COLOCATION MINING IN UNCERTAIN DATA SETS: A PROBABILISTIC APPROACH

M.Sheshikala[1], D. Rajeswara Rao[2], and Md. Ali Kadampur[3]

[1,3]S.R Engineering College
[2]Kl University

ABSTRACT

In this paper we investigate colocation mining problem in the context of uncertain data. Uncertain data is a partially complete data. Many of the real world data is Uncertain, for example, Demographic data, Sensor networks data, GIS data etc.,. Handling such data is a challenge for knowledge discovery particularly in colocation mining. One straightforward method is to find the Probabilistic Prevalent colocations (PPCs). This method tries to find all colocations that are to be generated from a random world. For this we first apply an approximation error to find all the PPCs which reduce the computations. Next find all the possible worlds and split them into two different worlds and compute the prevalence probability. These worlds are used to compare with a minimum probability threshold to decide whether it is Probabilistic Prevalent colocation (PPCs) or not. The experimental results on the selected data set show the significant improvement in computational time in comparison to some of the existing methods used in colocation mining.

KEYWORDS

Probabilistic Approach, Colocation Mining, Un-certain Data Sets

1. INTRODUCTION

Basically colocation mining is the sub-domain of data mining. The research in colocation mining has advanced in the recent past addressing the issues with applications, utility and methods of knowledge discovery. Many techniques inspired by data base methods (Join based, Join-less, Space Partitioning, etc.,) have been attempted to find the prevalent colocation patterns in spatial data. Fusion and fuzzy based methods have been in use. However due to growing size of the data and computational time requirements highly scalable and computationally time efficient framework for colocation mining is still desired. This paper presents a computational time efficient algorithm based on Probabilistic approach in the uncertain data.

Consider a spatial data set collected from a geographic space which consists of features like birds (of different types), rocks, different kinds of trees, houses, which is shown in Fig: 4. From this the frequent patterns on a spatial dimension can be identified, for example, < *bird, house* > and < *tree, rocks*>, the patterns are said to be colocated and they help infer a specific eco-system. This paper presents a computationally efficient method to identify such prevalent patterns from spatial data sets.

Since the object data is scattered in space (spatial coordinates) extracting information from it is quite difficult due to complexity of spatial features, spatial data types, and spatial relationships. For example, a cable service provider may be interested in services frequently requested by geographical neighbours, and thus gain sales promotion data. The subscriber of the channel is

DOI: 10.5121/ijci.2016.5101

located on a wide geographical positions and has wide ranging interest/preferences. Further in the process of collecting data there may be some missing links giving rise to uncertainty in the data. From the data mining point of view all this adds to complexity of analysis and needs to be handled properly. The paper addresses the uncertainty and data complexity issues in finding prevalent colocations.

The paper includes 1.The methods for finding the exact Probabilistic Prevalent colocations (PPCs). 2. Developing a dynamic programming algorithm to find Probabilistic Prevalent colocations (PPCs) which dramatically reduces the computation time. 3. Results of application of the proposed method on different data sets.

The remaining paper is organized as follows: In Section-1, we discuss the introduction, and related work is discussed in Section-2. In section-3 we discuss the definitions, and a block diagram to show the complete flow to find PPCs are discussed in section-4, In section-5 we discuss dynamic- programming algorithm for finding all Probabilistic Prevalent Colocations. We show the experiment results in Section-6. Finally, in section-7 we suggest future work.

2. RELATED WORK

Many methods have been extensively explored in order to find the Prevalent colocations in spatially Precise data. Some of these methods are:

2.1 Space Partitioning Method:

This approach finds the neigh-boring objects of a subset of features. It finds the partition centre points with base objects and decomposes the space from partitioning points using a geometric approach and then finds a feature within a distance threshold from the partitioning point in each area. This approach may generate incorrect colocation patterns, because it may miss some of the colocation instances across partition areas which can be identified from the below Fig:1.

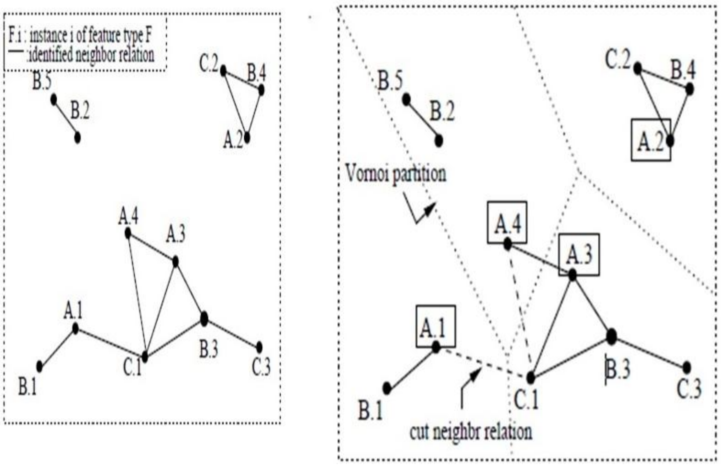


Fig. 1. Space Partitioning Approach

2.2. Join-Based Approach

This approach finds the correct and complete colocation instances, first it finds all neighboring pair objects (of size 2) using a geometric method, the method finds the instance of size k(> 2) colocations by joining the instances of its size k-1 subset colocation where the first k-2 objects are common. This approach is computationally expensive with the increase of colocation patterns and their instances as in Fig:2.

This approach finds the correct and complete colocation instances, first it finds all neighbouring pair objects (of size 2) using a geometric method, the method finds the instance of size k(> 2) colocations by joining the instances of its size k-1 subset colocation where the first k-2 objects are common. This approach is computationally expensive with the increase of colocation patterns and their instances as in Fig:2.

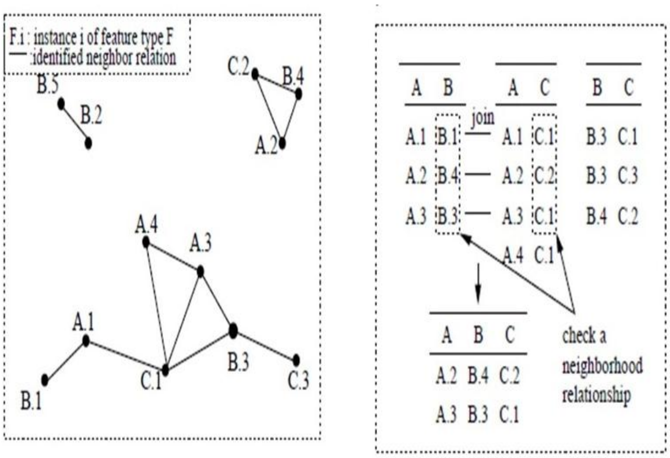
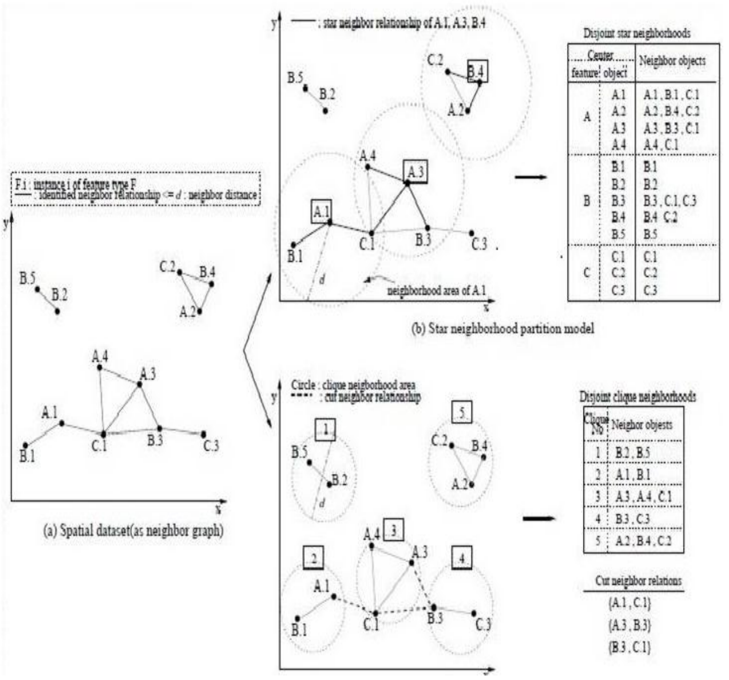


Fig. 2. Join-Based Approach

2.3. Join-Less Approach

The join-less approach puts the spatial neighbor relationship between instances into a compressed star neighborhood. All the possible table instances for every colocation pattern were generated by scanning the star neighbourhood, and by 3-time filtering operation. This join-less colocation mining algorithm is efficient since it uses an instance look-up schema instead of an expensive spatial or instance join operation for identifying colocation table instances, but the computation time of generating colocation table instances will increase with the growing length of colocation pattern as in Fig:3.


F.i : instance i of feature type F
—— : identified neighbor relationship <= d : neighbor distance
(a) Spatial dataset(as neighbor graph)
—— : star neighbor relationship of A.1, A.3, B.4
neighborhood area of A.1
(b) Star neighborhood partition model
Circle : clique neighborhood area
----- : cut neighbor relationship
A.1
A.2
A.3
A.4
B.1
B.2
B.3
B.4
B.5
C.1
C.2
C.3
d
x
y

Disjoint star neighborhoods

Center feature	Center object	Neighbor objects
A	A1	A1, B1, C1
	A2	A2, B4, C2
	A3	A3, B3, C1
	A4	A4, C1
B	B1	B1
	B2	B2
	B3	B3, C1, C3
	B4	B4 C2
	B5	B5
C	C1	C1
	C2	C2
	C3	C3

Disjoint clique neighborhoods

Clique No	Neighor objests
1	B2, B5
2	A1, B1
3	A3, A4, C1
4	B3, C3
5	A2, B4, C2

Cut neighbor relations

{A1, C1}
{A3, B3}
{B3, C1}

Fig: 3 Join-Less Approach

2.4. CPI-tree Algorithm

This algorithm proposed by Wnag et al in[11] developed in new structure called CPI-tree(colocation pattern instance tree) which could materialize the neighbor relationships of spatial data sets, and find all the table instances recursively from it. This method gives up Apriori like model, (i.e.) to generate size-k prevalence colocations after size(k-1) prevalence colocations, but Apriori candidate generate-test method reduces the number of candidate sets significantly and leads to performance gain.

2.5. Morimoto[8]

It was the first to define the problem in finding frequent neighbouring colocations in spatial databases based on number of instances of colocation, to measure the prevalence colocation but with a drawback not possessing the anti-monotone property.

2.6. Huang et al.[6]

In this paper a general framework was proposed for a prior-gen based colocation mining, in which minimum-participation ratio measure was taken instead of support, in which anti-monotone property which increases the computational efficiency. Later a paper[14],[16] was published which proposed a join-based algorithm to find prevalent colocation patterns, but as the size of the data set grows the number of joins increases. Later Huang et al. extended the problem to mining confident colocation patterns in which maximum participation ratio was taken instead of minimum participation ratio which is used to measure the prevalence of confident colocation.

2.7. Yoo etal.[9],[10]

Proposed two algorithms, one among these is partial-join algorithm and the other is join-less algorithm. These two algorithms discusses the information in which joins are used to identify k-size colocation table instances which were substituted by scanning the materialized neighborhood tables and looking-up size k-1 instances, but in this approach there are some repeated scanning of materialized neighborhoods.

2.8. Wang et al. [11]

A CPI-tree-based approach was developed by storing star-neighbourhoods in a more compact format and a prefix tree instead of a table, which reduces the repeated scans of materialized neighbourhoods as in[9]. In this paper [12] discovered colocation patterns from interval data. As different applications are growing the researchers are more devoted to extend the traditional frequent pattern mining to uncertain data sets. [1], [2], [3].

2.9. Chui et al.[3]

Proposed a method which accurately mine the frequent patterns maintaining the efficiency, later in paper [4], methods were used for finding the frequent items in very large uncertain data sets Besides the above representative colocation mining problem, in this paper we are closely related to finding the prevalent colocations using the Probabilistic approximation approach[13].

3. THE BASIC DEFINITIONS

3.1. Uncertain Data Sets

Uncertain data set is defined as the data that may contain errors or may only be partially complete. Many advanced technologies have been developed to store and record large quantities of data continuously. In many cases, For Example:

1. Demographic data sets, Provides only partially aggregated data sets because of privacy concerns.
2. The output of sensor networks is uncertain because of the noise in sensor inputs or errors in wireless transmission.
3. Geographic information systems may contain partial data because of privacy Concern.
4. Data collected from satellites.

Thus each aggregated record can be represented by a probability distribution. Many uncertain reasoning methods, such as fuzzy set theory, evidence theory, and neural networks, are powerful computational tools for data analysis and have good potential for data mining as well. But traditional spatial data mining and knowledge discovery did not pay attention to these characteristics. In this paper, on the basis of analysis of uncertainty in spatial data is analyzed briefly.

3.2. Probabilistic Approach

Probabilistic approaches enable variation and uncertainty to be quantified, mainly by using distributions instead of fixed values in risk assessment. A distribution describes the range of possible values and shows which values within the range are most likely. Probabilistic approach is used in the context of uncertain data as data is collected from a wider range of data sources.

Table . 1. A Sample Example Of Spatial Uncertain Data Set

Id if Instance w	Spatial Feature	Location	Probability
1	A1	in Fig.1	0.1
2	A2	in Fig.1	0.4
3	A3	in Fig.1	0.7
4	B1	in Fig.1	0.1
5	B2	in Fig.1	1
6	C1	in Fig.1	1
7	C2	in Fig.1	0.1
8	D1	in Fig.1	0.4
9	D2	in Fig.1	0.1

3.3. Spatial Data

Spatial data also known as geo-spatial data is the information which identifies the geographic location of features and boundaries on Earth, such as Forests, Oceans etc., Usually Spatial data is stored in terms of numeric values.

3.4. Colocation Mining

It is the process of finding patterns that are colocated in nearby regions. Co-location rule process finds the subsets of features whose instances are frequently located together in geographic space. Many important applications use colocation mining. For example:

1. NASA (studying the climatologically effects, land use classification),
2. National Institute of Health (predicting the spread of disease),
3. National Institute of Justice (finding crime hot spots),
4. Transportation agencies (detecting local instability in traffic).

It is found that classical data mining techniques are often inadequate for spatial data mining and different techniques need to be developed. For this we discuss the co-location pattern mining over spatial data sets.

3.5. Spatial Colocation Mining

It is a group of spatial features whose instances are frequently located around the geographic space. Let F= $\{f_1, f_2, \dots\dots\dots f_n\}$ be the set of features and Z= $\{P_1, P_2, \dots\dots\dots\dots\dots., P_n\}$ where $\{P_1, P_2, \dots\dots\dots\dots\dots., P_n\}$are the subsets of features $\{f_1, f_2, \dots\dots\dots f_n\}$ Let T be the threshold set {d, $\min\_ prev$, P_m} then C ∊ Z such that for C, T is valid. For example from the Fig:1 we can identify the features and instances related in a spatial data set

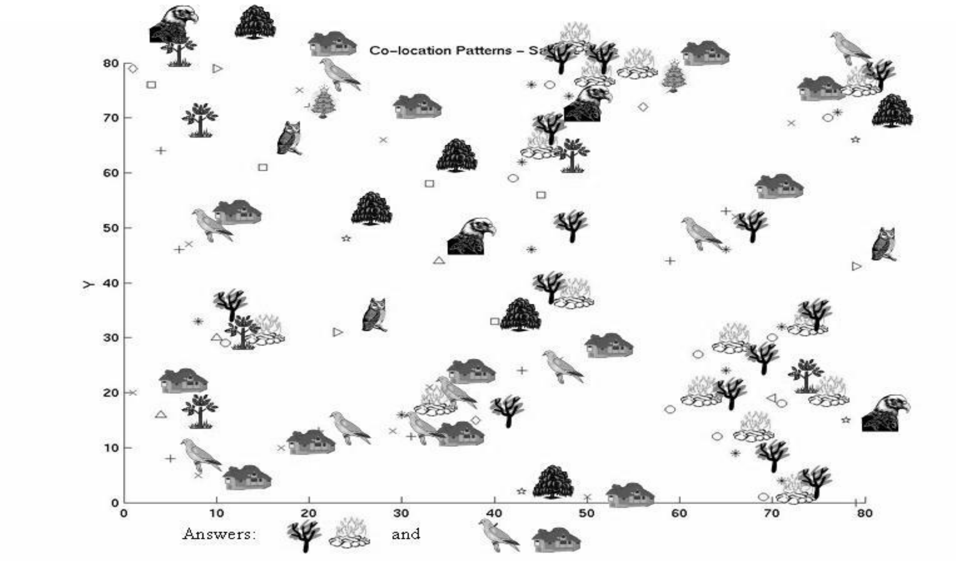


Fig: 4 Example of Spatial Colocation data

From the Fig:4 we can identify that there are different types of features like tree, Bird, Rocks and House and we have instances for the features like trees which are of various types of trees, and Birds which are like Eagle, Sparrow, Owl, and the Features like rock and house are having only one kind of instance. From the figure we can conclude that rocks and a type of tree is colocated, Sparrow and house are colocated.

From the Fig:4 we can identify that there are different types of features like tree, Bird, Rocks and House and we have instances for the features like trees which are of various types of trees, and Birds which are like Eagle, Sparrow, Owl, and the Features like rock and house are having only one kind of instance. From the figure we can conclude that rocks and a type of tree is colocated, Sparrow and house are colocated.

3.6. Instance of a Feature

The instances of a feature are the existential probability of the instance in the place location. If F is a feature then $F.i$ is an instance.

3.7. Spatially Uncertain Feature

A spatial feature contains the spatial instances, and a data set Z containing spatially uncertain features is called spatially Uncertain data set. If Z is a data set then set of features are A, B, C,...

3.8. Probability of Possible Worlds

For each colocation of k-size, c={$f_1, f_2, \ldots\ldots\ldots\ldots f_n$ of each instance $F.i$ there are two different possible worlds (i) one among them is that the instance is present (ii) and the other is absent.

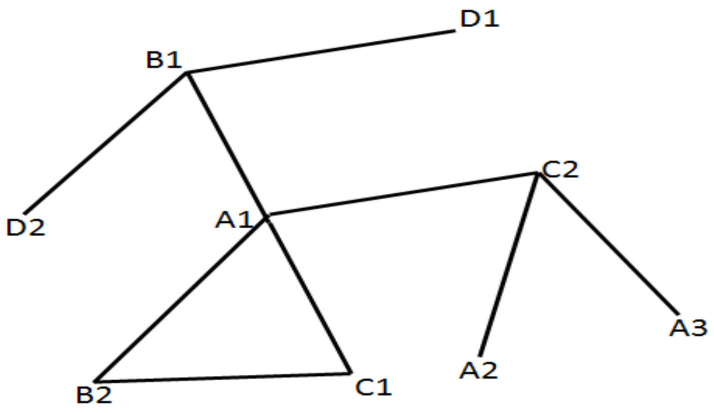


Fig: 5 Distribution of example spatial Instance

Take the set of features F=$\{f_1, f_2, \ldots\ldots\ldots f_n\}$ and the set of instances S= $\{S_{f_1}, S_{f_2}, \ldots\ldots\ldots\ldots, S_{f_n}\}$, where S_{f_i} $(1 <= i <= k)$ is the set of instances in S and there are $2^{|S|}$= $2^{\left|s_{f_1}, s_{f_2}, \ldots\ldots\ldots\ldots, s_{fn}\right|}$ possible worlds at most. Each Possible world w is associated with a probability P (w) that is the true world, where P (w) > 0.

3.9. Neib_tree

The Neib_tree is constructed for the Table-I which indicates the existence of the path from one feature to the other. If there is a path it indicates that a table instance is existing. This Neighbouring tree eliminates the duplicates can be seen in Fig:6.

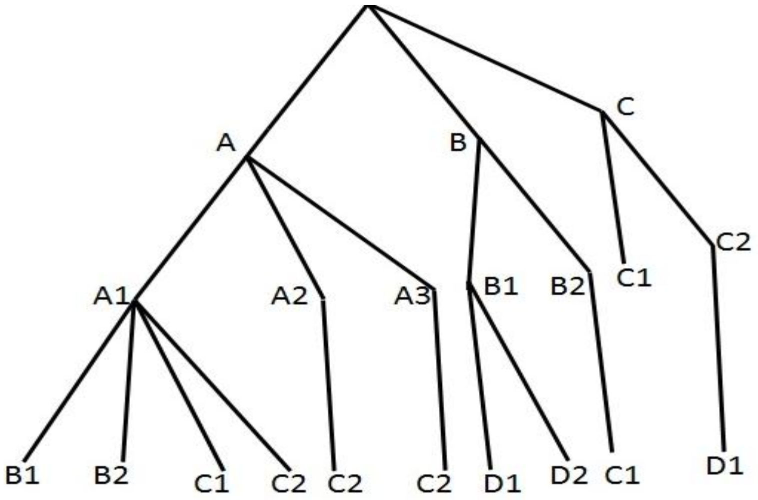


Fig: 6 Neib_tree for Fig:5

4. Block Diagram

Basic flow of co-location pattern mining: In this section, we present a flow diagram which describes the flow of identifying the Probabilistic Prevalent colocations. Given a Spatial data set, a neighbour relationship, and interest measure thresholds the basic colocation pattern mining involves 4 steps as in Fig: 3

Table 2. Computational Process Of Colocation (A,C)

A1	A2	A3	C1	C2
0	0	0	0	0
0	0	0	0	1
0	0	0	1	0
0	0	0	1	1
0	0	1	0	0
0	0	1	0	1
0	0	1	1	0
0	0	1	1	1
0	1	0	0	0
0	1	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	0
0	1	1	0	1
0	1	1	1	0
0	1	1	1	1
1	0	0	0	0
1	0	0	0	1
1	0	0	1	0
1	0	0	1	1
1	0	1	0	0
1	0	1	0	1
1	0	1	1	0
1	0	1	1	1
1	1	0	0	0
1	1	0	0	1
1	1	0	1	0
1	1	0	1	1
1	1	1	0	0
1	1	1	0	1
1	1	1	1	0
1	1	1	1	1

First candidate colocation patterns are generated and the colocation instances and spitted into two worlds from the spatial data set. Next, find the probabilities using minimum prevalence and compute summation of table instances of each colocation, Next find prevalent colocation using minimum probability.

5. THE BASIC ALGORITHM

The algorithm (Algorithm-1) is designed to find all PPCs with (min_prev, min_prob) pairing. The algorithm uses dynamic approach where in it prunes out the candidates which are not prevalent and works on the reduced search space to find the PPCs. It uses an approximation

Table 3. Computational Process Of Colocation(A,C)

Possible World$_w$	P(w_i)
w_1={C1}	0:1458
w_2={C1,C2}	0:0162
w_3={A3,C1}	0:3402
w_4={A3,C1,C2}	0:0378
w_5={A2,C1}	0:0972
w_6={A2,C1,C2}	0:0108
w_7={A2,A3,C1}	0:2268
w_8={A2,A3,C1,C2}	0:0252
w_9={A1,C1}	0:0162
w_{10}={A1,C1,C2}	0:0018
w_{11}={A1,A3,C1}	0:0378
w_{12}={A1,A3,C1,C2}	0:0042
w_{13}={A1,A2,C1}	0:0108
w_{14}={A1,A2,C1,C2}	0:0012
w_{15}={A1,A2,A3,C1}	0:0252
w_{16}={A1,A2,A3,C1,C2}	0:0028

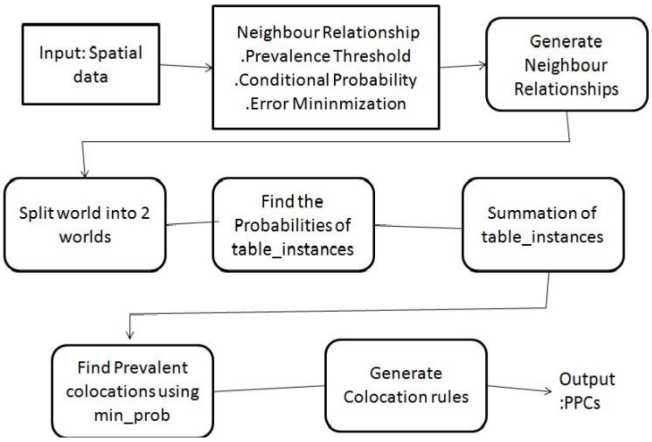

Input: Spatial data
Neighbour Relationship
.Prevalence Threshold
.Conditional Probability
.Error Mininmization
Generate Neighbour Relationships
Split world into 2 worlds
Find the Probabilities of table_instances
Summation of table_instances
Find Prevalent colocations using min_prob
Generate Colocation rules
Output :PPCs

Fig:7 Block diagram to find the PPCs

approach by accepting an initial error that would be tolerated in finding the PPCs and thereby speeds up the process of finding the PPCs. The algorithm is presented below:

Algorithm-1

Input:

$F = (f_1, f_2, \dots\dots\dots, f_n)$ a set of Spatial Features;
S: A spatially uncertain data set;
min_prev: A minimum prevalence threshold;
min_prob: A minimum Probability Threshold;
e: An Approximation error;
Probability of table instances:

$$P_r^{(c,f_1)}[0,0]_w = 1;$$
$$P_r^{(c,f_1)}[0,0]_w = 1;$$
$$P_r^{(c,f_1)}[0,0]_w = 0\ (1 < i < 1);$$

```
            $P_r^{(c,f_1)}[0,0]_w = 0\ (1 < i < 1);$
Output:-
($min\_prev$, $min\_prob$) PPCs.
Begin
    1) Read approximation error e.
    2) if e=1 STOP
    3) else
    4) Call Neib_tree_gen(F, S, NHR); // to identify table instances.
    5) Assign $P1 = F$, $k = 2$;
    6) While (not empty $P_{k-1}$ and $k \leq n$) do
        (i) for each colocation $"W"$ of size $'k'$ compute Probabilities of worlds from
            equation-3:
        (ii) Split W into W$_1$ and W$_2$ where
            $W_1 = f_1.j > (f_2.j, f_3.j \ldots\ldots, f_n.j)$ $W_2 = f_2, f_3 \ldots\ldots, f_n,$
            and W$_2 \subseteq$ w;
        (iii) for each set w=$(f_1.l, \ldots\ldots\ldots, f_n.l)$ compute Probability of table _instances as
            equation-4:.
        (iv) for each w compute Prevalence Probability
            $P(PR^R(c) \geq min\ prev)_{w_{1+w}}$ as equation-5:
        (v) Compute the summation of all Prevalence Probabilities
            $PPs = PPs + (P1 + P2 + \ldots\ldots\ldots + Pn)$
        (vi) if $(PPs \leq min\_prob)$ then c=c-C$_k$;
        (vii) $P_k$=sel_prev_colocation(C$_k$, $min\_prev$, $min\_prob$);
        (viii) $k = k + 1$;
        (ix) $end\ while$;
    7) STOP;
    8) Return $(P2 \cup P3 \ldots\ldots\ldots . \cup Pn)$
End.
```

$$P(W) = \prod_{i=1}^{n}\left(\prod_{(e\in s_{f_i})\in w} P(e) * \prod_{(e\in s_{f_i})\ni w}(1 - P(e))\right) \quad (3)$$

$$P^{(c,f_1)}[i,j]_w = \begin{cases} P^{(c,f_1)}[i,j]_w \ if, \ f_1.j \in table_{instance_{w\cup f_1^j}}(c) \\ P^{(c,f_1)}[i,j-1]_w .(1-p_j) + P^{(c,f_1)}[i-1,j-1]_w .p_j \\ \qquad if, f_1.j \in table_{instance_{w\cup f_1^j}}(c) \\ \qquad and \geq j.min\ \_previstrue \\ 0 \qquad otherwise, \end{cases}$$

$$P^{(c,f_1)}[i,j]_w = \begin{cases} P^{(c,f_1)}[i,j-1]_w \quad if, \ f_1.j \notin table_{instance_{w\cup f_1^j}}(c) \\ P^{(c,f_1)}[i,j-1]_w .(1-p_j) + P^{(c,f_1)}[i-1,j-1]_w .p_j \quad (4) \\ \qquad otherwise, \end{cases}$$

$$\left(\sum_{i=1}^{l_1} P^{(c,f_1)}[i,j]_w \left(\sum_{j=0}^{\frac{\lfloor 1-\min\_prev \rfloor}{\min\_prev} * i} P^{(c,\overline{f_1})}[j, l_1]_w\right)\right) \quad (5)$$

6. Tracing The Algorithm

6.1 Step 1,2,3. Reading the value of e

if the value of e is 1 then the algorithm stops and prints that all colocations are Prevalent. Otherwise if the value is in between 0<e<1 then execute steps from 4 to 14.

6.2. Step 4, 5: The Initializing Steps

After finding all neighbouring instance pairs, a Neib_tree can be generated using the method [5]. For example fig:2 are a Neib_tree generate from Fig:1 These Neib_tree consist of a set of features which are organized in ordered and branched form.

6.3. Step (i): Generating Coarse Combination instances from each collocation

This step computes the coarse combinations of different colocation of k-size. For example for colocation (A,C) we get a set of 24 combination instances out of 25 combinations whose probability is greater than zero.

6.4. Step (ii): Splitting of Colocation instances

Splitting of a colocation into two different worlds (i.e.)., colocation based on the set of features which has largest number of instances. W_1 is the set of possible worlds of ff1g and W_2 is that of possible set of worlds of $\{f_1, f_2, \dots\dots\dots, f_n\}$. For example in this paper the Colocation (A, C) are divided into 2 worlds out of which W_1 in consisting of all instances of {A} & {A, C} and W_2 consisting alone {C} instances (i.e.), {C1} & {C1, C2}.

6.5. Step (iii): Computing the Probability of table instances in world W_2

Computing the Probability of table instances W_2 where W_2 is consisting of ({C1},{C1,C2}) using the equations-(4) (i,e)., *for* $Pr(c, f_1)[i,j]\{C2\}$ and $Pr(c, \bar{f}_1)[i,j]\{C2\}$. After finding the Probabilities the values can be seen in TABLE IV and V :

Table 4. The Computation Of The $P(c, f_1)[i,j]\{C1\}$ And $P(c, \bar{f}_1)[i,j]\{C1\}$

	j =0	j =1	j =2	j =3
i=0	(1,1)	(0.7,1)	(0.7,0.6)	(0.7,0.24)
i =1	(0,0)	(0.3,0)	(0.3,0.4)	(0.3,0.52)
i =2	(0,0)	(0,0)	(0,0)	(0,0.24)
i =3	(0,0)	(0,0)	(0,0)	(0,0)

Table 5. The Computation Of The $P(c, f_1)[i, j]\{C1, C2\}$ And $P(c, \bar{f}_1)[i, j]\{C1, C2\}$

	j=0	j =1	j =3	j =4
i =0	(1,1)	(0.7,1)	(0.42,1)	(0.42,0.4)
i =1	(0,0)	(0.3,0)	(0.46,0)	(0.46,0.6)
i =2	(0,0)	(0,0)	(0.12,0)	(0.12,0)
i =4	(0,0)	(0,0)	(0,0)	(0,0)

6.6. Step (iv): Computing the Prevalence Probability of world W_2

After computing step-9 for each set colocation of k-size, now compute the prevalence probability from equation: 5 For example for colocation (A,C) if the min_prev is 0.5 then for the table_ instances { C1} the value is 0.205 and for{ C1, C2 } the value is 0.058.

6.7. Step (v): Summation of Prevalence Probabilities

After computing the prevalence Probability of all colocation then we make the summation of all Prevalence Probability. For example for colocation (A, C) the value of {C1} is 0.2052 and {C1, C2} is 0.058 and the summation of both { C1 } & {C1, C2} is 0.2632

6.8. Step (vi), (vii) : Checking with Minimum Probability

if the summation is less than the minimum probability then it is removed from Probabilistic Prevalent Colocations, Otherwise added to prevalent Colocation. From the above example if the min_prob is 0.3 then colocation (A, C) is filtered and if it is 0.2 then colocation is selected.

6.9. Step (ix)

The colocation size is increased and Steps from 6 to 13 are executed.

6.10. Step 7

Once all the Probabilistic Prevalent Colocations are identified the algorithm stops.

6.11. Step 8

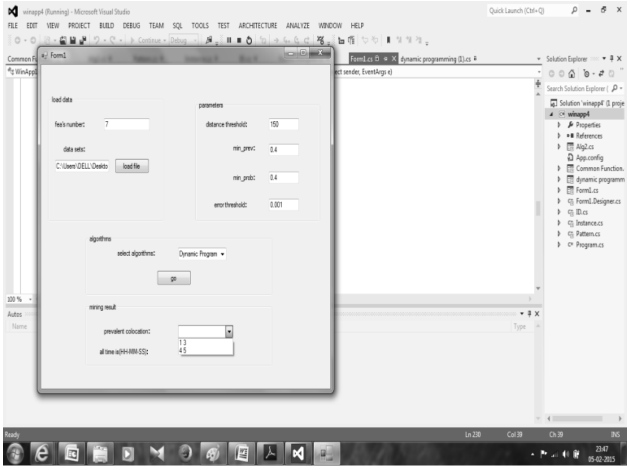
A Union of all Probabilistic Prevalent Colocations are written from a set of features.

7. RESULTS

The results are compared against a data set given in the following Table-VI which consists of 7 features with an average of 2 instances.

Table 6. A Synthetic Sample Data Set

Features	X-Coordinates	Y-Coordinates	Probability
0	328	1362	0.5
0	190	1140	0.4
0	392	1220	0.9
1	290	1264	0.1
1	330	1480	1
2	260	1278	0.1
3	185	1440	0.1
3	320	1500	0.4
3	330	1500	0.7
4	150	1580	0.1
4	150	1300	1
5	225	1300	1
5	260	1530	0.1
6	220	1650	0.4
6	60	1590	1


winapp4 (Running) - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM SQL TOOLS TEST ARCHITECTURE ANALYZE WINDOW HELP
Form1
load data
parameters
algorithms
mining result
load file
go
Ready

fea's number: 7

data sets: C:\Users\DELL\Deskto

distance threshold: 150

min_prev: 0.4

min_prob: 0.4

error threshold: 0.001

select algorithms: Dynamic Program

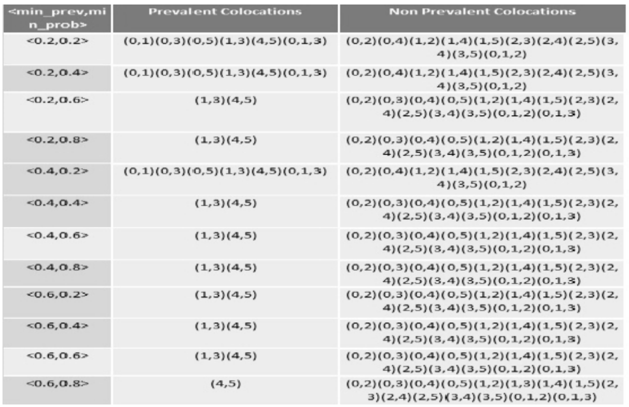
prevalent colocation:

1 3
4 5

all time is(HH-MM-SS):

Fig:8 PPCs for Table-VI with $min\_prev = 0.4$ and $min\_prob = 0.4$, d=150,and $\varepsilon = 0.001$

Likewise when the comparisons are made against the complete data set from Table-VI we get the following Prevalent and non-Prevalent colocations, varying the min_prev and min_prob for the distance threshold=150 which are shown in Fig:9.



<min_prev,min_prob>	Prevalent Colocations	Non Prevalent Colocations
<0.2,0.2>	(0,1)(0,3)(0,5)(1,3)(4,5)(0,1,3)	(0,2)(0,4)(1,2)(1,4)(1,5)(2,3)(2,4)(2,5)(3,4)(3,5)(0,1,2)
<0.2,0.4>	(0,1)(0,3)(0,5)(1,3)(4,5)(0,1,3)	(0,2)(0,4)(1,2)(1,4)(1,5)(2,3)(2,4)(2,5)(3,4)(3,5)(0,1,2)
<0.2,0.6>	(1,3)(4,5)	(0,2)(0,3)(0,4)(0,5)(1,2)(1,4)(1,5)(2,3)(2,4)(2,5)(3,4)(3,5)(0,1,2)(0,1,3)
<0.2,0.8>	(1,3)(4,5)	(0,2)(0,3)(0,4)(0,5)(1,2)(1,4)(1,5)(2,3)(2,4)(2,5)(3,4)(3,5)(0,1,2)(0,1,3)
<0.4,0.2>	(0,1)(0,3)(0,5)(1,3)(4,5)(0,1,3)	(0,2)(0,4)(1,2)(1,4)(1,5)(2,3)(2,4)(2,5)(3,4)(3,5)(0,1,2)
<0.4,0.4>	(1,3)(4,5)	(0,2)(0,3)(0,4)(0,5)(1,2)(1,4)(1,5)(2,3)(2,4)(2,5)(3,4)(3,5)(0,1,2)(0,1,3)
<0.4,0.6>	(1,3)(4,5)	(0,2)(0,3)(0,4)(0,5)(1,2)(1,4)(1,5)(2,3)(2,4)(2,5)(3,4)(3,5)(0,1,2)(0,1,3)
<0.4,0.8>	(1,3)(4,5)	(0,2)(0,3)(0,4)(0,5)(1,2)(1,4)(1,5)(2,3)(2,4)(2,5)(3,4)(3,5)(0,1,2)(0,1,3)
<0.6,0.2>	(1,3)(4,5)	(0,2)(0,3)(0,4)(0,5)(1,2)(1,4)(1,5)(2,3)(2,4)(2,5)(3,4)(3,5)(0,1,2)(0,1,3)
<0.6,0.4>	(1,3)(4,5)	(0,2)(0,3)(0,4)(0,5)(1,2)(1,4)(1,5)(2,3)(2,4)(2,5)(3,4)(3,5)(0,1,2)(0,1,3)
<0.6,0.6>	(1,3)(4,5)	(0,2)(0,3)(0,4)(0,5)(1,2)(1,4)(1,5)(2,3)(2,4)(2,5)(3,4)(3,5)(0,1,2)(0,1,3)
<0.6,0.8>	(4,5)	(0,2)(0,3)(0,4)(0,5)(1,2)(1,3)(1,4)(1,5)(2,3)(2,4)(2,5)(3,4)(3,5)(0,1,2)(0,1,3)

Fig: 9 PPCs for Table-VI with varying min_prev and min_prob, and d=150,and ε = 0.001

As expected, the smaller the min_prev and min_prob values lead to an increase in number of PPCs which increases the computation time as shown in Fig:10.

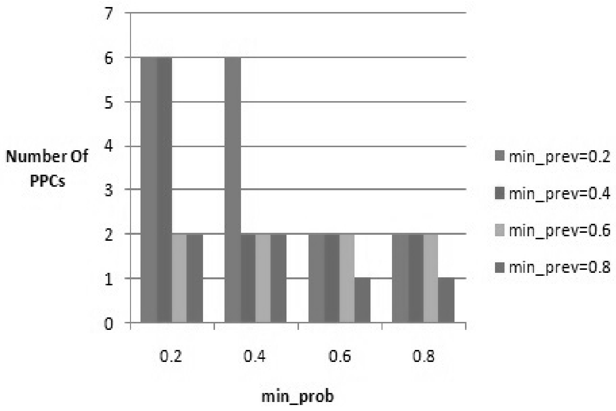


Fig: 10 Varying min_prev and min_prob and d=150, and ε = 0.001

From the graph below it is proved that the computation time for the improved Approximation algorithm works well when compared to dynamic algorithm: as shown in Fig: 11.

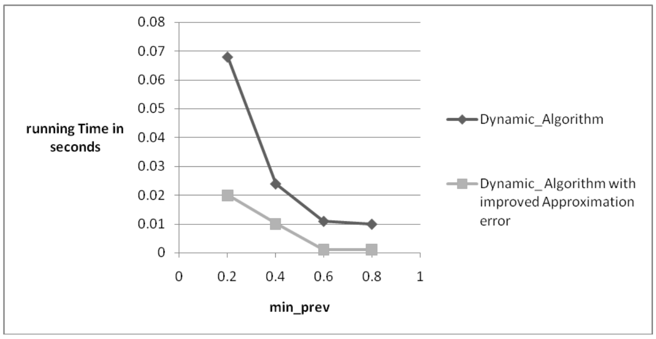


Fig.11: Varying min_prev and min_prob, d=150, and ε= 0.001

8. CONCLUSION

We have proposed a method for finding Probabilistic Prevalent Colocation in Spatially Uncertain data sets which are likely to be prevalent. We have given an approach in which the computation time is drastically reduced. Future Work can include the parallel computation for finding the Prevalent Colocation which are evaluated independently and this work can also be expanded to find the Probabilistic Prevalent colocations in other Spatially Uncertain data models, for example fuzzy data models and graphical spatial data. Further keeping in view the work can be extended to find the important sub functionalities in colocation mining to formulate colocation mining specific primitives for the next generation programmer which we can expect to evolve as a scripting language. In essence the scope of the work can cover data base technologies, parallel programming domain, graphical graph methods, programming language paradigms and software architectures.

REFERENCES

[1] C.C. Aggarwal et al, ”Frequent Pattern Mining with Uncertain Data,” Proc. 15th ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining(KDD), pp. 29-37, 2009.

[2] T. Bernecker, H-P Kriegel, M. Renz, F. Verhein, and A. Zuefle, ”Proba-bilistic Frequent Itemset Mining in Uncertain Databases,” Proc. 15th ACM SIGKDD Conf. Knowledge Discovery and Data Mining(KDD ‘09), pp. 119-127, 2009. [05]

[3] C.-K. Chui, B. Kao, and E. Hung, ”Mining Frequent Item sets from Uncertain Data,” Proc. 11th Pacific-Asia Conf. Knowledge Discovery and Data Mining(PAKDD), pp. 47-58, 2007.

[4] C.-K. Chui, B. Kao, ”A Decremental Approach for Mining Frequent Item sets from Uncertain Data,” Proc. 12th Pacific-Asia Conf. Knowledge Discovery and Data Mining(PAKDD), pp. 64-75, 2008.

[5] Y. Huang, H. Xiong, and S. Shekar, ”Mining Confident Co-Location Rules Without a Support Threshold,” Proc. ACM Symp. Applied Com-puting, pp. 497-501, 2003.

[6] Y. Huang, S. Shekar, and H. Xiong, ”Discovering Co-Location Patterns from Spatial Data Sets: A General Approach,” IEEE Trans. knowledge and Data Eng., vol. 16, no. 12, pp. 1472-1485, Dec. 2004.

[7] Y. Huang, J. Pei, and H. Xiong, ”Mining Co-Location Patterns with Rare Events from Spatial Data Sets,” Geoinformatics, vol. 10, no. 3, pp. 239-260, Dec. 2006.

[8] Y. Morimoto, ”Mining Frequent Neighbouring Class Sets in Spatial Databases,” Proc. Seventh ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining(KDD), pp. 353-358, 2001.

[9] J.S. Yoo, S. Shekar,J. Smith, and J.P. Kumquat, ”A Partial Join Approach for Mining Co-Location Patterns,” Proc. 12th Ann. ACM Int’l Workshop Geographic Information Systems (GIS), pp. 241-249, 2004.

[10] J.S. Yoo and S. Shekar, ”A Join less Approach for Mining Spatial Co-Location Patterns,” IEEE Trans. knowledge and Data Eng.(TKDE), vol. 18, no. 10, pp. 1323-1337, Dec. 2006.

[11] L. Wang, Y. Bao, J. Lu and J. Yip, ”A New Join-less Approach for Co-Location Pattern Mining,” Proc. IEEE Eighth ACM Int’l Conf. Computer and Information Technology (CIT), pp. 197-202, 2008.

[12] L. Wang, H. Chen, L. Zhao and L. Zhou, ”Efficiently Mining Co-Location Rules of Interval Data,” Proc. Sixth Int’l Conf. Advanced Data Mining and Applications, pp. 477-488, 2010.

[13] Q. Zhang, F. Li, and K. Yi, ”Finding Frequent Items in Probabilistic Data,” Proc. ACM SIGMOD Int’l Conf. Management of Data, pp. 819-832, 2008.

[14] L. Wang, P. Wu, and H. Chen, ”Finding Probabilistic Prevalent Colocations in Spatially Uncertain Data Sets,” IEEE Trans. knowledge and Data Eng.(TKDE), vol. 25, no. 4, pp. 790-804, Apr. 2013.