DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

Spectral Methods to Find Small Expansion Sets on Hypergraphs

Franz Rieger

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

Spectral Methods to Find Small Expansion Sets on Hypergraphs

Spektrale Methoden zum Finden kleiner Expansionsmengen auf Hypergraphen

Author:	Franz Rieger
Supervisor:	Prof. Susanne Albers
Advisor:	Dr. T.-H. Hubert Chan
Submission Date:	15. January 2019

I confirm that this bachelor's thesis in informatics is my own work and I have documented all sources and material used.

Munich, 15. January 2019 Franz Rieger

Acknowledgments

This thesis was written under the supervision of Dr. T.-H. Hubert Chan at the University of Hong Kong. TODO: ideas from him

Abstract

The problem of finding a small Edge Expansion on a graph can also be defined on hypergraphs. In this thesis approximation algorithms for obtaining sets with a small Edge Expansion are discussed and implemented.

Contents

Acknowledgments **iii**

Abstract **iv**

1 Introduction **1**

2 Notation **2**

3 Algorithms **4**
3.1 Brute force 4
3.2 Orthonormal vectors 4

4 Random Hypergraphs **7**
4.1 random edges with discard if not connected 7
4.2 connected edges with discard if not connected or not regular 7

5 Implementation **8**
5.1 Technologies 8
5.2 Code 8

6 Evaluation **9**

7 Applications **10**

8 Resume and Further Work **11**

List of Figures **12**

List of Tables **13**

1 Introduction

TODO: how to work with notation of next chapter here: minium TODO: example graphs (also example dataset?) TODO: Mincut, Sparsest Cut, Edge expansion

For normal graphs Np-Hard [**kaibel2004expansion**]

2 Notation

The notation used in this thesis is orientated on [**ChanLTZ16**].

A weighted, undirected hypergraph $H = (V, E, w)$ consists of a set of n vertices $V = \{v_1, \ldots, v_n\}$ and a set of m (hyper-)edges $E = \{e_1, \ldots, e_m | \forall i \in [i] : e_i \subseteq V \wedge e_i \neq \emptyset\}$ where every edge e is a non-empty subset of V and has a positive weight $w_e := w(e)$, defined by the weight function $w : E \rightarrow \mathbb{R}_+$.

The weight w_v of a vertex v is defined by summing up the weights of its edges: $w_v = \sum_{e \in E : v \in e} w_e$. Accordingly, a subset $S \subseteq V$ of vertices has weight $w_S := \sum_{v \in S}$ and a subset $F \subseteq E$ of edges has weight $w_F = \sum_{e \in F} w_e$. The set of edges which are cut by S is defined as $\partial S := \{e \in E : e \cap S \neq \emptyset \wedge e \cap V \setminus S \neq \emptyset\}$, which contains all the edges, which have at least one vertex in S and at least one vertex in $V \setminus S$. The edge expansion of a non-empty set of vertices $S \subseteq V$ is defined by

$$\Phi(S) := \frac{w(\partial S)}{w(S)}. \tag{2.1}$$

Observe that $\forall \emptyset \neq S \subset V : 0 \leq \Phi(S) \leq 1$. The first inequality holds because the edge-weights are positive. The second inequality holds because $W(S) \geq W(\partial S)$, as $W(S)$ takes at least every edge (and therefore the corresponding weight), which is also considered by $W(\partial S)$, into account.

With this, the expansion of a graph H is defined as

$$\Phi(H) := \min_{\emptyset \subsetneq S \subsetneq V} \max\{\Phi(S), \Phi(V \setminus S)\}. \tag{2.2}$$

Here again, $0 \leq \Phi(H) \leq 1$ holds. For not connected graphs $\Phi(H) = 0$, which can be verified by observing a S which only contains vertices of one connection component. Therefore, only connected graphs shall be of interest here. Observe that for a graph H, which is obtained by connecting two connection components with edge with small weight, $\Phi(H)$ takes a small value. For a fully connected graph with equal edge-weights, ∂S (and therefore $\Phi(S)$) will be big for every $S \subsetneq V$.

The weight matrix can be denoted as

$$W = \begin{pmatrix} w_{v_1} & 0 & 0 & \dots & 0 \\ 0 & w_{v_2} & 0 & \dots & 0 \\ 0 & 0 & w_{v_3} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & w_{v_n} \end{pmatrix} \in \mathbb{R}_{0+}^{n \times n}$$

The discrepancy ratio of a graph, given a non-zero vector $f \in \mathbb{R}^V$ is defined as

$$D_w(f) := \frac{\sum_{e \in E} w_e \max_{u,v \in e}(f_u - f_v)^2}{\sum_{u \in V} w_u f_u^2}$$

.

In the weighted space, in which the discrepancy ratio is defined like above, for two vectors $f, g \in \mathbb{R}^V$ the inner product is defined as $\langle f, g \rangle_w := f^T W g$. Accordingly, the norm is $||f||_w = \sqrt{\langle f, f \rangle_w}$. If $\langle f, g \rangle_w = 0$, f and g are said to be orthonormal in the weighted space.

3 Algorithms

In the following chapter different approaches for generating small expansion sets S will be discussed. TODO: why phi (S) and not phi(H)?

3.1 Brute force

One obvoius approach is to brute-force the problem:

Algorithm 1 Brute-force

$best\_S := null$
$lowest\_expansion :=$ inf
for $\emptyset \neq S \subsetneq V$ **do**
  $expansion := \Phi(S)$
  if $expansion < lowest\_expansion$ **then**
    $lowest\_expansion := expansion$
    $best\_S := S$
return $best\_S$

Correctness: This as this algorithm iterates over all $\emptyset \neq S \subsetneq V$, it computes $\arg\min_{\emptyset \subsetneq S \subsetneq V} \Phi(S)$.

TODO: what else to prove?

Complexity: There are $2^{|V|} - 2 = 2^n - 2 \in O(2^n)$ combinations for $\emptyset \neq S \subsetneq V$, namely all the $2^{|V|}$ subsets of V excluding the empty set $\emptyset$ and V itself. Therefore, this algorithm is of exponential time complexity in n and is therefore not efficient for larger graphs.

TODO: refine brute-force to only $\phi(S)$ not $\phi(H)$ possibly with $a < |S| < b$

3.2 Orthonormal vectors

As described in **[ChanLTZ16]**, the following algorithm can be used:

Fact 3.2.1 *Theorem 6.6 in [***ChanLTZ16***] Given an a hypergraph $H = (V, E, w)$ and k vectors $f_1, f_2, \dots, f_k$ which are orthonormal in the weighted space with $\max_{s \in [k]} D_w(f_s) \leq \xi$, the*

Algorithm 2 Small Set Expansion (according to Algorithm 1 in [**ChanLTZ16**])

function SmallSetExpansion($G := (V, E, w), f_1, \dots, f_k$)
  assert $\xi == \max_{s\in[k]}\{D_w(f_s)\}$
  assert $\forall f_i, f_j \in \{f_1, \dots, f_k\} \subset \mathbb{R}^n, i \neq j : f_i$ and f_j orthonormal in weighted space
  for $i \in V$ **do**
    for $s \in [k]$ **do**
      $u_i(s) := f_s(i)$
  for $i \in V$ **do**
    $\tilde{u}_i := \frac{u_i}{||u_i||}$
  $\hat{S} :=$ OrthogonalSeparator($\{\tilde{u}_i\}_{i\in V}, \beta = \frac{99}{100}, \tau = k$)
  for $i \in S$ **do**
    if $\tilde{u}_i \in \hat{S}$ **then**
      $X_i := ||u_i||^2$
    else
      $X_i := 0$
  $X :=$ sort $list(\{X_i\}_{i\in V})$
  $V := [i]_{\text{in order of X}}$
  $S := \arg\min_{\{P:O \text{ is prefix of } V\}} \phi(O)$
  return S

following holds. algorithm 2 constructs a random set $S \subsetneq V$ in polynomial time such that with $\Omega(1)$ probability, $|S| \leq \frac{24|V|}{k}$ and

$$\phi(S) \leq C \min\{\sqrt{r \log k}, k \log k \log\log k \sqrt{\log r}\} \cdot \sqrt{\xi},$$

where C is an absolute constant and $r := \max_{e\in E} |e|$.

Algorithm 3 Orthogonal Separator (combination of Lemma 18 and algorithm Theorem 10 in [**LouisM14**] (also Fact 6.7 in [**ChanLTZ16**]))

function ORTHOGONALSEPARATOR($\{\tilde{u}_i\}_{i \in V}, \beta = \frac{99}{100}, \tau = k$)
  $l := \lceil \frac{\log_2 k}{1 - \log_2 k} \rceil$
  $g \sim \mathcal{N}(0, I_n)$ where each component g_i is mutually independent and sampled from $\mathcal{N}(0,1)$
  $w :=$SAMPLEASSIGNMENTS(l, V, β)
  for $i \in V$ **do**
    $W(u) := w_1(u) w_2(u) \cdots w_j(u)$
  if $n \geq 2^l$ **then**
    $word := random(\{0,1\}^l)$ uniform
  else
    $words := set(w(i) : i \in V)$ no multiset
    $words \cup = \{w_1, \ldots, w_{|V| - |words|} \in \{0,1\}^l\}$ random choice
    $word := random(words)$ uniform
  $r := uniform(0,1)$
  $S := \{i \in V : ||i||^2 \geq r \wedge W(u) = word\}$
  return S

Algorithm 4 Sample Assignments (proof of Lemma 18 in [**LouisM14**])

function SAMPLEASSIGNMENTS(l, V, β)
  $\lambda := \frac{1}{\sqrt{\beta}}$
  for $j = 1, 2, \ldots, l$ **do**
    for $i \in V$ **do**
      $t_i := \langle g, \tilde{u}_i \rangle$
      $poisson\_count_i := N(t_i, \lambda)$ where N is a poisson process on $\mathbb{R}$
      if $poisson\_count_i \mod 2 == 0$ **then**
        $w_j(i) := 1$
      else
        $w_j(i) := 0$
  return w

4 Random Hypergraphs

TODO: Discuss different approaches of generating, their limitations

TODO: Analyze Φ for different random- classes? (and explain?)

4.1 random edges with discard if not connected

4.2 connected edges with discard if not connected or not regular

5 Implementation

5.1 Technologies

Python with Nump, Scipy as optimizer

5.2 Code

Own hypergraph implementation

TODO: how to reference code? Github? TODO: what all to explain

6 Evaluation

TODO: find constants by analyzing quality? Analyze runtime of code?

7 Applications

TODO: groups in social network discussions (how to cite discussion?) Learning?

8 Resume and Further Work

List of Figures

List of Tables