

# 一种基于连接度的空间线对象聚类算法

柳 盛<sup>1</sup> 吉根林<sup>1,2</sup> 李文俊<sup>2</sup>

(南京师范大学虚拟地理环境教育部重点实验室 南京 210046)<sup>1</sup>

(南京师范大学计算机科学与技术学院 南京 210046)<sup>2</sup>

**摘 要** 目前大多数聚类算法主要针对空间点对象且未考虑空间对象的拓扑关系。利用空间线对象相交关系定义了空间线对象连接度,提出一种基于连接度的空间线对象聚类算法 SLCC(Spatial Lines Clustering Algorithm Based on Connectivity)。该算法以 K-means 算法为基础,以空间线对象的连接度作为“距离”进行空间线对象聚类。实验结果表明,SLCC 算法能实现空间线对象的空间聚类,并具有较高的效率。

**关键词** 连接度,空间聚类,拓扑关系,线相交

中图法分类号 TP181 文献标识码 A

## Spatial Lines Clustering Algorithm Based on Connectivity

LIU Sheng<sup>1</sup> JI Gen-lin<sup>1,2</sup> LI Wen-jun<sup>2</sup>

(Key Laboratory of Virtual Geographic Environment of Ministry of Education, Nanjing Normal University, Nanjing 210046, China)<sup>1</sup>

(School of Computer Science and Technology, Nanjing Normal University, Nanjing 210046, China)<sup>2</sup>

**Abstract** At present, most spatial clustering algorithms focus on spatial points without considering spatial topological relations of spatial objects. The spatial line connectivity was defined by line intersection relations. Algorithm SLCC was proposed for clustering spatial lines based on spatial line connectivity. This algorithm, which is based on K-means, selects the spatial line connectivity as the distance between lines to cluster spatial lines. The experiment results show the algorithm is effective and efficient.

**Keywords** Spatial line connectivity, Spatial clustering, Topological relations, Line intersection

## 1 引言

空间聚类是空间数据挖掘的重要组成部分,是聚类研究在空间数据分析中的应用。空间聚类应用广泛,如地理信息系统、生态环境、军事、市场分析等领域。通过空间聚类可以从空间数据集中发现隐含的信息或知识,包括空间实体聚集趋势、分布特征和演变趋势等。如 Andrew 等利用空间聚类分析发现入侵物种分布的变化<sup>[1]</sup>, Wan 等利用空间聚类获得客户群落的划分<sup>[2]</sup>, Tanser 等利用空间聚类统计 HIV 病毒感染的传播情况<sup>[3]</sup>。

目前,国内外已有不少学者对空间聚类问题进行了较为深入的研究,并提出了多种算法。根据空间聚类所采用的思想不同,空间聚类算法主要可归纳为以下几种:(1)基于划分的方法,如 K-means<sup>[4]</sup>、CLARANS<sup>[5]</sup>等算法。(2)基于层次的方法,如 BIRCH<sup>[6]</sup>等算法。(3)基于密度的方法,如 DB-SCAN<sup>[7]</sup>等算法。(4)基于网格的方法,如 STING<sup>[8]</sup>等算法。(5)基于模型的方法,如 EM<sup>[9]</sup>等算法。(6)其它形式的空间聚类算法,如带约束的空间聚类算法及其上述算法的改进算法等。

上述空间聚类算法在空间聚类实际应用中仍然存在一定

的局限性,主要体现在:(1)应用的空间数据类型有限,主要针对空间点对象。而空间数据至少包括点、线、面、体等 4 种基本类型。(2)考虑空间属性不足。空间数据一般具有空间实体的位置、大小、形状、方位及拓扑关系等信息。

国内已有学者基于空间拓扑关系提出了相关的聚类算法<sup>[10-13]</sup>。本文在空间线对象相交的拓扑关系基础上,提出了线对象连接度的概念,在 K-means 算法的基础上,以线对象的连接度作为“距离”进行空间线对象聚类,实现了基于连接度的空间线对象聚类算法(Spatial Lines Clustering Algorithm Based on Connectivity, SLCC)。

本文第 2 节简要介绍了空间线对象连接度的定义及其计算方法;第 3 节是 SLCC 的聚类算法描述;第 4 节是实验结果与分析;最后是对全文的简单总结。

## 2 空间线对象连接度的计算

### 2.1 空间线对象连接度的定义

下面首先介绍点、线、线相交的定义,然后在此基础上介绍连接、连接步长、连接度的定义。

**定义 1**(点) 一个二维空间中的点对象  $P$  由点  $(x, y)$  构成。

到稿日期:2010-09-11 返修日期:2011-01-05 本文受国家自然科学基金(40871176)资助。

柳 盛(1981—),男,博士生,助教,主要研究方向为空间数据挖掘,E-mail:njnuls99@yahoo.com.cn;吉根林(1964—),男,博士,教授,博士生导师,主要研究方向为数据挖掘技术及其应用。

**定义 2(线)** 一个二维空间中的线对象  $L$  是仅由一对点  $P_1$  和  $P_2$  连接而成的线段。

**定义 3(线相交)** 两个不同的线对象  $L_1$  和  $L_2$ , 若存在一个点  $P$  同时属于  $L_1$  和  $L_2$ , 则  $L_1$  和  $L_2$  线相交, 反之  $L_1$  和  $L_2$  非线相交。

**定义 4(直接连接)** 两个不同的线对象  $L_1$  和  $L_2$ , 若  $L_1$  和  $L_2$  线相交, 则  $L_1$  和  $L_2$  是直接连接的。

**定义 5(间接连接)** 两个不同的线对象  $L_1$  和  $L_2$  非线相交, 若存在一个线对象  $L_i$ , 其中  $L_1$  和  $L_i$  线相交,  $L_i$  和  $L_2$  线相交; 或存在一个最短线对象序列  $L_{i1}, \dots, L_m (n \geq 2)$ , 其中  $L_{i1}$  和  $L_1$  线相交,  $L_m$  和  $L_2$  线相交, 任意的  $L_{ij}$  和  $L_{ij+1} (1 \leq j \leq n-1)$  线相交, 则  $L_1$  和  $L_2$  是间接连接的。

**定义 6(连接)** 两个不同的线对象  $L_1$  和  $L_2$ , 若  $L_1$  和  $L_2$  直接连接或间接连接, 则  $L_1$  和  $L_2$  是连接的, 反之  $L_1$  和  $L_2$  是非连接的。

**定义 7(直接连接步长)** 两个不同的线对象  $L_1$  和  $L_2$ , 若  $L_1$  和  $L_2$  是直接连接的, 则  $L_1$  和  $L_2$  直接连接步长为 1。

**定义 8(间接连接步长)** 两个不同的线对象  $L_1$  和  $L_2$  是间接连接的, 若是一个线对象  $L_i$ , 使得  $L_1$  和  $L_2$  间接连接, 则  $L_1$  和  $L_2$  间接连接步长为 2; 若是一个最短线对象序列  $L_{i1}, \dots, L_m (n \geq 2)$ , 使得  $L_1$  和  $L_2$  间接连接, 则  $L_1$  和  $L_2$  间接连接步长为  $n+1$ 。

**定义 9(连接步长)** 直接连接步长和间接连接步长统称为连接步长。

注: 两个不同的线对象  $L_1$  和  $L_2$ , 若  $L_1$  和  $L_2$  是非连接的, 则  $L_1$  和  $L_2$  的连接步长为  $\infty$ 。

**定义 10(直接连接度)** 两个不同的线对象  $L_1$  和  $L_2$ , 若  $L_1$  和  $L_2$  是直接连接的, 则  $L_1$  和  $L_2$  直接连接度为 1。

**定义 11(间接连接度)** 两个不同的线对象  $L_1$  和  $L_2$  是间接连接的, 若  $L_1$  和  $L_2$  间接连接步长为  $n (n \geq 2)$ , 则  $L_1$  和  $L_2$  间接连接度为  $1/n$ 。

**定义 12(连接度)** 线对象  $L$  与其他所有线对象直接连接度或间接连接度之和。

注: 若线对象  $L$  与其他线对象都是非连接的, 则  $L$  的直接连接度和连接度均为 0。

## 2.2 空间线对象连接度的计算

计算空间线对象连接度的步骤为: 首先根据空间线对象之间的相交关系判断算法(算法 1)计算出线对象相交矩阵 Minterseccion(算法 2), 接着根据 Minterseccion 求出线对象连接步长矩阵 Mcsteps(算法 3), 最后由 Mcsteps 获得每个线对象的连接度(算法 4)。

**算法 1 线对象相交判断算法**, 其主要思想是利用混合积分判断线对象是否相交<sup>[14,15]</sup>。

输入: 线对象  $L_1$  和  $L_2$ 。

输出: 线对象  $L_1$  和  $L_2$  是否相交, 若相交则返回 1, 否则返回 0。

方法:

设线对象  $L_1$  的端点坐标为  $P_1(x, y)$  和  $P_2(x, y)$ , 线对象  $L_2$  的端点坐标为  $Q_1(x, y)$  和  $Q_2(x, y)$ 。

第 1 步 计算  $\Delta_1$ 。

$$\Delta_1 = [(P_2(x) - P_1(x)) (Q_1(y) - P_1(y)) - (P_2(y) - P_1(y)) (Q_1(x) - P_1(x))] [(P_2(x) - P_1(x)) (Q_2(y) - P_1(y)) - (P_2(y) - P_1(y)) (Q_2(x) - P_1(x))];$$

第 2 步 计算  $\Delta_2$ 。

$$\Delta_2 = [(Q_2(x) - Q_1(x)) (P_1(y) - Q_1(y)) - (Q_2(y) - Q_1(y)) (P_1(x) - Q_1(x))] [(Q_2(x) - Q_1(x)) (P_2(y) - Q_1(y)) - (Q_2(y) - Q_1(y)) (P_2(x) - Q_1(x))];$$

第 3 步 判断  $\Delta_1$  和  $\Delta_2$  是否均为 0, 若均为 0 则表示线对象  $L_1$  和  $L_2$  相交。若  $L_1$  和  $L_2$  相交则返回 1, 否则返回 0。

### 算法 2 线对象相交关系矩阵求解算法

输入: 空间线对象  $L_1, \dots, L_n (n \geq 2)$ 。

输出: 线对象相交关系矩阵 Minterseccion。

方法:

```
for (i=1; i<=LineNumber; i++)
for (j=1; j<=LineNumber; j++)
//LineNumber 为线对象数目
{
    if (i==j) Minterseccion[i][j]=0;
    //自身不与自身相交
    else 判断  $L_i$  与  $L_j$  是否线相交, 如果  $L_i$  与  $L_j$  相交, 将 Minterseccion[i][j] 值设置为 1, 否则设置为 0。
}
```

### 算法 3 线对象连接步长计算算法

输入: 线对象相交关系矩阵 Minterseccion。

输出: 线对象连接步长矩阵 Mcsteps。

方法:

第 1 步 初始化线对象连接步长矩阵 Mcsteps。

```
for (i=1; i<= LineNumber; i++)
for (j=1; j<= LineNumber; j++)
{
    if (i==j) Mcsteps[i][j]=0; //0 表示线对象自身
    else Mcsteps[i][j]=∞; //∞ 表示不可达
}
```

第 2 步 迭代遍历直接连接线对象获得连接步长。

```
for (i=1; i<= LineNumber; i++)
{
    初始化队列 CQueue;
    初始化 Flags 数组, 将所有值设置为 0。
    //Flags 数组用来标识线对象是否进入过队列, 每个线对象仅有一次入队机会。Flags[i] 设置为 1 时, 表示该线对象  $L_i$  已经进入过队列。
    将 Flags[i] 设置为 1; //  $L_i$  不再入队
    for (j=1; j<= LineNumber; j++) {
        根据 Minterseccion [i][j] 判断  $L_i$  与  $L_j$  是否线相交, 若线相交则将 Mcsteps[i][j] 设置为 1; 将  $L_j$  入队; Flags[j] 设置为 1。
    } // end for
    while (CQueue 非空)
    {
        CQueue 出队;
         $L_k$  为出队线对象;
        若 Mcsteps[i][k] 等于 LineNumber, 则 CQueue 继续出队;
        for (j=1; j<= LineNumber; j++) {
            根据 Minterseccion [k][j] 判断  $L_k$  与  $L_j$  是否线相交, 若相交且  $L_j$  未进入过队列, 则将 Mcsteps[i][j] 设置为 Mcsteps[i][k]+1; 将  $L_j$  入队; Flags[j] 设置为 1。
        } // end for
    } //end while
}
```

### 算法 4 线对象连接度计算算法

输入:线对象连接步长矩阵 Mcsteps。  
 输出:线对象连接度集 ConnectivitySet。  
 方法:根据线对象连接度的定义,由线对象连接矩阵 Mcsteps 获得每个线对象的连接度。  
 for (i=1;i<=LineNumber;i++)  
 {  
   ConnectivitySet[i]=0.0;  
   for (j=1;j<=LineNumber;j++)  
   {  
 if (Mcsteps [i][j]!=0&&. Mcsteps [i][j]!= ∞)  
   {ConnectivitySet[i]+=1.0/ Mcsteps[i][j];}  
   }  
 }

### 3 SLCC 的聚类算法描述

通常 K-means 算法<sup>[4]</sup>的准则函数采用平方误差准则,其定义如下。

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

式中, $E$  是数据集中所有对象的平方误差和, $p$  表示给定的对象, $m_i$  表示簇  $C_i$  的平均值。

SLCC 算法采用 K-means 聚类算法作为基础,以线对象的连接度作为“距离”进行线对象聚类,其准则函数定义如下。

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

式中, $E$  是数据集中所有线对象的连接度误差和, $L$  表示给定的线对象, $Lc$  为线对象  $L$  的连接度, $m_i$  表示簇  $C_i$  所包含线对象连接度的平均值。 $|Lc-m_i|$  表示  $L$  与  $m_i$  的“距离”。

#### 算法5 SLCC 聚类算法

输入:空间线对象  $L_1, \dots, L_n (n \geq 2)$ , 聚类的个数  $k (k \geq 2)$ 。

输出:线对象聚类结果  $C=\{C_1, \dots, C_k\}$ 。

方法:

第1步 计算空间线对象  $L_1, \dots, L_n (n \geq 2)$  连接度获得线对象连接度集 ConnectivitySet;

第2步 根据式(2)的准则函数,运用 K-means 算法进行聚类, $C=K\text{-means\_Clustering}(\text{ConnectivitySet}, k)$ 。

### 4 实验结果与分析

为验证本文提出的基于连接度的空间线对象聚类算法 SLCC 的有效性,我们在 CPU 为 Intel Core2 双核 2.93GHZ,内存 2G 的微机上,将此算法用 Visual C# 加以实现,实验数据为合成数据。空间分布如图 1 所示,线对象分别为  $L_1, L_2, \dots, L_{10}, L_{11}$ 。

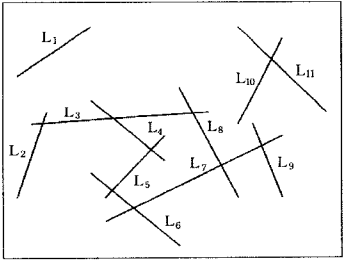


图1 线对象实验数据示意图

首先根据其相交关系计算出相交矩阵,然后根据相交矩阵计算出连接步长矩阵,如图 2 所示。表 1 为线对象  $L_1,$

$L_2, \dots, L_{10}, L_{11}$  的直接连接数和根据连接步长矩阵计算出的连接度。表 2 为根据线对象直接连接数(相交数)聚类的结果,表 3 为根据线对象连接度聚类的结果。用表 2 和表 3 的比较可以看出在衡量线对象之间的连通性时,依据线对象的连接度进行拓扑相交的聚类比仅仅按照线对象直接连接数聚类更加合理。

|   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| ∞ | 0 | 1 | 2 | 3 | 4 | 3 | 2 | 4 | ∞ | ∞ | ∞ |
| ∞ | 1 | 0 | 1 | 2 | 3 | 2 | 1 | 3 | ∞ | ∞ | ∞ |
| ∞ | 2 | 1 | 0 | 1 | 2 | 3 | 2 | 4 | ∞ | ∞ | ∞ |
| ∞ | 3 | 3 | 1 | 0 | 1 | 2 | 3 | 3 | ∞ | ∞ | ∞ |
| ∞ | 4 | 3 | 2 | 1 | 0 | 1 | 2 | 2 | ∞ | ∞ | ∞ |
| ∞ | 3 | 2 | 3 | 2 | 1 | 0 | 1 | 1 | ∞ | ∞ | ∞ |
| ∞ | 2 | 1 | 2 | 3 | 2 | 1 | 0 | 2 | ∞ | ∞ | ∞ |
| ∞ | 4 | 3 | 4 | 3 | 2 | 1 | 2 | 0 | ∞ | ∞ | ∞ |
| ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | 0 | 1 | ∞ |
| ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | 1 | 0 |

图2 线对象连接步长

表1 线对象直接连接数和连接度表

| 线对象名称           | 直接连接数 | 连接度  |
|-----------------|-------|------|
| L <sub>1</sub>  | 0     | 0    |
| L <sub>2</sub>  | 1     | 3.16 |
| L <sub>3</sub>  | 3     | 4.66 |
| L <sub>4</sub>  | 2     | 4.08 |
| L <sub>5</sub>  | 2     | 4    |
| L <sub>6</sub>  | 2     | 4.08 |
| L <sub>7</sub>  | 3     | 4.67 |
| L <sub>8</sub>  | 2     | 3.83 |
| L <sub>9</sub>  | 1     | 3.16 |
| L <sub>10</sub> | 1     | 1    |
| L <sub>11</sub> | 1     | 1    |

表2 基于直接连接数聚类结果

| 类别 | 包含线对象                                                                              |
|----|------------------------------------------------------------------------------------|
| 1  | L <sub>1</sub> , L <sub>10</sub> , L <sub>11</sub>                                 |
| 2  | L <sub>2</sub> , L <sub>8</sub> , L <sub>9</sub>                                   |
| 3  | L <sub>3</sub> , L <sub>4</sub> , L <sub>5</sub> , L <sub>6</sub> , L <sub>7</sub> |

表3 基于连接度聚类结果

| 类别 | 包含线对象                                                               |
|----|---------------------------------------------------------------------|
| 1  | L <sub>1</sub> , L <sub>2</sub> , L <sub>10</sub> , L <sub>11</sub> |
| 2  | L <sub>4</sub> , L <sub>5</sub> , L <sub>6</sub> , L <sub>8</sub>   |
| 3  | L <sub>3</sub> , L <sub>7</sub>                                     |

基于线连接度的空间聚类算法目前尚无文献报道,无法将 SLCC 算法与其他算法进行比较,算法 SLCC 的执行时间性能图如图 3 所示。

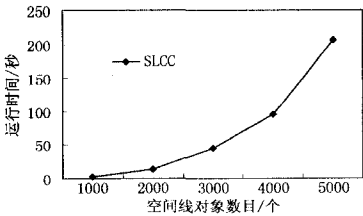


图3 算法 SLCC 运行时间性能

**结束语** 目前已有的空间聚类算法研究对象大多是空间点对象且未涉及空间对象拓扑关系。本文对空间线对象相交  
 (下转第 204 页)

**结束语** 由于 LGSSDR 算法对邻域参数选择比较敏感以及对邻域图边权值设定不够准确,在 LGSSDR 算法的基础上提出一种基于局部重构误差与全局保持的半监督维数约减方法。本算法不仅具有 LGSSDR 算法的优点,而且通过最小化局部重构误差的方法对邻域图边权值的设定更为精确,对邻域参数选择也更加容易,为算法的实际应用带来了很大的方便。

但本算法也存在一些不足之处。例如,算法假定样本空间的邻域是局部线性的,而此假定并不是在所有的样本空间中都成立。因此,如何减少此假定对算法的影响,扩大其适用范围,是需要进一步研究的内容。

## 参考文献

- [1] Shental N, Hertz T, Weinshall D, et al. Adjustment Learning and Relevant Component Analysis[C]// LNCS 2353. London: Springer-Verlag, 2002: 776-790
- [2] Bar-Hillel A, Hertz T, Shental N, et al. Learning a Mahalanobis Metric from Equivalence Constraints[J]. *Journal of Machine Learning Research*, 2005, 6(6): 937-965
- [3] King E P, Ng A Y, Jordan M I, et al. Distance Metric Learning, with Application to Clustering with Side-Information[C]// Advances in Neural Information Processing Systems 15. Cambridge: MIT Press, 2003: 505-512
- [4] Tang W, Zhong S. Pairwise Constraints-guided Dimensionality

Reduction[C]//Proc of the 2006 SIAM DM Workshop on Feature Selection for Data Mining. 2006: 59-66

- [5] Yeung D Y, Chang H. Extending the Relevant Component Analysis Algorithm for Metric Learning Using Both Positive and Negative Equivalence Constraints [J]. *Pattern Recognition*, 2006, 39(5): 1007-1010
- [6] Wu F, Zhou Y L, Zhang C S. Relevant Linear Feature Extraction Using Side-information and Unlabeled Data[C]//Proc of the 17th International Conference on Pattern Recognition. Washington: IEEE Computer Society, 2004: 582-585
- [7] Zhang D Q, Zhou Z H, Chen S C. Semi-supervised Dimensionality Reduction[C]//Proc of the 7th SIAM International Conference on Data Mining. 2007: 629-634
- [8] 韦佳, 彭宏. 基于局部与全局保持的半监督维数约减方法[J]. *软件学报*, 2008, 19(11): 2833-2842
- [9] Roweis S T, Saul L K. Nonlinear Dimensionality Reduction by Locally Linear Embedding[J]. *Science*, 2000, 290(22): 2323-2327
- [10] Georgiades A S, Belhumeur P N, Kriegman D J. From Few to Many: Illumination Cone Models for Face Recognition Under Variable Lighting and Pose[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2001, 23(6): 643-660
- [11] Sim T, Barker S, Bsat M. The CMU Pose, Illumination, and Expression Database[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003, 25(12): 1615-1618

(上接第 181 页)

关系进行了研究,提出了空间线对象连接度的定义,并在此基础上实现了 SLCC 算法。其中间接连接步长(定义 8)和间接连接度(定义 11)在实际应用中要与具体的应用背景相结合,本文未考虑线对象的长度、宽度等因素,也未考虑复杂线对象。SLCC 算法具有一定的应用价值,可以为道路建设的选址、公交线路的设定等提供相关的参考,一般来说良好的道路选址或者公交线路的设定,其道路或线路的连接度都比较高。空间线对象除了自身的属性外,还具有形状、长度、宽度、拓扑关系等空间属性,本文仅仅考虑了线对象相交关系,如何将线对象自身的属性和空间属性综合考虑,还有待深入研究。

## 参考文献

- [1] Andrew A, Thomas C, Korniss G. Ecological invasion: spatial clustering and the critical radius[J]. *Evolutionary Ecology Research*, 2007(9): 375-394
- [2] Wan L, Li Y, Liu W, et al. Application and study of Spatial Cluster and Customer Partitioning[C]//Proc. of the Fourth International Conference on Machine Learning and Cybernetics. Guangzhou, 2005: 1701-1706
- [3] Tanser F, Bärnighausen T, Cooke G S, et al. Localized spatial clustering of HIV infections in a widely disseminated rural South African epidemic [J]. *Int. Journal of Epidemiology*, 2009(4): 1008-1016
- [4] Lloyd S P. Least Squares Quantization in PCM [J]. *IEEE Trans. Information Theory*, 1982(28): 128-137
- [5] Ng A R, Han J. Efficient and effective clustering method for spatial data mining [C]//Proc. of the 1994 Intl. Conf. Very Large Databases. Santiago, Chile, 1994: 144-155
- [6] Zhang T, Ramakrishnan R, Livny M. BIRCH: an efficient data clustering method for very large databases [C]//Proc. of the

1996 Intl. Conf. Management of Data. Montreal, Canada, 1996: 103-114

- [7] Ester M, Kriegel H P, Sander J, et al. A density-based algorithm for discovering clusters in large spatial databases[C]//Proc. of the 1996 Intl. Conf. Knowledge Discovery and Data Mining. Portland, OR, 1996: 226-231
- [8] Wang W, Yang J, Muntz R. STING: A statistical information grid approach to spatial data mining[C]//Proc. of the 1997 Intl. Conf. Very Large Data Bases. Athens, Greece, 1997: 186-195
- [9] Dempster A, Laird N, Rubin D. Maximum likelihood from incomplete data via the EM algorithm [J]. *J. Royal Statistical Society*, 1977(39): 1-38
- [10] Ji G, Miao J, Bao P. A Spatial Clustering Algorithm Based on Spatial Topological Relations for GML Data[C]//Proc. of Intl. Conf. on Artificial Intelligence and Computational Intelligence. Shanghai China: IEEE Computer Society, 2009: 298-301
- [11] Ji G, Miao J, Yang M. A Novel Spatial Clustering Algorithm Based on Spatial Adjacent Relations for GML Data[C]//Proc. of Intl. Workshop on Education Technology and Computer Science. Wuhan, China: IEEE Computer Society, 2009: 278-281
- [12] Ji G, Zhang L. A Spatial Polygon Objects Clustering Algorithm Based on Topological Relations for GML Data[C]//Proc. of Intl. Conference on Information Engineering and Computer Science. Wuhan, China: IEEE Publishing, 2009: 363-366
- [13] Yang N, Ji G. A Spatial Lines Clustering Algorithm Based on Adjacent Relations for GML Data[C]//Proc. of Intl. Conf. on Information Engineering and Computer Science. Wuhan, China: IEEE Publishing, 2009: 3593-3596
- [14] 李永红, 华一新. 一种快速判断线段相交的方法[J]. *测绘通报*, 2003(7): 30-31
- [15] 杨娜, 吉根林. 一种基于相交关系的 GML 空间聚类算法[J]. *广西师范大学学报: 自然科学版*, 2009(3): 113-117