Hypergraph Partitioning Algorithm

Chandani Santosh Jain
PG Student
K. K. W. I. E. E. R.,
Nashik

S. M. Kamalapur, PhD
Associate Professor
K. K. W. I. E. E. R.,
Nashik

ABSTRACT

Hypergraph is an abstraction of graph in which edges are non-empty subset of vertex set. Hypergraph has edges that connect set of two or more vertices. Hypergraph are more suitable to represent complex relational objects in many real-world problems. There is need to make the partitions of the hypergraph to analyze the whole hypergraph. Multilevel partitioning techniques are used to obtain subgraph. Sometimes they are inadequate to follow global objective function. Here hypergraph partitioning is making partitions of vertex set into the subset of vertices which are distributed smoothly to form subgraphs and having minimum intersections between this subgraphs. The hypergraph partitioning problem is used in many scientific computing, social network analysis than the similar graph problem.

General Terms

Data mining, Graph

Keywords

Hypergraph, Hypergraph partitioning , Dense subgraph

1. INTRODUCTION

Graphs are used to explain pairwise relationship shared by objects. However, graph are not sufficient for effective representation of set of complex relational objects. Using graph, information loss may occur that is why hypergraph are more suitable to represent complex relational objects in many real-world problems. Hypergraph is an abstraction of graph in which edges are non-empty subset of vertex set. Hypergraph has edges that connect set of two or more vertices. Hypergraph develop the concept of edge by granting more than two vertices to be connected by a hyperedge.[1] We can say that Hyperedge is a set of subsets of vertices. A hypergrph is define by two sets : a set of vertices and a set of hyperedges. [2]

Let's take an example of a collection of songs. We have only information that who sang the song. Lets observe the difficulty in organizing collection of songs in various categories. Consider vertices represents songs and connected by an edge only if there is at least one common singers of their corresponding songs. Singer set is denoted byS contains three singers. Song set is denoted by V contain seven songs. Table 1 illustrates the available information about songs and corresponding singers. A Singer set S = {s_1,s_2,s_3} and a song set V = { v_1,v_2,...v_7}

Using information provided in table 1, an undirected graph shown in figure 1 is constructed, in which two vertices are joined together by an edge if there is at least one common singers of their corresponding songs. Method discussed above is quite natural, graph representation may miss out details such as a singer joined singing three or more songs or not.

This kind of information loss is unwanted because the songs by the same person likely belong to the same category. This kind of information is useful in forming relevant clusters.

Table 1. Relationship Of Singer Set And Song Set

	s_1	s_2	s_3
v_1	0	0	1
v_2	0	0	1
v_3	0	1	0
v_4	0	1	1
v_5	0	1	0
v_6	1	0	0
v_7	1	0	1

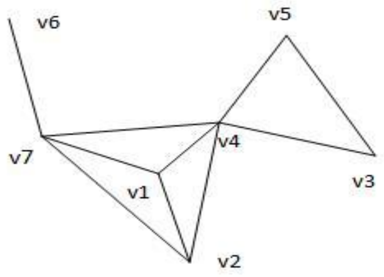

Fig.1 An undirected graph in which two songs are joined together by an edge if there is at least one singer in common.

Figure 2, shows hypergraph for the example of clustering few songs stated before. It is quite easy to develops a hypergraph with vertices representing the songs and the edges as singers. Each hyperedge contains all songs sung by its corresponding singers. We can also put positive weights on the edges to program our prior knowledge on singers' work if we have. For instance, for a person working on a broad range of fields, we may consign a comparatively small value to his subsequent edge. Now we can wholly represent the multifaceted relationships among objects by using hypergraphs. Simple graph is a hypergraph with each edge containing two vertices only. Hypergraphs can capture a relationship between a group of objects, whereas graphs can only capture binary relationship between objects.[3] Data is increasing day by day exponentially. Hence handling and resolving large data for practical applications is always needed. Different approaches are always required to reduce data.

Hypergraphs are better option to model complex unstructured relationships such as parallelization of complex and irregular applications from a range of domains including parallel scientific computing , sparse matrix reordering , socialnetwork analysis, clustering and recommendation, and database design. Other examples are Publishing data like co-authorship and co-citation, Collaborations in committee membership, movies etc, and chemical processes, social interactions and many more[13].

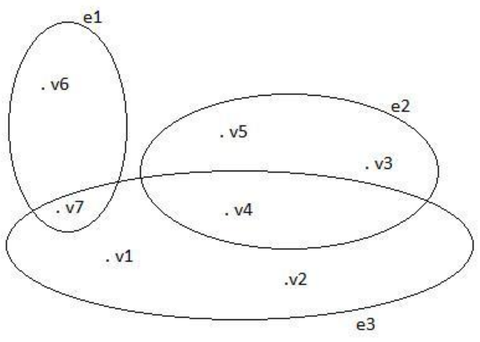

Fig.2 A hypergraph constructed for singer and song relationship

Hypergraph partitioning have applications in different discipline like, VLSI design, large database storage, task scheduling in multiprocessors, communities detection on web, image segmentation, query optimization, fixing cache locality in shared-memory systems. To reduce processing time, hypergraphs are used in representation of big data. While designing VLSI layout, it is essential to find optimum decomposition of hypergraph into k disjoint subsets.

Organization of this paper is as follows: Section I gives a brief introduction of proposed system. Section II focus on literature review, section III convey motivation behind proposed system. Section IV explains and contains block diagram and mathematical model of the proposed architecture. Section V elaborates data sets and results for the proposed method and section VI concludes the paper.

2. LITERATURE SURVEY

This section covers works done for hypergraph partitioning and summarizing their analysis. The existing hypergraph partition methods can be classified into three categories.

2.1 Iterative Partitioning

Conventional approaches for hypergraph partitioning are Kernighan-Lin algorithm (KL)[8] and Fiduccia Mattheyses (FM) [14] algorithms. KL algorithm tries to create cut between two balanced subgraph kept minimized. FM algorithm which partitions into unequal parts so that cut will be minimum. Both algorithms work in passes and lock vertices after moved. Actually, only move those vertices up to the maximum partial sum of gain. Difference between FM over KL is that is does not exchang pair of vertices, rather move only one vertex at each time. KL and FM are older algorithms but it is frequently found in use combined with other methods.

2.2 Spectral Partitioning

Spectral clustering techniques are also generalized to be used for hypergraphs. To represent relationship between vertices laplacian matrix[6] and modularity matrix[7] are constructed from hypergraph. Eigen vectors are formed of matrix representation of hypergraph. Zhou et al.[1] presented spectraltechnique applied to hypergraphs, Biological network and social network are analyzed using this developed algorithm. Rodriguez et al.[6] presented method in which hypergraphs are modeled using the generalization of the Laplacian matrix. Spectral Bisection is used when entire graph partitioning is to be performed. This method uses Adjacency matrix and diagonal Matrix. The disadvantage is high dimensional space required. The computational cost of partition increases rapidly as the size of a hypergraph.

2.3 Multi-level Partitioning

Computational burden and partition quality both are balanced using multilevel partitioning methods. First original graph is simplified, then initial partitioning is done. Final partition is achieved using refinement. Karypis et al. [8] presents multilevel hypergraph partitioning algorithm hMETIS [9] . METIS works directly on graph. Free download from website is made available for hMETIS.

Phase 1 : Coarsening phase : The graph G is first grained to a few hundred vertices. To find a good partition successive coarse graphs made. Lesser is memory required if coarsening done faster.

Phase 2 : Initial Partitioning : This is the easiest of the three phases. Random+FM, spectral, region growing, etc.algorithm can be used at this phase. Time required in this phase is very little.

Phase 3: Refinement: Also called as uncoarsening phase. Local refinement is done. Using partitioning done in phase 2, partition of larger graph is obtained. It reduces the cut and improves partitioning quality. Vertex swapping algorithms like KL, FM can be used. Efficiency of refinement algorithm is highly depends on coarsening.

hMETIS algorithm is used to divide large hypergraph into a predefined fixed number of subgraphs. Multiple iteration of algorithm improves the quality of partitioning. Lotfifar et al.[10] presented a multi-level sequential hypergraph partitioning algorithm. Feature Extraction Hypergraph Partitioning (FEHG) algorithm is used Coarsening, initial partitioning and uncoarsening are the three distinct phase used in this algorithm. Karypis et al. [11] Present a k-way partitioning algorithm. It also work under multilevel partitioning. Global optimize objective cannot be enforced on recursive bisection algorithm. The key problem is this algorithm trapped in local minima.

H Liu et al. [12] presented dense subgraph partition (DSP) of positive hypergraphs. The partitioning DSP naturally, correctly and quickly partitions positive hypergraph into dense subgraphs. DSP works in two layers of partitions. In first partition conditional core graphs are obtained. In second partition pseudo-disjoint subgraphs are obtained by using disjoint partition. Number of subgraph is determined automatically in DSP. List of dense subgraphs with decreasing densities is generated as a output of DSP. The result contains clusters and outliers also, inside that original hypergraph. DSP can be said as finds meaningful strong connections between vertices at multiple scales. DSP is inadequate to follow global objective function. Hence difficult to represent overall hypergraph picture. Besides, DSP is not able to partition a hypergraph into a defined number of subgraphs. In DSP, connection between different subgraphs are not necessarily weak. That means cost is not guaranteed to be minimum.

3. MOTIVATION

In DSP, connection between different subgraphs are not necessarily weak. That means cost is not guaranteed to be minimum.

In many hypergraph partitioning applications, input size is growing every year, to cope up with this high performance is necessary. Let's take the example of VLSI design. Total transistors needed in a classic VLSI design raised exponentially. Hence those algorithms working on scale of millions of vertices (as transistors), should be able to work with hundreds of millions in future. This scenario forces any partitioning algorithm must have almost linear worse case complexity. To improve the performance of existing dense subgraph partition of positive hypergraph, it is very important to find effective hypergraph partition technique. It finds clusters and outliers from local point of view. Global view in hypergraph partitioning at minimum cut cost is not addressed. Hence proposed work gives importance to global view at minimum cut cost while finding dense subgraphs.

4. PROPOSED ARCHITECTURE

This section includes details of proposed system i.e. mathematical model and block diagram of proposed system shown in figure 3.

4.1 Mathematical Model

Let S be a hypergraph partitioning system using which subgraphs of a hypergraph are constructed. The Proposed system S is defined as follows,

The mathematical model for the proposed system is as follows:

The proposed system S is defined as,

S= {V, E, O_1, O_2, O_3, O_4, O_5, F}

F is set of functions of the proposed system

F= {F_1, F_2, F_3, F_4, F_5}

V is the set of vertices in the dataset

V= {v_1, v_2, v_3, ….., v_n}

E is the set of edges in the dataset

E= {e_1, e_2, e_3, ….., e_m}

O_1 is the set of the preprocessed inputs

O_1= {D_1, D_2, D_3, …., D_n}

O_2 is the set of hyperedges forming hypergraph representing dataset

O_2= {h_1, h_2, h_3, …., h_n }

O_3 is the set of reward vector of vertices

O_3 = {r_1, r_2, r_3, ….., r_n }

O_4 is the set of histogram constructed for vertices

O_4= {g_1, g_2, g_3, …., g_n }

O_5 is the set of partitions generated

O_5= {p_1, p_2, p_3, …., p_j }

System functionality can be seen as first construct hypergraph and then partition hypergraph. Subgraphs are represented in the form of set of vertices.

Following rules used to define system functions and Table 2 elaborates functional dependency matrix

F_1 – It is a function used to perform data preprocessing

F_1 (V,E) → O_1

F_2 – It is a function used to perform hypergraph form input dataset

F_2 (O_1) → O_2

F_3 –It is a function used to calculate reward vector for vertices

F_3 (O_2) → O_3

F_4 –It is a function used to construct histogram for vertices

F_4 (O_3) → O_4

F_5 –It is a function used to construct partitions in the form of set of vertices

F_5 (O_4) → O_5

Table 2. Functional Dependency Matrix

	F1	F2	F3	F4	F5
F1	1	0	0	0	0
F2	1	1	0	0	0
F3	1	1	1	0	0
F4	1	1	1	1	0
F5	1	1	1	1	1

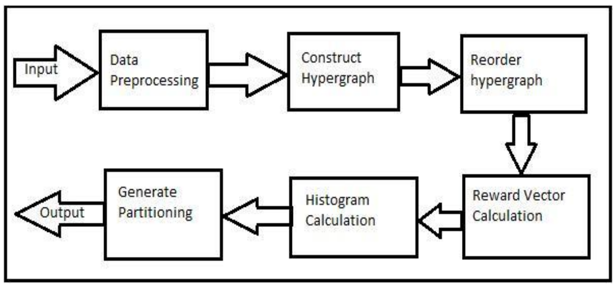

Fig.3 Block Diagram og Proposed system

The proposed hypergraph partitioning system consists of following steps

1. Read input dataset D
2. Construct hypergraph such that hyperedge for each vertex from input dataset by calculating set of adjacent vertices of each vertex.
3. Calculate reward $r_p(v)$ for each vertex v in partition p is equal to addition of weights W of all vertices in that edge.

$$r_e(v) \qquad w_v$$
$$v\, e$$

4. Reward vector is formed by knowing reward of each vertex.
5. Calculate integral histogram Y from reward vector for partition P contains vertices a1,a2,...a m such that

$$\{\, y_i \mid i \quad 1,..., n\}$$

where,

$$y_1 \quad r_p(a_1)$$

$$y_i \quad y_{i\ 1} \quad r_p\left(a_i\right) \text{ for i=2,...m}$$

6. Calculate α density of vi using histogram Y. Density αi of vi is initialized to reward vector of vi. For all vj in hi density calculated as

$$\frac{y_j}{j\ i\ \ 1}$$

7. Depending upon density partitions generated. If density is increasing then keep adding new vertex to older partition making a subgraph. Whenever density decreases from that vertex new subgraph is created. Subgraph will be created .

8. All partitions are generated in the form of subsets of vertices.

5. EXPERIMENTAL SETUP

Our Experimentation uses Intel processor and 6 GB RAM. The operating system is windows 8.1(64 bit) with C++. The proposed system works on the graph datasets, in the format of a text file and provides the output in the text file.

5.1 Dataset

In this section we discuss about dataset used. Proposed system uses Email-Enron dataset from Stanford Large Network Dataset Collection is used. Communication network provided in email-Enron consists of email communication. Vertices will be persons identified by email addresses. Communication between person i and j is denoted by an undirected edge between i and j.

5.2 Results and Analysis

Following table shows the results of proposed hypergraph partitioning technique on Email-enron dataset.

Table 3. Result Table

	Previous System	Proposed System
Time complexity	$O(n_e^3)$	$O(n_e p)$
Number of subgraphs obtained	1374	1178
Time required for partitions (s)	2.078	1.561

Where n_e=number of hyperedge, p= number of partitions

6. CONCLUSION

The existing graph partitioning methods can not follow global view due to multi level recursive coarsening. Also minimum cut cost is not guaranteed. Proposed work focus on global view. It takes care of minimum cost of cutting at the time of partitioning. Combination of global view and minimum cut cost will produce better quality partitions.

7. REFERENCES

[1] D. Zhou, J. Huang, and B. Scholkopf, “Learning with hypergraphs: Clustering, classification, and embedding,” in Proc. Adv. Neural Inf. Process. Syst., 2006, pp. 1601–1608.

[2] S. Kim, S. Nowozin, P. Kohli, and C. D. Yoo, “Higher-order correlation clustering for image segmentation,” inProc. Adv. Neural Inf. Process. Syst., 2011, pp. 1530–1538.

[3] R. Zass and A. Shashua, “Probabilistic graph and hypergraph matching,” inProc. IEEE Conf. Comput. Vis. Pattern Recog., 2008, pp. 1–8.

[4] B. Kernighan and S. Lin, “An efficient heuristic procedure for partitioning graphs,”Bell Syst. Tech. J., vol. 49, pp. 291– 307, 1970.

[5] C. Fiduccia and R. Mattheyses, “A linear-time heuristic for improving network partitions,” in Proc. 19th Conf. Des. Autom., 1982, pp. 175–181.

[6] J. Rodrıguez, “Laplacian eigenvalues and partition problems in hypergraphs,”Appl. Math. Letters, vol. 22, no. 6, pp. 916–921, 2009

[7] M. Newman, “Modularity and community structure in networks,”Proc. Nat. Acad. Sci., vol. 103, no. 23, pp. 8577– 8582, 2006.

[8] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar, “Multilevel hypergraph partitioning: Application in vlsi domain,” inProc. 34th Annu. Des. Autom. Conf., 1997, pp. 526–529.

[9] G. Karypis and V. Kumar, “hmetis: A hypergraph partitioning package, version 1.5. 3,” 1998.

[10]F. Lotfar and M. Johnson," A Multi Level Hypergraph Partitioning Algorithm using Rough Set Clustering", School of Engineering and Computing Sciences, Durham University, United Kingdom, 2015

[11] G. Karypis and V. Kumar," Multilevel k-way Hypergraph Partitioning", Department of Computer Science and Engineering, Army HPC Research Center, University of Minnesota, Minneapolis,2000, Vol. 11, No. 3, pp. 285-300

[12] H. Liu, L. J. Latecki, S. Yan ," Dense Subgraph Partition of Positive Hypergraphs", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 37, No. 3, March 2015

[13] C.Jain , S. Kamalapur ,"Hypergraph Partitioning Algorithm", cPGCON 2016