Beginner's Practical: Hypergraph Conductance-based Clustering

MARIIA MESHCHANINOVA, Heidelberg University, Germany

Hypergraph clustering is a problem of partitioning the nodes of a hypergraph into "true clusters" which are densely connected internally and sparsely connected externally. Conductance is a metric measuring the quality of a cut which is defined as the ratio of the weight of the cut to the minimal volume of one side of the cut. In this project, we implement a hypergraph clustering algorithm that heuristically tries to minimize the maximum value of the conductance over all cuts between clusters to find a high-quality hypergraph clustering. The algorithm is based on the existing shared-memory parallel algorithm for hypergraph partitioning `Mt-KaHyPar`.

1 INTRODUCTION

Hypergraph clustering is a problem of partitioning the nodes of a hypergraph into "true clusters" which are densely connected internally and sparsely connected externally. It is useful in many applications, such as semi-supervised learning [12], gene expression analysis [11] and image segmentation [5]. Conductance metric, which is defined as the ratio of the weight of a given cut to the minimal volume of one side of the cut, is widely used in graph clustering [2, 7] but to the best of our knowledge, there are no partitioners directly optimizing conductance-based metric of a hypergraph, while conductnace has been applied as a metric for evaluating the quality of a hypergraph clustering [1]. Therefore in this project, we implement a heuristic hypergraph clustering algorithm with the objective of minimizing the maximum value of the conductance over all cuts between the clusters while partitioning the hypargraph in at most k clusters.
Our algorithm is based on the existing shared-memory parallel hypergraph k-way partitioner `Mt-KaHyPar` and uses the same multilevel approach as the original algorithm. First, the hypergraph is coarsened in stages by detecting and merging local node clusters, then the coarsest hypergraph is partitioned into k clusters, and finally the initial partitioning is refined by uncoarsening the hypergraph nodes and applying a local search algorithm to improve the given objective. We implement two objective functions for the local search algorithm - *conductance_local* and *conductance_global* - and compare their performance on a benchmark set `ibm01...18` against a state of the art hypergraph partitioner `HyperModularity`.
In following, we define the used concepts in Section 2, then we describe related work on hypergraph partitioning and provide an overview of the existing conductance-based clustering algorithms in Section 3. In Section 4, we describe the implementation of the new objectives for optimizing the conductance and their integration in `Mt-KaHyPar`. Afterwards, we present the experimental results of our implementation in Section 5. Finally, we conclude the paper and discuss future work in Section 6.

2 PRELIMINARIES

In this section, we give an overview of the used concepts and notations and define conductance of a hypergraph.

Hypergraph is a generalization of a graph, where a hyperedge connects one, two or more nodes. Formally, an edge-weighted hypergraph $H = (V, E, \omega)$ consists of a non-empty set of nodes V, a set of hyperedges (also called *nets* or simply *edges*) $E \subseteq 2^V$ and a weight function for hyperedges $\omega : E \rightarrow \mathbb{N}$. In case of unweighted hyperedges, we set $\omega(e) = 1$ for all $e \in E$.

Author's address: Mariia Meshchaninova, mariia.meshchaninova@stud.uni-heidelberg.de,InformatikB.Sc.100%,4740180, Heidelberg University, Im Neuenheimer Feld 205, Heidelberg, Baden-Württemberg, Germany, 69120.

k-way clustering of a hypergraph $H = (V, E, \omega)$ is a partitioning of the set of nodes V into k disjoint subsets $V_1, \ldots, V_k$ called *clusters*. Empty clusters are allowed, so k is only an upper bound on the final number of clusters. Our algorithm is allowed to eventially move all the nodes of a cluster to other clusters if it would be beneficial for the objective function.

Cut ∂S of a hypergraph $H = (V, E, \omega)$ is partitioning of the set of nodes V into two disjoint non-empty subsets S and $V \backslash S$. A hyperedge $e \in E$ is called a **cutting edge** of the cut ∂S if it has at least one pin in S and at least one pin in $V \backslash S$. The weight of the cut is defined as the sum of the weights of the cutting edges of the cut ∂S:

$$\omega(\partial S) = \sum_{e \in \partial S} \omega(e)$$

Therefore in case of unweighted hyperedges, the weight of the cut is equal to the number of cutting edges of the cut.

Pin of a hyperedge $e \in E$ is a node $v \in V$ such that $v \in e$. A hyperedge with only one pin is called a **sinle-pin net**. In the implementation, we use number of pins $PinNum_i(e)$ of a hyperedge $e \in E$ in a cluster V_i to decide whether to move a node v from one cluster V_i to another cluster V_j.

Weighted degree of a node $v \in V$ is defined the same way in hypergraphs as in graphs. It is the sum of the weights of all hyperedges $e \in E$ that contain the node v:

$$\deg_\omega(v) = \sum_{e \in E : v \in E} \omega(e)$$

As during the coarsening phase of the algorithm we merge nodes, we also use **original weighted degree** of a node $v' \in V'$ of the coarsened hypergraph $\text{origDeg}_\omega(v')$ which is defined as the sum of the weighted degrees of all nodes of the initial hypergraph $v \in V$ that were merged into v' during the coarsening phase of the algorithm.

Volume of a hypergraph $H = (V, E, c, \omega)$ is defined as the sum of the weighted degrees of all nodes $v \in V$. Analogously, we define the **volume of a cluster** V_i as the sum of the weighted degrees of all nodes $v \in V_i$:

$$\text{vol}(H) = \sum_{v \in V} \deg_\omega(v) \qquad \text{vol}(V_i) = \sum_{v \in V_i} \deg_\omega(v)$$

And for the same reason as for the original weighted degree, we also define the **original volume** of a hypargraph and a cluster:

$$\text{origVol}(H) = \sum_{v \in V} \text{origDeg}_\omega(v) \qquad \text{origVol}(V_i) = \sum_{v \in V_i} \text{origDeg}_\omega(v)$$

This way, the original volume of a cluster or of the whole hypergraph in the coarsened hypergraph is equal to the corresponding volume in the initial - *original* - hypergraph.

Conductance of a cut ∂S of a hypergraph $H = (V, E, \omega)$ is defined as the ratio of the weight of the cut to the minimum volume of one side of the cut. The maximal conductance of a cut In a hypergraph is referred to as the **conductance of the hypergraph**:

$$\varphi(S) = \frac{\omega(\partial S)}{\min\{vol(S), vol(V \backslash S)\}} \qquad \varphi(V) = \max_{\emptyset \subsetneq S \subsetneq V} \varphi(S)$$

To use conductance as a quality metric for a hypergraph clustering with possibly more than two clusters, we define the **conductance of a hypergraph clustering** $V_1, \ldots, V_k$ as the maximum conductance of all cuts between the clusters. Conveniently, this definition can be simplified to the maximal conductance of a cut ∂V_i over all clusters V_i:

$$\varphi(V_1, \ldots, V_k) := \max_{\emptyset \neq I \subsetneq \{1,\ldots,k\}} \{\varphi(\cup_{i \in I} V_i)\} = \max_{i=1\ldots k} \varphi(V_i)$$

A proof of this nice statement can be found in Appendix A.

3 RELATED WORK

There are several state of the art algorithms for graph clustering that aim to minimize the resulting conductance. Some of them are heuristic like the diffusion based `NIBBLE` [10] and `hk-relax` [6]. Other, such as degree ratio-based `PC_de` [8] have a theoretical guaranatee of the resulting conductance.

As for the hypergraph-clustering, to the best of our knowledge, there are no existing algorithms that directly optimize a conductance-based objective. Instead, state of the art hypergraph clustering algorithms, e.g. `HyperModularity` which we use for comparison with our algorithm, deliver high quality hypergraph clustering by optimizing the modularity of the hypergraph [3, 9].

The algorithm `Mt-KaHyPar` that we use as a base for our implementation also optimizes the modularity of the hypergraph in its coarsening stage. But in general, it could be seen as a framework for hypergraph clustering with a user-implemented objective, which is optimized during the initial partitioning of the coarsest hypergraph and later in the uncoarsening stage consisting of label propagation and FR-refinement [4].

4 IMPLEMENTATION

5 EXPERIMENTAL RESULTS

6 CONCLUSION

A APPENDIX

Theorem A.1. *Concuctance of a k-way partition $V_1, \dots, V_k$ is the maximal conductance of a cut V_i for $i = 1, \dots k$.*

Proof. Per definition of conductance of a k-way partition, there exists a subset $\emptyset \subsetneq I \subsetneq \{1, \dots, k\}$ such that $\varphi(V_1, \dots V_k) = \varphi(\cup_{i\in I} V_i)$. Without loss of generality, we can assume that the volume of the union of the clusters V_i for $i \in I$ is less than or equal to the volume if the union of all the other clusters V_i for $i \notin I$. Then we can write:

$$\begin{aligned}
\varphi(V_1, \dots, V_k) &= \frac{\omega(\partial(\cup_{i\in I} V_i))}{\min(\mathrm{vol}(\cup_{i\in I} V_i), \mathrm{vol}(\cup_{i\notin I} V_i))} = \frac{\sum\{\omega(e) : e - \text{a cutting edge of } \partial(\cup_{i\in I} V_i)\}}{\mathrm{vol}(\cup_{i\in I} V_i)} \\
&\leq \frac{\sum_{i\in I} \omega(\partial V_i)}{\sum_{i\in I} \mathrm{vol}(V_i)} = \frac{\sum_{i\in I} \mathrm{vol}(V_i) \cdot \frac{\omega(\partial V_i)}{\mathrm{vol}(V_i)}}{\sum_{i\in I} \mathrm{vol}(V_i)} \leq \max_{i\in I} \frac{\omega(\partial V_i)}{\mathrm{vol}(V_i)} \\
&\leq \max_{i\in I} \frac{\omega(\partial V_i)}{\min(\mathrm{vol}(V_i), \mathrm{vol}(V \setminus V_i))} = \max_{i\in I} \varphi(V_i) \\
&\leq \varphi(V_1, \dots, V_k)
\end{aligned}$$

Thus, all inequalities are equalities, which means that the conductance of the k-way partition is equal to the maximal conductance of a cut V_i for $i = 1, \dots k$.

It is interesting to note that with this we have also proved that the conductance of a hypergraph partition is equal to the maximum ratio of the weight of a cut V_i to the volume of the cluster V_i over all the clusters $V_1, \dots V_k$. This property, however, was not used in the implementation due to late discovery. □

B GAIN FOR LOCAL SEARCH: NAIVE?

Information needed:

- the sum of all weights of the nets: $vol(V) = const$;
- for each node:

  - incident nets;
  - weighted degree;
- for each net e:
  - number of pins in e: $|e|$;
  - number of pins of e which are currently in cluster V_i for each cluster index i: $e.\#pins(V_i)$;
- for each cluster index i:
  - current volume: $vol(V_i)$;
  - weights sum of cutting edges of the current V_i: $V_i.out\_weight$.

Time needed: $O(k \cdot deg(v) + k \cdot \log k)$...

Calculating the gain for moving node v from V_i to V_j:

(1) (temporary) adjust volumes of V_i and V_j: $\Theta(1)$

$$vol(V_i)' \leftarrow vol(V_i) - \deg_\omega(v)$$

$$vol(V_j)' \leftarrow vol(V_j) + \deg_\omega(v)$$

(2) (temporary) adjust $Vi.out\_weight$ and analogously for $Vj.out\_weight$ (but with reversed signs of $\omega(e)$ and $0, |e| - 1$ in conditions in ifs): $\Theta(\deg(v))$
For each incident to v net $e \in E$:
if $1 = e.\#pins(V_i) < |e|$:
e is a cutting net for the cut V_i, v - its last pin in V_i
$\implies$ e won't be a cutting edge after removal of v:

$$Vi.out\_weight' \leftarrow V_i.out\_weight - \omega(e)$$

else if $1 < e.\#pins(V_i) = |e|$:
e was not a cutting net for the cut V_i, but it will be after the removal of v:

$$Vi.out\_weight' \leftarrow V_i.out\_weight + \omega(e)$$

(3) calculate new conductances of cuts V_i and (analogously) V_j: $\Theta(1)$

$$\varphi(V_i) = \frac{V_i.out\_weight'}{\min\{vol(V_i)', vol(V) - vol(V_i)'\}}$$

The question is, what should we take for gain from moving v:

(1) decrease in maximal conductance between the cuts V_i and V_j: *easy*;
(2) decrease in the overall maximal conductance: a PQ with k current conductances $\varphi(V_1), \ldots, \varphi(V_k)$ - additional $O(k \log k)$ for finding the best V_j.

$\implies$ maybe the best according to criterion 1 from the best according to criterion 2... (***I believe, I should try all these variants and possibly some other...***)

A PQ with k current conductances $\varphi(V_1), \ldots, \varphi(V_k)$ is needed (?) to calculate the contribution of a net to the overall conductance of the partition (?):

C CONTRIBUTION OF A NET TO THE OVERALL CONDUCTANCE

Objective function: conductance of the partition (also naive?):

- each edge e from the most expensive conduction-wise cut V_i contributes by

$$\frac{\omega(e)}{\max\{vol(V_i), vol(V) - vol(V_i)\}}$$

- all other edges contribute by 0 (to ensure that the sum of all contributions is the current conductance...)

REFERENCES

[1] Ali Aghdaei, Zhiqiang Zhao, and Zhuo Feng. Hypersf: Spectral hypergraph coarsening via flow-based local clustering. In *2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, pages 1–9. IEEE, 2021.

[2] Reid Andersen, Fan Chung, and Kevin Lang. Local graph partitioning using pagerank vectors. In *2006 47th annual IEEE symposium on foundations of computer science (FOCS'06)*, pages 475–486. IEEE, 2006.

[3] Philip S Chodrow, Nate Veldt, and Austin R Benson. Generative hypergraph clustering: From blockmodels to modularity. *Science Advances*, 7(28):eabh1303, 2021.

[4] Lars Gottesbüren, Tobias Heuer, Peter Sanders, and Sebastian Schlag. Scalable shared-memory hypergraph partitioning. *CoRR*, abs/2010.10272, 2020. URL https://arxiv.org/abs/2010.10272.

[5] Sungwoong Kim, Sebastian Nowozin, Pushmeet Kohli, and Chang Yoo. Higher-order correlation clustering for image segmentation. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper_files/paper/2011/file/98d6f58ab0dafbb86b083a001561bb34-Paper.pdf.

[6] Kyle Kloster and David F. Gleich. Heat kernel based community detection. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, page 1386–1395, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450329569. doi: 10.1145/2623330.2623706. URL https://doi.org/10.1145/2623330.2623706.

[7] Longlong Lin, Ronghua Li, and Tao Jia. Scalable and effective conductance-based graph clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4471–4478, 2023.

[8] Longlong Lin, Ronghua Li, and Tao Jia. Scalable and effective conductance-based graph clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4471–4478, 2023.

[9] Veronica Poda and Catherine Matias. Comparison of modularity-based approaches for nodes clustering in hypergraphs. *Peer Community Journal*, 4, 2024.

[10] Daniel A Spielman and Shang-Hua Teng. A local clustering algorithm for massive graphs and its application to nearly linear time graph partitioning. *SIAM Journal on computing*, 42(1):1–26, 2013.

[11] Ze Tian, TaeHyun Hwang, and Rui Kuang. A hypergraph-based learning algorithm for classifying gene expression and arraycgh data with prior knowledge. *Bioinformatics*, 25(21):2831–2838, 07 2009. ISSN 1367-4803. doi: 10.1093/bioinformatics/btp467. URL https://doi.org/10.1093/bioinformatics/btp467.

[12] Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006. URL https://proceedings.neurips.cc/paper_files/paper/2006/file/dff8e9c2ac33381546d96deea9922999-Paper.pdf.