Mining Frequent Spatial-Textual Sequential Patterns

Student Name: Krishan Kumar Arya

IIIT-D-MTech-CS-DE-14-MT12041
June 18, 2014

Indraprastha Institute of Information Technology
New Delhi

Thesis Committee
Dr. Vikram Goyal (Chair)
Dr. Ashish Sureka
Prof. R. K. Aggarwal

Submitted in partial fulfillment of the requirements
for the Degree of M.Tech. in Computer Science,
with specialization in Data Engineering

Keywords: Sequential Pattern Mining, PrefixSpan, Trajectory, Spatial-Textual, Textual-Spatial, Location Granularity, External Memory Algorithm, Dissimilar sequences.

Certificate

This is to certify that the thesis titled **"Mining Frequent Spatial-Textual Sequential Patterns"** submitted by **Krishan Kumar Arya** for the partial fulfillment of the requirements for the degree of *Master of Technology* in *Computer Science & Engineering* is a record of the bonafide work carried out by him under my guidance and supervision in the Data Engineering group at Indraprastha Institute of Information Technology, Delhi. This work has not been submitted anywhere else for the reward of any other degree.

Dr. Vikram Goyal
Indraprastha Institute of Information Technology, New Delhi

Abstract

Penetration of GPS-enabled devices has resulted into generation of a lot of Spatial-Textual data, which can be mined/analyzed to improve various location-based services. One such kind of data is activity-trajectory data, i.e. a sequence of locations visited by a user with each location having a set of activities performed by the user. In this thesis, we propose a mining framework along with algorithms for mining activity-trajectory data to find out Spatial-Textual sequencial patterns. The proposed framework is flexible in the sense that any algorithm from the existing sequence mining algorithms can be used as a core algorithm in our framework. We design and implement three different algorithms, namely, Spatial-Textual sequence mining algorithm, Textual-Spatial sequence mining algorithm and Hybrid sequence mining algorithm and find out their effectiveness for different location granularity and sensitivities. The experiment results shows Spatial-Textual approach outperforming other approaches in case of better location selectivity in the data. We also observe that the Spatial-Textual approach is able to handle much larger activity-trajectory data as compared to other approaches.

Acknowledgments

I wish to express my gratitude to all people involved in the successful completion of my M.Tech. Thesis, especially to my Thesis mentor **Dr. Vikram Goyal** for his guidance and critical reviews. Through his creativity, energy and expertise, he has given me a great appreciation for academic research and the joy of conducting original research. I would also like to thank him for his patience even when work and a medical emergency distracted me from my academic work. This thesis is totally dedicated to your creativity and enthusiasm of doing work.

My sincere thanks to **Prof. Shamkant B. Navathe** and **Prof. Sushil K. Prasad** who were very generous to devote their precious time, sharing their knowledge with me, and helping me out in every possible manner.

I would like to thank **Shilpi Jain** (M.Tech.-Data Engineering, 2012-2014) for her tremendous help in initial phase of this thesis project when I didnot know what is going to be the final outcome of this thesis.

A special thanks to all faculty members of **IIIT-Delhi**, especially to **Data Engineering** Department for imparting the best of knowledge and moulding my future. I am also thankful to all my batchmates and friends for being with me at each step when I need their support.

And finally, my deep gratitude to my family members for their unflinching emotional support during the whole period.

Krishan Kumar Arya
Indraprastha Institute of Information Technology, New Delhi

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Research Motivation and Aim

Sequential pattern mining is an important and very active research topic in data mining, with various applications, such as customer shopping transaction analysis, mining web logs, mining DNA sequences, mining e-bank customers etc. In Sequential Pattern Mining we are interested in finding frequent subsequences in sequence database. An example sequential pattern might be one in which customers typically bought a Bread and Butter, followed by the purchase of a Beer and Diaper, and then Milk.

There have been lot of sequential pattern mining algorithms proposed in the previous work that mine frequent subsequences from a large sequence database efficiently. These algorithms work in the fashion that they mine the entire database and obtain the results. The GSP(**G**eneralized **S**equential **P**atterns) algorithm generates patterns, scans each data sequence in the database to compute the frequencies of candidates, and then identifies candidates having enough supports as sequential patterns. The PrefixSpan (**Prefix**-projected **S**equential **pa**ttern mining) algorithm uses the pattern-growth methodology and finds the frequent patterns by scanning the sequences once. The database is then prefix-projected, according to the frequent items, into several smaller databases. and then finally, the complete set of frequent sequential patterns is found by recursively growing subsequence fragments in each projected database. The SPADE (**S**equential **PA**ttern **D**iscovery using **E**quivalence classes) algorithm searches in the lattice formed by id-list intersections and completes mining in three passes of database scanning. The data that can be provided to these algorithm can be of any type like it can be customer shopping sequences where the dataset contains sequences of purchasing behaviour of users, it can be sequences of web logs of users, it can be sequences of locations of users or movement of an object (i.e. trajectory) which can be sampled at consecutive timestamps (e.g., with the use of Global Positioning System (GPS) devices).

In this thesis, We are working on the data which is trajectories of objects with some activities i.e. we are having spatial locations of movements of objects or users and some activities associated with each of these locations means it describe the data as user can go from one location to another and do some activities. Here we call this type of dataset as Spatial-Textual dataset. The

increasing generality of location-acquisition technologies (GPS, GSM networks, etc.) is leading to the collection of large Spatial-Textual datasets and to the opportunity of discovering usable knowledge about movement behaviour, which furthers novel applications and services.

The problem here is to find frequent sequential patterns which tells that users or objects usually follow these trajectories by doing these sort of activities. These type of patterns are useful in suggesting future routes to the users for doing specific type of activities. Here, we move towards this direction and develop an extension of the sequential pattern mining paradigm that analyzes the trajectories of moving objects. we introduce trajectory patterns as concise descriptions of frequent behaviours, in terms of both space (i.e., the regions of space visited during movements) and time (i.e., the time difference between movements) and some events/activities. The movement routes of most objects are not defined but they often have a pattern. That pattern will help in various application domains like prediction of next terrorist activities, prediction of natural calamities, commercial applications etc.

The locations can be of any type like the actual location (i.e. exact latitude and longitude) or it can be the zip-code of the area or it can be the city name/state name/country name. For assigning label to the locations DBScan algorithm is used which work for given ρ value and minimum points. It gives us the clusters of locations and assign labels to them. We use these labels in the dataset for locations. Grid-based algorithm is also used for labeling the locations.

The task of mining such patterns becomes extremely challenging because of the association of events, and also both the temporal and the spatial dimensions have to be taken into account. The problem of frequent pattern mining is very old but still the problem of mining such data has not received a lot of attention.

For mining such type of patterns, an intelligent framework is proposed which scans the dataset and concludes the way it has to choose for mining frequent patterns. In this framework, any of the Sequential Patterns Mining algorithm can be used as the core algorithm and on top of this two new algorithms are being proposed called Spatial-Textual Mining and Textual-Spatial Mining. So this framework contains three algorithms for each core algorithm and this framework chooses the algorithm by scanning the dataset for one time by seeing the selectivity of either Spatial space or Textual space. If the selectivity of Spatial space is less for given support then it will choose Spatial-Textual Mining algorithm else if the selectivity of Textual space is less for given support then it will choose Textual-Spatial Mining algorithm otherwise it will choose Hybrid algorithm. After running these three algorithm over some Spatial-Textual datasets, it comes out that Spatial-Textual Mining algorithm may outperforms (in terms of execution time) Hybrid algorithm for some data having spatial selectivity less for given support. The Textual-Spatial Mining algorithm doesnt work good in either case because calculating Textual patterns is almost same as calculating overall patterns because Textual patterns doesnt prune the dataset

much.

Since the Spatial data alone is less in size rather than the full Spatial-Textual data so Spatial data can be fit into memory and spatial patterns can be found and then the full data is cut down in size on the concept that locations which are not frequent in spatial data cannot be frequent into spatial-textual data. So this approach is good in terms of memory requirement where we can fit the data into memory and find the patterns for the full data. Moreover if the spatial data is also not fitting into memory then external memory algorithm is also implemented for this. Since trajectories can be nearby to each other or can be far away to each other. By the use of this property of trajectories, we have chosen dissimilar trajectories into the chunks to run external memory algorithm efficiently. If we find patterns into dissimilar trajectories than the less number of false positives will be generated and it will reduce the execution time of the external memory algorithm.

Spatial-Textual database contains the trajectory of users associated with activities. Mining this type of data can be useful in various applications. Some of which are mentioned here:

- **Customer Shopping Sequences**

  Customers do shopping at various places. We can have data of customers who go at various locations and buy some products. By mining Spatial-Textual patterns, one can take a decision to open shopping mall or do publicity at various places for their products to achieve good profit.

- **Medical Treatment**

  People usually go at various hospitals or clinics for medical treatment of various diseases. One can use this data to get ST patterns to open hospitals/clinics at various locations of particular type like Dental, ENT etc.

- **Natural Disaster**

  Various types of natural disaster happens at different-different places. If we come to know the patterns of happening of these natural disaster then we can take further decision to provide some medical support to the people there and to warn people about the future natural disaster.

- **Science & Engineering Processes**

  Science and Engineering processes has various type of fields which contains locations and items type of data such as data from sensor networks. By mine this sort of data we can take a decision for the sensors like at which place which type of sensor is required.

- **Stocks and Markets**

Stocks are the major issue in today's world. A user involve in various types of market shares of various places and gains profit/loss. If we mine this sort of data to get users' patterns of locations with shares then this information can be used to take further decision.

- **Telephone & Calling Patterns**

  Users do call from different-different locations to various users. This type of data can be mined to extract patterns of users' call habits and from this info we can take various decisions like who is friend of whom, which group call frequently and mostly etc. from this various schemes can be launched for those users by the service providers.

- **Weblog Click Streams**

  Users use web services to surf various sites from various locations. So mining this type of data to extract users behavior of using sites from locations can be used by the internet provider/site owner to give good network connectivity at those locations and if user is using e-commerce site from different-different locations to buy some products then this info can be used for publicity purpose for the users for some products at some locations.

- **Sale Campaign Analysis**

  To do publicity for sale purpose, various patterns of purchasing habits of users are useful. To get these types of patterns, users data of buying products from different locations is used.

- **e-Bank Customers**

  Nowadays Bank customers use online banking to do various sort of transactions or various banking activity. Bank can do publicity of their scheme/service at various places to various users by mining the patterns of users behavior of using e-banking from various locations.

- **Spread of Disease**

  In today's world, various types of diseases are spreading at various different-different locations. If we come to know about the patterns of these type of diseases, then at those locations decisions can be taken to provide medical support of particular type.

- **Crime Hot spot**

  Crime of particular type is spreading at various locations. If patterns for these sort of crimes can be identified then immediate actions can be taken to provide police support to avoid future crimes and medical support can be provided.

- **Mobile Ad-Hoc Networks**

  As the recent denial-of-service attacks on several major Internet sites have shown us, no open computer network is immune from intrusions. The wireless ad-hoc network is particularly vulnerable due to its features of open medium, dynamic changing topology, cooperative algorithms, lack of centralized monitoring and management point, and lack of

a clear line of defense. Many of the intrusion detection techniques developed on a fixed wired network are not applicable in this new environment. So if we come to know about the patterns of denial of service of various sites from various locations then strict actions can be taken to stop such sort of activities.

- **Threat Assessment**

  Cyber attack is the major issue these days. It is being done by some attacker for various site such as mail accounts, social networking sites etc. from different-different locations. If these type of patterns can be identified then strict actions can be taken to stop an detect these sort of cyber attack in future.

- **Searching patterns in Geo-Spatial Data**

  People usually go from one location to another and at each and every location, they do some sort of activities. If patterns for these type of data can be known then user can get recommendations for future locations according to his/her interest of doing particular activity mean user can visit these locations and do these type of activities there.

Chapter 2

Related Work and Research Contributions

2.1 Related Work

This is most related to pattern discovery from sequential data, which include time series, event sequences, and Spatial-temporal trajectories.

In this section we briefly discuss about some related work that has been done in this field. The sequential pattern mining problem was first addressed by Agrawal and Srikant [1] in which they have given a simple algorithm to mine frequent patterns by candidate-generation-and-test-approach which may not be efficient in mining large sequence databases having numerous patterns and/or long patterns.

Another approach was given by Jian Pei et al. [4] in which they proposed a projection-based, sequential pattern-growth approach for mining of sequential patterns. In this approach, a sequence database is recursively projected into a set of smaller projected databases, and sequential patterns are grown in each projected database by exploring only locally frequent fragments. To avoid checking every possible combination of a potential candidate sequence, we need to fix the order of items within each element so that if one follows the order of the prefix of a sequence and projects only the suffix of a sequence, one can examine in an orderly manner all the possible subsequences and their associated projected database. The concept of prefix has been introduced for doing this. The major cost of prefix span is in database projection so to reduce this cost and to increase the performance they have introduced the concept of pseudoprojection in which an index position pointer may save physical projection of the suffix and, thus, save both space and time of generating numerous physical projected databases.

Another technique was given by Zaki [5] which has proved to be the most efficient one. Its key features are:

- It uses a vertical id-list database format, where each sequence is associated with a list of locations in which it occurs along with the time stamps. The id-lists for each sequence are kept in main memory.
- It uses a lattice-theoretic approach to decompose the original search space (lattice) into smaller lattices (equivalence classes) which can be processed independently in main memory.
- To generate a $(k+1)-$sequence, SPADE intersects two k-sequences with the same $(k-1)-$length prefix.
- All frequent sequences can be enumerated via simple id-list intersections.
- SPADE minimizes I/O costs by reducing database scans.

Conventional mining work for discovery of patterns in transactional databases, because the elements in transactional pattern are items that explicitly appear in pattern instances. On the other hand, location coordinates in a Spatial-temporal series are real numbers, which do not repeat themselves exactly in every pattern instance. To solve this problem Huiping Cao et al. [2] have proposed a new technique in which they have used segmentation algorithm(DP) to convert the location series to segment sequences. Then they apply a heuristic to find the candidate segments after which they followed two processes filtering and verification. The filtering process was used to prune the unnecessary segments on the basis of some threshold and the remaining segments were used for verification.

Koperski and Han [3] extended the concept given by Srikanta et al. to spatial databases. They defined some spatial predicates like close_to and near_by and then find association rules. To confine the number of discovered rules the concept of minimum support and minimum confidence are used. To minimize the number of costly spatial computations a novel two step technique for optimization during the search for associations was introduced. Computations starts at the high level of spatial predicates but more expensive, spatial computations are applied at lower concept levels only to those patterns that are large at higher level. To check the satisfiability at higher level the minimum bounding rectangles of the predicate pair should come in the range of minimum threshold.

2.2 Research Contributions

The main contributions of this thesis are:

- This thesis presents an idea of getting Spatial-Textual patterns from new type of activity-trajectory data which contains trajectories along with activities/events for each locations. We are able to tackle large data which may not fit in memory for getting frequent patterns but for our algorithm, it works.
- We propose a flexible framework which can use any type of Sequential pattern mining algorithm as core algorithm and runs these proposed algorithm on top of this to mine Spatial-Textual sequential patterns.
- We design three algorithms to mine patterns, namely Spatial-Textual, Textual-Spatial and Hybrid algorithm.
- If the data is large enough then an External-Memory algorithm to get patterns from this very large dataset is also proposed.
- We conduct experiments on the real world scenarios to verify the effectiveness of the framework and External-Memory algorithm.
- We propose an idea of choosing dissimilar Spatial-Textual sequences into the chunks to improve the performance of External-Memory algorithm.

Chapter 3

The Problem Definition and Proposed Solution Approach

3.1 Problem Definition

Let $I = (l_1, i_1), (l_2, i_2), \ldots, (l_m, i_n)$ be a set of all items along with their locations. An itemset is a non-empty subset of I. A sequence is an ordered list of itemsets. A sequence s is denoted by $\langle s_1, s_2, \ldots, s_l \rangle$, where s_j is an itemset. s_j is also called an element of the sequence, and denoted as $(x_1, x_2, \ldots, x_m)$, where x_k is an item-pair(with location). For brevity, the brackets are omitted if an element has only one item-location pair, i.e., element (x) is written as x. An item-location pair can occur at most once in an element of a sequence, but can occur multiple times in different elements of a sequence. The number of instances of item-location pairs in a sequence is called the length of the sequence. A sequence with length l is called an $l - sequence$. A sequence $\alpha = \langle a_1, a_2, \ldots, a_n \rangle$ is called a subsequence of another sequence $\beta = \langle b_1, b_2, \ldots b_m \rangle$ and β a supersequence of α, denoted as $\beta \sqsubseteq \alpha$, if there exist integers $1 \leq j_1 < j_2 < \ldots < j_n \leq m$ such that $a_1 \subseteq b_{j_1}, a_2 \subseteq b_{j_2}, \ldots, a_n \subseteq b_{j_n}$.

A sequence database S is a set of tuples $\langle sid, s \rangle$, where sid is a sequence_id and s a sequence. A tuple $\langle sid, s \rangle$ is said to contain a sequence α, if α is a subsequence of s. The support of a sequence α in a sequence database S is the number of tuples in the database containing α, i.e.,

$$support_s(\alpha) = |\langle sid, s \rangle | (\langle sid, s \rangle \in S) \wedge (\alpha \sqsubseteq s)| \tag{3.1}$$

It can be denoted as $support(\alpha)$ if the sequence database is clear from the context. Given a positive integer min_support as the support threshold, a sequence α is called a sequential pattern in sequence database S if $support_S(\alpha) \geq min\_support$ A sequential pattern with length l is called an $l - pattern$.

The problem here is to find frequent $Spatial - Textual$ patterns from given Spatial-Textual Data.

3.2 Proposed Solution

A framework to solve this type of problem is proposed. The framework shown in figure 3.1 contains 3 algorithms, named Spatial-Textual Mining (ST-Mining), Textual-Spatial Mining (TS-Mining), Hybrid Mining (H-Mining). This framework chooses one sequential Mining algorithm as the core algorithm for this framework and runs these 3 algorithms on top of this. This framework contains statistics extractor module which scans the Spatial-Textual sequences and calculates the stats for these sequences such as selectivity of the locations. This module passes this information to the next module which has 3 algorithms. This module takes support as the input and by seeing the selectivity of locations, it chooses one process plan (i.e. algorithm) for mining Spatial-Textual sequences. There is another module which contains basic sequence mining algorithms. We can choose any sequence mining algorithm as the core algorithm and can run selected process plan over this sequence mining algorithm. Finally this framework executes process plan with the selected core algorithm and given the Spatial-Textual patterns. To perform experiments, we have chosen PrefixSpan algorithm in which item-location pair is taken as a prefix instead of a single item.

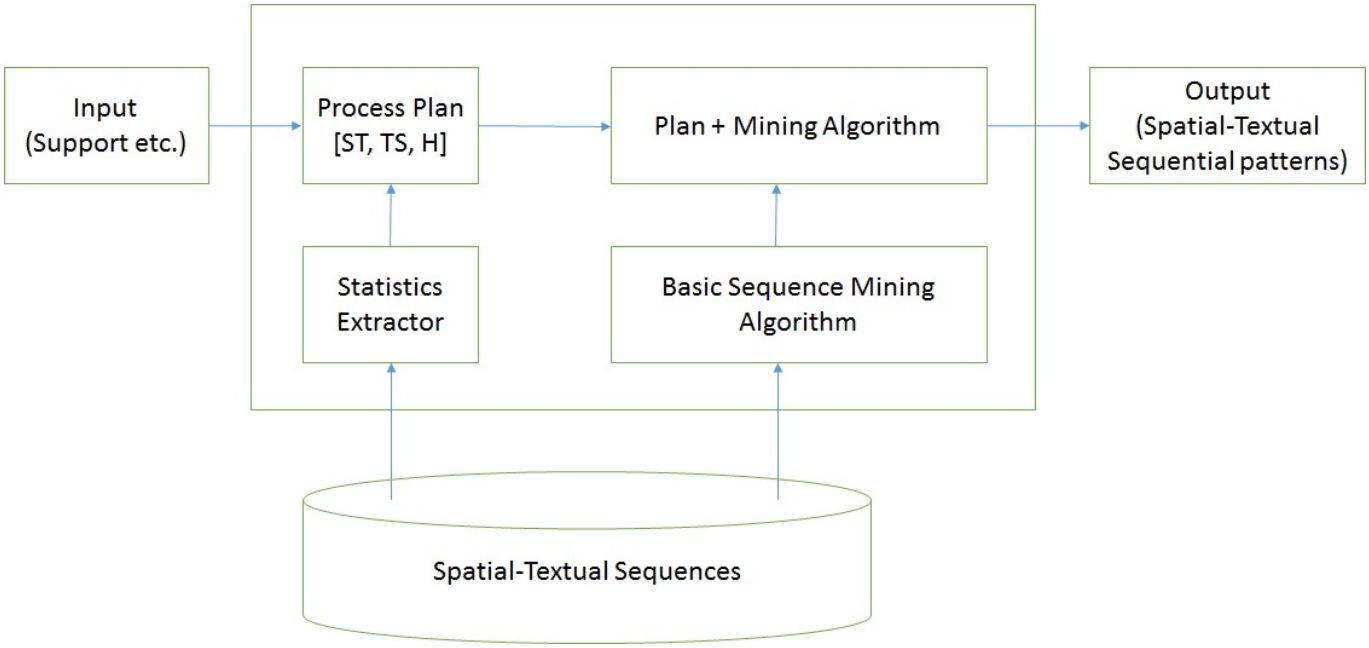

Figure 3.1: A Framework for Mining Spatial-Textual Sequences

Its a three step process:

(i) Cluster locations and generate cluster-IDs.

(ii) Apply Core Algorithm over Spatial/Textual sequences to get frequent Spatial/Textual sequences.

(iii) verify each frequent Spatial/Textual sequence to get Spatial-Textual/Textual-Spatial sequential patterns.

3.2.1 Location Clustering

The Spatial-Textual dataset contains trajectories of users with associated activities. Spatial series of locations is called trajectory. These spatial locations are actual locations means exact latitudes and longitudes. Since two trajectories can't be same in actual latitudes and longitudes. So there is need of assigning labels to the groups of locations means there is need of doing clustering over these locations so that labels can be assigned to them. Here DBScan [6] algorithm and Grid-based algorithm (Figure 3.2) is used for clustering the spatial locations and assigning labels to them. The DBScan [7] can identify clusters in large spatial dataset by looking at the local density of database elements, using only one input parameter. By using density distribution of nodes in the database, DBScan can categorize these nodes into separate clusters that defines the different classes. DBScan can find clusters of arbitrary shape, however clusters that lie close to each other tend to belong to the same class.

The key idea here is that for each point of a cluster, the neighborhood of a given radius (ε) has to contain at least a minimum number of points ($MinPts$), i.e. the density in the neighborhood has to exceed some threshold, where ε and $MinPts$ are input parameters. The shape of the neighborhood is determined by the choice of the distance for two points. DBScan uses a R^*-tree structure for more efficient queries. DBScan checks the $\varepsilon-$neighborhood of each point in the database. If the $\varepsilon-$neighborhood of a point has more than $MinPts$ points, a new cluster is created.

Grid-based algorithm initially make grids over the locations and if two locations are in the the same grid then the same id is assigned to those locations which is the grid-id. We use this grid-based approach for various grid sizes to study the location selectivity in the data.

3.2.2 Algorithms

This section contains Sequence mining algorithms that is designed to use in the framework shown in figure 3.1 as process plan. There are three algorithms for mining Spatial-Textual sequences, namely Hybrid algorithm, Spatial-Textual algorithm and Textual-Spatial algorithm. Hybrid algorithm is the basic sequence mining algorithm which takes the spatial and textual dimension simultaneously and generates the frequent Spatial-Textual sequential patterns. The other two algorithms are the variant of the Hybrid algorithm in which Spatial-Textual algorithm takes spatial dimension before the textual dimension and Textual-Spatial algorithm takes textual dimension before the spatial dimension.

Algorithm `ST-Mining()`

1 Scan the Spatial-Textual database and generate spatial sequences.
2 Call `CoreAlgo(`*Sequences* S, *Support* σ`)` over spatial sequences and get frequent spatial sequential patterns.
3 Prune Whole Spatial-Textual database from these frequent spatial sequential patterns.
4 Again call `CoreAlgo(`*Sequences* S, *Support* σ`)` and get frequent Spatial-Textual sequential patterns.
5 **return**

Procedure `CoreAlgo(`*Sequences* S, *Support* σ`)`

1 Scan Sequences S according to core algorithm and find each frequent sequential patterns.
2 **return**

Algorithm 1: Spatial-Textual Mining

Algorithm `TS-Mining()`

1 Scan the Spatial-Textual database and generate textual sequences.
2 Call `CoreAlgo(`*Sequences* S, *Support* σ`)` over textual sequences and get frequent textual sequential patterns.
3 Prune Whole Spatial-Textual database from these frequent textual sequential patterns.
4 Again call `CoreAlgo(`*Sequences* S, *Support* σ`)` and get frequent Spatial-Textual sequential patterns.
5 **return**

Procedure `CoreAlgo(`*Sequences* S, *Support* σ`)`

1 Scan Sequences S according to core algorithm and find each frequent sequential patterns.
2 **return**

Algorithm 2: Textual-Spatial Mining

Algorithm `H-Mining()`

1 Scan the Spatial-Textual database and map the pair of item and location to an integer.
2 Call `CoreAlgo(`*Sequences* S, *Support* σ`)` over these integer sequences and get frequent Spatial-Textual sequential patterns.
3 **return**

Procedure `CoreAlgo(`*Sequences* S, *Support* σ`)`

1 Scan Sequences S according to core algorithm and find each frequent sequential patterns.
2 **return**

Algorithm 3: Hybrid Mining

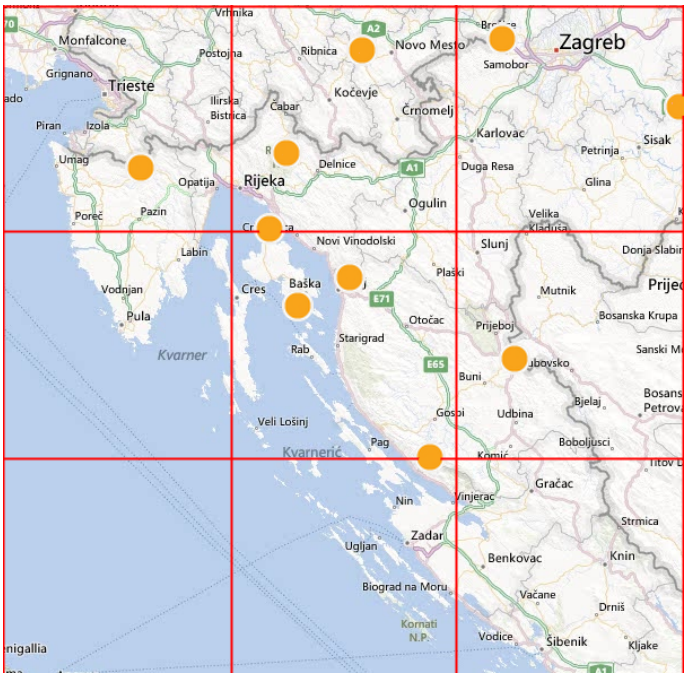

Figure 3.2: Grid-Based Clustering

3.2.3 Example

3.2.3.1 Hybrid Mining

This example shows how the *Hybrid* algorithm (*PrefixSpan* here) works on the given *Spatial* − *Textual* Data.

Given a set of sequences, where each sequence consists of a list of elements and each element consists of set of item-pairs.

10	$< l_1, a\ (l_2, a, b, c)(l_3, a, c)l_1, d\ (l_4, c, f) >$
20	$< (l_2, a, d)\ l_4, c(l_1, b, c)(l_3, a, e) >$
30	$< (l_1, e, f)(l_2, a, b)(l_4, d, f)l_3, c\ l_2, b >$
40	$< l_3, e\ l_4, g\ (l_2, a, f)(l_1, c)\ l_3, b(l_4, c) >$

Table 3.1: Example Dataset

Find all the frequent subsequences, i.e. the subsequences whose occurrence frequency in the set of sequences is no less than minimum support (Here $min\_sup = 2$ is considered)

- Find Length-1 sequential Patterns

Sequence	Count
l_2, a	4
l_2, b	2
l_3, a	2
l_3, c	2
l_4, c	3
l_3, e	2
l_1, c	2

Table 3.2: Length-1 Patterns

- Divide Search Space

Prefix	Projected database
l_2, a	$< \quad (\_l_2, b, c)(l_3, a, c) \quad l_1, d(l_4, c, f) \quad >$ $, < \quad (\_l_2, d) \quad l_4, c(l_1, b, c)(l_3, a, e) \quad >, <$ $(\_l_2, b)(l_4, d, f)\ l_3, c\ l_2, b >, < (\_l_2, f)\ l_1, c\ l_3, b\ l_4, c >$
l_4, c	$< (\_l_4, f) >, < (l_1, b, c)(l_3, a, e) >$
l_2, b	$< (l_2, c)(l_3, a, c)\ l_1, d(l_4, c, f) >, < (l_4, d, f)\ l_3, c\ l_2, b >$
l_3, a	$< (\_l_3, c)\ l_1, d(l_4, c, f) >, < (\_l_3, e) >$
l_3, c	$< l_1, d(l_4, c,) >, < l_2, b >$
l_4, f	$< l_3, c\ l_2, b >$
l_3, e	$< l_4, g(l_2, a, f)\ l_1, c\ l_3, b\ l_4, c >$
l_1, c	$< (l_3, a, e) >, < l_3, b\ l_4, c >$

Table 3.3: Projected Database

- Find subsets of sequential patterns

Prefix	Projected database
l_2, a	$< \quad (\_l_2, b, c)(l_3, a, c) \quad l_1, d(l_4, c, f) \quad >$ $, < \quad (\_l_2, d) \quad l_4, c(l_1, b, c)(l_3, a, e) \quad >, <$ $(\_l_2, b)(l_4, d, f)\ l_3, c\ l_2, b >, < (\_l_2, f)\ l_1, c\ l_3, b\ l_4, c >$

From projected database

$\_l_2, b$	$\_l_2, c$	l_3, a	l_3, c	l_1, d	l_4, c	l_4, f	$\_l_2, d$	l_1, b	l_1, c	l_3, e	l_4, d	l_2, b	$\_l_2, f$	l_3, b
2	1	2	2	1	2	2	1	1	2	1	1	1	1	1

From last table we are able to generate frequent sequences of prefix: l_2, a

$< (l_2, a, b) >< l_2, a\ l_3, a >< l_2, a\ l_3, c >< l_2, a\ l_4, c >< l_2, a\ l_4, f >< l_2, a\ l_1, c >$

Now take $< (l_2, a, b) >$

Prefix	Projected database
$< (l_2, a, b) >$	$< (\_l_2, c)(l_3, a, c)\ l_1, d(l_4, c, f) >, < (l_4, d, f)\ l_3, c\ l_2, b >$

From Projected database

$\_l_2, c$	l_3, a	l_3, c	l_1, d	l_4, c	l_4, f	l_4, d	l_2, b
1	1	2	1	1	2	1	1

Now take $< (l_2, a, b)\ l_3, c >$

Prefix	Projected database
$< (l_2, a, b)\ l_3, c >$	$< (\_l_3, a)\ l_1, d(l_4, c, f)\ l_2, b) >$

From projected database

$\_l_3, a$	l_1, d	l_4, c	l_4, f	l_2, b
1	1	1	1	1

Now take $< (l_2, a, b)\ l_4, f >$

Prefix	Projected database
$< (l_2, a, b)\ l_4, f >$	$< (\_l_4, c) >, < (\_l_4, d)\ l_3, d\ l_2, b) >$

From projected database

$\_l_4, c$	$\_l_4, d$	l_3, d	l_2, b
1	1	1	1

Final Frequent Sequences:

– Frequent-1

$< l_2, a >$	$< l_2, b >$	$< l_3, a >$	$< l_3, c >$	$< l_4, c >$
4	2	2	2	3

$< l_4, f >$	$< l_3, e >$	$< l_1, c >$
2	2	2

– Frequent-2

$< (l_2, a\ l_2, b) >$	$< l_2, a\ l_3, a >$	$< l_2, a\ l_3, c >$	$< l_2, a\ l_4, c >$	$< l_2, a\ l_4, f >$
2	2	2	2	2

$< l_2, a\ l_1, c >$	$< l_2, b\ l_3, c >$	$< l_2, b\ l_4, f >$
2	2	2

– Frequent-3

$< (l_2, a\ l_2, b)\ l_3, c >$	$< (l_2, a\ l_2, b)\ l_4, f >$
2	2

3.2.3.2 Spatial-Textual Mining

The Spatial-Textual data has two dimensions within the data. one is spatial dimension and the other is textual dimension. The one thing that we can do is just take the both dimensions simultaneously and run the existing sequence mining algorithm over this. But now the question arises, can we do better? The answer is yes. We propose two alternative in which we take one of the two dimension before the other.

Spatial sequences are smaller in size so it is better to work on spatial sequences first to speed up the mining process. In Spatial-Textual approach, spatial dimension is taken first and the get the frequent spatial patterns first. Now by the use of these frequent spatial patterns, prune the complete Spatial-Textual data as the locations which are not frequent in spatial dimension will not be frequent in Spatial-Textual patterns. Then we run sequence mining algorithm over this pruned Spatial-Textual sequences to find Spatial-Textual sequential patterns.

The Spatial-Textual data can be generated in a way that users visit locations but do activity very rarely. In this scenario, it may happen that the textual sequences may become smaller in size. So Textual-Spatial approach can be used for this scenario. The Textual-Spatial approach

is vice-versa of the Spatial-Textual approach in which we take textaul dimension first and prune the data on the basis of frequent textual sequential patterns.

Below is the working of the Spatial-Textual approach on the given $Spatial - Textual$ Data.

Given a set of sequences, where each sequence consists of an ordered list of transactions in which each transaction consists of set of pair of location and items.

10	$< l_1, a\ (l_2, a, b, c)(l_3, a, c) l_1, d\ (l_4, c, f) >$
20	$< (l_2, a, d)\ l_4, c(l_1, b, c)(l_7, a, e) >$
30	$< (l_1, e, f)(l_2, a, b)(l_4, d, f) l_3, c\ l_2, b >$
40	$< l_3, e\ l_6, g\ (l_2, a, f)(l_5, c)\ l_3, b(l_4, c) >$

Table 3.4: Example Dataset for ST-Mining Algorithm

Generate spatial sequences from the given Spatial-Textual Data.

10	$< l_1\ l_2\ l_3\ l_1\ l_4 >$
20	$< l_2\ l_4\ l_1\ l_7 >$
30	$< l_1\ l_2\ l_4\ l_3\ l_2 >$
40	$< l_3\ l_6\ l_2\ l_5\ l_3\ l_4 >$

Table 3.5: Spatial sequences for ST-Mining Algorithm

Find all the frequent spatial subsequences, i.e. the subsequences whose occurrence frequency in the set of spatial sequences is no less than minimum support (Here $min\_sup = 85\%$ is considered)

subsequence	support
$< l_2\ l_4 >$	4
$< l_4 >$	4
$< l_2 >$	4

Table 3.6: frequent Spatial subsequences for ST-Mining Algorithm

Prune whole Spatial-Textual Database and Find all the frequent spatial-Textual subsequences, i.e. the subsequences whose occurrence frequency in the set of pruned spatial-Textual sequences is no less than minimum support (Here $min\_sup = 85\%$ is considered)

10	$< (l_2, a, b, c)(l_4, c, f) >$
20	$< (l_2, a, d)\ l_4, c >$
30	$< (l_2, a, b)(l_4, d, f) l_2, b >$
40	$< (l_2, a, f)(l_5, c)(l_4, c) >$

Table 3.7: Pruned Spatial-Textual subsequences for ST-Mining Algorithm

subsequence	support
$< l_2 >$	4

Table 3.8: frequent Spatial-Textual subsequences for ST-Mining Algorithm

Chapter 4

Experimental Results

4.1 Datasets

The experimental results in this thesis have been collected using Foursquare dataset of trajectories. These trajectories contains spatial locations (latitude and longitude) with activities associated with them.

4.1.1 Dataset Preprocessing

The dataset is pure trajectory dataset in which each location contains the activities that user did at that location. Experiments are performed by applying labels to actual locations as well as to cluster of locations. For making clusters Grid-based approach as well as DBScan [7] algorithm is used. Since the hash of the location and the name of activities may be too long so the combination of these two as well as individual is mapped to integers. For mapping the following formulation is used :

Let $L = l_1, l_2, \ldots, l_m$ be a set of all locations and $I = i_1, i_2, \ldots, i_n$ be a set of all items/activities. Here we have m locations and n activities so we have reserved $m \times n$ integers for the combination of location with activity. where combination of i^{th} location to j^{th} activity will be assigned integer according to the following formula :

$$integer = ((n-1) \times i) + j \tag{4.1}$$

Total Number of Sequences	30755
Number of Transaction/Sequence	8
Number of items/Transaction	6
Total number of items(events)	50000

Table 4.1: Statistics of Spatial-Textual Sequences

For example If we have 50 locations and 100 activities if the given pair of location with activity is (l_3, i_5) then the integer value that will be assigned to this pair will be $((3-1)\times 100)+5 = 205$.

4.2 Comparison of Total Execution Time w.r.t. specific support

Figure 4.1 contains 4 graphs which shows the comparison of Total time taken by both the algorithms (i.e. Hybrid and Spatial-Textual Algorithm) on the basis of specific support. Here experiments are performed for various numbers of sequences and for various supports. Figure 4.1 (a), (b) and (c) shows that Spatial-Textual Algorithm takes less time than Hybrid algorithm for some support because for that support, the Spatial patterns are less in numbers and the dataset pruned so much that enables to run Spatial-Textual algorithm faster than Hybrid algorithm. Figure 4.1 (d) shows that as the support reduces, the Hybrid algorithm starts performing well than Spatial-Textual algorithm in terms of total execution time.

In all these graphs Spatial-Textual mining algorithm is able to give result for sequences greater than 20000 sequences but Hybrid algorithm is giving memory error. It clearly shows that Spatial-Textual mining algorithm prunes data so that it is able to find frequent patterns which may not be found by Hybrid algorithm due to memory constraint.

4.3 Comparison of Total Execution Time w.r.t. size of sequence database

Figure 4.2 contains 4 graphs which shows the comparison of Total time taken by both the algorithms (i.e. Hybrid and Spatial-Textual Algorithm) on the basis of size of sequences. Here experiments are performed for various numbers of sequences and for various supports. Figure 4.2 (a), (b), (c) and (d) shows that Spatial-Textual Algorithm takes less time than Hybrid algorithm for some support because for that support, the Spatial patterns are less in numbers and the dataset pruned so much that enables to run Spatial-Textual algorithm faster than Hybrid algorithm but as we reduces the supports, the difference between execution time decreases and for less support, Hybrid algorithm starts performing well. This is because as the support reduces, the spatial patterns are found more in numbers which does not prune the dataset more and Spatial-Textual algorithm starts taking more time than Hybrid algorithm.

4.4 Comparison of Core Algorithm execution Time w.r.t. specific support

Figure 4.3 contains 4 graphs which shows the comparison of PrefixSpan time taken by both the algorithms (i.e. Hybrid and Spatial-Textual Algorithm) on the basis of specific support. Here

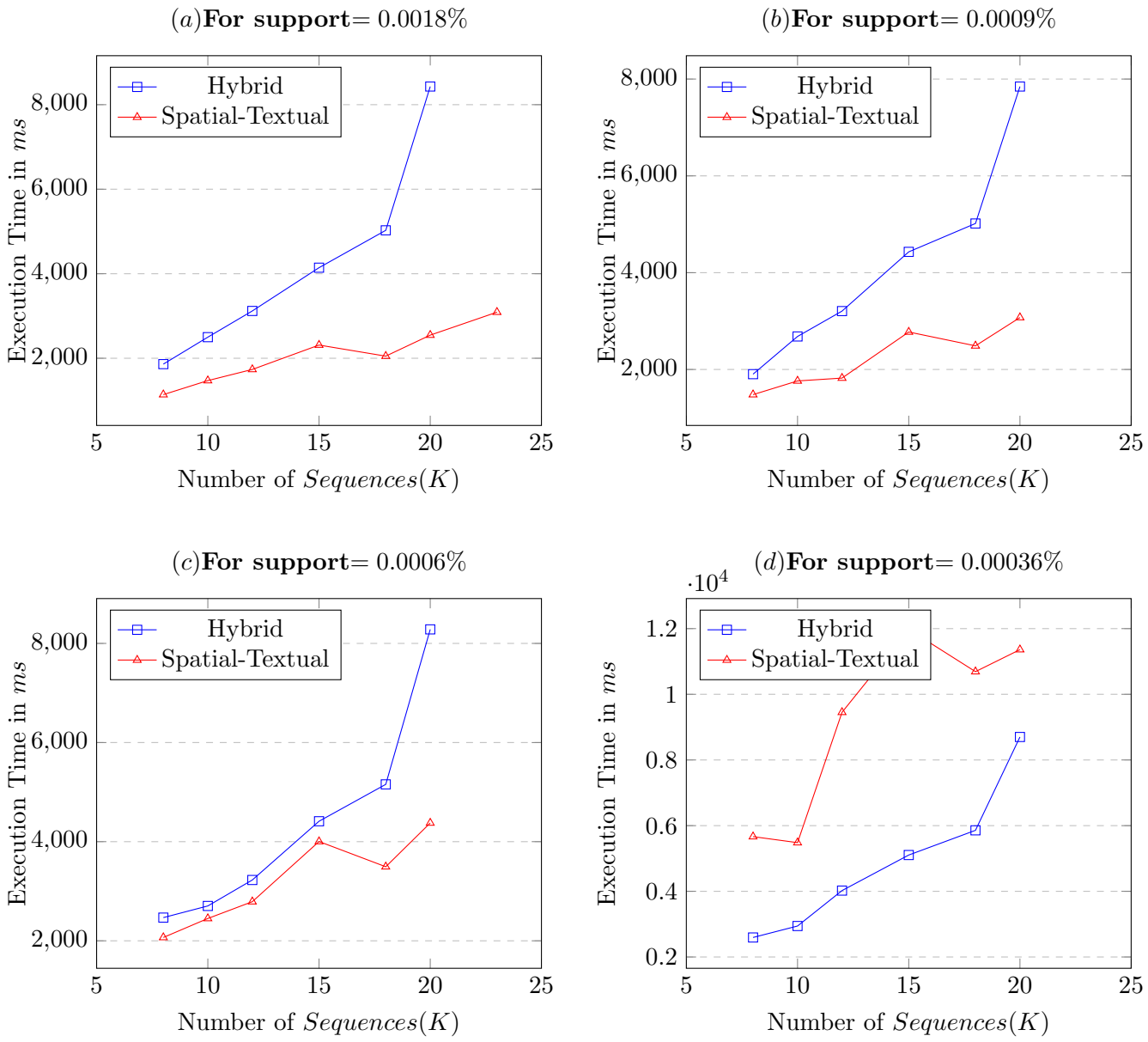

Figure 4.1: Total Time on basis of support

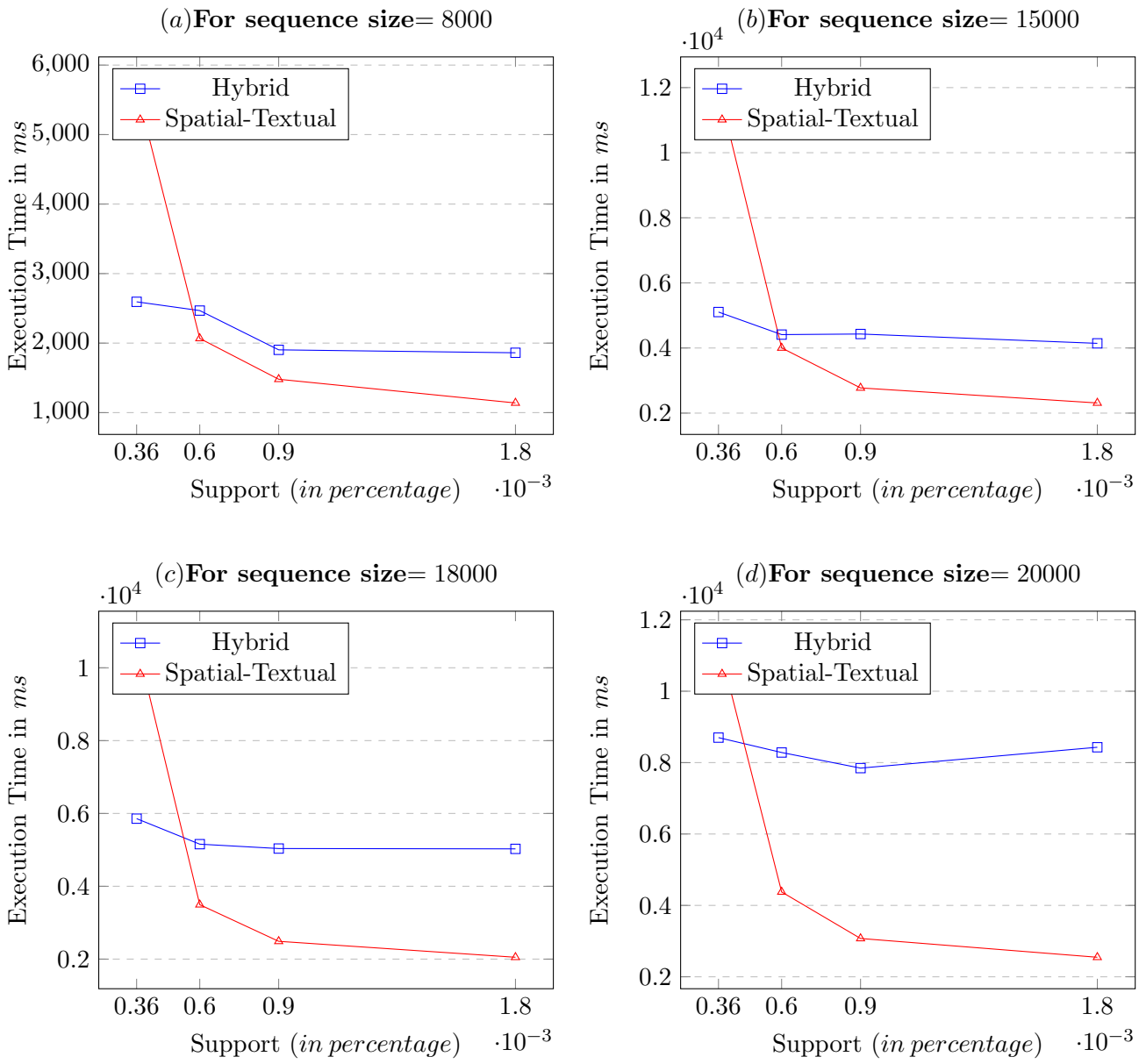

Figure 4.2: Total Time on basis of sequence size

we performed experiments for various numbers of sequences and for various supports. Figure 4.3 (a), (b), (c) and (d) shows that Spatial-Textual Algorithm takes less time or equal time than Hybrid algorithm because after pruning, the dataset may be cut down or not. If the dataset is cut down on the basis of spatial patterns than the Spatial-Textual algorithm will take less time than Hybrid algorithm else if the dataset is not cut down than it will be the exact same dataset and it will take same time as that of Hybrid algorithm. So this can be concluded by seeing the Core Algorithm execution time that Spatial-Textual algorithm prune the data.

4.5 Comparison of Core Algorithm execution Time w.r.t. size of sequence database

Figure 4.4 contains 4 graphs which shows the comparison of PrefixSpan time taken by both the algorithms (i.e. Hybrid and Spatial-Textual Algorithm) on the basis of size of sequences. Here experiments are performed for various numbers of sequences and for various supports. Figure 4.4 (a), (b), (c) and (d) shows that Spatial-Textual Algorithm takes less time or equal time than Hybrid algorithm because after pruning, the dataset may be cut down or not. If the dataset is cut down on the basis of spatial patterns than the Spatial-Textual algorithm will take less time than Hybrid algorithm else if the dataset is not cut down than it will be the exact same dataset and it will take same time as that of Hybrid algorithm. So this can be concluded by seeing the Core Algorithm execution time that Spatial-Textual algorithm prune the data.

4.6 Comparison of Total Execution Time w.r.t. location granularity

Figure 4.5 contains 4 graphs which shows the comparison of Total time taken by both the algorithms (i.e. Hybrid and Spatial-Textual Algorithm) on the basis of location granularity. Here experiments are performed for 15000 sequences and for various supports. Figure 4.5 (a), (b), (c) and (d) shows that Spatial-Textual Algorithm takes less time than Hybrid algorithm if the grid size is less and as the grid size starts increasing, the Hybrid algorithm starts perform well because for small grid size, the selectivity of location is less so spatial patterns are formed less in numbers that enables to run Spatial-Textual algorithm faster than Hybrid algorithm but as we increases grid size, the difference between execution time decreases and for more grid size, Hybrid algorithm starts performing well. This is because as the grid size increases, the spatial patterns are found more in numbers which does not prune the dataset more and Spatial-Textual algorithm starts taking more time than Hybrid algorithm.

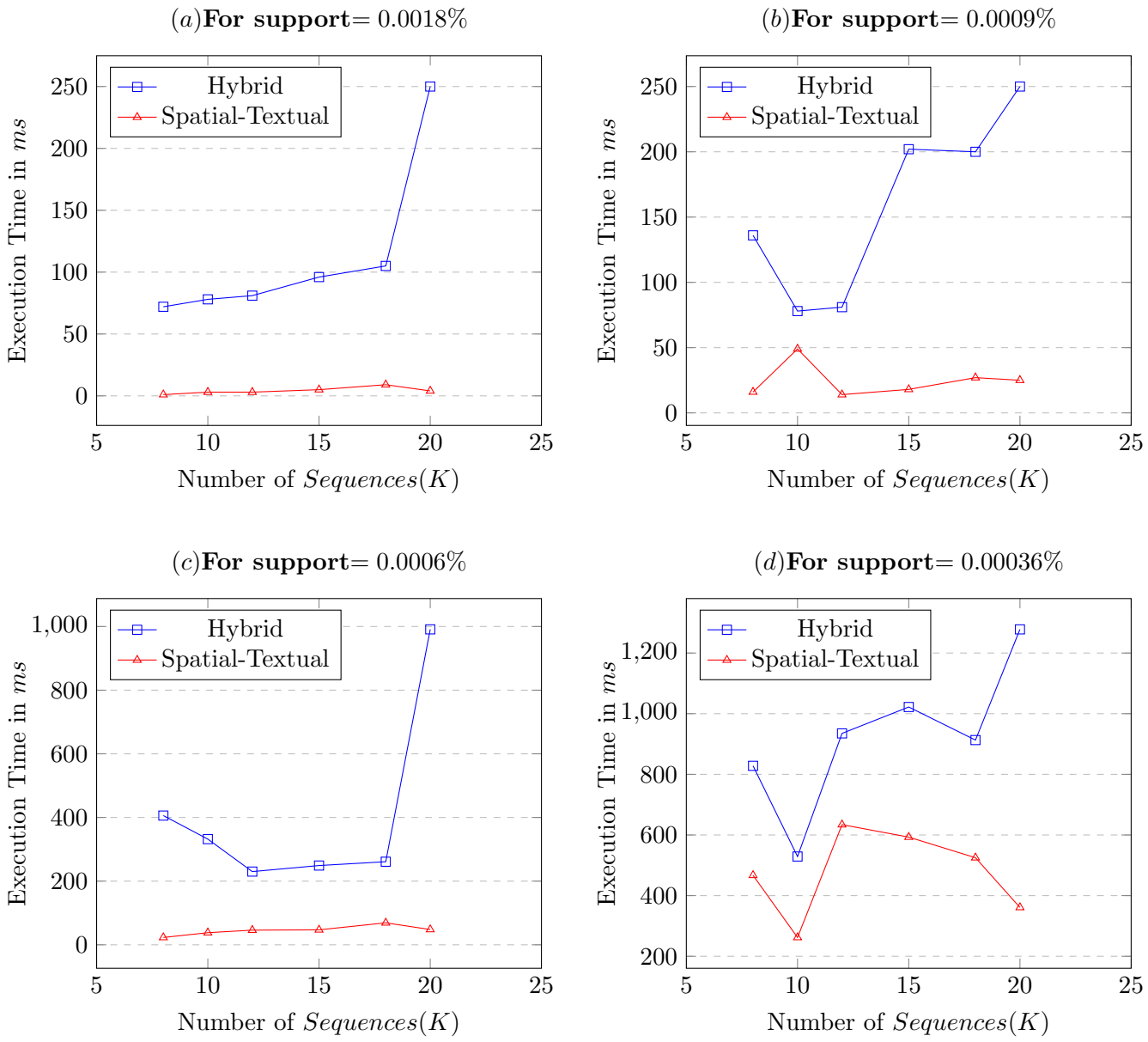

Figure 4.3: PrefixSpan Time on basis of support

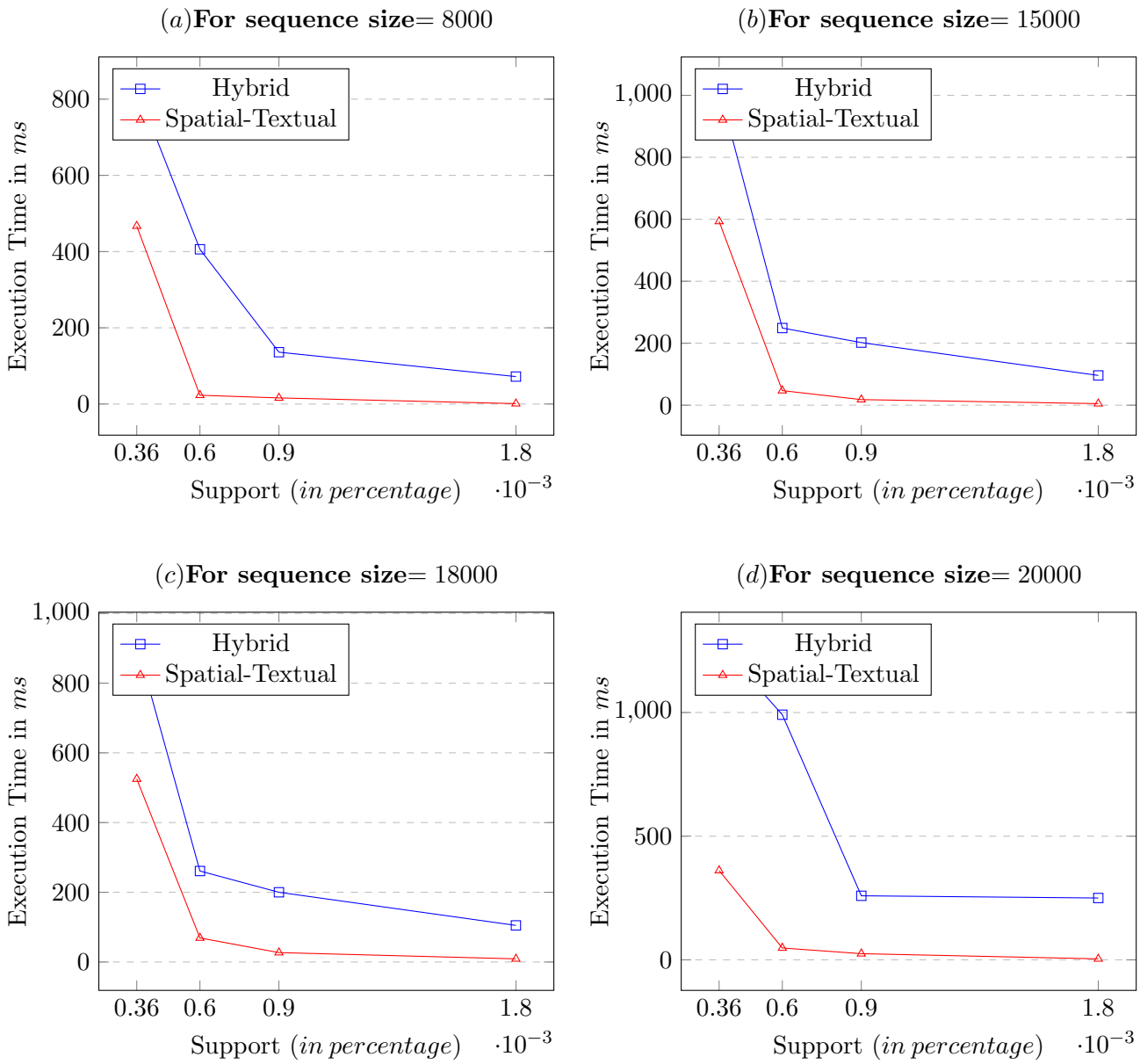

Figure 4.4: Total Time on basis of sequence size

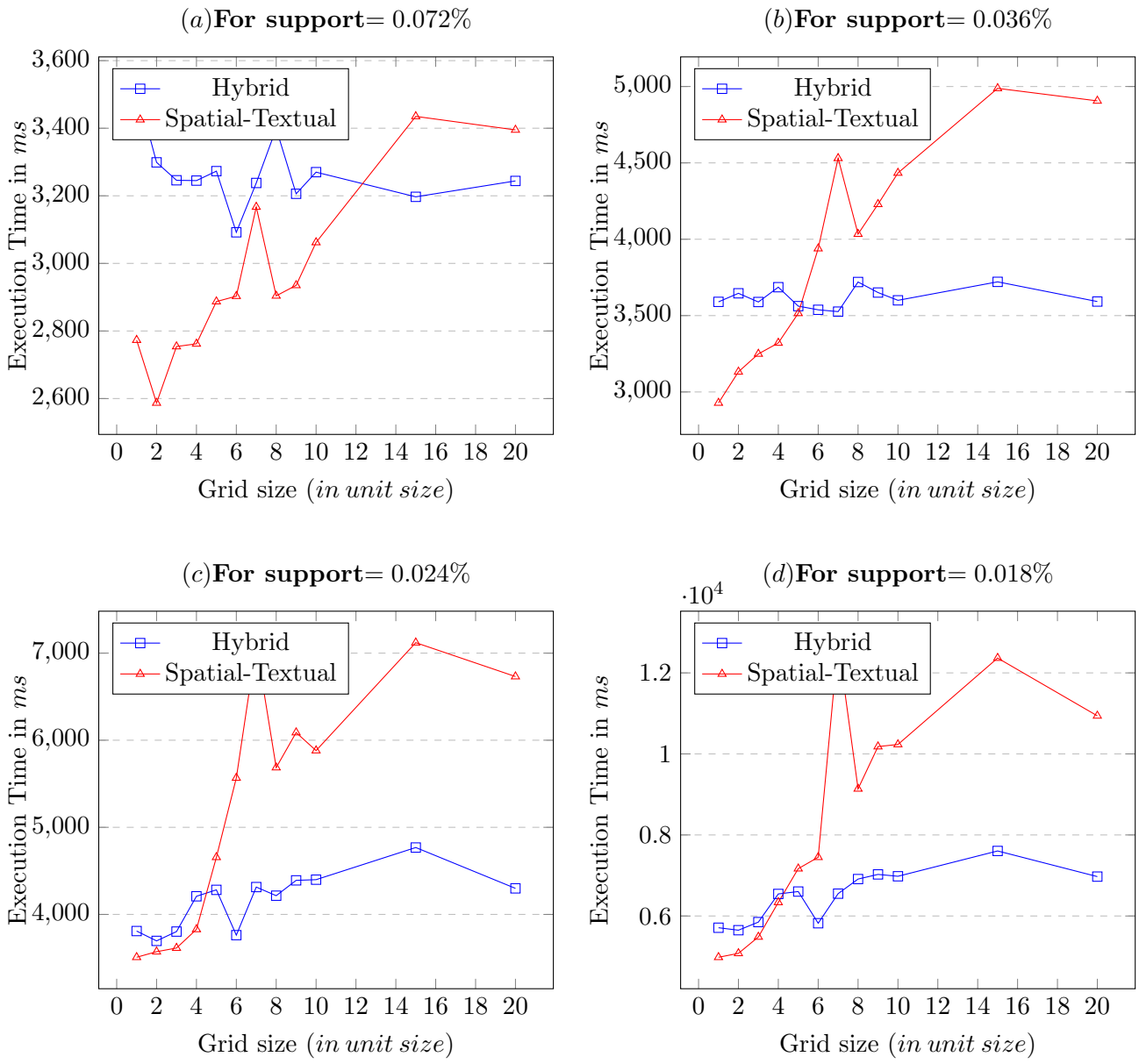

Figure 4.5: Total Time on basis of location granularity

4.7 Comparison of Core Algorithm execution Time w.r.t. location granularity

Figure 4.6 contains 4 graphs which shows the comparison of PrefixSpan time taken by both the algorithms (i.e. Hybrid and Spatial-Textual Algorithm) on the basis of location granularity. Here we performed experiments for 15000 sequences and for various supports. Figure 4.6 (a), (b), (c) and (d) shows that Spatial-Textual Algorithm takes less time or equal time than Hybrid algorithm because after pruning, the dataset may be cut down or not. If the dataset is cut down on the basis of spatial patterns than the Spatial-Textual algorithm will take less time than Hybrid algorithm else if the dataset is not cut down than it will be the exact same dataset and it will take same time as that of Hybrid algorithm. So this can be concluded that in terms of PrefixSpan time, Spatial-Textual algorithm will take either less time or equal time as that of Hybrid algorithm for all grid sizes as shown in the graphs of figure 4.6.

4.8 Analysis

We design three algorithms in which the performance of the Hybrid algorithm is same as of basic core sequence mining algorithm that is used by the framework. The Spatial-Textual algorithm works well for better location selectivity. We analyze the performance of the Spatial-Textual algorithm as follows:

Let there are n number of Spatial-Textual sequences. l be the different locations present in the dataset and i be the different items or activities present in the dataset. The time complexity of the Hybrid algorithm for particular support would depend on the n, l and i. The Spatial-Textual algorithm in the first phase runs only on the spatial sequences so the first phase of the ST-Mining will depend on the n and l. In the second phase, pruning is done on the basis of the frequent spatial patterns. If the location selectivity is better then this phase prunes the data much by taking into consideration that the locations which are not frequent in the spatial patterns will not be frequent in the final Spatial-Textual patterns. Pruning is done in horizontally as well as vertically. Let we prune the number of sequences by the factor of k_1 i.e. we prune vertically by the factor of k_1. In horizontal, we consider two dimensions i.e. spatial dimension as well as textual dimension. If we prune spatial dimension by the factor of k_2 and textual dimension by the factor of k_3 then the time complexity of this phase would depend on the n/k_1, l/k_2 and i/k_3. If the data has better location selectivity, then the factor k_2 will be much good and play a major role in the pruning. So if the factors k_1, k_2 and k_3 are good enough means we prune the data much then this third phase will take very less time to give frequent Spatial-Textual patterns. Since the first phase runs on the spatial sequences which are less in horizontal size and take less time to generate frequent spatial sequences for better location selectivity. So overall, for better location selectivity in the data and better pruning in the data, the Spatial-Textual algorithm outperforms other approaches.

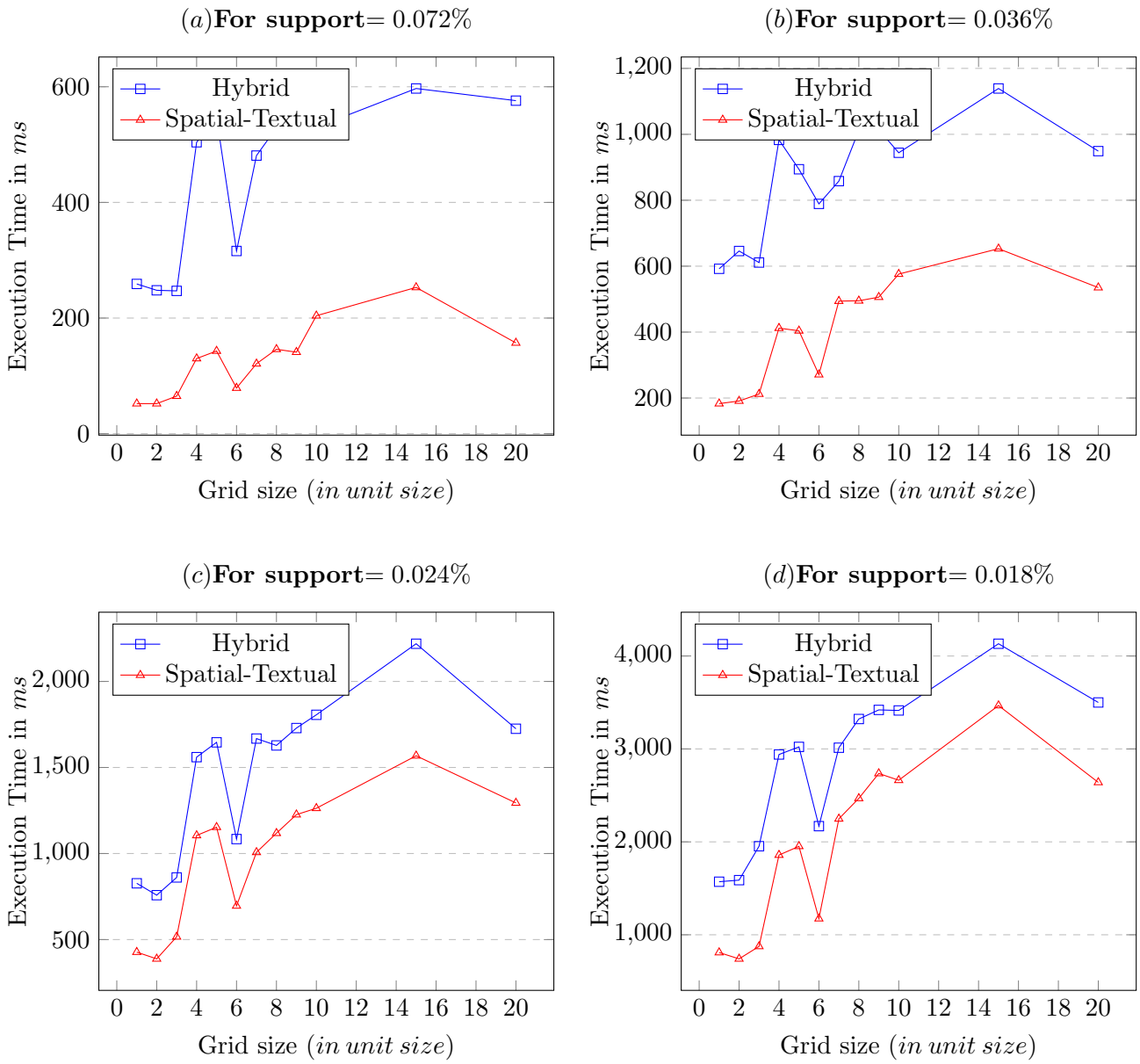

Figure 4.6: PrefixSPan Time on basis of location granularity

Figure 4.7 is the experimental results of the pruning. Figure 4.7 contains 3 graphs which shows the horizontal pruning, vertical pruning and transaction pruning by the Spatial-Textual Algorithm on the basis of location granularity. Here we perform experiments for 20000 sequences and for support=0.024%. These 20000 sequences contains 1056242 pair of location and items and 170624 transactions or locations. Figure 4.7 (a), (b) and (c) shows that Spatial-Textual Algorithm prunes the data much for better location selectivity and as the selectivity becomes poorer then it dost not prune the data much in either direction whether horizontally or vertically. Figure 4.7 (a) shows the vertical pruning in the data. Figure 4.7 (b) shows the horizontal pruning in the data i.e. pruning of the items from the data. Figure 4.7 (c) shows the pruning of the transactions i.e. how much locations are pruned in the data after taking spatial dimension first.

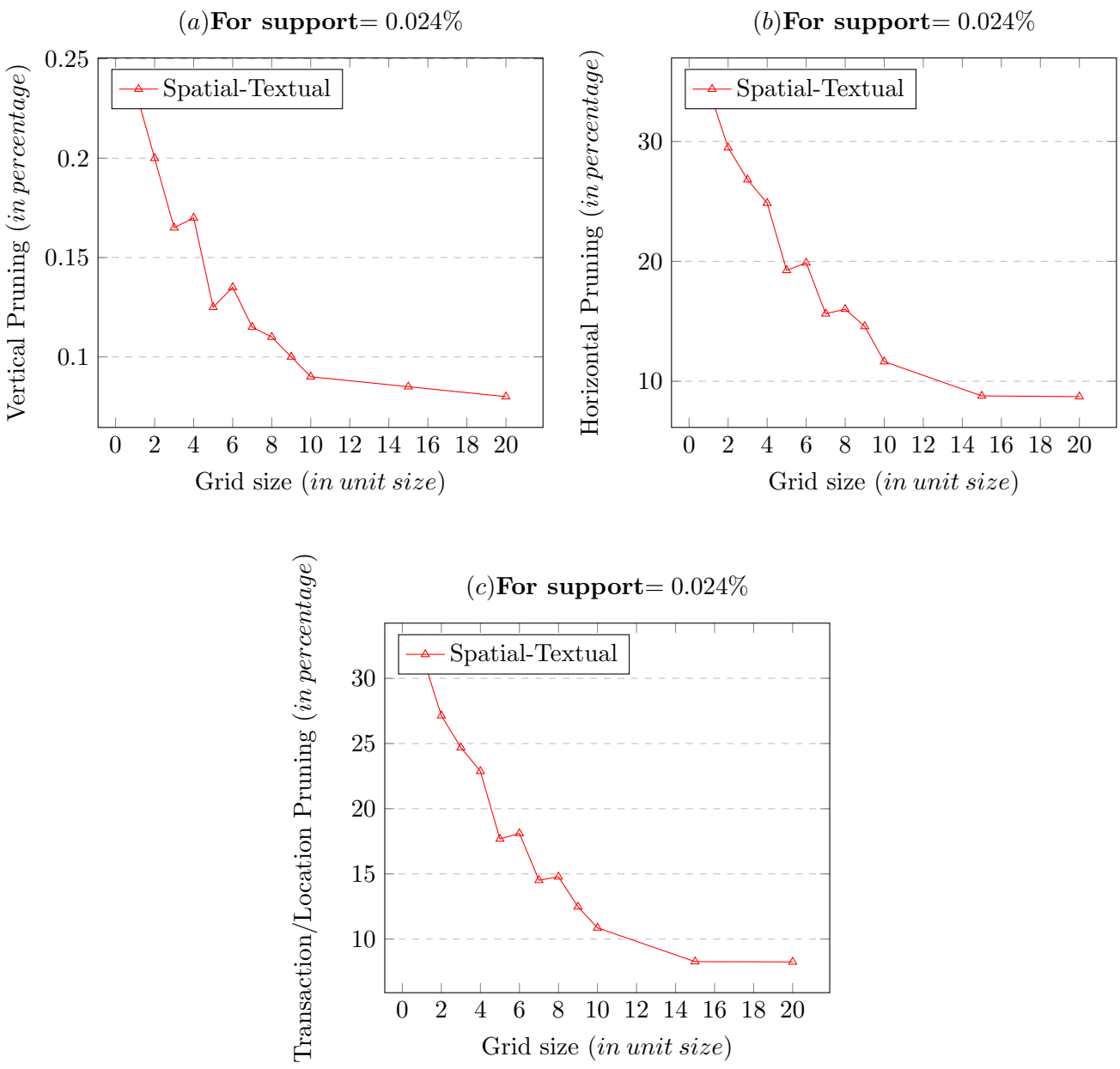

Figure 4.7: Pruning done by Spatial-Textual Algorithm

Chapter 5

External Memory Algorithm

Memory size nowadays increasing very rapidly so there is no difficulty in storing dataset into memory. Still, there might be some dataset that is too large for the main memory to accommodate in a batch. PrefixSpan [4] algorithm generate projected database also which increases recursively and may not fit into main memory.

Spatial-Textual mining algorithm initially runs on spatial sequences which are small in size than full Spatial-Textual data and may fit into main memory but there may be the case where spatial sequences also not fit into main memory. In this case, an External-Memory algorithm is proposed in which sequential patterns are discovered using a partition-and-validation technique. Spatial sequences is generated from the extra-large Spatial-Textual database. This Spatial sequences is partitioned so that each partition can be handled in main memory. The number of partition is minimized by reading as many data sequences into main memory as possible so that the number of false positives is minimized. The frequent spatial patterns are obtained from each of these partition and the true spatial patterns are identified with only one extra database pass through support against whole Spatial sequences. Now the Spatial-Textual database is pruned by the use of these frequent spatial sequences and again the same procedure is followed for finding frequent Spatial-Textual sequences. and again these frequent Spatial-Textual sequences are validated by another one extra pass over the pruned Spatial-Textual database.

The External-Memory algorithm works in the following steps:

(i) Scan the complete sequential database and generate partitions.

(ii) For each partition, generate frequent sequential patterns.

(iii) Validate frequent sequential patterns over complete sequential database.

Below is the algorithm for external memory implementation of ST-Mining algorithm. Here *min_support* is in the percentage.

Algorithm `External-Memory-ST-Mining()`

1 Scan the Spatial-Textual database and generate Spatial sequences.
2 Create partitions of spatial sequences.
3 **for** *each spatial sequence partition* **do**
4 Call `CoreAlgo(`*Sequences* S, *Support* σ`)` over spatial sequences and get frequent spatial sequential patterns.
end
5 Validate frequent spatial sequential patterns over full spatial sequences.
6 Prune Whole Spatial-Textual database from these frequent spatial sequential patterns.
7 Scan pruned Spatial-Textual sequences and create partitions of pruned Spatial-Textual sequences.
8 **for** *each Spatial-Textual sequence partition* **do**
9 Again call `CoreAlgo(`*Sequences* S, *Support* σ`)` and get frequent Spatial-Textual sequential patterns.
end
10 Validate frequent Spatial-Textual sequential patterns over full spatial sequences.
11 **return**

Procedure `CoreAlgo(`*Sequences* S, *Support* σ`)`

1 Scan Sequences S according to core algorithm and find each frequent sequential patterns.
2 **return**

Algorithm 4: External-Memory ST-Mining

5.1 Increasing the performance of External Memory Algorithm

This External-Memory algorithm is used to generate patterns over very large data which doesn't fit into memory. Since we are having Spatial-Textual data means we are having locations associated with our data so this can be useful to increase the performance of the External-Memory algorithm. In External-Memory algorithm, we consider the chunks of data which fits into memory and runs the Basic Sequence mining algorithm over these chunks and finally verifies the patterns. These chunks can have any number and type of sequences. Since here we are having Spatial-Textual sequences so we can have dissimilar type of sequences into each chunk. Since Spatial-Textual sequence is nothing but a trajectory, so dissimilarity in trajectory is defined in a way that two trajectories are said to be dissimilar if they are differing in their maximum location points of the trajectory and the location points are not close to each other. One thing that need to be kept in mind that the dissimilarity between trajectories are a relative measure.

We find the dissimilar sequences using following steps:

(i) For each sequence, find cluster-ID for each location of the trajectory.

(ii) Assign an ID to a trajectory, which is the grid-ID of that grid which contains maximum number of locations for that trajectory.

(iii) Choose chunk of sequences which contains maximum number of dissimilar trajectory-ID sequences.

Since we are using Grid-Based approach for labeling the locations. Parallel to this, we are assigning the label to each trajectory which is that grid number in which maximum number of locations of that trajectory are lying. At the time of picking the sequences into the chunk, we pick sequences (here trajectories) of different label IDs. This will create the chunk which is having more dissimilar sequences. The usefulness of this sort of chunk is that it will reduce the number of false positive patterns from every chunk and that reduced false positives need not to be verified at the end.

We have experimentally shown that chunks which are having dissimilar sequences, improves the performance of the External-Memory algorithm.

Figure 5.1 contains 3 graphs which shows the comparison of the time taken by both the approaches of External-Memory algorithm (i.e. Simple Serial chunk distribution and dissimilar sequences chunk distribution) on the basis of location granularity. Here we performed experiments for approx. 30000 sequences and for supports=0.072%. Figure 5.1 (a), (b) and (c) shows that dissimilar sequences chunk distribution approach takes less time than Simple Serial chunk distribution approach because dissimilar sequences chunk distribution approach will reduce the generation of false positive patterns and because of it that false positive patterns need not to be verified. So due to this, the performance of dissimilar sequences chunk distribution approach in External-Memory algorithm is increases as shown in the graphs of figure 5.1.

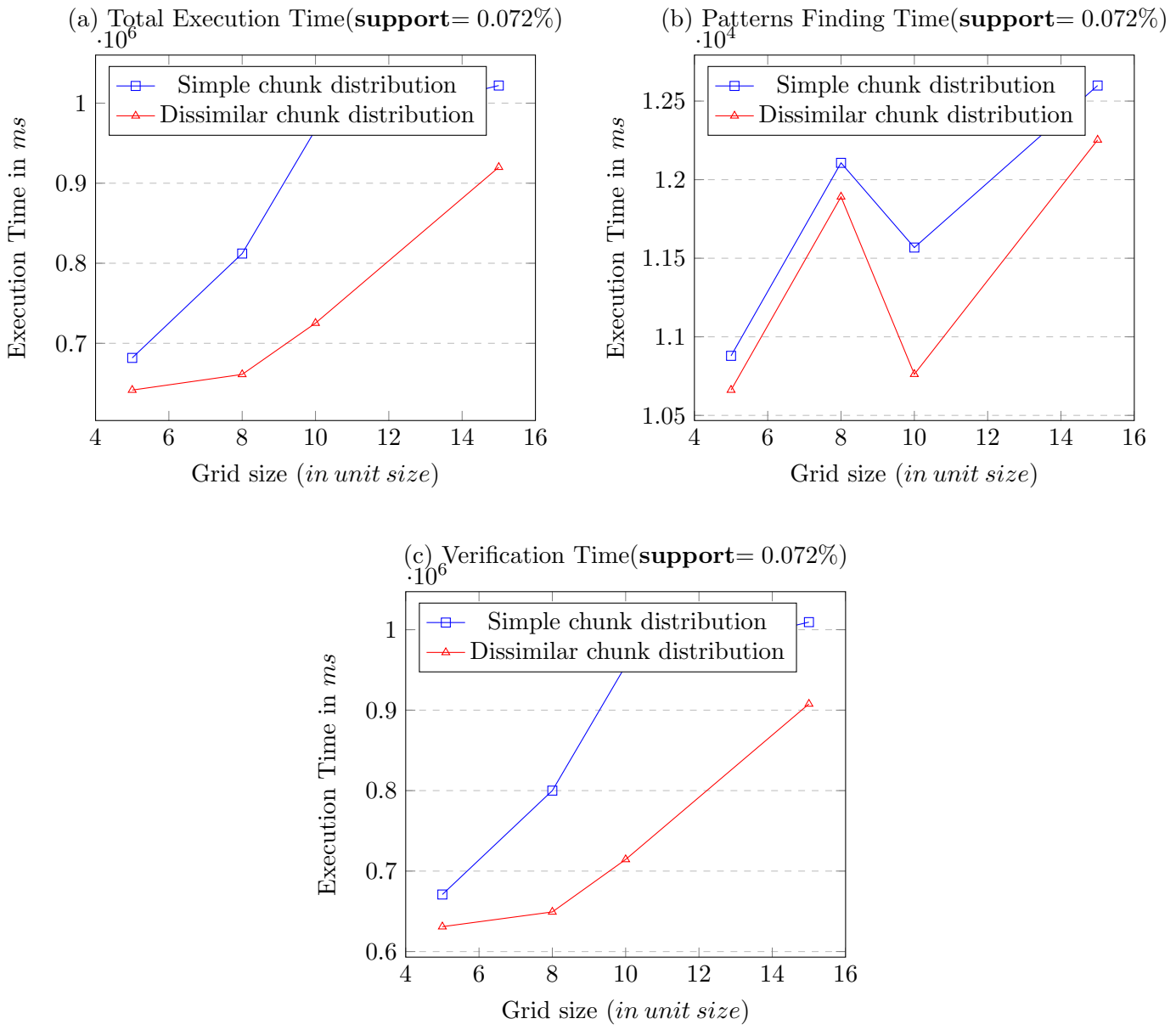

Figure 5.1: External-Memory Algorithm Execution Time on basis of location granularity

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Sequential pattern mining is very active research topic nowadays and extracting patterns from Spatial-Textual data is very interesting. This thesis concludes that locations usually have low selectivity so mining spatial data first and extracting frequent spatial patterns first helps in getting overall frequent spatial-Textual patterns very quickly. The most significant contribution of this thesis is the introduction and formalization of the spatial data associated with activities/events. Here a framework is proposed which can be used with any core sequential pattern mining algorithm along with Spatial-Textual Mining and Textual-Spatial Mining algorithm and this framework intelligently chooses the algorithm which it has to run for the given Spatial-Textual data by scanning it in first pass. Since Spatial-Textual data is activity-trajectory data so dissimilarity can be found very easily among the trajectories and these dissimilar trajectories improves the performance of External-Memory algorithm.

6.2 Future Directions

The framework presented in this thesis is used for mining Spatial-Textual data but this data is static means the trajectories are the past history of the users which are non-growing. The future work will be to find sequential patterns from self-growing Spatial-Textual sequences (incremental sequences), means the data is growing horizontally as well as vertically.

Bibliography

[1] Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering*, ICDE '95, pages 3–14, Washington, DC, USA, 1995. IEEE Computer Society.

[2] Huiping Cao, Nikos Mamoulis, and David W. Cheung. Mining frequent spatio-temporal sequential patterns. In *In ICDM*, pages 82–89, 2005.

[3] Krzysztof Koperski and Jiawei Han. Discovery of spatial association rules in geographic information databases. In MaxJ. Egenhofer and JohnR. Herring, editors, *Advances in Spatial Databases*, volume 951 of *Lecture Notes in Computer Science*, pages 47–66. Springer Berlin Heidelberg, 1995.

[4] Jian Pei, Jiawei Han, Behzad Mortazavi-asl, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Mei chun Hsu. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. pages 215–224, 2001.

[5] MohammedJ. Zaki. Spade: An efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1-2):31–60, 2001.

[6] Birant, Derya, and Alp Kut. ST-DBSCAN: An algorithm for clustering spatialtemporal data. Data & Knowledge Engineering 60.1 (2007): 208-221.

[7] Ester, Martin, et al. "A density-based algorithm for discovering clusters in large spatial databases with noise." KDD. Vol. 96. 1996.

[8] Fournier-Viger, P., Gomariz, A., Soltani, A., Lam, H., Gueniche, T. (2014) "SPMF: Open-Source Data Mining Platform." http://www.philippe-fournier-viger.com/spmf/

[9] Ashok Savasere, Edward Omiecinski, Shamkant B. Navathe "An Efficient Algorithm for Mining Association Rules in Large Databases." VLDB 1995: 432-444