Indian Journal of Science and Technology, Vol 10(7), DOI: 10.17485/ijst/2017/v10i7/106709, February 2017
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Computation Analysis for Finding Co–Location Patterns using Map–Reduce Framework

M. Sheshikala[1,2]*, D. Rajeswara Rao[1] and R. Vijaya Prakash[2]

[1]Department of Computer Science and Engineering, KL University, Vijayawada – 520002, Andhra Pradesh, India; marthakala08@gmail.com, rajeshduvvada@kluniversity.in
[2]Department of Computer Science and Engineering, SR Engineering College, Warangal – 506371, Telangana, India; vijprak@hotmail.com

Abstract

Objectives: The main objectives of the paper are 1. Generating the Neib_tree based on the number of features and instances. 2. Finding all the co-location patterns using Parallel Approach. 3. Improving the computation time by exploring Map-Reduce Framework. **Methods:** To generate Neib-tree is by Grid based approach. The method used find co-location patterns is by parallel approach which drastically increases the time complexity. The exploratory results are directed by utilizing manufactured information sets by taking the different data sets one with 25k, 50k and 75k features and an average of 20k instances each, which produces the computational analysis with a distance of 20km. **Findings:** This paper presents fast calculation of co-location patterns where these is helpful in finding the people suffering from a particular problem in a place and what are the patterns affecting the problem. The proposed work diminishes the calculation time by 1/n terms where n is the quantity of components as it uses a Map-Reduce system. This paper presents exact and fulfillment of the new approach. At long last, exploratory assessments utilizing manufactured information sets demonstrate the calculation is computationally more productive. **Applications**: The concept presented in this paper is helpful in different areas like medical Field, NASA, and etc., **Improvements**: The paper improves the time complexity and space complexity by using parallel join-less approach.

Keywords: Co-Location Mining, Map-Reduce, Participation Index, Participation Ratio, Time Complexity

Introduction

Spatial information mining is a procedure of discovering valuable and intriguing examples from spatial items. Separating fascinating examples from spatial articles is a troublesome errand since it incorporates spatial information sorts, spatial connections and spatial auto relationship.

A Spatial co-location design speaks to a subset of spatial elements whose occurrences are as often as possible co-situated in a spatial neighbourhood1. For instance, a school and library are as often as possible co-found. The co-location govern, (i.e.,) school >library predicts the nearness of library in ranges where school is found. In numerous applications finding spatial co-location designs assumes an imperative part.

Co-location lead revelation is a procedure to distinguish co-location designs from spatial information set. Co-location mining guideline presents challenges because of taking after reasons:

1. In the first place, discovering co-location occurrences is a troublesome errand since spatial articles are settled in a ceaseless space and they share neighbor connections. Here a lot of calculation is totally dedicated to discover the cases of co-location designs.

2. As spatial articles have no predefined exchange we can't characterize affiliation manage digging for producing co-location designs.

The present approach emerges the spatial neighbor connections with no loss of co-location cases and lessens the computational cost of distinguishing the co-location

*Author for correspondence

occasions utilizing an occurrence look-into composition, however as the quantity of components and their related examples builds the calculation time for creating the co-found examples additionally increments as the approach takes after the consecutive way.

In this paper, we propose a parallel join-less approach for co-location design mining which 1) finds the spatial neighbor connections with no loss of co-location occurrences. 2) Reduces the calculation cost of recognizing the co-location runs by utilizing a guide decrease structure.

The issue of mining co-location rules in view of spatial connections (e.g., closeness, nearness) was talked about in[1]. The work finds the subset of spatial components every now and again connected with a particular element, e.g.: coronary illness. Join Based approach finds the right and finish co-location occasions, this approach is computationally costly with the expansion of co-location designs and their examples.

It[2] has talked about the two calculations, one among these is halfway join calculation and the other is joining-less calculation, yet in this approach there are some rehashed filtering of appeared neighborhoods. Join-less calculation emerges neighbor connections of spatial information for productive co-location design mining utilizing both methodologies star neighborhood and inner circle neighborhood parceling. This calculation is proficient since it utilizes a case look-into mapping however as the quantity of elements and its occurrences expands the computational unpredictability of creating co-location guidelines will increment.

In[3] CPI-tree-based approach was produced by putting away star-neighborhoods in a smaller arrangement and a prefix tree rather than a table, which diminishes the rehashed outputs of appeared neighborhoods as in[4]. In this paper[5] found co-location designs from interim information. As various applications are developing the analysts are more given to broaden the customary successive example mining to indeterminate information sets.

In[6] proposed a strategy which precisely mines the continuous examples keeping up the productivity, techniques was utilized for finding the incessant things in substantial questionable information sets. Other than the above delegate co-location mining issue, in this paper we are firmly identified with finding the common co-locations utilizing the Probabilistic guess approach[7].

In[8] a general system was proposed for an earlier gen based co-location mining, in which least interest proportion measure was taken rather than support, in which against monotone property which builds the computational productivity. Later a paper[9] was distributed which proposed a join-based calculation to discover pervasive co-location designs; however as the measure of the information set develops the quantity of joins increments. Later Huang et al. extended the issue to mining sure co-location designs in which most extreme support proportion was taken rather than least interest proportion which is utilized to quantify the predominance of sure co-location.

2. Our Contributions

We ventured out presenting a parallel join-less calculation for co-location design revelation. The accompanying commitments are made in this work:

To start with, we propose a parallel way to deal with appear the neighbor connections of spatial information for productive co-location design mining. We exhibit a Grid-based apportioning approach for discovering this spatial neighborhood objects.

This calculation is productive since it is based upon Grid based segment model and it utilizes a parallel case[10] look-into outline for sifting co-location occurrences. It likewise has a coarse pruning step which can channel competitor co-locations parallel without finding the correct co-location occurrences.

Third, we apply the club parcel model to spatial information particularly bunched in neighborhood regions with another coarse separating composition. The whole handling is done parallel which radically decreases the calculation time.

We assess the arithmetical cost models to dissect the execution of this parallel join-less strategy utilizing Map-Reduce Framework[11]. At last assess out calculation utilizing manufactured information sets and genuine datasets.

3. Basic Concepts

In this we discuss the basic concepts and definitions of co-location pattern mining.

3.1 Neighbor Relationship

Accepting an arrangement of components F, its related occasion objects S and a neighbor relationship R over S a co-location C is a subset of spatial elements The neighbor connection R is an Euclidean separation with its edge d,

and two spatial items are neighbors on the off chance that they fulfill the neighbor relationship (W.1, X.1) ↔ remove (W.1, X.1) ≥d).

3.2 Instance of an Element

On the off chance that there is an element "F" then it can have "n" number of occurrences I to such an extent that n ≥ 1 connected with the element. For instance, in Figure 1 Feature W has 3 occurrences W.1, W.2, W.3.

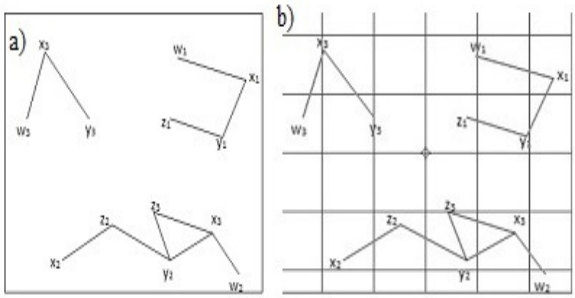

Figure 1. (a) A sample input spatial data set (b) Grid partitioning for neighborhood.

3.3 Co-Location Instance

Co-location is an arrangement of articles related in an inner circle occasion (i.e.,) . For instance, in Figure 1 the co-location occurrences for the arrangement of elements (W, X, Y) are (W.3, X.3, Y.3) since for the element W, W.3 in an example, X.3 is an occasion of highlight sort X and the case of highlight Y is Y.3

3.4 Participation Ratio

The support proportion of highlight i in a co-location C= { f_1,f_2… f_i}is characterized as the quantity of unmistakable objects of highlight i to the aggregate number of objects of i . For instance, in Figure 1((W, X), W) is (W.3. X.3) (W.1, X.1) (W.2, X.3).

3.5 Participation Index

The base of interest proportion over all co-location components is Participation index.(i.e.,) - (1)

For instance in condition (1), at that point PI(c) = min {P(c, W), P(c, X), P(c, Y) } = 2/3

3.6 Map Reduce

Map Reduce[10] is a framework that allows parallel processing among large datasets across several data nodes using a simple programming model.

Map-Reduce job task is to split the given input data into individual lumps, which is completely processed by a parallel approach. The Reducer is then assigned this as input which is the output of mapper. Both the input and output are stored in the file system which is handled by the framework. Monitoring[11] and re-executing[12] the failed tasks and also scheduling the tasks are taken care by this framework[13].

4. Implementation of Map-Reduce Algorithm

4.1 Finding the Neighboring Paths

In this parallel approach mapper[14] assigns the task of generating neighboring paths to different data nodes, as such one data node finds the neighboring path in the region assigned to it, another data node finds the paths in the neighboring paths assigned to it, likewise based on the number of data nodes assigned the paths are generated.

The following Pseudocode-1 gives the information to find neighboring paths.

Pseudo code -1: Generating neighboring relationships using Mapper and Reducer:

1. Proc**Mapper** (key, Value=0)
2. grid value ← find region(o)
3. emit (grid value, o) if dist<= user_threshold
4. end proc.
5. Proc**Reducer** (key=grid_no, value=[o])
6. Distance ← euclidean_distance (x, y) coordinates
7. dist ← Distance ('neighbour distance threshold')
8. if Distance (x, y) ≤ user_threshold
9. eliminate(o)
10. end if
11. add(o)

In the above Map-Reduce procedure the neighboring pairs are generated using a method called Grid Partitioning. The work to be done by the mapper is to allocate the spatial objects in different partitions to different data nodes and find the distance between those objects using Euclidean distance[15]. All the data nodes then compare the distance with a user threshold value and a path is established by the reducer if the specifies distance is greater than or equal to the threshold value.

4.2 Finding Co-Location Patterns

In this section we discuss how to generate the co-location patterns at Mapper-Reducer side.

Step 1: Mapper generates the candidate co-locations by giving one feature to one data node, for example, from Figure 1. One data node generates (W, X), (W, Y), (W, Z) for the feature W, and another data node generates (X, Y), (X, Z) for the feature X, here we eliminate (X, W) since it is already generated by the feature W removing the redundancy. Like-wise all candidate co-location are generated by different data nodes at the same time, and all this candidate co-locations are returned by the Reducer.

Step 2: Mapper generates the Star Instances (SI_k) by using different data nodes. For example, from Figure 1. One data node generates (w3, x3), (w1, x1), (w2, x3) for the feature W in association with feature X, like-wise the other data nodes find the corresponding neighboring paths of all the features with their corresponding features. Later all these are grouped and collected by the reducer.

Step n: The same process is repeated to find the prevalent co-locations and to generate co-location rules.

The following Pseudocode-2 explains how to generate co-location rules at Mapper and Reducer Side.

4.2.1 Mapper Side

1. Proc**Mapper** (o, I, F) initialization.
2. generate_candidate_colocations(o, C_k)
3. Filter_star_instances(SI_k)
4. if k=2 then collect_the_instancesCI_k=SI_k
5. P_k=Select_prevalent_co-locations(C_k, CI_k, min_prev)
6. else if k>=2
7. Select_Coarse_Prevelent_Co_locations(PI_k)
8. Filter_clique_instances(C_k, SI_k)
9. Repeat step 5
10. generate_co-location_rules(R_k(F) , min_prob)
11. end if
12. end

4.2.2 Reducer Side

1. Proc **Reducer**(o, I, f)
2. collect_candidate_co-locations(o,C_k)
3. group_star_instances(SI_k)
4. if k=2
5. return SI_k=CI_k
6. else if k>=2
7. return clique_instances(CI_k)
8. return Coarse_Prevelent_Co_locations(PI_k, min_prev)
9. return co-location_rules(R_k(F))
10. end if

5. Experimental Results

In image processing we take an image which represent different objects and give it to Mat Lab for processing where each and every row is processed and different objects are identified and output is given to text file in raw format where x and y coordinates are separated by comma and each coordinate represent feature associated with different instances.In this exploratory study, we have utilized manufactured spatial information sets with various number of features consider such ,40 which is having around 300 instances by and large. The client characterized neighbor separation is 20km. Figure 2, we utilized 3 group hubs, one is ace and two are slaves. We have executed these models on virtual framework specially designed for this work. Additionally, the apache-hadoop[16] structure was utilized for co-location design mineworker. We have dissected the co-location designs with various least pervasiveness limits. The results are compared and shown in Figure 3.

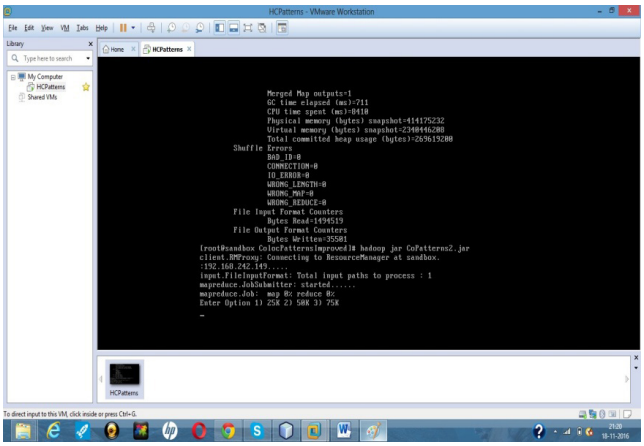

Figure 2. Implementation of different data sets with hadoop framework.

Table 1. Threshold based model, co-location pattern miner and Hadoop based model

#Instancessize	Prevalence Threshold	Threshold Based Patterns	Co-location Miner	Hadoop Based Miner
#100000	0.3	467	398	299
#200000	0.4	646	654	367
#500000	0.5	724	701	543
#700000	0.6	892	811	671

Table 1 describes the different prevalence threshold and its corresponding co-location patterns in different traditional models i.e., Threshold based model, Co-location pattern miner and Hadoop based model.

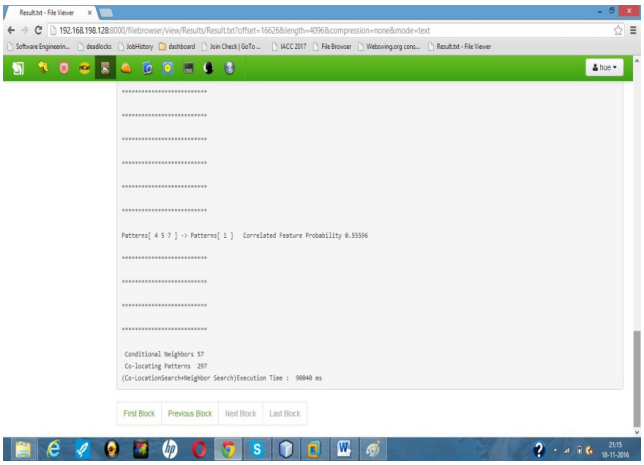

Figure 3. Computation results with hadoop framework.

6. Conclusion and Future Work

In this paper we proposed another parallelized approach for finding the co-location rules. To begin with we isolate the given spatial information set into partitions by utilizing Grid approach. At that point we apply the parallel calculation with a specific end goal to create the co-location rules. The time taken to process the neighboring ways is 1/n times less when contrasted with the before proposed approach where "n" is the quantity of bunch hubs. This paper additionally gives the computational results different information sets. Later on work, we will minimize the co-location designs alongside Mapper and reducer execution time utilizing another information structure.

7. References

1. Huang Y, Shekar S, Xiong H. Discovering co-location patterns from spatial data sets: a general approach. Institute of Electrical and Electronics Engineers (IEEE) Transaction Knowledge and Data Engineering. 2004 Dec; 16(12):1472–85. https://doi.org/10.1109/TKDE.2004.90
2. Huang Y, Pei J, Xiong H. Mining co-location patterns with rare events from spatial data sets. GeoInformatica. 2006 Sep; 10(3):239–60. https://doi.org/10.1007/s10707-006-9827-8
3. Wang L, Bao Y, Lu J, Yip J. A new join-less approach for co-location pattern mining. In the Proceedings of the 8th Institute of Electrical and Electronics Engineers (IEEE) International Conference Computer and Information Technology (CIT); 2008 Jul 8–11. p. 197–202.
4. Yoo JS, Shekar S, Smith J, Kumquat JP. A partial join approach for mining co- location patterns. In the Proceedings of the 12th Annual Association for Computing Machinery (ACM) International Workshop Geographic Information Systems (GIS); 2004 Nov 12–13. p. 241–9. https://doi.org/10.1145/1032222.1032258
5. Wang L, Wu P, Chen H. Finding probabilistic prevalent co-locations in spatially uncertain data sets. Institute of Electrical and Electronics Engineers (IEEE) Transaction Knowledge and Data Engineering (TKDE); 2013 Apr; 25(4):790–804. https://doi.org/10.1109/TKDE.2011.256
6. Morimoto Y. Mining frequent neighboring class sets in spatial databases. In the Proceedings of the Seventh Association for Computing Machinery (ACM) SIGKDD International Conference Knowledge Discovery and Data Mining (KDDM), San Francisco, California; 2001 Aug 26–29. p. 353–8. https://doi.org/10.1145/502512.502564
7. Manjula KR, Keshari AK, Pahlazani A. An approach to perform uncertainty analysis on a spatial dataset using clustering and distance based outlier detection technique. Indian Journal of Science and Technology. 2015 Dec; 8(35):1–7. https://doi.org/10.17485/ijst/2015/v8i35/71972
8. Huang Y, Xiong H, Shekar S. Mining confident co-location rules without a support threshold. In the Proceedings of the Association for Computing Machinery (ACM) Symposium in Applied Computing, Melbourne, Florida, USA; 2003. p. 497–501.
9. Yoo JS, Shekar S. A join less approach for mining spatial co-location patterns. Institute of Electrical and Electronics Engineers (IEEE) Transaction Knowledge and Data Engineering (TKDE). 2006 Dec; 18(10):1323–37. https://doi.org/10.1109/TKDE.2006.150
10. White T. Hadoop: the definitive guide. O'reilly; 2012 May.
11. Park H, Cha G, Chung C. Multi-way spatial joins using r-trees: methodology and performance evaluation. International Symposium on Advances in Spatial Databases; 1999 Jun 25. p. 229–50.
12. Sajana T, Sheela Rani CM, Narayana KV. A survey on clustering techniques for big data mining. Indian Journal of Science and Technology. 2016 Jan; 9(3):1–12. https://doi.org/10.17485/ijst/2016/v9i3/75971
13. Rarimala M, Lopez D. K-neighborhood structural similarity approach or spatial clustering. Indian Journal of Science and Technology. 2015 Sep; 8(23):1–11.
14. Agrawal R. Design and development of data classification methodology for uncertain data. Indian Journal of Science and Technology. 2016 Jan; 9(3):1–12. https://doi.org/10.17485/ijst/2016/v9i3/72262

15. Anuradha K, Sairam N. Spatio-temporal based approaches for human action recognition in static and dynamic background: a survey. Indian Journal of Science and Technology. 2016 Feb; 9(5):1–12. https://doi.org/10.17485/ijst/2016/v9i5/72065

16. Kumar S, Suseendran G. Incremental quality based reverse ranking for spatial data. Indian Journal of Science and Technology. 2016 Jan; 9(1):1–10.