomas, and R. Agrawal. Integrating association rule mining with relational database systems: Alternatives and implications. SIGMOD'98, Seattle, WA.**
- **K. Yoda, T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama. Computing optimized rectilinear regions for association rules. KDD'97, Newport Beach, CA, Aug. 1997.**
- **M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. Parallel algorithm for discovery of association rules. Data Mining and Knowledge Discovery, 1:343-374, 1997.**

文献: 频繁模式挖掘 (外延)

- S. Brin, R. Motwani, and C. Silverstein. Beyond market basket: Generalizing association rules to correlations. SIGMOD'97, 265-276, Tucson, Arizona.
- J. Han and Y. Fu. Discovery of multiple-level association rules from large databases. VLDB'95, 420-431, Zurich, Switzerland.
- M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A.I. Verkamo. Finding interesting rules from large sets of discovered association rules. CIKM'94, 401-408, Gaithersburg, Maryland.
- F. Korn, A. Labrinidis, Y. Kotidis, and C. Faloutsos. Ratio rules: A new paradigm for fast, quantifiable data mining. VLDB'98, 582-593, New York, NY.

文献: 频繁模式挖掘 (外延)

- **B. Lent, A. Swami, and J. Widom. Clustering association rules. ICDE'97, 220-231, Birmingham, England.**
- **R. Meo, G. Psaila, and S. Ceri. A new SQL-like operator for mining association rules. VLDB'96, 122-133, Bombay, India.**
- **R.J. Miller and Y. Yang. Association rules over interval data. SIGMOD'97, 452-461, Tucson, Arizona.**
- **A. Savasere, E. Omiecinski, and S. Navathe. Mining for strong negative associations in a large database of customer transactions. ICDE'98, 494-502, Orlando, FL, Feb. 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, 1-12, Seattle, Washington.**
- **J. Pei, A.K.H. Tung, J. Han. Fault-Tolerant Frequent Pattern Mining: Problems and Challenges. SIGMOD DMKD'01, Santa Barbara, CA.**

文献: 挖掘最大模式和闭项集

- **R. J. Bayardo. Efficiently mining long patterns from databases. SIGMOD'98, 85-93, Seattle, Washington.**
- **J. Pei, J. Han, and R. Mao, "CLOSET: An Efficient Algorithm for Mining Frequent Closed Itemsets", Proc. 2000 ACM-SIGMOD Int. Workshop on Data Mining and Knowledge Discovery (DMKD'00), Dallas, TX, May 2000.**
- **N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. ICDT'99, 398-416, Jerusalem, Israel, Jan. 1999.**
- **M. Zaki. Generating Non-Redundant Association Rules. KDD'00. Boston, MA. Aug. 2000**
- **M. Zaki. CHARM: An Efficient Algorithm for Closed Association Rule Mining, SIAM'02**

文献: 基于约束的频繁模式挖掘

- **G. Grahne, L. Lakshmanan, and X. Wang. Efficient mining of constrained correlated sets. ICDE'00, 512-521, San Diego, CA, Feb. 2000.**
- **Y. Fu and J. Han. Meta-rule-guided mining of association rules in relational databases. KDOOD'95, 39-46, Singapore, Dec. 1995.**
- **J. Han, L. V. S. Lakshmanan, and R. T. Ng, "Constraint-Based, Multidimensional Data Mining", COMPUTER (special issues on Data Mining), 32(8): 46-50, 1999.**
- **L. V. S. Lakshmanan, R. Ng, J. Han and A. Pang, "Optimization of Constrained Frequent Set Queries with 2-Variable Constraints", SIGMOD'99**

文献: 基于约束的频繁模式挖掘

- **R. Ng, L.V.S. Lakshmanan, J. Han & A. Pang. “Exploratory mining and pruning optimizations of constrained association rules.” SIGMOD’98**
- **J. Pei, J. Han, and L. V. S. Lakshmanan, "Mining Frequent Itemsets with Convertible Constraints", Proc. 2001 Int. Conf. on Data Engineering (ICDE'01), April 2001.**
- **J. Pei and J. Han "Can We Push More Constraints into Frequent Pattern Mining?", Proc. 2000 Int. Conf. on Knowledge Discovery and Data Mining (KDD'00), Boston, MA, August 2000.**
- **R. Srikant, Q. Vu, and R. Agrawal. Mining association rules with item constraints. KDD'97, 67-73, Newport Beach, California.**

文献: 序列模式挖掘方法

- **R. Agrawal and R. Srikant. Mining sequential patterns. ICDE'95, 3-14, Taipei, Taiwan.**
- **R. Srikant and R. Agrawal. Mining sequential patterns: Generalizations and performance improvements. EDBT'96.**
- **J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal, M.-C. Hsu, "FreeSpan: Frequent Pattern-Projected Sequential Pattern Mining", Proc. 2000 Int. Conf. on Knowledge Discovery and Data Mining (KDD'00), Boston, MA, August 2000.**
- **H. Mannila, H Toivonen, and A. I. Verkamo. Discovery of frequent episodes in event sequences. Data Mining and Knowledge Discovery, 1:259-289, 1997.**

文献: 序列模式挖掘方法

- **J. Pei, J. Han, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu, "PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth", Proc. 2001 Int. Conf. on Data Engineering (ICDE'01), Heidelberg, Germany, April 2001.**
- **B. Ozden, S. Ramaswamy, and A. Silberschatz. Cyclic association rules. ICDE'98, 412-421, Orlando, FL.**
- **S. Ramaswamy, S. Mahajan, and A. Silberschatz. On the discovery of interesting patterns in association rules. VLDB'98, 368-379, New York, NY.**
- **M.J. Zaki. Efficient enumeration of frequent sequences. CIKM'98. November 1998.**
- **M.N. Garofalakis, R. Rastogi, K. Shim: SPIRIT: Sequential Pattern Mining with Regular Expression Constraints. VLDB 1999: 223-234, Edinburgh, Scotland.**

文献: 空间, 多媒体, 文本和 Web 数据 库频繁模式挖掘

- **K. Koperski, J. Han, and G. B. Marchisio, "Mining Spatial and Image Data through Progressive Refinement Methods", *Revue internationale de gomatique (European Journal of GIS and Spatial Analysis)*, 9(4):425-440, 1999.**
- **A. K. H. Tung, H. Lu, J. Han, and L. Feng, "Breaking the Barrier of Transactions: Mining Inter-Transaction Association Rules", *Proc. 1999 Int. Conf. on Knowledge Discovery and Data Mining (KDD'99)*, San Diego, CA, Aug. 1999, pp. 297-301.**
- **J. Han, G. Dong and Y. Yin, "Efficient Mining of Partial Periodic Patterns in Time Series Database", *Proc. 1999 Int. Conf. on Data Engineering (ICDE'99)*, Sydney, Australia, March 1999, pp. 106-115.**

文献: 空间, 多媒体, 文本和 Web 数据库频繁模式挖掘

- H. Lu, L. Feng, and J. Han, "Beyond Intra-Transaction Association Analysis: Mining Multi-Dimensional Inter-Transaction Association Rules", ACM Transactions on Information Systems (TOIS'00), 18(4): 423-454, 2000.
- O. R. Zaiane, M. Xin, J. Han, "Discovering Web Access Patterns and Trends by Applying OLAP and Data Mining Technology on Web Logs," Proc. Advances in Digital Libraries Conf. (ADL'98), Santa Barbara, CA, April 1998, pp. 19-29
- O. R. Zaiane, J. Han, and H. Zhu, "Mining Recurrent Items in Multimedia with Progressive Resolution Refinement", Proc. 2000 Int. Conf. on Data Engineering (ICDE'00), San Diego, CA, Feb. 2000, pp. 461-470.

文献: 用于分类和数据方计算的频繁模式挖掘

- K. Beyer and R. Ramakrishnan. Bottom-up computation of sparse and iceberg cubes. SIGMOD'99, 359-370, Philadelphia, PA, June 1999.
- M. Fang, N. Shivakumar, H. Garcia-Molina, R. Motwani, and J. D. Ullman. Computing iceberg queries efficiently. VLDB'98, 299-310, New York, NY, Aug. 1998.
- J. Han, J. Pei, G. Dong, and K. Wang, “Computing Iceberg Data Cubes with Complex Measures”, Proc. ACM-SIGMOD'2001, Santa Barbara, CA, May 2001.
- M. Kamber, J. Han, and J. Y. Chiang. Metarule-guided mining of multi-dimensional association rules using data cubes. KDD'97, 207-210, Newport Beach, California.
- K. Beyer and R. Ramakrishnan. Bottom-up computation of sparse and iceberg cubes. SIGMOD'99
- T. Imielinski, L. Khachiyan, and A. Abdulghani. Cubegrades: Generalizing association rules. Technical Report, Aug. 2000

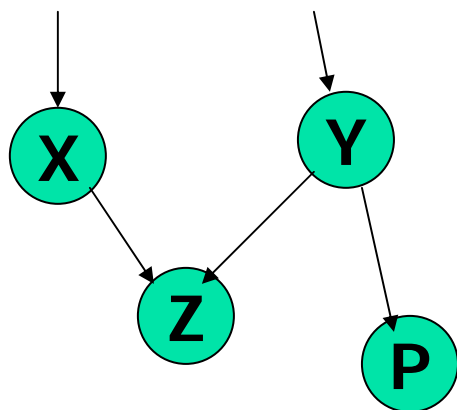
Chapter 6. 分类: Advanced Methods



- 贝叶斯信念网络
- 后向传播分类 Classification by Backpropagation
- 支持向量机 Support Vector Machines
- Classification by Using Frequent Patterns
- Lazy Learners (or Learning from Your Neighbors)
- 其他分类方法
- Additional Topics Regarding Classification
- Summary

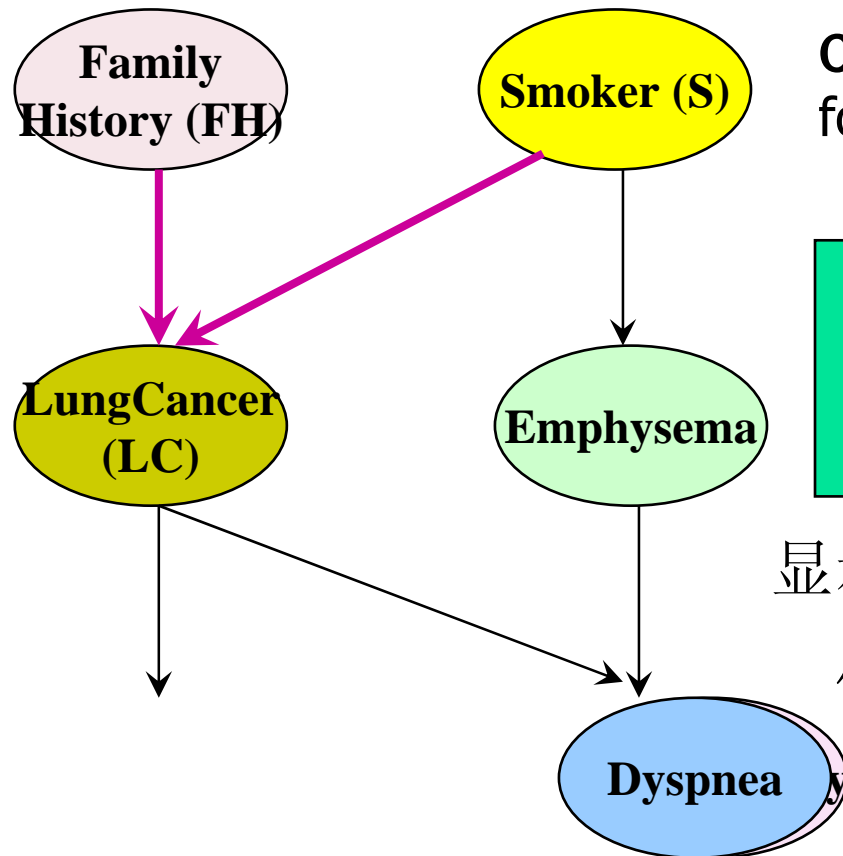
贝叶斯信念网络

- **Bayesian belief networks** (又称为 **Bayesian networks, probabilistic networks**): 允许变量子集间定义类条件独立
- (有向无环) 因果关系的图模型
 - 表示变量间的依赖关系
 - 给出了一个联合概率分布



- Nodes: 随机变量
- Links: 依赖关系
- X,Y 是Z的双亲, Y is the parent of P
- Z 和 P间没有依赖关系
- 没有环

贝叶斯信念网络: An Example



CPT: Conditional Probability Table
for variable LungCancer:

	(FH, S)	(FH, ~S)	(~FH, S)	(~FH, ~S)
LC	0.8	0.5	0.7	0.1
~LC	0.2	0.5	0.3	0.9


显示父母的每个可能组合的条件概率
从CPT推倒 **X**的特定值得概率

$$P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i \mid \text{Parents}(Y_i))$$

训练贝叶斯网路:几种方案

- Scenario 1: 给定网络结构和所有变量观察: 只计算CPT
- Scenario 2: 网络结构已知, 某些变量隐藏: 梯度下降法(贪心爬山), i.e., 沿着准则函数的最速下降方向搜索解
 - 权重初始化为随机值
 - 每次迭代中, 似乎是对目前的最佳解决方案前进, 没有回溯
 - 每次迭代中权重被更新, 并且收敛到局部最优解
- Scenario 3: 网络结构未知, 所有变量可知: 搜索模型空间构造网络拓扑
- Scenario 4: 未知结构, 隐藏变量: 目前没有好的算法
- D. Heckerman. [A Tutorial on Learning with Bayesian Networks](#). In *Learning in Graphical Models*, M. Jordan, ed.. MIT Press, 1999.

Chapter 6. 分类: Advanced Methods

- Bayesian Belief Networks
- Classification by Backpropagation 
- Support Vector Machines
- Classification by Using Frequent Patterns
- Lazy Learners (or Learning from Your Neighbors)
- Other Classification Methods
- Additional Topics Regarding Classification
- Summary

用反向传播分类

- 反向传播：一种神经网络学习算法
- 最早是由心理学家和神经学家开创的，开发和测试神经元计算模拟
- 神经网络：一组连接的输入/输出单元，其中每个连接都与一个权重关联
- 通过调整权重来学习，能够输入元组的正确类别标号
- 又被称为连接者学习 **connectionist learning**

神经网络作为分类器

- 弱点
 - 学习时间很长
 - 需要很多参数（常靠经验确定），如网络的结构
 - 可解释性差：很难解释权重和网络中“隐藏单元”的含义
- 优势
 - 对噪音数据的高承受能力
 - 分类未经训练的模式的能力
 - 非常适合处理连续值的输入/输出
 - 成功地应用于现实数据, **e.g.**, 手写字符识别
 - 算法是固有并行的
 - 已经发展了一些从训练好的神经网络提取规则的技术

多层前馈神经网络

Output vector

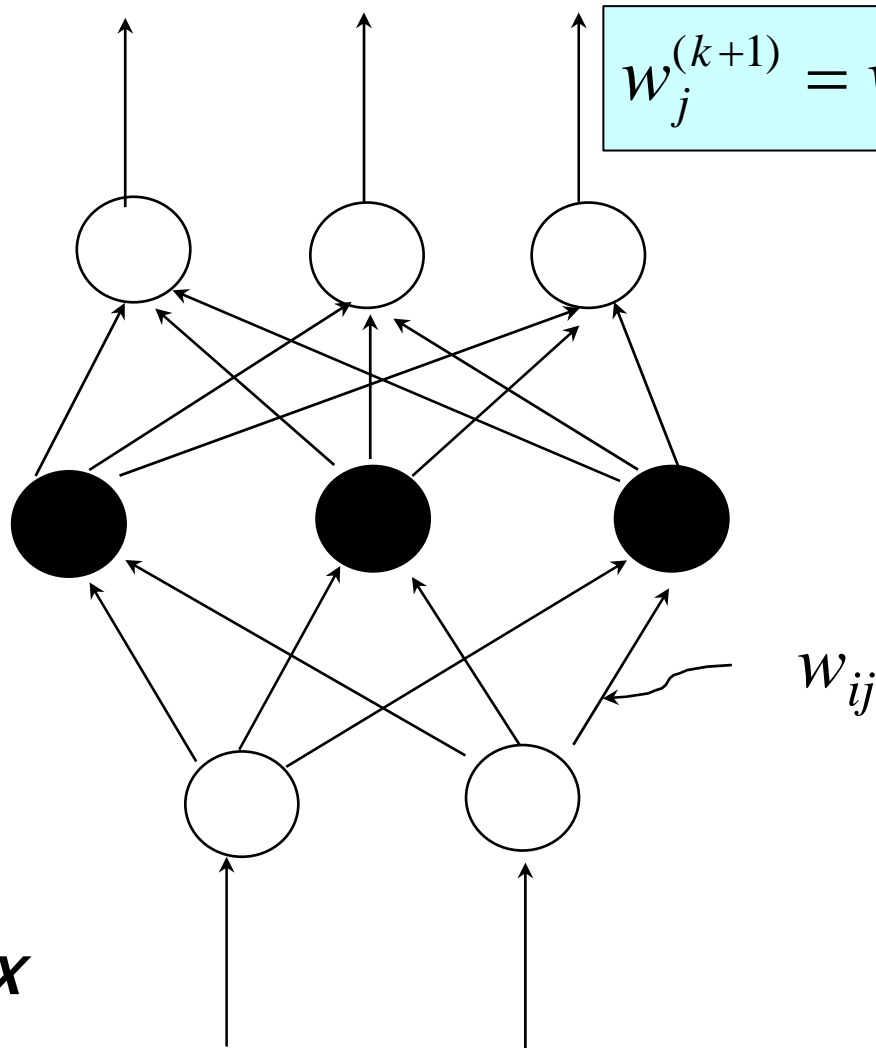
输出层

隐藏层

输入层

Input vector: X

$$w_j^{(k+1)} = w_j^{(k)} + \lambda(y_i - \hat{y}_i^{(k)})x_{ij}$$



多层前馈神经网络

- 网络的输入对应于每个训练元组的测量属性
 - 输入同时传给称作输入层的单元
- 加权后同时传递给隐藏层
- 隐藏层的数目是任意的, 通常只有一个
- 最后一个隐藏层的输出权重后作为输入传递给称为输出层, 此处给出网络的预测
- 前馈**feed-forward**: 权重都不反馈到输入单元或前一层的输出单元
- 从统计学观点, 网络进行一个非线性回归; 给定足够的隐藏单元和训练数据, 可以逼近任何函数

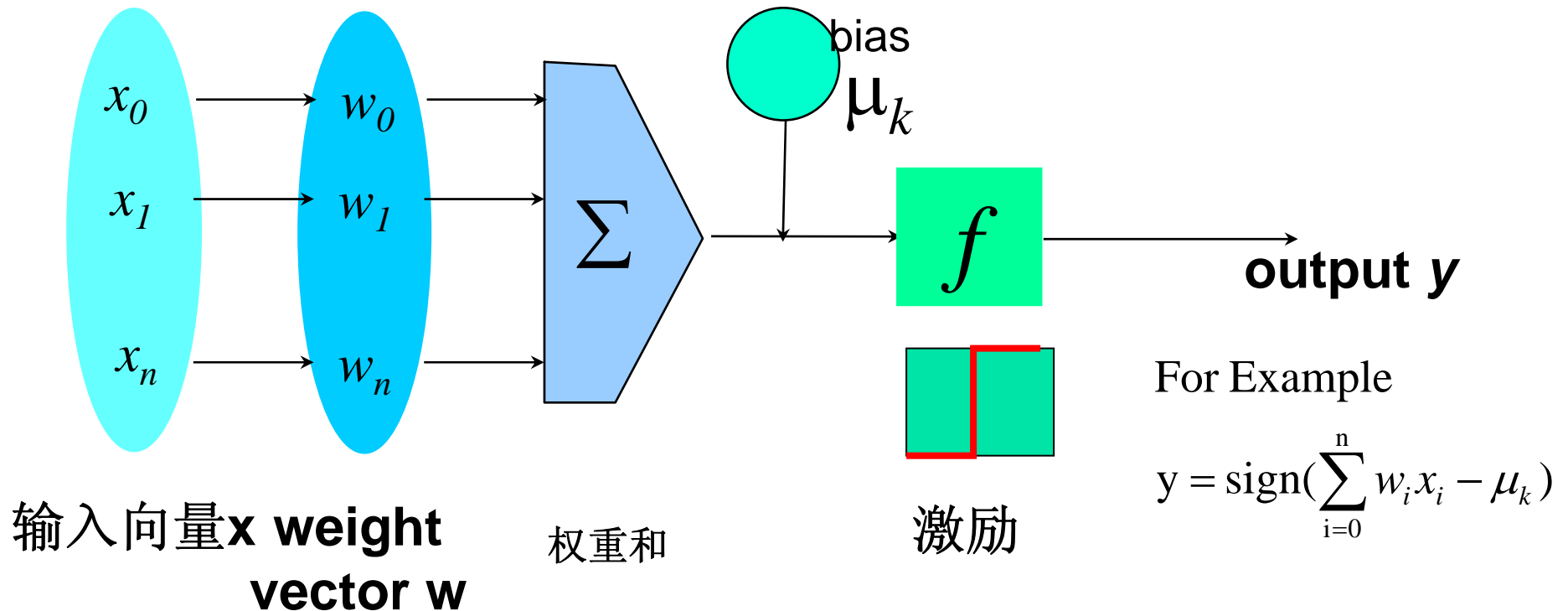
定义网络拓扑

- 确定网络拓扑: 给定输入层的单元数, 隐藏层数(if > 1), 每个隐藏层的单元数, 输出层的单元数
- 规格化训练元组的输入值 [0.0—1.0]
 - 对于离散值, 可重新编码, 每个可能的值一个输入单元并初始化0
- 输出, 如果涉及超过两个类别则一个输出单元对应一个类别
- 一旦一个训练好的网络其准确率达不到要求时, 用不同的网络拓扑和初始值重新训练网络

反向传播Backpropagation

- 迭代地处理训练数据 & 比较网络的预测值和实际的目标值
- 对每个训练元组, 修改权重最小化目标的预测值和实际值之间的**mean squared error**
- 这种修改后向进行: 从输出层开始, 通过每个隐藏层直到第一个隐藏层
- 步骤
 - 初始化权重为一个小的随机数, 以及偏倚 **biases**
 - 向前传播输入 (应用激励函数)
 - 向后传播误差 (更新权重和偏倚)
 - 停止条件 (当误差非常小, etc.)

神经元：一个隐藏/输出层单元



- 一个 n -维输入向量 \mathbf{x} 被映射到变量 y ，通过非线性函数映射
- 单元的输入是前一层的输出。被乘上权重后求和且加上此单元的偏倚。然后应用一个非线性激励函数。

后向传播算法


```
1) 初始化 network 的权和偏置。
2) while 终止条件不满足 {
3) for samples 中的每个训练样本 X {
4) // 向前传播输入
5) for 隐藏或输出层每个单元 j {
6)  $I_j = \sum_i w_{ij} O_i + \theta_j$ ; // 相对于前一层 i, 计算单元 j 的净
   输入
7)  $O_j = 1 / (1 + e^{-I_j})$ ; } // 计算单元 j 的输出
8) // 后向传播误差
9) for 输出层每个单元 j
10)  $Err_j = O_j(1 - O_j)(T_j - O_j)$ ; // 计算误差
11) for 由最后一个到第一个隐藏层, 对于隐藏层每个单元 j
12)  $Err_j = O_j(1 - O_j) \sum_k Err_k w_{kj}$ ; // 计算关于下一个较高层 k 的误差
13) for networ 中每个权  $w_{ij}$  {
14)  $\Delta w_{ij} = (l) Err_j O_i$ ; // 权增值
15)  $w_{ij} = w_{ij} + \Delta w_{ij}$ ; } // 权更新
16) for networ 中每个偏差  $\theta_j$  {
17)  $\Delta \theta_j = (l) Err_j$ ; // 偏差增值
18)  $\theta_j = \theta_j + \Delta \theta_j$ ; } // 偏差更新
19) }}
```

图 7.9 后向传播算法

效率和可解释性

- 向后传播的效率: 每次迭代 $O(|D| * w)$, $|D|$ 为元组数, w 个权重, 最坏的情况下迭代的次数可能是元组数的指数
- 为了更容易理解: 通过网络修剪提取规则
 - 简化网络结构, 去除对训练的网络有最弱影响的权重连接
 - 对连接, 单元, or 活跃值聚类
 - 输入和活跃值集合用来推导描述输入和隐藏层间关系的规则
- Sensitivity analysis: 评估一个给定的输入变量对网络输出的影响。从中获得的知识可以表示为规则。
 - IF X 减少5% THEN Y增加...

Chapter 6. 分类: Advanced Methods

- 贝叶斯信念网络
- 后向传播分类 Classification by Backpropagation
- 支持向量机 Support Vector Machines 
- Classification by Using Frequent Patterns
- Lazy Learners (or Learning from Your Neighbors)
- 其他分类方法
- Additional Topics Regarding Classification
- Summary

分类:一个数学映射

- **Classification:** 预测分类的类标签

- E.g., 个人主页分类

- $x_i = (x_1, x_2, x_3, \dots)$, $y_i = +1$ or -1

- x_1 : # of word “homepage”

- x_2 : # of word “welcome”

- $x \in X = \mathcal{R}^n$, $y \in Y = \{+1, -1\}$,

- 推导一个函数 $f: X \rightarrow Y$

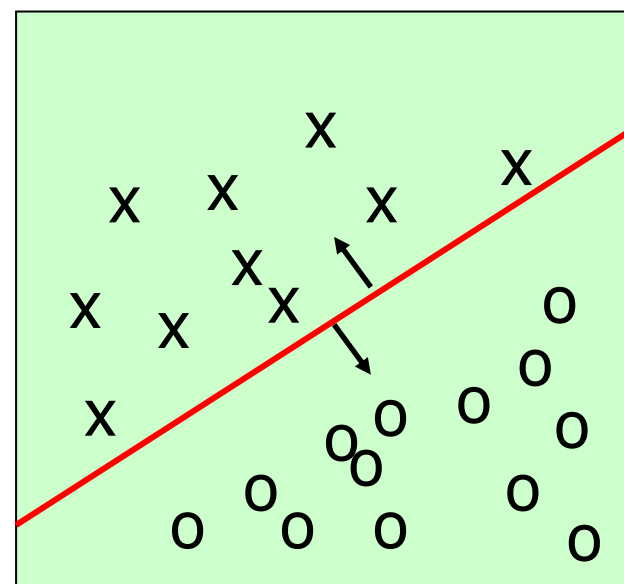
- 线性分类

- 二元分类问题

- 红线上面的点属于 class ‘x’

- 下面的点属于 class ‘o’

- Examples: SVM, Perceptron, Probabilistic Classifiers



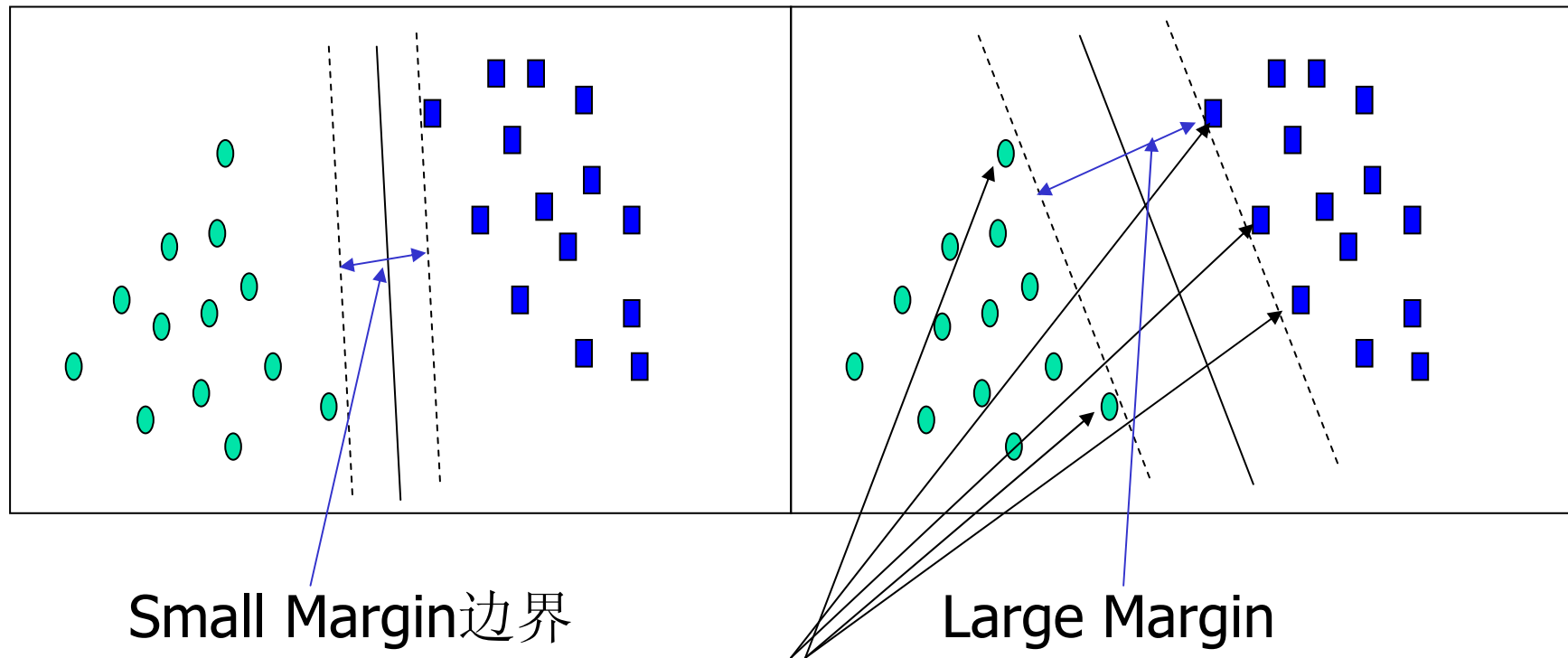
SVM—Support Vector Machines

- 一个相对新的分类方法，适用于linear and nonlinear data
- 使用一个非线性映射把原始训练数据变换到高维空间中
- 在新的维上, 搜索线性优化分离超平面**hyperplane** (i.e., “决策边界”)
- 用一个合适的足够高维的映射, 两类数据总是可以被超平面分开
- SVM 使用**support vectors** (“基本” 选练元组) 和边缘 **margins** (由支持向量定义)发现超平面

SVM—历史和应用

- Vapnik and colleagues (1992)—基础工作来自于Vapnik & Chervonenkis' statistical learning theory in 1960s
- Features: 训练慢但是准确度高，由于能够建模非线性决策边界 (margin maximization)
- Used for: 分类和数值预测
- 应用:
 - 手写数字识别, object recognition, speaker identification, 基准时间序列预测检验

支持向量机的一般哲学

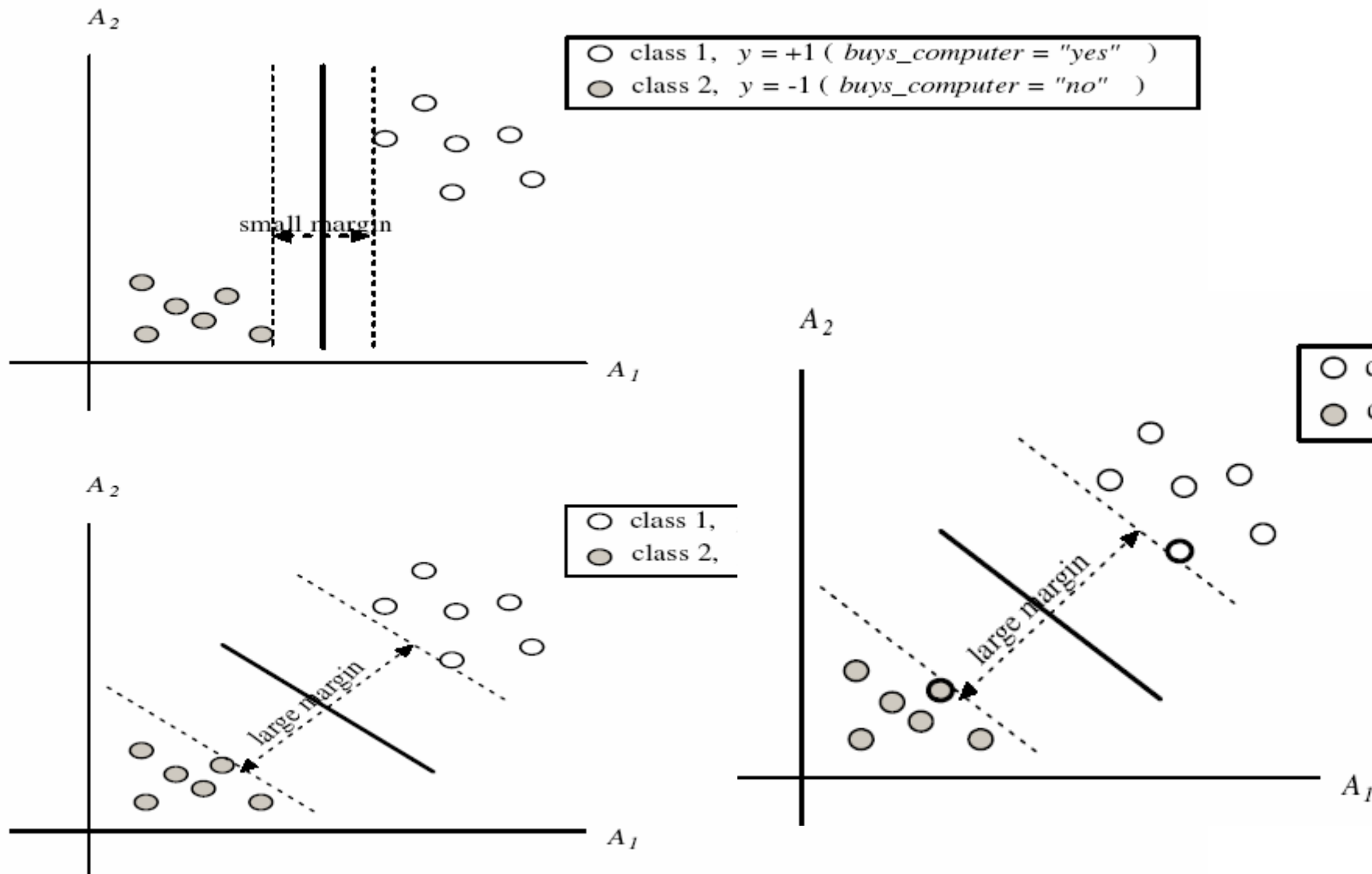


Small Margin边界

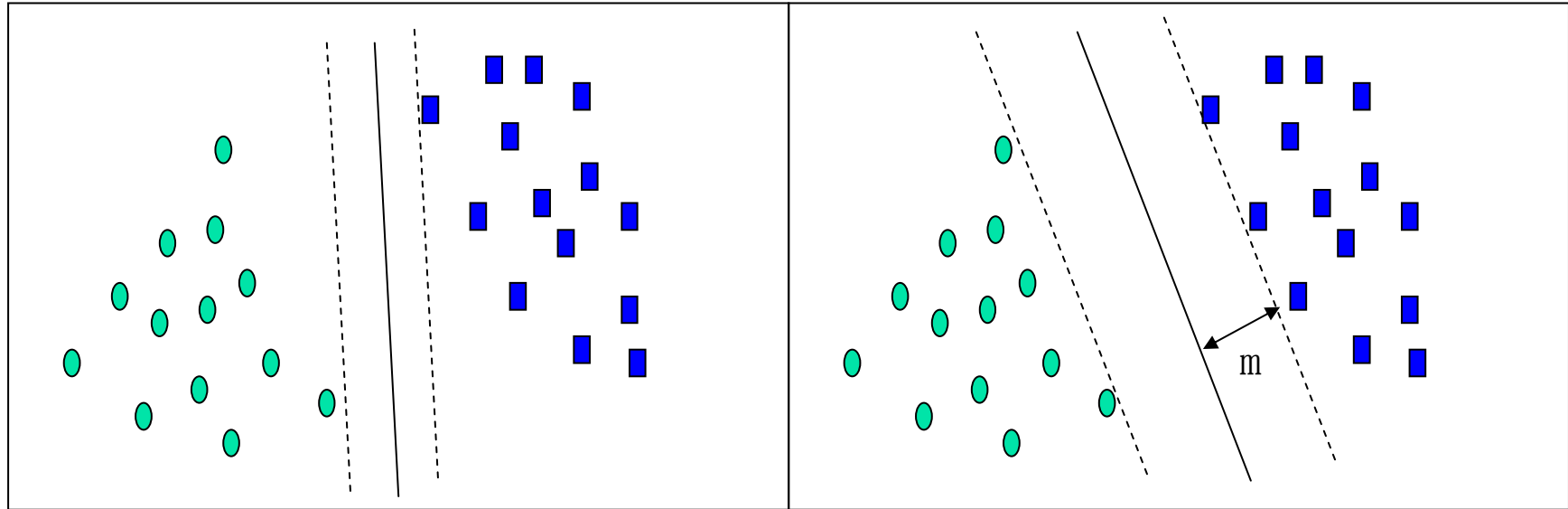
Large Margin

Support Vectors

SVM—Margins and Support Vectors



SVM—当数据线性可分时



D 为 $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_{|D|}, y_{|D|})$, 其中 \mathbf{x}_i 带标签 y_i 的训练元组

有无数条直线(hyperplanes) 可以分离两个类, 但我们需要发现最好的一个(对未知数据有最小化的分类误差)

SVM searches for the hyperplane with the largest margin, i.e., maximum marginal hyperplane (MMH)

SVM—线性可分

- 一个分离超平面可以写成

$$\mathbf{W} \bullet \mathbf{X} + b = 0$$

$\mathbf{W} = \{w_1, w_2, \dots, w_n\}$ 权重向量和标量 b (bias)

- 对于2-D, 可以写成

$$w_0 + w_1 x_1 + w_2 x_2 = 0$$

- 超平面定义了边缘的边界:

$$H_1: w_0 + w_1 x_1 + w_2 x_2 \geq 1 \quad \text{for } y_i = +1, \text{ and}$$

$$H_2: w_0 + w_1 x_1 + w_2 x_2 \leq -1 \quad \text{for } y_i = -1$$

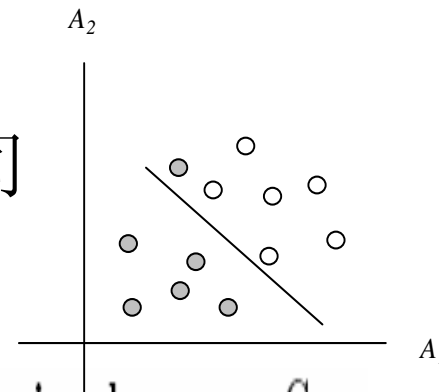
- 任何一个位于超平面 H_1 or H_2 (i.e., the sides defining the margin) 的样本为 **support vectors**
- 最大边缘是 $2 / \|\mathbf{w}\| \rightarrow \max$
- 是一个 **constrained (convex) quadratic optimization problem**:
二次目标函数和线性约束 \rightarrow *Quadratic Programming (QP)* \rightarrow Lagrangian multipliers

Why Is SVM Effective on High Dimensional Data?

- 训练后的分类器的**complexity**由支持向量数而不是数据维度刻画
- 支持向量**support vectors**是基本的/临界的训练元组——离决策边界最近 (MMH)
- 如果其他的样本删掉后重新训练仍然会发现相同的分离超平面
- 支持向量的数目可用于计算（**svm**分类器）期望误差率的上界 (upper), 其独立于数据维度
- 一个只有少量支持向量的**svm**有很好的推广性能, 即使数据的维度很高时

SVM—线性不可分

- 把原始输入数据变换到一个更高维的空间



Example 6.8 Nonlinear transformation of original input data into a higher dimensional space. Consider the following example. A 3D input vector $\mathbf{X} = (x_1, x_2, x_3)$ is mapped into a 6D space Z using the mappings $\phi_1(\mathbf{X}) = x_1, \phi_2(\mathbf{X}) = x_2, \phi_3(\mathbf{X}) = x_3, \phi_4(\mathbf{X}) = (x_1)^2, \phi_5(\mathbf{X}) = x_1x_2$, and $\phi_6(\mathbf{X}) = x_1x_3$. A decision hyperplane in the new space is $d(\mathbf{Z}) = \mathbf{WZ} + b$, where \mathbf{W} and \mathbf{Z} are vectors. This is linear. We solve for \mathbf{W} and b and then substitute back so that we see that the linear decision hyperplane in the new (\mathbf{Z}) space corresponds to a nonlinear second order polynomial in the original 3-D input space,

$$\begin{aligned} d(\mathbf{Z}) &= w_1x_1 + w_2x_2 + w_3x_3 + w_4(x_1)^2 + w_5x_1x_2 + w_6x_1x_3 + b \\ &= w_1z_1 + w_2z_2 + w_3z_3 + w_4z_4 + w_5z_5 + w_6z_6 + b \end{aligned}$$

- Search for a linear separating hyperplane in the new space

SVM: 不同的核函数

- 计算变换后数据的点积, 数学上等价于应用一个核函数 $K(\mathbf{X}_i, \mathbf{X}_j)$ 于原始数据, i.e., $K(\mathbf{X}_i, \mathbf{X}_j) = \Phi(\mathbf{X}_i) \cdot \Phi(\mathbf{X}_j)$
- Typical Kernel Functions

Polynomial kernel of degree h : $K(\mathbf{X}_i, \mathbf{X}_j) = (\mathbf{X}_i \cdot \mathbf{X}_j + 1)^h$

Gaussian radial basis function kernel : $K(\mathbf{X}_i, \mathbf{X}_j) = e^{-\|\mathbf{X}_i - \mathbf{X}_j\|^2 / 2\sigma^2}$

Sigmoid kernel : $K(\mathbf{X}_i, \mathbf{X}_j) = \tanh(\kappa \mathbf{X}_i \cdot \mathbf{X}_j - \delta)$

- SVM 也可用于多类数据 (> 2)和回归分析(需要附加参数)

SVM vs. Neural Network

■ SVM

- Deterministic algorithm
- Nice generalization properties
- Hard to learn – 使用 quadratic programming techniques 批量学习
- Using kernels can learn very complex functions

■ Neural Network

- Nondeterministic algorithm
- Generalizes well but doesn't have strong mathematical foundation
- Can easily be learned in incremental fashion
- To learn complex functions—use multilayer perceptron (nontrivial)

SVM Related Links

- SVM Website: <http://www.kernel-machines.org/>
- Representative implementations
 - **LIBSVM**: an efficient implementation of SVM, multi-class classifications, nu-SVM, one-class SVM, including also various interfaces with java, python, etc.
 - **SVM-light**: simpler but performance is not better than LIBSVM, support only binary classification and only in C
 - **SVM-torch**: another recent implementation also written in C

Chapter 6. 惰性学习

- Bayesian Belief Networks
- Classification by Backpropagation
- Support Vector Machines
- Classification by Using Frequent Patterns
- Lazy Learners (or Learning from Your Neighbors)
- Other Classification Methods
- Additional Topics Regarding Classification
- Summary



Lazy vs. Eager Learning

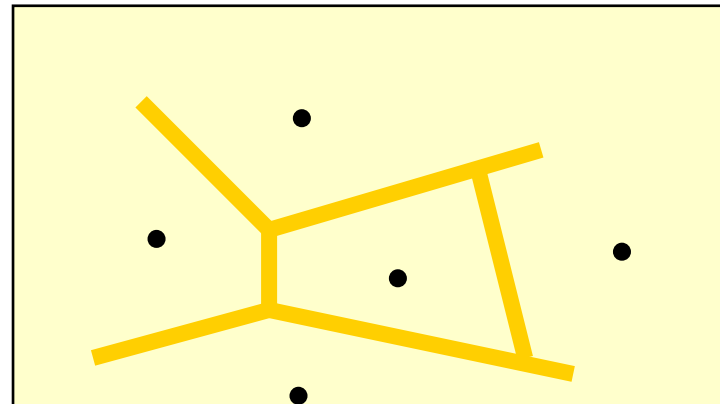
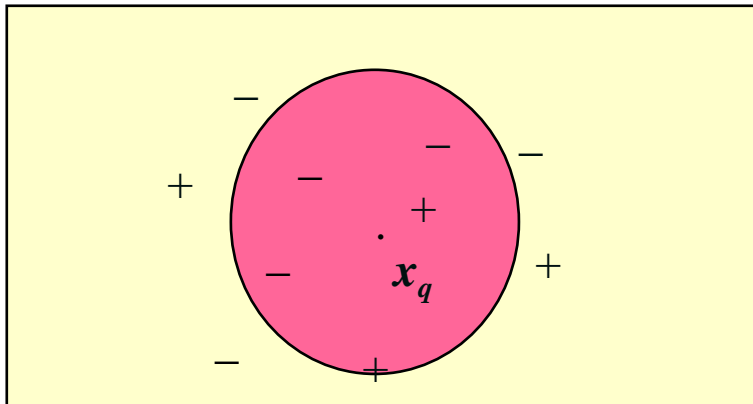
- Lazy vs. eager learning
 - **Lazy learning** (e.g., 基于实例的学习): 仅存储数据 (或稍加处理) 直到碰到检验元组才开始处理
 - **Eager learning** (前面介绍的方法): 给定训练数据, 在遇到待处理的新数据前构造分类模型
- **Lazy:** 训练用时很少, 预测用时多
- 准确性
 - 惰性学习方法可以有效地利用更丰富的假设空间, 使用多个局部线性函数来对目标函数形成一个隐式的全局逼近
 - **Eager:** 必须限于一个假设, 它覆盖了整个实例空间

Lazy Learner:基于实例的方法

- Instance-based learning:
 - Store training examples and delay the processing (“lazy evaluation”) until a new instance must be classified
- 典型的方法
 - k-nearest neighbor approach
 - 实例表示为欧氏空间中的点.
 - Locally weighted regression
 - Constructs local approximation
 - 基于案例的推理Case-based reasoning
 - 使用符号表示和知识为基础的推理

k -最近邻算法

- 所有的样本对应于 n -D 空间的点
- 通过Euclidean distance, $\text{dist}(\mathbf{X}_1, \mathbf{X}_2)$ 定义最近邻居
- 目标函数可以是discrete- or real- 值
- 对于离散值, k -NN 返回与目标元组最近的 k 个训练样本的多数类
- Voronoi diagram: the decision surface induced by 1-NN for a typical set of training examples




k -NN Algorithm的讨论

- k -NN: 元组的未知实值的预测时
 - 返回与未知元组 k 个最近邻居的平均值（对应属性）
- Distance-weighted nearest neighbor algorithm
 - 根据与目标元组的距离权重组合 k 个近邻的贡献
 - Give greater weight to closer neighbors $w \equiv \frac{1}{d(x_q, x_i)^2}$
- Robust to noisy data by averaging k -nearest neighbors
- Curse of dimensionality: 邻居间的距离会被无关联的属性影响
 - 坐标轴伸缩或去除次要的属性

基于案例的推理 (CBR)

- **CBR:** 使用一个问题解的数据库来求解新问题
- 存储符号描述(tuples or cases)—不是Euclidean space的点
- 应用: 顾客-服务台 (产品有关的诊断), 合法裁决
- Methodology
 - 实例表示为复杂的符号描述(e.g., function graphs)
 - 搜索相似的案例, 组合多个返回的例子
 - Tight coupling between case retrieval, knowledge-based reasoning, and problem solving
- Challenges
 - Find a good similarity metric
 - Indexing based on syntactic similarity measure, and when failure, backtracking, and adapting to additional cases

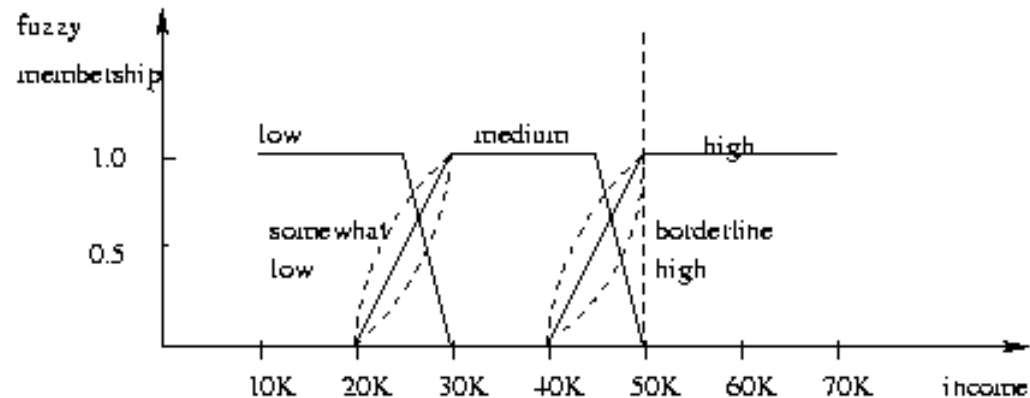
Chapter 6. 分类: 其他方法

- Bayesian Belief Networks
- Classification by Backpropagation
- Support Vector Machines
- Classification by Using Frequent Patterns
- Lazy Learners (or Learning from Your Neighbors)
- Other Classification Methods 
- Additional Topics Regarding Classification
- Summary

遗传算法 (GA)


- Genetic Algorithm: 模仿生物进化
- 使用随机产生的规则组成一个最初的**population**
 - 每个规则有一系列位表示
 - E.g., if A_1 and $\neg A_2$ then C_2 can be encoded as 100
 - 如果一个属性有 $k > 2$ 个值, 使用k位
- 基于适者生存原理, 最适合的规则及其后代组成新的种群
- 规则的拟合度用它在训练样本的准确率来评估
- 通过交叉和突变来产生后代
- 此过程持续下去, 直到种群**P**进化到其中的每个规则满足给定的拟合度阈值
- 算法慢, 但易于并行

Fuzzy Set Approaches



- Fuzzy logic 使用 $[0.0, 1.0]$ 真值来表示类的成员的隶属度
- 属性值被转化成模糊值. Ex.:
 - 对于每个离散类别收入{low, medium, high}, x 被分配一个模糊的隶属值, e.g. \$49K 属于 "medium income" 0.15, 属于 "high income" 的隶属值是0.96
 - 模糊隶属值的和不一定等于1.
- 每个可用的规则为类的隶属贡献一票
- 通常, 对每个预测分类的真值求和, 并组合这些值

Chapter 6. 分类: Advanced Methods

- Bayesian Belief Networks
- Classification by Backpropagation
- Support Vector Machines
- Classification by Using Frequent Patterns
- Lazy Learners (or Learning from Your Neighbors)
- Other Classification Methods
- Additional Topics Regarding Classification 
- Summary

多类分类

- 分类时设计多个类别 (i.e., > 2 Classes)
- Method 1. **One-vs.-all (OVA)**: 每次学习一个分类器
 - 给定 m 个类, 训练 m 个分类器, 每个类别一个
 - 分类器 j : 把类别 j 的元组定义为 *positive* & 其他的为 *negative*
 - 为分类样本 \mathbf{X} , 所有分类器投票来集成
- Method 2. **All-vs.-all (AVA)**: 为每一对类别学习一个分类器
 - Given m classes, construct $m(m-1)/2$ binary classifiers
 - 使用两个类别的元组训练一个分类器
 - 为分类元组 \mathbf{X} , 每个分类器投票. \mathbf{X} is assigned to the class with maximal vote
- Comparison
 - All-vs.-all tends to be superior to one-vs.-all
 - Problem: Binary classifier is sensitive to errors, and errors affect vote count

多类分类的Error-Correcting Codes

- 最初目的是在数据传输的通讯任务中通过探索数据冗余来修正误差。例：

- A 7-bit codeword associated with classes 1-4

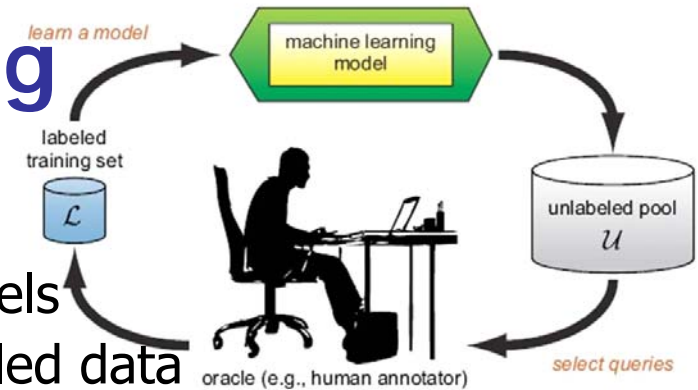
Class	Error-Corr. Codeword						
C_1	1	1	1	1	1	1	1
C_2	0	0	0	0	1	1	1
C_3	0	0	1	1	0	0	1
C_4	0	1	0	1	0	1	0

- 给定未知元组 \mathbf{X} , 7个分类器的结果为: 0001010
- Hamming distance: # 两个码字间不同位数的和
- $H(\mathbf{X}, C_1) = 5$, 检查 $[1111111]$ & $[0001010]$ 间不同位数和
- $H(\mathbf{X}, C_2) = 3$, $H(\mathbf{X}, C_3) = 3$, $H(\mathbf{X}, C_4) = 1$, thus C_4 as the label for \mathbf{X}
- Error-correcting codes can correct up to $(h-1)/h$ 1-bit error, where h is the minimum Hamming distance between any two codewords
- If we use 1-bit per class, it is equiv. to one-vs.-all approach, the code are insufficient to self-correct
- When selecting error-correcting codes, there should be good row-wise and col.-wise separation between the codewords

半监督分类

- Semi-supervised: 使用有标签和无标签数据构造分类器
- Self-training:
 - Build a classifier using the labeled data
 - Use it to label the unlabeled data, and those with the most confident label prediction are added to the set of labeled data
 - 重复以上过程
 - Adv: 容易理解; disadv: 可能增大误差
- Co-training: Use two or more classifiers to teach each other
 - 每个学习者使用元组的相互独立的特征集合来训练一个好的分类器 F_1
 - 然后 f_1 and f_2 用来预测未知元组 X 的类别标签
 - Teach each other: The tuple having the most confident prediction from f_1 is added to the set of labeled data for f_2 , & vice versa
- Other methods, e.g., joint probability distribution of features and labels

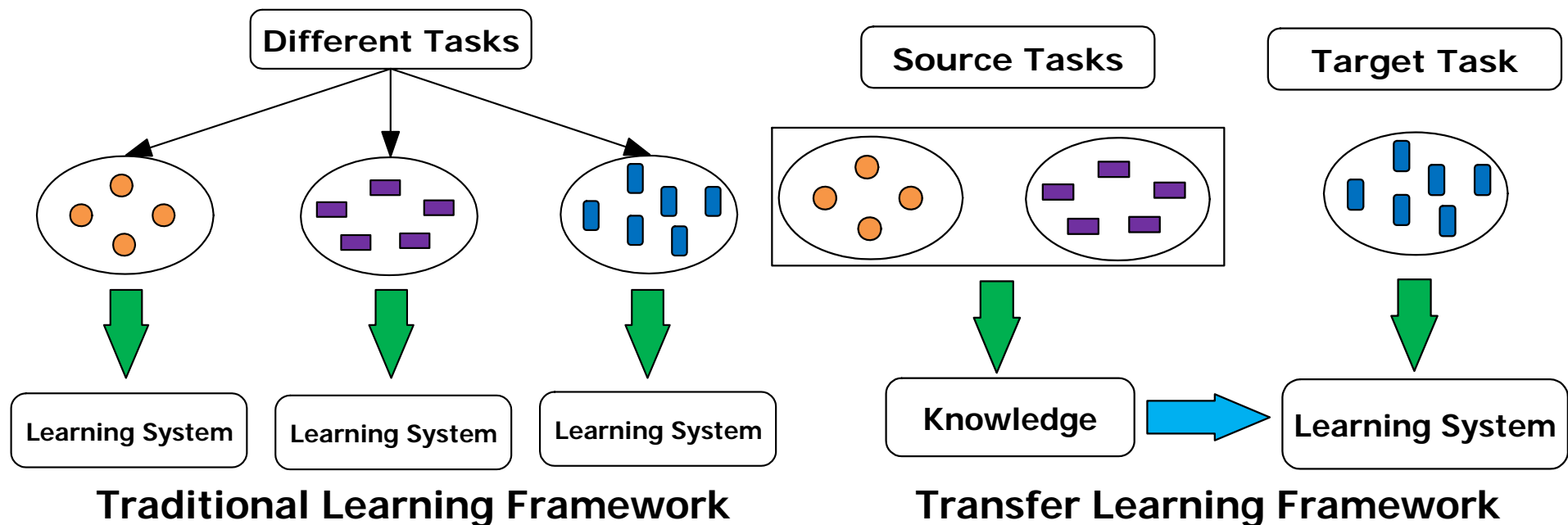
主动学习 Active Learning



- 获取类标签是昂贵
- Active learner: query human (oracle) for labels
- Pool-based approach: Uses a pool of unlabeled data
 - \mathcal{L} : \mathcal{D} 中有标签的样本子集, \mathcal{U} : \mathcal{D} 的一个未标记数据集
 - 使用一个查询函数小心地从 \mathcal{U} 选择1或多个元组, 并咨询标签an oracle (a human annotator)
 - The newly labeled samples are added to \mathcal{L} , and learn a model
 - Goal: Achieve high accuracy using as few labeled data as possible
- Evaluated using *learning curves*: Accuracy as a function of the number of instances queried (# of tuples to be queried should be small)
- Research issue: How to choose the data tuples to be queried?
 - Uncertainty sampling: choose the least certain ones
 - Reduce *version space*, the subset of hypotheses consistent w. the training data
 - Reduce expected entropy over \mathcal{U} : Find the greatest reduction in the total number of incorrect predictions

迁移学习：概念框架


- Transfer learning: Extract knowledge from one or more source tasks and apply the knowledge to a target task
- Traditional learning: 每一个任务建立分类器
- Transfer learning: Build new classifier by applying existing knowledge learned from source tasks



迁移学习: Methods and Applications

- 应用:数据过时或分布的变化时, e.g., Web document classification, e-mail spam filtering
- *Instance-based transfer learning*: Reweight some of the data from source tasks and use it to learn the target task
- TrAdaBoost (Transfer AdaBoost)
 - 假定源和目标数据用相同的属性和类别描述, but rather diff. distributions
 - Require only labeling a small amount of target data
 - 训练中使用源数据: When a source tuple is misclassified, reduce the weight of such tuples so that they will have less effect on the subsequent classifier
- Research issues
 - Negative transfer: When it performs worse than no transfer at all
 - Heterogeneous transfer learning: Transfer knowledge from different feature space or multiple source domains
 - Large-scale transfer learning

Chapter 6. 分类:频繁模式

- Bayesian Belief Networks
- Classification by Backpropagation
- Support Vector Machines
- Classification by Using Frequent Patterns 
- Lazy Learners (or Learning from Your Neighbors)
- Other Classification Methods
- Additional Topics Regarding Classification
- Summary

关联分类

- 关联分类: 主要步骤

- 挖掘关于频繁模式(属性-值对的联结) 和类标签间的强关联
- 产生以下形似的关联规则

$$P_1 \wedge p_2 \dots \wedge p_l \rightarrow "A_{\text{class}} = C" (\text{conf}, \text{sup})$$

- 组织规则, 形成基于规则的分类器

- 为什么有效?

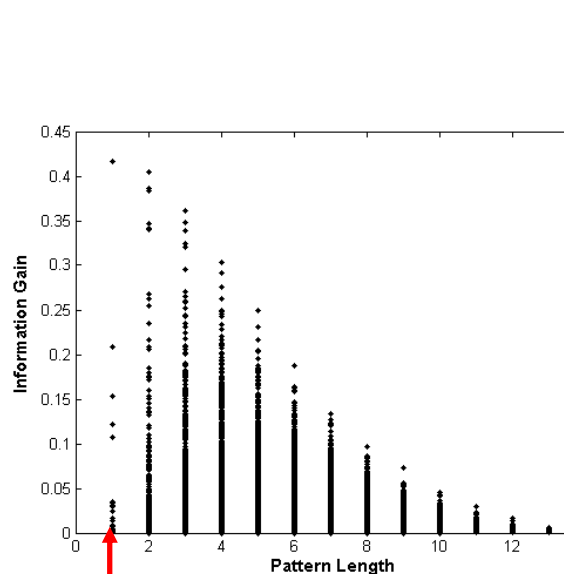
- 可以发现 (在多个属性间) 高置信度的关联, 可以克服决策树规约引入的约束, 决策树一次考虑一个属性
- 研究发现, 关联分类通常比某些传统的分类方法更精确, 例如C4.5

典型的关联分类方法

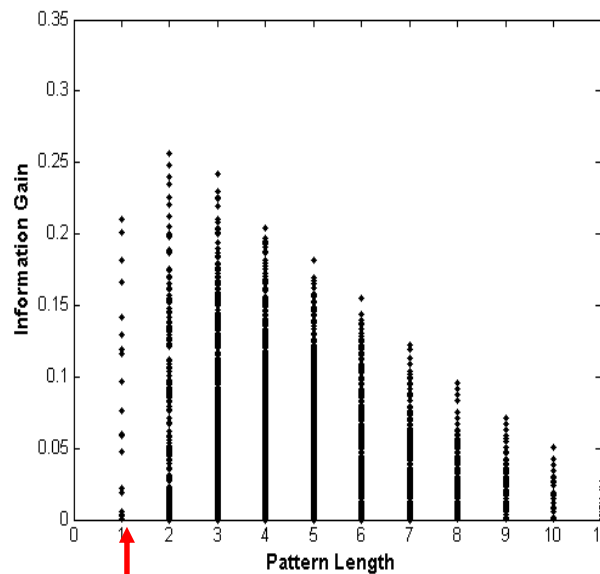
- **CBA** (Classification Based on Associations: Liu, Hsu & Ma, KDD'98)
 - 挖掘可能关联规则: Cond-set (属性-值 的集合) \rightarrow class label
 - 建立分类器: 基于置信度和支持度的下降序组织规则
- **CMAR** (Classification based on Multiple Association Rules: Li, Han, Pei, ICDM'01)
 - 分类: 多个规则的统计分析
- **CPAR** (Classification based on Predictive Association Rules: Yin & Han, SDM'03)
 - 产生预测性规则 (FOIL-like analysis) 允许覆盖的元组以降低权重形式保留下来构造新规则
 - (根据期望准确率) 使用最好的k 个规则预测
 - 更有效 (产生规则少), 精确性类似CMAR

频繁模式 vs. 单个特征

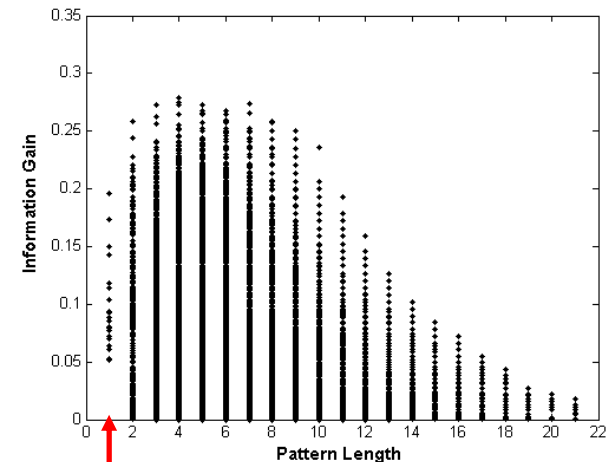
某些频繁模式的判别能力高于单个特征.



(a) Austral



(b) Cleve



(c) Sonar

Fig. 1. Information Gain vs. Pattern Length

经验结果

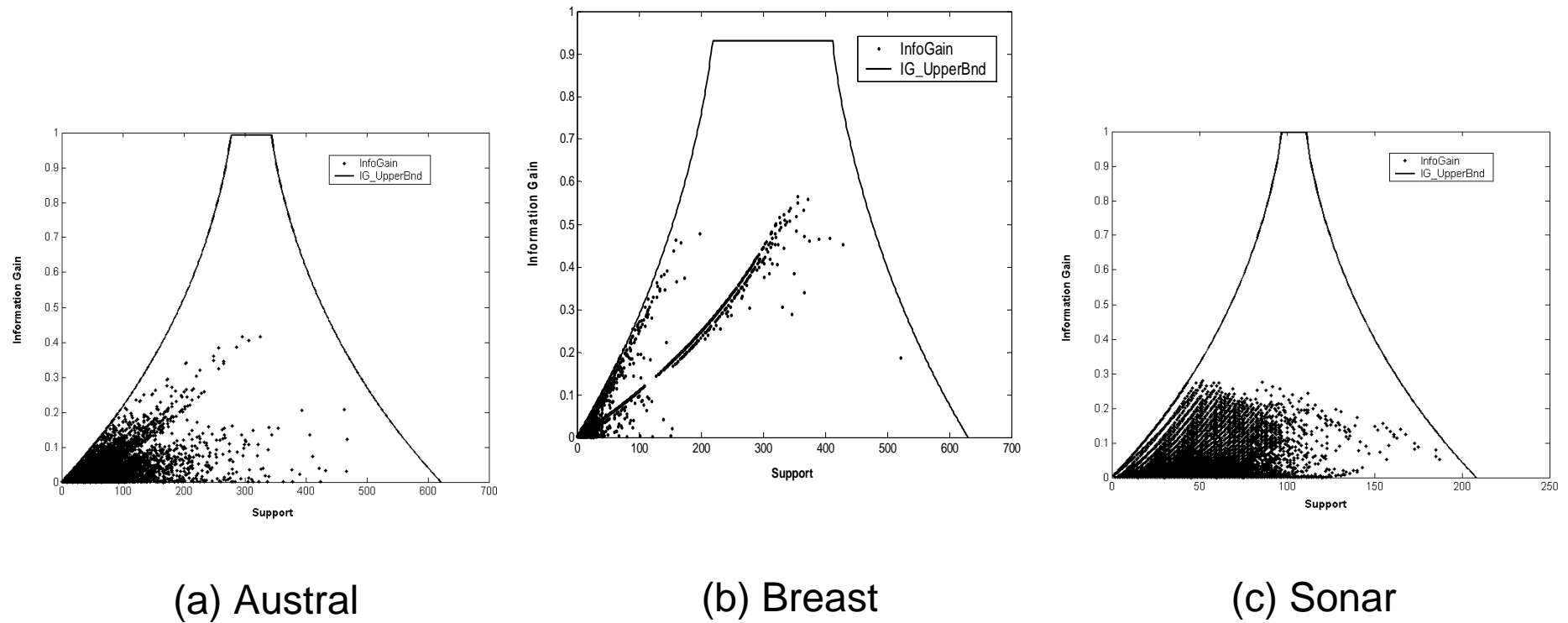


Fig. 2. Information Gain vs. Pattern Frequency

特征选择Feature Selection

- 给定频繁模式集合, 存在non-discriminative和redundant 的模式, 他们会引起过度拟合
- 我们希望选出discriminative patterns, 并且去除冗余
- 借用Maximal Marginal Relevance (MMR)的概念
 - A document has high marginal relevance if it is both relevant to the query and contains minimal marginal similarity to previously selected documents

实验结果

Table 1. Accuracy by SVM on Frequent Combined Features vs. Single Features

Data	Single Feature			Freq. Pattern	
	<i>Item_All</i>	<i>Item_FS</i>	<i>Item_RBF</i>	<i>Pat_All</i>	<i>Pat_FS</i>
anneal	99.78	99.78	99.11	99.33	99.67
austral	85.01	85.50	85.01	81.79	91.14
auto	83.25	84.21	78.80	74.97	90.79
breast	97.46	97.46	96.98	96.83	97.78
cleve	84.81	84.81	85.80	78.55	95.04
diabetes	74.41	74.41	74.55	77.73	78.31
glass	75.19	75.19	74.78	79.91	81.32
heart	84.81	84.81	84.07	82.22	88.15
hepatic	84.50	89.04	85.83	81.29	96.83
horse	83.70	84.79	82.36	82.35	92.39
iono	93.15	94.30	92.61	89.17	95.44
iris	94.00	96.00	94.00	95.33	96.00
labor	89.99	91.67	91.67	94.99	95.00
lymph	81.00	81.62	84.29	83.67	96.67
pima	74.56	74.56	76.15	76.43	77.16
sonar	82.71	86.55	82.71	84.60	90.86
vehicle	70.43	72.93	72.14	73.33	76.34
wine	98.33	99.44	98.33	98.30	100
zoo	97.09	97.09	95.09	94.18	99.00

Table 2. Accuracy by C4.5 on Frequent Combined Features vs. Single Features

Dataset	Single Features		Frequent Patterns	
	<i>Item_All</i>	<i>Item_FS</i>	<i>Pat_All</i>	<i>Pat_FS</i>
anneal	98.33	98.33	97.22	98.44
austral	84.53	84.53	84.21	88.24
auto	71.70	77.63	71.14	78.77
breast	95.56	95.56	95.40	96.35
cleve	80.87	80.87	80.84	91.42
diabetes	77.02	77.02	76.00	76.58
glass	75.24	75.24	76.62	79.89
heart	81.85	81.85	80.00	86.30
hepatic	78.79	85.21	80.71	93.04
horse	83.71	83.71	84.50	87.77
iono	92.30	92.30	92.89	94.87
iris	94.00	94.00	93.33	93.33
labor	86.67	86.67	95.00	91.67
lymph	76.95	77.62	74.90	83.67
pima	75.86	75.86	76.28	76.72
sonar	80.83	81.19	83.67	83.67
vehicle	70.70	71.49	74.24	73.06
wine	95.52	93.82	96.63	99.44
zoo	91.18	91.18	95.09	97.09

Scalability Tests

Table 3. Accuracy & Time on Chess Data

<i>min_sup</i>	#Patterns	Time (s)	SVM (%)	C4.5 (%)
1	N/A	N/A	N/A	N/A
2000	68,967	44.703	92.52	97.59
2200	28,358	19.938	91.68	97.84
2500	6,837	2.906	91.68	97.62
2800	1,031	0.469	91.84	97.37
3000	136	0.063	91.90	97.06

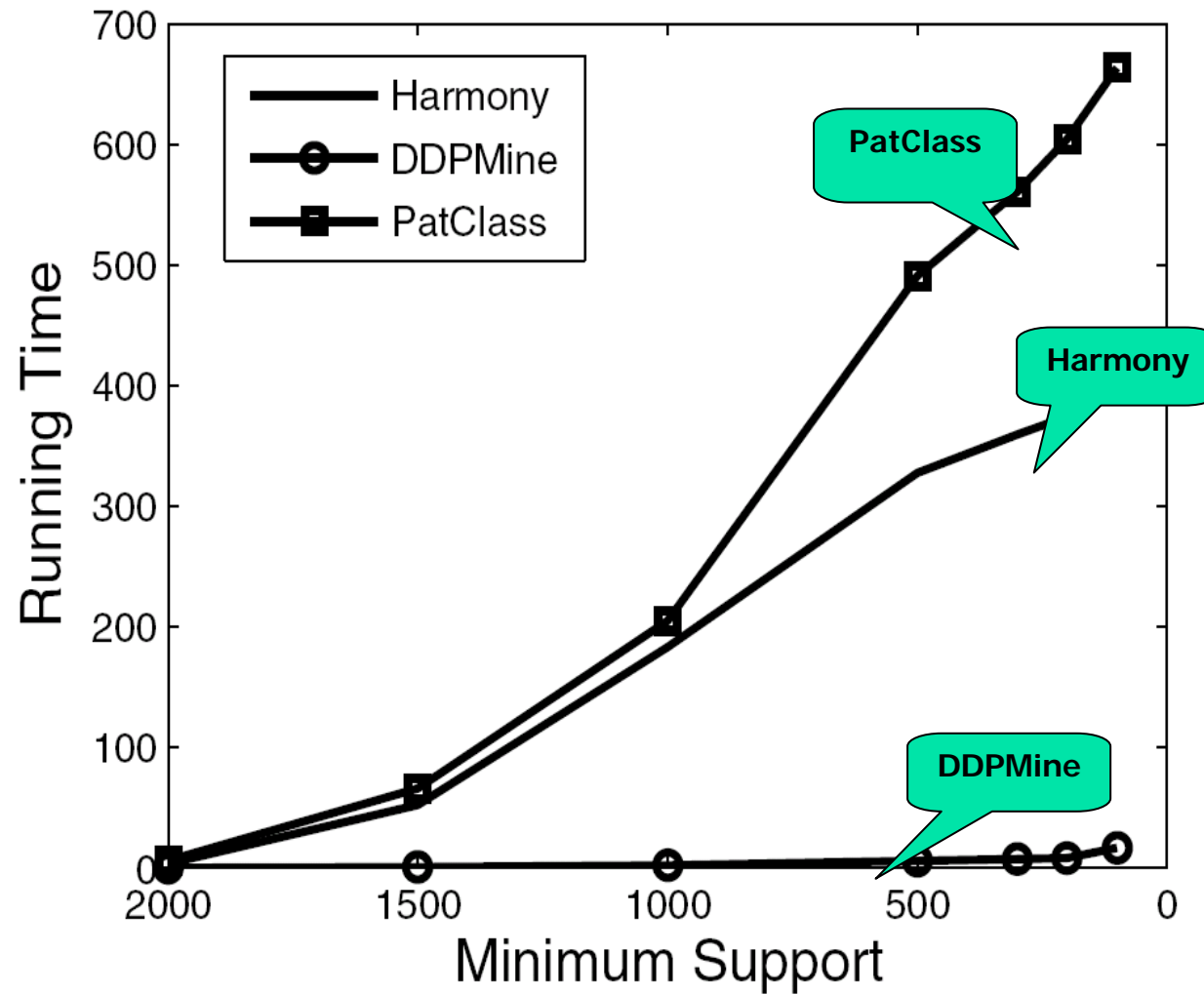
Table 4. Accuracy & Time on Waveform Data

<i>min_sup</i>	#Patterns	Time (s)	SVM (%)	C4.5 (%)
1	9,468,109	N/A	N/A	N/A
80	26,576	176.485	92.40	88.35
100	15,316	90.406	92.19	87.29
150	5,408	23.610	91.53	88.80
200	2,481	8.234	91.22	87.32

基于频繁模式的分类

- H. Cheng, X. Yan, J. Han, and C.-W. Hsu, "Discriminative Frequent Pattern Analysis for Effective Classification", ICDE'07
- Accuracy issue问题
 - Increase the discriminative power
 - Increase the expressive power of the feature space
- Scalability issue问题
 - It is computationally infeasible to generate **all feature combinations** and filter them with an information gain threshold
 - Efficient method (DDPMine: FPtree pruning): H. Cheng, X. Yan, J. Han, and P. S. Yu, "Direct Discriminative Pattern Mining for Effective Classification", ICDE'08

DDPMine Efficiency: Runtime



PatClass: ICDE'07
Pattern
Classification Alg.

Summary

- Effective and advanced classification methods
 - Bayesian belief network (probabilistic networks)
 - Backpropagation (Neural networks)
 - Support Vector Machine (SVM)
 - Pattern-based classification
 - Other classification methods: lazy learners (KNN, case-based reasoning), genetic algorithms, rough set and fuzzy set approaches
- Additional Topics on Classification
 - Multiclass classification
 - Semi-supervised classification
 - Active learning
 - Transfer learning

References (1)

- C. M. Bishop, Neural Networks for Pattern Recognition. Oxford University Press, 1995
- C. J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2(2): 121-168, 1998
- H. Cheng, X. Yan, J. Han, and C.-W. Hsu, Discriminative Frequent pattern Analysis for Effective Classification, ICDE'07
- H. Cheng, X. Yan, J. Han, and P. S. Yu, Direct Discriminative Pattern Mining for Effective Classification, ICDE'08
- N. Cristianini and J. Shawe-Taylor, Introduction to Support Vector Machines and Other Kernel-Based Learning Methods, Cambridge University Press, 2000
- A. J. Dobson. An Introduction to Generalized Linear Models. Chapman & Hall, 1990
- G. Dong and J. Li. Efficient mining of emerging patterns: Discovering trends and differences. KDD'99

References (2)

- R. O. Duda, P. E. Hart, and D. G. Stork. Pattern Classification, 2ed. John Wiley, 2001
- T. Hastie, R. Tibshirani, and J. Friedman. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer-Verlag, 2001
- S. Haykin, Neural Networks and Learning Machines, Prentice Hall, 2008
- D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. Machine Learning, 1995.
- V. Kecman, Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic, MIT Press, 2001
- W. Li, J. Han, and J. Pei, CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules, ICDM'01
- T.-S. Lim, W.-Y. Loh, and Y.-S. Shih. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. Machine Learning, 2000

References (3)

- B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining, p. 80-86, KDD'98.
- T. M. Mitchell. Machine Learning. McGraw Hill, 1997.
- D.E. Rumelhart, and J.L. McClelland, editors, Parallel Distributed Processing, MIT Press, 1986.
- P. Tan, M. Steinbach, and V. Kumar. Introduction to Data Mining. Addison Wesley, 2005.
- S. M. Weiss and N. Indurkha. Predictive Data Mining. Morgan Kaufmann, 1997.
- I. H. Witten and E. Frank. Data Mining: Practical Machine Learning Tools and Techniques, 2ed. Morgan Kaufmann, 2005.
- X. Yin and J. Han. CPAR: Classification based on predictive association rules. SDM'03
- H. Yu, J. Yang, and J. Han. Classifying large data sets using SVM with hierarchical clusters. KDD'03.

SVM—Introductory Literature

- “Statistical Learning Theory” by Vapnik: extremely hard to understand, containing many errors too.
- C. J. C. Burges. [A Tutorial on Support Vector Machines for Pattern Recognition](#). *Knowledge Discovery and Data Mining*, 2(2), 1998.
 - Better than the Vapnik’s book, but still written too hard for introduction, and the examples are so not-intuitive
- The book “An Introduction to Support Vector Machines” by N. Cristianini and J. Shawe-Taylor
 - Also written hard for introduction, but the explanation about the mercer’s theorem is better than above literatures
- The neural network book by Haykins
 - Contains one nice chapter of SVM introduction

Notes about SVM—Introductory Literature

- “Statistical Learning Theory” by **Vapnik**: difficult to understand, containing many errors.
- C. J. C. **Burges**. [A Tutorial on Support Vector Machines for Pattern Recognition](#). *Knowledge Discovery and Data Mining*, 2(2), 1998.
 - Easier than Vapnik’s book, but still not introductory level; the examples are not so intuitive
- The book [An Introduction to Support Vector Machines](#) by **Cristianini and Shawe-Taylor**
 - Not introductory level, but the explanation about Mercer’s Theorem is better than above literatures
- [Neural Networks and Learning Machines](#) by **Haykin**
 - Contains a nice chapter on SVM introduction

Chapter 6. 分类: 基本概念



- 分类: 基本概念
- 决策树归纳
- 贝叶斯分类
- 基于规则的分类
- 模型评价与选择
- 提高分类准确率的技术:集成方法Ensemble Methods
- Summary

有监督 vs. 无监督学习

- 有监督学习 (分类)

- 监督：训练数据（观察，测量等）都带有标签，指示观察的类别
- 根据训练集分类新数据

- 无监督学习 (聚类)

- 训练集的类别（标签）未知
- 给定一个观察，测量等的集合，目标是建立数据中存在的数据的类或簇

预测问题： 分类vs.数值预测

- 分类

- 预测分类的类标签(离散 **or** 名义)
- 基于训练数据和类标签 构造一个模型，并分类新数据

- 数值预测

- 建连续值函数/模型, 预测未知/缺失值

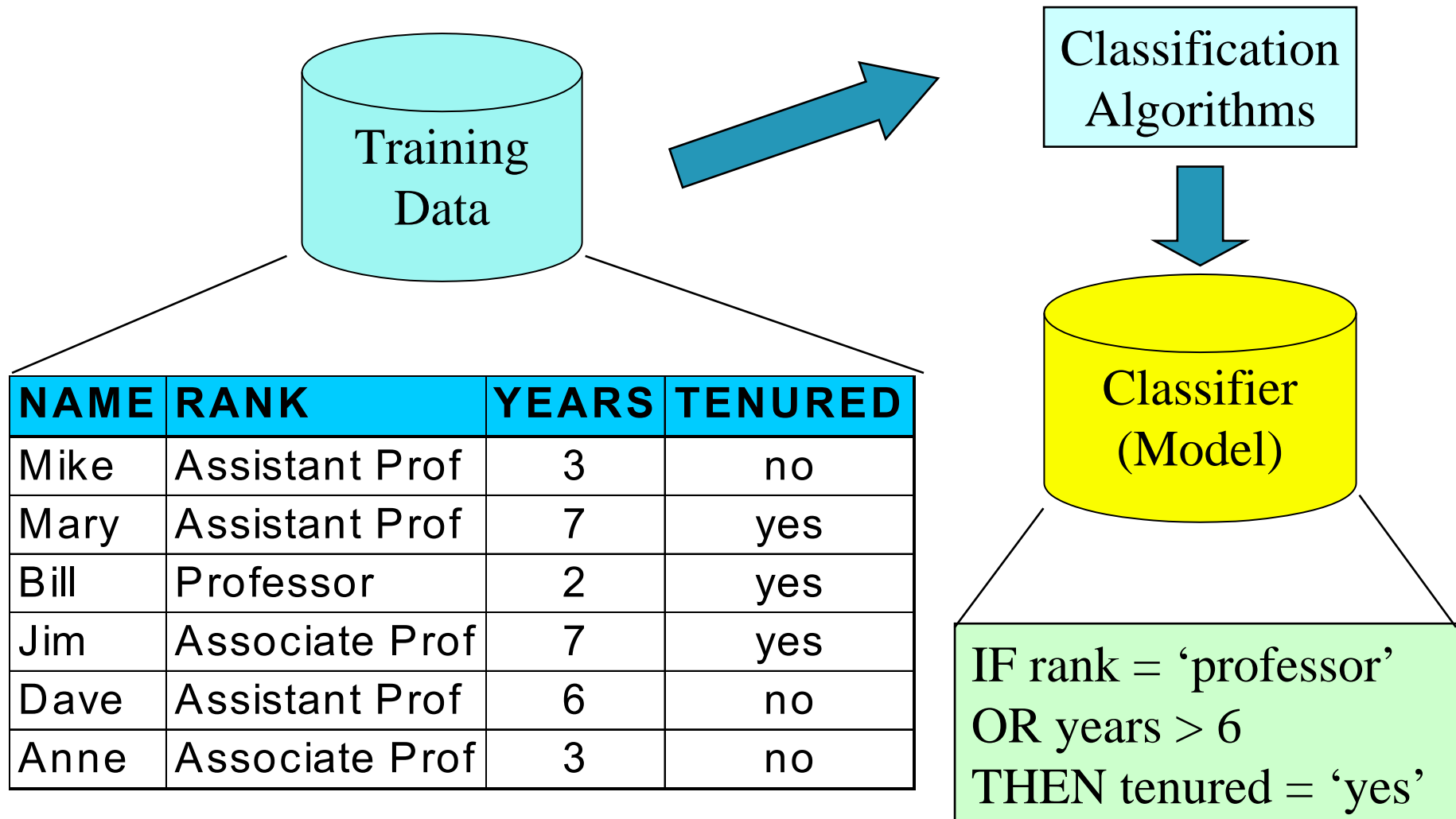
- 典型应用

- 信用卡/贷款审批:
- 医疗诊断: 肿瘤是癌或良性?
- 欺诈检测: 交易欺诈?
- 网页分类: 这是哪一类?

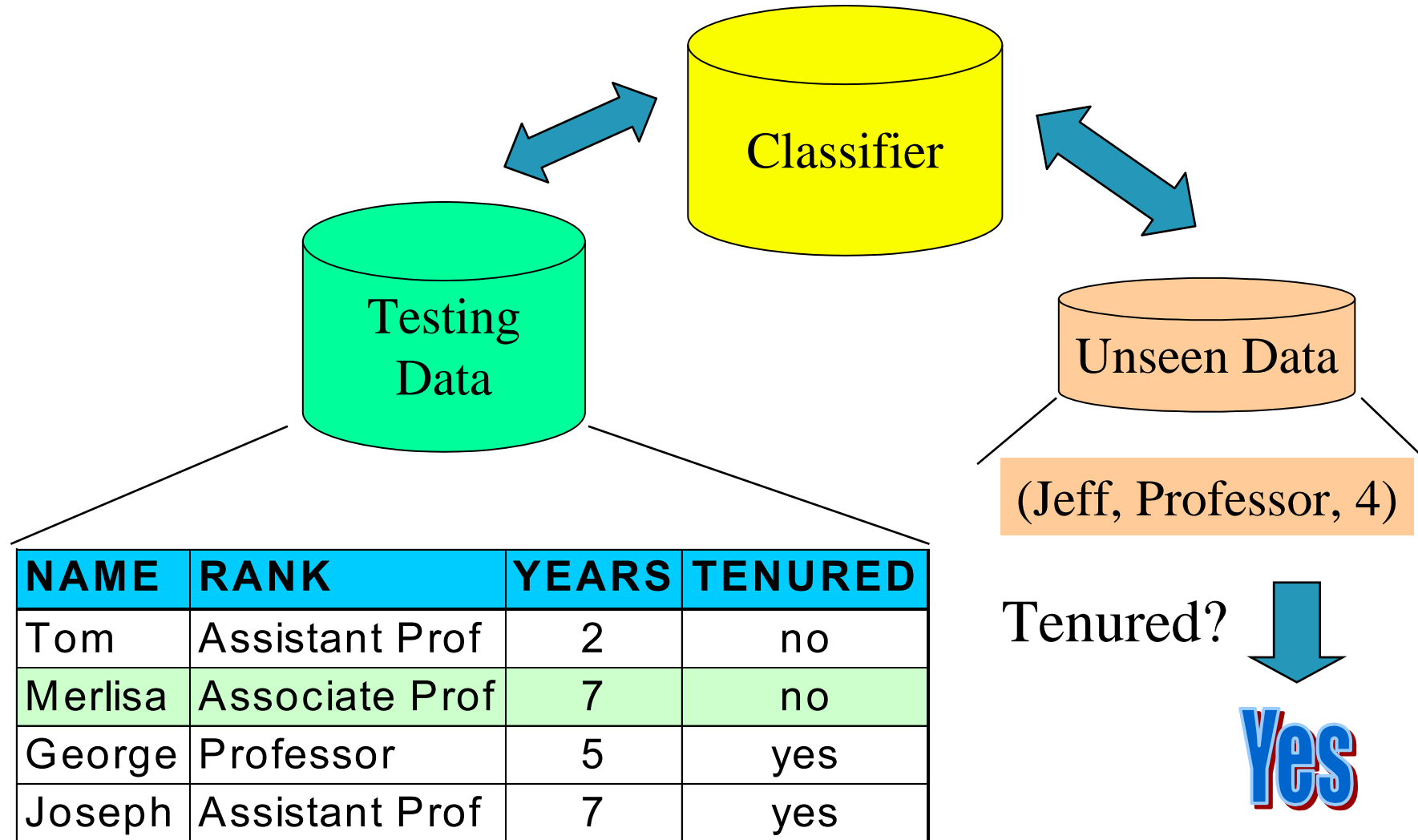
分类: 一个两步的过程

- **模型构建**: 描述一组预先定义的类
 - 假定每个元组/样本 属于一个类, 由类标签属性设定
 - 用于构建模型的元组集合称为训练集 **training set**
 - 模型可以表示为分类规则, 决策树, 数学公式
- **模型使用**: 分类将来/未知对象
 - **估计模型的准确率**
 - **测试集**: 独立于训练集的样本 (避免过分拟合 **overfitting**)
 - 比较测试样本的已知标签/由模型预测 (得到) 标签
 - **准确率**: 测试样本集中模型正确预测/分类的样本的比率
 - 如果准确率合时, 使用模型来分类标签为未知的样本

Process (1): 模型构建




Process (2): Using the Model in Prediction



Issues: Evaluating Classification Methods

- Accuracy
 - classifier accuracy: predicting class label
 - predictor accuracy: guessing value of predicted attributes
- Speed
 - time to construct the model (training time)
 - time to use the model (classification/prediction time)
- Robustness: handling noise and missing values
- Scalability: efficiency in disk-resident databases
- Interpretability
 - understanding and insight provided by the model
- Other measures, e.g., goodness of rules, such as decision tree size or compactness of classification rules

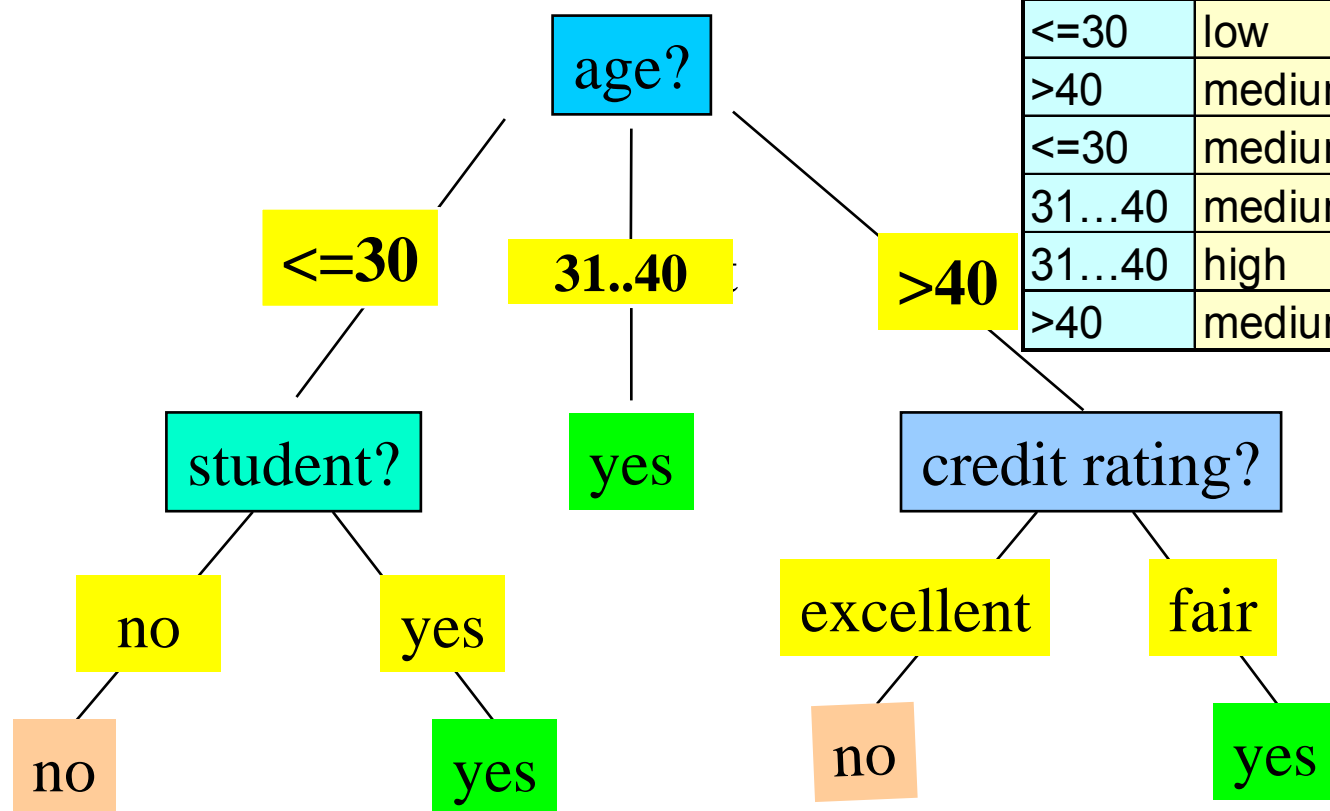
Chapter 6. 分类:决策树归纳

- 分类: 基本概念
- 决策树归纳 
- 贝叶斯分类
- 基于规则的分类
- 模型评价与选择
- 提高分类准确率的技术:集成方法Ensemble Methods
- Summary

决策树归纳: 例子

- 训练集: 购买计算机
- 结果:

age	income	student	信誉	购买计算机
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no



决策树归纳的算法

- 基本算法 (贪心算法)
 - 树构建：自顶向下递归地分治方式
 - 开始，所有的训练样本位于根节点
 - 属性是分类属性(若是连续值,事先离散化)
 - 基于选择的属性，样本被递归地分割
 - 基于启发式/统计测来选择测试属性 (例如 信息增益)
- 终止划分的条件
 - 一个给定节点的所有样本属于一个类别
 - 没有属性剩下，用于进一步划分 -运用多数投票来标记此节点
 - 没有样本剩下

输出：一棵决策树。

方法：

- (1) 创建一个节点 N ;
- (2) **if** D 中的元组都是同一类 C **then**
- (3) 返回 N 作为叶节点, 以类 C 标记;
- (4) **if** $attribute_list$ 为空 **then**
- (5) 返回 N 作为叶节点, 标记为 D 中的多数类; //多数表决
- (6) 使用 $attribute_selection_method(D, attribute_list)$, 找出“最好”的 $splitting_criterion$;
- (7) 用 $splitting_criterion$ 标记节点 N ;
- (8) **if** $splitting_attribute$ 是离散值的并且允许多路划分 **then** //不限于二叉树
- (9) $attribute_list \leftarrow attribute_list - splitting_attribute$; //删除划分属性
- (10) **for** $splitting_criterion$ 的每个输出 j // 划分元组并对每个划分产生子树
- (11) 设 D_j 是 D 中满足输出 j 的数据元组的集合; //一个划分
- (12) **if** D_j 为空 **then**
- (13) 加一个树叶到节点 N , 标记为 D 中的多数类;
- (14) **else** 加一个由 $Generate_decision_tree(D_j, attribute_list)$ 返回的节点到节点 N ;
- end for**
- (15) 返回 N ;

属性选择度量

- 属性选择度量
 - 分裂规则，决定给定节点上的元组如何分裂
 - 具有最好度量得分的属性选定位分裂属性
- 三种度量
 - 信息增益、增益率、**Gini**指标
- 数学符号
 - D 为元组的训练集，元组属于 m 个不同的类 $C_i(i=1,,m)$
 - $C_{i,D}$ 是 D 中的 C_i 类的元组集合
 - $|C_{i,D}|$ 和 $|D|$ 分别表示各自的元组个数

属性选择度量: 信息增益(ID3/C4.5)

- 选择具有最高信息增益的属性
- 令 p_i 为D中的任一元组属于类 C_i 概率, 估计为 $|C_{i,D}|/|D|$
- 分类D中元组需要的期望信息(entropy) :

$$Info(D) = - \sum_{i=1}^m p_i \log_2(p_i)$$

- (利用 A 分裂D 为v个部分后)分类D 需要的信息为:

$$Info_A(D) = \sum_{j=1}^v \frac{|D_j|}{|D|} \times Info(D_j)$$

- 以属性A分枝得到的信息增益

$$Gain(A) = Info(D) - Info_A(D)$$

属性选择: 信息增益

■ Class P: 买电脑 = “yes”

■ Class N: 买电脑 = “no”

$$Info(D) = -\frac{9}{14} \log_2 \left(\frac{9}{14} \right) - \frac{5}{14} \log_2 \left(\frac{5}{14} \right) = 0.940$$

$$\begin{aligned} Info_{age}(D) &= \frac{5}{14} \times \left(-\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5} \right) \\ &\quad + \frac{4}{14} \times \left(-\frac{4}{4} \log_2 \frac{4}{4} - \frac{0}{4} \log_2 \frac{0}{4} \right) \\ &\quad + \frac{5}{14} \times \left(-\frac{3}{5} \log_2 \frac{3}{5} - \frac{2}{5} \log_2 \frac{2}{5} \right) \\ &= 0.694 \text{ bits.} \end{aligned}$$

age	income	student	credit_rating	comp
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no

$$Gain(age) = Info(D) - Info_{age}(D) = 0.246$$

$$Gain(income) = 0.029$$

$$Gain(student) = 0.151$$

$$Gain(credit_rating) = 0.048$$

计算信息增益-连续值属性

- 令 A 为连续属性
- 必须为 A 确定一个最佳分裂点 *best split point*
 - 上升序排序 A
 - 典型地, 每对相邻值的中点是一个可能的分裂点
 - $(a_i + a_{i+1})/2$ is the midpoint between the values of a_i and a_{i+1}
 - 具有最小期望信息需求的点选为 A 的分裂点
- Split:
 - $D1$ 为 D 中元组满足 $A \leq \text{split-point}$, $D2$ 是元组满足 $A > \text{split-point}$

增益率 (C4.5)

- 信息增益倾向于有大量不同取值的属性（划分更细，更纯）
 - 极端：每个划分子集只有一个样本，即一个类
 - 此时 $\text{Info}(d)=0$
- C4.5 (ID3 后继) 使用增益率来克服这一问题(规范化信息增益)

$$\text{SplitInfo}_A(D) = - \sum_{j=1}^v \frac{|D_j|}{|D|} \times \log_2 \left(\frac{|D_j|}{|D|} \right)$$

- $\text{GainRatio}(A) = \text{Gain}(A) / \text{SplitInfo}(A)$

- Ex
$$\text{SplitInfo}_{\text{income}}(D) = -\frac{4}{14} \times \log_2 \left(\frac{4}{14} \right) - \frac{6}{14} \times \log_2 \left(\frac{6}{14} \right) - \frac{4}{14} \times \log_2 \left(\frac{4}{14} \right) = 1.557$$

- $\text{gain_ratio}(\text{income}) = 0.029 / 1.557 = 0.019$

- 具有最大增益率的属性选为分裂属性

Gini Index指标 (CART)

- 数据 D 包含 n 类别的样本, gini指标, $gini(D)$ 定义为

$$p_j \text{ 类别 } j \text{ 在 } D \text{ 中的频率} \quad gini(D) = 1 - \sum_{j=1}^n p_j^2$$

- 数据集 D 基于属性 A 分裂为子集 D_1 和 D_2 , $gini$ 指标定义为

$$gini_A(D) = \frac{|D_1|}{|D|} gini(D_1) + \frac{|D_2|}{|D|} gini(D_2)$$

- 不纯度减少: $\Delta gini(A) = gini(D) - gini_A(D)$
- 具有最小 $gini_{split}(D)$ 的属性(or不纯度减少最大的) 用于分裂节点 (需要枚举所有可能的分裂情况)

计算 Gini Index 指标

- D 有 9个元组买电脑 = “yes” / 5 个买电脑 = “no”

$$gini(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459$$

- 设属性income分裂D为包含10个元组的 D_1 : {low, medium} / 4个元组的 D_2

$$\begin{aligned} & Gini_{income \in \{low, medium\}}(D) \\ &= \frac{10}{14} Gini(D_1) + \frac{4}{14} Gini(D_2) \\ &= \frac{10}{14} \left(1 - \left(\frac{7}{10}\right)^2 - \left(\frac{3}{10}\right)^2 \right) + \frac{4}{14} \left(1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2 \right) \\ &= 0.443 \\ &= Gini_{income \in \{high\}}(D). \end{aligned}$$

$Gini_{\{low, high\}} = 0.458$; $Gini_{\{medium, high\}} = 0.450$. 因此{low, medium} / {high}分裂, 由于其有最小的Gini index

- 假设所有属性都是连续值, 需要其他技术, e.g., 聚类, 来获得可能的分裂点

比较属性选择度量

- 通常三种度量获得较好的结果
 - 信息增益**Information gain**:
 - 偏向于多值属性
 - 增益率**Gain ratio**:
 - 倾向于不平衡的分裂，其中一个子集比其他小得多
 - **Gini index**:
 - 偏向于多值属性
 - 当类数目较大时，计算困难
 - 倾向于导致大小相等的分区和纯度

其他属性选择度量

- CHAID: 一种流行的决策树算法, 基于独立 χ^2 检验的选择度量
- C-SEP: 某些情况下比信息增益gini指标更好
- G-statistic: 非常近似于 χ^2 分布
- MDL (最小描述长度) (i.e., 首选最简单的解):
 - 最佳树为需要最小二进位的树 (1) 编码树, (2) 编码树的异常
- 多元划分 (基于多变量组合来划分)
 - CART: 基于属性的线性组合来发现多元划分
- 哪一个是最好的?
 - 大部分可以获得较好结果, 没有一个显著地优于其他

过拟合与数剪枝

- 过拟合Overfitting: 一棵归纳的树 可能过分拟合训练数据
 - 分枝太多,某些反映训练数据中的异常, 噪音/孤立点
 - 对未参与训练的样本的低精度预测
- 两种处理方法
 - 先剪枝: 提前终止树构造
 - 如果对一个节点的分裂会产生低于给定的阈值的度量, 划分停止
 - 选择一个合适的阈值很难
 - 后剪枝: 从完全生长的树中剪去树枝—得到一个逐步修剪树
 - 例如, 最小化代价复杂度 (树节点个数和错误率的函数)
 - 使用不同于训练集的数据来确定哪一个 “**best pruned tree**”

决策树归纳的增强

- 允许连续值属性
 - 动态地定义新的离散值属性，其把连续值属性分成离散的区间
- 处理缺失属性值
 - 分配属性的最常见值
 - 为每一个可能的值分配概率
- 属性构造
 - 基于现有的稀少出现的属性创建新的属性，
 - 这减少了分散，重复和复制

大型数据库中分类

- 分类—被统计学和机器学习研究人员广泛地研究一个经典问题
- 可伸缩性:以合理的速度分类由带有数百个属性的百万个样本组成的数据集
- 为什么决策树归纳受欢迎?
 - 相对快的训练速度 (与其他分类方法相比)
 - 转换为简单、易于理解的分类规则
 - 可用 **SQL** 查询来访问数据库
 - 与其它方法可比的分类精度
- **RainForest** (VLDB'98 — Gehrke, Ramakrishnan & Ganti)
 - Builds an AVC-list (attribute, value, class

RainForest雨林的可扩展性框架

- 可扩展性和确定质量树的标准相分离
- 建并维持 AVC-list: **AVC** (属性-值, 类标号)
- **AVC集** (of an attribute X)
 - 把训练集投影到属性 X 和类标签上, 给出属性 X 的
每个值上的类标签计数
- **AVC组群** (在节点 n)
 - 节点 n 上所有预测属性的AVC集合----组群

Rainforest: 训练集和AVC集

Training Examples

age	income	student	credit_rating	comp
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no

AVC-set on *Age*

Age	Buy_Computer	
	yes	no
<=30	2	3
31..40	4	0
>40	3	2

AVC-set on *income*

income	Buy_Computer	
	yes	no
high	2	2
medium	4	2
low	3	1

AVC-set on *Student*

student	Buy_Computer	
	yes	no
yes	6	1
no	3	4

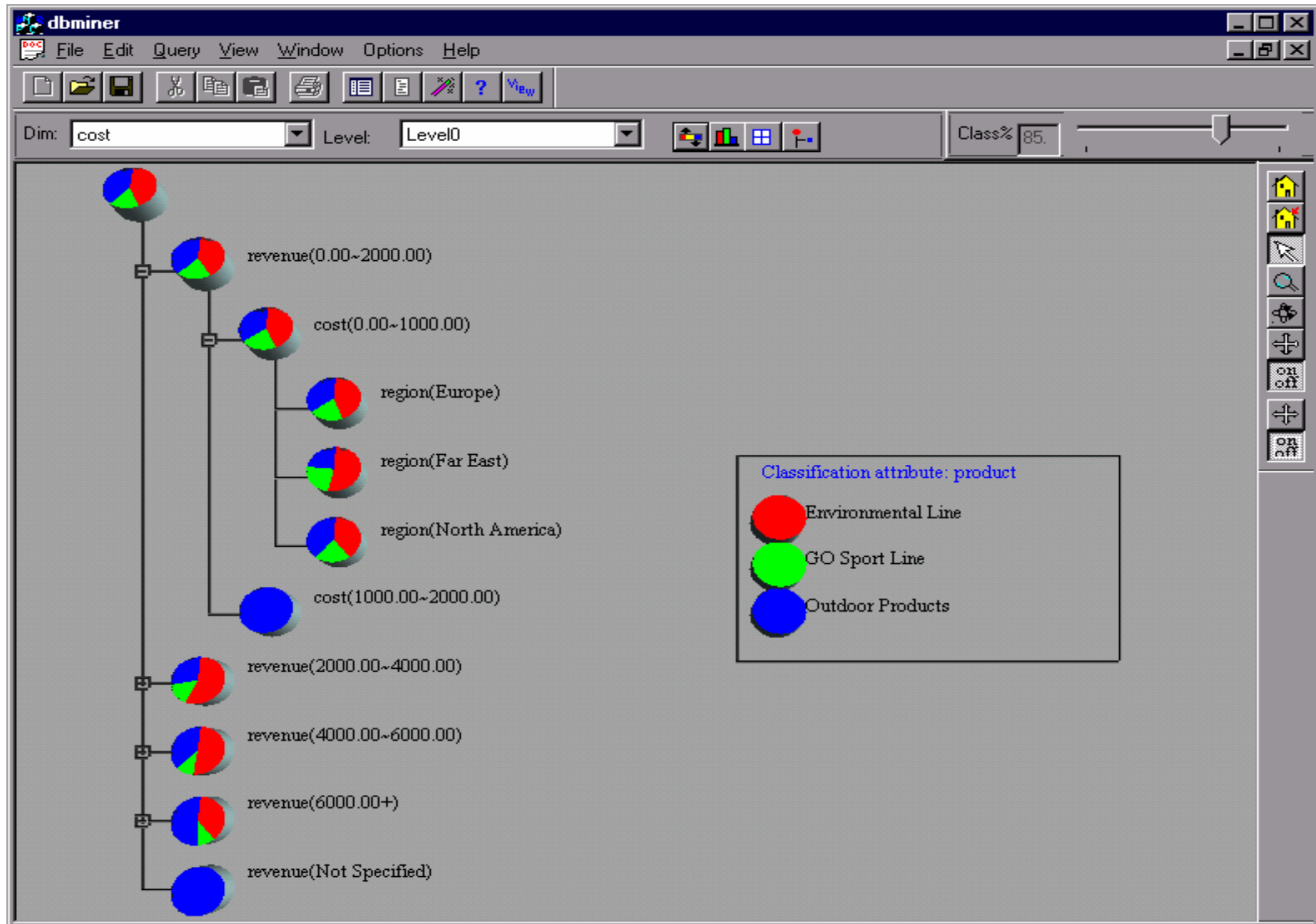
AVC-set on *credit_rating*

Credit rating	Buy_Computer	
	yes	no
fair	6	2
excellent	3	3

BOAT (Bootstrapped Optimistic Algorithm for Tree Construction)

- 使用一个叫做 ***bootstrapping*** 自助法的统计技术多个更小的样本集（子集），每一个可放入内存
- 每个子集产生一个树，导致多个树
- 考察这些树并用他们构造一个新树 T'
 - 事实证明， T' 非常接近于使用全部数据集构造的树
- **Adv:** 只要求扫描DB两遍,并且是一个增量算法.

分类结果的陈述/表示




决策树可视化SGI/MineSet 3.0



SGI公司和美国Stanford大学联合开发的多任务数据挖掘系统。

MineSet以先进的可视化显示方法闻名于世

Chapter 6. 分类: 贝叶斯分类

- 分类: 基本概念
- 决策树归纳
- 贝叶斯分类 
- 基于规则的分类
- 模型评价与选择
- 提高分类准确率的技术: 集成方法 Ensemble Methods
- Summary

贝叶斯理论

- 令 \mathbf{X} 为数据样本: 类标签未知
- 令 \mathbf{H} 为一个假设在: \mathbf{X} 属于类别 \mathbf{C}
- 分类就是确定 $P(\mathbf{H}|\mathbf{X})$ (*后验概率*),
 - 给定观察数据 \mathbf{X} , 假设 \mathbf{H} 成立的概率
- $P(\mathbf{H})$ (*先验概率*)——最初的概率
 - 例, 不管年龄和收入等条件 \mathbf{X} 将会购买计算机
- $P(\mathbf{X})$: 样本数据 \mathbf{x} 被观察到的概率
- $P(\mathbf{X}|\mathbf{H})$ (可能性),
 - 假设 \mathbf{H} 成立, 那么观测到样本 \mathbf{X} 的概率
 - E.g., 已知 \mathbf{X} 购买计算机, \mathbf{X} 为31..40且中等收入的概率

贝叶斯理论 Bayesian Theorem

- 给定训练数据 \mathbf{X} , 假设 H 的后验概率 $P(H|\mathbf{X})$ 满足贝叶斯理论

$$P(H | \mathbf{X}) = \frac{P(\mathbf{X} | H) P(H)}{P(\mathbf{X})} = P(\mathbf{X} | H) \times P(H) / P(\mathbf{X})$$

- 通俗地说, 这可以写成

posteriori = likelihood x prior/evidence

- 预测 \mathbf{X} 属于类别 C_2 当且仅当概率 $P(C_i|\mathbf{X})$ 是所有 $P(C_k|\mathbf{X})$ for all the k classes 最大的
- 实际困难: 需要许多可能性的初步知识, 计算成本显著

Naïve Bayesian Classifier

- **D**为训练数据集（包含类别标签），并且每个元组表示为一个**n**-维的属性向量**X** = (**x**₁, **x**₂, ..., **x**_n)
- 假定有 **m** 个类别 **C**₁, **C**₂, ..., **C**_m.
- 分类就是推导最大的后验概率, i.e., the maximal **P(C_i|X)**
- 可以由贝叶斯理论计算
$$P(C_i|\mathbf{X}) = \frac{P(\mathbf{X}|C_i)P(C_i)}{P(\mathbf{X})}$$
- 由于对所有类**P(X)**是常量，只需要最大化

$$P(C_i|\mathbf{X}) = P(\mathbf{X}|C_i)P(C_i)$$

朴素贝叶斯分类器的推导

- 一个简单假定：属性是条件独立的 (**i.e.**, 属性间没有依赖关系):

$$P(\mathbf{X}|C_i) = \prod_{k=1}^n P(x_k | C_i) = P(x_1 | C_i) \times P(x_2 | C_i) \times \dots \times P(x_n | C_i)$$

- 这样极大地减少了计算代价：只需要统计类的分布
- 若 \mathbf{A}_k 是分类属性
 - $P(\mathbf{x}_k|C_i) = C_i$ 类中 \mathbf{A}_k 取值为 \mathbf{x}_k 的元组数/ $|C_i|$ (类 C_i 的大小)
- 若 \mathbf{A}_k 是连续值, $P(\mathbf{x}_k|C_i)$ 通常基于均值 μ 标准差 σ 的高斯分布计算

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- $P(\mathbf{x}_k|C_i) =$
 $P(\mathbf{X}|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i})$

朴素贝叶斯分类: 训练数据集

两个类别:

C1:buys_computer =
'yes'

C2:buys_computer =
'no'

数据样本

X = (age <=30,
Income = medium,
Student = yes
Credit_rating = Fair)

age	income	student	credit_rating	com
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no

Naïve Bayesian Classifier: 例子

- $P(C_i)$: $P(\text{buys_computer} = \text{"yes"}) = 9/14 = 0.643$
 $P(\text{buys_computer} = \text{"no"}) = 5/14 = 0.357$
- Compute $P(X|C_i)$ for each class
 - $P(\text{age} = \text{"<=30"} | \text{buys_computer} = \text{"yes"}) = 2/9 = 0.222$
 - $P(\text{age} = \text{"<= 30"} | \text{buys_computer} = \text{"no"}) = 3/5 = 0.6$
 - $P(\text{income} = \text{"medium"} | \text{buys_computer} = \text{"yes"}) = 4/9 = 0.444$
 - $P(\text{income} = \text{"medium"} | \text{buys_computer} = \text{"no"}) = 2/5 = 0.4$
 - $P(\text{student} = \text{"yes"} | \text{buys_computer} = \text{"yes"}) = 6/9 = 0.667$
 - $P(\text{student} = \text{"yes"} | \text{buys_computer} = \text{"no"}) = 1/5 = 0.2$
 - $P(\text{credit_rating} = \text{"fair"} | \text{buys_computer} = \text{"yes"}) = 6/9 = 0.667$
 - $P(\text{credit_rating} = \text{"fair"} | \text{buys_computer} = \text{"no"}) = 2/5 = 0.4$
- **$X = (\text{age} \leq 30, \text{income} = \text{medium}, \text{student} = \text{yes}, \text{credit_rating} = \text{fair})$**
 - $P(X|C_i) : P(X|\text{buys_computer} = \text{"yes"}) = 0.222 \times 0.444 \times 0.667 \times 0.667 = 0.044$
 - $P(X|\text{buys_computer} = \text{"no"}) = 0.6 \times 0.4 \times 0.2 \times 0.4 = 0.019$
 - $P(X|C_i) * P(C_i) : P(X|\text{buys_computer} = \text{"yes"}) * P(\text{buys_computer} = \text{"yes"}) = 0.028$
 - $P(X|\text{buys_computer} = \text{"no"}) * P(\text{buys_computer} = \text{"no"}) = 0.007$

贝叶斯分类: Why?

- 一个统计学分类器: 执行概率预测, *i.e.*, 预测类成员的概率
- 基础: 基于贝叶斯理论
- **Performance**: 一个简单的贝叶斯分类器, 朴素贝叶斯分类器, 可以与决策树和经过挑选的神经网络分类器相媲美
- 增量: 每次训练的样本可以逐步增加/减少一个假设是正确的可能性——先验知识可与观测数据相结合
- **Standard**: 即使贝叶斯方法是难以计算的, 最优决策制定提供标准 (其他方法可以衡量)

避免零概率问题

- 朴素贝叶斯要求每个条件概率非零. 然而,预测的概率可能为零

$$P(X | C_i) = \prod_{k=1}^n P(x_k | C_i)$$

- Ex. 假定有1000 元组, e=low (0), income= medium (990), and income = high (10)
- Use **Laplacian correction**校准 (or Laplacian estimator估计法)

- *Adding 1 to each case*

$$\text{Prob}(\text{income} = \text{low}) = 1/1003$$

$$\text{Prob}(\text{income} = \text{medium}) = 991/1003$$


$$\text{Prob}(\text{income} = \text{high}) = 11/1003$$

- 校准的 “corrected” 概率估计很接近未校准的

Naïve Bayesian Classifier:评论

- Advantages
 - Easy to implement
 - Good results obtained in most of the cases
- Disadvantages
 - Assumption: 类条件独立性, 损失精度
 - 实际中, 变量间存在依赖
 - E.g., 医院: 患者: 简介: 年龄, 家族病史等
症状: 发烧, 咳嗽等疾病: 肺癌, 糖尿病等
 - Dependencies among these cannot be modeled by Naïve Bayesian Classifier
- How to deal with these dependencies? Bayesian Belief Networks

Chapter 6. 分类:基于规则的分类

- 分类: 基本概念
- 决策树归纳
- 贝叶斯分类
- 基于规则的分类 
- 模型评价与选择
- 提高分类准确率的技术:集成方法Ensemble Methods
- Summary

使用IF-THEN 规则分类

- 使用 IF-THEN 规则表示知识

R: IF *age* = youth AND *student* = yes THEN *buys_computer* = yes

- 规则前件/前提 vs. 规则结论

- 评估规则: 覆盖率 *coverage* and 准确率 *accuracy*

- $n_{\text{covers}} = \# \text{ 规则R覆盖的元组数 } \% \text{ 给定元组, 规则的前提满足一覆盖元组}$

- $n_{\text{correct}} = \# \text{ R正确分类的元组数}$

$\text{coverage}(R) = n_{\text{covers}} / |D|$ / %D: 训练数据集

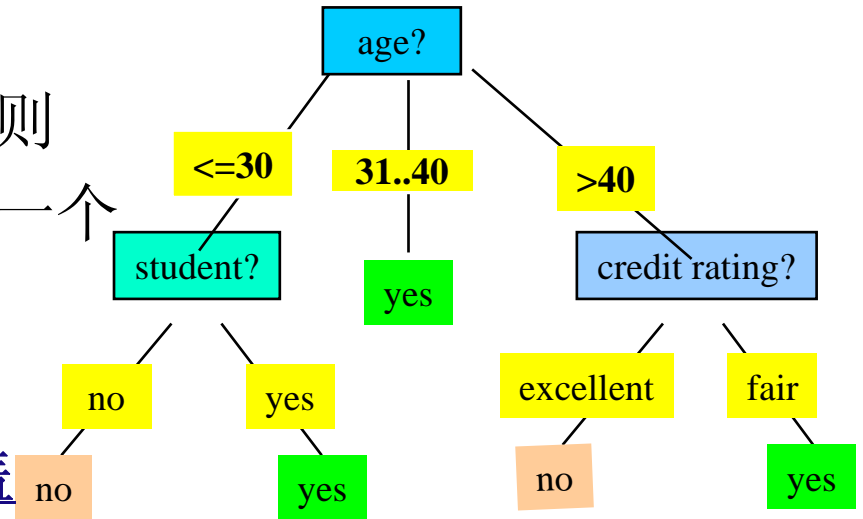
$\text{accuracy}(R) = n_{\text{correct}} / n_{\text{covers}}$

- 如果超过1条规则被触发,需要解决冲突

- 规模序Size ordering: 最高优先权赋予“最苛刻”的规则(即, 最多属性测试)
- 基于类的序: 每个类的错误分类代价的下降序
- 基于规则的序(决策表): 根据一些规则的质量度量或由专家建议, 规则被组织成一个长的优先级列表

从决策树提取规则

- 规则比一棵大的决策树更容易理解
- 从根到每个叶子的路径产生一个规则
- 沿路径的每个属性值对一起形成了一个联合: 叶节点形成规则后件
- 规则是互斥的和穷举的
 - 没有冲突规则, 每个元组被覆盖



- Example: Rule extraction from our *buys_computer* decision-tree

IF <i>age</i> = young AND <i>student</i> = no	THEN <i>buys_computer</i> = no
IF <i>age</i> = young AND <i>student</i> = yes	THEN <i>buys_computer</i> = yes
IF <i>age</i> = mid-age	THEN <i>buys_computer</i> = yes
IF <i>age</i> = old AND <i>credit_rating</i> = excellent	THEN <i>buys_computer</i> = no
IF <i>age</i> = old AND <i>credit_rating</i> = fair	THEN <i>buys_computer</i> = yes

顺序覆盖算法的规则归纳

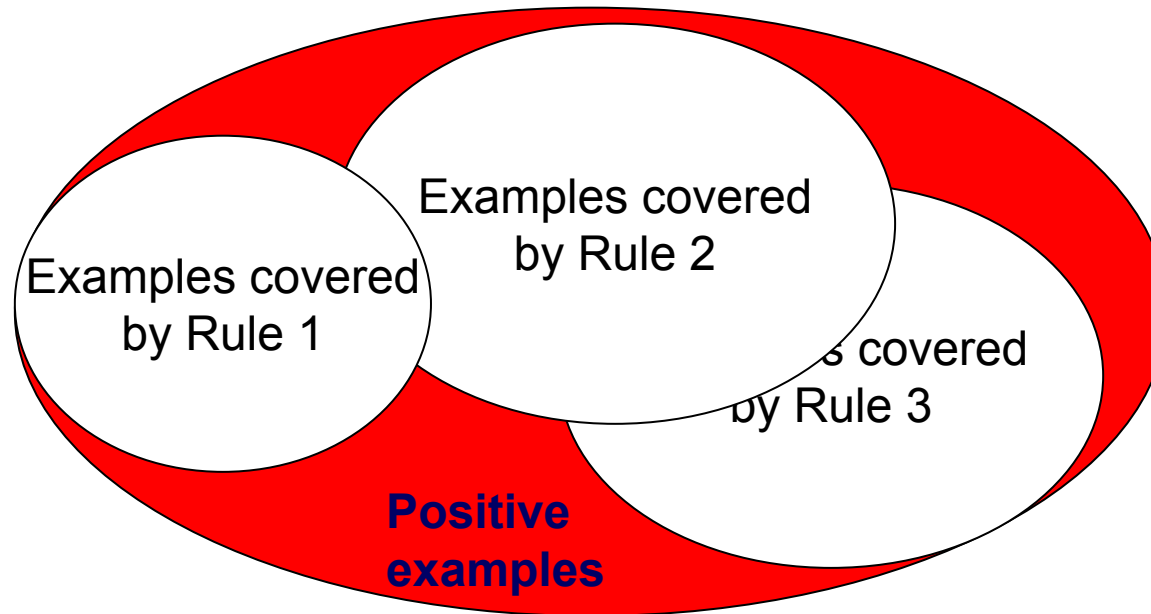
- 顺序覆盖算法：直接从训练数据抽取规则
- 典型的算法：FOIL, AQ, CN2, RIPPER
- 规则被顺序地学习，类 C_i 的规则将尽量覆盖 C_i 的元组，不或少覆盖其他类的元组
- Steps:
 - 一次学习一个规则
 - 每学习一个规则，删除此规则覆盖的元组
 - 对剩下的元组重复该过程直到终止条件，e. g., 没有训练样本/返回的规则的质量低于用户给定的阈值
- 与决策树对照：同时学习一组规则

顺序覆盖算法

while (enough target tuples left)

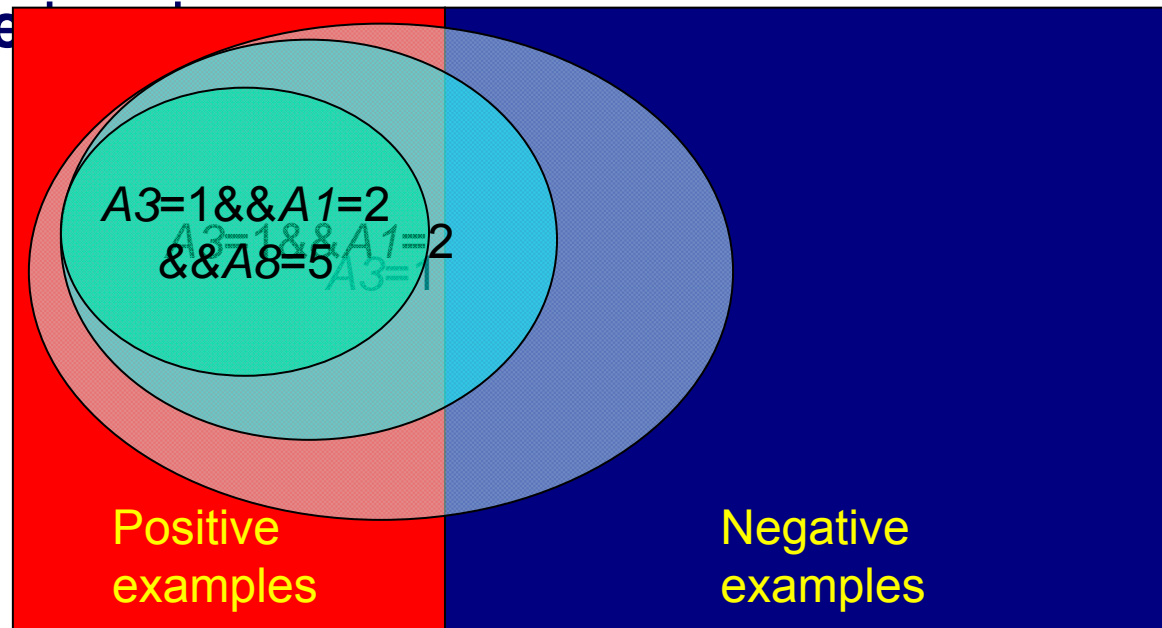
产生一个规则

删除这个规则覆盖的元组



Rule Generation

- To generate a rule
while(true)
 找到最好的谓词 p
 if 规则质量度量(p) > threshold **then** add p to current rule
 else



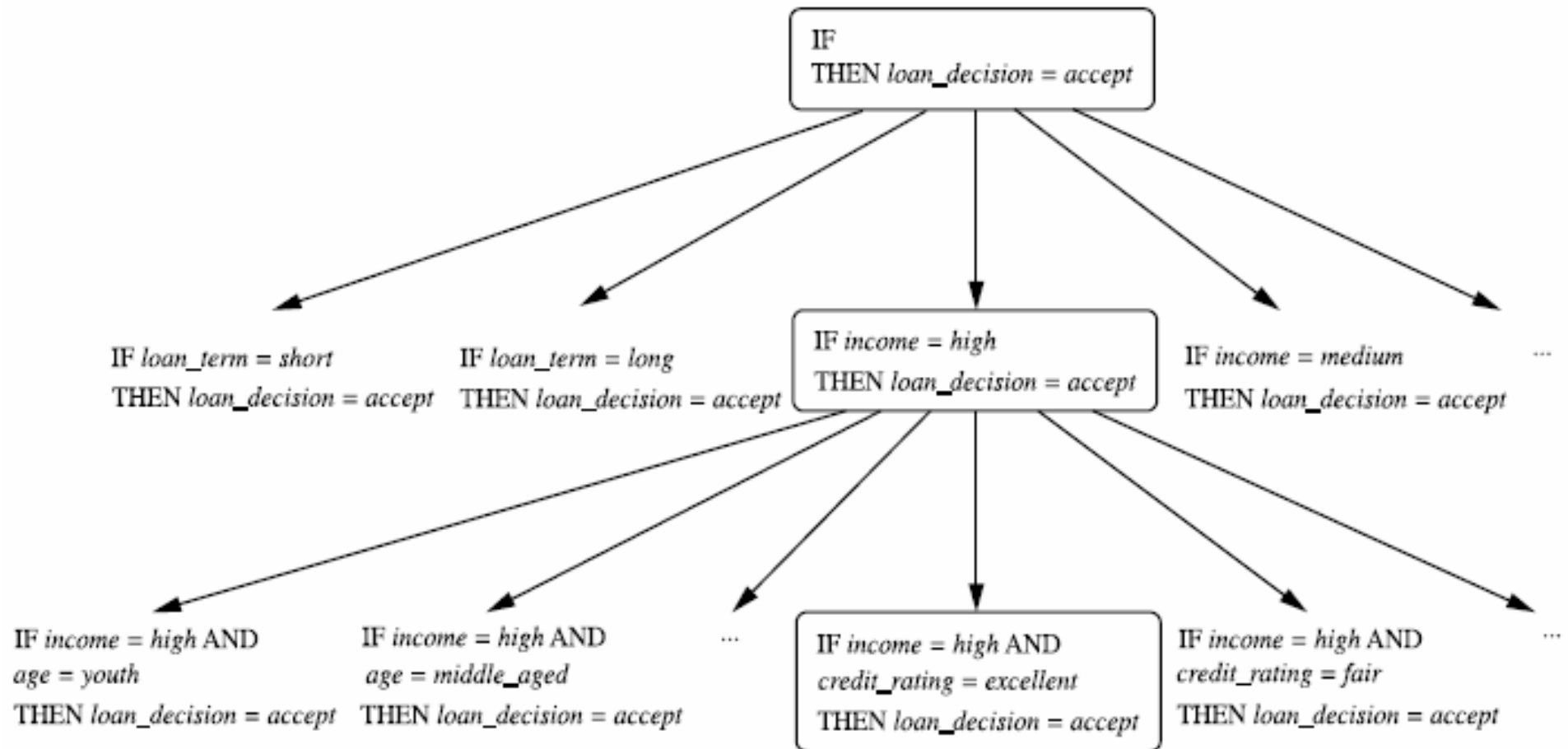
输出：IF-THEN规则的集合。

方法：

- (1) $Rule_set = \{\}$; // 学习的规则的初始集为空
- (2) **for** 每个类 c **do**
- (3) **repeat**
- (4) $Rule = \text{Learn_One_Rule}(D, Att_vals, c)$;
- (5) 从 D 中删除 $Rule$ 覆盖的元组;
- (6) **until** 终止条件满足;
- (7) $Rule_set = Rule_set + Rule$; // 将新规则添加到规则集
- (8) **endfor**
- (9) 返回 $Rule_Set$;

如何学习一个规则？

- 从可能的最一般的规则开始: `condition = empty`
- 采用贪心的深度优先策略添加新属性（于规则中）
 - 选择对“规则质量”提高最大的那个属性



A general-to-specific search through rule space.

规则质量度量与剪枝

- 规则质量度量: 同时考虑 覆盖率和准确率
 - Foil-gain (in FOIL & RIPPER): 评价扩展条件的info_gain

$$FOIL_Gain = pos' \times (\log_2 \frac{pos'}{pos' + neg'} - \log_2 \frac{pos}{pos + neg})$$


- 偏向于具有高准确率并覆盖许多正元组的规则
 - 正用于学习规则的类的元组—**正元组**；其余为**负元组**
 - Pos(neg):规则覆盖的正（负）元组数
- 基于一个独立的测试集进行规则剪枝（即删除一个属性测试）

$$FOIL_Prune(R) = \frac{pos - neg}{pos + neg}$$

Pos/neg are #被R覆盖的正/负元组.

If 规则R 剪枝后 $FOIL_Prune$ 较高, 那么剪枝R

Chapter 6. 分类:模型评价与选择

- 分类: 基本概念
- 决策树归纳
- 贝叶斯分类
- 基于规则的分类
- 模型评价与选择 
- 提高分类准确率的技术:集成方法Ensemble Methods
- Summary

模型评价与选择

- 评价指标: 怎样度量准确率?考虑其他指标??
- 使用测试集（带标签）代替训练集评估准确度
- 估计分类器准确率的方法:
 - Holdout method, random subsampling
 - 交叉验证 Cross-validation
 - 自助法（解靴带） Bootstrap
- Comparing classifiers:
 - 置信区间 Confidence intervals
 - 代价效益分析和ROC曲线
 - Cost-benefit analysis and ROC Curves

分类器评价指标: 混淆矩阵

混淆矩阵 **Confusion Matrix**:

Actual class \ Predicted class	C_1	$\neg C_1$
C_1	True Positives (TP)	False Negatives (FN)
$\neg C_1$	False Positives (FP)	True Negatives (TN)

- 感兴趣的类定为“**正类**”或“**阳性类**”，对应的为“**负/阴性类**”
 - 正样本/负样本
- 给定 m 个类, $\mathbf{CM}_{i,j}$ 表示 # 类 i 的样本被分类器分到类别 j 的个数
- 可以提供额外的行/列提供“合计”和“识别率”

例子:

Actual class \ Predicted class	buy_computer = yes	buy_computer = no	Total
buy_computer = yes	6954	46	7000
buy_computer = no	412	2588	3000
Total	7366	2634	10000

分类器评价指标: 准确度, 误差率, 灵敏性 Sensitivity, 特效性 Specificity

A \	C	$\neg C$	
C	T	F	P
$\neg C$	F	N	N
	P	N	All

- 分类器准确度, or 识别率: 测试元组被正确识别的比例

$$\text{Accuracy} = (TP + TN) / \text{All}$$

- 误差率: $1 - \text{accuracy}$, or
 $\text{Error rate} = (FP + FN) / \text{All}$

- **Class Imbalance Problem** 类分布不平衡问题:

- One class may be *rare*, e.g. fraud, or HIV-positive

- **Sensitivity:** True Positive recognition rate

- **Sensitivity** = TP / P

- **Specificity:** True Negative recognition rate

- **Specificity** = TN / N

分类器评价指标:

Precision and Recall, and F-measures

- **Precision:** 正确 – 被分类器标记为正类的样本中实际上属于“正类”的比例

$$precision = \frac{TP}{TP + FP}$$

- **Recall:** completeness完全 – what % of positive tuples did the classifier label as positive?

$$recall = \frac{TP}{TP + FN}$$

- Perfect score is 1.0

- 精度和召回率逆关系

- **F measure (F_1 or F-score):**精度和召回的调和平均值,

$$F = \frac{2 \times precision \times recall}{precision + recall}$$

- F_β :精确度和召回率的加权量

- assigns β times as much weight to recall as to precision

$$F_\beta = \frac{(1 + \beta^2) \times precision \times recall}{\beta^2 \times precision + recall}$$

分类器评价指标: 例子

真实类\预测类	cancer = yes	cancer = no	Total	Recognitio n(%)
cancer = yes	90	210	300	30.00 (<i>sensitivity</i>)
cancer = no	140	9560	9700	98.56 (<i>specificity</i>)
Total	230	9770	10000	96.40 (<i>accuracy</i>)

- $Precision = 90/230 = 39.13\%$

$$Recall = 90/300 =$$

评测分类器的正确率:

Holdout & Cross-Validation Methods

■ Holdout method

- 给定数据随机分成两个部分
 - 训练集 (e.g., 2/3) 用于模型构造
 - 测试集 (e.g., 1/3) 用于正确率估计
- 随机抽样: a variation of holdout
 - 重复holdout k 次, accuracy = 所有正确率的平均值

■ Cross-validation (k -fold, $k = 10$ 最常用)

- 随机分割数据为 k 互不相交的子集, 每一个大小近似相等
- 在 i -th 迭代中, 使用 D_i 为测试集其他的为训练集
- 留一法: k folds where $k = \#$ of tuples, for small sized data
- *Stratified cross-validation*: 每个部分分层使得每个子集中类分布近似于原始数据

评测分类器的正确率: Bootstrap

■ Bootstrap

- 对于小样本数据，效果很好
- 从给定样本中又放回的均匀抽样 *with replacement*
 - i.e., 每次一个样本被选中, 把它加入训练集并且等可能得被再次选中

■ 多个自助法, 最常用的是 .632 bootstrap

- 含 d 个样本的数据集有放回抽样 d 次, 产生 d 个样本的训练集. 没有被抽到的样本组成测试集. 大约63.2% 的样本被抽中, 剩余的36.8% 形成测试集(因为 $(1 - 1/d)^d \approx e^{-1} = 0.368$)
- 重复抽样过程 k 次, 总体准确率为:

$$Acc(M) = \frac{1}{k} \sum_{i=1}^k (0.632 \times Acc(M_i)_{test_set} + 0.368 \times Acc(M_i)_{train_set})$$

估计置信区间: 分类器 M_1 vs. M_2

- 假定有连个分类器 M_1 and M_2 , 那一个更好?
- 用10-fold cross-validation获得 $\overline{err}(M_1)$ $\overline{err}(M_2)$
- 这些平均误差率仅仅是未来数据总体误差的一种估计
- 2个错误率之间差别如果是否是偶然的?
 - 使用统计显著性检验
 - 获得估计误差的**confidence limits**置信界

估计置信区间: **Null Hypothesis**

- 执行 10-fold cross-validation
- 假定样本服从 $k-1$ 个自由度的 **t distribution** ($k=10$)
degrees of freedom
- Use **t-test** (or **Student's t-test**)
- 零假设 **Null Hypothesis**: M_1 & M_2 相同（即没有区别）
- 如果可以拒绝 null hypothesis, 那么
 - 可以断定 M_1 & M_2 间的不同是统计上显著的
 - Chose model with lower error rate

估计置信区间: t-test

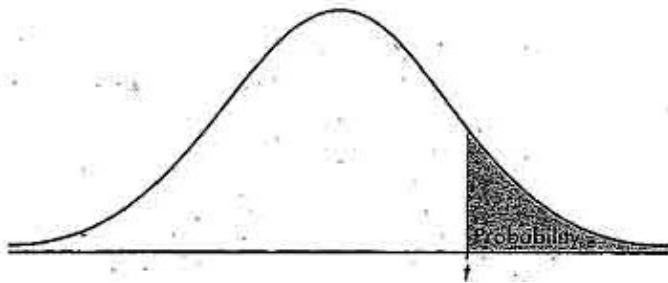
- 当只有一个测试集时: 成对比较 **pairwise comparison**
 - 对于10倍交叉验证中的 i^{th} round, 使用相同的样本分割 来计算 $err(M_1)_i$ and $err(M_2)_i$ and $\overline{err}(M_1)$ and $\overline{err}(M_2)$
 - 然后求平均over 10
 - **t-test comp** $t = \frac{\overline{err}(M_1) - \overline{err}(M_2)}{\sqrt{var(M_1 - M_2)/k}}$ **k-1 degrees of freedom:** 其中

$$var(M_1 - M_2) = \frac{1}{k} \sum_{i=1}^k \left[err(M_1)_i - err(M_2)_i - (\overline{err}(M_1) - \overline{err}(M_2)) \right]^2$$

- 如果有两个测试集: **t-test**
where $var(M_1 - M_2) = \sqrt{\frac{var(M_1)}{k_1} + \frac{var(M_2)}{k_2}}$

where k_1 & k_2 are # of cross-validation samples used for M_1 & M_2 , resp.

估计置信区间: Table for t-distribution



- Symmetric
- **Significance level**, e.g., $sig = 0.05$ or 5% means M_1 & M_2 are *significantly different* for 95% of population
- **Confidence limit**, $z = sig/2$

TABLE B: t-DISTRIBUTION CRITICAL VALUES

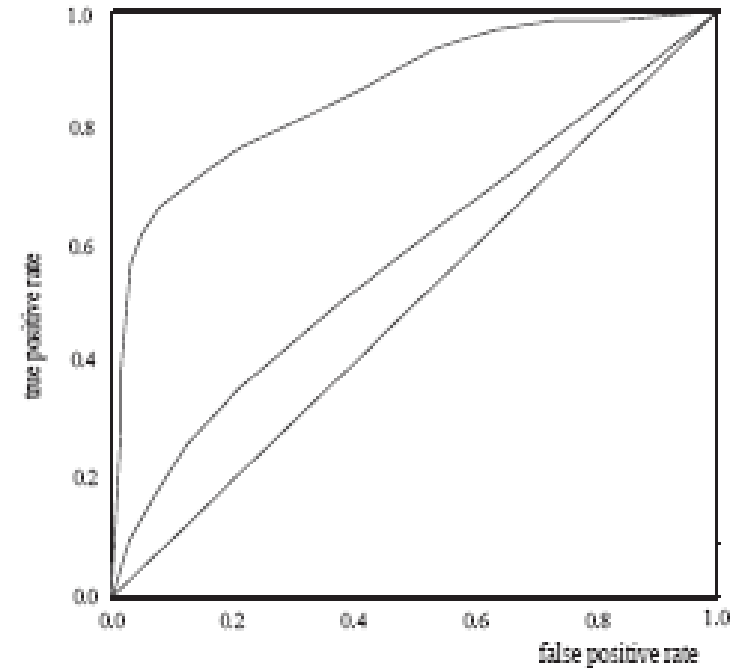
df	Tail probability p											
	.25	.20	.15	.10	.05	.025	.02	.01	.005	.0025	.001	.0005
1	1.000	1.376	1.963	3.078	6.314	12.71	15.89	31.82	63.66	127.3	318.3	636.6
2	.816	1.061	1.386	1.886	2.920	4.303	4.849	6.965	9.925	14.09	22.33	31.60
3	.765	.978	1.250	1.638	2.353	3.182	3.482	4.541	5.841	7.453	10.21	12.92
4	.741	.941	1.190	1.533	2.132	2.776	2.999	3.747	4.604	5.598	7.173	8.610
5	.727	.920	1.156	1.476	2.015	2.571	2.757	3.365	4.032	4.773	5.893	6.869
6	.718	.906	1.134	1.440	1.943	2.447	2.612	3.143	3.707	4.317	5.208	5.959
7	.711	.896	1.119	1.415	1.895	2.365	2.517	2.998	3.499	4.029	4.785	5.408
8	.706	.889	1.108	1.397	1.860	2.306	2.449	2.896	3.355	3.833	4.501	5.041
9	.703	.883	1.100	1.383	1.833	2.262	2.398	2.821	3.250	3.690	4.297	4.781
10	.700	.879	1.093	1.372	1.812	2.228	2.359	2.764	3.169	3.581	4.144	4.587
11	.697	.876	1.088	1.363	1.796	2.201	2.328	2.718	3.106	3.497	4.025	4.437
12	.695	.873	1.083	1.356	1.782	2.179	2.303	2.681	3.055	3.428	3.930	4.318
13	.694	.870	1.079	1.350	1.771	2.160	2.282	2.650	3.012	3.372	3.852	4.221
14	.692	.868	1.076	1.345	1.761	2.145	2.264	2.624	2.977	3.326	3.787	4.140
15	.691	.866	1.074	1.341	1.753	2.131	2.249	2.602	2.947	3.286	3.733	4.073
16	.690	.865	1.071	1.337	1.746	2.120	2.235	2.583	2.921	3.252	3.686	4.015
17	.689	.863	1.069	1.333	1.740	2.110	2.224	2.567	2.898	3.222	3.646	3.965
18	.688	.862	1.067	1.330	1.734	2.101	2.214	2.552	2.878	3.197	3.611	3.922
19	.688	.861	1.066	1.328	1.729	2.093	2.205	2.539	2.861	3.174	3.579	3.883
20	.687	.860	1.064	1.325	1.725	2.086	2.197	2.528	2.845	3.153	3.552	3.850
21	.686	.859	1.063	1.323	1.721	2.080	2.189	2.518	2.831	3.135	3.527	3.819
22	.686	.858	1.061	1.321	1.717	2.074	2.183	2.508	2.819	3.119	3.505	3.792
23	.685	.858	1.060	1.319	1.714	2.069	2.177	2.500	2.807	3.104	3.485	3.768
24	.685	.857	1.059	1.318	1.711	2.064	2.172	2.492	2.797	3.091	3.467	3.745
25	.684	.856	1.058	1.316	1.708	2.060	2.167	2.485	2.787	3.078	3.450	3.725
26	.684	.856	1.058	1.315	1.706	2.056	2.162	2.479	2.779	3.067	3.435	3.707
27	.684	.855	1.057	1.314	1.703	2.052	2.158	2.473	2.771	3.057	3.421	3.690
28	.683	.855	1.056	1.313	1.701	2.048	2.154	2.467	2.763	3.047	3.408	3.674
29	.683	.854	1.055	1.311	1.699	2.045	2.150	2.462	2.756	3.038	3.396	3.659
30	.683	.854	1.055	1.310	1.697	2.042	2.147	2.457	2.750	3.030	3.385	3.646
40	.681	.851	1.050	1.303	1.684	2.021	2.123	2.423	2.704	2.971	3.307	3.551
50	.679	.849	1.047	1.299	1.676	2.009	2.109	2.403	2.678	2.937	3.261	3.496
60	.679	.848	1.045	1.296	1.671	2.000	2.099	2.390	2.660	2.915	3.232	3.460
80	.678	.846	1.043	1.292	1.664	1.990	2.088	2.374	2.639	2.887	3.195	3.416
100	.677	.845	1.042	1.290	1.660	1.984	2.081	2.364	2.626	2.871	3.174	3.390
1000	.675	.842	1.037	1.282	1.646	1.962	2.056	2.330	2.581	2.813	3.098	3.300
∞	.674	.841	1.036	1.282	1.645	1.960	2.054	2.326	2.576	2.807	3.091	3.291
	50%	60%	70%	80%	90%	95%	96%	98%	99%	99.5%	99.8%	99.9%
	Confidence level C											

Significance

- M_1 & M_2 是否显著得不同?
 - Compute t . Select *significance level* (e.g. $sig = 5\%$)
 - Consult table for t-distribution: Find t value corresponding to $k-1$ degrees of freedom (here, 9)
 - t-分布对称: 通常显示分布的上百分点 % \rightarrow 查找值
confidence limit $z=sig/2$ (here, 0.025)
 - If $t > z$ or $t < -z$, 那么 t 的值位于拒绝域:
 - **Reject null hypothesis** that mean error rates of M_1 & M_2 are same
 - Conclude: statistically significant difference between M_1 & M_2
 - **Otherwise**, conclude that any difference is **chance**


模型选择: ROC Curves

- **ROC** (Receiver Operating Characteristics) curves: 图形比较分类模型
- 源于信号检测理论
- true positive rate和false positive rate间的折衷
- ROC 曲线下的面积就是模型正确率的度量
- 测试元组递减序排列: 最可能属于正类的排在最顶端
- The closer to the diagonal line (i.e., the closer the area is to 0.5), the less accurate is the model

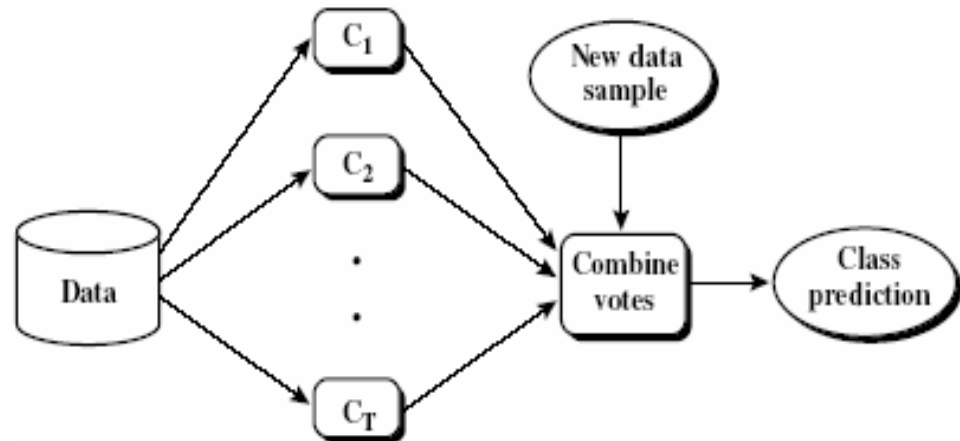


- 垂直坐标表示the true positive rate
- 水平坐标表示the false positive rate
- 同时显示对角线
- A model with perfect accuracy will have an area of 1.0

Chapter 6. 分类:集成方法

- 分类: 基本概念
- 决策树归纳
- 贝叶斯分类
- 基于规则的分类
- 模型评价与选择
- 提高分类准确率的技术:集成方法Ensemble Methods
- Summary 

集成方法: Increasing the Accuracy



- 集成方法 Ensemble methods
 - 使用多个模型的组合来提高accuracy
 - 组合多个学习的模型, M_1, M_2, \dots, M_k , 来获得一个提高的模型 M^*
- Popular ensemble methods
 - 装袋Bagging: 多个分类器的结果进行多数表决
 - 提升Boosting: 多个分类器的结果权重投票
 - 集成Ensemble: combining a set of heterogeneous classifiers

装袋Bagging

- 训练
 - 给定包含 d 个元组的数据 D , 在第 i 次迭代, 从 D 中有放回抽取 d 个样本组成训练集 D_i (i.e., bootstrap), 从 D_i 学习一个分类器 M_i
- 分类: 分类一个未知样本 X
 - 每个分类器 M_i 给出预测结果
 - 装袋分类器 M^* 计算投票, 把得票最多的类分配给 X
- 预测: 每个分类器预测的值的平均值
- 正确性Accuracy
 - 常常优于 D 上单个分类器的正确率
 - 对噪音数据: 不会很差, 更健壮
 - Proved improved accuracy in prediction

提升 Boosting

- 类比:咨询几个医生,在原来的诊断准确性的基础上分配权重,加权诊断的组合为结果
- **Boosting**如何工作?
 - Weights 分配给每个训练样本
 - 迭代学习一系列分类器
 - 学习 M_i 后,权重更新使得,后续得分分类器 M_{i+1} 更关注于 M_i 错误分类的训练样本
 - 最后的分类器 M^* 组合了每个独立分类器的投票,其中每个分类器的权重势其正确率的函数
- 可以扩充**Boosting** 算法用于数值预测
- 与**bagging**比较: **Boosting**倾向于得到更高的准确率,但有过拟合错误分类数据的风险

Adaboost (Freund and Schapire, 1997)

- 数据集含 d class-labeled 元组, $(\mathbf{X}_1, y_1), \dots, (\mathbf{X}_d, y_d)$
- 最初, 每个元组的权重为 $1/d$
- 在 k 轮中产生 k classifiers. 在第 i 轮,
 - 从 D 有放回抽取训练集 D_i (大小相等)
 - 每个元组被选中的概率基于其权重
 - 分类模型 M_i 学习自 D_i
 - 使用 D_i 为测试集计算误差率
 - 如果一个元组被错分, 权重增加, o.w. 否则下降
- 误差率: $err(\mathbf{X}_j)$ 为错误分类元组 \mathbf{X}_j 误差, 分类器 M_i 误差率是元组错误分类的权重和:

$$error(M_i) = \sum_j^d w_j \times err(\mathbf{X}_j)$$

- 分类器 M_i 投票权重为

$$\log \frac{1 - error(M_i)}{error(M_i)}$$

随机森林 Random Forest (Breiman 2001)

- Random Forest:
 - 每个分类器为 *decision tree* , 在每个结点上使用随机选出的属性来分裂产生判定树
 - 分类时, 每棵树投票得票最多的类返回结果
- 两种构造方法:
 - Forest-RI (*random input selection*): 每个结点随机选F 个属性为分裂的候选.用CART方法产生最大尺寸的树
 - Forest-RC (*random linear combinations*): 以现有属性的线性组合来产生新属性 (降低了单个分类器间的相关性)
- 准确率比得上Adaboost, 对误差和孤立点更稳健
- 每次分裂时对选出的候选属性数目不敏感, **faster than bagging or boosting**

分类类别不平衡数据集

- 类别不平衡问题.
- 传统的方法假定平衡的类别分布和相等的错误代价: 不适合
- 二元分类中典型的方法处理不平衡数据:
 - 过采样**Oversampling**: 对正类数据过/多采样
 - **Under-sampling**: 随机减少负类的样本
 - 阈值-移动**Threshold-moving**: 移动判定阈值 t , 使得少数类元组更容易识别, 减少 (昂贵的) 假阴性错误的机会
 - 集成技术:
- Still difficult for class imbalance problem on multiclass tasks

预测误差的度量

- 度量预测准确率: 度量预测值与真实值的距离

- 损失函数: 度量 y_i 和预测值 y_i' 间的误差

- 绝对误差Absolute error: $|y_i - y_i'|$

- 平方误差Squared error: $(y_i - y_i')^2$

- 检验误差 (泛化误差generalization error):

平均绝对误差: $\frac{\sum_{i=1}^d |y_i - y_i'|}{d}$

均方误差Mean squared error:

$$\frac{\sum_{i=1}^d (y_i - y_i')^2}{d}$$

- Relative absolute error: $\frac{\sum_{i=1}^d |y_i - y_i'|}{\sum_{i=1}^d |y_i - \bar{y}|}$

$$\frac{\sum_{i=1}^d (y_i - y_i')^2}{\sum_{i=1}^d (y_i - \bar{y})^2}$$

- Relative squared error:

- 均方误差夸大了离群点

Popularly use (square) root mean-square error, similarly, root relative squared error

What Is Prediction?

- (Numerical) 预测类似于分类
 - 构建一个模型
 - 利用模型来估计给定输入的连续或排序的值
- 与分类的不同
 - 分类是预测类别标签
 - 预测是模型连续值函数
- Major method for prediction: regression
 - 模型一个或多个预测变量和相应变量间的关系
- Regression analysis
 - 线性和多元回归
 - 非线性回归
 - 其他方法: generalized linear model, Poisson regression, log-linear models, regression trees

线性回归

- Linear regression: 包含一个响应变量 y 和一个预测变量 x

$$y = w_0 + w_1 x$$

- Method of least squares: estimates the best-fitting straight line

$$w_1 = \frac{\sum_{i=1}^{|D|} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{|D|} (x_i - \bar{x})^2} \quad w_0 = \bar{y} - w_1 \bar{x}$$

- 多元线性回归: 包含多个预测变量
 - Training data is of the form $(\mathbf{X}_1, y_1), (\mathbf{X}_2, y_2), \dots, (\mathbf{X}_{|D|}, y_{|D|})$
 - 对2-D数据, 有 $y = w_0 + w_1 x_1 + w_2 x_2$
 - 通常用统计软件包求解 SAS, S-Plus
 - 多个非线性函数可以表示成上面这种形式

非线性回归

- 某些非线性模型可以用多项式函数
- 多项式回归模型可以变换为线性回归模型. 例如

$$y = w_0 + w_1 x + w_2 x^2 + w_3 x^3$$

借助新变量: $x_2 = x^2$, $x_3 = x^3$

$$y = w_0 + w_1 x + w_2 x_2 + w_3 x_3$$

- 其他函数,如幂函数, 也可以转化为线性函数
- **Some models are intractable nonlinear (e.g., 指数相求和)**
 - 可能通过更复杂的公式综合计算, 得到最小二乘估计

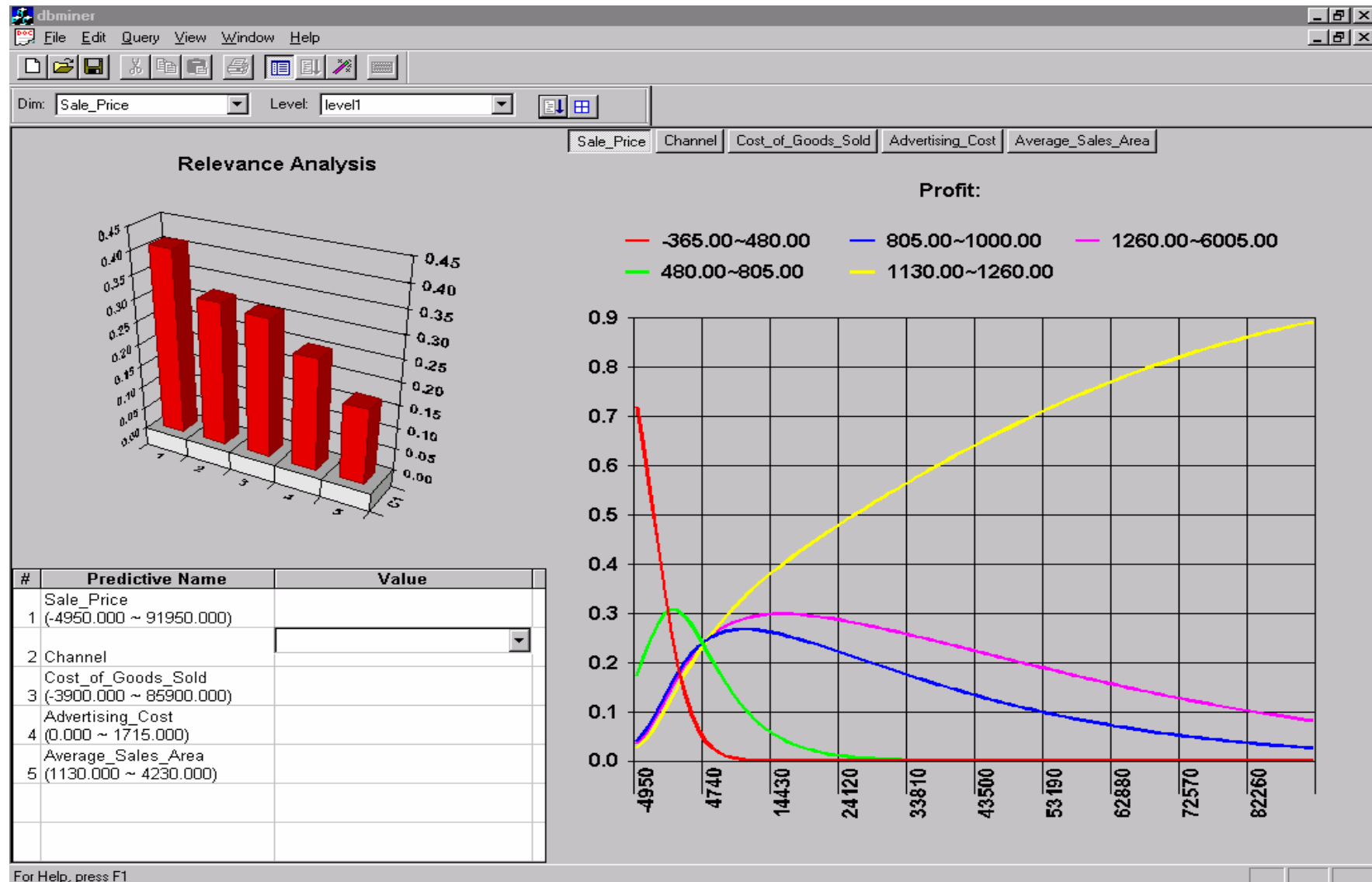
Other Regression-Based Models

- Generalized linear model:
 - Foundation on which linear regression can be applied to modeling categorical response variables
 - Variance of y is a function of the mean value of y , not a constant
 - Logistic regression: models the prob. of some event occurring as a linear function of a set of predictor variables
 - Poisson regression: models the data that exhibit a Poisson distribution
- Log-linear models: (for categorical data)
 - Approximate discrete multidimensional prob. distributions
 - Also useful for data compression and smoothing
- Regression trees and model trees
 - Trees to predict continuous values rather than class labels

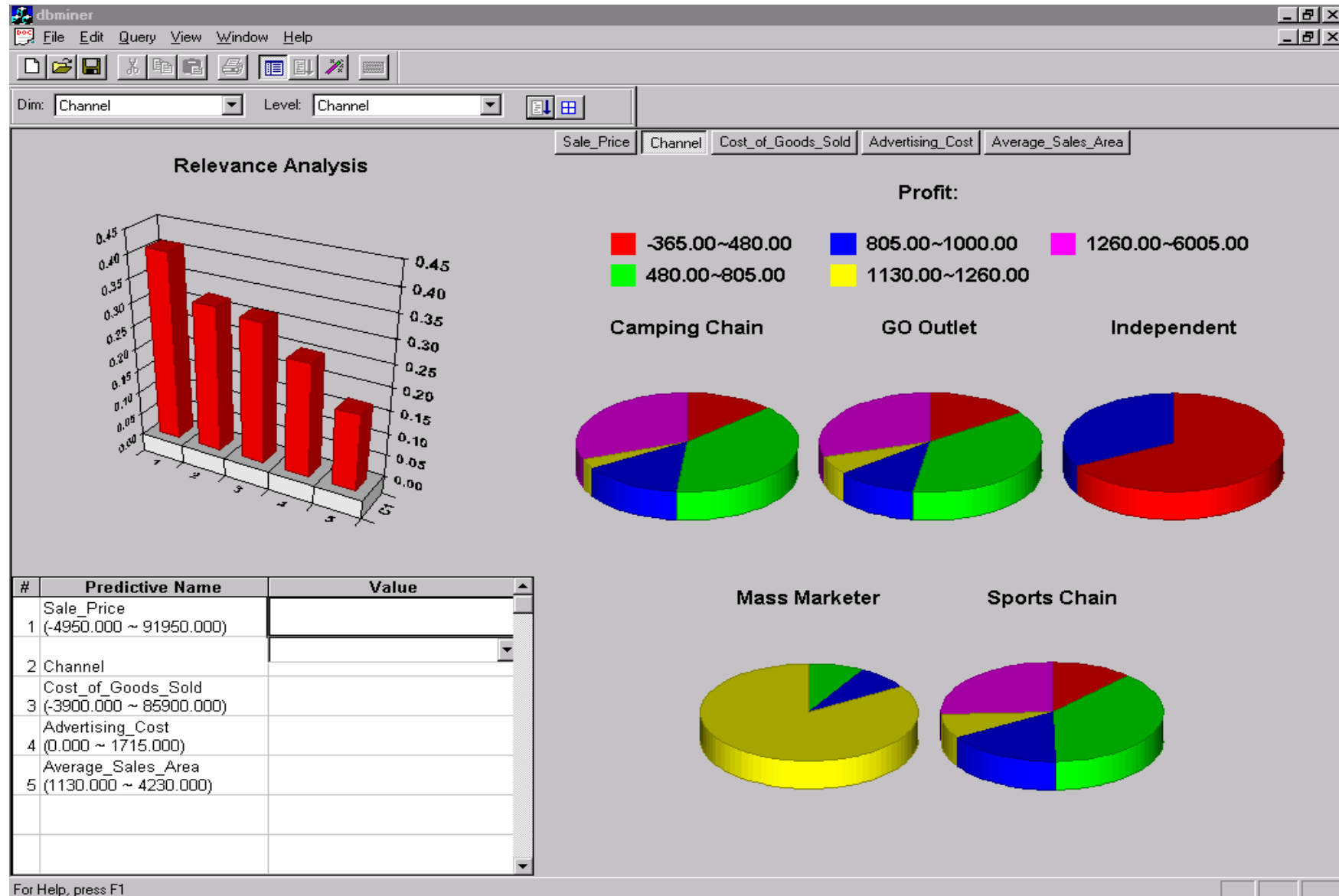
Regression Trees and Model Trees

- Regression tree: proposed in CART system (Breiman et al. 1984)
 - CART: Classification And Regression Trees
 - Each leaf stores a *continuous-valued prediction*
 - It is the *average value of the predicted attribute* for the training tuples that reach the leaf
- Model tree: proposed by Quinlan (1992)
 - Each leaf holds a regression model—a multivariate linear equation for the predicted attribute
 - A more general case than regression tree
- Regression and model trees tend to be more accurate than linear regression when the data are not represented well by a simple linear model


Prediction: Numerical Data



Prediction: Categorical Data



Chapter 6. 分类: 基本概念

- 分类: 基本概念
- 决策树归纳
- 贝叶斯分类
- 基于规则的分类
- 模型评价与选择
- 提高分类准确率的技术:集成方法Ensemble Methods
- Summary 

Summary (I)

- **Classification** is a form of data analysis that extracts **models** describing important data classes.
- Effective and scalable methods have been developed for **decision tree induction**, **Naive Bayesian classification**, **rule-based classification**, and many other classification methods.
- **Evaluation metrics** include: accuracy, sensitivity, specificity, precision, recall, F measure, and F_β measure.
- **Stratified k-fold cross-validation** is recommended for accuracy estimation. **Bagging** and **boosting** can be used to increase overall accuracy by learning and combining a series of individual models.

Summary (II)

- **Significance tests** and **ROC curves** are useful for model selection.
- There have been numerous **comparisons of the different classification** methods; the matter remains a research topic
- No single method has been found to be superior over all others for all data sets
- Issues such as accuracy, training time, robustness, scalability, and interpretability must be considered and can involve trade-offs, further complicating the quest for an overall superior method

References (1)

- C. Apte and S. Weiss. **Data mining with decision trees and decision rules.** Future Generation Computer Systems, 13, 1997
- C. M. Bishop, **Neural Networks for Pattern Recognition.** Oxford University Press, 1995
- L. Breiman, J. Friedman, R. Olshen, and C. Stone. **Classification and Regression Trees.** Wadsworth International Group, 1984
- C. J. C. Burges. **A Tutorial on Support Vector Machines for Pattern Recognition.** *Data Mining and Knowledge Discovery*, 2(2): 121-168, 1998
- P. K. Chan and S. J. Stolfo. **Learning arbiter and combiner trees from partitioned data for scaling machine learning.** KDD'95
- H. Cheng, X. Yan, J. Han, and C.-W. Hsu, **Discriminative Frequent Pattern Analysis for Effective Classification**, ICDE'07
- H. Cheng, X. Yan, J. Han, and P. S. Yu, **Direct Discriminative Pattern Mining for Effective Classification**, ICDE'08
- W. Cohen. **Fast effective rule induction.** ICML'95
- G. Cong, K.-L. Tan, A. K. H. Tung, and X. Xu. **Mining top-k covering rule groups for gene expression data.** SIGMOD'05

References (2)

- A. J. Dobson. **An Introduction to Generalized Linear Models**. Chapman & Hall, 1990.
- G. Dong and J. Li. **Efficient mining of emerging patterns: Discovering trends and differences**. KDD'99.
- R. O. Duda, P. E. Hart, and D. G. Stork. **Pattern Classification**, 2ed. John Wiley, 2001
- U. M. Fayyad. **Branching on attribute values in decision tree generation**. AAAI'94.
- Y. Freund and R. E. Schapire. **A decision-theoretic generalization of on-line learning and an application to boosting**. J. Computer and System Sciences, 1997.
- J. Gehrke, R. Ramakrishnan, and V. Ganti. **Rainforest: A framework for fast decision tree construction of large datasets**. VLDB'98.
- J. Gehrke, V. Gant, R. Ramakrishnan, and W.-Y. Loh, **BOAT -- Optimistic Decision Tree Construction**. SIGMOD'99.
- T. Hastie, R. Tibshirani, and J. Friedman. **The Elements of Statistical Learning: Data Mining, Inference, and Prediction**. Springer-Verlag, 2001.
- D. Heckerman, D. Geiger, and D. M. Chickering. **Learning Bayesian networks: The combination of knowledge and statistical data**. Machine Learning, 1995.

References (3)

- T.-S. Lim, W.-Y. Loh, and Y.-S. Shih. **A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms.** Machine Learning, 2000.
- J. Magidson. **The Chaid approach to segmentation modeling: Chi-squared automatic interaction detection.** In R. P. Bagozzi, editor, Advanced Methods of Marketing Research, Blackwell Business, 1994.
- M. Mehta, R. Agrawal, and J. Rissanen. **SLIQ : A fast scalable classifier for data mining.** EDBT'96.
- T. M. Mitchell. **Machine Learning.** McGraw Hill, 1997.
- S. K. Murthy, **Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey,** Data Mining and Knowledge Discovery 2(4): 345-389, 1998
- J. R. Quinlan. **Induction of decision trees.** *Machine Learning*, 1:81-106, 1986.
- J. R. Quinlan and R. M. Cameron-Jones. **FOIL: A midterm report.** ECML'93.
- J. R. Quinlan. **C4.5: Programs for Machine Learning.** Morgan Kaufmann, 1993.
- J. R. Quinlan. **Bagging, boosting, and C4.5.** AAAI'96

References (4)

- R. Rastogi and K. Shim. **Public: A decision tree classifier that integrates building and pruning.** VLDB'98.
- J. Shafer, R. Agrawal, and M. Mehta. **SPRINT : A scalable parallel classifier for data mining.** VLDB'96.
- J. W. Shavlik and T. G. Dietterich. **Readings in Machine Learning.** Morgan Kaufmann, 1990.
- P. Tan, M. Steinbach, and V. Kumar. **Introduction to Data Mining.** Addison Wesley, 2005.
- S. M. Weiss and C. A. Kulikowski. **Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems.** Morgan Kaufman, 1991.
- S. M. Weiss and N. Indurkha. **Predictive Data Mining.** Morgan Kaufmann, 1997.
- I. H. Witten and E. Frank. **Data Mining: Practical Machine Learning Tools and Techniques**, 2ed. Morgan Kaufmann, 2005.
- X. Yin and J. Han. **CPAR: Classification based on predictive association rules.** SDM'03
- H. Yu, J. Yang, and J. Han. **Classifying large data sets using SVM with hierarchical clusters.** KDD'02

SCALABLE DECISION TREE INDUCTION

Methods

- **SLIQ** (EDBT'96 — Mehta et al.)
 - Builds an index for each attribute and only class list and the current attribute list reside in memory
- **SPRINT** (VLDB'96 — J. Shafer et al.)
 - Constructs an attribute list data structure
- **PUBLIC** (VLDB'98 — Rastogi & Shim)
 - Integrates tree splitting and tree pruning: stop growing the tree earlier
- **RainForest** (VLDB'98 — Gehrke, Ramakrishnan & Ganti)
 - Builds an AVC-list (attribute, value, class label)
- **BOAT** (PODS'99 — Gehrke, Ganti, Ramakrishnan & Loh)
 - Uses bootstrapping to create several small samples

Data Cube-Based Decision-Tree Induction

- Integration of generalization with decision-tree induction (Kamber et al.'97)
- Classification at primitive concept levels
 - E.g., precise temperature, humidity, outlook, etc.
 - Low-level concepts, scattered classes, bushy classification-trees
 - Semantic interpretation problems
- Cube-based multi-level classification
 - Relevance analysis at multi-levels
 - Information-gain analysis with dimension + level

第7章 聚类分析

- 什么是聚类（**Clustering**）分析？
- 聚类分析中的数据类型
- 主要聚类方法分类
- 划分方法（**Partitioning Methods**）
- 层次方法（**Hierarchical Methods**）
- 基于密度的方法（**Density-Based Methods**）
- 基于网格的方法（**Grid-Based Methods**）
- 基于模型的聚类方法（**Model-Based Clustering Methods**）
- 孤立点分析（**Outlier Analysis**）
- 小结

什么是聚类分析？

- 聚类: 数据对象的集合/簇 (**cluster**)
 - 同一簇中的对象彼此相似
 - 不同簇中的对象彼此相异
- 聚类分析
 - 将数据对象分组成为多个类或簇
- 聚类是**无指导的**分类: 没有预先定义的类
- 典型应用
 - 作为洞察数据内部分布的独一无二的工具
 - 作为其它算法的预处理步骤

聚类的一般应用

- 模式识别
- 空间数据分析
 - 聚类产生**GIS**(地理信息系统)的专题地图**thematic maps**
 - 在空间数据挖掘中检测空间聚类并解释它们
- 图象处理
- 经济科学 (特别是市场研究)
- **WWW**
 - 文本分类
 - **Web**日志数据聚类, 发现类似访问模式群

聚类应用的例子

- 市场营销:

帮助市场营销者发现他们的基本顾客的不同组群，然后利用这一知识制定有针对性的营销计划

- 国土利用

在地球观测数据库中识别类似的国土使用区域

- 保险

对汽车保险持有者的分组

- 城市规划

根据房子的类型，价值，和地理位置对一个城市中房屋的分组

- 地震研究

应当将观测到的地震震中沿大陆板块断裂进行聚类

什么是好的聚类方法？

- 一个好的聚类方法应当产生高质量的聚类
 - 类内相似性高
 - 类间相似性低
- 聚类结果的质量依赖于方法所使用的相似性度量和它的实现.
- 聚类方法的质量也用它发现某些或全部隐藏的模式的能力来度量

数据挖掘对聚类的要求

- 可伸缩性
 - 有的算法当数据对象少于**200**时处理很好,但对大量数据对象偏差较大
 - 大型数据库包含数百万个对象
- 处理不同属性类型的能力
 - 许多算法专门用于数值类型的数据
 - 实际应用涉及不同的数据类型,**i.e.** 混合了数值和分类数据
- 发现任意形状的聚类
 - 基于距离的聚类趋向于发现具有相近尺度和密度的球状簇
 - 一个簇可能是任意形状的

数据挖掘对聚类的要求(续)

- 用于决定输入参数的领域知识最小化
 - 许多聚类算法要求用户输入一定的参数,如希望产生的簇的数目.聚类结果对于输入参数十分敏感
 - 参数难以确定,增加了用户的负担,使聚类质量难以控制
- 处理噪声数据和孤立点的能力
 - 一些聚类算法对于噪音数据敏感,可能导致低质量的聚类结果
 - 现实世界中的数据库大都包含了孤立点,空缺,或者错误的数据
- 对于输入记录的顺序不敏感
 - 一些聚类算法对于输入数据的顺序是敏感的,以不同的次序输入会导致不同的聚类

数据挖掘对聚类的要求(续)

- 高维性 (**high dimensionality**)
 - 许多聚类算法擅长处理低维的数据,可能只涉及两到三维
 - 数据库或者数据仓库可能包含若干维或者属性,数据可能非常稀疏,而且高度偏斜
- 整合用户指定的约束
 - 现实世界的应用可能需要在各种约束条件下进行聚类
 - 要找到既满足特定的约束,又具有良好聚类特性的数据分组是一项具有挑战性的任务
- 可解释性和可用性
 - 用户希望聚类结果是可解释的,可理解的,和可用的
 - 聚类可能需要和特定的语义解释和应用相联系

第7章. 聚类分析

- 什么是聚类（**Clustering**）分析？
- 聚类分析中的数据类型
- 主要聚类方法分类
- 划分方法（**Partitioning Methods**）
- 层次方法（**Hierarchical Methods**）
- 基于密度的方法（**Density-Based Methods**）
- 基于网格的方法（**Grid-Based Methods**）
- 基于模型的聚类方法（**Model-Based Clustering Methods**）
- 孤立点分析（**Outlier Analysis**）
- 小结

数据结构

- 数据矩阵
 - (two modes)

$$\begin{bmatrix} x_{11} & \dots & x_{1f} & \dots & x_{1p} \\ \dots & \dots & \dots & \dots & \dots \\ x_{i1} & \dots & x_{if} & \dots & x_{ip} \\ \dots & \dots & \dots & \dots & \dots \\ x_{n1} & \dots & x_{nf} & \dots & x_{np} \end{bmatrix}$$

- 相异度矩阵
(Dissimilarity matrix)
 - (one mode)

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \dots & \dots & 0 \end{bmatrix}$$

评估聚类的质量

- 有一个单独的“质量”函数, 它度量聚类的“好坏”.
- 很难定义“足够类似”或“足够好”
 - 对此问题是相当主观的.
- 相异度/相似度矩阵
 - 相似性用距离函数表示, 通常记作 $d(i, j)$
- 对于区间标度变量, 二元变量, 标称变量, 序数和比例标度变量, 距离函数的定义通常是很不相同的.
- 根据应用和数据语义, 不同的变量应赋予不同的权.

聚类分析的数据类型

- 区间标度变量(**Interval-scaled variables**)
- 二元变量(**Binary variables**)
- 标称(名词性), 序数, 和比例标度变量(**Nominal, ordinal, and ratio variables**)
- 混合类型变量(**Variables of mixed types**)

区间标度变量

- 区间标度变量：一种粗略线形标度的连续度量
- 为了避免度量单位的影响，数据标准化

- (1)计算平均绝对偏差：

$$s_f = \frac{1}{n}(|x_{1f} - m_f| + |x_{2f} - m_f| + \dots + |x_{nf} - m_f|)$$

其中 $m_f = \frac{1}{n}(x_{1f} + x_{2f} + \dots + x_{nf})$.

- (2)计算标准化的度量值 (*z-score*)

$$z_{if} = \frac{x_{if} - m_f}{s_f}$$

- 使用平均绝对偏差比使用标准差更具有鲁棒性

对象之间的相似性/相异性

- 通常, 使用距离来度量两个数据对象之间的相似性/相异性
- 常用的距离包括:闵可夫斯基(Minkowski) 距离:

$$d(i, j) = \sqrt[q]{(|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + \dots + |x_{ip} - x_{jp}|^q)}$$

其中 $i = (x_{i1}, x_{i2}, \dots, x_{ip})$ 和 $j = (x_{j1}, x_{j2}, \dots, x_{jp})$ 是两个 p -维数据对象(q 正整数)

- 如果 $q = 1$, d 是曼哈坦 (Manhattan) 距离

$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \dots + |x_{ip} - x_{jp}|$$

对象之间的相似性/相异性

- 如果 $q = 2$, d 是欧几里德(Euclidean)距离 :

$$d(i, j) = \sqrt{(|x_{i_1} - x_{j_1}|^2 + |x_{i_2} - x_{j_2}|^2 + \dots + |x_{i_p} - x_{j_p}|^2)}$$

- 距离的性质

- 非负性: $d(i, j) \geq 0$
- 自身到自身的距离为0: $d(i, i) = 0$
- 对称性: $d(i, j) = d(j, i)$
- 三角不等式: $d(i, j) \leq d(i, k) + d(k, j)$

- 也可以使用加权的距离, 如加权的欧几里德距离

$$d(i, j) = \sqrt{w_1(|x_{i_1} - x_{j_1}|^2 + w_2|x_{i_2} - x_{j_2}|^2 + \dots + w_p|x_{i_p} - x_{j_p}|^2)}$$

二元变量

- 二元变量(binary variable)只有两个状态0或1. 0表示该变量为空, 1表示该变量存在
 - 例如, 描述病人的变量 $smoker$, 1表示病人抽烟, 而0表示病人不抽烟
- 计算二元变量的相似度
 - 假定所有二元变量具有相同的权重, 则得到一个两行两列的可能性表(contingency table)

		对象 j		
		1	0	sum
对象 i	1	q	r	$q+r$
	0	s	t	$s+t$
	sum	$q+s$	$r+t$	p

二元变量

- 对称的: 二元变量的两个状态具有同等价值,并具有相同的权重
 - 例: 性别是对称的二元变量
- 恒定的相似度: 基于对称的二元变量的相似度, 当一些或全部二元变量编码改变时, 计算结果不会发生变化
- 对称的二元变量的相异度计算-----简单匹配系数

$$d(i, j) = \frac{r + s}{q + r + s + t}$$

- 不对称的: 二元变量的两个状态的输出不是同样重要
 - 例: 疾病检查结果的肯定和否定.

二元变量(续)

- 根据惯例, 比较重要的输出结果是出现几率较小的
 - 例: **HIV**阳性是比较重要的结果,出现几率较小, 而**HIV**阴性(正常情况)出现几率较大
- 通常, 比较重要的输出结果编码为**1**, 另一种结果编码为**0**
- 两个都取**1**的情况(正匹配)比两个都取**0**的情况(负匹配)更有意义. ----非恒定的相似度
- 对于非对称的相似度, 负匹配数目**t**被忽略.
 - 采用**Jaccard**系数

$$d(i, j) = \frac{r+s}{q+r+s} = 1 - \frac{q}{q+r+s} = 1 - Jaccard(i, j)$$

二元变量之间的相异度

■ 例

Name	Gender	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	P	N	N	N	N

- **gender** 是对称的
- 其余都不是对称的
- **Y**和**P**的值设置为**1**, 而 **N**的值设置为 **0**

$$d(\text{jack}, \text{mary}) = \frac{0 + 1}{2 + 0 + 1} = 0.33$$

$$d(\text{jack}, \text{jim}) = \frac{1 + 1}{1 + 1 + 1} = 0.67$$

$$d(\text{jim}, \text{mary}) = \frac{1 + 2}{1 + 1 + 2} = 0.75$$

标称变量-分类变量，名义变量

- 标称变量(Nominal variable)是二元变量的拓广, 它可以取多于两种状态值, 如, **red, yellow, blue, green**
- 方法1: 简单匹配

- m : 状态取值匹配的变量数目, p : 变量总数

$$d(i, j) = \frac{p - m}{p}$$

- 方法 2: (可以用非对称的二元变量对标称变量编码) 使用大量二元变量,
 - 对M个标称状态的每一个, 创建一个新的二元变量. 对于一个有特定状态值的对象, 对应状态值的二元变量值置1, 其余二元变量的值置0

序数型变量

- 序数型变量(**ordinal variable**)可以是离散的,也可以是连续的
 - 离散的序数型变量类似于**标称变量**,但**序数型变量**的M个状态是以有意义的序列排序
 - 连续的序数型变量看起来像一个未知刻度的连续数据的集合.
 - 值的相对顺序是必要的,而其实际的大小则不重要
 - 将区间标度变量的值域划分为有限个区间,从而将其值离散化,也可以得到序数型变量
- 序数型变量的值可以映射为秩(**rank**).
 - 例如,假设变量f有 M_f 个状态,这些有序的状态定义了一个排列 $1, \dots, M_f$

序数型变量(续)

- 相异度计算可以用类似于区间标度变量的方法处理
 - 设第 i 个对象 f 的值为 x_{if} , 用对应的秩 r_{if} 替代 x_{if} , 其中 $r_{if} \in \{1, \dots, M_f\}$
 - 将每个变量的值域映射到 $[0, 1]$ 区间, 以便每个变量都具有相同的权重: 用下式替换 r_{if}

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

- 使用区间标度变量计算距离的方法计算相异度, z_{if} 作为第 i 个对象 f 的值

比例标度变量

- 比例标度变量(**Ratio-scaled variable**)非线性的刻度上取正的度量值，例如指数标度，近似地遵循如下的公式

$$Ae^{Bt} \text{ 或 } Ae^{-Bt}$$

- 相异度计算：
 - 采用与处理区间标度变量同样的方法 — *不是好的选择!*
(为什么?—标度可能被扭曲)
 - 进行对数变换

$$y_{if} = \log(x_{if})$$

- 将 x_{if} 看作连续的序数型数据, 将其秩作为区间标度值
- 方法的选取取决于应用, 但后两种方法比较有效

混合类型变量

- 数据库可能包含所有六种类型
 - 对称的二元变量, 不对称的二元变量, 标称的, 序数的, 区间的, 比例标度的
- 如何计算混合类型变量描述的对象的相关度?
 - 方法1: 将变量按类型分组, 对每种类型的变量单独进行聚类分析
 - 如果这些分析得到兼容的结果, 这种方法是可行的
 - 在实际的应用中, 这种方法行不通
 - 方法2: 将所有的变量一起处理, 只进行一次聚类分析.
 - 将不同类型的变量组合在单个相关度矩阵中, 把所有变量转换到共同的值域区间[0.0, 1.0]上

混合类型变量(续)

- 假设数据集包含 p 个不同类型的变量, 对象 i 和 j 之间的相异度 $d(i,j)$ 定义为

$$d(i,j) = \frac{\sum_{f=1}^p \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^p \delta_{ij}^{(f)}}$$

其中, 如果 x_{if} 或 x_{jf} 缺失(即对象 i 或对象 j 没有变量 f 的度量值)或者 $x_{if}=x_{jf}=0$, 且变量 f 是不对称的二元变量, 则指示项 $\delta_{ij}^{(f)} = 0$; 否则, 指示项 $\delta_{ij}^{(f)} = 1$

- 变量 f 对 i 和 j 之间相异度的计算方式与其具体类型有关
 - 如果 f 是二元或标称变量:
如果 $x_{if}=x_{jf}$, $d_{ij}^{(f)} = 0$; 否则 $d_{ij}^{(f)} = 1$

混合类型变量(续)

- 如果 f 是区间标度变量, 使用规格化的距离

$$d_{ij}^{(f)} = \frac{|x_{if} - x_{jf}|}{\max_h x_{hf} - \min_h x_{hf}}$$

- 如果 f 是序数型或比例标度型变量

- 计算秩 r_{if} 和

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

- 将 z_{if} 作为区间标度变量对待

向量对象

- 向量x和y
 - 余弦度量

$$\cos(x, y) = \frac{x^t \cdot y}{\|x\| \cdot \|y\|} = \frac{\sum_{i=1}^p x_i y_i}{\sqrt{\sum_{i=1}^p x_i^2 \cdot \sum_{i=1}^p y_i^2}}$$

- Tanimoto系数 $s(x, y) = \frac{x^t \cdot y}{x^t \cdot x + y^t \cdot y - x^t \cdot y}$

第8章. 聚类分析

- 什么是聚类（**Clustering**）分析？
- 聚类分析中的数据类型
- 主要聚类方法分类
- 划分方法（**Partitioning Methods**）
- 层次方法（**Hierarchical Methods**）
- 基于密度的方法（**Density-Based Methods**）
- 基于网格的方法（**Grid-Based Methods**）
- 基于模型的聚类方法（**Model-Based Clustering Methods**）
- 孤立点分析（**Outlier Analysis**）
- 小结

主要聚类方法的分类

1. 划分方法(Partitioning method): 构造 n 个对象数据库 D 的划分, 将其划分成 k 个聚类满足如下的要求:
 - 1) 每个组至少包含一个对象
 - 2) 每个对象必须属于且只属于一个组
 - 在某些模糊划分技术中, 第二个要求可以放宽
 - 基本方法: 首先创建一个初始划分. 然后采用一种迭代的重定位技术, 尝试通过在划分间移动对象来改进划分
 - 好的划分的一般准则: 在同一个类中的对象之间尽可能“接近”或相关, 而不同类中的对象之间尽可能“远离”或不同

主要聚类方法的分类(续)

- 全局最优: 穷举所有可能的划分
- 启发式方法: k -平均值(k - means)和 k -中心点(k - medoids)算法
 - k -平均值(MacQueen'67): 每个簇用该簇中对象的平均值来表示
 - k -中心点或 PAM (Partition around medoids) (Kaufman & Rousseeuw'87): 每个簇用接近聚类中心的一个对象来表示
- 这些启发式算法适合发现中小规模数据库中的球状聚类
- 对于大规模数据库和处理任意形状的聚类,这些算法需要进一步扩展

主要聚类方法的分类(续)

- 2. 层次方法(Hierarchy method): 对给定数据对象集合进行层次的分解
 - 两种层次方法
 - 凝聚方法, 也称为自底向上的方法: 开始将每个对象作为单独的一个组; 然后继续地合并相近的对象或组, 直到所有的组合并为一个(层次的最上层), 或者达到一个终止条件
 - 分裂方法, 也称为自顶向下的方法: 开始将所有的对象置于一个簇; 在迭代的每一步, 一个簇被分裂为更小的簇, 直到最终每个对象在单独的一个簇中, 或者达到一个终止条件

主要聚类方法的分类(续)

- 层次方法的缺点: 一个步骤一旦完成便不能被撤消.
 - 该规定可以避免考虑选择不同的组合, 减少计算代价
 - 问题: 不能更正错误的决定
- 改进层次聚类结果的措施
 - 在每层划分中, 仔细分析对象间的“联接”, 例如**CURE**和**Chameleon**中的做法
 - 综合层次凝聚和迭代的重新定位方法。首先用自底向上的层次算法, 然后用迭代的重新定位来改进结果。例如在**BIRCH**中的方法

主要聚类方法的分类(续)

3. 基于密度的方法(Density-based method): 基于密度函数
 - 基本思想: 只要临近区域的密度(对象或数据点的数目)超过某个阈值, 就继续聚类. 也就是说, 对给定类中的每个数据点, 在一个给定范围的区域中必须至少包含一定数目的点
 - 该方法可以用来过滤“噪音”数据, 发现任意形状的簇
 - **DBSCAN**是一个有代表性的基于密度的方法, 它根据一个密度阈值来控制簇的增长
 - **OPTICS**是另一个基于密度的方法, 它为自动的, 交互的聚类分析计算一个聚类顺序

主要聚类方法的分类(续)

- 4. 基于网格的方法(Grid-based method): 把对象空间量化为有限数目的单元, 形成了一个网格结构. 所有的聚类操作都在这个网格结构(即量化的空间)上进行
 - 这种方法的主要优点是它的处理速度很快, 其处理时间独立于数据对象的数目, 只与量化空间中每一维的单元数目有关
 - STING是基于网格方法的一个典型例子
 - CLIQUE和WaveCluster这两种算法既是基于网格的, 又是基于密度的

主要聚类方法的分类(续)

5. 基于模型的方法(Model-based Method): 基于模型的方法为每个簇假定了一个模型, 寻找数据对给定模型的最佳拟合

第7章. 聚类分析

- 什么是聚类（**Clustering**）分析？
- 聚类分析中的数据类型
- 主要聚类方法分类
- 划分方法（**Partitioning Methods**）
- 层次方法（**Hierarchical Methods**）
- 基于密度的方法（**Density-Based Methods**）
- 基于网格的方法（**Grid-Based Methods**）
- 基于模型的聚类方法（**Model-Based Clustering Methods**）
- 孤立点分析（**Outlier Analysis**）
- 小结

划分方法

- 划分方法: 构造 n 个对象数据库 D 的划分, 将其划分成 k 个聚类
- 给定 k , 找 k 个 *clusters* 对于选定的划分标准它是最优的
 - 全局最优(Global optimal): 枚举所有的划分
 - 启发式方法(Heuristic methods): k -平均(k -means)和 k -中心点(k -medoids)算法
 - k -平均(MacQueen'67): 每个簇用簇的重心(簇的平均值) 表示
 - k -中心点或PAM (Partition around medoids) (Kaufman & Rousseeuw'87): 每个簇用接近聚类中心的一个对象来表示

k -平均聚类算法

- 算法： k -平均

- (1) 任意选择 k 个对象作为初始的簇中心;
- (2) repeat
- (3) 根据簇中对象的平均值, 将每个对象(重新)赋给最类似的簇;
- (4) 更新簇的平均值, 即重新计算每个簇中对象的平均值;
- (5) until 不再发生变化

- 通常, 采用平方误差准则作为收敛函数, 其定义如下

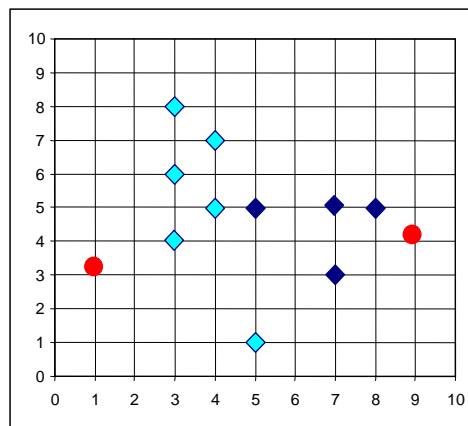
$$E = \sum_{i=1}^k \sum_{p \in C_i} |p - m_i|^2$$

其中, m_i 是簇 C_i 的平均值

该准则试图使生成的结果簇尽可能紧凑, 独立

k -平均聚类算法(续)

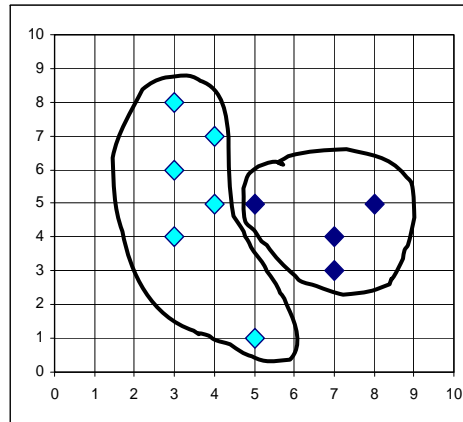
■ 例



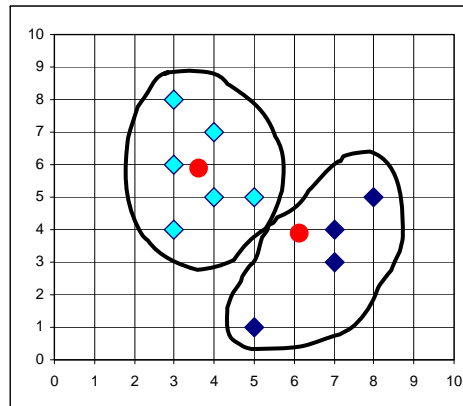
K=2

任意选择 **K** 个对象作为初始聚类中心

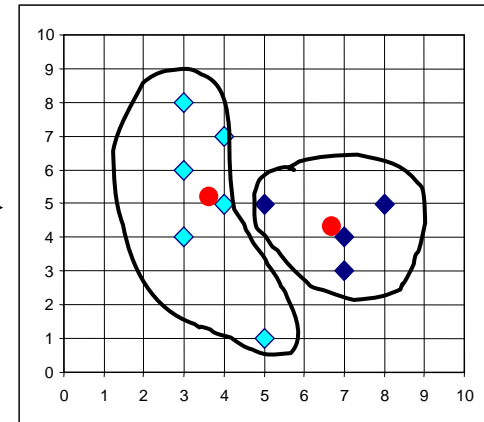
将每个对象赋给最类似的中心



重新赋值

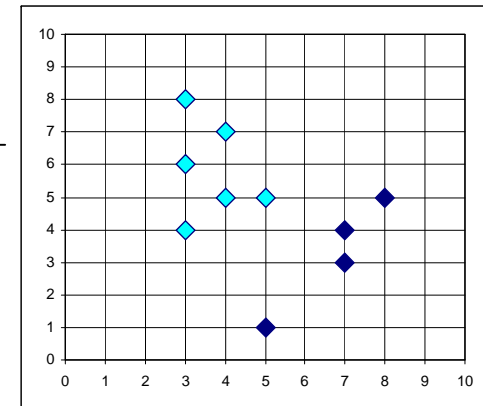


更新簇的平均值



重新赋值

更新簇的平均值



k -平均聚类算法(续)

- 优点: 相对有效性: $O(tkn)$,
其中 n 是对象数目, k 是簇数目, t 是迭代次数; 通常,
 $k, t \ll n$.
 - 比较: PAM: $O(k(n-k)^2)$, CLARA: $O(ks^2 + k(n-k))$
- 当结果簇是密集的, 而簇与簇之间区别明显时, 它的效果较好
- Comment: 常常终止于局部最优.
- 全局最优可以使用诸如确定性的退火(*deterministic annealing*)和遗传算法(*genetic algorithms*)等技术得到

k -平均聚类算法(续)

- 弱点

- 只有在簇的平均值(*mean*)被定义的情况下才能使用.可能不适用于某些应用,例如涉及有分类属性的数据
- 需要预先指定簇的数目 k ,
- 不能处理噪音数据和孤立点(*outliers*)
- 不适合用来发现具有非凸形状(*non-convex shapes*)的簇

k -平均方法的变种

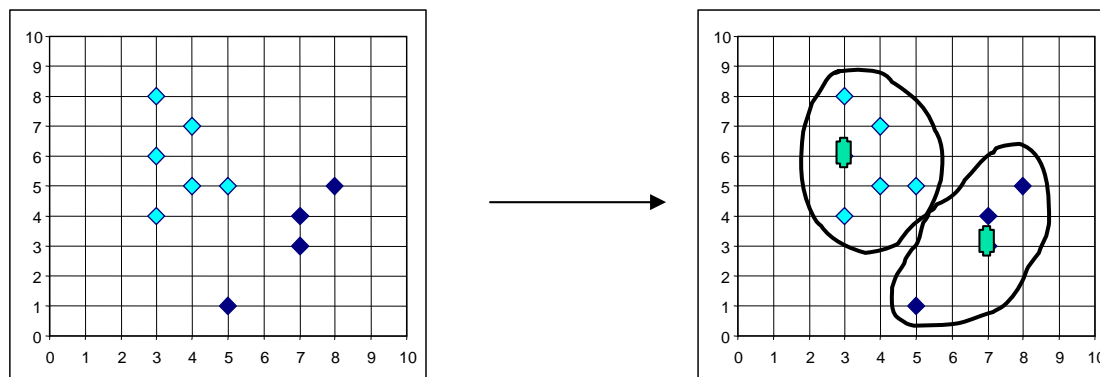
- k -平均方法的变种, 它们在以下方面有所不同
 - 初始 k 个平均值的选择
 - 相异度的计算
 - 计算聚类平均值的策略
- 较好的策略: 先用层次凝聚算法决定簇的数目, 并产生初始聚类, 然后用迭代重定位改进聚类结果
- 处理分类属性: k -模(k -modes) 方法(Huang'98)
 - 用模(modes众数)替代聚类的平均值
 - 使用新的相异性度量方法来处理分类对象
 - 用基于频率的方法来修改聚类的模
- k -原型(k -prototype)方法: k -平均和 k -模的结合, 处理具有数值和分类属性的数据

k -平均方法的变种(续)

- **EM(Expectation Maximization, 期望最大)算法**
 - 以另一种方式对 k -means方法进行了扩展: 不把对象分配给一个确定的簇, 而是根据对象与簇之间隶属关系发生的概率来分派对象
- 怎样增强 k -means算法的可扩展性?
 - 数据分成三种区域:
 1. 可废弃的: 一个对象与某个簇的隶属关系是确定的
 2. 可压缩的: 一个对象不是可废弃的, 但属于某个**紧密**的子簇
 3. 必须在主存: 既不是可废弃的, 又不是可压缩的
 - 迭代的算法只包含可压缩的对象和必须在主存中的对象的聚类特征, 从而将一个基于二级存储的算法变成了基于主存的算法

k -中心点聚类方法

- k -平均值算法对孤立点很敏感!
 - 因为具有特别大的值的对象可能显著地影响数据的分布.
- k -中心点(k -Medoids): 不采用簇中对象的平均值作为参照点, 而是选用簇中位置最中心的对象, 即中心点(medoid)作为参照点.



k -中心点聚类方法(续)

- 找聚类中的代表对象(中心点)
- **PAM (Partitioning Around Medoids, 1987)**
 - 首先为每个簇随意选择一个代表对象, 剩余的对象根据其代表对象的距离分配给最近的一个簇; 然后反复地用非代表对象来替代代表对象, 以改进聚类的质量
 - **PAM** 对于较小的数据集非常有效, 但不能很好地扩展到大型数据集
- **CLARA (Kaufmann & Rousseeuw, 1990)** 抽样
- **CLARANS (Ng & Han, 1994)**: 随机选样

k -中心点聚类方法(续)

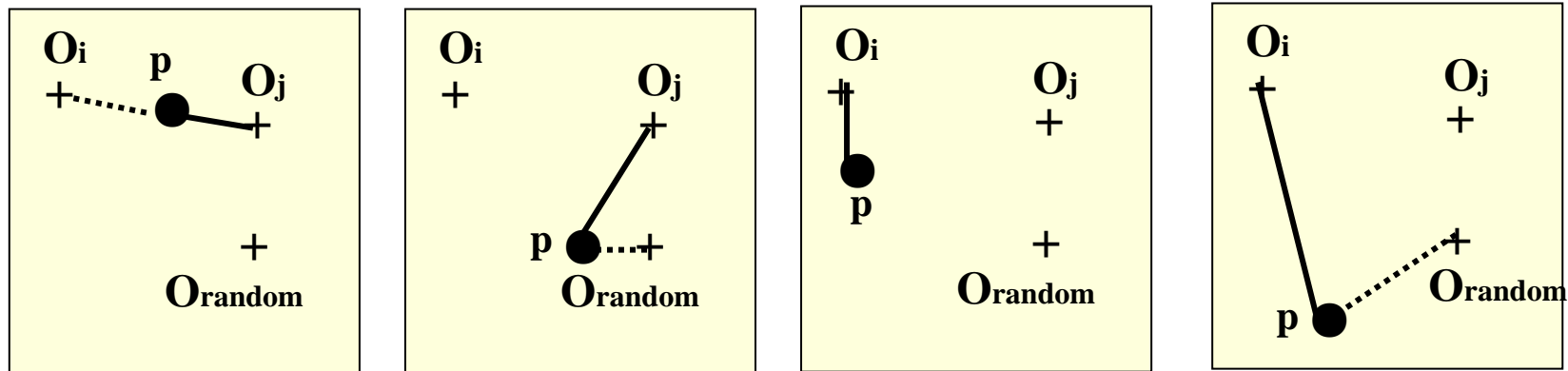
- 基本思想:

- 首先为每个簇随意选择一个代表对象; 剩余的对象根据其代表对象的距离分配给最近的一个簇
- 然后反复地用非代表对象来替代代表对象, 以改进聚类的质量
- 聚类结果的质量用一个代价函数来估算, 该函数评估了对象与其参照对象之间的平均相异度

k -中心点聚类方法(续)

- 为了判定一个非代表对象 O_{random} 是否是当前一个代表对象 O_j 的好的替代, 对于每一个非代表对象 p , 考虑下面的四种情况:
 - 第一种情况: p 当前隶属于代表对象 O_j . 如果 O_j 被 O_{random} 所代替, 且 p 离 O_i 最近, $i \neq j$, 那么 p 被重新分配给 O_i
 - 第二种情况: p 当前隶属于代表对象 O_j . 如果 O_j 被 O_{random} 代替, 且 p 离 O_{random} 最近, 那么 p 被重新分配给 O_{random}
 - 第三种情况: p 当前隶属于 O_i , $i \neq j$. 如果 O_j 被 O_{random} 代替, 而 p 仍然离 O_i 最近, 那么对象的隶属不发生变化
 - 第四种情况: p 当前隶属于 O_i , $i \neq j$. 如果 O_j 被 O_{random} 代替, 且 p 离 O_{random} 最近, 那么 p 被重新分配给 O_{random}

k -中心点聚类方法(续)



重新分配给 O_i

2. 重新分配给 O_{random}

3. 不发生变化

4. 重新分配给 O_{random}

● 数据对象

+ 簇中心

— 替代前

..... 替代后

图8-3 k -中心点聚类代价函数的四种情况

k -中心点聚类方法(续)

■ 算法: k -中心点

(1) 随机选择 k 个对象作为初始的代表对象;

(2) **repeat**

(3) 指派每个剩余的对象给离它最近的代表对象所代表的簇;

(4) 随意地选择一个非代表对象 O_{random} ;

(5) 计算用 O_{random} 代替 O_j 的总代价 S ;

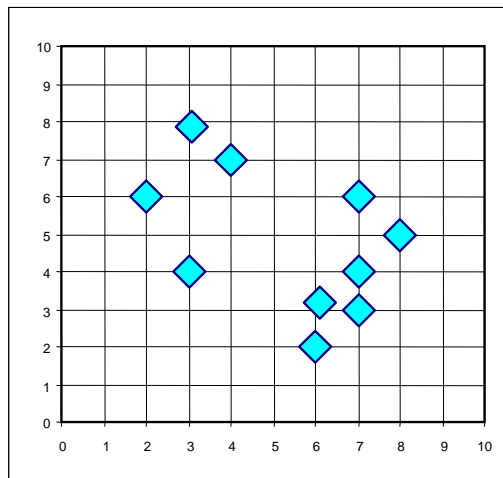
(6) 如果 $S < 0$, 则用 O_{random} 替换 O_j , 形成新的 k 个代表对象的集合;

(7) **until** 不发生变化

PAM

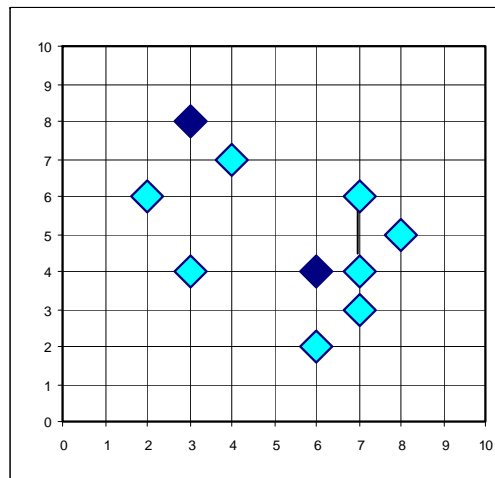
- **PAM (Partitioning Around Medoids) (Kaufman and Rousseeuw, 1987)**
 - 是最早提出的 k -中心点聚类算法
 - 基本思想:
 - 随机选择 k 个代表对象
 - 反复地试图找出更好的代表对象: 分析所有可能的对象对, 每个对中的一个对象被看作是代表对象, 而另一个不是. 对可能的各种组合, 估算聚类结果的质量

PAM(续)



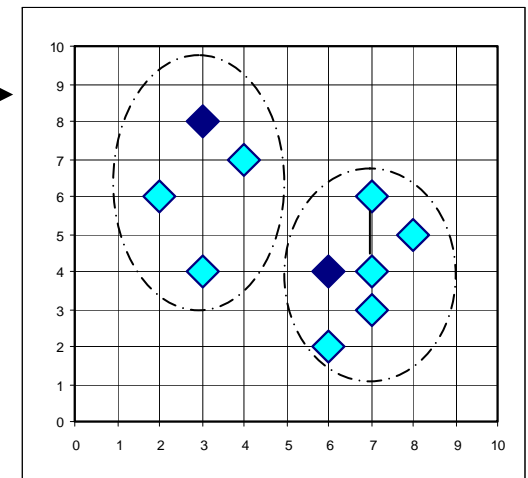
$K=2$

Arbitrary
choose k
object as
initial
medoids



Total Cost = 26

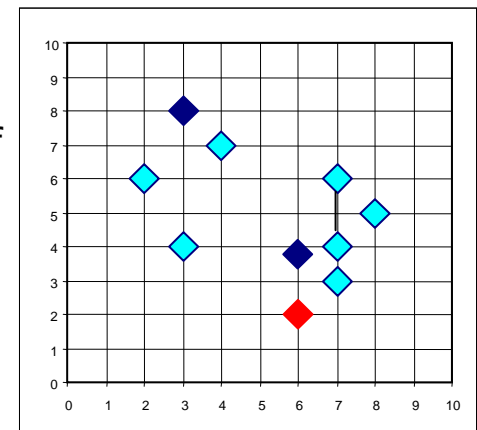
Assign
each
remainin
g object
to
nearest
medoids



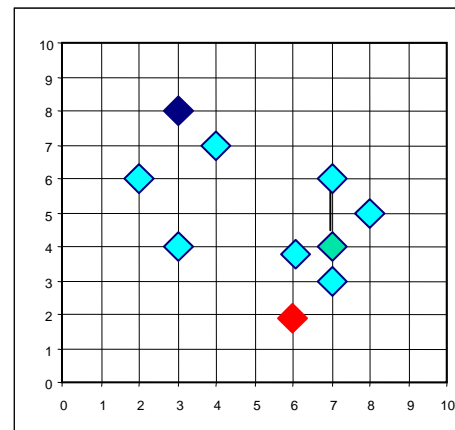
Total Cost = 20

Randomly select a
nonmedoid object, O_{random}

Compute
total cost of
swapping



Swapping O
and O_{random}
If quality is
improved.



Do loop
Until no
change

PAM(续)

- 当存在噪音和孤立点时, PAM 比 k -平均方法更健壮. 这是因为中心点不象平均值那么容易被极端数据影响
- PAM对于小数据集工作得很好, 但不能很好地用于大数据集
 - 每次迭代 $O(k(n-k)^2)$

其中 n 是数据对象数目, k 是聚类数

→ 基于抽样的方法,

CLARA(Clustering LARge Applications)

CLARA (Clustering Large Applications) (1990)

- *CLARA* (Kaufmann and Rousseeuw in 1990)
 - 不考虑整个数据集, 而是选择数据的一小部分作为样本
- 它从数据集中抽取多个样本集, 对每个样本集使用*PAM*, 并以最好的聚类作为输出
- 优点: 可以处理的数据集比 *PAM* 大
- 缺点:
 - 有效性依赖于样本集的大小
 - 基于样本的好的聚类并不一定是 整个数据集的好的聚类, 样本可能发生倾斜
 - 例如, O_i 是最佳的 k 个中心点之一, 但它不包含在样本中, *CLARA* 将找不到最佳聚类

CLARANS (“Randomized” CLARA) (1994)

- **CLARANS** (A Clustering Algorithm based on Randomized Search) (Ng and Han’94)
- **CLARANS**将采样技术和**PAM**结合起来
 - **CLARA**在搜索的每个阶段有一个固定的样本
 - **CLARANS**任何时候都不局限于固定样本,而是在搜索的每一步带一定随机性地抽取一个样本
- 聚类过程可以被描述为对一个图的搜索,图中的每个节点是一个潜在的解,也就是说 **k medoids**
 - 相邻节点: 代表的集合只有一个对象不同
- 在替换了一个代表对象后得到的聚类结果被称为当前聚类结果的邻居

CLARANS(续)

- 如果一个更好的邻居被发现, **CLARANS**移到该邻居节点, 处理过程重新开始, 否则当前的聚类达到了一个局部最优
- 如果找到了一个局部最优, **CLARANS**从随机选择的节点开始寻找新的局部最优
- 实验显示**CLARANS**比**PAM**和**CLARA**更有效
- **CLARANS**能够探测孤立点
- 聚焦技术和空间存取结构可以进一步改进它的性能 (**Ester et al.'95**)

第7章. 聚类分析

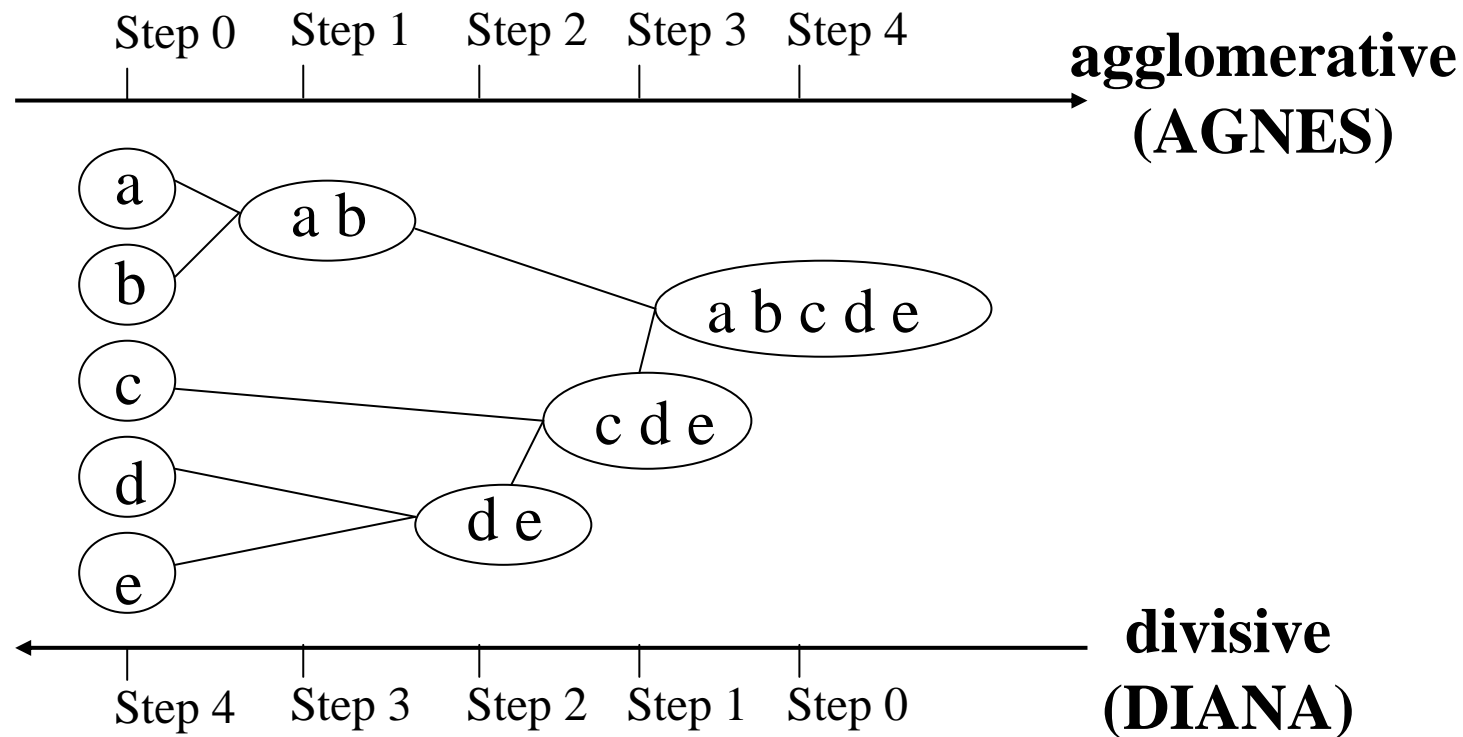
- 什么是聚类（**Clustering**）分析？
- 聚类分析中的数据类型
- 主要聚类方法分类
- 划分方法（**Partitioning Methods**）
- 层次方法（**Hierarchical Methods**）
- 基于密度的方法（**Density-Based Methods**）
- 基于网格的方法（**Grid-Based Methods**）
- 基于模型的聚类方法（**Model-Based Clustering Methods**）
- 孤立点分析（**Outlier Analysis**）
- 小结

层次方法

- 层次的聚类方法将数据对象组成一棵聚类的树
- 根据层次分解是自底向上, 还是自顶向下形成, 层次的聚类方法可以进一步分为凝聚的(**agglomerative**)和分裂的(**divisive**)层次聚类
- 纯粹的层次聚类方法的聚类质量受限于如下特点: 一旦一个合并或分裂被执行, 就不能修正
- 最近的研究集中于凝聚层次聚类和迭代重定位方法的集成
- 使用距离矩阵作为聚类标准. 该方法不需要输入聚类数目 k , 但需要终止条件

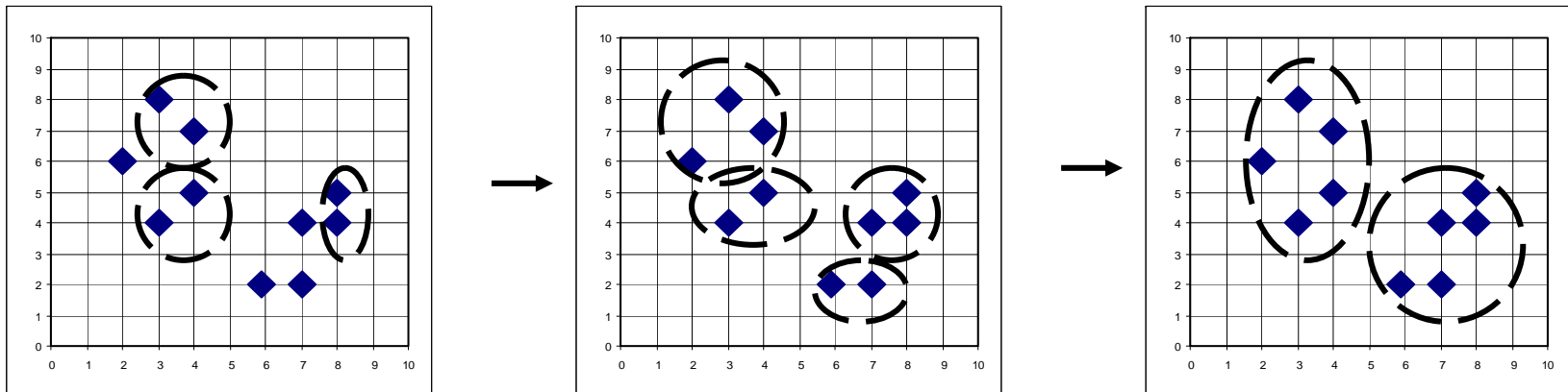
层次方法(续)

- 凝聚的(agglomerative)和分裂的(divisive)层次聚类图示



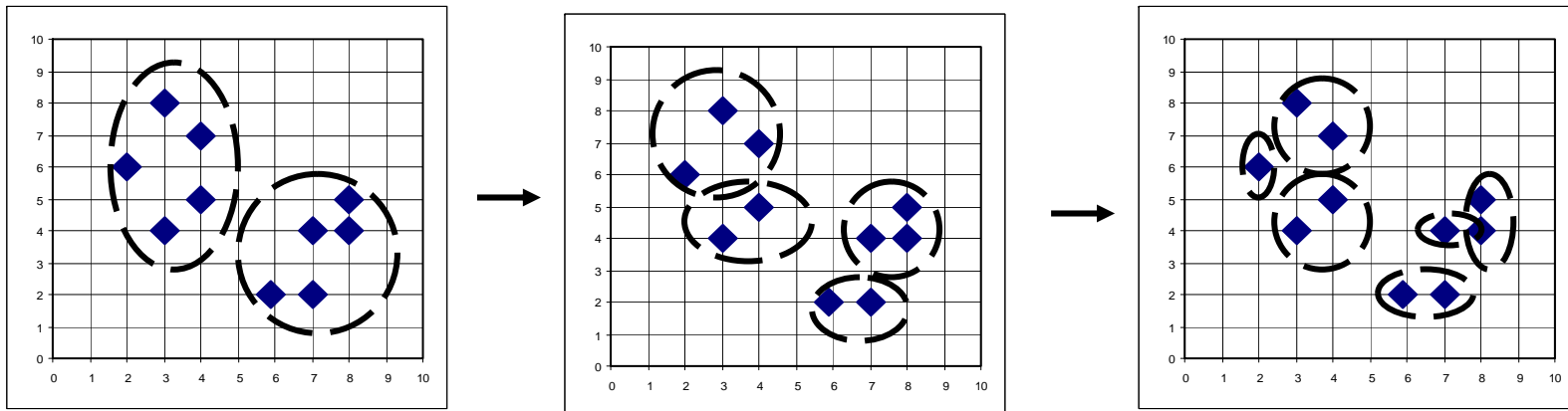
AGNES (Agglomerative Nesting)

- 由 **Kaufmann**和**Rousseeuw**提出(1990)
- 已在一些统计分析软件包中实现 . 如 **Splus**
- 使用单链接(**Single-Link**)方法和相异度矩阵
- 合并具有最小相异度的节点
- 以非递减的方式继续
- 最终所有的节点属于同一个簇



DIANA (Divisive Analysis)

- 由 **Kaufmann**和**Rousseeuw**提出 (1990)
- 已在一些统计分析软件包中实现 . 如 **Splus**
- 是 **AGNES**的逆
- 最终每个节点自己形成一个簇



层次方法(续)

- 四个广泛采用的簇间距离度量方法

- 最小距离: $d_{min}(C_i, C_j) = \min_{p \in C_i, p' \in C_j} |p - p'|$
- 最大距离: $d_{max}(C_i, C_j) = \max_{p \in C_i, p' \in C_j} |p - p'|$
- 平均值的距离: $d_{mean}(C_i, C_j) = |m_i - m_j|$
- 平均距离: $d_{avg}(C_i, C_j) = \sum_{p \in C_i} \sum_{p' \in C_j} |p - p'| / n_i n_j$

其中, $|p - p'|$ 是两个对象 p 和 p' 之间的距离

m_i 是簇 C_i 的平均值, n_i 是簇 C_i 中对象的数目

层次方法(续)

- 层次聚类的主要缺点

- 不具有很好的可伸缩性: 时间复杂性至少是 $O(n^2)$, 其中 n 对象总数
- 合并或分裂的决定需要检查和估算大量的对象或簇
- 不能撤消已做的处理, 聚类之间不能交换对象. 如果某一步没有很好地选择合并或分裂的决定, 可能会导致低质量的聚类结果

层次方法(续)

- 改进层次方法的聚类质量的方法：将层次聚类和其他的聚类技术进行集成, 形成多阶段聚类
 - **BIRCH (1996)**: 使用 **CF-tree**对对象进行层次划分, 然后采用其他的聚类算法对聚类结果进行求精
 - **ROCK1999**: 基于簇间的互联性进行合并
 - **CHAMELEON (1999)**: 使用动态模型进行层次聚类
 - **CURE (1998)**: 采用固定数目的代表对象来表示每个簇, 然后依据一个指定的收缩因子向着聚类中心对它们进行收缩

BIRCH (1996)

- **Birch (Balanced Iterative Reducing and Clustering using Hierarchies):** 利用层次方法的平衡迭代归约和聚类由Zhang, Ramakrishnan和Livny 提出 (SIGMOD'96)
- 两个重要概念
 - 聚类特征(Clustering Feature, CF)
 - 聚类特征树(Clustering Feature Tree, CF树)
- 聚类特征
 - 聚类特征(CF)是一个三元组, 给出对象子类的信息的汇总描述
 - 设某个子类中有N个d维的点或对象 $\{o_i\}$, 则该子类的CF定义如下
$$CF = (N, L\vec{S}, SS)$$

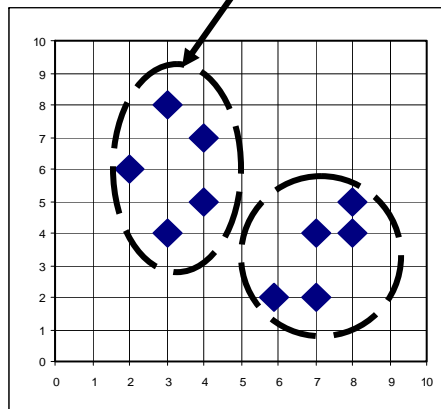
聚类特征

Clustering Feature: $CF = (N, \vec{LS}, SS)$

N : 数据点数目

LS : $\sum_{i=1}^N \vec{X}_i$

SS : $\sum_{i=1}^N \vec{X}_i^2$



$CF = (5, (16,30), (54,190))$

(3,4)

(2,6)

(4,5)

(4,7)

(3,8)

BIRCH的CF树

- 聚类特征

- 从统计学的观点来看，聚类特征是对给定子类统计汇总：子聚类的0阶, 1阶和 2阶矩(**moments**)
- 记录了计算聚类 and 有效利用存储的关键度量, 并有效地利用了存储, 因为它汇总了关于子类的信息, 而不是存储所有的对象

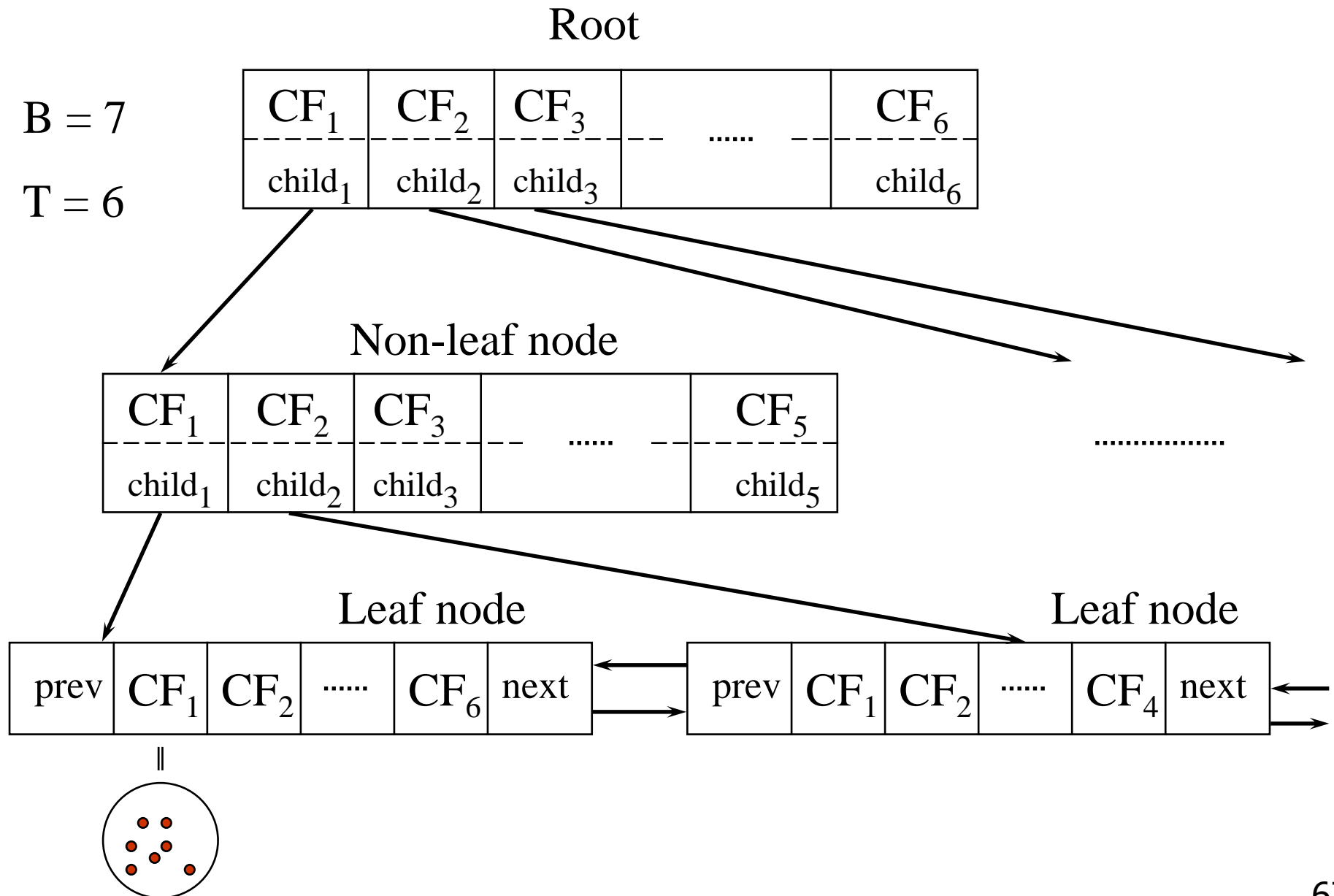
- CF 树是高度平衡的树，它存储了层次聚类的聚类特征

- 树中的非叶节点有后代或“孩子”
- 非叶节点存储了其孩子的CF的总和，即汇总了关于其孩子的聚类信息

- CF树有两个参数 ----影响CF树的大小

- 分支因子**B**: 定义非树叶节点的孩子的最大个数
- 阈值**T**: 给出了存储在树的叶子节点中的子类的最大直径

CF Tree



BIRCH (续)

- BIRCH增量地构造一棵 CF 树(Clustering Feature Tree), CF 树是一个

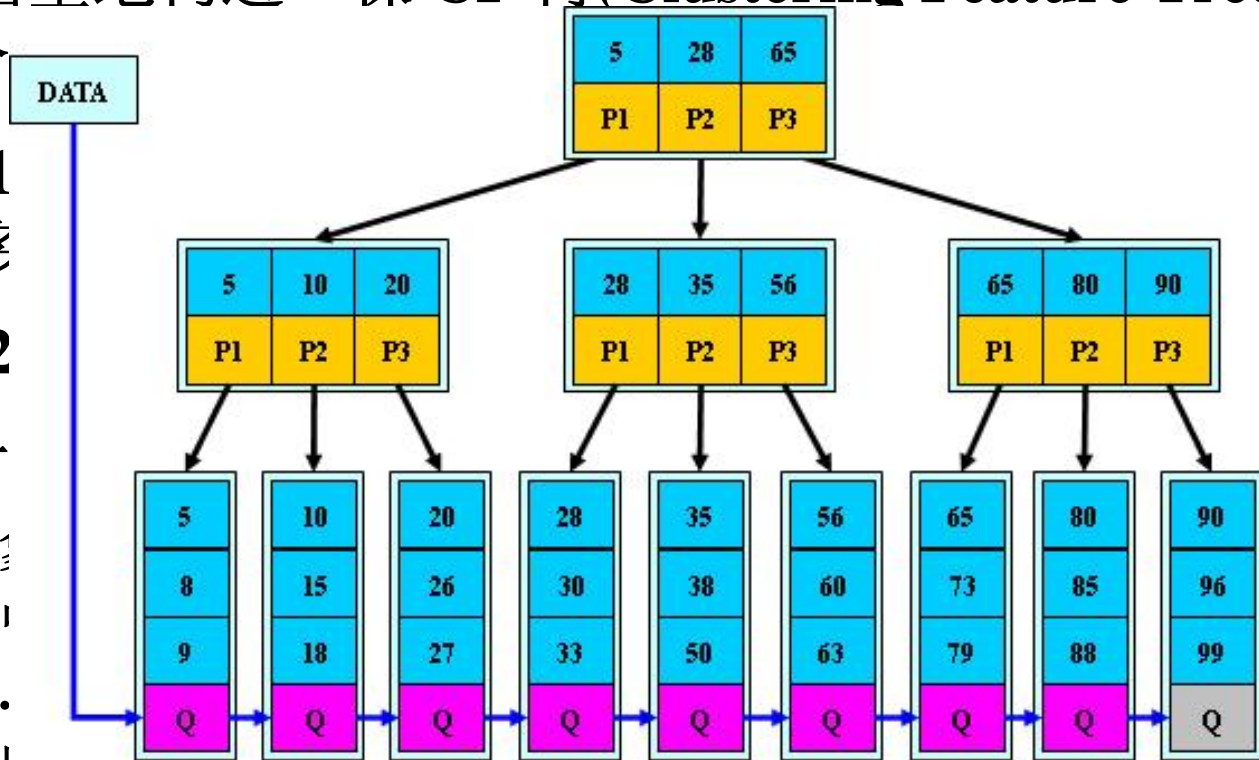
- 阶段 1

- 阶段 2

- 在阶段一

- 一个对子节点被分裂.

- 通过修改阈值, CF树的大小可以改变



树(数

在叶
他节点)
以于B+

BIRCH (续)

- 重建过程从旧树的叶子节点建造一个新树。这样，重建树的过程不需要重读所有的对象 ---- 建树只需读一次数据
- 在阶段二被采用任何聚类算法，例如典型的划分方法
- **BIRCH**的性能
 - 支持增量聚类
 - 线性可伸缩性: 计算复杂性 $O(n)$, 单遍扫描, 附加的扫描可以改善聚类质量
 - 较好的聚类质量
- 缺点
 - 只能处理数值数据
 - 对数据的输入次序敏感
 - **Cf**树结点不总是对应于[用户考虑的]自然簇(参数**B**和**T**)
 - 簇非球形时效果不好(使用半径/直径控制簇边界)

CURE(1998)

- **CURE (Clustering Using REpresentatives)** : 由 **Guha, Rastogi** 和 **Shim**提出(1998)
- 绝大多数聚类算法或者擅长处理球形和相似大小的聚类, 或者在存在孤立点时变得比较脆弱
- **CURE**解决了偏好球形的问题, 在处理孤立点上也更加健壮
- **CURE**采用了一种新的层次聚类算法
 - 选择基于质心和基于代表对象方法之间的中间策略. 它不用单个质心或对象来代表一个簇, 而是选择了数据空间中固定数目的具有代表性的点
 - 首先选择簇中分散的对象, 然后根据一个特定的收缩因子向簇中心“收缩”

CURE(续)

- 每个簇有多于一个的代表点使得**CURE**可以适应非球形的任意形状的聚类
- 簇的收缩或凝聚可以有助于控制孤立点的影响
- **CURE**的优点
 - **CURE**对孤立点的处理更加健壮
 - 能够识别非球形和大小变化较大的簇
 - 对于大规模数据库,它 also 具有良好的伸缩性,而且没有牺牲聚类质量
- 针对大型数据库,**CURE**采用了随机取样和划分两种方法的组合
 - 首先划分一个随机样本,每个划分被部分聚类
 - 然后对这些结果簇聚类,产生希望的结果

Cure(续)

- CURE算法核心:

- 从源数据对象中抽取一个随机样本 S .
- 将样本 S 分割为 p 个划分, 每个的大小为 s/p
- 将每个划分局部地聚类成 s/pq 个簇
- 删除孤立点
 - 通过随机选样
 - 如果一个簇增长太慢, 就删除它.
- 对局部聚类进行聚类.
- 用相应的簇标签来标记数据

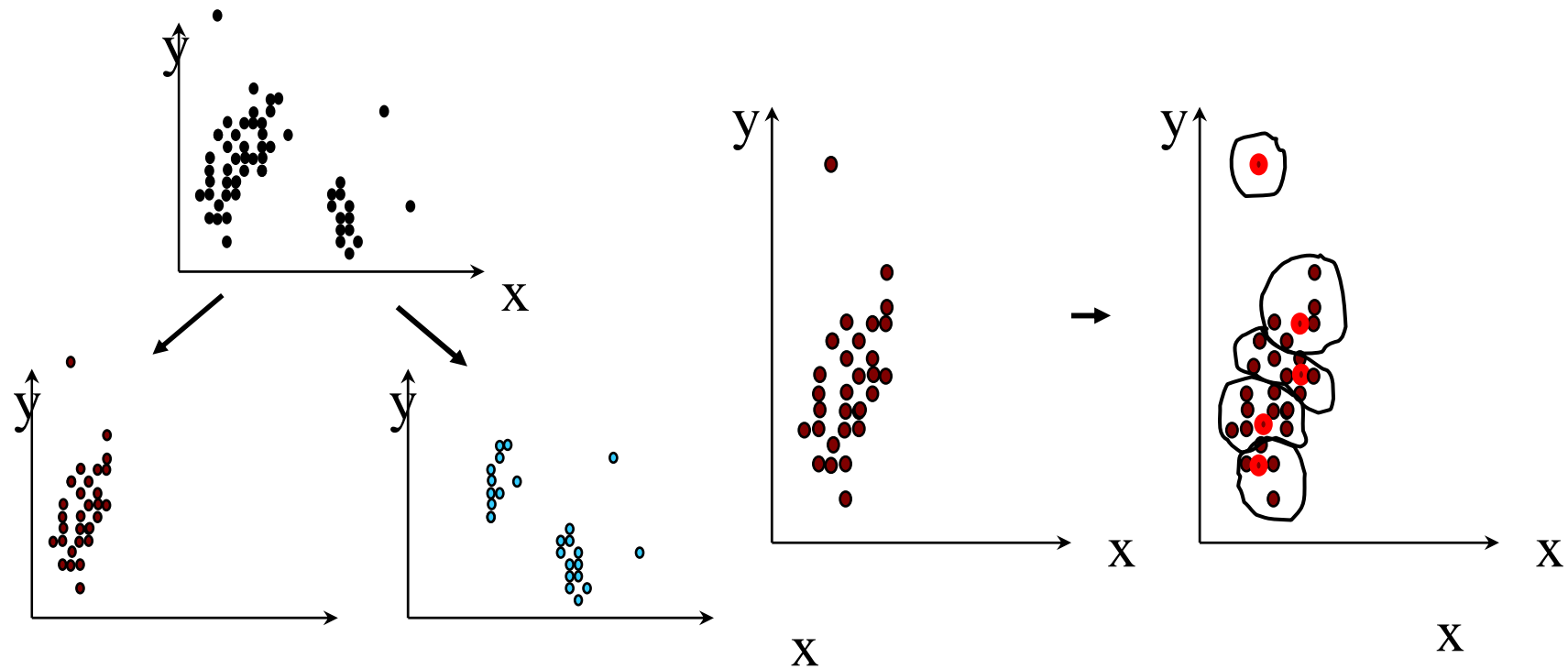
CURE: 例

■ $s = 50$

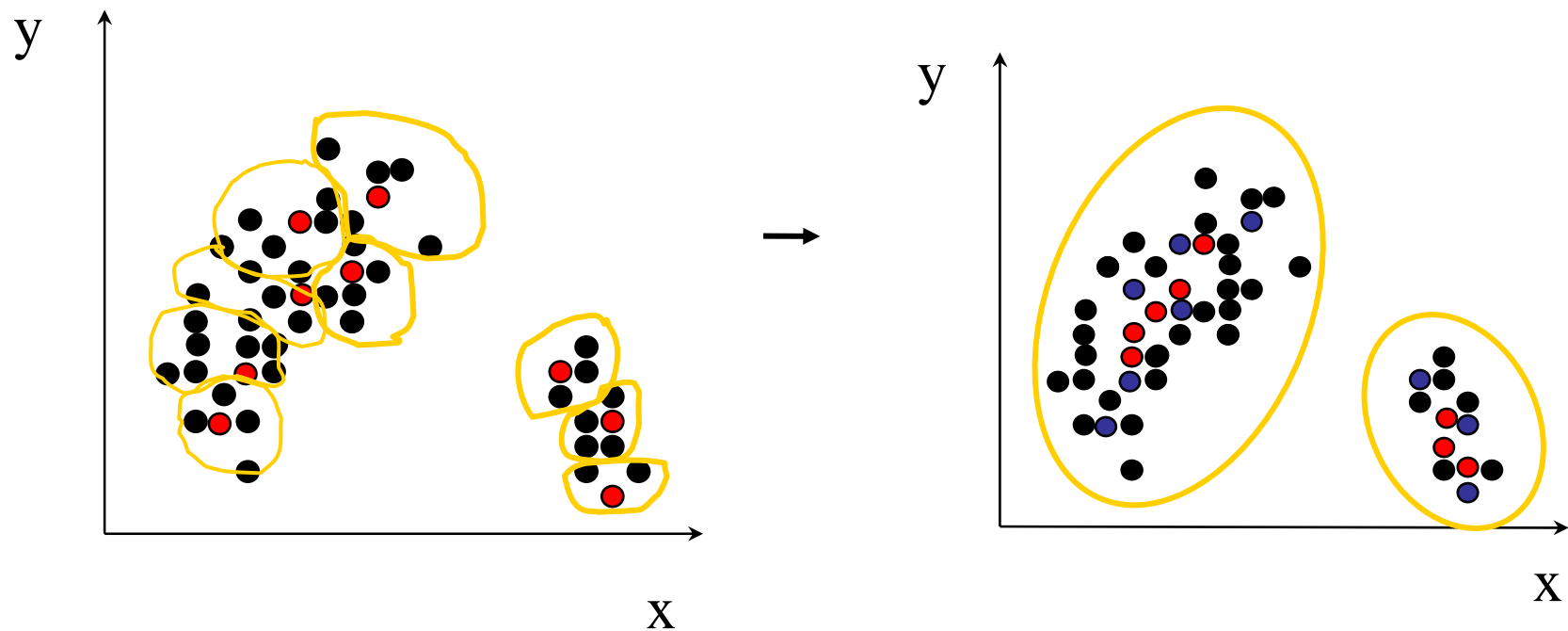
■ $p = 2$

■ $s/p = 25$

■ $s/pq = 5$



CURE: 例(续)



- 多个代表点向重心 以因子 α 移动, 进行收缩或凝聚
- 多个代表点描述了每个簇的形状

对分类数据聚类: ROCK

- **ROCK(RObust Clustering using linKs)** 由S. Guha, R. Rastogi, K. Shim提出 (ICDE'99).

- 使用**链接(link)**度量相似性/接近性

- 链接: 两个对象间共同的近邻的数目

- 不是基于距离的

- 计算复杂性: $O(n^2 + nm_m m_a + n^2 \log n)$

- 基本思想:

- 相似性函数:**Jaccard**系数

$$Sim(T_1, T_2) = \frac{|T_1 \cap T_2|}{|T_1 \cup T_2|}$$

设 $T_1 = \{1,2,3\}$, $T_2 = \{3,4,5\}$

$$Sim(T_1, T_2) = \frac{|\{3\}|}{|\{1,2,3,4,5\}|} = \frac{1}{5} = 0.2$$

Rock(续)

- 两个点 p_i 和 p_j 是近邻, 如果 $\text{sim}(p_i, p_j) \geq$ 用户指定阈值
- $\text{link}(p_i, p_j)$ 是两个点 p_i 和 p_j 共同的近邻的数目
- 两个簇 C_i 和 C_j 的互连性被定义为两个簇间交叉链 (**cross link**) 的数目

$$\sum_{p_q \in C_i, p_r \in C_j} \text{link}(p_q, p_r)$$

- **ROCK**首先根据相似度阈值和共享近邻的概念, 从给定的数据相似度矩阵构建一个稀疏的图, 然后在这个稀疏图上运行一个层次聚类算法

we define the *goodness measure* $g(C_i, C_j)$ for merging clusters C_i, C_j

$$g(C_i, C_j) = \frac{\text{link}[C_i, C_j]}{(n_i + n_j)^{1+2f(\theta)} - n_i^{1+2f(\theta)} - n_j^{1+2f(\theta)}}$$

goodness measure is maximum is the best pair of clusters to be merged at any given step.

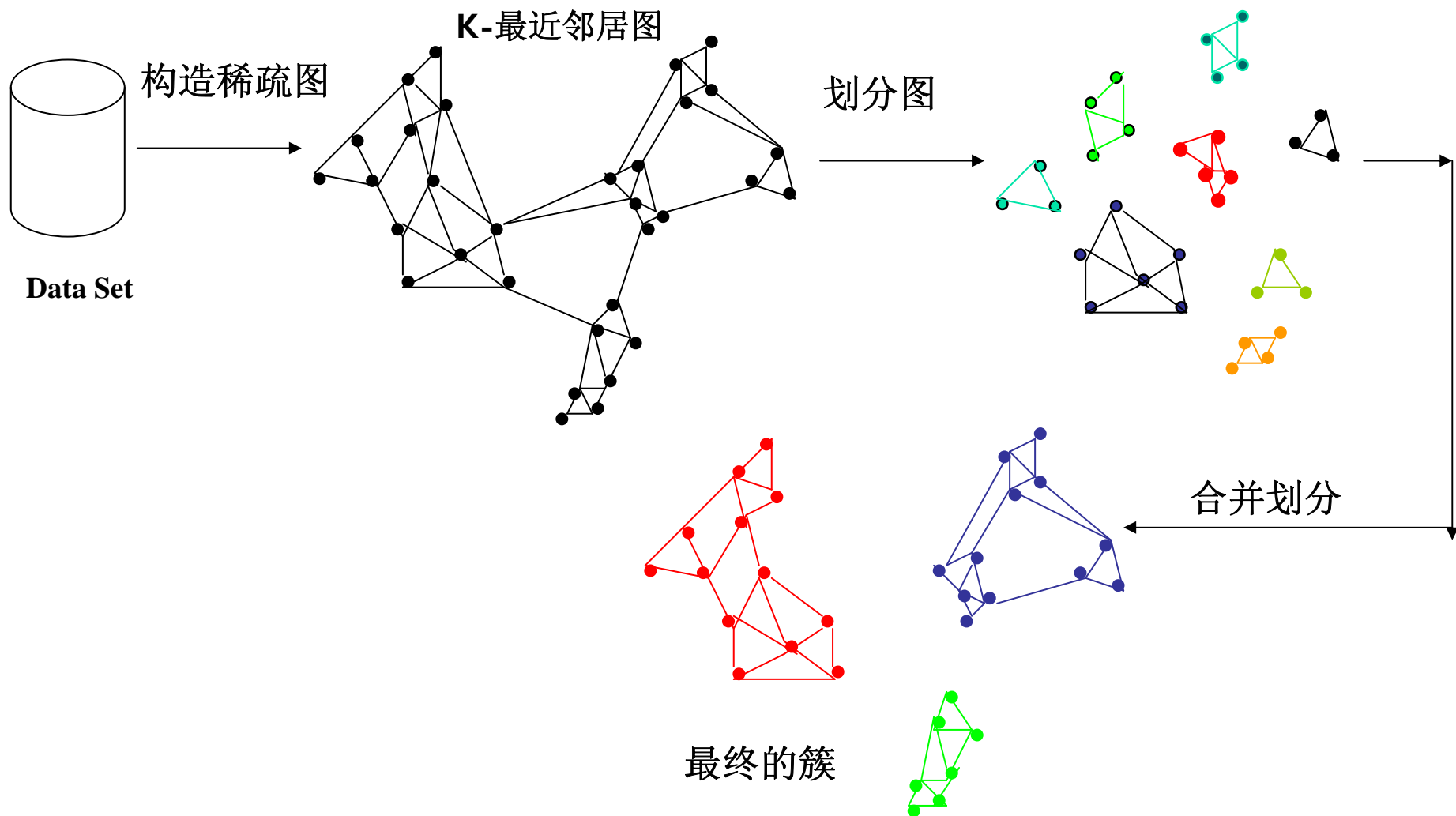
CHAMELEON

- **CHAMELEON** :一个利用动态模型的层次聚类算法 (Hierarchical clustering using dynamic modeling) 由G. Karypis, E.H. Han, and V. Kumar'99 提出
- 对**CURE**和**ROCK**缺点的观察:
 - **Cure**忽略了关于两个不同簇中对象的聚集互连性的信息
 - **Rock**强调对象间互连性, 却忽略了关于对象间近似度的信息
- **CHAMELEON**基于动态模型度量相似性
 - 如果两个簇间的互连性和近似度与簇内部对象间的互连性和近似度高度相关, 则合并这两个簇

CHAMELEON(续)

- 两阶段算法
 1. 使用图划分算法: 将数据对象聚类为大量相对较小的子类
逐步用图划分算法把k近邻图分成 相对较小de子簇, 最小化割边。
 2. 使用凝聚的层次聚类算法: 通过反复地合并子类来找到真正的结果簇
- 既考虑互连性, 又考虑簇间的近似度, 特别是簇内部的特征, 来确定最相似的子类.
- 这样, 它不依赖于静态的用户提供的模型, 能够自动地适应被合并的簇的内部特征
 - 割边最小化——簇c划分为两个子簇 C_i 和 C_j 时需要割断的边的加权和最小。
 - 割边用 $EC_{\{C_i, C_j\}}$ 表示, 评估 C_i 和 C_j 的簇间的绝对互联性。

CHAMELEON图示



CHAMELEON(续)

- **k-最近邻图 G_k** : 图中的每个点代表一个数据对象, 如果一个对象是另一个对象的**k**个最类似的对象之一, 在这两个点之间存在一条边
- **k-最近邻图 G_k 动态地捕捉邻域的概念**: 一个对象的邻域半径由对象所在区域的密度所决定
 - 在一个密集区域, 邻域的定义范围相对狭窄; 在一个稀疏区域, 它的定义范围相对较宽
- 区域的密度作为边的权重被记录下来
 - 一个密集区域的边趋向于比稀疏区域的边有更大的权重

CHAMELEON(续)

- Chameleon通过两个簇的相对互连性 $RI(C_i, C_j)$ 和相对接近度 $RC(C_i, C_j)$ 来决定簇间的相似度
 - $RI(C_i, C_j)$ 定义为 C_i 和 C_j 之间的绝对互联性关于两个簇的内部互连性的规范化

$$RI(C_i, C_j) = \frac{|EC_{\{C_i, C_j\}}|}{\frac{1}{2}(|EC_{C_i}| + |EC_{C_j}|)}$$

其中, $EC_{\{C_i, C_j\}}$ 是包含 C_i 和 C_j 的簇分裂为 C_i 和 C_j 的割边, EC_{C_i} (或 EC_{C_j}) 是它的最小截断等分线的大小 (即将图划分为两个大致相等的部分需要切断的边的加权和)

CHAMELEON(续)

- $RC(C_i, C_j)$ 定义为 C_i 和 C_j 之间的绝对接近度关于两个簇的内部接近度的规范化

$$RC(C_i, C_j) = \frac{\bar{S}_{EC_{\{C_i, C_j\}}}}{\frac{|C_i|}{|C_i| + |C_j|} \bar{S}_{EC_{C_i}} + \frac{|C_j|}{|C_i| + |C_j|} \bar{S}_{EC_{C_j}}}$$

其中, $\bar{S}_{EC(C_i, C_j)}$ 是连接 C_i 和 C_j 顶点的边的平均权重

$\bar{S}_{EC_{C_i}}$ 是 C_i 的最小二等分的边的平均权重

第7章. 聚类分析

- 什么是聚类（**Clustering**）分析？
- 聚类分析中的数据类型
- 主要聚类方法分类
- 划分方法（**Partitioning Methods**）
- 层次方法（**Hierarchical Methods**）
- 基于密度的方法（**Density-Based Methods**）
- 基于网格的方法（**Grid-Based Methods**）
- 基于模型的聚类方法（**Model-Based Clustering Methods**）
- 孤立点分析（**Outlier Analysis**）
- 小结

基于密度的方法

- 基于密度聚类 (**Density-Based Clustering**)
- 主要特点:
 - 发现任意形状的聚类
 - 处理噪音
 - 一遍扫描
 - 需要密度参数作为终止条件
- 一些有趣的研究:
 - **DBSCAN**: Ester, et al. (KDD'96)
 - **OPTICS**: Ankerst, et al (SIGMOD'99).
 - **DENCLUE**: Hinneburg & D. Keim (KDD'98)
 - **CLIQUE**: Agrawal, et al. (SIGMOD'98)

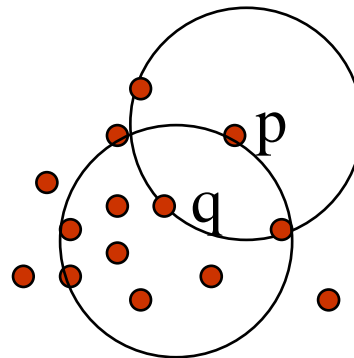
密度概念

- ϵ -邻域: 给定对象半径 ϵ 内的领域
- 核心对象 (Core object): 一个对象的 ϵ -邻域至少包含最小数目MinPts个对象
- 直接密度可达的(Directly density reachable, DDR): 给定对象集合D, 如果p是在q的 ϵ -邻域内, 而q是核心对象, 我们说对象p是从对象q直接密度可达的
- 密度可达的(density reachable): 存在一个从p到q的DDR对象链
- 密度相连的(density-connected): 如果对象集合D中存在一个对象o, 使得对象p和q是从o关于 ϵ 和MinPts密度可达的, 那么对象p和q是关于 ϵ 和MinPts密度相连的

基于密度的聚类: 背景I

- 两个参数:
 - Eps : 邻域的最大半径
 - $MinPts$: 在 Eps -邻域中的最少点数
 - $N_{Eps}(p)$: $\{q \text{ belongs to } D \mid dist(p,q) \leq Eps\}$
- 直接密度可达的: 点 p 关于 $Eps, MinPts$ 是从点 q 直接密度可达的, 如果
 - 1) p 属于 $N_{Eps}(q)$
 - 2) 核心点条件:

$$|N_{Eps}(q)| \geq MinPts$$



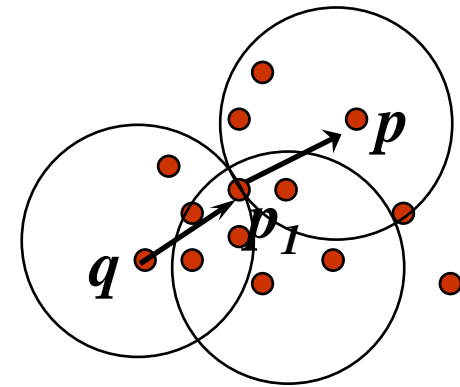
$MinPts = 5$

$Eps = 1 \text{ cm}$

基于密度的聚类: 背景II

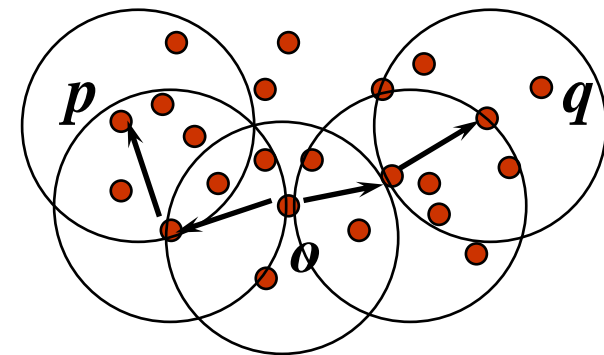
- 密度可达:

- 点 p 关于 $Eps, MinPts$ 是从 q 密度可达的, 如果 存在一个节点链 $p_1, \dots, p_n, p_1 = q, p_n = p$ 使得 p_{i+1} 是从 p_i 直接密度可达的



- 密度相连的:

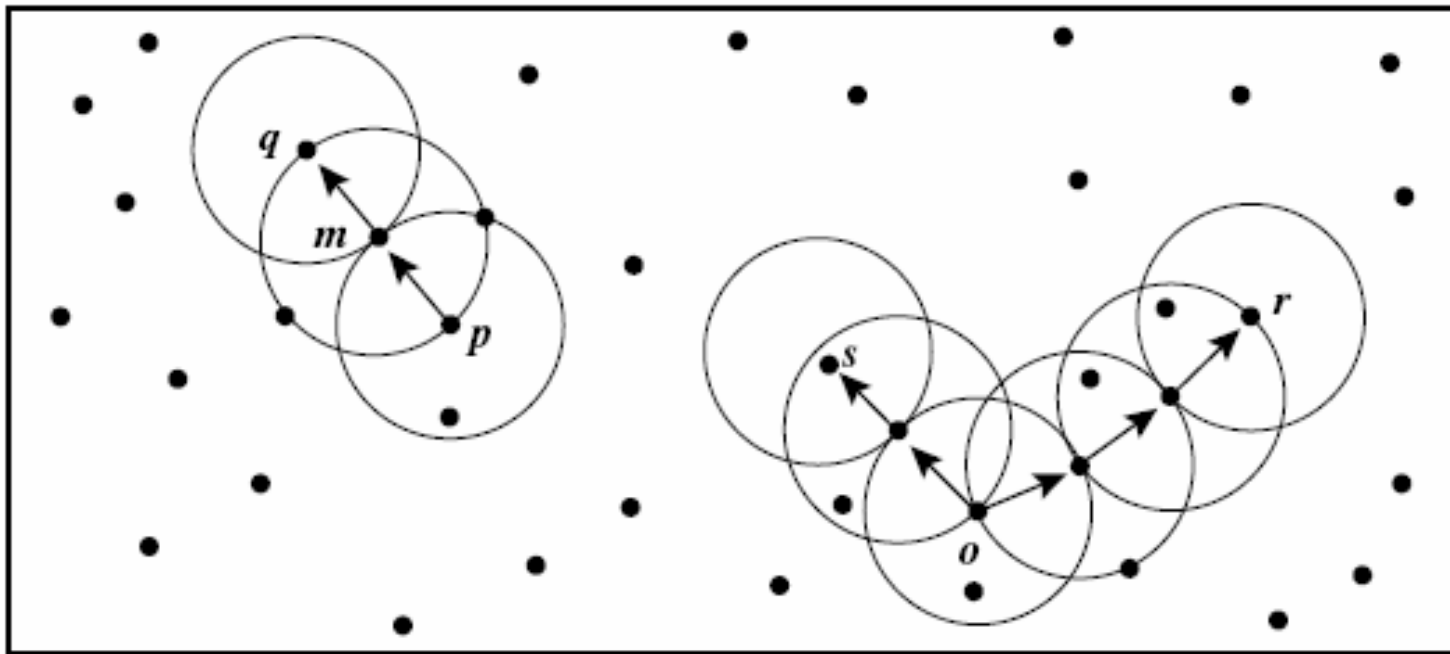
- 点 p 关于 $Eps, MinPts$ 与点 q 是密度相连的, 如果 存在点 o 使得, p 和 q 都是关于 $Eps, MinPts$ 是从 o 密度可达的



例子

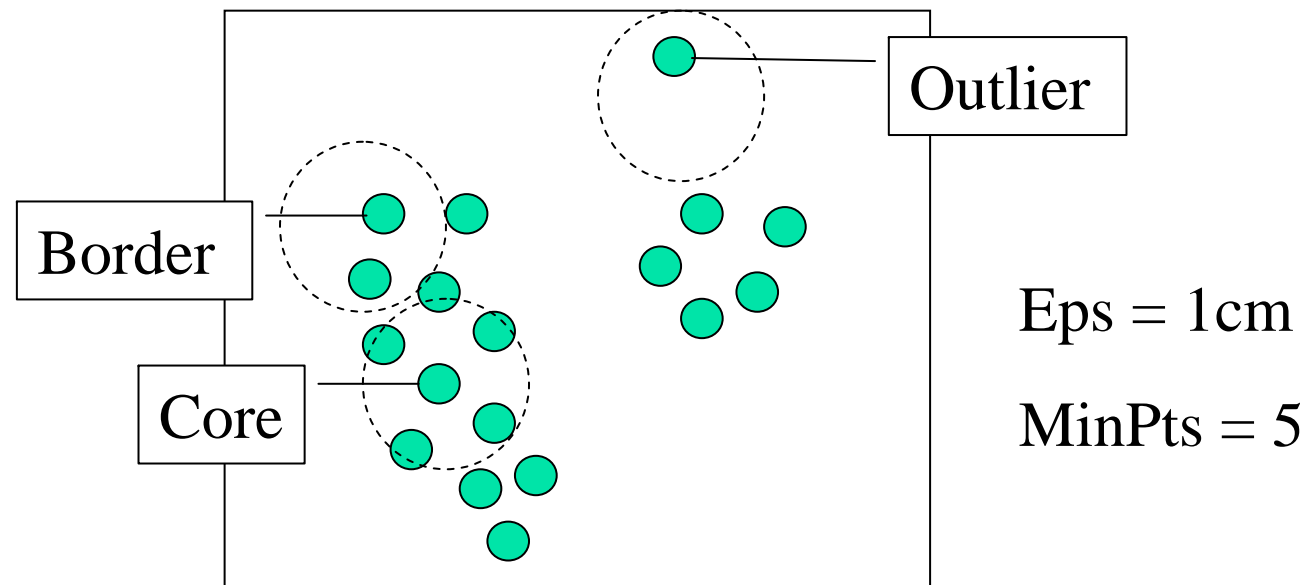
■ MinPts=3

- q 是从 p 密度可达； p 不是从 q 密度可达（ q 非核心）
- s 和 r 从 o 密度可达； o 从 r 密度可达；
- r, s, o 密度相连



DBSCAN(1996)

- **DBSCAN(Density Based Spatial Clustering of Applications with Noise)** 一个基于密度的聚类算法
- 可以在带有“噪音”的空间数据库中发现任意形状的聚类



DBSCAN(续)

■ 算法

- 任意选取一个点 p
- 得到所有从 p 关于 Eps 和 $MinPts$ 密度可达的点.
- 如果 p 是一个核心点, 则找到一个聚类.
- 如果 p 是一个边界点, 没有从 p 密度可达的点, **DBSCAN** 将访问数据库中的下一个点.
- 继续这一过程, 直到数据库中的所有点都被处理.

■ DBSCAN的复杂度

- 采用空间索引, 复杂度为 $O(n \log n)$, 否则为 $O(n^2)$

■ DBSCAN的缺点:

- 对用户定义的参数是敏感的, 参数难以确定(特别是对于高维数据), 设置的细微不同可能导致差别很大的聚类.

(数据倾斜分布) 全局密度参数不能刻画内在的聚类结构

OPTICS (1999)

- **OPTICS(Ordering Points To Identify the Clustering Structure)**
 - Ankerst, Breunig, Kriegel, 和 Sander 提出(SIGMOD'99)
 - 为自动和交互的聚类分析计算一个簇次序(cluster ordering).
 - 这个次序代表了数据的基于密度的聚类结构。它包含了信息, 等同于从一个广域的参数设置所获得的基于密度的聚类
 - 可用于自动和交互聚类分析, 包括发现内在聚类结构
 - 可以使用图形或可视化技术表示

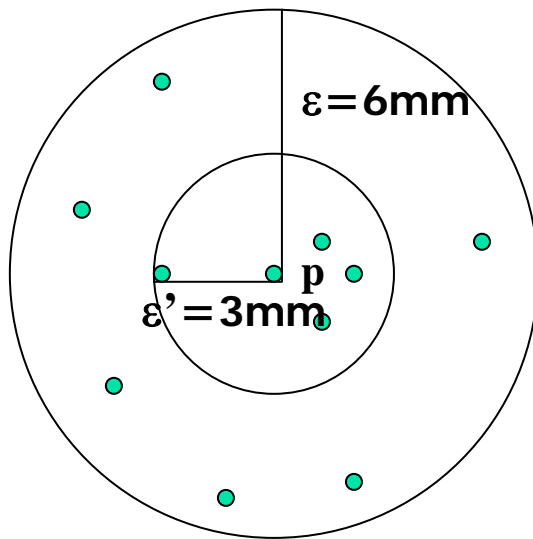
OPTICS(续)

- 考虑DBSCAN, 对于一个恒定的MinPts值, 关于高密度的(即较小的 ϵ 值)的聚类结果被完全包含在根据较低密度所获得的密度相连的集合中
- 扩展DBSCAN算法来同时处理一组距离参数值
- 为了同时构建不同的聚类, 应当以特定的顺序来处理对象. 优先选择最小的 ϵ 值密度可达的对象, 以便高密度的聚类能被首先完成
- 每个对象需要存储两个值
 - 对象p的**核心距离(core-distance)**是使得p成为核心对象的最小 ϵ 。如果p不是核心对象, p的核心距离没有定义
 - 对象q关于另一个对象p的**可达距离(reachability-distance)**是p的核心距离和p与q的欧几里得距离之间的较大值. 如果p不是一个核心对象, p和q之间的可达距离没有定义

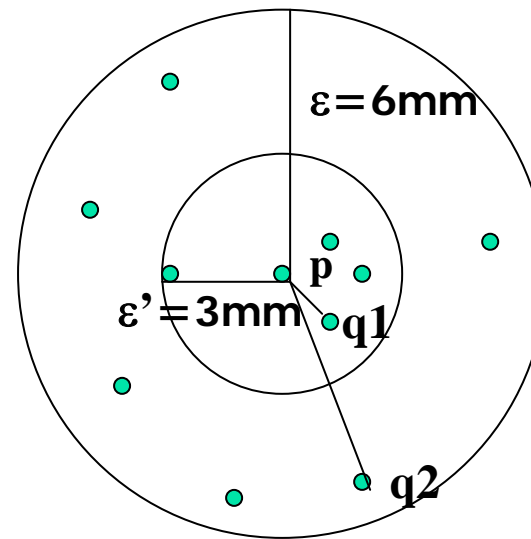
OPTICS(续)

■ 例: 设 $\epsilon=6(\text{mm})$, $\text{MinPts}=5$.

- p 的核心距离是 p 与第四个最近的数据对象之间的距离 ϵ' .
- q_1 关于 p 的可达距离是 p 的核心距离(即 $\epsilon'=3\text{mm}$), 因为它比从 p 到 q_1 的欧几里得距离要大.
- q_2 关于 p 的可达距离是从 p 到 q_2 的欧几里得距离, 它大于 p 的核心距离



p 的核心距离



可达距离 $(p, q1) = \epsilon' = 3\text{mm}$

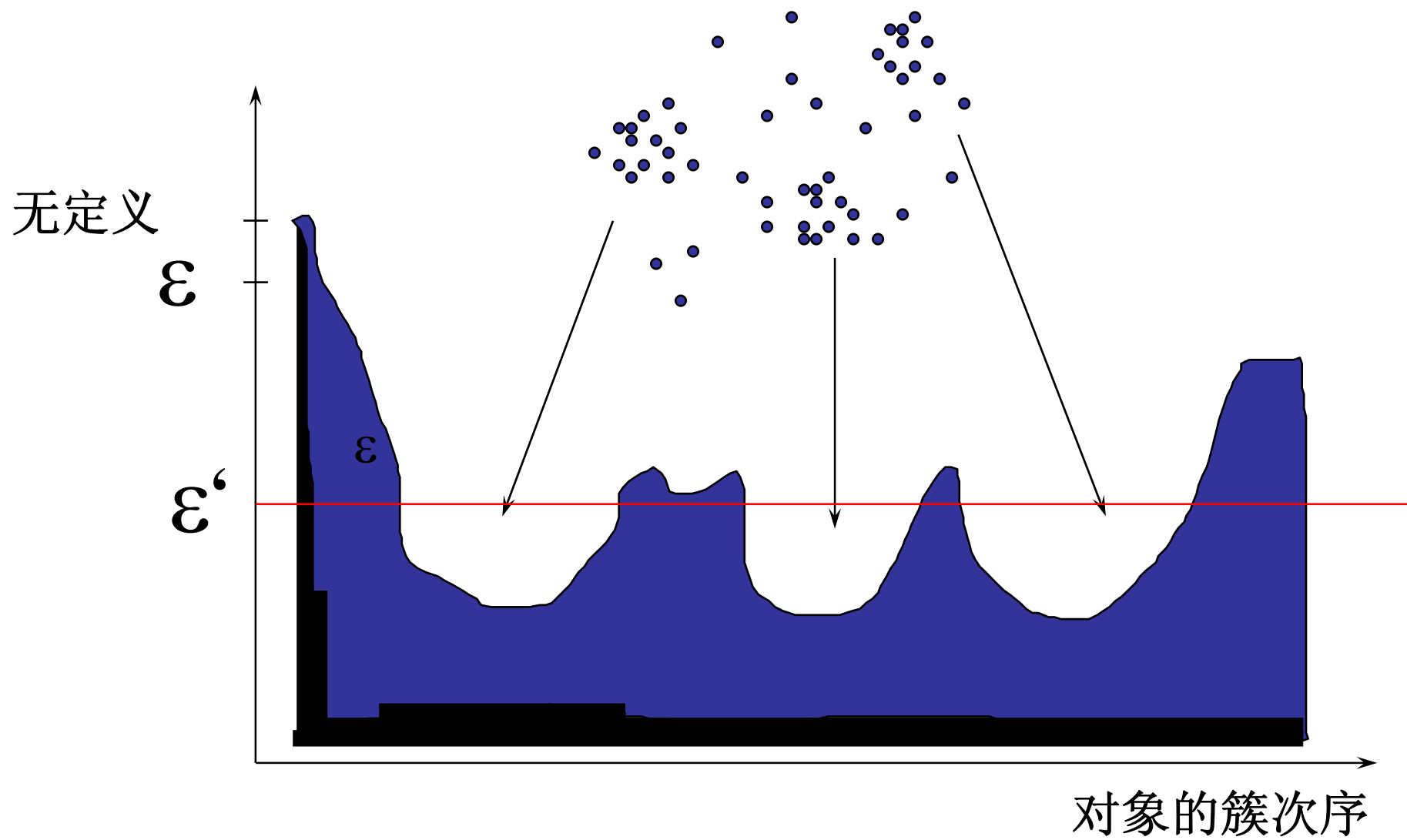
可达距离 $(p, q2) = d(p, q2)$

OPTICS(续)

Definition 7: (results of the OPTICS algorithm)

- Let DB be a database containing n points. The OPTICS algorithm generates an ordering of the points $o: \{1..n\} \rightarrow DB$ and corresponding reachability-values $r: \{1..n\} \rightarrow \mathbf{R}_{\geq 0}$.
 - 已经提出了一种算法, 基于**OPTICS**产生的次序信息来抽取聚类. 对于小于在生成该次序中采用的距离 ϵ 的任何距离 ϵ' , 为提取所有基于密度的聚类, 这些信息是足够的
- 一个数据集合的聚类次序可以被图形化地描述, 有助于理解
-
- 由于**OPTICS**算法与**DBSCAN**在结构上的等价性, 它具有和**DBSCAN**相同的时间复杂度, 即当使用空间索引时, 复杂度为 $O(n \log n)$

可达距离



DENCLUE(1998)

- **DENCLUE(DENsity-based CLUstEring)** 由**Hinneburg** 和 **Keim** (**KDD'98**)提出, 是基于密度分布函数的聚类方法
- 主要特点
 - 坚实的数学基础, 概括了其他的聚类方法, 包括基于划分的, 层次的, 及基于位置的方法
 - 适用于具有大量噪音的数据集
 - 可用于高维数据集任意形状的聚类, 它给出了简洁的数学描述
 - 明显快于现有算法 (比 **DBSCAN** 快 **45**倍)
 - 但是, 需要大量参数, 要求对密度参数 σ 和噪音阈值 ξ 进行仔细的选择

Denclue: 技术要点

- 使用栅格单元, 但只保存实际存放数据点的栅格单元信息, 并且在一个基于树的存取结构中管理这些单元.
- 影响函数(**Influence function**): 描述数据点在其邻域的影响.
- 数据空间的整体密度可以被模拟为所有数据点的影响函数的总和
- 聚类可以通过确定**密度吸引点 (density attractor)**来得到.
- 密度吸引点是全局密度函数的局部最大值.

DENCLUE(续)

- 设 \mathbf{x} 和 \mathbf{y} 是 d 维特征空间 F^d 中的对象. 数据对象 \mathbf{y} 对 \mathbf{x} 的**影响函数**是一个函数 $f^y_B: F^d \rightarrow R^+_0$, 它是根据一个基本的影响函数 f_B 来定义的

$$f^y_B(x) = f_B(\mathbf{x}, \mathbf{y})$$

- 原则上, 影响函数可以是一个任意的函数, 它由某个邻域内的两个对象之间的距离来决定
- 例如欧几里得距离函数, 用来计算一个方波影响函数(square wave influence function):

$$f_{Square}(x, y) = \begin{cases} 0 & \text{如果 } d(x, y) > \sigma \\ 1 & \text{其它} \end{cases}$$

DENCLUE(续)

- 高斯影响函数

$$f_{Gauss}(x, y) = e^{-\frac{d(x, y)^2}{2\sigma^2}}$$

- 一个对象 $x \in F^d$ 的密度函数被定义为所有数据点的影响函数的和. 给定 n 个对象, $D = \{x_1, \dots, x_n\} \subset F^d$, 在 x 上的密度函数定义如下

$$f_B^D(x) = \sum_{i=1}^n f_B^{x_i}(x)$$

DENCLUE(续)

- 例如, 根据高斯影响函数得出的密度函数是

$$f_{Gauss}^D(x) = \sum_{i=1}^N e^{-\frac{d(x, x_i)^2}{2\sigma^2}}$$

- 根据密度函数, 我们能够定义该函数的梯度和密度吸引点(全局密度函数的局部最大)
- 一个点 x 是被一个**密度吸引点** x^* 密度吸引的, 如果存在一组点 $x_0, x_1, \dots, x_k, x_0=x, x_k=x^*$, 对 $0 < i < k$, x_{i-1} 的梯度是在 x_i 的方向上
- 对一个连续的, 可微的影响函数, 用梯度指导的爬山算法能用来计算一组数据点的密度吸引点

密度吸引点

Def. 2 (Gradient)

The gradient of a function $f_B^D(x)$ is defined as

$$\nabla f_B^D(x) = \sum_{i=1}^N (x_i - x) \cdot f_B^{x_i}(x).$$

In case of the Gaussian influence function, the gradient is defined as:

$$\nabla f_{Gauss}^D(x) = \sum_{i=1}^N (x_i - x) \cdot e^{-\frac{d(x, x_i)^2}{2\sigma^2}}.$$

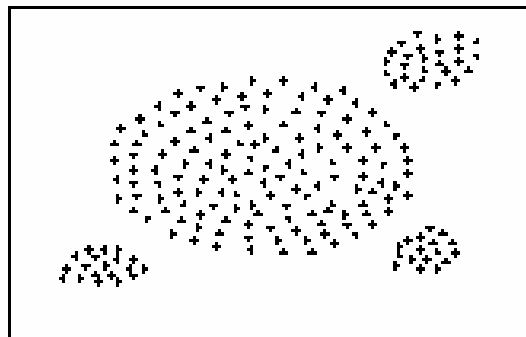
Def. 3 (Density-Attractor)

A point $x^* \in F^d$ is called a *density-attractor* for a given influence function, iff x^* is a local maximum of the density-function f_B^D .

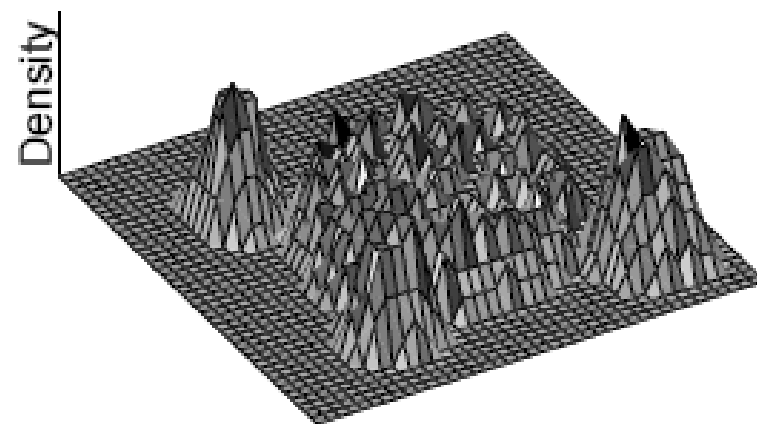
A point $x \in F^d$ is *density-attracted* to a density-attractor x^* , iff $\exists k \in N : d(x^k, x^*) \leq \epsilon$ with

$$x^0 = x, \quad x^i = x^{i-1} + \delta \cdot \frac{\nabla f_B^D(x^{i-1})}{\|\nabla f_B^D(x^{i-1})\|}.$$

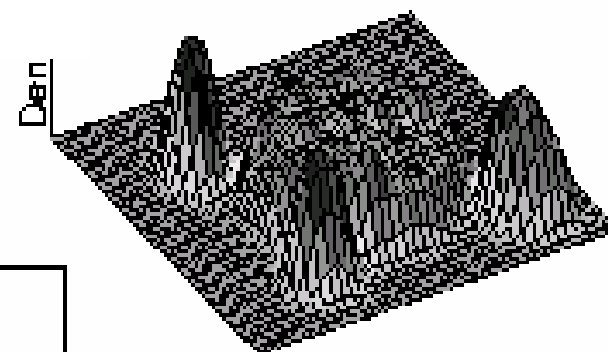
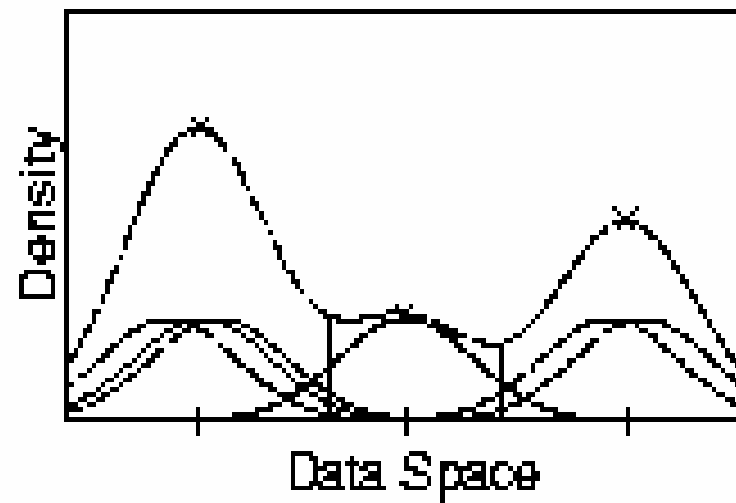
密度吸引点



(a) Data Set



(b) Square Wave



(c) Gaussian

中心定义的簇和任意形状的簇

- 密度吸引点 x^* 的**中心定义的簇** (center-defined cluster) 是一个被 x^* 密度吸引的子集 C , 在 x^* 的密度函数不小于一个阈值 ξ ; 否则(即如果它的密度函数值小于 ξ), 它被认为是孤立点
- 一个**任意形状的簇** (arbitrary-shape cluster) 是子集 C 的集合, 每一个是各自密度吸引子密度吸引的, 有不小于阈值 ξ 的密度函数值, 从每个区域到另一个都存在一条路径 P , 该路径上每个点的密度函数值都不小于 ξ

中心定义的簇和任意形状的簇

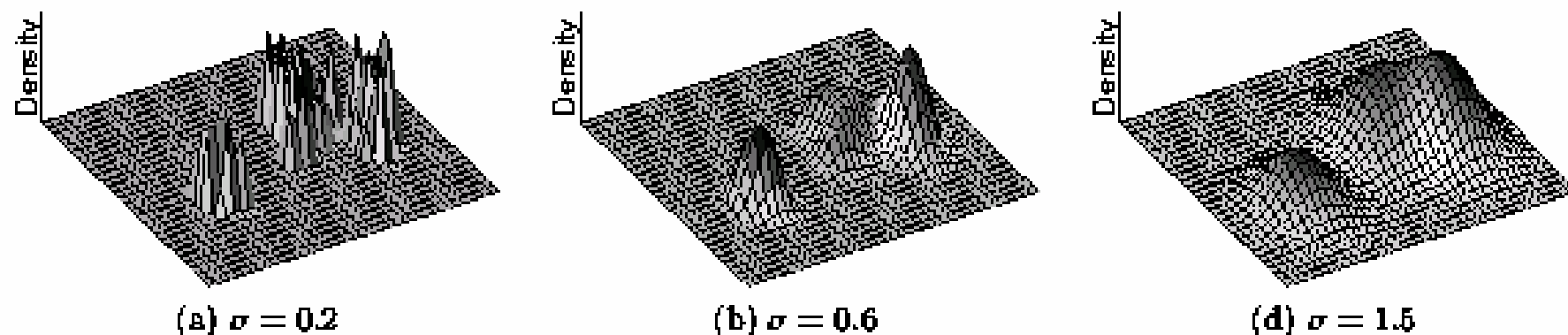


Figure 3: Example of Center-Defined Clusters for different σ

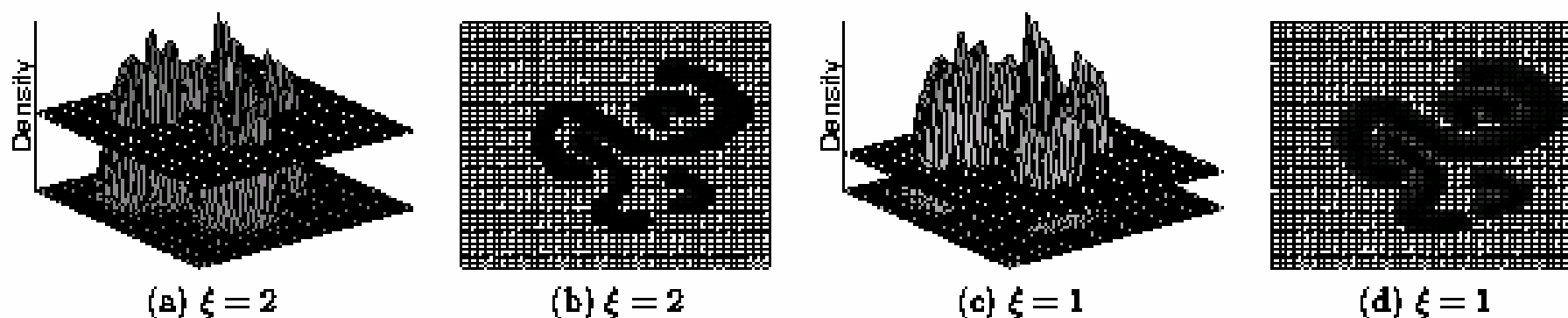


Figure 4: Example of Arbitrary-Shape Clusters for different ξ

第8章. 聚类分析

- 什么是聚类（**Clustering**）分析？
- 聚类分析中的数据类型
- 主要聚类方法分类
- 划分方法（**Partitioning Methods**）
- 层次方法（**Hierarchical Methods**）
- 基于密度的方法（**Density-Based Methods**）
- 基于网格的方法（**Grid-Based Methods**）
- 基于模型的聚类方法（**Model-Based Clustering Methods**）
- 孤立点分析（**Outlier Analysis**）
- 小结

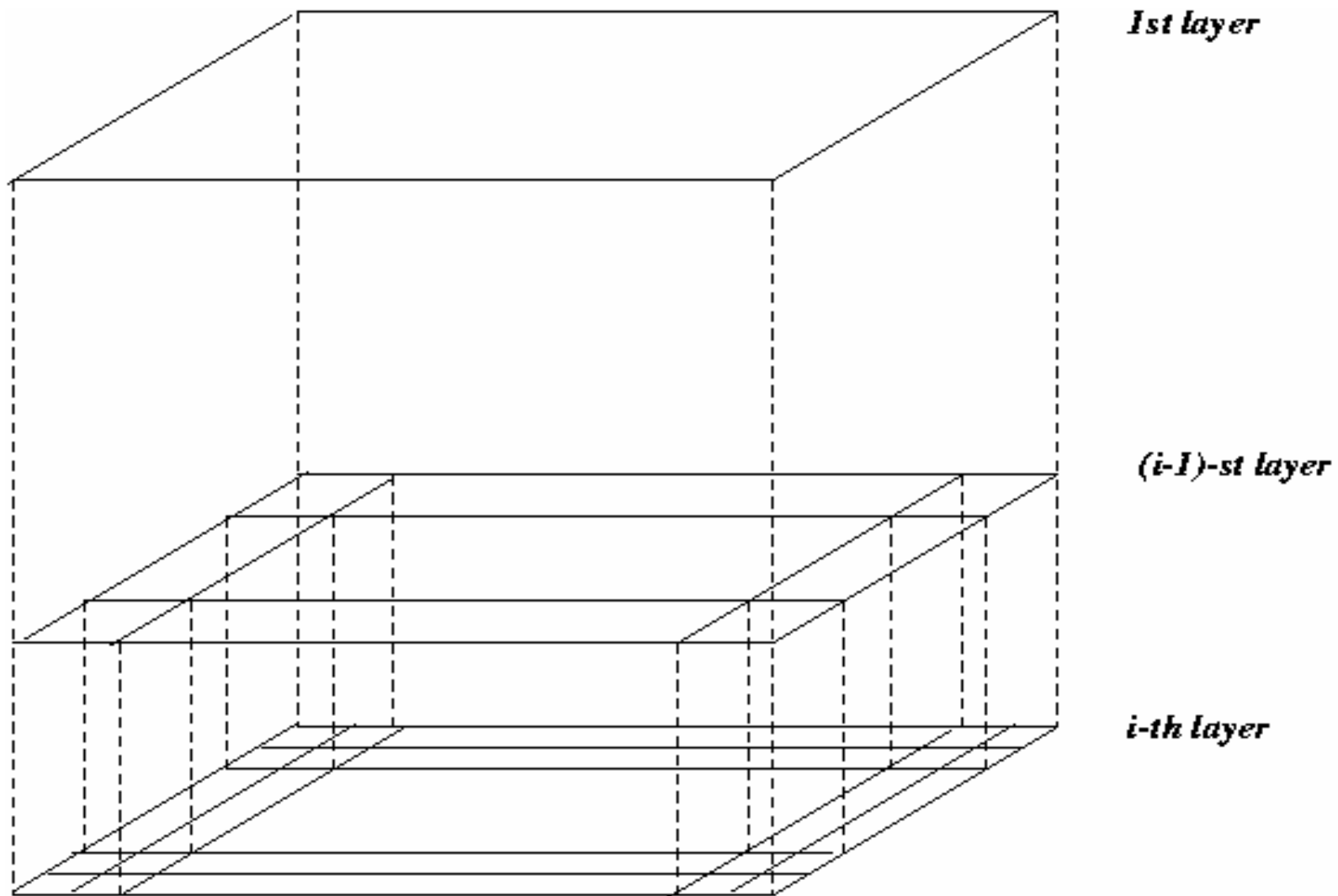
基于网格的聚类方法

- 使用多分辨率的网格数据结构
- 一些有趣的方法
 - **STING** (a STatistical INformation Grid approach)
由Wang, Yang 和 Muntz 提出(1997)
 - **WaveCluster** 由 Sheikholeslami, Chatterjee, 和 Zhang 提出(VLDB'98)
 - 采用小波方法的多分辨率的聚类方法
 - **CLIQUE**: Agrawal, et al. 提出(SIGMOD'98)

STING: 统计信息网格

- **STING(STatistical INformation Grid)**是一个基于网格的多分辨率聚类技术, 由**Wang, Yang** 和 **Muntz** 提出 (**VLDB'97**)
- 空间区域划分为矩形单元
- 多个级别的矩形单元, 对应不同级别的分辨率. 这些单元形成了一个层次结构: 每个高层单元被划分为多个低一层的单元
- 预先计算和存储关于每个网格单元属性的统计信息 (如平均值, 最大值, 和最小值), 用于回答查询

STING(续)



STING(续)

- 高层单元的统计参数可以很容易地从低层单元的计算得到. 这些统计参数包括:
 - 属性无关的参数**count**; 属性相关的参数**m**(平均值), **s**(标准偏差), **min**(最小值), **max**(最大值)
 - 该单元中属性值遵循的分布类型: 正态的, 一致的, 指数的, 无(分布未知)
- 分布的值可以由用户指定, 也可以通过假设检验(如 χ^2 检验)来获得
- 最底层单元的参数**count**, **m**, **s**, **min**, 和**max**直接进行计算

STING(续)

- 使用自顶向下的方法回答空间数据查询
 - 在层次结构选定一层作为查询处理的开始点——通常, 该层包含少量的单元
 - 对当前层次的每个单元, 计算置信度区间(或者估算其概率), 用以反映该单元与给定查询的关联程度
 - 删除不相关的单元, 进一步处理不考虑它们
 - 结束当前层的考查后, 就处理下一层
 - 重复这一过程, 直到最低层

STING(续)

■ 优点:

- 基于网格的计算是独立于查询的: 存储在每个单元中的统计信息是不依赖于查询的汇总信息
- 网格结构有利于并行处理和增量更新
- 效率很高: **STING**扫描数据库一次来计算单元的统计信息, 因此产生聚类的时间复杂度是 $O(n)$, 其中, n 是对象的数目

层次结构建立后, 查询处理时间是 $O(K)$, K 是最底层网格单元的数目, 通常远远小于 n

■ 缺点:

- **所有的聚类边界或者是水平的, 或者是坚直的, 没有斜的分界线.** 尽管该技术有快速的处理速度, 但可能降低簇的质量和精确性

WaveCluster (1998)

- 由Sheikholeslami, Chatterjee, 和Zhang (VLDB'98) 提出
- 采用小波变换聚类: 是一种多分辨率的聚类算法, 对特征空间采用小波变换(wavelet transform)
 - 小波变换是一种信号处理技术, 它将信号压缩到不同频率的子波段.
- 既是基于网格的方法, 又是基于密度的方法 **grid-based and density-based**
- 输入参数:
 - 每维单元的数目
 - 小波和应用小波变换的次数

WaveCluster (1998)

- 如何使用小波变换找出聚类
 - 首先通过在数据空间上强加一个多维网格结构来汇总数据
 - 用 n -维向量空间表示这些多维空间数据对象
 - 对特征空间施加小波变换, 找出特征空间中的稠密区域
 - 使用小波变换多次, 得到由细到粗不同尺度的聚类
- 小波变换
 - 将信号分解到不同频率的子波段(可以用于 n -维信号)
 - 变换后的数据在不同的分辨率下保留对象之间的相对距离.
 - 数据的自然聚类变得更加容易识别

WaveCluster

- 为什么小波变换对聚类是有用的

- 提供了无指导的聚类

它采用了帽形(**hat-shape**)过滤, 强调点密集的区域, 而忽视在密集区域外的较弱的信息----特征空间中的密集区域成为了附近点的**吸引点(attractor)**, 距离较远的点成为抑制点(**inhibitor**). 这意味着数据的聚类自动地显示出来, 并“清理”了周围的区域

- 能够自动地排除孤立点

- 多分辨率

- 图8.16显示了不同分辨率的小波变换结果, 从细的尺度到粗的尺度. 在每一个层次, 显示了原始数据分解得到的四个子波段. 左上像限显示的子波段强调了每个数据点周围的平均邻域. 右上像限内的子波段强调了数据的水平边. 左下像限中的子波段强调了垂直边, 而右下像限中的子波段强调了转角

Quantization

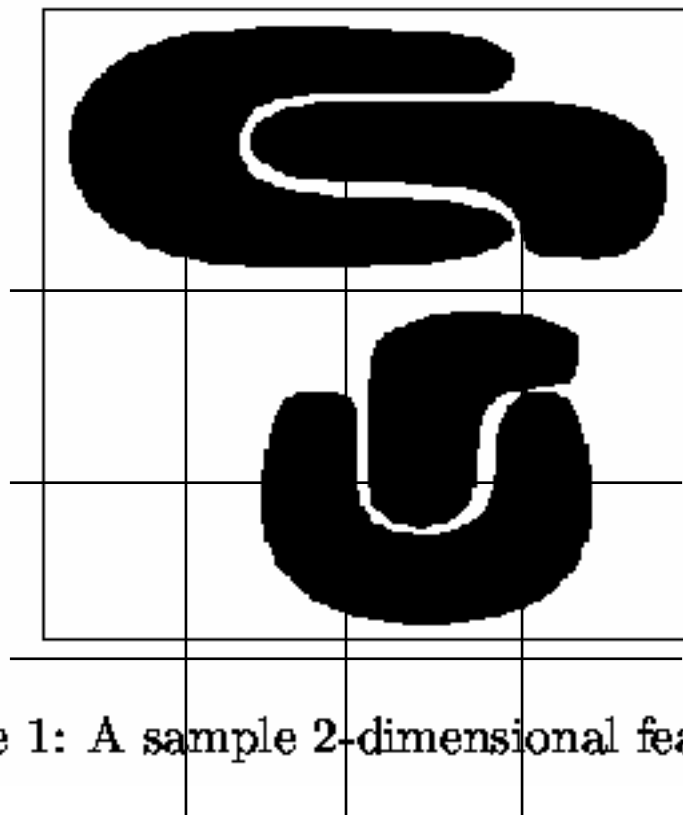


Figure 1: A sample 2-dimensional feature space.

Transformation

- 量化数据m-D grid structure, then wavelet transform
 - a) scale 1: high resolution
 - b) scale 2: medium resolution
 - c) scale 3: low resolution

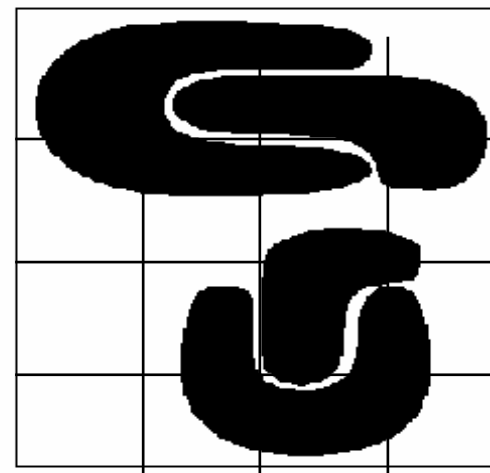


Figure 1: A sample 2-dimensional feature space.

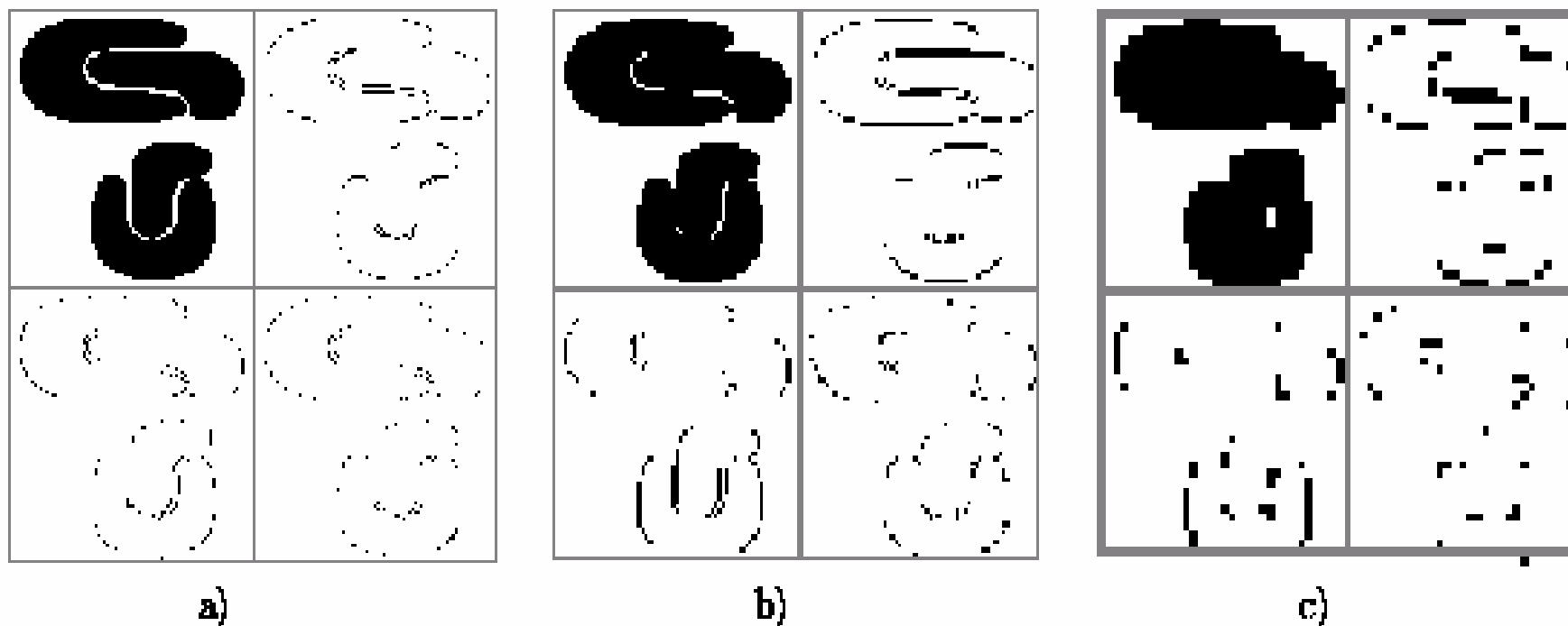


图7-16 特征空间的多分辨率的结果

WaveCluster

- 主要特点:
 - 复杂性 $O(N)$
 - 检测不同的尺度下的任意形状聚类
 - 对噪音不敏感, 对孤立点不敏感
 - 通常只能处理低维数据

第7章. 聚类分析

- 什么是聚类（**Clustering**）分析？
- 聚类分析中的数据类型
- 主要聚类方法分类
- 划分方法（**Partitioning Methods**）
- 层次方法（**Hierarchical Methods**）
- 基于密度的方法（**Density-Based Methods**）
- 基于网格的方法（**Grid-Based Methods**）
- 基于模型的聚类方法（**Model-Based Clustering Methods**）
- 孤立点分析（**Outlier Analysis**）
- 小结

基于模型的聚类方法

- 试图优化给定的数据和某些数学模型之间的拟合
- 三类基于模型的方法：
 - 统计学方法
 - 期望最大化方法
 - 概念聚类
 - 神经网络方法

EM 方法

- 每个分布代表一个簇， k 个概率分布的混合模型表示整个数据，估计分布参数拟合数据
- EM — 一种流行的迭代求精算法， k -means方法的一种扩展
 - 根据权重 (prob. distribution) 把对象指派到簇cluster
 - 基于权重度量计算新的均值
- 基本想法
 - 对参数向量进行初始估计/猜测
 - 反复地根据参数向量产生的混合密度对每个对象重新打分
 - 重新打分后的对象又用来更新参数向量
 - 如果根据他们的分数赋予一个特定的成员（分布），对象属于相同的聚类
- 算法收敛速度快，但可能达不到全局最优

The EM Algorithm

- 初始, 随机选择k 聚类中心
- 根据如下两步迭代求精
 - 期望步: 用以下概率将数据点 X_i 指派到cluster C_i

$$P(X_i \in C_k) = p(C_k|X_i) = \frac{p(C_k)p(X_i|C_k)}{p(X_i)},$$

- 最大化:
 - Estimation of model parameters

$$m_k = \frac{1}{N} \sum_{i=1}^N \frac{X_i P(X_i \in C_k)}{\sum_j P(X_i \in C_j)}.$$

概念聚类

- 概念聚类
 - 一种机器学习聚类方法
 - 给出一组未标记的对象，它产生一个分类模式
 - 为每组对象找出特征描述
- **COBWEB (Fisher'87)**
 - 一种简单的、流行的增量概念聚类算法
 - 以一个分类树的形式创建层次聚类

COBWEB

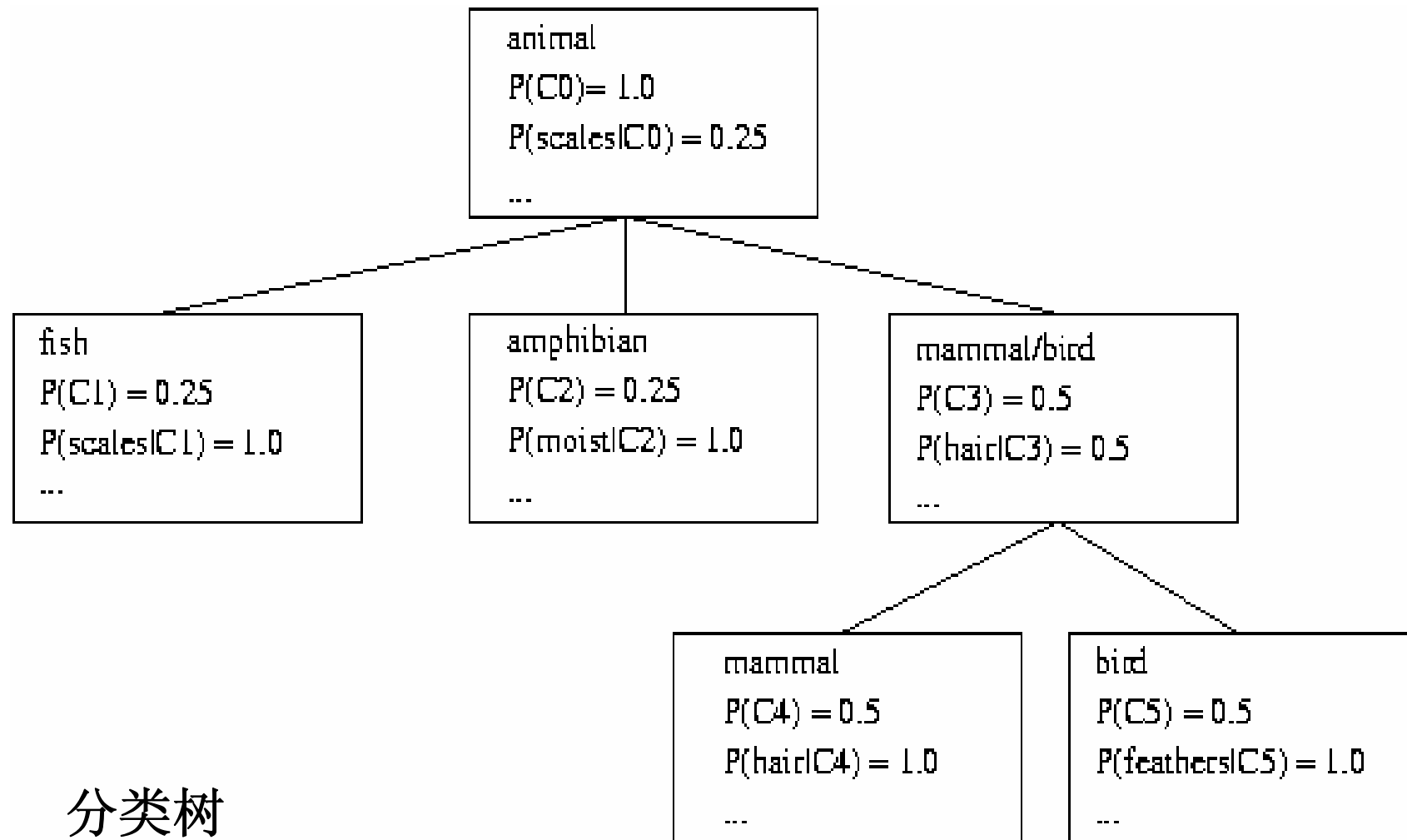
- 分类树

- 分类树中的每个节点对应一个概念, 包含该概念的一个概率描述, 概述了被分在该节点下的对象
- **概率描述**包括概念的概率和形如 $P(A_i = V_{ij} / C_k)$ 的条件概率, 这里 $A_i = V_{ij}$ 是一对属性和值, C_k 是概念类
- 为了用分类树对一个对象进行分类, 采用了一个部分匹配函数来沿着最佳匹配节点的路径在树中向下移动

- 分类树VS判定树

- 判定树标记分支, 而非节点, 而且采用逻辑描述符, 而不是概率描述符

COBWEB 聚类方法



分类树

COBWEB

- COBWEB采用了一个启发式估算度量——**分类效用** (Category Utility, CU)来指导树的构建, 分类效用定义如下

$$\frac{\sum_{i=1}^n P(C_k) \left[\sum_i \sum_j P(A_i = V_{ij} | C_k)^2 - \sum_i \sum_j P(A_i = V_{ij})^2 \right]}{n}$$

- **n**是在树的某个层次上形成一个划分{C1, C2, ..., Cn}的节点, 概念或“种类”的数目
- 概率 $P(A_i = V_{ij} / C_k)$ 表示类内相似性. 该值越大, 共享该属性-值的类成员比例越大, 该属性-值对类成员的预见性就越大
- $P(C_k / A_i = V_{ij})$ 表示类间相异性. 该值越大, 共享该属性-值却在其它类中的对象就越少, 该属性-值对类的预见性就越大

COBWEB

- **COBWEB**将对象增量地加入到分类树中
 - 给定一个新的对象, **COBWEB**沿着一条适当的路径向下, 修改计数, 寻找可以分类该对象的最好节点
 - 基于将对象临时置于每个节点, 计算结果划分的分类效用.
 - **COBWEB**也计算为给定对象创建一个新的节点所产生的分类效用.
 - 与基于现存节点的结果相比较, 根据产生最高分类效用的划分, 对象被置于一个已存在的类, 或者为它创建一个新类
 - 要注意**COBWEB**可以自动修正划分中类的数目, 它不需要用户提供这样的输入参数

COBWEB

- **COBWEB的限制**
 - 属性上的概率分布是彼此独立的 假定太强, 因为相关性可能存在
 - 不适合对大型数据库中的数据进行聚类: 倾斜的树, 计算概率分布的代价太高
- **COBWEB的一个扩展CLASSIT可以用于连续数据的聚类, 但也有同样的问题**
- **AutoClass (Cheeseman and Stutz, 1996)**
 - 使用**Bayesian** 统计分析估计聚类的个数
 - 在产业界很流行
- **将概念聚类用于数据挖掘需要进一步研究**

神经网络方法

- 神经网络方法将每个簇描述为一个标本(**exemplar**)
- 标本作为聚类的“原型”, 不一定对应一个特定的数据实例或对象
- 根据某些距离函数, 新的对象可以被分配给标与其本最相似的簇. 被分配给一个簇的对象的属性可以根据该簇的标本的属性来预测
- 两个比较著名的方法
 - 竞争学习(**competitive learning**)
 - 自组织特征映射(**Self-organizing feature maps**)这两种方法都涉及有竞争的神经单元

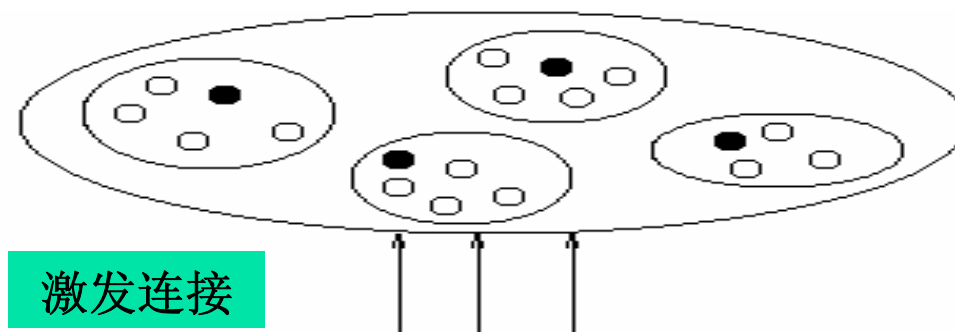
神经网络方法

■ 竞争学习

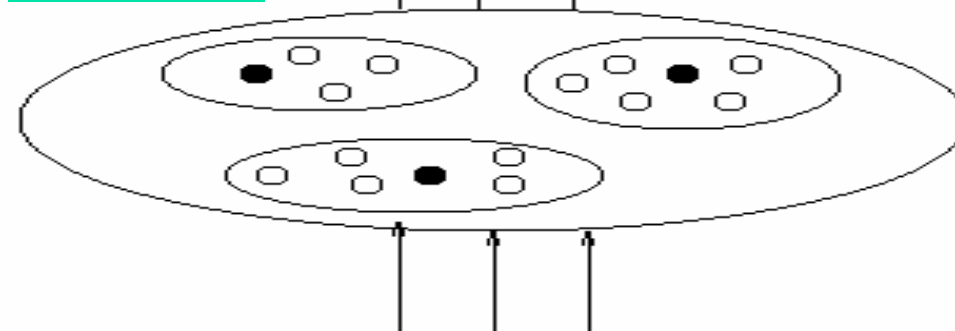
- 涉及若干个单元(“神经元” **neurons**)的层次结构
- 它们以一种“胜者全取(**winner-take-all**)”的方式对系统当前处理的对象进行竞争
- 下图显示了一个竞争学习系统的例子
 - 每个圆圈代表一个单元, 在一个簇中获胜的单元成为活跃的(以实心圆点表示), 而其它是不活跃的(以空心圆点表示)
 - 各层之间的连接是激发(**excitatory**)——在某个给定层次中的单元可以接收来自低一层次所有单元的输入
 - 在某个给定层次中, 一个簇中的单元彼此竞争, 对低一层的输出模式做出反应. 一个层次内的联系是抑制(**inhibitory**), 以便在任何簇中只有一个单元是活跃的

竞争学习结构

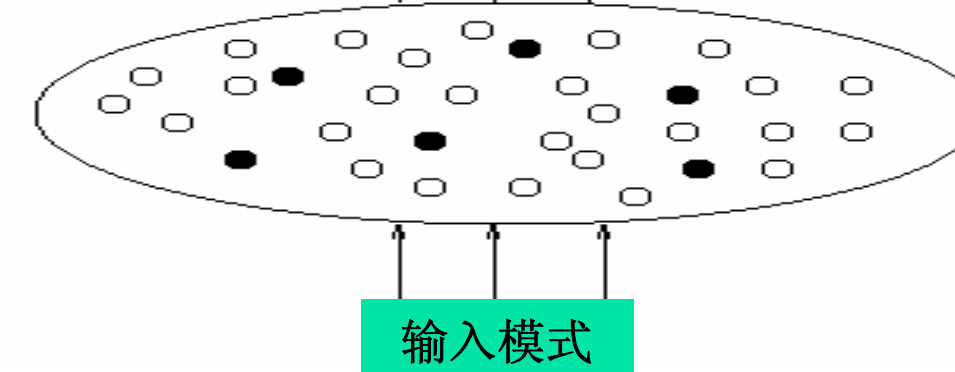
层3
抑制簇



层2
抑制簇



层1
输入单元



输入模式

自组织特征映射

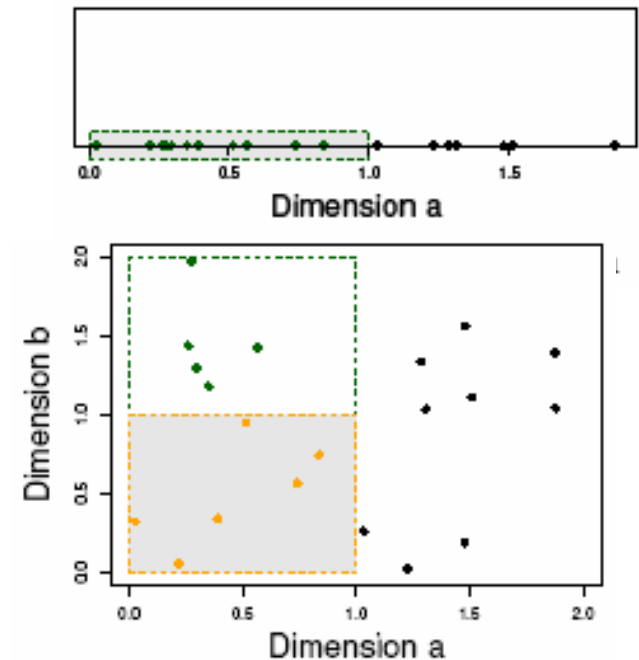
- 自组织特征映射 (**Self-organizing feature maps,SOM**)
 - 也是通过若干个单元竞争当前对象来进行聚类
 - 权重向量最接近当前对象的单元成为获胜的或活跃的单元
 - 为了更接近输入对象, 获胜单元及其最近的邻居的权重进行调整
 - **SOM**被认为类似于大脑的处理过程
 - 对在二或三维空间中直观化高维数据是有用的

聚类高维数据

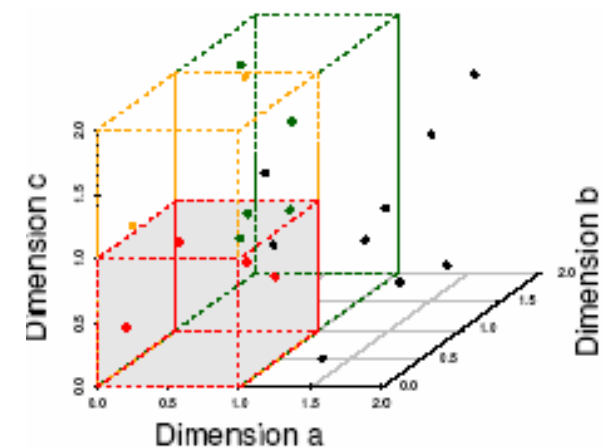
- 聚类高维数据，应用广泛: **text documents, DNA micro-array data**; 重要挑战:
 - 多个不相关的维度掩盖聚类
 - 距离函数变得没有意义—由于 **equi-distance**(高维空间，数据变稀疏)
 - 聚类可能存在于某些子空间中
- 特征变换: 仅当大部分维度与聚类相关时有效
 - **PCA & SVD**有效，当特征高度相关/冗余
- 特征选择: 缠绕**wrapper** 或 过滤方法
 - 当数据有很好的聚类结构时，很有效
- 子空间聚类: 在所有可能的子空间中寻找**clusters**
 - **CLIQUE, ProClus, and frequent pattern-based clustering**

维数灾难

- (graphs adapted from Parsons et al. KDD Explorations 2004)
- 一维的数据相对压缩的
- 增加一个维度将沿此维“伸展”数据点, 使得数据更分散
- 增加更多的维度将使得数据更稀疏—高维数据非常稀疏
- 距离变得没有意义—**due to equi-distance**



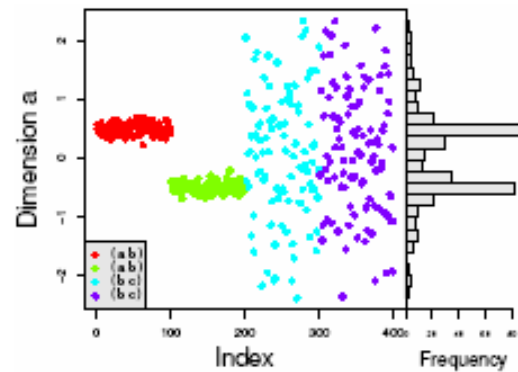
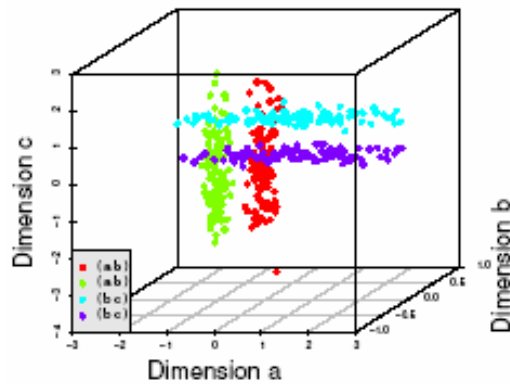
(b) 6 Objects in One Unit Bin



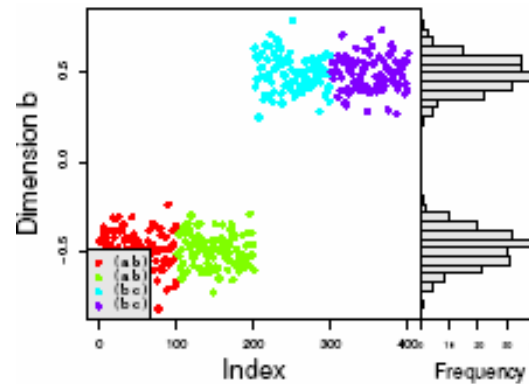
(c) 4 Objects in One Unit Bin

子空间聚类，为什么？

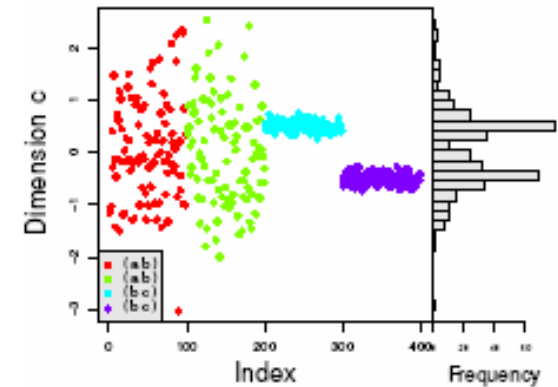
- 聚类可能只存在于某些子空间
- 子空间聚类: find clusters in all the subspaces



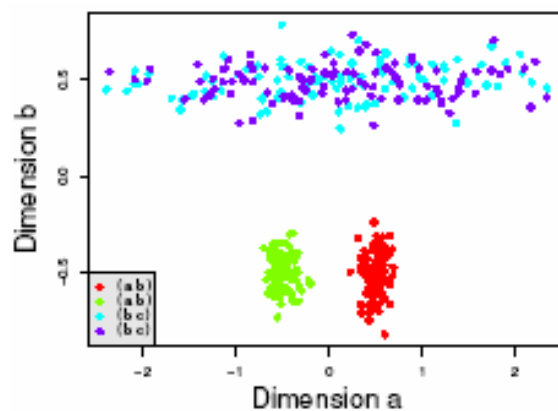
(a) Dimension a



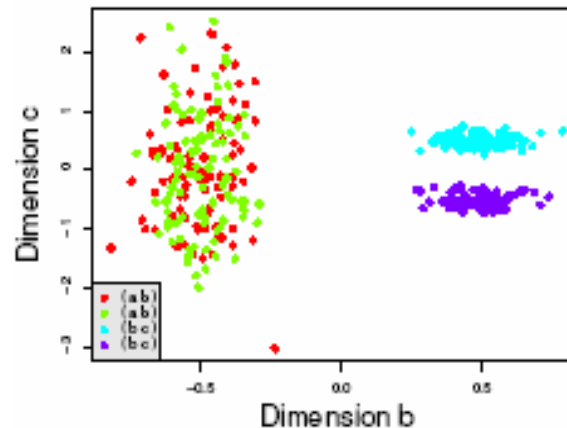
(b) Dimension b



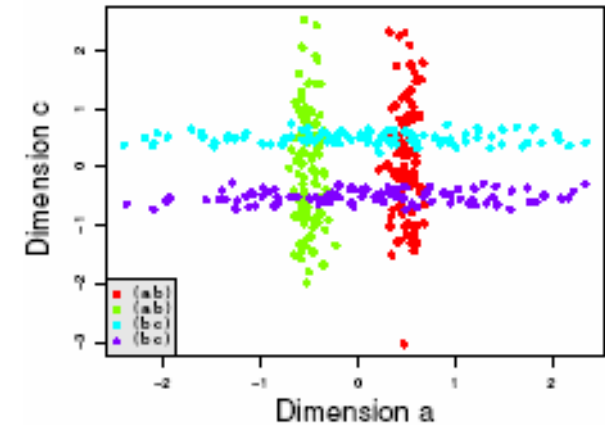
(c) Dimension c



(a) Dims a & b



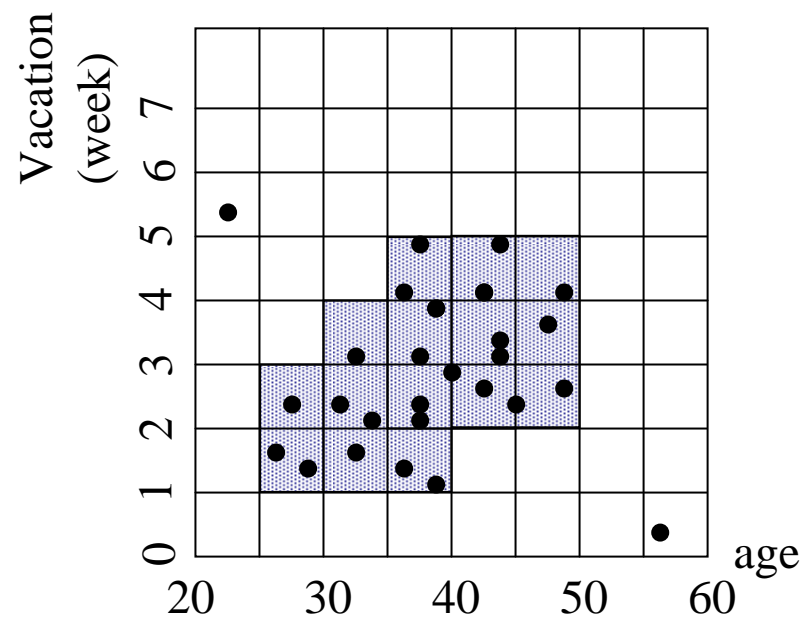
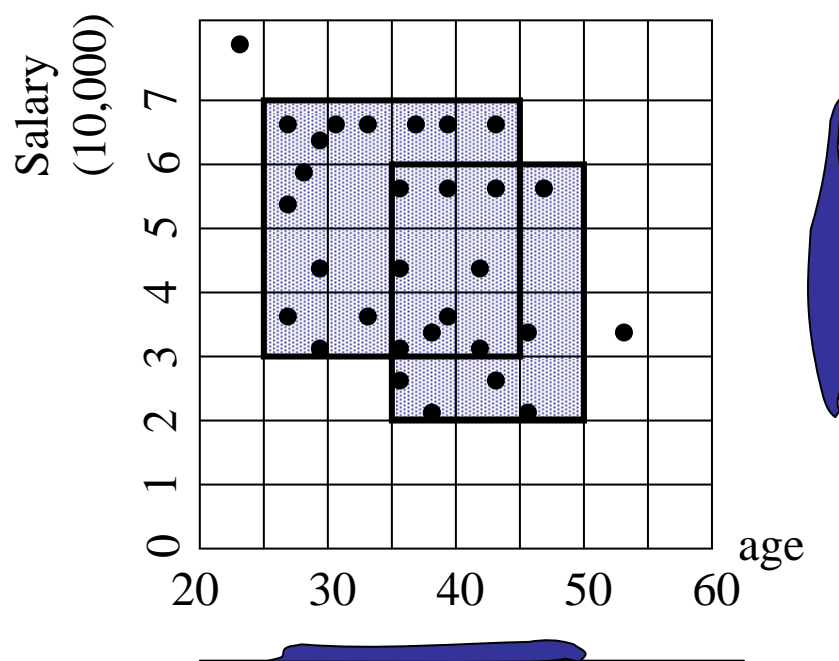
(b) Dims b & c



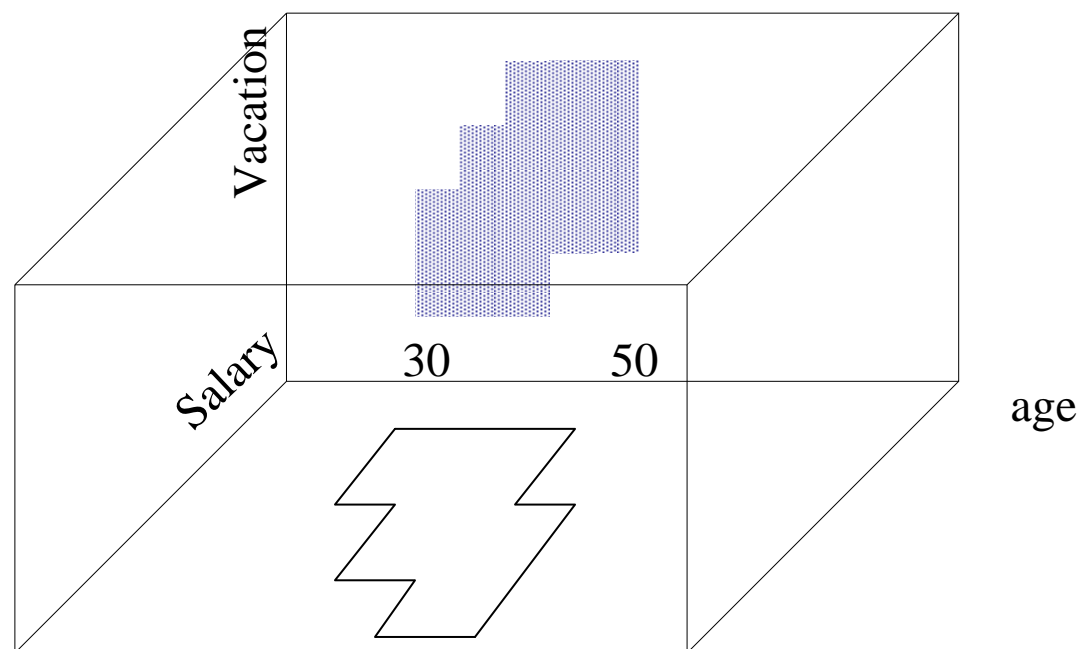
(c) Dims a & c

CLIQUE

- **CLIQUE(Clustering In QUEst)**综合了基于密度和基于网格的聚类方法. 由Agrawal, Gehrke, Gunopulos, Raghavan (SIGMOD'98)提出, 对于大型数据库中的高维数据的聚类非常有效.
- 自动识别高维数据空间的子空间, 在子空间上聚类比在原空间上更好
- 基本思想: **CLIQUE**是基于密度和基于网格的聚类方法
 - 它将每个维划分成相同个数的等长区间
 - 它将m-维数据空间划分成不重叠的长方形单元
 - 一个单元是**稠密**的, 如果包含在该单元中的数据点占全部数据点的比例超过输入的模式参数
 - 簇是相连的密集单元的最大集合



$\tau = 3$



CLIQUE

- CLIQUE所采用的先验性质(Apriori property)如下
 - 如果一个 k 维单元是密集的, 那么它在 $k-1$ 维空间上的投影也是密集的
 - 也就是说, 给定一个 k 维的候选密集单元, 如果我们检查它的 $k-1$ 维投影单元, 发现任何一个不是密集的, 那么我们知道第 k 维的单元也不可能是密集的
 - 可以从 $k-1$ 维空间中发现的密集单元来推断 k 维空间中潜在的或候选的密集单元. 通常, 最终的结果空间比初始空间要小很多

CLIQUE : 主要步骤

- 划分数据空间,并找出划分的每个单元中的点数.
- 使用Apriori性质, 识别包含聚类的子空间
- 识别聚类 :
 - 确定所有感兴趣的子空间中的稠密单元
 - 确定所有感兴趣的子空间中的连接的稠密单元.
- 产生聚类的最小描述
 - 对每个簇, 确定覆盖相连的密集单元的最大区域
 - 然后确定最小的覆盖

CLIQUE的优缺点

■ 优点

- 它自动地找出高维的子空间, 高密度的聚类存在于在这些子空间中
- 对元组的输入顺序不敏感, 无需假设任何规范的数据分布
- 它随输入数据的大小线性地扩展, 当数据的维数增加时具有良好的可扩展性

■ 缺点

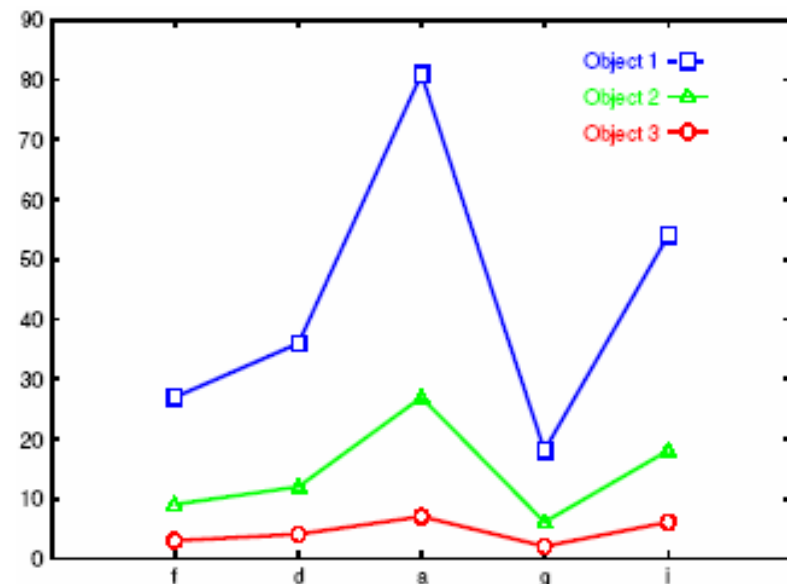
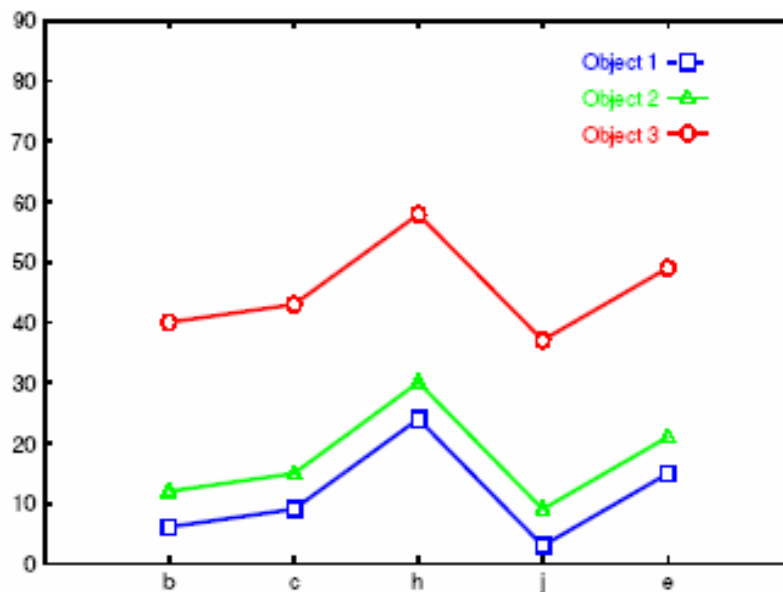
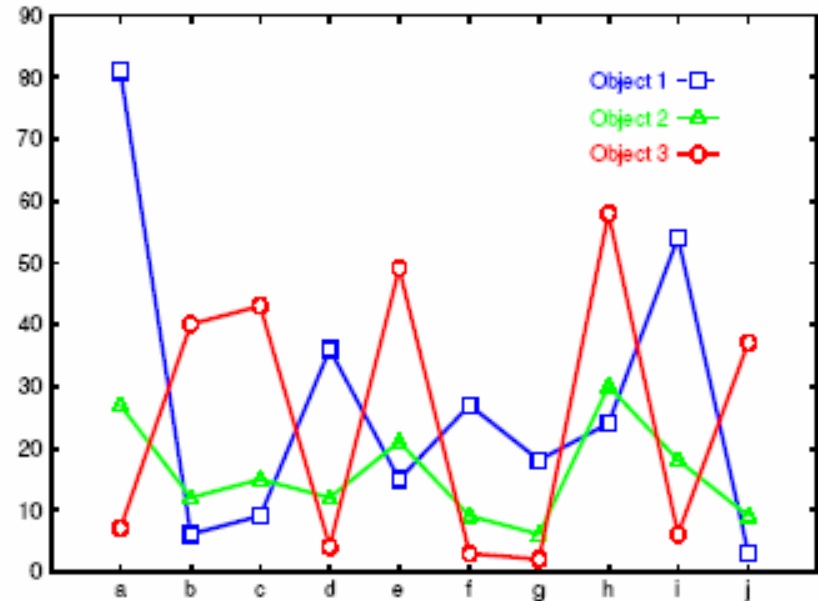
- 由于方法大大简化, 聚类结果的精确性可能会降低

基于频繁模式的方法

- 基本思想：发现的频繁模式也可能预示簇
- 典型的方法
 - 基于频繁相的文本聚类
 - 提取相，每个文档用项的集合表示
 - 项由单个或多个词组成
 - 挖掘频繁项集
 - 从频繁项集的集合中精选出的子集可以看成聚类
 - 精选子集覆盖所有的文档
 - 不同子集覆盖的部分间的重叠小
 - 模式相似性聚类微阵列数据 (pClustering)

基于模式相似性聚类 (p -Clustering)

- 原始的微阵列数据包含3个基因在多个条件（维度）下的表达值
 - 难以发现规律
- 特定的维度子集形成了有趣的模式
 - **shift** 平移模式（关于y）
 - **scaling** 缩放模式（关于y）



Why p -Clustering?

- 微阵列数据分析需要

- 针对数千个attributes 聚类
- 发现 **shift** 和 **scaling** 的模式

- 用Euclidean 距离来聚类 — 不能发现平移模式

- 1) 使用引入的新属性 $A_{ij} = a_i - a_j$? — 将会引入 $N(N-1)$ 维

- 2) 提出的Bi-cluster使用均方残差得分 $h(I, J)$

$$H(IJ) = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (d_{ij} - d_{iJ} - d_{IJ} + d_{IJ})^2$$

$$d_{ij} = \frac{1}{|J|} \sum_{j \in J} d_{ij}$$

$$d_{Ij} = \frac{1}{|I|} \sum_{i \in I} d_{ij}$$

- 子矩阵是 δ -cluster, 如果 $H(I, J) \leq \delta$ 对某个 $\delta > 0$

- 应用随机算法发现簇

$$d_{IJ} = \frac{1}{|I||J|} \sum_{i \in I, j \in J} d_{ij}$$

- bi-cluster的问题

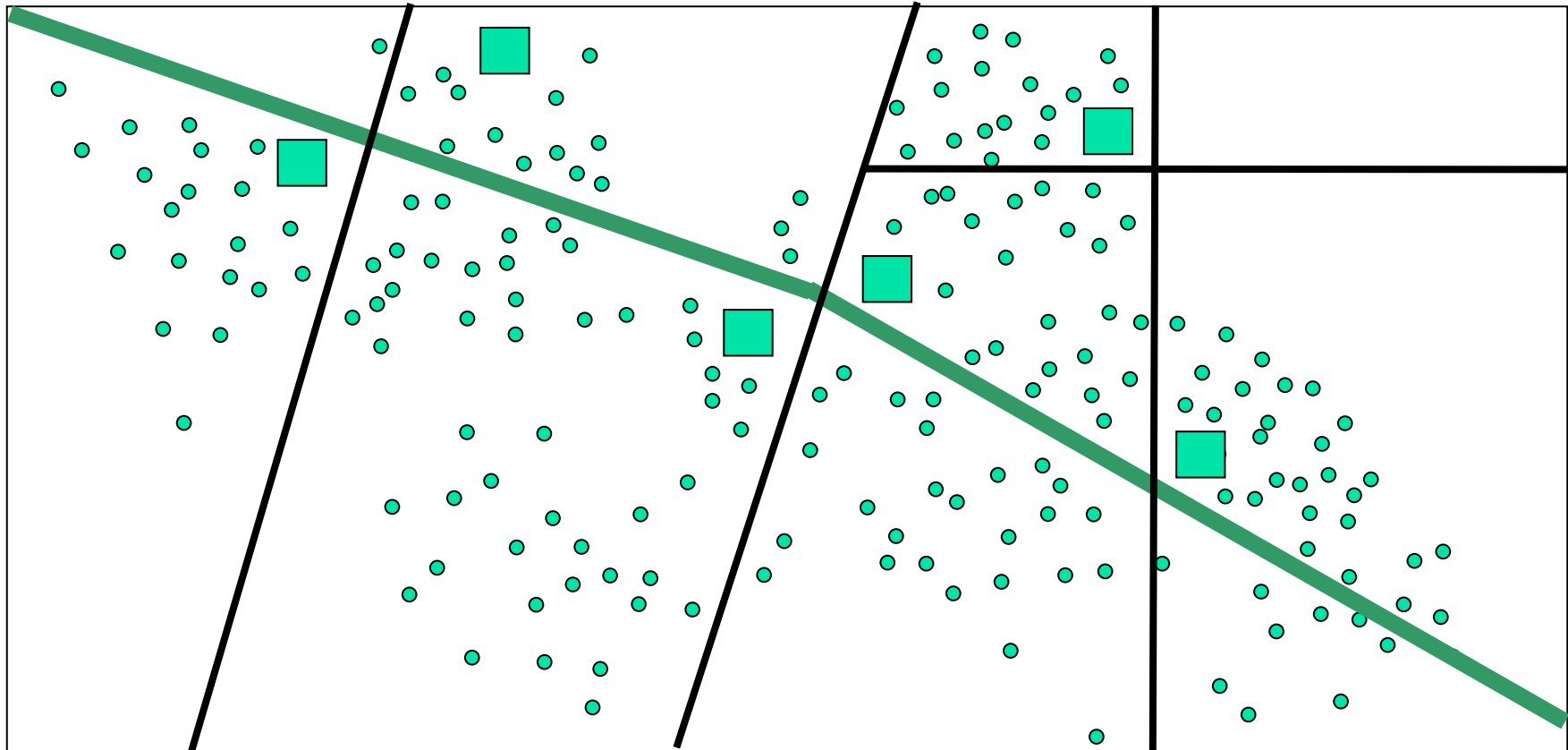
- 不能向下传递,即子矩阵不一定满足条件
- 由于取平均值, 可能包含孤立点 within δ -threshold

$$pScore \left(\begin{bmatrix} d_{xa} & d_{xb} \\ d_{ya} & d_{yb} \end{bmatrix} \right) = | (d_{xa} - d_{xb}) - (d_{ya} - d_{yb}) | \quad (p\text{-Clustering})$$

- 给定集合O中对象x, y 和T中特征a, b, pScore是基于2x2 matrix
- (O, T)是 δ -pCluster, 如果(O, T)的任何2x2 矩阵X, $pScore(X) \leq \delta (\delta > 0)$
- δ -pCluster的特点
 - 向下闭包特性
 - 得到的簇更同源, 相比于bi-cluster (要求两两组合满足条件)
- 已经提出模式增长的算法来挖掘
- 对缩放模式, 对下式取对数可以导出pScore $\frac{d_{xa} / d_{ya}}{d_{xb} / d_{yb}} < \delta$

基于约束的聚类分析?

- 需要用户反馈: 用户对应用需求有更清楚的认识
- 更少参数但更多的用户约束, e.g., ATM 分配问题: obstacle & desired clusters

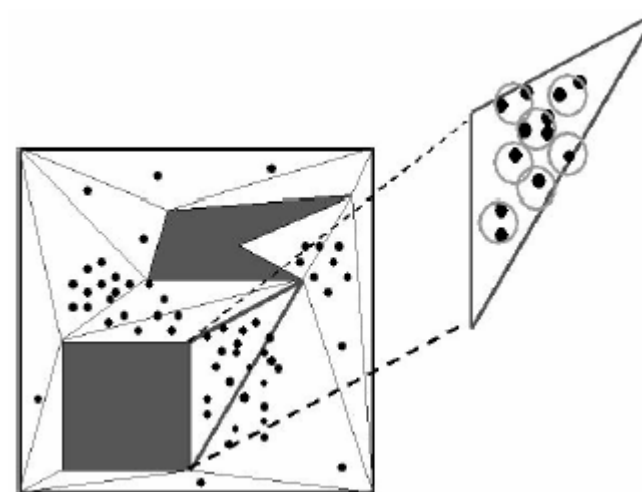
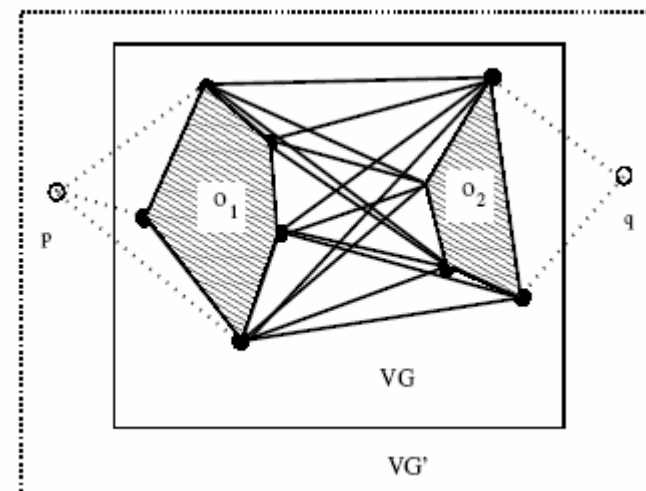


约束聚类的分类

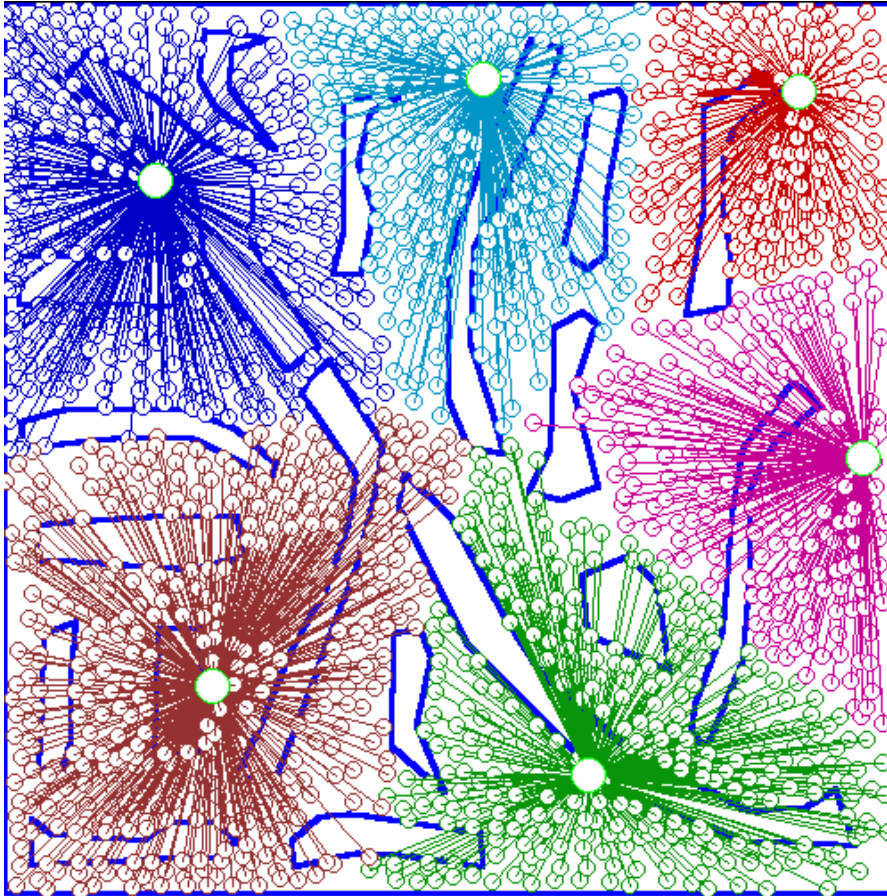
- 根据约束的种类:
 - 个体对象的约束 (对数据对象进行选择)
 - 房地产应用: 价值\$300K以上的房子聚类
 - 聚类参数的约束
 - # of clusters, MinPts, etc.
 - 距离或相似度函数的约束
 - Weighted functions, obstacles (e.g., rivers, lakes)
 - 用户指定的约束
 - 服务站设置: 每个簇包含至少500高价值客户和5000普通客户
 - 半监督聚类:
 - 给定某个小训练集作为约束

含障碍对象的聚类

- K-中心点方法更合适，k-means某个质心（分配ATM）可能是湖中心
- 构造可见图（最短路径）
- 三角划分，形成微簇
- 两类连接索引（基于最短路径）
 - VV index: 任意一对障碍物定点
 - MV index: 任意一对微簇和障碍物定点



An Example: Clustering With Obstacle Objects



Not Taking obstacles into account



Taking obstacles into account

用户约束的聚类

- 例：确定 k 个投递中心
- 方法
 - 划分数据 k 组，寻找一个满足约束的初始解
 - 迭代地调整微簇来改进结果 (e.g., 移动 δ μ -clusters 从 C_i 到 C_j)
 - 处理“死锁deadlock” (需要时分裂微簇)
- 对数据预处理形成微簇，可以提高效率

第7章. 聚类分析

- 什么是聚类（**Clustering**）分析？
- 聚类分析中的数据类型
- 主要聚类方法分类
- 划分方法（**Partitioning Methods**）
- 层次方法（**Hierarchical Methods**）
- 基于密度的方法（**Density-Based Methods**）
- 基于网格的方法（**Grid-Based Methods**）
- 基于模型的聚类方法（**Model-Based Clustering Methods**）
- 孤立点分析（**Outlier Analysis**）
- 小结

孤立点分析

- 什么是孤立点?
 - 对象的集合, 它们与数据的其它部分不一致
 - 孤立点可能是度量或执行错误所导致的
 - 孤立点也可能是固有的数据变异性的结果
- 问题
 - 给定一个 n 个数据点或对象的集合, 及预期的孤立点的数目 k , 发现与剩余的数据相比是相异的, 例外的, 或不一致的前 k 个对象
- 两个子问题:
 - 定义在给定的数据集合中什么样的数据可以被认为是不一致的
 - 找到一个有效的方法来挖掘这样的孤立点

孤立点分析

- 应用：
 - 信用卡欺诈检测
 - 电信欺诈检测
 - 顾客分割：确定极低或极高收入的客户的消费行为
 - 医疗分析：发现对多种治疗方式的不寻常的反应
- 孤立点的定义是非平凡的
 - 如果采用一个回归模型, 余量的分析可以给出对数据“极端”的很好的估计
 - 当在时间序列数据中寻找孤立点时, 它们可能隐藏在趋势的, 周期性的, 或者其他循环变化中, 这项任务非常棘手
 - 当分析多维数据时, 不是任何特别的一个, 而是维值的组合可能是极端的. 对于非数值型的数据（如分类数据）, 孤立点的定义要求特殊的考虑

孤立点分析

- 采用数据可视化方法来进行孤立点探测如何？
 - 不适用于包含周期性曲线的数据
 - 对于探测有很多分类属性的数据, 或高维数据中的孤立点效率很低
- 方法
 - 统计学方法
 - 基于距离的方法
 - 基于偏差的方法
 - 基于密度的方法

基于统计学的孤立点检测

- 对给定的数据集假设了一个分布或概率模型(例如, 正态分布), 然后根据模型采用不一致性检验(**discordancy test**)来确定孤立点
- 检验要求的参数
 - 数据集参数: 例如, 假设的数据分布
 - 分布参数: 例如平均值和方差
 - 和预期的孤立点的数目
- 统计学的不一致性检验需要检查的两个假设
 - 工作假设(**working hypothesis**)
 - 替代假设(**alternative hypothesis**)

基于统计学的孤立点检测

- 工作假设 H 是一个命题: n 个对象的整个数据集合来自一个初始的分布模型 F ,
 - 即 $H: O_i \in F, i = 1, 2, \dots, n$
- 不一致性检验验证一个对象 O_i 关于分布 F 是否显著地大(或者小)
- 依据关于数据的可用知识, 已提出不同的统计量用于不一致性检验
- 假设某个统计量被选择用于不一致性检验, 对象 O_i 的该统计量的值为 V_i , 则构建分布 T
 - 估算显著性概率 $SP(V_i) = Prob(T > V_i)$
 - 如果某个 $SP(V_i)$ 是足够的小, 那么 O_i 是不一致的, 工作假设被拒绝. 替代假设被采用, 它声明 O_i 来自于另一个分布模型 G

基于统计学的孤立点检测

- 结果非常依赖于模型 F 的选择
 - O_i 可能在一个模型下是孤立点, 在另一个模型下是非常有效的值
- 替代分布在决定检验的能力上是非常重要的
- 不同的替代分布
 - 固有的替代分布(**inherent alternative distribution**): 所有对象来自分布 F 的工作假设被拒绝, 而所有对象来自另一个分布 G 的替代假设被接受
 - 混合替代分布(**mixture alternative distribution**): 不一致的值不是 F 分布中的孤立点, 而是来自其他分布的污染物
 - 滑动替代分布(**slippage alternative distribution**): 所有的对象(除了少量外)根据给定的参数, 独立地来自初始的模型 F , 而剩余的对象是来自修改过的 F 的独立的观察

基于统计学的孤立点检测

- 检测孤立点有两类基本的过程
 - 批(**block**)过程: 或者所有被怀疑的对象都被作为孤立点对待, 或者都被作为一致数据而接受
 - 连续的过程:
该过程的一个例子是内部出局(**inside-out**)过程
 - 主要思想
 - 首先检验最不可能是孤立点的对象. 如果它是孤立点, 那么所有更极端的值都被认为是孤立点; 否则, 检验下一个极端的对象, 依次类推
 - 该过程往往比批过程更为有效

基于统计学的孤立点检测

■ 缺点

- 绝大多数检验是针对单个属性的,而许多数据挖掘问题要求在多维空间中发现孤立点
- 统计学方法要求关于数据集合参数的知识(如,数据分布),但是在许多情况下,数据分布可能是未知的
- 当没有特定的检验时,统计学方法不能确保所有的孤立点被发现;或者观察到的分布不能恰当地被任何标准的分布来模拟

基于距离的孤立点检测

- 为了解决统计学方法带来的一些限制，引入了基于距离的孤立点的概念
- 基于距离的孤立点：
 - $DB(p, d)$ -孤立点是数据集 T 中的一个对象 o ，使得 T 中的对象至少有 p 部分与 o 的距离大于 d
- 将基于距离的孤立点看作是那些没有“足够多”邻居的对象。这里的邻居是基于距给定对象的距离来定义的
- 对许多不一致性检验来说，如果一个对象 o 根据给定的检验是一个孤立点，那么对恰当定义的 p 和 d ， o 也是一个 $DB(p, d)$ 孤立点
 - 例如，如果离平均值偏差3或更大的对象被认为是孤立点，假设一个正态分布，那么这个定义能够被一个 $DB(0.9988, 0.13 \sigma)$ 孤立点所概括

基于距离的孤立点挖掘算法

■ 基于索引的算法

- 采用多维索引结构, R 树或 k - d 树, 来查找每个对象 o 在半径 d 范围内的邻居
- 设 M 是一个孤立点的 d -邻域内的最大对象数目. 一旦对象 o 的 $M+1$ 个邻居被发现, o 就不是孤立点
- 最坏情况下的复杂度为 $O(kn^2)$, 这里 k 是维数, n 是数据集合中对象的数目
- 建造索引的任务是计算密集的

■ 嵌套循环算法

- 嵌套-循环算法和基于索引的算法有相同的计算复杂度, 但它避免了索引结构的构建, 试图最小化I/O的次数
- 它把内存的缓冲空间分为两半, 把数据集合分为若干个逻辑块. 通过精心选择逻辑块装入每个缓冲区域的顺序, I/O效率能够改善

基于距离的孤立点挖掘算法

- 基于单元(**cell-based**)的算法
 - 为了避免 $O(n^2)$ 的计算复杂度, 为驻留内存的数据集合开发了基于单元算法. 它的复杂度是 $O(c^k + n)$, 这里 c 是依赖于单元数目的常数, k 是维数
- 方法
 - 数据空间被划分为单元, 单元的边长等于 $d / 2\sqrt{k}$
 - 每个单元有两层围绕着. 第一层的厚度是一个单元, 而第二层的厚度是 $[2\sqrt{k} - 1]$
 - 算法逐个单元地对孤立点计算, 而不是逐个对象地进行计算. 对一个给定的单元, 它累计三个计数——单元中对象的数目 **cell_count**, 单元和第一层中对象的数目 **cell+_1_layer_count**, 及单元和两个层次中的对象的数目 **cell+_2_layer_count**

基于距离的孤立点挖掘算法

- 确定孤立点

设 M 是一个孤立点的 d -邻域中可能存在的孤立点的最大数目

- 如果 $\text{cell_+_1_layer_count}$ 大于 M , 那么该单元中所有的对象可以从进一步的考察中移走, 因为它们不可能是孤立点
- 如果 $\text{cell_+_2_layers_count}$ 小于或等于 M , 那么单元中所有的对象被认为是孤立点
- 否则, 对单元中的每个对象 o , 检查 o 的第二层中的对象. 只有那些 d -邻域内的对象不超过 M 个的点是孤立点

基于偏离的孤立点检测

- 通过检查一组对象的主要特征来确定孤立点
- 与给出的描述偏离的对象被认为是孤立点
- 序列异常技术(**sequential exception technique**)
 - 模仿人类从一系列推测类似的对象中识别异常对象的方式
- 术语
 - 异常集(**exception set**): 它是偏离或孤立点的集合, 被定义为某类对象的最小子集, 这些对象的去除会导致剩余集合的相异度的最大减少
 - 相异度函数(**dissimilarity function**): 是满足如下条件的任意函数: 当给定一组对象时, 如果对象间相似, 返回值就较小。对象间的相异度越大, 函数返回的值就越大

基于偏离的孤立点检测

例: 给定 n 个对象的子集合 $\{x_1, \dots, x_n\}$, 一个可能的相异度函数是集合中对象的方差

- 基数函数(cardinality function):
 - 一般是给定的集合中对象的数目
- 平滑因子(smoothing factor):
 - 一个为序列中的每个子集计算的函数.
 - 它估算从原始的数据集合中移走子集合可以带来的相异度的降低程度.
 - 平滑因子值最大的子集是异常集
- 一般的寻找异常集的任务可以是NP完全的(即, 难处理的).

基于偏离的孤立点检测

- 一个顺序的方法在计算上是可行的, 能够用一个线性的算法实现
 - 不考虑估算当前子集关于其补集的相异度, 该算法从集合中选择了子集合的序列来分析
 - 对每个子集合, 它确定其与序列中前一个子集合的相异度差异
 - 为了减轻输入顺序对结果的任何可能的影响, 以上的处理过程可以被重复若干次, 每一次采用子集合的一个不同的随机顺序
 - 在所有的迭代中有最大平滑因子值的子集合成为异常集

基于偏离的孤立点检测

- **OLAP 数据方技术**
 - 使用数据方识别大型多维数据中的异常区域

Summary

- **Cluster analysis** groups objects based on their **similarity** and has wide applications
- Measure of similarity can be computed for **various types of data**
- Clustering algorithms can be **categorized** into partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods
- **Outlier detection** and analysis are very useful for fraud detection, etc. and can be performed by statistical, distance-based or deviation-based approaches
- There are still lots of research issues on cluster analysis, such as **constraint-based clustering**

References (1)

- R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. SIGMOD'98
- M. R. Anderberg. Cluster Analysis for Applications. Academic Press, 1973.
- M. Ankerst, M. Breunig, H.-P. Kriegel, and J. Sander. Optics: Ordering points to identify the clustering structure, SIGMOD'99.
- P. Arabie, L. J. Hubert, and G. De Soete. Clustering and Classification. World Scietific, 1996
- M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases. KDD'96.
- M. Ester, H.-P. Kriegel, and X. Xu. Knowledge discovery in large spatial databases: Focusing techniques for efficient class identification. SSD'95.
- D. Fisher. Knowledge acquisition via incremental conceptual clustering. Machine Learning, 2:139-172, 1987.
- D. Gibson, J. Kleinberg, and P. Raghavan. Clustering categorical data: An approach based on dynamic systems. In Proc. VLDB'98.
- S. Guha, R. Rastogi, and K. Shim. Cure: An efficient clustering algorithm for large databases. SIGMOD'98.
- A. K. Jain and R. C. Dubes. Algorithms for Clustering Data. Printice Hall, 1988.

References (2)

- L. Kaufman and P. J. Rousseeuw. Finding Groups in Data: an Introduction to Cluster Analysis. John Wiley & Sons, 1990.
- E. Knorr and R. Ng. Algorithms for mining distance-based outliers in large datasets. VLDB'98.
- G. J. McLachlan and K.E. Bkassford. Mixture Models: Inference and Applications to Clustering. John Wiley and Sons, 1988.
- P. Michaud. Clustering techniques. Future Generation Computer systems, 13, 1997.
- R. Ng and J. Han. Efficient and effective clustering method for spatial data mining. VLDB'94.
- E. Schikuta. Grid clustering: An efficient hierarchical clustering method for very large data sets. Proc. 1996 Int. Conf. on Pattern Recognition, 101-105.
- G. Sheikholeslami, S. Chatterjee, and A. Zhang. WaveCluster: A multi-resolution clustering approach for very large spatial databases. VLDB'98.
- W. Wang, Yang, R. Muntz, STING: A Statistical Information grid Approach to Spatial Data Mining, VLDB'97.
- T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH : an efficient data clustering method for very large databases. SIGMOD'96.