Discovering Fuzzy Spatial Association Rules

Esen Kacar, Nihan Kesim Cicekli

Department of Computer Engineering,
M.E.T.U., Ankara, Turkey
esenkacar@yahoo.com
nihan@ceng.metu.edu.tr

Abstract. Discovering interesting, implicit knowledge and general relationships in geographic information databases is very important to understand and to use these spatial data. One of the methods for discovering this implicit knowledge is mining spatial association rules. A spatial association rule is a rule indicating certain association relationships among a set of spatial and possibly non-spatial predicates. In the mining process, data is organized in a hierarchical manner. However, in real-world applications it may not be possible to construct a crisp structure for this data, rather some fuzzy structures should be used. Fuzziness, i.e. partial belonging of an item to more than one sub-item in the hierarchy, could be applied to the data itself, and also to the hierarchy of spatial relations. This paper shows that, strong association rules can be mined from large spatial databases using fuzzy concept and spatial relation hierarchies.

1 Introduction

With a variety of applications, large amounts of spatial and related non-spatial data are collected and stored in Geographic Information Databases. Spatial Data Mining, (i.e., discovering interesting, implicit knowledge and general relationships in large spatial databases) is an important task for the understanding and the usage of these spatial data. Studies for knowledge discovery in spatial databases can be categorized as: generalization based knowledge discovery, clustering, spatial association rules and image mining. They all have their own techniques for extraction knowledge not explicitly stored in databases.

There has been considerable amount of work on generalization based knowledge discovery, clustering and image mining [4,5,6,7,8]. However relatively little attention has been given to mining spatial association rules [1]. In this paper we also address the problem of mining spatial association rules in geographic information databases.

A spatial association rule, is a rule of the form "$X \rightarrow Y$" where X and Y are sets of predicates where some are spatial, some are non-spatial [1]. Like other kinds of association rules, spatial association rules are also associated with a *minimum support* and a *minimum confidence*. The *support* of a rule can be defined as the number of spatial objects that satisfy this pattern. The *confidence* of a rule can be defined as the probability of the consequent pattern occurs if antecedent occurs.

We implemented the algorithm proposed in [1] for mining strong spatial association rules. In order to enhance the power of spatial association rule mining, we have also implemented the suggested extensions. These extensions include (*i*) mining non-spatial consequent association rules and (*ii*) using multiple concept hierarchies. Additionally, we have provided the ability of mining spatial association rules at multiple levels of spatial relationship hierarchy.

Our main contribution to the algorithm is adding fuzziness to the hierarchies used in the system. Our observations showed that neither the concept hierarchies nor the spatial relationship hierarchies are always crisp in structure in real-world applications. Fuzzy data mining and discovery of fuzzy generalized association rules have been studied in [3]. Referring to this study, the concept hierarchy and the spatial relationship hierarchy definitions are changed, and also the calculations performed in spatial association rule mining algorithm are modified accordingly.

The rest of the paper is organized as follows. In section 2, we summarize the algorithm we have used in this paper. In section 3, we describe our implementation. Section 4 presents how fuzziness is incorporated into the implementation. We define a fuzzy concept hierarchy and a fuzzy spatial relations hierarchy and we show how mining of spatial association rules is done using these two fuzzy hierarchies. After explaining the modifications that we propose to the existing algorithm we conclude our paper in section 5.

2 Spatial Association Rules

Generalization-based spatial data mining methods [4,5,6] discover spatial and non-spatial relationships at a general concept level, where the spatial objects are expressed as merged spatial regions [4] or clustered spatial points [5,6]. However, these methods cannot discover rules concerning structure of spatial objects and spatial/spatial or spatial/non-spatial relationships. An algorithm, proposed in [1] discovers spatial association rules representing object/predicate relationships containing spatial predicates. The examples of Spatial Association Rules (*SAR*s) are as follows:

- Spatial consequent with non-spatial/spatial antecedent(s)

  is_a (X, gas_station) → *close_to* (X, highway)

- Non-spatial consequent with spatial antecedent(s)

  is_a (X, house) ∧ *close_to* (X, beach) → *is_expensive* (X)

In general, a "**Spatial Association Rule**" is defined as a rule of the form

$$P_1 \wedge \ldots P_m \rightarrow Q_1 \wedge \ldots Q_n \qquad (c\%)$$

where at least one of the $P_1 \wedge \ldots P_m$, $Q_1 \wedge \ldots Q_n$ is a spatial predicate and $c\%$ is the confidence of the rule. The confidence indicates that the c percent of the objects satisfying the antecedent of the rule also satisfy the consequent.

In the algorithm proposed in [1], the data is organized in a hierarchy called *Concept Hierarchy.* Due to this hierarchical structure, it is possible to find multiple level *SAR*s. For instance, some of the concept hierarchies for relations used in our implementation are defined as follows:

- A concept hierarchy for *LivingAreas*:

(LivingAreas (LargeLivingAreas (
States (Texas, Arizona…),Counties (King, Lewis…) …) …)
(SmallLivingAreas (
Cities (Beacon NY, Bedford OH,…)…)…)…) …)

- A concept hierarchy for *Transportations*:
(Transportation(Interstates(Washington-Pacific, Oregon-Pacific, Wyoming-Mtn…)))

- A concept hierarchy for *Waters*:
(Waters (Rivers (Shores (DoubleLineShorelines (1140, 1145…))
Canals (Navigable (436,773…), Other (2156,722…))
River (375, 442,….))))

Similarly, the spatial predicates are also structured in a hierarchical manner (Figure 1). The predicate "*g_close_to*" is a predefined predicate for "*generalized close to*" and covers all the spatial relations referred in the algorithm. Thus it is possible to find *SAR*s both using spatial approximations (like "*g_close_to*") and referring multiple levels of spatial hierarchy (like at the first level using "*not_disjoint*", "*close_to*" or at the second level using *intersects, inside*, *covers*).

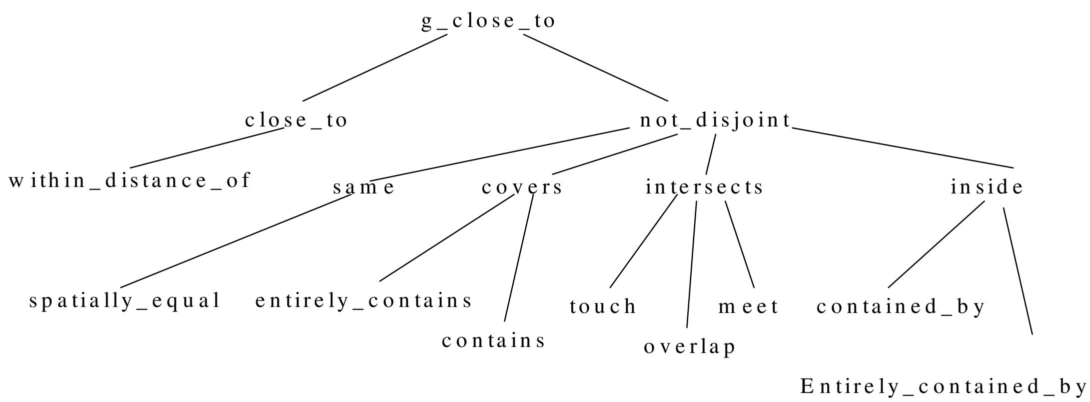


Figure 1. The hierarchy of spatial relations

In the algorithm, the rules that are supposed to be discovered are presented in an SQL-like query structure. An example of such a query is as follows:

```
discover Spatial Association Rules
inside USSample
from Transportation T , LivingAreas LA
in relevance to Water W
where g_close_to(W.geo, X.geo) AND X in (LA, T)
and W.River_Type = 'Single line perennial'
and LA.AnnulRain > 40
and LA.AveTemp > 30
```

This query means that, it is expected to find spatial association rules in "*USSample*" warehouse, related with *Water* and the features, *LivingAreas* and *Transportation*, which satisfy the *g_close_to* relationship with *Water* feature. The selection condition for *Water* is being in type "Single Line Perennial" and for *LivingAreas* is having annual rain greater than 40 and average temperature greater than 30.

In the execution of this query, first the set of all the relevant data is retrieved. The data sets retrieved are (1) Waters: only those whose type is "Single Line Perennial", (2) Transportations: all kinds of transportation, since there are no selection conditions on this feature, (3) LivingAreas: whose annual rain is greater than 40 and average temperature is greater than 30.

Next, the "*generalized close to*" (*g_close_to*) relationships between the instances of the main entity, Water, and other two types of entities (i.e. *LivingAreas* and *Transportation*) are computed. Here, to obtain a less expensive computation, the MBR data structure and plane-sweeping algorithm, or R* trees and other approximations can be used. In our application, however, we used the spatial relationship query functions of Geomedia tool of Intergraph Corporation[10]. The derived spatial predicates (i.e. *g_close_to*) are collected in a table, Table1. For each entity, the support values are calculated and those whose support value is below the required minimum support threshold, are eliminated by removing their columns from table. Here, since the support values for both "*LivingAreas*" and "*Transportation*" are above this threshold, none of the columns is removed.

Water	**Transportation**	**LivingAreas**
$Water_1$	$Transportation_1$, $Transportation_2$	$LivingAreas_1$
$Water_2$		$LivingAreas_2$, $LivingAreas_3$
$Water_3$	$Transportation_3$	

Table1. The computed *g_close_to* relation.

At this level of computation, some interesting large item sets can be discovered. For instance, rules like the following can be discovered at this level:

is_a(X,SLP_River) → *g_close_to*(X,Transportation) (78%)
is_a(X,SLP_River) ∧ *g_close_to*(X,Transportation) → *g_close_to*(X,LivingAreas) (56%)

After the elimination of those items whose support value is below the *minimum support,* the refined computation is performed on larger predicate sets. Each *g_close_to* is replaced with the actual spatial relation such as *touch*, *overlap*, *contains,* etc. A new table, Table2 is constructed.

Water	**Transportation**	**LivingAreas**
$Water_1$	touch(T_1), within_distance_of(T_2)	within_distance_of(LA_1)
$Water_2$		touch(LA_2), meet(LA_3)
$Water_3$	overlap(T_3)	

Table2. Detailed spatial relationships for large sets.

Table2 forms a base for the computation of detailed spatial relationships at multiple concept levels. Afterwards, Table3 is constructed. This table contains the large k-predicate sets where k=1, 2, … For example, for each row of Table2, if *Transportation* attribute is non empty, the related spatial relation count is incremented by 1 for each *SAR* predicate. In this way the 1-predicate rows (i.e. k=1) are constructed. If the support (count) value is below the threshold, then this row is removed from table. Then the 2-predicate rows are formed by pair-wise computation of 1-predicates.

k	**large k-predicate sets**	**Count**
1	touch(Transportation)	144
1	within_distance_of(Transportation)	168
1	meet(LivingAreas)	542
2	touch(Transportation), meet(LivingAreas)	103

Table3. Large k-predicate sets at the top concept level.

The *SAR*s can be directly computed from Table3. For example,

is_a(X, SLP_River) ∧ *touch*(X, Transportation) → *meet*(X, LivingAreas) (71%)

Here, *is_a*(X, SLP_River) is added to the rule, since this condition is specified in the input query by the forth and sixth lines, specifically by the phrases "**`in relevance to Water W`**" and "**`W.River_Type = 'Single line perennial'`**". Using the given support values in calculation 103/144 gives the confidence of the rule, 71%.

After mining rules at the highest level of the concept hierarchy, the large k-predicate sets for the intermediate levels are also computed in the same way. For this computation a new table, which is similar to Table3, is constructed. However, the attributes of Table3 are replaced with their sub-items in this level of the concept hierarchy. For example, *LivingAreas* is replaced with *BigLivingAreas* or *SmallLivingAreas*; *Transportation* is replaced with *Interstate*.

k	large k-predicate sets	Count
1	touch(Interstate)	144
1	within_distance_of(Interstate)	168
1	meet(BigLivingAreas)	358
1	meet(SmallLivingAreas)	184
2	touch(Interstate), meet(BigLivingAreas)	78

Table4. Large k-predicate sets at the second concept level.

After construction of this table the related *SAR*s can be found in the same way as above. However the key point is that, the minimum support and the minimum confidence values should be decreased in order to conclude enough interesting rules.

3 The Implementation of the Algorithm

The algorithm explained in the previous section, discovers spatial association rules with only spatial predicates in the consequent. However, according to the definition of a spatial association rule, the discovered rules can have a non-spatial consequent too. We added this feature to the algorithm as well. For this, the requested non-spatial consequent should be specified in the query with an additional phrase. For instance, the example query in the previous section is extended with the phrase "**`with consequent of W.AnnulWaterAmount >= Average`**".

For this kind of queries, the implementation of the algorithm is slightly changed. First, the relevant data are retrieved, and Table1 is constructed. However, this time a non-spatial attribute column is added to Table1. (e.g. The column *AnnulWaterAmount* will be added to Table1). And for each *Water* in the table, the value of this non-spatial attribute is also determined. Therefore at this level, the coarse *SAR*s can also contain the non-spatial attribute since Table1 will now have a non-spatial attribute column. For example:

is_a(X,SLP_River) → *AnnulWater*(X, HigherThanAverage) (78%)

After refined computations in Table2, Table3 is constructed in the same way as described in the previous section. However, this time k-predicate sets include the non-spatial attribute too.

k	large k-predicate sets	Count
1	touch(Transportation)	144
1	AnnulWater(High)	452
1	AnnulWater(Average)	364
2	touch(Transportation), AnnulWater(High)	135

Table3. Large k-predicate sets at the top concept level with non-spatial attribute.

Therefore the non-spatial attribute can appear in the consequent of the *SAR*s mined.

$$is\_a(\text{X, SLP\_River}) \wedge touch(\text{X, Transportation}) \rightarrow AnnulWater(\text{X,High}) \quad (93\%)$$

We also implemented a way to define and use multiple concept hierarchies in the mining process. In addition, we realized the need for finding *SAR*s at intermediate levels of spatial relationship hierarchy, for example at "*not_disjoint*", "*close_to*" level in Figure 1. For this extension, the implementation is not changed till the construction of Table2. At this point instead of leaf level spatial relation predicates for each item, the related intermediate level items are used. The table becomes like:

Water	Transportation	LivingAreas
Water_1	not_disjoint(T_1), close_to(T_2)	close_to(LA_1)
Water_2		not_disjoint (LA_2), not_disjoint (LA_3)
Water_3	not_disjoint (T_3)	

Table2. Detailed spatial relationships at second level of spatial relationship hierarchy.

As it can be predicted, Table3 and associated *SAR*s contain these new spatial predicates. And the confidence thresholds should be determined according to these new predicates.

4 Extending the Algorithm with Fuzzy Hierarchies

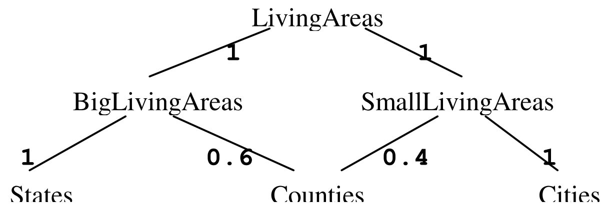
Mining generalized association rules is to discover the relationships between items belonging to different taxonomic structures. In many real-world applications, the taxonomic structures may not be necessarily crisp, rather certain fuzzy taxonomic structures reflecting partial belonging of one item to another may pertain.

Fuzzy extension to crisp taxonomic structures in mining association rules is studied in [3]. They extended the classical Srikant and Agrawal's algorithm (including Apriori algorithm and Fast algorithm) [9] to allow discovering the relationships between data attributes upon all levels of fuzzy taxonomic structures. In this way both fuzzy and crisp association rules can be discovered.

Mining spatial association rules algorithm in [1] uses two types of hierarchies: concept hierarchy and spatial relationship hierarchy. These hierarchies are defined by an expert who knows the characteristics of the data and they are given as an input to the algorithm. Since these hierarchies may not always be crisp, we have extended the algorithm with fuzzy hierarchies as described in [3]. Thus, we are able to mine spatial association rules with fuzzy concept and spatial relationship hierarchies.

4.1 Fuzzy Concept Hierarchies

In the concept hierarchies, the fuzziness is included to the *isa* relationship between an item and its sub-items. That is, an item (or a concept) can be defined as the child of more than one parent with different weights. In this way we are defining a partial belonging of a sub-item to its parent item. For example, the concept hierarchy of *LivingAreas* is modified as:



With this hierarchy, we state that a *County* is considered to be a *BigLivingArea* by 60% and a *SmallLivingAreas* by 40%. According to the algorithm in [3], for this kind of taxonomic structures, the degrees of leaf nodes with respect to their ancestors are placed in a table as shown in Figure 2.

States	1/States, 1/BigLivingAreas, 1/LivingAreas
Counties	1/Counties, 0.6/BigLivingAreas, 0.4/SmallLivingAreas, 0.6/LivingAreas
Cities	1/Cities, 1/SmallLivingAreas, 1/LivingAreas

Figure 2. Leaf-nodes and their ancestors' degrees

The calculations of these degrees are described in detail in [3]. However, as an example, the degree of *Counties*, according to *LivingAreas* is calculated as:

$$\mu(\textit{Counties}, \textit{LivingAreas}) = \max(\min(0.6, 1), \min(0.4, 1)) = 0.6$$

While *LivingAreas* is the parent of *BigLivingAreas* and *SmallLivingAreas* at degrees 1, either *BigLivingAreas* or *SmallLivingAreas* is a parent of *Counties* in a partial degree. Therefore, the degree of *Counties* being a child of *LivingAreas* is partial naturally, which is 0.6, the maximal between the path *Counties-BigLivingAreas-LivingAreas* and the path *Counties-SmallLivingAreas-LivingAreas*.

In our implementation, after the construction of Table1, while concluding interesting large item sets, the calculations are slightly changed because of the fuzzy concept hierarchy. This time all *States*, *Counties* and *Cities* should be counted as individually and be multiplied with their degree values. The count (support) of *LivingAreas* is found as:

count (*LivingAreas*) = count(*States*) * 1 + count(*Counties*) * 0.6 + count(*Cities*) * 1

where the multipliers (1, 0.6, 1) are the degree of each item with respect to *LivingAreas*.

As expected, the confidence of the rules at this level decreases. After Table2 is constructed, similar calculations are done for finding 1-predicate sets of Table3. However 2-predicate sets are counted in a different way. For instance, considering our running example query, the count value of 2-predicate set [*intersects* (Transportation), *contained_by*(LivingAreas)] is found as follows:

support = count(Water, *intersects* (Interstates), *contained_by* (States)) * min(1,1) +
count(Water, *intersects* (Interstates), *contained_by* (Counties)) * min(1,0.6) +
count(Water, *intersects* (Interstates), *contained_by* (Cities)) * min(1,1)

where, count(W, *intersects* (T), *contained_by* (LA)) means the number of W attribute which intersects with the T attribute and also contained by the LA attribute. The multipliers in the function shows the minimum of the degrees of the attributes according to their parents referring to the tables like in Figure 2. For our example, this table contains *Interstates* = 1/*Transportation* and *Counties* = 0.6/*LivingAreas*.

After all 2-predicate sets are counted as explained above, the construction of Table3 is finished and the first level *SAR*s can be found by using this table as described in previous sections.

Similar calculations are performed in order to find spatial association rules at the intermediate levels of the concept hierarchy. This time, the table in Figure 2 is used for finding the degree of each attribute according to the parents at the specified level. The minimum support and minimum confidence values are determined by concerning these fuzzy values and the specified level number.

4.2 Fuzzy Spatial Relationship Hierarchy

The predicates at the leaf level of the hierarchy of spatial relations (see Figure 1) are predefined spatial functions. We construct this taxonomic structure from the nine built-in spatial functions. The definitions of some functions gave us the motivation for adding fuzziness to the spatial relationship hierarchy. For example the function *touch* returns the following shaded objects when it is called by the boundary object.

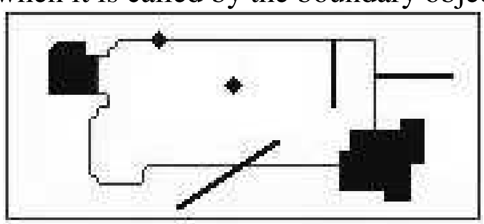
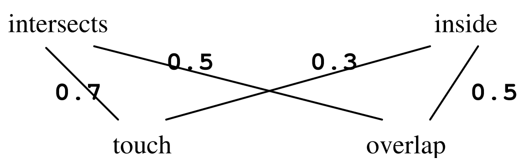


Figure 3. Spatial Relation "*touch*"

Figure shows the returned features from "*touch*" spatial operator. Since some of the returned objects can also be perceived as "*inside"*, *touch* operator can be defined fuzzily as the child of both "*intersects*" and "*inside*" in the spatial relationship hierarchy shown in Figure 1. Similar reasoning exists in the definition of "*overlap*" spatial operator. Therefore, the part of the hierarchy given in Figure 1 is modified as:



In the implementation, first a table similar to the one in Figure 2 is constructed for the fuzzy spatial relationship hierarchy and the degrees are determined according to the algorithm in [3]. After the construction of Table1, the count operation for *g_close_to* is changed in order to use the degree values calculated. Values are changed since the degree of *touch* and *overlap* are changed from 1 to 0.7 and 0.5, respectively. For example, if *Transportation* satisfies 3 spatial relation operators (*within_distance_of*, *touch*, *meet*) with specified *Water*, the total count of *g_close_to* is calculated as:

count(*g_close_to*) = count(*within_distance_of*) * 1 +
count(*touch*) * 0.7 +
count(*meet*) * 1

and the interesting large item sets are concluded according to these values. Here again, the multipliers are the degrees of each operator according to "*g_close_to*" predicate.

There is no change in the construction of Table2 and the counting 1-predicate item sets of Table 3. However in calculating 2-predicate item sets, fuzzy spatial calculations take place. For example, the calculation of the support value for the following 2-predicate item set

[*not_disjoint*(Transportation), *close_to*(LivingAreas)]

is as follows:

count(Water, *touch*(Interstates), *within_distance_of*(LivingAreas)) * min(0.7,1) +
count(Water, *meet* (Interstates), *within_distance_of* (LivingAreas)) * min(1,1)

As mentioned before, we are able to find *SAR*s at the intermediate levels of the spatial relationship hierarchy just like we did with the concept hierarchy. During mining *SAR*s in these intermediate levels, the fuzzy values are determined according to the specified level value and calculations and the support values are changed due to these fuzzy values.

4.3 Combining the Two Fuzzy Hierarchies

In the preceding two sections we described how to make concept hierarchies and spatial relationship hierarchy fuzzy separately. However one may need to have both fuzzy hierarchies together in an application. For such applications, the input query is changed as follows:

```
discover fuzzy Spatial Association Rules
using CH_1
referring SR_1
inside USSample
…
```

Here, the **CH_1** gives the name of the (fuzzy) concept hierarchy, which will be used in the mining process. Similarly, the **SR_1** is the name of the (fuzzy) spatial relationship hierarchy. These two hierarchies can have fuzzy or crisp structure. Therefore the concluded *SAR*s can be constructed using a fuzzy or crisp concept hierarchy with a fuzzy or crisp spatial relationship hierarchy.

It is important to note that in this study, we assumed that, the user has a good knowledge of data and of the hierarchies used in the mining process. Because the minimum support and the minimum confidence values at each level, should be determined by the user.

5 Conclusion

In this paper we have integrated two existing algorithms [1,3] for mining fuzzy spatial association rules. The algorithm in [1] presents a method for mining strong spatial association rules in large spatial databases. The algorithm in [3] proposes mining generalized association rules using fuzzy taxonomic structures. By observing the similarities between the concept hierarchies of [1] and the taxonomic structures defined in [3], we combined the two approaches in *SAR* mining. We applied fuzziness not only on concept hierarchies but also on spatial relationship hierarchy because of the definitions of the spatial relation operators.

When fuzziness is incorporated to the hierarchies, the calculations performed for spatial association rule mining are changed according to the algorithm in [3]. This fuzzy extension also brings the necessity of the determination of new support and confidence values.

At the end, we have been able to mine spatial association rules in large spatial databases using fuzzy or crisp concept hierarchies with a fuzzy or crisp spatial relationship hierarchy at multiple levels of concept and spatial relationship taxonomic structures.

References:

1. K.Koperski, J.Han. Discovery of Spatial Association Rules in Geographic Information Databases. *Symposium on Spatial Databases*, 1995
2. J. Han, Y. Fu, Mining Multiple Level Association Rules in Large Databases. *IEEE Transactions on Knowledge and Data Engineering, Vol. 11*, September/October 1999
3. G. Chen, Q. Wei, E.E. Kerre. Fuzzy Data Mining: Discovery of Fuzzy Generalized Association Rules. *Fuzzy Terms*, pp.45-66
4. W. Lu, J.Han, and B.C. Ooi. Discovery of General Knowledge in Large Spatial Databases**.** In *Proc. Far East Workshop on Geographic Information Systems* pp.275-289, Singapore, June 1993
5. R. Ng, and J. Han. Efficient and effective clustering method for spatial data mining. In *Proc. 1994 Int. Conf. VLDB*, pp. 144-155, Santiago, Chile, Sept. 1994
6. M.Ester, H. Kriegel, J. Sander, X. Xu. Clustering for Mining in Large Spatial Databases. *In Special Issue on Data Mining, KI-Journal Science Tech Publishing, Vol.1, 1998*
7. K.Koperski, J.Han, J. Adhikasy. Spatial Data Mining: Progress and Challenges. *Survey Paper.*
8. W. Wong, J. Yong, R. Muntz. An approach to Active Spatial Data Mining Based on Statistical Information. IEEE *Transactions on Knowledge and Data Engineering, Vol. 12 No.5*, September/October 2000.
9. R. Agrawal, R. Srikant. Fast Algorithms for mining Association Rules. In *Proceedings of the 20th Int'l Conference on Very Large Databases*, Santiago, Chile, September 1994
10. *Intergraph 's Mapping and GIS Solutions, Geomedia* http://www.intergraph.com/gis/geomedia/