



**The University of Sydney**

**An Efficient Approach for Mining  
Positive and Negative Relationships in Spatial Databases**

Technical Report Number 551

July 2004

Bavani Arunasalam, Sanjay Chawla and Pei Sun

ISBN 1 86487 648 4

**School of Information Technologies  
University of Sydney NSW 2006**

# An Efficient Approach for Mining Positive and Negative Relationships in Spatial Databases

Bavani Arunasalam

Sanjay Chawla

Pei Sun

University of Sydney  
School of Information Technologies  
Sydney, NSW, Australia  
{bavani,chawla,psun2712}@it.usyd.edu.au

3rd July 2004

## Abstract

We propose an efficient approach, called *Negative Pruning MaxPI (NP\_MaxPI)*, which can simultaneously mine complex spatial relationships which include both positive and negative patterns. Our approach is based on showing that for a certain support-like measure, itemsets which have both positive and negative features satisfy a weak monotonic property. This is used as a basis for aggressively pruning the candidate space of itemsets. Furthermore we evaluate the correctness of our results using a standard statistical tool, known as Ripley's *K*-function. Finally we carry out extensive experiments on a large extract of the Sloan Digital Sky Survey (SDSS) Database to demonstrate the utility of our approach for large scale data exploration.

**Keywords:** Spatial Data Mining, Postive and Negative Features, Astronomy database, Ripley's *K*-Function

## 1 Introduction

An important problem in spatial statistics is that of characterizing a distribution of spatially indexed points. Given a set of points  $S = \{x_i\}$ , is there a way to quantify that the points in  $S$  exhibit a *random*, *correlated* or *negatively correlated* behavior? Scientists working in diverse fields including ecology, geology and astronomy are interested

in inferring such information from their data sets as it often provides insights about the underlying mechanisms at play. For example, astrophysicists are trying to understand the structure of the universe from the point distribution of different morphological types of galaxies - the two main types being elliptical and spiral [5].

The standard tool to test for such patterns is the two-point correlation function  $\zeta$ . Intuitively  $\zeta$  calculates the likelihood for finding a point near the vicinity of another given point and can be expressed as

$$\zeta(\delta A) = \frac{N_{obs}(\delta A)}{N_{background}(\delta A)} - 1$$

where  $N_{obs}(\delta A)$  is the number of points observed in a small area  $\delta A$  and  $N_{background}(\delta A)$  is the expected number of points that will be observed assuming that the data is sampled from a given background distribution. Now the test to characterize the data set can be summarized as

$$\zeta = \begin{cases} < 0 & \text{then S is negatively correlated} \\ \approx 0 & \text{then S agrees with background} \\ > 0 & \text{then S is positively correlated} \end{cases}$$

In order to calculate the exact form of  $\zeta$  several assumptions have to be explicitly made. For example, if  $\zeta$  does not depend upon the location the process is called homogeneous otherwise it is called inhomogeneous. Similarly  $\zeta$  may only depend upon the distance between the points

(isotropic) and not also the direction (an-isotropic). Based on these assumptions and the underlying domain expertise several forms of  $\zeta$  have been proposed in the literature, especially in the astrostatistics community [5].

Our objective is to use techniques from spatial association rule mining to determine significant local spatial patterns and then use statistical techniques to determine if these patterns are indeed substantive.

A spatial pattern is one between features in a metric space, defined in terms of the co-locational trends over that space. An example is to determine the confidence of the ‘where there’s smoke there’s fire’ with respect to a set of coordinates, each representing the feature, smoke or fire. The task here would be to determine whether fire occurs in the neighbourhood of smoke more than is randomly likely.

Neighbourhoods are defined in terms of cliques (also known as neighbour-sets). A clique is defined as any set of items such that all items in that set co-locate. For example, in Figure 1 and Table 1, the co-locational pattern  $\{A, C\}$  occurs 3 times, v, vii and ix. In spatial data, two features are typically said to co-locate if they are positioned within a distance  $d$  of one another. As has been assumed in Figure 1,  $d$  is usually a constant, but it may also be defined as varying locally within the space or with respect to a given feature.

One factor unique to spatial data is that the number of transactions a single item may participate in is potentially unbounded, while in a market-basket this is limited to one (obviously, two people may purchase the same toothpaste product, but not the same exact tube). Self-co-location is also more likely in spatial data. The upper limit in a marketbasket is multiple purchases of only one product, which is less likely than an equivalent spatial situation of an area of monoculture forest. Similarly, there may be direct relationships between features that don’t co-locate, as between animals displaying territorial behaviour, making such relationships intrinsically more interesting in spatial data.

A complex relationship is simply any combination of these different relationships. It is important to note that while the relationships are defined as complex, the phenomena they represent are often very simple, for example:

1. Crime is more likely to co-occur on streets with no lighting near a subway station.
2. Many feral (non-native) animals in an area of forest

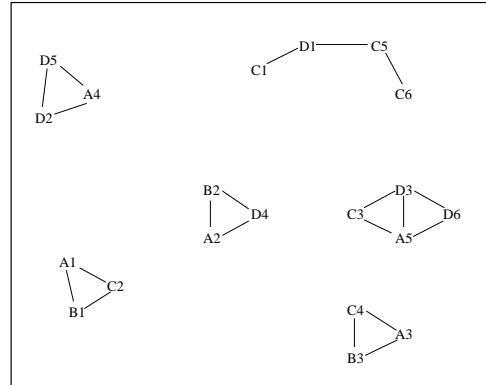


Figure 1: An Example of spatial co-location patterns

No	Clique
i	$C_1, D_1$
ii	$C_5, D_1$
iii	$C_5, C_6$
iv	$A_4, D_2, D_5$
v	$A_5, C_3, D_3$
vi	$A_5, D_3, D_6$
vii	$A_1, B_1, C_2$
viii	$A_2, B_2, D_4$
ix	$A_3, B_3, C_4$

Table 1: Cliques in Figure1

implies the absence of native ones.

3. Dogs only act as pack animals in non-urban environments

While much research has been done on *positive* association rule mining, very little progress has been achieved in mining negative patterns. The reason for this is that each positive pattern of length  $k$  gives rise to  $O(2^k)$  negative patterns making the search space exponentially larger than the space for positive patterns. In this paper we present an efficient algorithm to discover complex relationships in spatial databases.

## 1.1 Main Contributions

1. We show that for a certain class of measures  $M$ , a **monotonic-like property holds for positive and negative patterns**. This allows us to design a level-

wise algorithm to prune both positive and negative candidate patterns. Our approach is based on the following observation:

Suppose  $\{A, B\}$  is a 2-pattern and let  $\sigma(A, B)$  be its support. Then if  $\sigma(A, B)$  is greater than 50%, because  $\sigma(A, -B) = \sigma(A) - \sigma(A, B)$ , it would follow that  $\sigma(A, -B)$  is less than 50%. The notation "-B" represents the absence of B. However, one rarely chooses a support level this high though a confidence level greater than 50% is quite natural. The problem with using just the *confidence* is that it does not follow any monotonic property. Thus we need a measure that uses high threshold values like the confidence and still enjoys a monotonic property. In this paper we use the Maximal Participation Index (MaxPI) measure [4]. The MaxPI of a set  $(A, B)$  is defined as

$$MaxPI(A, B) = \max\left\{\frac{\sigma(A, B)}{\sigma(A)}, \frac{\sigma(A, B)}{\sigma(B)}\right\}$$

It has been shown before that MaxPI has a weak-monotonic property. In Section 6 we prove that (under mild assumptions)  $MaxPI(A, B) > t \geq 0.5$  then  $MaxPI(A, -B) < t$  or  $MaxPI(-A, B) < t$ . This allows us to prune large number of candidate negative patterns.

2. In an earlier work we had introduced a taxonomy of complex spatial relationships that are worthy of being mined [1]. For example, we define a relation  $A^+ \rightarrow -B$  as a multi-feature exclusive relationship. An example of such a relationship is  $Elliptical^+ \rightarrow -Spiral$ . We will show how to efficiently mine these relationships using our approach.
3. One of the weaknesses of association rule mining is that the number of rules that are generated far exceeds the "genuine" numbers of patterns that are interesting. We follow an approach analogous to the filter-refine strategy popular in spatial databases. We will use the MaxPI to generate candidate patterns and use the Ripley's K-function (a form of two-point correlation function  $\zeta$ ) to validate (or filter out) the patterns.
4. Finally, we have carried out detail experiments on a large extract of the Sloan Digital Sky Survey

database to show how our approach can be used in practice. We discover patterns which are known to be genuine and others which may turn out to be "interesting" thus confirming the role of data mining as a tool for credible hypothesis generation.

## 2 Related work

Association rules are considered one of the major success stories of data mining research [9]. Association Rules are traditionally described in the framework of market basket analysis. Given a set of items I and a set of transactions T consisting of subsets of I, an Association Rule is a relationship of the form  $A \rightarrow^{s,c} B$ , where A and B are subsets of I while s and c are the minimum support and confidence of the rule. A is called the antecedent and B the consequent of the rule. The support  $\sigma(A)$  of a subset A of I is defined as the percentage of transactions which contain A and the confidence of a rule  $A \rightarrow B$  is  $\frac{\sigma(A \cup B)}{\sigma(A)}$ . Most algorithms for association rule discovery take advantage of the anti-monotonicity property exhibited by the support level: If  $A \subset B$  then  $\sigma(A) \geq \sigma(B)$ .

Our focus is to apply the principle of Apriori in spatial data. In [7] the first extension of the Apriori paradigm to spatial data was proposed. However in their method they materialized all the possible spatial relationships that they intended to mine. This is equivalent to determining the universe of candidate interesting relationships. Thus in some ways their technique was hypothesis driven rather than hypothesis generating. An efficient algorithm to mine a kind of spatial co-locations was proposed in [8]. The concepts of neighborhood, participation ratio, participation index were defined. Instead of support, the participation index was used as a pruning measure in the conventional Apriori-like technique.

The drawback of the above method is that some confident co-location rules with low support are also pruned. In order to solve this problem, [4] proposed the concept of maximal participation index and it was used as pruning measure to replace participation index. We will discuss these measures in detail in the section 5, as they are central to our approach.

An algorithm to mine both positive and negative association rules was proposed in [2]. Negative rules are generated from infrequent item sets and interest is used as

a further pruning measure. Their algorithm involves no spatial component.

### 3 Notations used

**Absence:** The absence of an item is represented by negation, for example  $-A$ . (Note: this is not the equivalent of the set-theory,  $\neg A$ , meaning the presence of any item other than  $A$ ).

**Self-co-location:** Multiple instances of a feature (multiple items) are represented by a '+' following the feature, for example  $A+$ .

### 4 Types of Relationships

In this section we define different types of relationships and give examples for each type.

#### 4.1 Positive(Simple) Relationships

This is the most common type of relationship mined.

**Definition 1:** A positive relationship (multi-feature co-location) in spatial data is a set of features that co-locate at a ratio greater than some predefined threshold. In spatial data, the confidence of a positive relationship  $A \rightarrow B$ , is given by the fraction of unique  $A$ 's that co-occur in a clique containing the feature  $B$ .

Example:  $S_a$  Spiral Galaxies  $\rightarrow$   $S_b$  Spiral Galaxies

#### 4.2 Self-co-location / Self-exclusion

This is the measure of which a feature tends to co-locate with itself. Formally, it is the average cardinality of an item in a clique with respect to the expected cardinality of a random distribution. A galaxy is a simple example of stars self-co-locating. Extreme self-exclusion will be a perfectly uniform distribution with respect to the data space.

**Definition 2:** A feature is defined as self-co-locating in spatial data if the items representing that feature

co-locate with each other at a ratio greater than some predefined threshold.

Example: Elliptic galaxies tend to cluster more strongly.  $E \rightarrow E+$ .

**Definition 3:** A feature is defined as self-excluding in spatial data if the items representing that feature co-locate with each other at a ratio less than some predefined threshold. Example: Spiral galaxies are likely to occur in isolated clusters.

#### 4.3 One-to-Many relationships

Here, the user is specifically interested in the cardinality of a relationship between two features. For example it is noted that in SDSS data, some lone elliptical galaxies( $A$ ) are within unusually dense clusters of spiral galaxies( $B$ ),  $B+ \rightarrow A$ .

**Definition 4:** Two features are defined as having a one-to-many relationship in spatial data if one feature occurs multiple times in the presence of the other feature, greater than some pre-defined threshold. Included within this definition are two way one-to-many (many-to-many) relationships. Example: Some elliptic galaxies are within dense clusters of spiral galaxies.

Example:  $S+ \rightarrow E$ .

#### 4.4 Multi-feature exclusive relationships

These are exclusive relationships with respect to two or more features. In terms of a transactional representation, they are negative rules, which are explored in [2].

**Definition 5:** A multi-feature exclusive relationship in spatial data is defined as where a feature is absent from a given co-location at a ratio greater than a predefined threshold.

Example: Clusters of elliptic galaxies tend to dominate their surroundings and exclude other types of galaxies (no cause and effect is implied).  $E+ \rightarrow -S$ .

## 4.5 Complex relationships

These are any combination of two or more spatial relationship types.

**Definition 6:** A complex relation in spatial data is any relationship containing two or more of the properties defined in definitions 1-5.

The independent application of the above rules may produce complex relationships such as  $A \rightarrow B$  and  $A \rightarrow -C, B$ , but will not produce complex relationships such as  $A+, -C \rightarrow B+$ .

## 5 Maximal Participation Ratio

In this section we will briefly describe the notion of Maximal Participation Index (maxPI) as described in [4] where more details can be found.

### 5.1 Participation ratio

Given a co-location pattern  $L$  and a feature  $f \in L$ , the participation ratio of  $f$ ,  $pr(L, f)$ , can be defined as the support of  $L$  divided by the support of  $f$ . For example, in Figure 1, the support of  $\{A, B, C\}$  is 2 and the support of  $C$  is 6, so  $pr(\{A, B, C\}, C) = 2/6$ .

### 5.2 Maximal participation index

Given a co-location pattern  $L$ , the maximal participation index of  $L$ ,  $maxPI(L)$  can be defined as the maximal participation ratio of all the features in  $L$ , i.e.

$$maxPI(L) = \max_{f \in L} \{pr(L, f)\}$$

For example, in Figure 1,

$$maxPI(\{A, B, C\}) = \max\left(\frac{2}{5}, \frac{2}{3}, \frac{2}{6}\right) = \frac{2}{3}$$

A high maximal participation index indicates that at least one spatial feature (which we call the *maxfeature*) strongly implies the pattern. By using maxPI, rules with low frequency but high confidence can be found, which would otherwise be pruned by a support threshold [4].

## 5.3 Mining rules with low support and high confidence using maxPI

As discussed above the maxPI could be used to generate rules with low support and high confidence. For example if  $A$  is an infrequent feature the support of  $\{A, B\}$  will be very low and hence will be pruned by the support threshold. However the confidence of the rule  $A \rightarrow B$ , which is given as,  $conf(A \rightarrow B) = \frac{\sigma(A \cup B)}{\sigma(A)}$ , could be high.

The maxPI directly addresses this problem.  $MaxPI(A, B)$  is given as

$$\begin{aligned} maxPI\{A, B\} &= \max(pr(\{A, B\}, A), pr(\{A, B\}, B)) \\ &= \max(conf(A \rightarrow B), conf(B \rightarrow A)) \end{aligned}$$

This shows that a high confidence for the rule  $A \rightarrow B$  will lead to high maxPI, which will prevent rules with low support and high confidence from being pruned.

## 5.4 The weak monotonic property of maxPI

Maximal participation index is not monotonic with respect to the pattern containment relations. For example, in Figure 1,  $(maxPI(\{A, C\}) = 3/5 < (maxPI(\{A, B, C\}) = 2/3)$ . Interestingly, as pointed out in [4] the maximal participation index does have the following weak monotonic property:

If  $P$  is a  $k$ -colocation pattern, then there exists at most one  $(k-1)$  subpatterns  $P'$  of  $P$  such that  $maxPI(P') < maxPI(P)$ .

Relying on this weak monotonic property, the Apriori-like algorithm can be modified to mine confident patterns by using a maxPI threshold.

## 6 The NP\_MaxPI algorithm

The *maxPI* measure was introduced to discover *positive* co-location patterns with high confidence and low support. However our goal is to mine complex relationships which include both positive and negative patterns. We now show that *maxPI* is a good candidate to help achieve our goal, i.e., simultaneously discover both positive and negative patterns.

The main challenge in mining complex relationships is the high processing cost due the large number of candidate itemsets which include positive and negative features. Each positive candidate  $k$ -pattern will give rise to

$O(2^k)$   $k$ -patterns which contain a negative feature.

Let  $F=\{A,B,C,D\}$  be the set of all features in the spatial database. Then for mining complex relationships, the candidate 1- itemset would be  $\{A,B,C,D,-A,-B,-C,-D\}$ . Hence as the number of features in the spatial database increases, the candidate 1-itemsets is doubled. We propose an approach which effectively reduces the number of candidate itemsets when mining for positive and negative patterns. Our approach is based on Lemma 1 which shows how a large number of negative patterns can be pruned when a positive pattern is greater than the threshold  $t$  where  $t \geq 0.5$ . Since maxPI is based is on the confidence, such a high threshold is not unusual for maxPI.

**Definition:** Let  $F_k = \{f_1, f_2, \dots, f_k\}$  be a  $k$ -pattern, then  $F_{k-l} \equiv \{f_1, \dots, f_{l-1}, -f_l, f_{l+1}, \dots, f_k\}$ .

**Lemma 1** Let  $F_k$  be a  $k$ -pattern and  $t \geq 0.5$ . If  $maxPI(F_k) \geq t$  and  $maxPI(F_k) = pr(F_k, j)$  then  $MaxPI(F_{k-l}) < t$  for every  $f_l \in F_k, f_l \neq f_j$ .

**Proof:** By definition,  $MaxPI(F_{k-l}) =$

$$Max \left\{ Max_{i=1, i \neq l}^k \left\{ \frac{\sigma(f_1, f_2, \dots, -f_l, \dots, f_k)}{\sigma(f_i)} \right\}, \frac{\sigma(f_1, f_2, \dots, -f_l, \dots, f_k)}{\sigma(-f_l)} \right\} \left( \binom{2m}{2} - \binom{m}{2} - m - \binom{m}{2} \right) = m^2 - m$$

In spatial data, the number of instances of absence of a feature  $f_l$ , i.e.,  $\sigma(-f_l)$ , will be equal to the number of points in the area of observation which do not contain that feature. This will always be greater than the support of positive features. Therefore,

$$MaxPI(F_{k-l}) = Max_{i=1, i \neq l}^k \left\{ \frac{\sigma(f_1, f_2, \dots, -f_l, \dots, f_k)}{\sigma(f_i)} \right\}$$

But from the assumption  $MaxPI(F_k) = \frac{\sigma(f_1, f_2, \dots, f_k)}{f_j}$ . This implies that  $\sigma(f_j) \leq \sigma(f_i)$  for every  $f_i \in F_k, i \neq j$ .

Hence

$$MaxPI(F_{k-l}) = \frac{\sigma(f_1, \dots, -f_l, \dots, f_k)}{\sigma(f_j)} = \frac{\sigma(f_1, \dots, f_{l-1}, f_{l+1}, \dots, f_k)}{\sigma(f_j)} - \frac{\sigma(f_1, f_2, \dots, f_{l-1}, f_l, f_{l+1}, \dots, f_k)}{\sigma(f_j)} \quad (1)$$

Since  $\frac{\sigma(f_1, f_2, \dots, f_{l-1}, f_l, f_{l+1}, \dots, f_k)}{\sigma(f_j)} = MaxPI(F_k) > t \Rightarrow \frac{\sigma(f_1, f_2, \dots, f_{l-1}, f_{l+1}, \dots, f_k)}{\sigma(f_j)} > t$  because of support's monotonic property .

Therefore for  $t \geq 0.5$ , it follows from (1) that  $MaxPI(F_{k-l}) < t$ .

**Corollary 1** Let  $F = \{f_1, f_2, \dots, f_m\}$  be the set of all features in a spatial database. Then,  $0 \leq P \leq m^2 - m$

where  $P$  is the size of the candidate 2-itemsets pruned because of Lemma 1.

When mining for positive and negative patterns, for every  $f_i \in F$ , we also have to consider  $-f_i$ . This increases the feature set size to  $2m$ .

Hence the number of candidate itemset of size 2 generated from this feature set is  $\binom{2m}{2}$  ( because of maxPI no pruning is done at the first level).

From this set we remove candidates of the form  $(-f_i, -f_j)$  and  $(f_i, -f_i)$ . This reduce the size of the candidate set to  $\binom{2m}{2} - \binom{m}{2} - m$ .

**Case 1 :** All positive candidate item sets of size 2 are greater than the threshold  $t$ . Then by Lemma 1 all the candidate itemsets of size 2 with negative features would be pruned. Hence the additional number of pruning will be

$$\left( \binom{2m}{2} - \binom{m}{2} - m - \binom{m}{2} \right) = m^2 - m$$

**Case 2 :** All positive candidate item sets of size 2 are less than the threshold  $t$ .

Then all negative candidate item sets will be checked and hence there will be no additional pruning because of Lemma 1.

## 6.1 NP\_MaxPI Algorithm: Example

With the help of an example we describe the details of the *Negative Pruning MaxPI* (NP\_MaxPI) algorithm for mining complex spatial relationships.

1. Create a clique set such that every element represents a set of point forming a clique in the dataset. Table 2 gives an example of a clique set.
2. Create a transaction set in which each every element represents the types of point in the corresponding row of the clique set. Table 2 gives the transaction set generated from the given clique set.
3. Generate candidate itemsets of size one with positive and negative features. For the given example, the candidate 1-itemsets are

---

**Input:** Transaction table,  $t$

**Output:** Confident complex patterns

```
k=1
 $F_k \leftarrow \{f_1, f_2, \dots, f_n\} \cup \{-f_1, -f_2, \dots, -f_n\}$ 
            $\cup \{f_1^+, f_2^+, \dots, f_n^+\}$ 
While  $F_k \neq \phi$ 
  k=k+1
  //generate candidate itemsets
   $C_k = \text{maxPIgen}(F_{k-1})$ 
  for each  $c \in C_k$ 
    if everyfeature in c is positive
      OR check_negative(c) then
        generate  $F_k$  from  $C_k$  using MaxPI
      end if
    end for
  End While
```

---

Figure 2: *Algorithm NP\_MaxPI*. We generate candidate itemsets in the same way as apriori-gen( $F_{k-1}$ ), except that we use the weak monotonic property instead of monotonic property.

---

```
check=true
for each negative feature  $-f_i \in c$ 
  cp=c-(-fi) + fi
  if  $cp \in F_k$  and  $f_i \notin \text{maxFeature}(cp)$  then
    check = false
  end if
end for
return check
```

---

Figure 3: Function *check\_Negative()*

$\{A, -A, B, -B, C, -C, D, -D\}$ . For simplicity, we ignore the self\_col-locations such as  $A+$ . Since maxPI for all the 1-itemsets are 1, we do not prune any candidates at this level.

4. Generate candidate 2-itemsets from 1-itemsets, automatically pruning patterns such as  $\{A, -A\}$  and patterns with all negative features such as  $\{-A, -B\}$ .
5. Check maxPI of the candidate 2-itemsets and generate frequent 2-itemsets. While checking for negative candidate itemsets use Lemma 1 for additional pruning. Tables 3 and 4 show the process of checking candidate itemsets of sizes 2 and 3 with a threshold of 60%. The column 'checked' shows whether the candidate itemset was checked against the transaction table or not. When using the MaxPI algorithm all the given candidate itemsets would have been checked. However NP\_MaxPI avoids these checks because of Lemma 1. This increases the efficiency of mining complex patterns. The column *MaxFeature* denotes the feature for which the participation ratio of the candidate itemset was greater than the threshold. For example in Table 3, for candidate item AB,  $\text{pr}(\{A,B\},A)=5/6 > t$  and  $\text{pr}(\{A,B\},B)=5/6 > t$ . Therefore by Lemma 1, we do not have to check  $\text{maxPI}\{-A, B\}$  and  $\text{maxPI}\{A, -B\}$ . If we consider AC,  $\text{pr}(\{A,C\},A)=3/6 < t$  and  $\text{pr}(\{A,C\},C)=3/4 > t$ . Hence by Lemma 1, we do not have check  $\text{maxPI}\{-A, C\}$ . However we need to check A-C.
6. Repeat steps 4 and 5 for the consequent levels until there are no more frequent itemsets.

## 7 Experiments, Results and Analysis

We have carried out three sets of experiments to demonstrate the correctness, scalability and "applicability" of NP\_MaxPI.

**Correctness** We applied the NP\_MaxPI approach on a small data set to obtain an output  $O$  which contains patterns of the form  $\{A, B\}$  and  $\{C, -D\}$  (for a

candidate itemset	checked	Pr	MaxPI	MaxFeature	Pruned
AB	Y	5/6,5/6	5/6	A,B	N
A-B	N				
AC	Y	3/6,3/5	3/4	C	N
A-C	Y	3/6	3/6	-	Y
AD	Y	4/6,4/6	4/6	A,D	N
A-D	N				
-AB	N				
-AC	N				
-AD	N				
BC	Y	3/6,3/4	3/4	C	N
B-C	Y	3/6	3/6	-	Y
BD	Y	4/6,4/6	4/6	B,D	N
B-D	N				
-BC	N				
-BD	N				
CD	Y	2/4,2/6	2/4	-	Y
C-D	Y	2/4	2/4	-	Y
-CD	Y	4/6	4/6	D	N

Table 3: Mining candidate 2- itemset

candidate 3-itemset	checked	pr	MaxPI	MaxFeature	Pruned
ABC	Y	2/6,2/6,2/4	2/4	-	Y
ABD	Y	3/6,3/6,3/6	3/6	-	Y
ACD	Y	0/6,0/4,0/6	0	-	Y
A-CD	Y	4/6,4/6	4/6	A,D	N
BCD	Y	1/6,1/4,1/6	1/4	-	Y

Table 4: Mining candidate 3- itemset

fixed threshold). We then applied the Ripley’s K-function test to check whether  $A$  and  $B$  are positively correlated and  $C$  and  $D$  are negatively correlated. This is how we demonstrate the correctness of NP\_MaxPI.

**Scalability** We created a large synthetic data set in order to compare the MaxPI and the NP\_MaxPI algorithm. In particular we wanted to empirically test how much additional pruning was being achieved by the use of Lemma 1.

**Applicability** We extracted a large sample from the SDSS Database and applied the NP\_MaxPI algo-

rithm to test the effectiveness of our approach on a real data set.

## 7.1 NP\_MaxPI and Ripley’s K-Function

Ripley’s K function is a tool used to analyze the spatial pattern of point data. The K function is given by

$$K(t) = \lambda^{-1} E$$

where  $E$  is the number of events within distance  $t$  of a randomly chosen event and  $\lambda$  is the density (number per unit area) of events.

No	Clique	Transaction
i	$A_1, B_1, C_1$	$A, B, C$
ii	$A_2, B_2, C_2$	$A, B, C$
iii	$C_2, D_1$	$C, D$
iv	$A_3, C_3$	$A, C$
v	$A_3, B_6, D_2$	$A, B, D$
vi	$A_4, B_3, D_3$	$A, B, D$
vii	$A_5, B_5, D_4$	$A, B, D$
viii	$B_4, C_4, D_5$	$B, C, D$
ix	$A_6, D_6$	$A, D$

Table 2: Clique Set

If A is the area of the study region and N is the observed number of points then,  $\lambda = N/A$ .

Let  $d_{ij}$  be the distance between the  $i^{\text{th}}$  and the  $j^{\text{th}}$  points and  $I(d_{ij})$  be the indicator function such that  $I(d_{ij})=1$  if  $d_{ij} \leq t$  and 0 if  $d_{ij} > t$ .

The K function is estimated using the formula

$$K(t) = \lambda^{-1} \sum_{i=1}^N \sum_{j=1, j \neq i}^N w(l_i, l_j) I(d_{ij}) / N$$

where  $w(l_i, l_j)$  provides the edge correction.

Ripley's K function could be used to test complete spatial randomness i.e. test whether the observed events are consistent with a homogeneous Poisson process. If so,  $K(t) = \pi t^2$  for all  $t$ . i.e. under complete randomness  $t = (K(t)/\pi)^{1/2}$  for all  $t$ .

[6] suggests the generalization of K(t) function to multivariate spatial point process as follows :

$$K_{ij}(t) = \lambda_j^{-1} E$$

Where E is the number of type j events within distance t of a randomly chosen type i event.  $K_{ij}(t)$  function could be estimated similar to the univariate K(t) function as follows

$$K_{ij}(t) = (\lambda_i \lambda_j A^{-1}) \sum_{k=1}^N \sum_{l=1}^N w(ik, jl) I(d_{ik, jl})$$

Under complete spatial randomness  $K_{ij}(t) = \pi t^2$   
Hence  $L_{ij}(t) = (K_{ij}(t)/\pi)^{1/2} = t$

Neighbourhood Distance =4
$A \rightarrow D$ conf=83.33%
$D+ \rightarrow A$ conf=100.00%
$D+ \rightarrow -C$ conf=75.00%
$C+ \rightarrow -B$ conf=100.00%

Table 5: Rules from the synthetic data set

t	$L_{AD}$	$L_{DC}$	$L_{CB}$
0.00	0.00	0.00	0.00
0.50	0.00	0.00	0.00
1.00	0.00	0.00	0.00
1.50	0.00	0.00	0.00
2.00	2.02	0.00	0.00
2.50	3.50	1.81	2.33
3.00	4.52	2.52	2.33
3.50	4.52	3.11	3.37
4.00	4.52	3.11	3.37

Table 6: Results from Ripley's function

This shows that values of  $L_{ij}(t) - t > 0$  indicate attraction between the i type point and j and values  $< 0$  indicate repulsion.

We applied the *NP\_MaxPI* algorithm to a very small synthetic data set with 20 points and generated a set of complex rules with min confidence 70%. The rules are given in Table 5.

We applied Ripley's K function to the synthetic data set and calculated the  $L_{ij}$  values for different values of t where i, and j were the types of features in each of the rules in table 5. The different values of  $L_{ij}$  are given in table 6.

Table 6 shows that for type A and D,  $L_{AD} \geq t$  which indicates that these two types of objects are positively correlated. When comparing types C and D, we find that  $L_{CD} \leq t$  which shows negative correlation between the types. Similarly types C and B show negative correlation. These confirm the rules in table 5.

## 7.2 Performance evaluation of the algorithm

We generated synthetic datasets of sizes ranging from 170K to 1 million transactions with ten different fea-

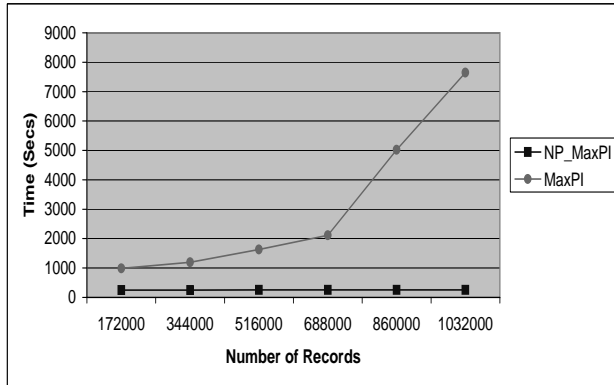


Figure 4: comparison of efficiency

Objects	Symbol
GALAXY-LRG-EARLY	A
GALAXY-LRG-LATE	B
GALAXY-MAIN-EARLY	C
GALAXY-MAIN-LATE	D
QSO	F

Table 7: List of object types and their symbols

tures. These datasets were generated using the following method. With each of the 10 features as the first feature, a random number of transactions were generated. Each transaction was generated with a random size from 1 to 10. Depending on the size of the transaction the consequent features were chosen randomly from the feature set. We then applied the MaxPI algorithm and *NP\_MaxPI* algorithm on the datasets to generate confident complex patterns. Figure 4 shows the time taken by both these methods to mine complex patterns from datasets of different sizes. It could be seen that the time taken by the *MaxPI* algorithm increases rapidly with the size of the dataset, while the time taken by the *NP\_MaxPI* algorithm increases only by few seconds because of the extensive pruning of the negative patterns using Lemma 1. We confirmed that the patterns generated by the two method were identical by using the unix *diff* utility.

## 7.3 Application of NP\_MaxPI algorithm on SDSS database

### 7.3.1 Data Preparation

The data for this experiment was obtained from the SDSS Data Release 1, from the online catalogue service. The online data contained information about over 150000 celestial objects. Among the hundreds of attributes stored in the database for each object we extracted the following attributes for our experiment

- The object's ID,
- coordinates (as a unit vector),
- object type, -its primary target flags,
- redshift,
- the difference between the u & r light magnitudes (used to separate galaxy types).

The distances between objects were calculated using Hubble's Law given by  $D = c * z / H_0$ , where  $c$  is the speed of light,  $z$  is the redshift and  $H_0$  is the Hubble's constant and is  $71 kmsec^{-1} Mpc^{-1}$  [11].

To ensure that the results for measuring the distance to each object would be as accurate as possible, only objects with a  $zConf$  value  $> 0.95$  (i.e. the object's redshift is  $>95\%$  certain) and  $zWarning = 0$  (i.e. there's no problem with the redshift) are used. This filtering cuts down the number of objects to around 117000.

The 'object type' attribute classifies the objects into 25 categories. However the current online data has objects only in 17 of these categories. Among the various object types, we extracted only the galaxies since 90% of the objects were classified as 'Galaxies'. We further classified the galaxies into 'main' galaxies and Luminous Red Galaxies (LRG). The main galaxies are closer (to the Earth) and brighter than the LRG. In the SDSS database the LRG were flagged with a particular value. We classified the 'main' and LRG galaxies further into Early and Late galaxies using the UV & red light magnitudes. A  $u-r$  value greater than or equal to 2.22 indicate Early galaxy and less than 2.22 indicate Late galaxies. From Hubbles Tuning Fork model [5] it follows that early galaxies are elliptical in shape and late are spiral / irregular. Table 7

Neighbourhood Distance :1 megaparsec
min confidence : 70%
$B+ \Rightarrow -C - F$
$D+ \Rightarrow -C - F$
$A+ \Rightarrow -B - D - F$
$B \Rightarrow -A - C - F$
$C \Rightarrow -B - D - F$
$C+ \Rightarrow -B - D - F$
$A \Rightarrow -B - C - D - F$

Table 8: Rules from SDSS database

lists the different types of galaxies and the corresponding symbol used in this paper.

### 7.3.2 Rules Generated from SDSS Database

We applied the NP\_maxPI algorithm to the data set extracted from the SDSS database and some of the interesting rules generated are shown in Table 7. The entire result set could be obtained from <http://www.cs.usyd.edu.au/~chawla/sdss.html>. From Table 2 we see that Feature A and C are early galaxies and hence they are elliptical in shape. Features B and D are late galaxies and hence they are spiral in shape.

Among the rules in table 8, the rules to be noted are  $A+ \Rightarrow -B - D - F$  and  $C+ \Rightarrow -B - D - F$ . These rules show that A+ is negatively correlated with B,D and F but not C . Similarly C+ is negatively correlated with B,D, and F and not A. These rules conform to the well know fact that when elliptical galaxies co-locate the spiral galaxies are excluded.

## 8 Summary and Conclusion

In this paper we demonstrated the problem of generating complex patterns in spatial databases. We then presented an efficient approach ,  $NP\_MaxPI$ , which mines for complex patterns by extensively pruning candidate negative patterns. The results of our experiments show (i) the correctness of our approach using Ripley’s K function (ii) the significant performance improvement over MaxPI algorithm and (iii) the efficacy of our approach on real

dataset by generating confident complex pattern in SDSS spatial database.

## 9 Acknowledgements

Thanks to Chris Bowman for helping us create the data set.

## References

- [1] R.Munro, S. Chawla, and P.Sun. Complex Spatial Relationships, In Proc. of the IEEE ICDM,pages 227-234,2003.
- [2] X.Wu, C.Zhang, and S.Zhang. Mining Both Positive and Negative association Rules. InProc. 19th International Conference on Machine Learning (ICML-2002),2002.
- [3] J. Gray, A.S. Szalay, A. Thakar, P. Kunszt, C. Stoughton, D. Slutz, and J. vandenBerg. Data Mining the SDSS SkyServer Database. Microsoft Tech Report, MSR-TR-2002-01,2002.
- [4] Y.Huang, H.Xiong, and S.Shekhar. Mining confident co-location rules without a support threshold. In Proc. 18th ACM Symposium on Applied Computing (ACM SAC),2003.
- [5] V.J.Martin and E.Saar. Statistics of the Galaxy Distribution. Chapman & Hall/CRC, 2002.
- [6] P.Dixon.Ripley’s K function.Department of Statistics,Iowa State University,2001.
- [7] HanK.Koperski and J. Han. Discovery of Spatial Association Rules in Geographic Information Databases. In Proc. 14th International Symposium on Large Spatial Databases(SSD95), Maine, 1995,pp.47-66.
- [8] S.Shekhar and Y. Huang. Discovering Spatial Co-location Patterns: A Summary of Results. In Proc. 7th International Symposium on Spatial and Temporal Databases(SSD01, L.A,CA, 2001

- [9] R.Agrawal and R.Srikant. Fast algorithms for mining association rules. In Proc. 20th International on Very Large Data Bases(VLDB),1994.
- [11] D.N.Spergel,M.Bolte and W.Freedman.The age of the Universe.In Proc.Natl. Sci.USA.Vol 94,pp 6579-6584,June 1997.