基于粗糙集的空间信息预处理方法

郭 平 叶 莲 范 丽
(重庆大学计算机学院 重庆400044)

摘 要 空间数据可以通过不同的方法获取,但是获得的原始数据一般不适于直接应用,通常还需要进行预处理加工。本文通过对空间信息预处理一般过程的研究,提出了预处理中的几个关键问题,并将Rough集理论用于空间数据预处理,解决遗漏值补齐、属性离散化等问题,同时给出了相应的算法。

关键词 数据预处理,空间数据处理,Rough集应用,遗漏值补齐,属性离散化

The Preprocess Method of Spatial Information Based on Rough Set

GUO Ping YE Lian FAN Li
(College of Computer Science, Chongqing University, Chongqing 400044)

Abstract The spatial data may be extracted by different methods. But the original data cannot be applied to application directly, it need to be pretreated. This paper gives the research on the common course of pretreatment of spatial information, and presents several pivotal problems in pretreatment. The problem of complementarity of missing value and discrete of attribution can be resolved through the theory of Rough set applied to pretreatment of spatial data, and the according algorithm is presented as well.

Keywords Data pretreatment, Spatial data process, Rough set application, Complementarity of missing vatue, Discrete of attribution

1 引言

空间信息是人们认识自然和改造自然的重要数据,一般可分为空间数据和属性数据。空间数据用来表示空间对象的位置、形态、大小、分布等几何特征方面的信息,是对现实世界中存在的具有定位意义的事物和现象的描述。属性数据则是表示空间对象特征、类属等方面信息的数据。

空间信息预处理是为GIS处理与应用提供高质量空间信息的重要手段与方法。空间信息由于其自身的特点,其预处理方法和技术与一般的数据的处理方法和技术存在差异。除需要进行数据中噪声滤除、缺失值补齐等处理外,还包括属性选择、连续属性离散化、数据理解等处理。另外,由于空间数据库技术不成熟,大量的空间数据还是通过关系数据库来管理的,因此,以合适的方式将空间数据库存储在关系数据库中并进行管理也是空间信息预处理必须考虑的问题。

与空间信息预处理相关的研究已被广泛地进行,例如基于云理论的方法,将非数字型数据用云模型表达建立概念层次结构。用最大方差法数据离散化方法MaxVar与云模型相结合,从数字型数据中产生云模型表达的软离散区间和概念层次结构,以及根据面向对象的类结构生成概念树等等。

本文首先研究了空间信息预处理系统的基本结构,然后讨论了空间信息的逻辑预处理、数据转换等方法,重点研究了以粗糙集理论为基础的决策表补齐和决策表离散化问题,给出了相应的算法,并对算法进行了分析。

2 空间信息预处理系统的结构

空间信息预处理的目的包括两方面:一是将空间信息进行提取并转换成易于存储和管理的数据格式,二是提高空间信息的质量。

2.1 空间信息结构

空间信息可以分为空间数据与属性数据两个方面。属性数据使用关系数据库能够进行较好的管理,空间数据的管理要复杂得多。

根据在计算机系统中对空间数据的存储组织、处理方法的不同,空间数据又分为图形数据和图像数据两种基本形式。图形数据以点、线、面和体来表示空间对象,并以一定的参照系中的坐标来定位空间对象。图像数据以像素为空间对象的基本构成单位,空间对象以点的不同灰度值和不同的颜色值来表示。

在空间数据存储与管理方面,图形数据的组织方式是:点是最基本的数据单位,用以定位一些特定的空间对象,并且是组成其它空间对象的基础;线是点的集合,一般地,考虑以点为端点的线段;面是由线(段)围成的区域,且是封闭的,因此常用一组相互连接的线来表示面;体是由面围成的封闭区域,常用一组面来表示。

在应用中,图形数据常按一定的特征被组织成图层,即将具有同一特征的空间对象组织为一个图层。例如,将城区的所有街道放在一个图层中便于对城区的街道进行管理。专题图是应用中另一种常用的组织图形数据的方法,它将为某个特定主题服务的空间对象组织在一起以方便应用和管理。例如,将城区的街道与公交线路放在一个专题图中,便于对公共交通进行管理与查询。

图像数据的存储与管理的基础是点及其灰度和颜色,因此,点及点间的关系是构成图像数据的基础。

2.2 空间数据的质量

空间信息处理中，空间数据质量的优劣决定着整个系统的质量与相关应用的成败。提高空间数据质量是空间信息预处理的目标之一。一般来说，评价空间数据质量的标准包括：

(1)位置精度。空间对象的坐标数据与实体真实位置的接近程度，表现为空间坐标数据的精度，高程精度，形状再现精度等。

(2)属性精度。实体的属性或特征值与真实值相符的程度，通常取决于数据的类型，且常与位置精度有关。

(3)时间精度。数据的现实性，可以通过数据采集、更新时间和更新强度来表现。

(4)逻辑一致性。数据之间关系上的可靠性，包括数据结构、数据内容、空间标识以及拓扑性质上的内在一致性。

(5)数据完整性。包括数据值范围、数据的分层、实体类型、属性数据和名称等方面的数据完整性。

2.3 空间数据预处理过程

空间数据从获取到应用需要经过多个方面的预处理，可表示为如图1所示的流程。在空间数据的编辑处理中，包括了数据的信号加强和去噪声、空间数据的格式转换、图形坐标变换和数据的结构化表示等。本文中仅讨论逻辑预处理、数据格式转换、缺失值补齐和离散化。

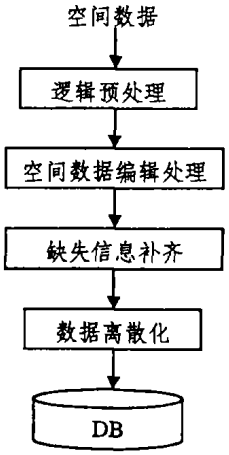


图1 空间数据预处理一般过程

3 空间数据逻辑预处理和结构化表示

3.1 空间数据逻辑预处理

空间数据的逻辑预处理[1]包括对空间数据进行分幅、分层、分专题要素等处理。

(1)*分幅* 对于空间数据，无论是由遥感得到的卫星照片或是由测量所得的地图，首先需要进行分幅处理，以便于数据的存储、检索、显示与分析。分幅的目的在于将原始数据集按一定的规则划分为多个数据集。基本的分幅方法为水平分幅与垂直分幅。水平分幅按平面的定位坐标划分，如图2所示。水平分幅的方式可以是矩形，也可以是按行政区划等。垂直分幅是按高程坐标划分，加入了高程信息，如将某一高程范围内的空间对象分在同一图幅中。

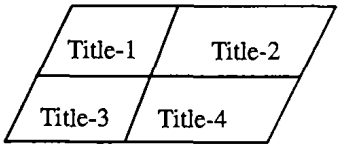


图2 水平分幅

(2)*分层* 按空间数据的内容和某些特征来组织空间数据形成分层，不同的内容属于不同的层。分层的目的是为了更有效组织和管理空间数据，主要分层依据是国家相关信息分类标准。例如可以根据植被、水系和居民地将地图分为三层。

分层一般宜细不宜粗，层与层之间的关联应尽量少，不同层的组合可得到不同的空间数据集。

(3)*分专题要素* 专题要素是指地理实体具有的各种性质，如地形的坡度、坡向、某地的年降雨量、土地酸碱类型、人口密度等等。将空间数据按专题要素来组织，能方便地进行数据管理和应用。如，按降雨量专题要素组织的专题图能方便地对洪涝灾害等的预测和预报提供支持。

逻辑预处理中的分幅、分层和分专题要素一般不是孤立的，通常在分幅的基础上再分层或分专题要素。针对不同的需求可以有多种不同的组合形式。

3.2 空间数据的结构化表示[1]

目前，空间信息的存储和管理主要采用两库模式：属性数据利用关系数据库进行存储，空间数据用各 GIS 软件提供的专用图形/图像数据库进行存储。在 SQL Server 和 Oracle 等关系数据库产品中通过图形/图像数据类型也能存储和管理空间数据。然而，这些存储管理方式都是以图形/图像数据结构来管理空间数据的，是非结构化数据结构，给操作与处理带来了许多的不便。将非结构化的数据格式转换成结构化的数据格式是空间数据预处理中必须考虑和要解决的问题之一。这里，分别讨论将图形、图像格式的数据转换成结构化的数据格式并存入关系数据库的方法。

(1)*图形数据的结构化表示* 对于2维空间的图形数据，它们是由点、线段和面(多边形)组成的，根据图形的构成特征分别构造点关系表、线段关系表和多边形关系表来表示和存储，如图3所示。

点关系表

点编号	名称	坐标	属性
Node1			
Node2			
…			

线段关系表

线段编号	名称	结点编号	属性
Line1		node1,2,…	
Line2		node5,6,…	
…			

多边形关系表

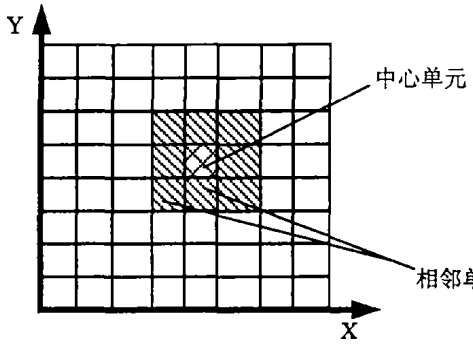
多边形编号	名称	线段编号	属性
Polygon1		Line1,2,3	
Polygon2		Line5,6,7	
…			

图3 图形数据的结构化

这样，就可以用处理属性数据的方法来处理图形数据，将数据的非结构化转换为结构化表示，便于数据的存储与管理。

(2)*图像数据的结构化表示* 对于图像数据，可以采用如下的方法来存储。将要记录的单元作为中心单元，它周围的八个单元为相邻单元。先将各个单元的信息存放在信息表中，每一行的各列依次为 X,Y 坐标值及属性，然后再建立中心单元与相邻单元的关联信息表，每一行的各列依次为 X,Y 坐标值，中心单元的属性值，相邻单元的属性值。图4给出了一个用

这种方式结构化的例子。图4(a)为需要结构化表示的图像，图4(b)为按单元存储的信息表，其中 $a,b,\cdots,d$ 为每个单元的属性，图4(c)为中心单元与相邻单元的关联信息表，其中 $a_1,a_2,\cdots,a_8$ 为与中心单元相邻的8个单元对应的属性 a，其余类似。



(a) 图像、中心单元与相邻单元

X	Y	a	b	…	d
…	…			…	
4	5	2	5	…	7
5	5	1	6	…	8
6	5	2	6	…	8
…	…			…	

(b) 单元信息表

X	Y	a	a_1	a_2	…	a_8	b	b_1	b_2	…	b_8	…	d_8
…	…	…	…	…	…	…	…	…	…	…	…	…	
5	5	1	1	2		8	6	5	3		1	…	
…	…	…	…	…	…	…	…	…	…	…	…	…	

(c) 关联信息表

图4 图像数据结构化表示示例

4 空间数据补齐

存入空间数据库中的空间数据若有遗漏信息，就需要在预处理阶段进行补充，这在 Rough 集理论中称为决策表补齐。

4.1 Mean Completer 算法

根据信息表中其余实例在该属性上的取值的分布情况来对一个遗漏属性值进行估计补充是一种常用的属性补齐方法。

Mean Completer 算法将信息表中的属性分为数值属性和非数值属性来分别处理。如果遗漏的属性值是数值型的，就根据该属性在其他所有实例的取值的平均值来补充该遗漏的属性值；如果遗漏的属性值是非数值型的，就用该属性在其他所有实例上取值最多的值(及出现频率最高的取值)来补充该遗漏的属性值，如图5所示。

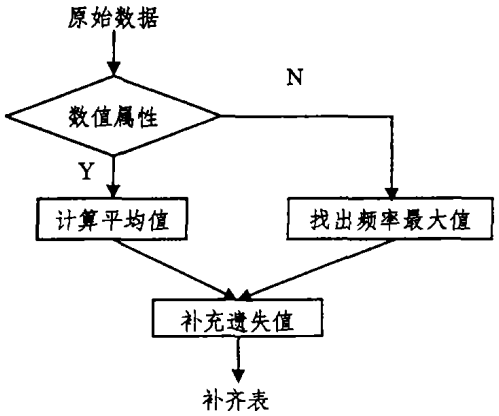


图5 Mean Completer 算法

Mean Completer 算法简单直接，在此基础上，还可以演绎出 Conditioned Mean Completer 等算法。这类算法的基本出发点是以最大概率可能的取值来填补遗漏的属性值。

4.2 基于 Rough 集的不完备数据分析方法(E_ROUSTIDA)

空间数据基本反映了它所涉及领域的基本特征，尽管系统中可能存在遗失的数据。因此，对不完备信息表中的遗失数据值的填补的要求是：应该尽可能反映此信息表中数据的基本特征以及隐含的内在规律。根据 Rough 集理论中数据不可分辨关系来对不完备的数据进行补齐处理能满足这样的要求。

ROUSTIDA 算法是利用 Rough 集理论对遗失数据进行填补的一个有效算法，其基本思想是：遗失数据值的填补应使补齐后的信息表产生的分类规则具有尽可能高的支持度，产生的规则尽量集中，即使具有遗失值的对象与信息系统的其他相似对象的属性值尽可能保持一致，使属性值之间的差异尽可能保持最小。

令信息系统为 $S=\langle U,A\rangle$，$A=\{a_i\mid i=1,\cdots,m\}$ 是属性集，$U=\{x_1,x_2,\cdots,x_n\}$ 是论域，$a_i(x_j)$ 是样本 x_j 在属性 a_i 上的取值。扩充 S 的可辨识矩阵得矩阵 M，称为扩充可辩识矩阵，其定义为：

$$M(i,j)=\{a_k\mid a_k\in A\wedge a_k(x_i)\neq a_k(x_j)\wedge a_k(x_i)\neq *\ \wedge a_k(x_j)\neq *\}$$

其中，$i,j=1,\cdots,n$；“*”表示遗失值。

设 $x_i\in U$，则对象 x_i 的遗失属性集 MAS_i，无差别对象集 NS_i 和 S 的遗失对象集 MOS 分别定义为：

$$MAS_i=\{a_k\mid a_k(x_i)=*,k=1,\cdots,m\}$$

$$NS_i=\{j\mid M(i,j)=\Phi,i\neq j,j=1,\cdots,n\}$$

$$MOS=\{i\mid MAS_i\neq\Phi,i=1,\cdots,n\}$$

基于扩充可辩识矩阵 M 的不完备数据分析算法 ROUSTIDA 如下：

E_ ROUSTIDA // 扩展 ROUSTIDA 算法

输入：不完备信息系统 $S^0=\langle U^0,A,V,f^0\rangle$；
输出：完备的信息系统 $S^r=\langle U^r,A,V,f^r\rangle$；
步骤：
步骤1：计算初始可辨识矩阵 M^0，MAS_i^0，和 MOS^0；令 $r=0$；
步骤2：
A. 对于所有 $i\in MOS^r$，计算 NS_i^r；
B. 产生 S^{r+1}。
(1)对于 $i\notin MOS^r$ 有：$a_k(x_i^{r+1})=a_k(x_i^r)),k=1,2,\cdots,m$；
(2)对于所有 $i\in MOS^r$，对所有 $k\in MAS_i^r$ 作循环；
a)如果 $NS_i^r=1$，设 $j\in NS_i^r$，若 $a_k(x_j^r)=*$，则 $a_k(x_i^{r+1})=*$；
否则 $a_k(x_i^{r+1})=a_k(x_i^r)$；
b)否则，

(i)如存在 j_0 和 $j_1 \in NS_i^r$，满足 $(a_k(x_{j_0}^r) \neq *) \wedge (a_k(x_{j_1}^r) \neq *) \wedge (a_k(x_{j_1}^r) \neq a_k(x_{j_0}^r))$，则 $a_k(x_i^{r+1}) = *$；

(ii)否则，如存在 $j_0 \in NS_i^r$，满足 $(a_k(x_{j_0}^r) \neq *)$，则 $a_k(x_i^{r+1}) = a_k(x_{j_0}^r)$；

(iii)否则，$a_k(x_i^{r+1}) = *$；

C. 如果 $S^{r+1} = S^r$，结束循环转步骤3。

否则，计算 M^{r+1}，MAS^{r+1} 和 MOS^{r+1}；$r = r+1$；转步骤2；

步骤3：如果信息系统还有遗失值，可用 Mean Completer 算法处理；

步骤4：结束。

下面为一简单例子说明上述 E_ROUSTIDA 算法的实施过程。

不完备信息系统 S

U	A1	A2	A3	A4
X1	4	*	1	2
X2	3	1	*	*
X3	*	1	1	*
X4	2	1	4	3
X5	*	1	3	4

中间结果

U	A1	A2	A3	A4
X1	4	1	1	2
X2	3	1	*	4
X3	*	1	1	2
X4	2	1	4	3
X5	3	1	3	4

E_ROUSTIDA 处理结果

U	A1	A2	A3	A4
X1	4	1	1	2
X2	3	1	1	4
X3	3	1	1	2
X4	2	1	4	3
X5	3	1	3	4

5 空间数据的决策表离散化

对连续属性值进行离散化处理是空间数据预处理的重要步骤，离散化后的属性值能方便地存入数据库中便于后续的处理和利用。

5.1 属性区间离散化

通过将属性域划分为区间，离散化技术可以用来减少连续属性值的个数。区间的标号可以替代实际的数据值。许多离散化技术都可以递归使用，以便提供属性值的分层或多分解划分——概念分层。对于给定的数值属性，概念分层定义了该属性的一个离散化。通过收集并用较高层的概念(如 young，middle-age 和 senior)替换较低层的概念(如属性 age 的数字值)，概念分层可以用来归约数据。通过这种数字概化，尽管细节丢失了，但概化的数据更有意义、更容易解释，并且所需的空间比原数据少。在归约的数据上进行挖掘，与在大的、未概化的数据上挖掘相比，所需的 I/O 操作更少，并且更有效。对于同一个属性可以定义多个概念分层，以适合不同用户的需要。

5.2 基于 Rough 集的信息表离散化

利用 Rough 集来进行连续属性离散化可归结为利用选取的断点来对条件属性构成的空间进行划分的问题，划分的原则是：使得每个划分后的区域对应的决策属性值相同。对多个连续属性值的断点选取过程也是合并属性的过程，通过合并属性值，减少属性值的个数，减小问题的复杂度，有利于提高知识获取过程中所得到的规则知识的适应度。常用的划分方法有：

(1)等距离划分算法在每个属性上，根据用户给定的参数来把属性值简单地划分为距离相等的断点段，不考虑每个断点段中属性值个数的多少。

(2)等频率划分算法根据用户给定的参数 k 把 m 个对象分成段，每段中有 m/k 个对象。

(3)最少断点集算法。由于求最少数目的断点集是 NP 完全问题，因此常采用启发式算法。

对信息表 $S = \langle U, C \cup d \rangle$，其中 C 称为条件属性，d 称为决策属性，构造一个信息表 $S^* = \langle U^*, R^* \rangle$ 如下：

$$U^* = \{(x_i, x_j) \in U * U \mid d(x_i) \neq d(x_j)\}$$

$R^* = \{P_r^a \mid a \in C, P_r^a$ 是属性 a 的第 r 个断点 $[c_r^a, c_{r+1}^a]\}$

其中：对于任意 P_r^a，如果 $[c_r^a, c_{r+1}^a] \subseteq [\min(a(x_i), a(x_j)), \max(a(x_i), a(x_j))]$，那 $P_r^a((x_i, x_j)) = 1$；否则，$P_r^a((x_i, x_j)) = 0$。

最少断点集启发式算法：

第1步：根据原来的信息表 S 构造新的信息表 S^*；

第2步：初始化最佳断点集 $CUT = \Phi$；

第3步：选取信息表 S^* 所有列中1的个数最多的断点加入到 CUT 中，去掉此断点所在的列和在此断点上值为1的所有行；

第4步：如果信息表中 S^* 的元素不为空，则转第3步；否则停止。

可知：CUT 即是所求的最少断点集。

(4)属性重要性离散化算法。属性的重要性是建立在属性的分类能力上的，为了衡量条件属性的重要性程度，可从表中删除这一属性，再来考察信息系统的分类会产生怎样的变化：如果去掉某属性会相应地改变分类，则说明该属性重要(改变的程度越大，重要性越高)；反之说明该属性的重要性越低。

基于属性重要性的离散化算法：

第1步：首先根据条件属性的重要性由小到大对条件属性 $V_i (i = 1, \cdots, n)$ 进行排序；在属性重要性相同的情况下，按条件属性断点个数由多到少对条件属性进行排序。

第2步：对每个属性 $V_i \in A$ 进行下面的过程；

第3步：对属性 V_i 中的每一个断点 $C_j (j = 1, \cdots, i_j)$，考虑它的存在性：把信息系统中与 C_j 相邻的两个属性值的较小值改为较大值，如果信息系统不引入冲突，则 $C_{ui} = C_{ui} \setminus \{c_j\}$；否则，把修改过的属性值还原。

这个算法通过对每一个断点进行判定，去掉冗余的断点。从而得到简化的信息系统。这种算法的特点在于由得到的断点集求得的信息系统不会引入冲突。由于断点的判定是从它所属的属性的重要性由小到大的顺序进行的，因此属性重要性较小的属性的断点被淘汰的可能性更大。这里，离散化的过程同时又是属性约简的过程，因为如果一个属性的所有断点都被去掉，则该属性也可以被去掉。

总结 空间信息是人类使用最多、最频繁的信息。RS 和 GPS 技术的发展为空间信息的搜集提供了有效的方法。然而将来自 RS 和 GPS 的信息直接用于空间分析和应用存在许多的困难，信息的预处理成为一个必不可少的环节。本文讨论了空间信息预处理的一般过程，并将 Rough 集理论用于空间数据预处理，为获得完备的、一致的空间信息提供了技术支持，为空间信息的进一步应用奠定了基础。

参 考 文 献

1 李满春，任建武. GIS 设计与实现. 科学出版社，2003
2 张晶，邬伦. 地理信息系统. 电子工业出版社，2002
3 王国胤. Rough 集理论与知识获取. 西安交通大学出版社，2001
4 Han Jiawei，Kamber M. Data Mining Concepts and Techniques. 机械工业出版社，2001
5 邸凯昌，李德仁，李德毅. Rough 集理论及其在 GIS 属性分析和知识发现中的应用. 武汉测绘科技大学学报，1999，24(1)：6～10
6 邸凯昌著. 空间数据挖掘和知识发现. 武汉大学出版社，2001
7 王家耀著. 空间信息系统原理. 科学出版社，2001
8 DeMers M N 著. 武法东，付宗堂，王小牛，等译. 地理信息系统基本原理，电子工业出版社，2001
9 Hinneburg A，Wawryniuk M，Keim D A. HD-Eye：Visual Mining of High-Dimensional Data. IEEE Computer Graphics & Applications Journal，September，1999，19(5)：22～31
10 Ester M，Kriegel H-P，Sander J. Knowledge discovery in spatial databases. In：Advances in Artificial Intelligence，23rd Annual German Conf. on Artificial Intelligence，Bonn，Germany，1999. 61～74