

52-56

关系型数据库中的归纳依赖关系*

Induced dependency in relational databases

欧阳为民^{1,2}(安徽大学计算中心 合肥230039)¹蔡庆生²(中国科学技术大学计算机系 合肥230027)²

TP311.13

摘要 In this paper, we mainly discuss the induced dependency which play an important role in knowledge discovery, and define a set of static and temporal induced dependencies. Integration of inductive data model with relational data model is important for research and application of KDD.

关键词 Data dependency, Induced dependency, Knowledge discovery

1 引言

数据依赖是数据库的一个重要概念,函数依赖则是一种常见的依赖关系.本文主要论述另一种对知识发现有重要作用的归纳依赖(induced dependency,简称ID).考察下列关系数据库模式:

Student, Name	StudID	Department
Brown	3	computing
Green	5	math
Whilte	6	educaion
Grey	8	ecology
Black	9	math

Enrollment, StudID	Course	Year	Grad
3	DB1	90	pass
3	OS2	89	pass
5	DB1	90	fail
6	DB1	90	pass
6	PH3	90	Pass
8	OS1	90	fail
9	IDB	91	pass

Lecture, Teacher	Department	Location
ALN	ecology	building2
JDP	education	building1
JFR	computing	building3
JGH	math	building4
NGC	computing	building3
RSN	math	building3
TJR	computing	building3

Allocation, Course	Year	Teacher
DB1	90	JFR
DB2	89	NGC
IDB	91	JFR
OS1	90	TJR
OS2	89	NGC
PH3	90	TJR

相应的已知函数依赖关系及外部码依赖关系(合在一起简称为FD关系)为:

- Student, StudID $\rightarrow$ Student (Name, Department) (1)
 Enrollment (StudID, Course, Year) $\rightarrow$ Enrollment, Grad (2)
 Allocation (Course, Year) $\rightarrow$ Allocation, Teacher (3)
 Lecture, Teacher $\rightarrow$ Lecture (Department, Location) (4)

- Enrollment, StudID $\rightarrow$ Student, StudID (5)
 Enrollment (Course, Year) $\rightarrow$ Allocation (Course, Year) (6)
 Allocation, Teacher $\rightarrow$ Lecture, Teacher (7)

归纳依赖关系可以不严格地定义为没有事先在数据库模式中说明的与数据库中当前数据相吻合的函数依赖关系.注意,归纳依赖关系不必在演绎意义上为真,只要对已知数据是正确的就可以了.更准确地说,归纳依赖关系仅是大量数据的一种如实反映.对客体模型的任何推论均可视为归纳,因而易于产生偏差.在上述实例数据库中,我们有如下一些归纳依赖关系:

- Student, Name $\rightarrow$ Student, StudID (8)
 Allocation, Course $\rightarrow$ Allocation, Teacher (9)

这种归纳依赖关系也存在于不同关系之间.例如:

Allocation, Course $\rightarrow$ Lecture, Department

就是关系 Allocation 与 Lecture 之间的一种归纳依赖关系.其意为:给定 Course 就可以确定开设这门课的教师所在的系.也就是说,当存在关系间的归纳依赖关系时,就好象这些关系已按照其候选关键字进行了链接(join).这样一来,函数依赖就可作为归纳依赖定义的一部分.显然,这要求我们利用数据的任何已知结构.然而,即使将观察或推理局限在外部码依赖关系上,如果两个关系间存在多个可将两者链接在一起的外部码,归纳依赖的语义解释就会产生歧义.这时,我们可以通过定义用户视图,即明确指出在所发现的归纳依赖中所涉及的外部码依赖关系以解决歧义问题.在上例中,假定在关系 Alloca-

* 本文得到国家自然科学基金项目与国家教委博士点基金资助.欧阳为民 副教授,在职博士生,主要研究方向:KDD,人工智能及其应用,蔡庆生 教授,博士生导师,主要研究方向:机器学习,知识发现,人工智能.

tion 和 Lecture 上通过外部码(7)定义用户视图 AlloLect,从而式子(9)改写为:

$$\text{AlloLect. Course} \rightarrow \text{AlloLect. Department} \quad (10)$$

2 静态归纳依赖

本节以上述关系型数据库为例,讨论关系型数据库中若干静态归纳依赖关系及其性质。

2.1 完全归纳依赖

关系模式 R 为属性 $A_1, A_2, A_3, \dots, A_n$ 的集合,其中 $A_i (1 \leq i \leq n)$ 均有其相应的值域 D_i 。 R 上的元组 t 为笛卡尔集 $D_1 \times D_2 \times D_3 \times \dots \times D_n$ 的元素。 R 上的关系 r 为定义在 R 上的元组的有限集合。设 r 为 R 上的关系, X 和 Y 为 R 的子集,若 r 中不存在两个元组在 X 中的属性值相等,而在 Y 中的属性值必不等,则称 Y 函数依赖于 X , 记作 $X \rightarrow Y$ 。据此,结合归纳的含义,我们给出完全归纳依赖关系的定义。

定义1 令 r 为定义在 R 上的非空关系, X 和 Y 为 R 的子集。在 X 和 Y 之间存在完全归纳依赖关系,如果:(1) r 满足 $X \rightarrow Y$; (2) R 上的任何 FD 关系不蕴涵 $X \rightarrow Y$ 。

上述实例数据库中,我们有归纳依赖关系 $\text{Student}(\text{Name}, \text{Department}) \rightarrow \text{Student.StudID}$,但它不是最小化的,最小化可以通过强加如下要求达到:

对 R 中的关系 r , 如果 $X \rightarrow Y$, 并且不存在 X 的任何真子集 X' , 有 $X' \rightarrow Y$, 那么 $X \rightarrow Y$ 是最小化的归纳依赖关系。归纳依赖关系 $\text{Student.StudID} \rightarrow \text{Student.Name}$ 不是完全归纳依赖关系,因为条件(2)要求归纳依赖关系不同于 FD 关系,也不属于 FD 关系的传递闭包。显然,归纳依赖关系的定义必须使得 FD 依赖关系的集关系成为 ID 关系的子集。然而,在实践中, ID 关系和 FD 关系有着不同的作用,因而条件(2)要求它们互不相同。而且,这一要求与归纳结论不必演绎为真的观点是一致的。不过,在某些场合有必要将函数依赖与归纳依赖放在一起考虑。此时,我们将这两种依赖关系的集合称为依赖关系集,其中的成员记为 $X \rightarrow^+ Y$ 。

定义2(依赖关系集的成员) 依赖关系 $X \rightarrow^+ Y$ 在依赖关系集中,如果它满足下列条件之一:(1) $X \rightarrow Y$; (2) $X \rightarrow \neg Y$ 。

2.2 值敏感归纳依赖关系

值敏感归纳依赖关系(Value Sensitive Induced Dependency, 简称 VSID)进一步扩展了归纳依赖关系的概念。在有的场合,只是某属性集的某个属性值

子集可以推导出另一属性集中的属性值,或者某属性仅能推出另属性集的可以识别的属性值子集。考虑前述 Lecture 关系,在属性 Department 和 Location 之间不存在 FD 关系与 ID 关系,但对 Lecture 作如下标识在方括号中的选择后,却存在值敏感归纳依赖关系:

$$\text{Lecture. Department}[\text{'computing'}] \rightarrow \text{Lecture. Location} \quad (11)$$

该式可解释为:所有计算机系的职员都在同一栋楼。类似地,我们还可以将限制加在依赖关系的目标上。例如:

$$\text{Lecture. Department} \rightarrow \neg \text{Lecture. Location}[\text{'building2'}, \text{'building3'}] \quad (12)$$

其含义为:所有位于 building2 或 building3 的系就在该楼,或者说,没有一个系的职员同时在这两栋楼 building2 和 building3 中。VSID 关系可形式化定义如下:

定义3 令 r 为定义在 R 上的非空关系, X, Y 为 R 的子集, D_x, D_y 分别为 X, Y 的值域, D'_x, D'_y 分别为 D_x, D_y 的真子集, X 和 Y 之间关于 D'_x, D'_y 存在值敏感归纳依赖关系,如果 $X \rightarrow \neg Y$ 对 r' 成立,其中 $r' = \alpha_{x \in D'_x, y \in D'_y}(r)$ 。

说明:(1)当 D'_x, D'_y 分别与 D_x, D_y 相同时,值敏感归纳依赖关系就是归纳依赖关系;(2)限制 X 的值域并不要求相应限制 Y 的值域,反之亦然;(3)对完全归纳依赖关系的弱化程度 D'_x, D'_y 相对于 D_x, D_y 减少程度。

2.3 弱归纳依赖关系

在某些场合,特别是在数据量很大的场合,不存在归纳依赖关系并不意味着属性值间没有令人感兴趣的关系(interesting correlation)。让我们再次考察关系 Lecture,其中只有在两种情况下(数学系的职员),职员的位置不能从其所在的系推导出。于是有:

$$\text{Lecture. Department} \rightarrow \neg \text{Lecture. Location} \quad (13)$$

很明显,令人感兴趣关系的确定是与上下文有关的,就是说这种关系并不总是成立的。那么,究竟存在多少例外时,我们可以接受呢?通常我们采用阈值的方法来处理噪音数据和例外情况,即对低于一定数量的例外是可以忽略的。其它象置信度因子等处理不确定性的方法均可运用。当然,我们应根据特定的应用需要选择合适的方法。这里,我们采用百分数来描述关系在多大程度上是成立的。式(13)可改写为:

$$\text{Lecture. Department} \rightarrow \neg (71\%) \text{Lecture. Location} \quad (14)$$

定义4 令 r 为定义在 R 上的非空关系, X, Y 为 R 的子集, 在 X 和 Y 之间存在信任度为 k 的弱归纳依赖关系 $X \rightarrow^-(k) Y$, 如果对任意给定的信任值 $k' (k' \leq k)$ 有 $X \rightarrow^- Y$ 。

2.4 弱值敏感归纳依赖关系

弱值敏感归纳依赖关系是上述弱归纳依赖关系与值敏感归纳依赖关系的结合, 即某属性的属性值子集在某种程度上可以决定另一属性的属性值, 弱 VSID 关系在实践中是十分重要的, 但其有效实现很困难, 有待进一步研究。

2.5 归纳依赖关系的若干性质

与完全归纳依赖关系相比, 值敏感归纳依赖关系和弱归纳依赖关系分别是基于水平划分和信任度衍生出来的, 值敏感归纳依赖关系可以视为水平弱化, 因为它不要求全部元组都满足指定的性质, 所选择运算必须建立一个满足完全归纳依赖的关系。类似地, 弱归纳依赖关系是以信任度的方式弱化的, 因为目标属性只要以指定的信任级别满足依赖关系。

归纳依赖关系具有下列运算性质, 其中 X, Y, Z 为属性集, D'_1, D'_2 和 D'_3 分别为 D_1, D_2 和 D_3 的子集。

- 1) 定义 $X \rightarrow^+ Y$ 蕴涵 $X \rightarrow Y$ 或者 $X \rightarrow^- Y$
- 2) 反身律 $X \rightarrow^- X$ 不成立, 因为有 $X \rightarrow X$, 以及完全归纳依赖关系的条件(2)
- 3) 增强律 $X \rightarrow^- Y$ 蕴涵 $XZ \rightarrow^- Y$
- 4) 加法律 $X \rightarrow^- Y$ 且 $X \rightarrow^- Z$ 蕴涵 $X \rightarrow^- YZ$
- 5) 投影律 $X \rightarrow^- YZ$ 而 蕴涵 $X \rightarrow^+ Y$
 $X \rightarrow^- Y$ 不成立
- 6) 传递律 $X \rightarrow^- Y$ 且 $Y \rightarrow^- Z$ 蕴涵 $X \rightarrow^- Z$
 $X \rightarrow^- Y$ 且 $Y \rightarrow Z$ 蕴涵 $X \rightarrow^+ Z$
 $X \rightarrow Y$ 且 $Y \rightarrow^- Z$ 蕴涵 $X \rightarrow^+ Z$
- 7) 伪传递律 $X \rightarrow^- Y$ 且 $YZ \rightarrow^- W$ 蕴涵 $XZ \rightarrow^- W$
 $X \rightarrow^- Y$ 且 $YZ \rightarrow W$ 蕴涵 $XZ \rightarrow^+ W$
 $X \rightarrow Y$ 且 $YZ \rightarrow^- W$ 蕴涵 $XZ \rightarrow^+ W$

对值敏感归纳依赖关系, 只要所用值域子集是指定属性集的同一个值域子集, 上述规律同样是成立的, 以加法律为例:

$X[D'_1] \rightarrow^- Y[D'_1], X[D'_1] \rightarrow^- Z[D'_1]$ 蕴涵 $X[D'_1] \rightarrow^- Y[D'_1]Z[D'_1]$

但是, $X[D'_1] \rightarrow^- Y[D'_1], X[D'_1] \rightarrow^- Z[D'_1]$ 不蕴涵 $X[D'_1] \rightarrow^- Y[D'_1]Z[D'_1]$

也不蕴涵 $X[D'_1] \rightarrow^- Y[D'_1]Z[D'_1]$

3 时态归纳依赖关系

上节我们讨论的归纳依赖关系是静态的, 即数据库中的数据不随时间发生变化。为处理随时间变化的数据, 我们有必要引入所谓的时态归纳依赖关系 (Temporal Induced Dependency, 简称 TID)。这种归纳依赖关系既可用于时态数据库, 也可用于具有时变特征的关系型数据库。

提出 TID 关系的动机在于自动分析大量时变数据, 并试图从中提取有意义的属性/元组间的关系 (一般都具有时间约束)。根据所拥有的数据的语义, 这类关系可表现为因果关系、必要的约束或巧合。通常, 计算机系统不负责识别这类数据间的联系, 而将责任留给有关的人类领域专家。作为例子, 我们考察下列时态关系数据库:

LOG: Computer	Status	Reason	Start	End
VAX-1	online		00:00	10:00
VAX-2	online		00:00	10:00
VAX-3	online		00:00	14:00
VAX-4	offline	not in use	00:00	14:00
VAX-1	fault	unknown	10:00	10:15
VAX-2	offline	maintenance	10:00	11:10
VAX-1	online		10:15	18:00
VAX-2	online		11:10	12:40
VAX-2	fault	disk error	12:40	13:10
VAX-2	online		13:10	18:00
VAX-3	fault	cable error	14:00	14:10
VAX-4	online		14:00	∞
VAX-3	online		14:10	14:15
VAX-3	fault	cable error	14:15	14:25
VAX-3	online		14:25	14:35
VAX-3	fault	cable error	14:35	∞
VAX-1	fault	unknown	18:00	18:10
VAX-2	offline	maintenance	18:00	18:25
VAX-1	online		18:10	∞
VAX-2	online		18:25	∞

观察上述数据可知, 当 VAX-2 因维护需要而停用时, VAX-1 原因不明地出现故障。注意, 这并未建立其间的因果联系, 而只是提供了一种可供进一步探究的可能联系。运用 Allen 的时间间隔-间隔约束方法, 可将上述可能联系形式化地记为:

Log. (Computer, Status)[VAX-1, fault]-starts-
Log. (Computer, Reason)[VAX-2, maintenance]

该式可解释为: Computer = 'VAX-1' 且 Status = 'fault' 的全部元组与 Computer = 'VAX-2' 且 Reason = 'maintenance' 的元组具有相同的起始时间。

3.1 时间间隔与事件之间的关系

时态关系用于描述时态关系数据库中的两个元组在时间上的联系。各种时态关系及其含义如表1所示。该表最后一列表示间隔-间隔, 点-间隔, 点-点关

系,这些关系又是通过如下点之间关系表达的:(1) a.start 与 b.start;(2)a.start 与 b.end;(3)a.end 与 b.start;(4)a.end 与 b.end.

表1

关 系	逆 关 系	条 件
a before b	b after a	a.end < b.start
a meets b	b met by a	a.end = b.start
a overlaps b	b overlapped by a	a.start < b.start < a.end < b.end
a equals b		a.start = b.start, a.end = b.end
a starts b	b started by a	a.start = b.start, a.end < b.end
a finished by b	b finishes a	a.start < b.start, a.end = b.end
a contains b	b during a	a.start < b.start, a.end < b.end
a begins b	b begun by a	a.start = a.end = b.start
a ended by b	b ends a	a.start = a.end = b.end

3.2 时态数据的归纳模式

第2节描述的静态归纳依赖关系也如下刻画:对 X 中的属性的所有实例 x 和 Y 中的属性的所有实例 y 有:

$$\forall x, \forall y: X \rightarrow \neg Y$$

类似地,静态值敏感归纳依赖关系亦可如下描述:对 X 中的属性的实例 x' 的某个子集和 Y 中的属性的实例 y' 的某个子集有:

$$\forall x', \forall y': X \rightarrow \neg Y$$

如要从随时间变化而变化的数据中得出相应的结论,仅仅简单地增加一个时间因素是不够的,设有布尔函数 $\Phi(\theta, x, y)$, θ 为时态关系,如果时态关系 θ 通过两个元组的起点与终点时间描述了两者的关系,那么布尔函数 $\Phi(\theta, x, y)$ 为真。借此,我们便可以研究对归纳依赖关系施加时态约束后所产生的作用和影响。例如,我们考察下列情形:

(1) Y 中属性的时间与值的所有变化均可从 X 的任何变化推知,即

$$\forall x, \forall y: X \rightarrow \neg Y \wedge \Phi(\theta, x, y)$$

这种约束太强了,几乎没有什么适用场合。即使有,也是十分明显的关系,不必劳神去发现。于是,我们如下弱化约束:

(2) Y 中所有变化的时间和值可从某个 X 的变化推出,即 X 解释 Y 中的变化:

$$\exists x, \forall y: X \rightarrow \neg Y \wedge \Phi(\theta, x, y)$$

(3) X 中的所有变化至少蕴涵了 Y 中的其时间与值可导出的相应变化,即 X 的变化以 Y 的变化为条件:

$$\forall x, \exists y: X \rightarrow \neg Y \wedge \Phi(\theta, x, y)$$

(4) Y 中的某些变化的时间和值可以从 X 推知,即 X 也许能解释 Y 中的某个变化:

$$\exists x, \forall y: X \rightarrow \neg Y \wedge \Phi(\theta, x, y)$$

注意,如果 $X \rightarrow \neg Y$ 表示的是一一对应关系,我们便可逆转该依赖关系,即有 $Y \rightarrow \neg X$ 。此时,情况(2)与情况(3)除了时态关系 θ 是互逆的,两者可以认为是等价的。

较之第一种情况,后三种情况所表达的因果关系略为宽松一些,但在现实世界数据库中,其约束仍然较强,不易满足,因而实用性不够。于是,我们进一步放宽归纳依赖的约束。如认定只有那些发生在特定时间的变化才是相关的,即推理的有效性时间与时间有关。

(5) Y 的所有值均可从 X 推出,只要它们在 X 决定的时间变化,即如果变化发生在指定时间, X 就能决定 Y 的值:

$$\forall x, \forall y, \Phi(\theta, x, y): X \rightarrow \neg Y$$

仿情形(2)、(3)、(4)对情形(1)的弱化方法,我们对情形(5)作相应的约束弱化。

(6) Y 中所有变化的值可从某个 X 的变化 x 推出,只要它们在相应 x 确定的时间变化,即 Y 中的变化由某个 X 中的变化解释:

$$\forall y, \exists x, \Phi(\theta, x, y): X \rightarrow \neg Y$$

(7) X 中的所有变化至少蕴涵了 Y 中的一个其值可导出的相应变化 y, 只要该 y 在 X 确定的时间变化,即 X 的变化以 Y 的变化为条件:

$$\forall x, \exists y, \Phi(\theta, x, y): X \rightarrow \neg Y$$

(8) Y 中的某些变化的时间和值可以从 X 推知,而且它们还要满足相应的时态关系,即 X 也许能解释 Y 中的某个变化:

$$\exists x, \exists y, \Phi(\theta, x, y): X \rightarrow \neg Y$$

上述例子显然未能穷尽所有可能情况,但已可以说明从数据库中自动推出与时间有关的关系是相

当复杂的,同时,我们也可以看到三种形式的约束弱化方法:(1)原因从全称量词到存在量词;(2)结果从全称量词到存在量词;(3)约束从与时间无关到与时间有关。

下面以情形(5)为基准,提出时态归纳依赖关系,然后,再提出若干弱化形式。

3.3 时态归纳依赖关系

现在,我们通过扩展普通关系模式的方法来获得所需的时态关系模式,在普通关系模式的属性集上增加两个属性:S和E,用以表征某一元组在某段时间的有效性。令H为一时态关系模式,h为定义在H上的关系,对H中候选码属性的任意子集X,在任意时间t, $\sigma_{X=x}(0 \leq t \leq E(h))$ 最多只有一个元组,这就是说,不允许主码相同的元组有重叠的时间段。

令Q为重复合H中每个属性而形成的关系模式,q为定义在Q上的一个关系,其中

$$q = \sigma_{A \in D_A, h \in D_h, A \in D_A, h \in D_h} (h \times h')$$

h为定义在H上的一个关系,除了相应属性A变为A',h'与h相同。 D'_A 为X的值域 D_A 的子集, D'_Y 为Y'的值域 D_Y 的子集。

定义5 关系h在属性子集X和Y之间存在一时态归纳依赖关系 $X[D'_A] \rightarrow Y'[D'_Y]$,如果属性子集X和Y之间的值敏感归纳依赖关系 $X[D'_A] \rightarrow Y'[D'_Y]$ 在关系q中成立。

说明:(1)如果 $D'_A = D_A$,即对X的值域没有限制,那么称上述时态归纳依赖关系为完全时态归纳依赖关系(FTID);(2)如果 $\emptyset$ 包含了较多的时态关

系,那么时态归纳依赖关系的利用便较不易,此时可对 $\emptyset$ 中的关系进行抽象,形成更高层的时态关系。

上述时态归纳依赖关系可对原因和结果分别进行如下弱化:

对原因进行弱化 原因弱化是指原因的变化没有引起相应的结果发生变化,即X值的变化不影响Y的值。

定义6 在属性子集X和Y($S_1, E_1 \in X, Y$)之间存在一原因弱化时态归纳依赖关系 $X \rightarrow Y[D'_Y]$,如果存在一Y值域 D_Y 的子集 D'_Y ,时态归纳依赖关系 $X \rightarrow Y'[D'_Y]$ 在关系q中对部分X成立。

对结果进行弱化 结果弱化是指结果的变化独立于任何指定的原因,即Y值的变化不影响X的值。

定义7 在属性子集X和Y($S_1, E_1 \in X, Y$)之间存在一原因弱化时态归纳依赖关系 $X[D'_A] \rightarrow Y$,如果存在一X值域 D_A 的子集 D'_A ,时态归纳依赖关系 $X[D'_A] \rightarrow Y$ 在关系q中对部分Y成立。

3.4 时态归纳依赖关系实例

现在,回到前述实例数据库,考察如下时态归纳依赖关系是否成立:

$$\text{Log(Computer, Status)[VAX-1, fault]} \rightarrow \text{Log(Computer, Reason)[VAX-2, maintenance]} \quad (15)$$

属性子集X, Y分别为{Computer, Status}, {Computer, Reason},其值域子集 D'_A, D'_Y 分别为{VAX-1, fault}, {VAX-2, maintenance}。q = $\sigma_{A \in D'_A, h \in D'_Y, A \in D_A, h \in D_h} (h \times h')$,即如下表:

computer	Status	Computer	Reason	start	End	start'	End
VAX-1	fault	VAX-2	maintenance	10:00	10:15	10:00	11:10
VAX-1	fault	VAX-2	maintenance	18:00	18:10	18:00	18:25

按照定义5可知,式(15)所述的时态归纳依赖关系成立。根据定义,同样可以推得下列时态归纳依赖关系:

$$\text{Log(Computer, Status)[VAX-4, online]} \rightarrow \text{Log(Computer, Reason)[VAX-3, failure]} \quad (16)$$

$$\text{Log(Computer, Status)[VAX-4, online]} \rightarrow \text{Log(Computer, Reason)[VAX-3, failure]} \quad (17)$$

注意:我们考察的对象是元组对,这就将问题的复杂性大为降低。如果考察三个元组之间的关系,问题的复杂性就会显著增加。

结束语 本文刻画了在关系数据库中进行数据归纳的一些问题,给出了一系列关于静态和动态归纳依赖关系的定义。归纳数据模型与普通关系数据模型的整合对KDD的研究与应用具有重要意义。

参考文献

- [1] M. Ke, M. Ali, Induction in Database Systems, J. Appl. Intell., 1(3)1991
- [2] W. J. Frawley et al., Knowledge Discovery in Databases, An Overview, AI Magazine, 13(3)
- [3] J. Han et al., Discovery of Quantitative Rules from Large Databases, Proc. Fifth Int'l Symp. Methodologies for Intelligent Systems
- [4] J. F. Roddick, J. D. Patrick, Temporal Semantics in Information Systems-A Survey, Information Systems, 17(3)1992
- [5] J. F. Roddick et al., Handling Discovered Structure in Database Systems, IEEE Trans. on Knowledge and Data Engineering, 8(2)1996