

Enumeration of Maximal Clique for Mining Spatial Co-location Patterns

Ghazi Al-Naymat
School of Information Technologies
The University of Sydney
NSW 2006, Australia
ghazi@it.usyd.edu.au

Abstract

This paper presents a systematic approach to mine co-location patterns in Sloan Digital Sky Survey (SDSS) data. SDSS Data Release 5 (DR5) contains 3.6 TB of data. Availability of such large amount of useful data is an opportunity for application of data mining techniques to generate interesting information. The major reason for the lack of such data mining applications in SDSS is the unavailability of data in a suitable format. This work illustrates a procedure to obtain additional galaxy types from an available attributes and transform the data into maximal cliques of galaxies which in turn can be used as transactions for data mining applications. An efficient algorithm GridClique is proposed to generate maximal cliques from large spatial databases. It should be noted that the full general problem of extracting a maximal clique from a graph is known as NP-Hard. The experimental results show that the GridClique algorithm successfully generates all maximal cliques in the SDSS data and enables the generation of useful co-location patterns.

1. Introduction

With the rapid invention of advanced technology, researchers have been collecting large amounts of data on a continuous or periodic basis in many fields. This data becomes the potential for researchers to discover useful information and knowledge that has not been seen before. In order to process this data and extract useful information, the data needs to be organised in a suitable format. Hence data preparation plays a very important role in the data mining process.

The focus of this study is mainly on data preparation for Sloan Digital Sky Survey (SDSS) astronomy dataset [5]. The SDSS is the most motivated astronomical survey project ever undertaken. The survey maps in detail one-quarter of the entire sky, determining the positions and ab-

solute brightness of more than 100 million celestial objects. The first official Data Release (DR1) of SDSS was in June 2003. Since then there have been many new releases including the fourth major release (DR4) in June 2005 that provides images, imaging catalogs, spectra, and redshift. The latest version (DR5) with 3.6 TB of data, which includes measures of 200 million unique celestial objects.

Availability of such large amount of useful data is an obvious opportunity for application of data mining techniques to generate interesting information. However, while much research has been done by the astronomical researchers, a feeble effort has been made to apply data mining techniques on SDSS data. That is because the SDSS data format is not suitable for mining purposes, which are the main motivation of this paper.

As mentioned in [10] spatial databases store spatial attributes about objects, and hence, SDSS is a large spatial dataset as it contains many attributes for each object. One of the most significant problems in spatial data mining is to find object types that frequently co-locate with each other in large databases. The co-location means objects that are found in the neighborhood of each other. The proposed approach mines co-location patterns in SDSS data and uses these patterns to generate interesting information about different types of galaxies. In this work only the galaxies existed in SDSS is used. However, this approach could be generalised to be used with any other celestial objects.

The data preparation is in two folds. First, extracting the galaxies in SDSS data and categorising them into “Early” and “Late” type galaxies. Second, a new algorithm, *Grid-Cliqu* is proposed to generate co-location patterns (maximal cliques) from the data. A clique is any set of spatial objects such that all the objects in the set co-locate. A maximal clique is a clique which is not a subset of any other clique. Figure 1 depicts some examples of spatial co-locations. In this figure, a line between items indicates that items are co-located. The second column of Table 1 displays the maximal clique patterns, which are depicted in Figure 1.

The full general problem of extracting maximal cliques

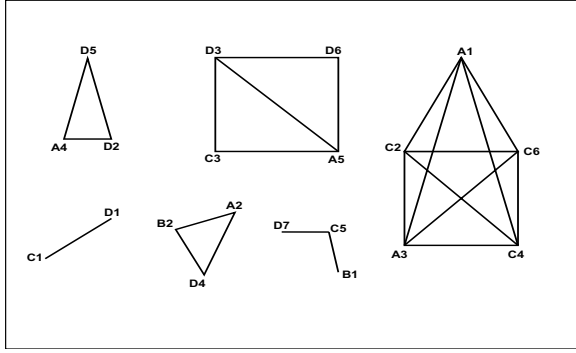


Figure 1. Cliques in a plane

Table 1. Maximal clique patterns

ID	Maximal Cliques	Transactions
1	C1,D1	C,D
2	C5,D7	C,D
3	B1,C5	B,C
4	A4,D2,D5	A,D+
5	A5,C3,D3	A,C,D
6	A5,D3,D6	A,D+
7	A2,B2,D4	A,B,D
8	A1,A3,C2,C4,C6	A+,C+

from a graph is known as NP-Hard. *GridClique* efficiently extracts maximal cliques in a given spatial database with capability to divide the space into grid structure based on a predefined distance. The use of the grid structure plays a vital role for reducing the search space.

The maximal cliques generated by *GridClique* could be represented as transactions as given in Column 3 of Table 1. The (+) sign in the transaction column means more than one item from the same type are included. For example, clique (A5,D3,D6) contains three objects, two of them from the same type, namely D. Hence, the transaction is given by A,D+, which can be used by association rule mining techniques. Association rule mining techniques that proposed by [1, 3] can generate useful rules, which will be interpreted as relationships between objects. Arunasalam et al. [3] classified spatial relationships into four different types. (1) “Positive”, which describes the existence of object in the neighborhood of another object. For example, the existence of spiral galaxy type Sa implies the existence of spiral galaxy type Sb. (2) “Negative”, which describes the absence of an object in the neighborhood of another object. For example, Elliptic galaxies tend to exclude spiral galaxies. (3) “Self-Co-location”, which reflects the presence of many instances of the same type in the same neighborhood. For

example, Elliptical galaxies have a tendency to co-locate closely to each others. (4) “Complex”, which is a combination of the previous relationships.

1.1 Basic Definitions and Concepts

This section gives a brief definitions of the concepts that are used in this paper.

Definition 1 Clique: A set of spatial objects S is said to be a clique if every object in S co-locates with every other object in S .

For example, in Figure 1, (A4,D2,D5) form a clique as each object co-locates with each other. Similarly (C5,D7) form another clique.

Definition 2 Maximal Clique: A clique that is not a subset of any other clique in the same graph.

In Figure 1, (A4,D2,D5) form a maximal clique as it is not a subset of another clique. However, (A2,D4) is not a maximal clique since it is a subset of the clique (A2,B2,D4).

Definition 3 Clique’s Cardinality: It is the number of clique’s members. In other words, it is the clique’s size.

1.2 Contributions and Paper Outline

This paper provides three major contributions to the field of astronomy and spatial mining. These contributions are summarized as follows:

1. An efficient algorithm called *GridClique* is proposed to generate all maximal clique patterns that exist in large spatial databases.
2. The experiments confirm the utility and usefulness of the *GridClique* algorithm in generating interesting patterns from large spatial datasets.
3. The benefit of using complex relationships in finding valid rules is proposed.

The rest of the paper is organized as follows: Section 2 demonstrates the related work in spatial data mining. Section 3 talks about the extracted data and the galaxy categorisation process. Section 4 gives in detail about the proposed algorithm *GridClique*. Section 5 illustrates the experimental setup and the results discussion. Section 6 concludes the paper.

2. Related Work

The expression spatial data was defined as location-based data by Judd [7]. However, a simple definition of

a spatial database is a collection of data that contains information on an observable fact of interest, such as forest condition or pollution, and the location of an observable fact on the Earth. Spatial data consists of two types of attributes; the normal attributes, which are defined as non-spatial attributes, and spatial attributes that describe instances's location and shape, ...etc.

In addition, there are different types of spatial patterns defined by the relations among objects, such as *clique* patterns. Generally, when at least two objects are located in the same area, the relation between them is called co-location. Therefore, co-location relations in the database show the subsets of object types and their locations within a given distance. For example, clique determines subset of items that are co-located in the same neighborhood. [3].

Huang et al. [6] defined the co-location pattern as the presence of spatial feature in the neighborhood of instances of other spatial features. They developed an algorithm for mining valid rules in spatial databases, and they found useful and good rules based on some measures. However, they were looking for positive rules without any complexity. Their method does not give complex rule such as $A \rightarrow B+$ or $B \rightarrow -C$. In fact, they did not obtain that because they prune most items based on their prevalence measure "participation index".

Furthermore, for the purposes of finding the maximal clique, Huang's method [6] finds cliques that do not have any of their members participating in any other cliques. This violates the definition of the maximal clique, which is a group of items having a common relationship (distances between them $\leq t$), even though a subset of that clique appears in another clique, but not all clique's members. Assume clique-1 contains three items $\{A1, B1, C1\}$ and clique-2 contains $\{A1, B2, D1\}$, item $\{A1\}$ is a common item between the two cliques, which is possible. In the previous example, Huang's method counts them as one clique because of the use of the prevalence measure. Therefore, they lose one clique which might give more details later on. In contrast, this study allows some redundancy as one of the reasons to form complex rules, therefore, it counts these cliques as two.

Morimoto [9] defined new co-location patterns called k-neighboring class set. He used in his method the number of pattern instances as a prevalence measure. The latter does not satisfy the anti-monotone property because in this work there are some overlapping. As a result, the number of instances increases by the size of the pattern. Hence, Morimoto [9] dealt with this case by using a constraint to acquire the same benefit as the anti-monotone property. The constraint was "Any point object must belong to only one instance of a k-neighboring class set". Based on this constraint he faced a problem, which is obtaining different support of the same k-neighboring class set

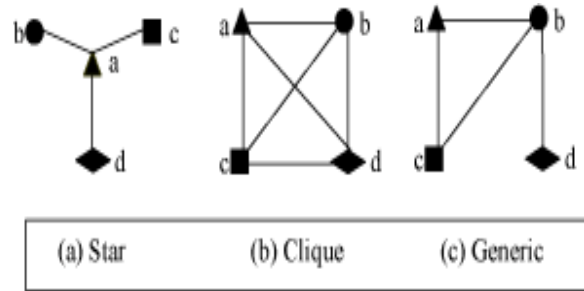


Figure 2. Three different spatial patterns.

because the selectivity process counts the instances from the same feature as one instance. In fact, if the order of the instances is changed, then both the value of the support and the co-location patterns will be changed. Consequently, Morimoto's [9] method also can not give the opportunity to generate complex rules because of that constraint.

Zhang et al. [13] enhanced the proposed algorithm in [6]. However, their approach is finding spatial star, clique, and generic patterns as shown in Figure 2. In [13] they used a grid structure over space and they defined relationships among objects if the distance between them is \leq threshold ϵ . Their method of finding object's neighbors is done by extending object's coordinates by ϵ to form a disk, then all grids that intersect with this disk will be hashed and all items inside these grids will be checked using the Euclidean distance to make sure that these items are close to the center of the star. In summary, their method for finding star patterns based on two steps: hashing and mining. However, these two steps were not enough to find clique patterns. Therefore, they added another mining step to find clique patterns. The proposed algorithm *GridClique* finds maximal clique patterns in two steps after placing the objects on the plain.

Many researchers have used association rule mining techniques in finding relationships between items and objects [2]. In this research, association rule mining technique can be used to extract valid rules, and introduce them to the astronomy domain, because *GridClique* algorithm generates transactional dataset, which is the desirable format for rule mining techniques. However, this beyond the scope of this paper.

3. Data Extraction and Categorisation

This section illustrates the method of extracting attributes from the SDSS database and using them to categorise galaxy objects. A view called *SpecPhoto* which is derived from a table called *SpecPhotoAll* is used. The latter is a join between the *PhotoObjAll* and *SpecObjAll* tables. In

Table 2. The SDSS schema

No	Field name	Field description
1.	specObjID	Unique ID
2.	z	Final RedShift
3.	ra	Right ascention
4.	dec	Declination
5.	cx	x of Normal unit vector
6.	cy	y of Normal unit vector
7.	cz	z of Normal unit vector
8.	primTarget	prime target categories
9.	objType	object type : Galaxy =0
10.	modelMag_u	Ultraviolet magniutde
11.	modelMag_r	Red Light magnitude

other words, *SpecPhoto* is view of joined *Spectro* and *PhotoObjects* that have the clean spectra.

The concern was to extract only the galaxy objects from the SDSS using parameter(object type=0). The total number of galaxy type objects stored in the SDSS catalog is (507,594). However, to ensure the accuracy for calculating the distance between objects and the earth which leads to calculate the X , Y , and Z coordinates for each object, some parameters are used, such as $zConf < 0.95$ (the rigid objects) and $zWarning = 0$ (correct RedShift). Therefore, the number of objects is reduced to (442,923).

SDSS release 5 provides a table called *Neighbors*. This table contains all objects that are located within 0.5 arcmins, this makes it not useful in this study because there is no ability to choose any distance that would form the neighborhood relation between objects. For example, in our experiments (1, ..., 5) mega-parsec (distances) are used as the thresholds to check whether objects are close to each other or not. Table 2 discloses the extracted fields from the SDSS (DR5) that used during the preparation process.

- **Data extraction:** The data was obtained from SDSS (DR5) [11]. This data is extracted from the online catalog services using several SQL statements and tools, which offered by the catalog. These tools are accessible from the SDSS site ¹.

- **Data transformation:** The result from the previous task should go next through the data transformation process, which is the last task of the data preparation process and it makes it possible to use the extracted dataset. This process plays an important role in accessing the dataset by *Oracle10g* database using some SQL commands. After preparing the corrected data, the final step is uploading this data to ODBMS which is used in this research.

Few tables were created to store the extracted data,

¹<http://cas.sdss.org/dr5/en/tools/search/sql.asp>

also a few stored procedures were created to do some other special tasks such as, categorising galaxy objects and putting the data into the right format.

- **New attributes creation:** With all the necessary fields, the next step is to calculate the exact value of the X , Y , and Z coordinates which are not explicitly showed in the SDSS data. Firstly, the distance D between objects and the earth, is calculated. It is calculated using Hubble's law and z value of each object as what Equation 1 shows. Secondly, by considering the unit vectors cx , cy , and cz , and multiplying them by the D , the value of X , Y and Z coordinates are calculated as in Equations 2, 3, and 4, respectively.

$$D \approx \frac{c \times z}{H_o} \quad (1)$$

Where c is the speed of light, z is the object RedShift, and H_o is Hubbles' constant. Currently the best estimate for this constant is $71 \text{ kms}^{-1} \text{ Mpc}^{-1}$ [4, 8].

$$X = D \times cx \quad (2)$$

$$Y = D \times cy \quad (3)$$

$$Z = D \times cz \quad (4)$$

- **Galaxies Categorisation:** Different parameters were used to categorise galaxy types. Based on the difference between Ultraviolet U and Red light magnitude R , galaxies are categorized as either "Early" or "Late". If the difference is greater than or equal to 2.22 the galaxy is "Early", otherwise "Late". The value of the r -band *Petrosian* magnitude indicates if the galaxy is "Main" (close to the earth) or "Luminous Red Galaxies" (*LRG*). That is by checking the value of r -band. if $r\text{-band} \leq 17.77$, that indicates that the object is Main galaxy otherwise it is LRG [12]. The four galaxy types that found are **Main-Late**, **Main-Early**, **LRG-Late**, and **LRG-Early**.

4. Generating Maximal Cliques

The aim of this algorithm is to extract the maximal clique patterns that exist in any undirected graph. It is developed using an index structure through grid implementation. Table 3 contains 10 objects and their X and Y coordinates, this information will be used to explain the functionality of the algorithm in the following sections. It should be noted that SDSS is three dimensional dataset, but in the example two dimensions are used for the sack of simplicity.

Input: Set of points (P_1, \dots, P_n) , Threshold t

Output: A List of maximal cliques in transactional data format.

$[CliqueID, Clique'sMembers]$

Ex: $[1, \{A, B, C\}]$

Step 1: Generating grid structure.

1. $d \leftarrow t$
2. $GridMap \leftarrow \phi$
3. $PointList \leftarrow \{P_1, \dots, P_n\}$
4. $S = PointList.Size()$
5. for each $P_i \in PointList \dots i \leq S$
6. Get the coordinates of each point $P^{k_x}, P^{k_y}, P^{k_z}$
7. Generate the composite key (GridKey).
8. if $GridKey \in GridMap$
9. $GridMap \leftarrow P_i$
10. else
11. $GridMap \leftarrow new\ CompKey$
12. $GridMap.CompKey \leftarrow P_i$

Step 2: Getting the neighbors.

1. for each $p_i \in GridMap$
2. $p_i.list \leftarrow \phi$
3. $NeighborGrids \leftarrow \phi, NeighborList \leftarrow \phi$
3. Generate the neighborhood grids for p_i .
4. if $NeighborGrids_i.size() > 1$ then
5. for each $p_j \in NeighborGrids_j$
6. if $dis(p_i, p_j) \leq d$
7. p_i, p_j are neighbors
8. $p_i.list \leftarrow p_j$
9. $NeighborList \leftarrow p_i.list$

Step 3: Prune non neighbor items.

1. $TempList \leftarrow \phi, CliqueList \leftarrow \phi$
2. for each $Record_i \in NeighborList$
3. $RecordItems \leftarrow \phi$
4. $RecordItems \leftarrow Record_i$
5. for each $p_i \in RecordItems$
6. for each $p_j \in RecordItems$
7. if $dis(p_i, p_j) \leq d$
8. p_i, p_j are neighbors
9. $Templist \leftarrow p_j$
10. $CliqueList \leftarrow Templist$

Figure 3. Pseudocode of the GridClique algorithm.

Table 3. Example: Dataset of two dimensions.

Object type	X-Coordinate	Y-Coordinate
A1	2.5	4.5
A2	6	4
A3	2	9
B1	1.5	3.5
B2	5	3
B3	5	4
C1	2.5	3
C2	6	3
D1	3	9
D2	7	1.5

4.1 GridClique algorithm

Figure 3 reveals the pseudocode of the *GridClique* algorithm. This section also shows an example of how the algorithm works. By assuming that all objects are spatial point objects, items locations will be the same as what Figure 4(a) depicts.

Edges in each subgraph are generated by calculating the distance between objects and if the distance is $\leq t$ the edge will be created. Each subgraph forms a pattern. Therefore, all the results of this algorithm are patterns having objects that are close to each other. The following are the algorithm steps:

1. Dividing the space as a grid structure and concurrently places each point into its particular grid cell. The grid cell size is $d \times d$, where d is the same as the threshold t and it is one of the inputs for this algorithm as shown in Figure 4(b).
2. Finding each object's neighbor list. This step is the most important one, and it is the most crucial step for the complexity issue. Similarly, it uses the same technique to check the neighborhood feature between points based on the Euclidean distance. However, the number of checked points depends on the density of the grid and the content of the nine neighbor cells². According to the example in Figure 4, also because this sample contains 10 objects, a list for each object is created except for those objects that are located lonely. Our concern was just to find co-location patterns that have number of members ≥ 2 (i.e. cardinality ≥ 2). Because one object will not form the desired relationship. Consequently, no need to count objects that do not have connections (i.e. relationship) with at least one another object. But in our example all objects share relationships with others.

²Number of neighbor cells is 27 if the data is 3D.

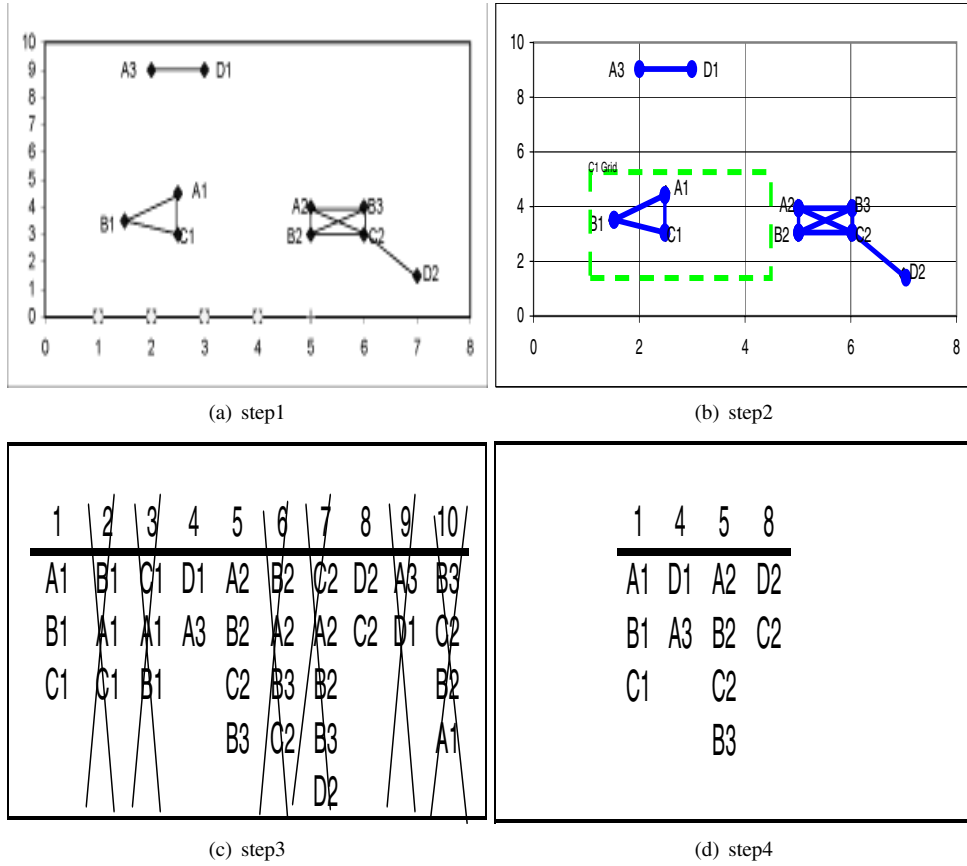


Figure 4. Steps for finding maximal cliques

For example, object $\{A1\}$ has a relationship with $\{B1, C1\}$ and object $\{A2\}$ with $\{B2, B3, C2\}$. It can be seen that these objects share the same location, this means $\{A1, B1, C1\}$ are co-located because the distance between them is \leq threshold t . Figure 4(c) shows some redundancy and this gives us the chance to prune the complete list in the next step when one of its objects violates the co-location conditions (distance between objects is $\leq t$), because same members from another list can be obtained from another list.

3. Pruning any neighbor list that contains at least one object violates the co-location condition. For example, list 7 is pruned because two of its members $\{A2, D2\}$ are not close to each other as shown in Figure 4(c).

As a result of the previous steps, list of maximal cliques will be formed. For example, $\{A1, B1, C1\}$ forms a maximal clique, and so forth for lists (4, 5, 8) in Figure 4(d).

4.2 GridClique algorithm analysis

This section discusses the GridClique algorithm completeness, correctness, and complexity.

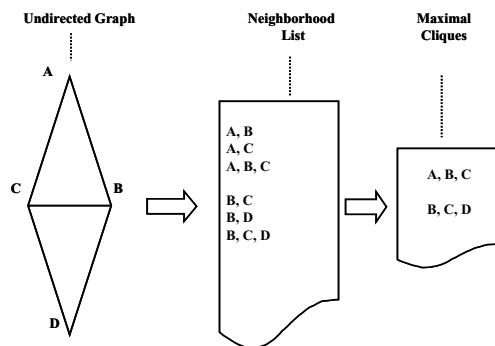


Figure 5. Example of two maximal cliques used to show the correctness of the proposed algorithm.

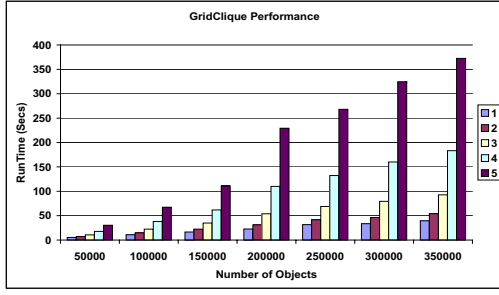


Figure 6. GridClique’s runtime using 5 different distances.

Completeness: All objects in neighbor lists appear as set or subset in maximal clique lists. After acquiring the entire neighbors for each point, another check among these neighbors is done to assure that all points are neighbors to each other. Intuitively, doing that results to have repeated neighbor lists. Therefore, this ensures finding all maximal cliques in any given graph.

Correctness: Every subset of a maximal clique appears in the neighbors list. Thus, all maximal cliques that appear in maximal clique’s list will not be found as a subset in another maximal clique. That is, the definition of maximal clique. Figure 5 displays an undirected graph and the neighborhood list and the existed maximal clique patterns. It is very clear that the pair $\{A, D\}$ does not appear in the neighborhood list, because the distance between A and D does not satisfy the threshold.

As a result, the pair $\{A, D\}$ will not be included in the maximal cliques’ list. In other words, any subset of any maximal clique appears in the neighborhood list and it will not appear as an independent maximal clique. By this, the correctness of the proposed algorithm is shown.

Complexity: Firstly, assume there is an N points and d cells, and assume that all points are uniformly distributed. Hence, on average there is N/d points per cell. Also, assume each cell has l neighbors. Then to create the neighborhood list of one point $l(N/d)$ points need to be examined to check if they are within distance t . Since the total number of points is N , thus the cost is $O(N^2l/d)$. And since $d \gg l$, an assumption, that this part of the algorithm is sub-quadratic, can be stated.

Secondly, the pruning neighborhood lists stage. Again assume that on average the length of each neighborhood list is k . Then for each neighborhood list, k other lists have to

be examined to check if a point is in others neighborhood list or not. Therefore, for each point, k other neighborhood lists are examined as well as within each one, up to k points will be checked. Consequently, the cost is $O(N(k^2))$.

Finally, the total cost is the cost to put the points in cell ($O(N)$), the cost to create the neighborhood lists $O(N^2l/d)$, and the cost to prune the lists $O(N(k^2))$. The total complexity of the algorithm is $O(N(Nl/d + k^2 + 1))$.

5. Experiments and Results Discussion

Experiments are carried out to confirm the achieved results from the proposed algorithm using SDSS data.

5.1 Experimental setup

All experiments were carried out on a Windows XP operated PC with a Pentium-4 (3 GHz) processor and 1 GB main memory. The data structures and algorithm were implemented in Java and compiled with the GNU compiler.

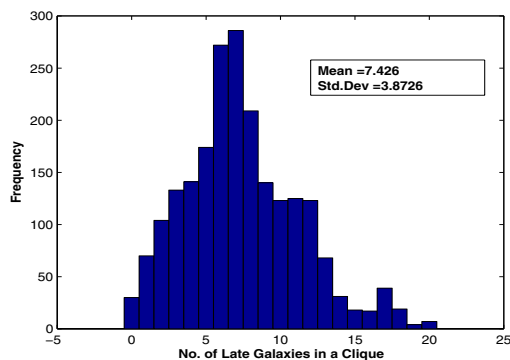
5.2 Scalability of GridClique algorithm

Figure 6 demonstrates the runtime of the *GridClique* algorithm with various numbers of objects (galaxies) and distance values. It illustrates that the runtime increases slightly as the number of objects and distance increase. The distance is increased by 1 Mpc every time, whereas the number of objects is increased by 50K objects. The maximum number of records was 350K.

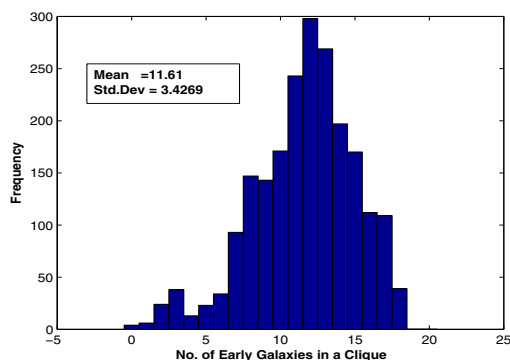
To explain further, when the distance increases the grid size increases. Also by increasing number of objects at the same time, it allows more objects to appear in the same grid’s cell or in the neighbor grids area. Therefore, the two factors (distance, records number) affect the runtime of the *GridClique* algorithm.

5.3 Galaxy types in large cliques

We applied the *GridClique* algorithm on the “Main” galaxies extracted from SDSS to generate maximal cliques with neighborhood distance as 4 Mpc. We selected the cliques with the largest cardinality(22). Figure 7 shows the distribution of “Early” and “Late” type galaxies in the reported cliques. These results show that large cliques consist of more “Early” type galaxies (Elliptic) than “Late” type galaxies (Spiral). This conforms to the patterns given by [5].



(a) MainLate



(b) MainEarly

Figure 7. The existence of galaxies in the universe.

6. Conclusion

A systematic approach to mine co-location patterns in Sloan Digital Sky Survey (SDSS) data, which contains 3.6 TB of data, is proposed. In addition, this study demonstrated a procedure to obtain additional galaxy types from available attributes and transformed the data into maximal cliques of galaxies that can be used as transactions for data mining applications. The proposed algorithm *GridClique* enumerates maximal clique patterns efficiently. These patterns are used as input data to an association rule mining technique to generate useful rules to the astronomy domain. In the preliminary results of applying association rules mining technique, some interesting rules were found. For example, the existence of an Elliptic galaxy entails the absence of a spiral galaxy, which is a well known fact in astronomy³.

³This work will be disclosed as part of the future work. The initial results are excluded due to the space limitation.

Acknowledgments

The author wishes to acknowledge the generosity of A/P. Sanjay Chawal and Bavani Arunasalam⁴ for their valuable support in this work.

References

- [1] R. Agrawal, T. Imielinsk, and A. Swami. Mining association rules between sets of items in large databases. In *SIGMOD '93: Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pages 207–216, New York, NY, USA, 1993. ACM Press.
- [2] 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.
- [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] D.N.Spergel, M.Bolte, and W.Freedman. The age of the universe. *Proceedings of the National Academy of Science*, 94:6579–6584, 1997.
- [5] J. Gray, D. Slutz, A. S. Szalay, A. R. Thakar, J. vandenBerg, P. Z. Kunszt, and C. Stoughton. Data mining the sdss sky-server database. Technical Report MSR-TR-2002-01, Microsoft Research, 2002.
- [6] 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*. ACM Press, New York, 2003.
- [7] D. D. Judd. What's so special about spatial data?. retrieved april 21, 2005 from <http://www.eomonline.com/archives/jan04/judd.html>, 2005.
- [8] H. M. and S. Churchman. Hubble's law. retrieved march 12, 2005, from <http://map.gsfc.nasa.gov/>, 1999.
- [9] Y. Morimoto. Mining frequent neighboring class sets in spatial databases. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 353 – 358. ACM Press-New York, 2001.
- [10] S. Sekhar and S. Chawla. *Spatial Databases:A Tour*. Prentice Hall, 2003.
- [11] S. D. S. Survey. Sdss - sloan digital sky survey. retrieved august 5, 2005 from <http://cas.sdss.org/dr5/en/help/download/>, 2006.
- [12] V.J.Martin and E.Saar. *Statistics of the Galaxy Distribution*. Chapman and Hall/CRC, 2002.
- [13] 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.

⁴Sanjay and Bavani with the University of Sydney.