Vague数据库Skyline查询技术研究

赵法信 金义富

(岭南师范学院信息科学与技术学院 湛江 524048)

摘 要 Skyline查询处理是近年来数据库领域的一个热门研究方向。由于现实世界中普遍存在着大量不精确、不确定的信息,Skyline查询也随之成为模糊数据处理中的一个重要内容。在已有研究的基础上,讨论了基于Vague关系数据模型的Skyline查询,其用于查询给定Vague关系中的任意元组确定不被该关系中的任意其它元组所支配的程度,并给出了相关的计算公式和查询算法,该算法可直接作用于Vague关系数据库,而无需对Vague关系数据库对应的所有可能性状态逐一进行扫描,具有较高的执行效率。在此基础上,还进一步讨论了带有预选择条件的Skyline查询的计算方法。

关键词 Vague集,Vague关系数据模型,Skyline,查询

中图法分类号 TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.9.048

Study on Skyline Query for Vague Database

ZHAO Fa-xin JIN Yi-fu

(School of Information Science and Technology, Lingnan Normal University, Zhanjiang 524048, China)

Abstract Skyline query processing has recently received a lot of attention in the field of database. Due to a lot of information is often imprecise and uncertain in the real world, Skyline queries have become an important content of fuzzy data processing. Based on the existing research, Skyline query processing based on the Vague relational data model was discussed. In this framework, Skyline queries aim at computing the extent to which any tuple of a given relation is not dominated by any other tuples of the same relation. And the corresponding query formula and query algorithm were given. The key for efficiency lies in the fact that the algorithm does not require to make computations explicitly over all the possible worlds, but works directly on the Vague relational databases. On the basis, processing method of Skyline query with preselection condition was discussed.

Keywords Vague set, Vague relational data model, Skyline, Query

1 引言

不精确、不确定的信息和数据普遍存在于现实世界中。为了能够在数据库中处理这些具有模糊性的信息和数据,众多的研究已经致力于用模糊集理论[1]扩展关系数据库模型。这类包含不精确属性值的数据库被称为模糊数据库[2]。同时,针对模糊数据库的数据处理方法研究也引起了许多学者的注意[3]。

为了解决模糊集理论不能同时表示支持和反对的证据的问题,Gau和Buehrer提出了Vague集理论[4],作为模糊集的进一步扩展,Vague集具有更好的表达数据模糊性的能力[5],并在模糊控制、决策分析及专家系统等领域取得了较模糊集理论更好的效果[6,7]。但相对模糊集理论而言,有关Vague集的研究还处于起步阶段,特别是将Vague集应用于数据库领域方面的研究则更少[8,9]。

Skyline查询[10]也称为Pareto,是一种典型的多目标优化问题,已成功应用于市场分析、决策制定、数据库可视化、数据挖掘等领域。2001年,Borzsonyi等人[10]首次将Skyline查询引入到数据库领域,主要用于从数据集中提取不被其它任何数据点支配的数据点集合。近年来,模糊Skyline查询作为当前模糊数据查询研究的一个重要方面,已成为数据库和网络计算等领域的一个研究热点。王意洁等人[11]综述了现有的各种不确定数据集上的集中式和分布式Skyline查询方法,分析了各种算法的原理和优缺点;Pei等人[12]研究了面向概率数据的Skyline查询,并给出了一种概率Skyline模型来解决对象出现概率不确定的情况。Khalefa等人[13]研究了基于不完备数据的Skyline查询,并设计了一个专门用于不完备数据的"ISkyline"算法。Pivert等人[14]也在给定的不确定数据库模型下,研究了基于该模型的Skyline查询。

本文研究基于Vague关系数据模型的Skyline查询技术,用于查询给定Vague关系中的任意元组可能/确信不被该关系中的任意其它元组所支配的程度,并以此查询语义为基础,讨论了基于该模型的Skyline查询计算方法,同时为避免引发模糊数据库所对应的可能性状态数量的"组合爆炸"问

到稿日期:2015-01-03 返修日期:2015-02-18 本文受广东省自然科学基金项目(S2012010010438)资助。

赵法信(1974-),男,博士,高级工程师,主要研究方向为智能数据处理、数据挖掘,E-mail:zfx0405@163.com;金义富(1969-),男,博士,教授,主要研究方向为智能信息处理与数据挖掘。

题，给出了一个可直接作用于整个 Vague 关系数据库的 Skyline 查询计算算法。在此基础上，还进一步讨论了带有 where 子句的 Skyline 查询的计算方法。

2 Vague 关系数据模型

定义 1(Vague 集[4]) 设 U 为论域，其中的任意一个元素用 u 表示。U 中的一个 Vague 集 V 用一个真隶属函数 t_V 和一个假隶属函数 f_V 表示，$t_V(u)$是从支持 u 的证据中所导出的 u 的隶属度下界，$f_V(u)$则是从反对 u 的证据中所导出的 u 的否定隶属度下界，$t_V(u)$和 $f_V(u)$都将区间[0,1]中的一个实数与 U 中的每一个元素联系起来，即

$t_V: U \to [0,1]$ $f_V: U \to [0,1]$

其中，$t_V(u)+f_V(u) \leqslant 1$。

假设 $U=\{u_1,u_2,\cdots,u_n\}$，那么论域 U 中的一个 Vague 集 V 可以表示如下：

$$V=\sum_{i=1}^{n}[t_V(u_i),1-f_V(u_i)]/u_i, \forall u_i \in U$$

其中，$t_V(u) \leqslant \mu_V(u) \leqslant 1-f_V(u)$且 $1 \leqslant i \leqslant n$。

这里 u 的隶属度$[t_V(u),1-f_V(u)]$是区间[0,1]的一个子区间。也就是说，尽管可能并不知道 u 的精确隶属度，但它落在某个子区间却是确定的。

对于 Vague 值$[t_V(u),1-f_V(u)]/u$ 来说，对象 u 的 Vague 值是一个区间$[t_V(u),1-f_V(u)]$。例如，如果$[t_V(u),1-f_V(u)]=[0.6,0.9]$，则可得出 $t_V(u)=0.6$，$1-f_V(u)=0.9$ 和 $f_V(u)=0.1$。这可以解释为“对象 u 隶属于 Vague 集 V 的程度是 0.6，对象 u 不隶属于 Vague 集 V 的程度是 0.1”。

下面给出部分 Vague 集的基本定义：

定义 2 设 Vague 集 A 和 Vague 集 B 的并集是 Vague 集 C，记作 $C=A \cup B$，其中 C 的真、假隶属度函数分别为：

$t_C=\max(t_A,t_B)$

$1-f_C=\max(1-f_A,1-f_B)=1-\min(f_A,f_B)$

定义 3 设 Vague 集 A 和 Vague 集 B 的交集是 Vague 集 C，记作 $C=A \cap B$，其中 C 的真、假隶属度函数分别为：

$t_C=\min(t_A,t_B)$

$1-f_C=\min(1-f_A,1-f_B)=1-\max(f_A,f_B)$

定义 4 Vague 集 V 的补集 V' 定义为：

$t_{V'}=f_V, 1-f_{V'}=1-t_V$

定义 5(Vague 关系数据模型[8]) 设 $A_i(1 \leqslant i \leqslant m)$是分别定义在论域 U_i 上的属性。那么定义在关系模式 $R(A_1,A_2,\cdots,A_m)$上的 Vague 关系 r 可视为这些属性域笛卡尔积的 Vague 子集，即：

$r \subseteq V(U_1) \times V(U_2) \times \cdots \times V(U_m)$

其中，$V(U_i)$表示论域 U_i 上所有 Vague 子集的集合。经典数据库对应于现实世界的一种状态，但对于 Vague 数据库而言，由于其所含信息的模糊性，它可能对应于现实世界的多种状态。表 1 所列的 Vague 关系 Person 可解释为 16(2×2×2×2)种可能性状态，如果在表 1 的第一条元组中取工作年限为 10；第二条元组取工作年限为 8，取职称为副教授；第三条元组取工作年限为 6，那么由此组成的一个可能性状态所对应的可能度为$[0.6,0.8] \wedge [0.5,0.6] \wedge [0.7,0.8] \wedge [0.4,0.6]=[0.4,0.6]$。

表 1 Vague 关系 Person

姓名	工作年限	职称
A	[0.8,0.9]/9+[0.6,0.8]/10	副教授
B	[0.5,0.6]/8+[0.7,0.8]/9	[0.7,0.8]/副教授+[0.4,0.5]/讲师
C	[0.4,0.6]/6+[0.7,0.9]/5	讲师

3 Skyline 查询

数据库中传统的 Skyline 查询是以 Pareto 方法[10]为基础的。下面给出传统 Skyline 查询的定义。

定义 6(Skyline) 设 $P=\{P_1,P_2,\cdots,P_n\}$是数据库的原子查询偏好集合，用 $t >_{P_i} s$ 表示“元组 t 满足偏好 P_i 的程度优于元组 s”，用 $t \geqslant_{P_i} s$ 表示“元组 t 满足偏好 P_i 的程度不比元组 s 差”。根据 Skyline 查询的一般原则，一个元组 t 支配另一个元组 s 当且仅当 t 和 s 满足以下条件：

$\forall i \in \{1,\cdots,n\}, t \geqslant_{P_i} s$，并且 $\exists k \in \{1,\cdots,n\}, t >_{P_k} s$

也就是说，如果元组 t 满足每一个查询偏好的程度都不小于 s 满足相应查询偏好的程度，并且至少有一个偏好，使得 t 满足该偏好的程度大于 s 满足该偏好的程度，就有 t 支配 s 成立。下面用文献[15]中的偏好 SQL 语言来说明以上 Skyline 查询语义。

例 1 给定一手机数据库，其中包含数据表"*phone*"，其关系模式为(品牌，尺寸，颜色，价格)，如表 2 所列。

表 2 关系 phone

ID	品牌	尺寸	颜色	价格
1	苹果	5.5	白色	6000
2	三星	5	白色	2200
3	小米	4.5	白色	1490
4	华为	4	黑色	1100
5	HTC	5	白色	2880
6	三星	4	白色	1500
7	联想	5	灰色	1800

考虑以下查询：

select * from phone where 价格<3000

preferring (品牌='小米' else 品牌='华为' else 品牌='三星') and (颜色='黑色' else 颜色='白色')；

查询结果保留不被 preferring 子句支配的元组，这里第 2、5、6、7 条元组因为被第 3、4 条元组所支配，所以不是所需要的查询结果；而第 3、4 条元组之间不存在相互支配关系，因此最终的查询结果是第 3 条元组和第 4 条元组，如表 3 所列。

表 3 Skyline 查询结果关系 Result1

ID	品牌	尺寸	颜色	价格
3	小米	4.5	白色	1490
4	华为	4	黑色	1100

4 基于 Vague 关系数据模型的 Skyline 查询

4.1 Vague Skyline 查询

由于 Vague 数据库中含有大量不精确、不确定的信息，无法将传统 Skyline 查询的定义直接应用于 Vague 数据库，因此需要根据 Vague 数据库的特点，对传统的 Skyline 查询进行扩展，研究相应的 Vague Skyline 查询定义及处理方法。

以传统 Skyline 查询的定义为基础，以“查询给定 Vague 关系中的任意元组确定不被该关系中的任意其它元组所支配的程度"为查询语义，给出 Vague Skyline 查询的定义如下。

定义 7(Vague Skyline) 设将要查询的 Vague 关系模式为$(A_1,\cdots,A_n)$,$A_{p1},\cdots,A_{pk}$是查询中偏好所涉及的属性。R是 Skyline 查询结果集。根据定义 6 中 Skyline 查询的定义,将元组 t 被另一个元组 s 所支配记为$t\prec s$,此时有:

$\forall i\in\{1,\cdots,k\}, s.A_{pi}\geqslant_{A_{pi}} t.A_{pi}$,并且$\exists j\in\{1,\cdots,k\}$, $s.A_{pi}\geqslant_{A_{pi}} t.A_{pi}$

同时,将元组 t 不被另一个元组 s 所支配记为 $t\not\prec s$,此时有$\neg(t\prec s)$成立。

在模糊查询中,由于查询结果的不唯一性,一般采用"可能度"P表示查询结果元组 t 满足查询条件的程度,用"必要度"N表示确信查询结果元组 t 满足查询条件的程度,且有$N(t)=1-NP(t)$,其中 NP 表示查询结果元组 t 不满足查询条件的程度,当 $N(t)>0$ 时,有 $P(t)=1$,详情请参阅文献[8]。当查询结果较多时,用户希望查询结果能够以满足查询条件的程度进行排序,由 P 和 N 的语义可知,采用将查询结果以 $N(t)$和 $P(t)$为第一关键字和第二关键字进行排序的方法即能够很好地满足用户的排序要求,本文也将采用这种方法。

根据 Vague Skyline 的定义,本文用 $Poss(t)$表示"结果关系 R 中元组 t 不被 R 中的其它元组 s 所支配的可能度",用 $NPoss(t)$表示"结果关系 R 中元组 t 被任意其它元组 s 支配的可能度",用 $N(t)$表示"确信元组 t 不被结果关系 R 中的任意其它元组 s 所支配的程度",来讨论 Vague Skyline 查询的具体计算方法。

(1)可能度 $Poss(t)$的计算方法

对于 t 的每个可能性状态 π_i/t_i,计算每一个元组 $s(s\neq t)$ 中存在的可能性状态 s_j 不支配 t_i 的可能度。最后的可能度 $Poss(t)$是 t 的所有可能性状态所对应的可能度 $Poss(\pi_i/t_i)$的最大值。即

$$Poss(t)=\max_{\pi_i/t_i\in rep(t)} Poss(\pi_i/t_i) \tag{1}$$

其中,$rep(t)$表示元组 t 的所有可能性状态的集合,并且有

$Poss(\pi_i/t_i)=\min(\pi_i,\min_{s\in R\setminus\{t\}} Poss(t_i\not\prec s))$

当$\{\pi_j/s_j\in rep(s)\mid t_i\not\prec s_j\}=\emptyset$时,有:

$Poss(t_i\not\prec s_j)=0$

否则,有:

$$Poss(t_i\not\prec s_j)=\max_{\pi_j/s_j\in rep(s)\mid t_i\not\prec s_j}\pi_j$$

(2)可能度 $NPoss(t)$的计算方法

根据 $Poss(t)$的计算方法,同样也可得到 $NPoss(t)$的计算公式:

$$NPoss(t)=\max_{\pi_i/t_i\in rep(t)} NPoss(\pi_i/t_i) \tag{2}$$

其中

$NPoss(\pi_i/t_i)=\min(\pi_i,\max_{s\in R\setminus\{t\}} Poss(t_i\prec s))$

当$\{\pi_j/s_j\in rep(s)\mid t_i\prec s_j\}=\emptyset$时,有:

$Poss(t_i\prec s)=0$

否则,有:

$$Poss(t_i\prec s)=\max_{\pi_j/s_j\in rep(s)\mid t_i\prec s_j}\pi_j$$

例 2 给定关系模式(品牌,颜色),偏好(小米>华为>三星),(黑色>白色>其它颜色),有以下元组:

$t_1=$〈{[1,1]/三星+[0.7,0.9]/小米},白色〉

$t_2=$〈华为,{[1,1]/黑色+[0.6,0.8]/灰色}〉

$t_3=$〈{[1,1]/小米,[0.5,0.6]/三星},白色〉

先计算 $Poss(t_1)$,元组 t_1 所对应的可能性状态有:

$t_{11}=$[1,1]/〈三星,白色〉

$t_{12}=$[0.7,0.9]/〈小米,白色〉

对于 t_1 的第一个可能性状态 t_{11},有 $Poss(t_{11}\not\prec t_2)=$[0.6,0.8](对应于 t_2 的可能性状态〈华为,灰色〉),$Poss(t_{11}\not\prec t_3)=$[0.5,0.6](对应于 t_3 的可能性状态〈三星,白色〉)。

对于 t_1 的第二个可能性状态 t_{12},有 $Poss(t_{12}\not\prec t_2)=$[1,1](对应于 t_2 的可能性状态〈华为,黑色〉),该可能性状态是完全可能的并且不支配 t_{12}),$Poss(t_{12}\not\prec t_3)=$[1,1](对应于 t_3 的可能性状态〈小米,白色〉),因而最后可得:

$Poss(t_1)=\max(\min([1,1],\min([0.6,0.8],[0.5,0.6])),\min([0.7,0.9],\min([1,1],[1,1])))=[0.7,0.9]$

(3)必要度 $N(t)$的计算方法

结果关系 R 中的元组 t 不被 R 中的其它元组 s 支配的必要度等于 1 减去 t 被 s 支配的可能度,即

$$N(t)=1-NPoss(t) \tag{3}$$

例 3 以例 2 中的数据和查询为例。涉及到 $N(t_1)$的计算,有

$Poss(t_1\prec t_2)=$[1,1],对应于 t_1 中的[1,1]/〈三星,白色〉和 t_2 中的[1,1]/〈华为,黑色〉。

$Poss(t_1\prec t_3)=$[1,1],对应于 t_1 中的[1,1]/〈三星,白色〉和 t_3 中的[1,1]/〈小米,白色〉

最后有:

$N(t_1)=1-\max(Poss(t_1\prec t_2),Poss(t_1\prec t_3))$

$=1-\max([1,1],[1,1])=0$

(4)查询结果元组的排序方法

对于 R 中的每一个元组 t,可根据式(1)和式(3)计算出元组 t 不被 R 中的其它元组 s 所支配的可能度 $Poss(t)$和必要度 $N(t)$。由于 $N(t)$表示确信元组 t 不被该关系中的任意其它元组所支配的程度,因此当 $N(t)>0$ 时,有 $Poss(t)=1$ 成立。

计算出结果元组 t 的 $N(t)$和 $Poss(t)$后,即可使用下面的方法对查询结果进行排序:

1)首先根据 $N(t)$值进行降序排序;

2)对于 $N(t)=0$ 的元组,再根据 $Poss(t)$的值进行降序排序。

例 4 给定 Vague 关系 r,其关系模式为 $R(A,B,C,D)$,如表 4 所列。这个数据库来自于不同的数据库融合或可信度较低的数据源。

表 4 Vague 关系 r

ID	A	B	C	D
1	{[1,1]/a3+[0.2,0.4]/a2}	{[1,1]/b2+[0.5,0.7]/b1}	c1	{[1,1]/90+[0.7,0.9]/130+[0.3,0.5]/145}
2	{[1,1]/a1+[0.3,0.5]/a3}	b2	{[1,1]/c1+[0.7,0.9]/c3}	{[1,1]/80+[0.7,0.9]/90+[0.5,0.7]/125}
3	a2	{[1,1]/b1+[0.6,0.8]/b3}	c1	150
4	{[1,1]/a3+[0.7,0.9]/a1}	b2	{[1,1]/c3+[0.4,0.6]/c2}	{[1,1]/85+[0.7,0.9]/90}

考虑以下查询:

select * from r

preferring (B='b1'else B='b2') and (A='a1' else A='a2' else A='a3') and (C='c1' else C='c2' else C='c3')

根据上节给出的元组 t 不被其它元组 s 所支配的可能度 $Poss(t)$和必要度 $N(t)$的计算公式，可计算出关系 r 中的 4 条元组的可能度和必要度如下：

$Poss(t_1)=[0.5,0.7], N(t_1)=0$

$Poss(t_2)=1, N(t_2)=[0.5,0.7]$

$Poss(t_3)=1, N(t_3)=[0.2,0.4]$

$Poss(t_4)=[0.7,0.9], N(t_4)=0$

由上述排序方法，根据$(N(t), Poss(t))/t$ 的值对 Skyline 查询结果中的元素进行排序，显然有：

$([0.5,0.7],1)/t_2 \succ ([0.2,0.4],1)/t_3 \succ (0,[0.7,0.9])/t_4 \succ (0,[0.5,0.7])/t_1$

具体如表 5 所列。

表 5　Skyline 查询结果关系 *res*

ID	A	B	C	D
2	{[1,1]/a1+[0.3,0.5]/a3}	b2	{[1,1]/c1+[0.7,0.9]/c3}	{[1,1]/80+[0.7,0.9]/90+[0.5,0.7]/125}
3	a2	{[1,1]/b1+[0.6,0.8]/b3}	c1	150
4	{[1,1]/a3+[0.7,0.9]/a1}	b2	{[1,1]/c3+[0.4,0.6]/c2}	{[1,1]/85+[0.7,0.9]/90}
1	{[1,1]/a3+[0.2,0.4]/a2}	{[1,1]/b2+[0.5,0.7]/b1}	c1	{[1,1]/90+[0.7,0.9]/130+[0.3,0.5]/145}

4.2　Vague Skyline 查询相关算法

直接利用式(1)和式(3)计算 Skyline 查询结果元组的 $Poss(t)$和 $N(t)$的算法如下：

输入：Vague 关系 r

输出：结果元组 t 所对应的 Poss(t)和 N(t)

```
Begin
(1)for r 中的每一个元组 t
  1)Poss(t)=0; p0=0;
  2)for t 中的每一个可能性状态 πi/ti
    ①p1=πi; p2=0;
    ②for r 中每一个元组 s≠t
      I)p3=0; p4=0;
      II) for s 中的每一个可能性状态 πj/sj
          if(ti ≺sj)
            p3=max(p3, πj);
          else
            p4=max(p4, πj);
      III)p1=min(p1, p3);
      IV)p2=max(p2, p4);
    ③Poss(t)=max(Poss(t), p1);
    ④p0=max(p0, min(πi, p2);
  3)N(t)=1-p0;
(2) 输出 Skyline 查询结果元组的 Poss(t)和 N(t)
End
```

显然，这个算法的时间复杂度在 $O(n^2)$之内，其中 n 为 r 的基数。尽管在算法执行过程中需要计算所有(t,s)中元组 t 和元组 s 所对应的所有可能性状态，但由于 Vague 数据库中不确定元组所对应的可能性状态是有限的，因此由该计算过程所增加的时间复杂度是有限的。重要的是，算法执行并不需要计算 Vague 关系所对应的所有可能性状态，从而避免了 Vague 数据库所对应的可能性状态数量"组合爆炸"现象的出现，大大提高了算法的执行效率。需要指出的是，本文是将 Skyline 查询作为一个独立的查询来讨论的，其它情况将另文研究。

上述算法还可通过引入一些剪枝条件来进一步提高效率。例如，在计算元组 t 时，当 p_3 和 p_4 皆等于 1 时可中止最里层的循环。如果用户仅关心必要度，当 $Poss(t\prec s)=1$ 时(此时有 $N(t)=0$)，也可以中止元组 t 的计算。

5　基于 where 子句的 Skyline 查询

在偏好 SQL 语言[15]中，查询会涉及到 where 子句，用于选择那些需要计算 Skyline 的元组。在 Vague 数据库环境下，where 子句的选择条件可能会作用于一个含有 Vague 信息的属性。在这种情况下，只要与选择条件相关的属性和与偏好相关的属性之间是相互独立的(这也是本文所讨论内容的前提条件)，就不会对上节中所讨论的 Skyline 查询方法的有效性产生任何影响。但 where 子句中的选择条件会影响计算 Skyline 的过程中与选择结果元组相关的可能度和必要度，这就需要对上节中用于计算可能度的式(1)和计算必要度的式(2)进行修正。

为了清晰起见，这里仅考虑 where 子句选择条件 *Cond* 形式为"$A\ \theta\ v$"时(A 为属性，v 为常量)的情况，其它形式选择条件的处理方法可参见文献[8]，由文献[8]可得到计算满足条件 *Cond* 的元组 t 的可能度 *CPoss* 和必要度 *CN* 的公式如下所示：

$$CPoss(t)=\max_{\pi_k/t.A_k\in rep(t.A)}\min(\pi_k,\mu(t.A_k))$$

$$CN(t)=1-\max_{\pi_k/t.A_k\in rep(t.A)}\min(\pi_k,1-\mu(t.A_k))$$

其中，$rep(t.A)$表示元组 t 中 A 属性所对应的可能性状态的集合。当 $t.A_k\theta\ v$ 成立时，$\mu(t.A_k)=1$，否则 $\mu(t.A_k)=0$。

在筛选出满足 where 子句选择条件的元组并计算出相关的可能度 *CPoss* 和必要度 *CN* 后，即可对式(1)和式(2)进行修正，然后再进一步进行 Skyline 查询的相关计算。

对式(1)进行修正可以得出：

$$Poss(t)=\min(CPoss(t),\ \max_{\pi_i/t_i\in rep(t)}Poss(\pi_i/t_i))$$

其中，$Poss(\pi_i/t_i)=\min(\pi_i,\ \min_{s\in R\setminus\{t\}}Possc(t_i\not\prec s))$，且

$$Possc(t_i\not\prec s))=\max(1-CN(s),\min(CPoss(s),Poss(t_i\not\prec s)))$$

$Poss(t_i\not\prec s)$的定义和 4.1 节中的公式相同。

同样，在当前环境下，也可以对式(2)进行修正以得出元组 t 被其他元组 s 所支配的可能度 $NPoss(t)$，具体如下：

$$NPoss(t)=\max(1-CN(t),\min(CPoss(t),\ \max_{\pi_i/t_i\in rep(t)}NPoss(\pi_i/t_i)))$$

其中

$$NPoss(\pi_i/t_i)=\min(\pi_i,\ \max_{s\in R\setminus\{t\}}\min(CPoss(s),Poss(t_i\not\prec s))$$

$Poss(t_i\prec s)$ 的定义和 4.1 节中的公式相同。

结果关系 R 中的元组 t 不被 R 中的其它元组 s 支配的必要度公式与式(3)相同，即 $N(t)=1-NPoss(t)$。

例 5　仍以表 4 所列的 Vague 关系 r 为例，在例 4 所给的查询基础上增加 where 子句，具体如下：

select * from r where $D\geqslant 120$

preferring (B='b1' else B='b2') and (A='a1' else A='a2' else A='a3') and (C='c1' else C='c2' else C='c3')

(下转第 248 页)

[7] Lin T, Zha H B. Riemannian manifold learning [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2008, 30 (5): 796-809

[8] Luo D J, Ding C, Nie F P, et al. Cauchy graph embedding[C]// Proceedings of ICML2011. 2011: 553-560

[9] Zhang Z Y, Wang J, Zha H Y. Adaptive manifold learning [J]. IEEE Trans. PAMI, 2012, 34(2): 253-265

[10] Qiao H, Zhang P, Wang D, et al. An explicit nonlinear mapping for manifold learning [J]. IEEE Trans. Cybernetics, 2013, 43 (1): 51-63

[11] 刘辉,杨俊安,王一. 基于流形学习的声目标特征提取方法研究[J]. 物理学报, 2011, 60(7): 1-7
Liu H, Yang J A, Wang Y. A novel approach to research on feature extraction of acoustic targets based on manifold learning [J]. Acta Phys. Sin., 2011, 60(7): 1-7

[12] He X, Yan S, Hu Y, et al. Face recognition using Laplacianfaces [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(3): 328-340

[13] 刘利,韦佳,马千里. 基于流形学习的图像检索研究进展[J]. 北京交通大学学报, 2010, 34(5): 164-171
Liu L, Wei J, Ma Q L. State-of-the-art on image retrieval based on manifold learning[J]. Journal of Beijing Jiaotong University, 2010, 34(5): 164-171

[14] De Silva V, Tenenbaum J B. Global versus local methods in nonlinear dimensionality reduction[J]. Advances in Neural Information Processing Systems, 2003, 15: 721-728

[15] De Silva V, Tenenbaum J B. Sparse multidimensional scaling using landmark points[R]. Stanford, CA: Dept. Math., Stanford University, 2004

[16] Garey M R, Johnson D S. Computers and intractability: a guide to the theory of NP-completeness [M]. WH Freeman & Co. New York, NY, USA, 1979

[17] Karp R. Reducibility among combinatorial problems[M]// Complexity of Computer Computations. 1972: 85-103

(上接第 239 页)

由 where 子句的选择条件所得到的预选择结果关系是 $\{(0,[0.7,0.9])/t_1,(0,[0.5,0.7])/t_2,(1,1)/t_3\}$,这里元素的表示形式为$(CN(t_i),CPoss(t_i))/t_i$。在此基础上,再进一步计算 Skyline 查询,结果如下:

$Poss(t_1)=[0.6,0.8], N(t_1)=0$

$Poss(t_2)=[0.5,0.7], N(t_2)=0$

$Poss(t_3)=1, N(t_3)=[0.3,0.5]$

将结果中的元素用$(N(t),Poss(t))/t$表示。排序后的结果为:

$([0.3,0.5],1)/t_3 \succ (0,[0.6,0.8])/t_1 \succ (0,[0.5,0.7])/t_2$

结束语 本文基于 Vague 关系数据模型,讨论了 Vague 数据库 Skyling 查询的处理方法,该方法用于查询给定 Vague 关系中的任意元组确定不被该关系中的任意其它元组所支配的程度,并给出了相关的计算公式和实现算法。该实现算法直接对 Vague 数据库而不是分别对 Vague 数据库所对应的所有可能性状态进行操作,避免了模糊数据库所对应的可能性状态数量"组合爆炸"问题的发生,在很大程度上降低了 Skyling 查询计算过程的复杂性。在此基础上,还进一步给出了带有 where 子句的 Skyline 查询的计算方法。一些用于经典数据库[16]或概率数据库[12]或不完备数据库[13]的 Skyline 查询技术是否可以用于 Vague 数据库框架还需要进一步的研究。另外,在后续的研究中,我们计划以本文的研究工作为基础,采取批量查询的形式,在大规模模拟实例的基础上测试 Vague 数据库 Skyline 查询处理方法的有效性和效率。

参 考 文 献

[1] Zadeh L A. Fuzzy sets[J]. Information and Control, 1965, 8(3): 338-353

[2] Ma Z M, Mili F. Handling fuzzy information in extended possibility-based fuzzy relational databases [J]. International Journal of Intelligent Systems, 2002, 17(10): 925-942

[3] Bosc P, Pivert O. Modeling and Querying Uncertain Relational Databases: a Survey of Approaches Based on the Possible Worlds Semantics [J]. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2010, 18(5): 565-603

[4] Gau W L, Buehrer D J. Vague sets[J]. IEEE Transactions on Systems, Man, and Cybernetics, 1993, 23(2): 610-614

[5] Lu A, Ng W. Vague sets or intuitionist fuzzy sets for handling vague data: which one is better [M]// Conceptual Modeling-Ek 2005. Springer, 2005: 401-416

[6] 郝忠孝,李松. Vague 时间段关系与 Vague 区域关系的表示和复合推理[J]. 计算机学报, 2014, 37(8): 1743-1753
Hao Zhong-Xiao, Li Song. Representation and Compound Reasoning of the Vague Temporal Interval Relations and the Vague Region Relations [J]. Chinese Journal of Computers, 2014, 37 (8): 1743-1753

[7] 欧阳春娟,李斌,李霞,等. 基于 Vague 集相似度量的图像隐写系统安全性测度[J]. 计算机学报, 2012, 35(7): 1510-1521
Ouyang Chun-juan, Li Bin, LI Xia, et al. A New Security Evaluation for Steganographic System Based on Vague Set Similarity Measure[J]. Chinese Journal of Computers, 2012, 35(7): 1510-1521

[8] 赵法信,马宗民,吕艳辉. 基于 Vague 数据库的代数查询语言[J]. 小型微型计算机系统, 2008, 29(10): 1893-1899
Zhao Fa-xin, Ma Zong-min, Lv Yan-hui. Vague Databases Based Algebraic Query Language [J]. Journal of Chinese Computer Systems, 2008, 29(10): 1893-1899

[9] 赵法信,金义富. 基于异构双极信息的模糊查询研究[J]. 计算机科学, 2013, 40(7): 153-156, 181
Zhao Fa-xin, Jin Yi-fu. Study on Fuzzy Query of Heterogeneous Bipolarity information[J]. Computer Science, 2013, 40(7): 153-156, 181

[10] Borzsonyi S, Kossmann D, Stocker K. The skyline operator [C]// Proc of the Int Conf on Data Engineering. Los Alamitos, CA: IEEE Computer Society, 2001: 421-430

[11] 王意洁,李小勇,杨永滔,等. 不确定 Skyline 查询技术研究[J]. 计算机研究与发展, 2012, 49(10): 2045-2053
Wang Yi-jie, Li Xiao-yong, Yang Yong-tao, et al. Reasearch on Uncertain Skyline Query Processing Technique[J]. Journal of Computer Research and Development, 2012, 49(10): 2045-2053

[12] Pei J, Jiang B, Lin X, et al. Probabilistic skylines on uncertain data[C]// Proc. of VLDB 2007. New York: ACM, 2007: 15-26

[13] Khalefa M E, Mokbel M F, Levandoski J J. Skyline query processing for incomplete data[C]// Proc. of ICDE 2008. Piscataway, NJ: IEEE, 2008: 556-565

[14] Pivert O, Prade H. Skyline Queries in an Uncertain Database Model Based on Possibilistic Certainty[C]// SUM 2014. 2014: 280-285

[15] Kießling W, Kostler G. Preference SQL- Design, implementation, experiences[C]// Proc. of VLDB 2002. 2002: 990-1001

[16] Bartolini I, Ciaccia P, Patella M. Efficient sort-based skyline evaluation[J]. ACM Transaction on Database Systems, 2008, 33 (4): 1-49