

Fast Mining of Complex Spatial Co-location Patterns using GLIMIT

Florian Verhein
The School of Information Technologies
University of Sydney
fverhein@it.usyd.edu.au

Ghazi Al-Naymat
The School of Information Technologies
University of Sydney
ghazi@it.usyd.edu.au

Abstract

Most algorithms for mining interesting spatial co-locations integrate the co-location / clique generation task with the interesting pattern mining task, and are usually based on the Apriori algorithm. This has two downsides. First, it makes it difficult to meaningfully include certain types of complex relationships – especially negative relationships – in the patterns. Secondly, the Apriori algorithm is slow. In this paper, we consider maximal cliques – cliques that are not contained in any other clique. We use these to extract complex maximal cliques and subsequently mine these for interesting sets of object types (including complex types). That is, we mine interesting complex relationships. We show that applying the GLIMIT itemset mining algorithm to this task leads to far superior performance than using an Apriori style approach.

1 Introduction

A spatial dataset often describes *Geo-spatial* or “*Astro-spatial*” (astronomy related) data. In this work, we use a large astronomical dataset containing the location of different *types* of galaxies. Datasets of this nature provide opportunities and challenges for the use of data mining techniques to generate interesting patterns.

One such pattern is the *co-location* pattern. A co-location pattern is a group of objects (such as galaxies) so that each is located in the neighborhood (within a given distance) of another object in the group.

A *clique* is a special type of co-location pattern. A clique is a group of objects such that *all* objects in that group are co-located with each other. In other words, given a predefined distance, if a group of objects lie within this distance from every other object in the group, they form a clique. Figure 1 shows 8 different objects $\{A1, A2, A3, B1, B2, B3, B4, C1\}$. The set $\{B1, B2, A3\}$ is a clique. However, $\{B1, B2, A3, C1\}$ is not, because $C1$ is not co-located with $B2$ and $A3$. Simi-

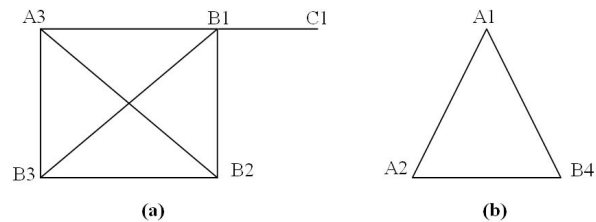


Figure 1. Clique example.

larly, $\{B1, B2, A3, C1\}$ is a co-location pattern but it is not a clique.

In this paper we consider *maximal cliques*. A maximal clique is a clique that does not appear as subset of another clique in the same co-location pattern (and therefore the entire dataset, as each object is unique). For example, in Figure 1, $\{A1, A2, B4\}$ forms a maximal clique as it is not a subset of any other clique. However, $\{A3, B2, B3\}$ is not a maximal clique since it is a subset of the clique $\{A3, B1, B2, B3\}$ (which in turn is a maximal clique). The second column of Table 1 shows all the maximal cliques in Figure 1.

In our dataset, each row corresponds to an object (galaxy) and contains its type as well as its location. We are interested in mining relationships between the *types* of objects. Examples of object types in this dataset are “early-type” galaxies and “late-type” galaxies. To clarify, we are not interested in co-locations of *specific* objects, but rather, co-locations of their *types*. Finding complex relationships between such types is useful information in the astronomy domain. In Figure 1, there are three types: $\{A, B, C\}$.

In this paper we focus on using maximal cliques to allow us to mine interesting *complex spatial relationships* between the object types.

A *complex spatial relationship* includes not only whether an object type, say A , is present in a (maximal) clique, but also:

- Whether *more than one* object of its type is present in the (maximal) clique. This is called a *positive type* and is denoted by $A+$.
- Whether objects of a particular type are not present in

ID	Maximal Cliques	Raw Maximal Cliques	Non-Complex Relationships	Complex Without Negative Relationships	Complex With Negative Relationships
1	{A3, B1, B2, B3}	{A, B, B, B}	{A, B}	{A, B, B+}	{A, B, B+, -C}
2	{B1, C1}	{B, C}	{B, C}	{B, C}	{-A, B, C}
3	{A1, A2, B}	{A, A, B}	{A, B}	{A, A+, B}	{A, A+, B, -C}

Table 1. Representing maximal cliques of Figure 1 as complex relationships

a *maximal clique* – that is, the absence of types. This is called a *negative type* and is denoted by $-A$.

The inclusion of *positive* and / or *negative types* makes a relationship *complex*. This allows us to mine patterns that say, for example, that A occurs with multiple B 's but not with a C . That is, the presence of A may imply the presence of multiple B 's and the absence of C . This is interesting in the astronomy domain. The last two columns of Table 1 show examples of (maximal) complex relationships.

We are not interested in *maximal* complex patterns (relationships) in themselves, as they provide only local information (that is, about a maximal clique). We are however interested in *sets* of object types (including complex types), that appear across the entire dataset (that is, amongst many maximal cliques). In other words, we are interested in mining *interesting complex spatial relationships* (sets), where “interesting” is defined by a global measure. We use a variation of the *minPI* [7] measure to define interestingness:

$$\text{minPI}(P) = \min_{t \in P} \{N(P)/N(\{t\})\} \quad (1)$$

Here P is a set of complex types we are evaluating, and $N(\cdot)$ is the number of maximal cliques that contain the set of complex types. Note that we count the occurrences of the pattern (set of complex types) only in the maximal cliques. This means that if the *minPI* of a pattern is above α , then we can say that whenever any type $t \in P$ occurs in a maximal clique, the entire pattern P will occur at least a fraction *alpha* of those maximal cliques. *minPI* is superior to simply using $N(P)$ because it scales by the occurrences of the individual object types, thus reducing the impact of a non-uniform distribution on the object types.

In this work we focus on *maximal cliques* because:

- The process of forming complex positive relationships makes sense. Suppose we extract a clique that is not maximal, such as $\{A1, B4\}$ from Figure 1. We would not generate the positive relationship $\{A+, B\}$ from this, even though each of $\{A1, B4\}$ are co-located with $\{A2\}$. So we get the correct pattern only once we have considered the maximal cliques.
- Negative relationships are possible. For example, consider the maximal clique in row 1 of Table 1. If we did not use *maximal* cliques, then we would also consider $\{B1, B2, B3\}$, and from this we would *incorrectly* infer that the complex relationship $\{B, B+, -A\}$ exists.

However, this is not true because A is co-located with each of $\{B1, B2, B3\}$. *Therefore, using non-maximal cliques will generate incorrect negative patterns.*

- Each maximal clique will be considered as a single instance (transaction) for the purposes of counting. In other words, we avoid multiply counting the same objects within a maximal clique automatically.
- Mining maximal cliques reduces the number of cliques by removing all redundancy. Also, it is possible to mine for maximal cliques directly. Finally, because negative types cannot be inferred until the maximum clique is mined, it does not make sense to mine cliques that are not maximal.

1.1 Problem Statement

Given the set of maximal cliques, find all interesting and complex patterns that occur amongst the set of maximal cliques. More specifically, find all sets of object types, including positive and negative (that is, complex) types that are interesting as defined by their *minPI* being above a threshold.

This problem therefore becomes an *itemset mining* task. In order to do this very quickly, we use the GLIMIT algorithm [8] as we shall describe in Section 2.3.

Including negative types makes the problem much more difficult, as it is typical for spatial data to be sparse. This means that the absence of a type can be very common. We will show that this is not a problem for our approach. In contrast, approaches relying on an Apriori style algorithm find this very difficult.

1.2 Contributions

We make the following contributions:

- We will show that GLIMIT can be used to mine complex, interesting co-location patterns very efficiently in very large datasets. We demonstrate that GLIMIT can be almost three orders of magnitude faster than using an Apriori based approach.
- We introduce the concept of *maximal cliques*. We describe how the use of *maximal cliques* makes more sense than simply using cliques, and we showed that they allow the use of negative patterns.

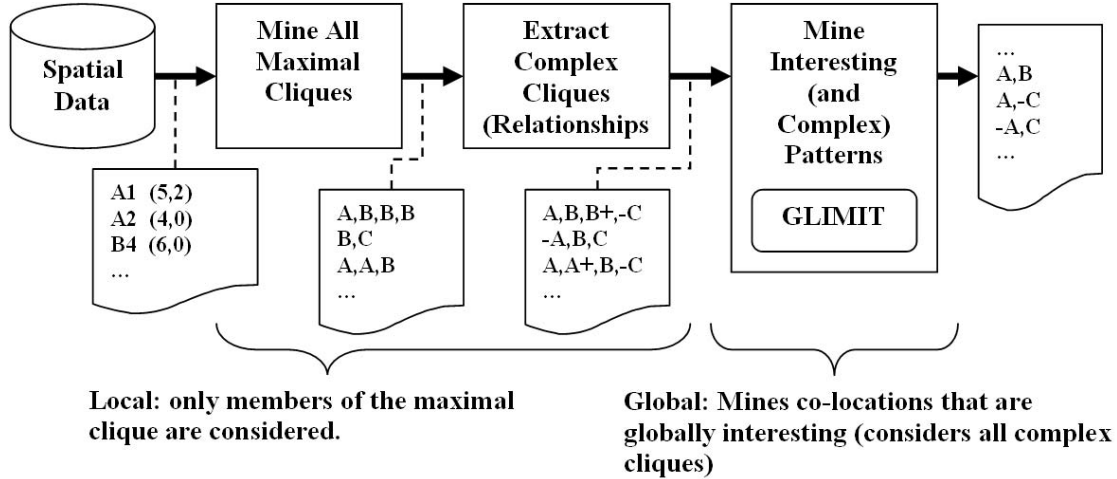


Figure 2. The complete mining process.

The rest of the paper is organized as follows: Section 2 gives further details of our approach. Section 3 contains our experiments and an analysis of the results. Section 4 puts our contributions in context of related work and we conclude in Section 5.

2 Our Approach

Figure 2 shows the overall flowchart of our method. First, a maximal clique mining algorithm finds all *maximal cliques*, and strips them of the object identifiers (producing raw maximal cliques as shown in Table 1). One pass is then made over the raw maximal cliques in order to extract complex relationships. We describe this in Section 2.2. This produces maximal complex cliques. Each of these complex, maximal cliques is then considered as a *transaction*, and an *interesting itemset mining algorithm*, using *minPI* as the interestingness measure, is used to extract the interesting complex relationships. We describe this in Section 2.3.

As shown in Figure 2, the clique generation and complex relationship extraction are local procedures, in the sense that they deal only with individual maximal cliques. In contrast, the interesting pattern mining is global – it finds patterns that occur across the entire space. Secondly, we consider subsets of maximal cliques only in the last step – after the complex patterns have been extracted.

2.1 Maximal Cliques

Consider a set of objects O with fixed locations. Given an appropriate distance measure $d : O \times O \rightarrow \mathbb{R}$ we can define a graph G as follows; let O be the vertices and construct an edge between two objects $o_1 \in O$ and $o_2 \in O$ if $d(o_1, o_2) \leq \tau$, where τ is a chosen distance. A *co-location pattern* is a connected subgraph.

Definition 1 (Clique) A clique $C \in O$ is any fully connected subgraph of G . That is, $d(o_1, o_2) \leq \tau \forall \{o_1, o_2\} \in C \times C$.

As we have mentioned in Section 1 we use maximal cliques so that we can define and use complex patterns meaningfully and to avoid double counting.

Definition 2 (Maximal Clique) A maximal clique C_M is a clique that is not a subset (sub-graph) of any other clique.

The mining of maximal cliques is done directly – it does not require mining all sub-cliques first. It is described in [2].

2.2 Extracting Complex Relationships

A relationship is called complex if it consists of *complex types* as defined in Section 1.

Extracting a complex relationship R from a maximal clique C_M is straightforward – we simply use the following rules for every type t :

1. If C_M contains an object with type t , $R = R \cup t$.
2. If C_M contains more than one object of type t , $R = R \cup t+$.
3. If C_M does not contain an object of type t , $R = R \cup -t$.

Note that if R includes a positive type $A+$, it will also *always* include the basic type A . This is necessary so that maximal cliques that contain $A+$ will also be counted as containing A when we mine for interesting patterns.

Recall that the negative type only makes sense if we use *maximal cliques*. The last three columns of Table 1 show the result of applying Rule 1, Rule 1 and Rule 2, and all three rules, respectively.

2.3 Mining Interesting Complex Relationships

In *itemset mining*, the dataset consists of a set of transactions T , where each transaction $t \in T$ is a subset of a set of *items* I ; that is, $t \subseteq I$. In our work, the set of complex maximal cliques (relationships) becomes the set of transactions T . The items are the object types – including the complex types such as $A+$ and $-A$. For example, if the object types are $\{A, B, C\}$, and each of these types is present and absent in at least one maximal clique, then $I = \{A, A+, -A, B, B+, -B\}$. An interesting itemset mining algorithm mines T for interesting itemsets. The support of an itemset $I' \subseteq I$ is the number of transactions containing the itemset: $support(I') = |\{t \in T : I' \subseteq t\}|$. So called *frequent itemset mining* uses the support as the measure of interestingness. For reasons described in Section 1 we use *minPI* (see Equation 1) which, under the mapping described above, is equivalent to

$$minPI(I') = \min_{i \in I'} \{support(I') / support(\{i\})\}$$

Since *minPI* is *anti-monotonic*, we can easily prune the search space for interesting patterns.

GLIMIT [8] is a very fast and efficient itemset mining algorithm that has been shown to outperform Apriori [1] and FP-Growth [4]. GLIMIT works by first transposing the dataset, so that each row, known as an *itemvector*, corresponds to an item. GLIMIT then makes one pass over the result (the *itemvectors*). It is an *item enumeration* algorithm, which means that it searches through the space of possible itemsets. It does this in a bottom up (the size of the itemsets increases along a branch of the search) fashion, so is suitable for measures that possess some form of *anti-monotonic* property [8]. The search progresses in a depth first fashion, which enables very little space to be used – specifically, space linear in the size of the dataset [8]. GLIMIT uses a framework defined by the following functions and operator [8]:

- $g(\cdot)$ performs a transformation on the transposed dataset.
- \circ is an operator that combined two *itemvectors* together to create a new *itemvector* corresponding to the union of the two itemsets.
- $m_{I'} = f(\cdot)$ is a measure on an itemset $I' \subseteq I$ (evaluated over the corresponding *itemvector*) that depends only on that itemset.
- $M_{I'} = F(\cdot)$ is a measure on an itemset $I' \subseteq I$ that uses $f(\cdot)$ and may depend on said itemset as well as any of its subsets.

The *minPI* measure can be incorporated into GLIMIT as follows (let $I' = \{1, 2, \dots, q\}$ for simplicity): $g(\cdot)$ is the

identity function (there is no transformation on the dataset), $\circ = \cap$ and $f(\cdot) = |\cdot|$ (the set size). This means that $m_{I'} = support(I')$. Finally, $M_{I'} = F(m_{I'}, m_1, \dots, m_q) = \min_{i \in I'} \{m_{I'} / m_i\}$.

We use GLIMIT with the above instantiations of its framework to mine interesting co-locations, as shown in Figure 2. For comparison, we will also use an Apriori [1] implementation.

The Apriori [1] and Apriori-like algorithms are *bottom up item enumeration* type itemset mining algorithms. Apriori works in a breadth first fashion, making one pass over the dataset for each level expanded. This is in contrast to GLIMIT, which makes only one pass over the entire dataset. In Apriori, a *candidate generation* step generates candidate itemsets (itemsets that may be interesting) for the next level, followed by a dataset pass (*support counting*) where each candidate itemset is either confirmed as interesting, or discarded. The support counting step is computationally intensive as subsets of the transactions need to be generated. GLIMIT operates on a completely different principle [8].

3 Experiments

We used a real life two dimensional astronomy dataset from the the Sloan Sky Digital Survey (SDSS)¹. We extracted all galaxies from this dataset, giving a total of 365,425 objects. There were 12 *types* of galaxies. The distance threshold used for generating the maximal cliques was 1 Mega-parsec².

A total of 121,506 maximal cliques (transactions) were generated in 39.6 seconds. This is quite a large dataset. We processed these in a number of ways (refer to Table 1 for examples of these) as described in Section 2.2:

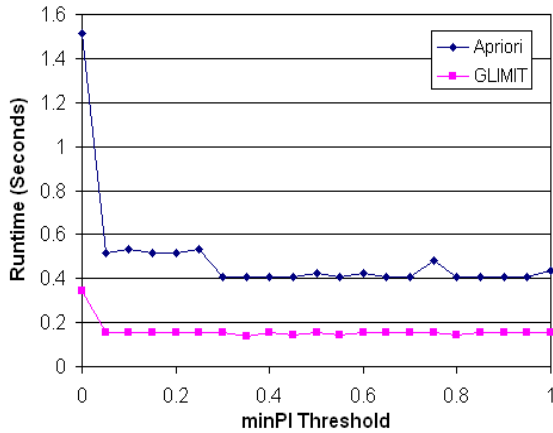
- **Non-Complex:** We removed duplicate items (object types) in the maximal cliques.
- **Complex w/o Negative:** We included *positive* types: if an object type A occurred more than once in a maximal clique, we replaced it with A and $A+$.
- **Complex w Negative:** The same as **Complex w/o Negative**, but we also included *negative* types.

The following table describes the resulting sets of maximal cliques we used for mining interesting patterns:

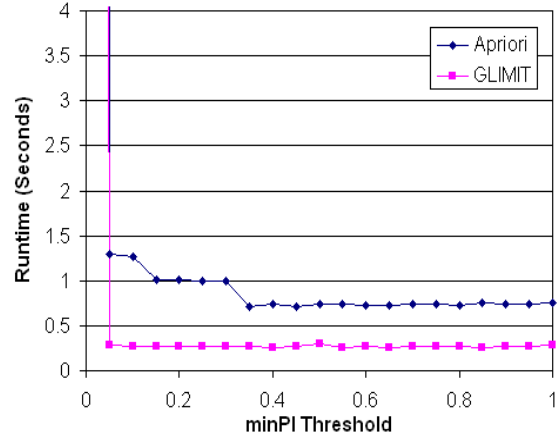
Maximal Clique Set	Items	Average Size (Transaction Width)
Non-Complex	12	1.87
Complex w/o Negative	21	2.69
Complex w Negative	33	13.69

¹<http://cas.sdss.org/dr6/en/tools/search/sql.asp>

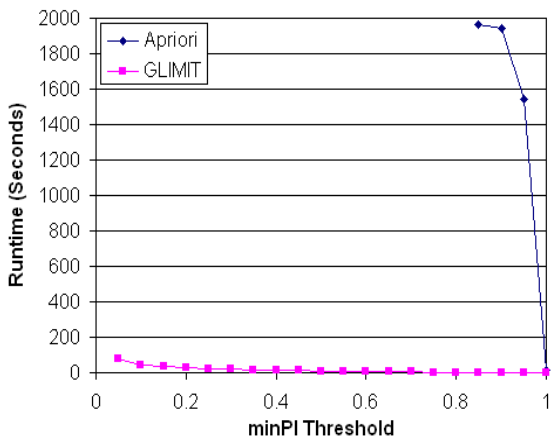
²The mega-parsec is an astronomical distance measure. See <http://csep10.phys.utk.edu/astr162/lect/distances/distcales.html> for details.



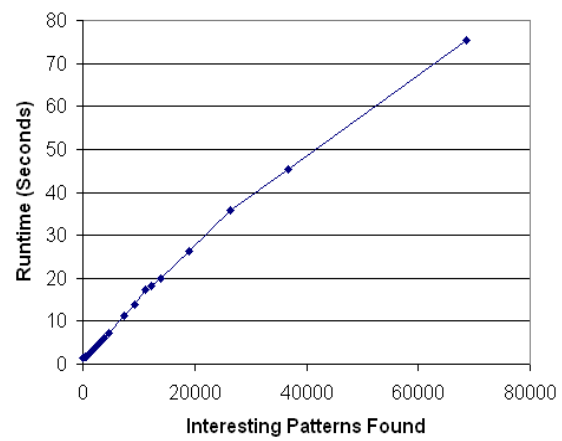
(a) Runtime on non-complex maximal cliques.



(b) Runtime on complex maximal cliques without negative patterns.



(c) Runtime on complex maximal cliques with negative patterns.



(d) The runtime of GLIMIT on complex maximal cliques with negative patterns, versus the number of interesting patterns found.

Figure 3. Computational Performance. The MinPI threshold was changed in increments of 0.05.

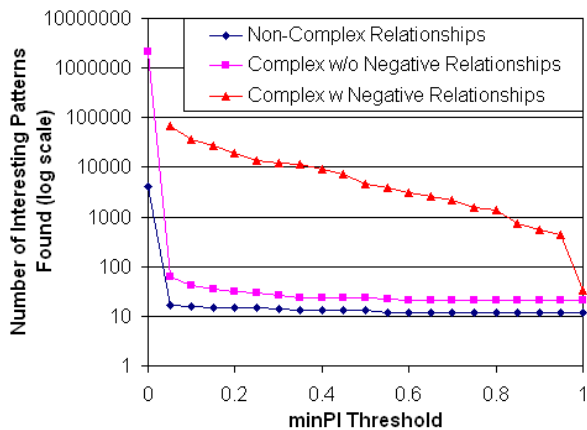


Figure 4. Number of interesting patterns found.

Note that the the “Complex w Negative” dataset is very large. It has 121, 506 transactions (like the others), but each transaction has an average size of 13.7.

Since most co-location mining algorithms are based on the Apriori algorithm, we use this as the comparison. That is, we evaluate both GLIMIT and Apriori for the interesting pattern mining task of Figure 2.

Figure 4 shows the number of interesting patterns found on the different sets of cliques.

Figures 3(a), 3(b) and 3(c)³ show the runtime⁴ of the pattern mining. It is clear that GLIMIT easily outperforms the Apriori technique. In particular, we draw the readers attention to the difference between the run-times when negative items are involved; namely, Figure 3(c). **For example, with a *minPI* threshold of 0.85, Apriori takes 33 min-**

³We set an upper limit of 2,000 seconds (33 minutes)

⁴Programs were implemented in Java and run on a laptop with a 2.0GHz Pentium 4M processor and Windows XP Pro.

utes (1967 seconds), while GLIMIT takes only 2 seconds. This is a difference of almost three orders of magnitude!

As can be seen from the previous table, the use of negative types increases the average transaction width substantially. This has a large influence to the runtime of the Apriori algorithm, due to the support counting step where all subsets (of a particular size) of a transaction must be generated. This is not true of GLIMIT, which runs in roughly linear time in the number of interesting patterns found, as can be seen in Figure 3(d). The “non-complex” and “complex w/o negative” datasets, due to their small average transaction width, may be considered easy. The “complex w negative” dataset is very difficult for Apriori, but very easy for GLIMIT. Indeed, even with a *minPI* threshold of 0.05 it takes only 76 seconds to mine 68, 633 patterns.

4 Related Work

Huang et al. [5] defined the co-location pattern as the presence of a spatial feature in the neighborhood of instances of other spatial features. They developed an algorithm for mining valid rules in spatial databases using an Apriori based approach. Their algorithm does not separate the co-location mining and interesting pattern mining steps like our approach does. Also, they did not consider complex relationships or patterns.

Monroe et al. [6] used cliques as a co-location pattern. Similar to our approach, they separated the clique mining from the pattern mining stages. However, they did not use maximal cliques. They treated each clique as a transaction and used an Apriori based technique for mining association rules. Since they used cliques (rather than maximal cliques) as their transactions, the counting of pattern instances is very different. They considered complex relationships within the pattern mining stage. However, their definition of negative patterns is very different – they used infrequent types while we base our definition on the concept of absence in *maximal cliques*. They also used a different measure, namely, *maxPI*.

Arunasalam et al. [3] used a similar approach to [6]. They proposed an algorithm called NP_maxPI which also used the MaxPI measure. The proposed algorithm prunes the candidate itemsets using a property of maxPI. They also used an Apriori based technique to mine complex patterns. A primary goal of their work was to mine patterns which have low support and high confidence. As with the work of [6], they did not use maximal cliques.

Zhang et al. [9] enhanced the algorithm proposed in [5] and used it to mine special types of co-location relationships in addition to cliques, namely; the *spatial star*, and *generic* patterns.

To the best of our knowledge, previous work has used Apriori type algorithms for mining interesting co-location

patterns. We use GLIMIT [8] as the underlying pattern mining algorithm as already discussed in Section 2.3.

To the best of our knowledge, no previous work has used the concept of *maximal cliques*.

5 Conclusion

In this paper we considered mining complex spatial co-location patterns. Most work in this area has used an Apriori style algorithm to do this. We showed that GLIMIT is a much better choice, especially when complex patterns are involved. Furthermore, we introduced the idea of maximal cliques, which is fundamental to our work. We argued that complex patterns only make sense in the context of maximal cliques. Using maximal cliques also allowed us to easily split the clique generation from the interesting pattern mining tasks and avoid redundant cliques.

References

- [1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proceedings of 20th International Conference on Very Large Data Bases VLDB*, pages 487–499. Morgan Kaufmann, 1994.
- [2] G. Al-Naymat, S. Chawla, and B. Arunasalam. Enumeration of maximal clique for mining spatial co-location patterns. tr 615. Technical report, School of Information Technologies, University of Sydney, Australia, 2007.
- [3] B. Arunasalam, S. Chawla, and P. Sun. Striking two birds with one stone: Simultaneous mining of positive and negative spatial patterns. In *Proceedings of the Fifth SIAM International Conference on Data Mining*, pages 173–182, 2005.
- [4] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *2000 ACM SIGMOD Intl. Conference on Management of Data*, pages 1–12. ACM Press, May 2000.
- [5] Y. Huang, H. Xiong, S. Shekhar, and J. Pei. Mining confident co-location rules without a support threshold. In *Proceedings of the 18th ACM Symposium on Applied Computing ACM SAC*, 2003.
- [6] R. Munro, S. Chawla, and P. Sun. Complex spatial relationships. In *Proceedings of the 3rd IEEE International Conference on Data Mining, ICDM 2003*, pages 227–234. IEEE Computer Society, 2003.
- [7] S. Shekhar and Y. Huang. Discovering spatial co-location patterns: A summary of results. *Lecture Notes in Computer Science*, 2121:236+, 2001.
- [8] F. Verhein and S. Chawla. Geometrically inspired itemset mining. In *ICDM*, pages 655–666. IEEE Computer Society, 2006.
- [9] X. Zhang, N. Mamoulis, D. W. Cheung, and Y. Shou. Fast mining of spatial collocations. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 384 – 393. ACM Press-New York, 2004.