Resumen

Aquí va el resumen...

ABSTRACT

Aquí inicia el abstract...

AGRADECIMIENTOS

Aquí van los agradecimientos...

Contents

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

El capítulo 1 inicia aquí...

1.1 Presentation

1.2 Objectives

1.2.1 General objective

1.2.2 Particular objective

1.3 Justification

1.4 Limitations and delimitations of the project

1.5 Research Problem

1.6 Hypothesis

1.7 Project organization

CHAPTER 2

THEORY AND CONCEPTUAL FRAMEWORK

2.1 Preliminaries

Description of concepts

Definition 2.1.1. *orthogonal complement*

Theorem 2.1.1. *For every $n \times n$ symmetric real matrix, the eigenvalues are real and the eigenvectors can be chosen real and orthonormal.*

Theorem 2.1.2 (Courant-Fisher Formula)**.** *Let A be an $n \times n$ real symmetric matrix with eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ and corresponding eigenvectors $v_1, v_2, \ldots, v_n$. Then*

$$\lambda_1 = \min_{\|x\|=1} x^T A x = \min_{x \neq 0} \frac{x^T A x}{x^T x},$$

$$\lambda_2 = \min_{\substack{\|x\|=1 \\ x \perp v_1}} x^T A x = \min_{x \neq 0} \frac{x^T A x}{x^T x},$$

$$\lambda_n = \lambda_{max} = \max_{\substack{\|x\|=1 \\ x \perp v_1}} x^T A x = \max_{\substack{x \neq 0 \\ x \perp v_1}} \frac{x^T A x}{x^T x}.$$

In general, for $1 \leq k \leq n$, let S_k denote the span of $v_1, v_2, \ldots, v_k$ (with $S_0 = \{0\}$). Then

$$\lambda_k = \min_{\substack{\|x\|=1 \\ x \in S_{k-1}^{\perp}}} x^T A x = \min_{\substack{x \neq 0 \\ x \in S_{k-1}^{\perp}}} \frac{x^T A x}{x^T x}.$$

2.2 Graphs and Laplacian Matrices

For the rest of the chapter, let $G = (V,E)$ be a graph, where $V = \{v_1, v_2, ..., v_n\}$ is the non-empty set of nodes (or vertices) and E is the set of edges, composed by pairs of the form (v_i, v_j), where $v_i, v_j \in V$.

It is assumed that all graphs are undirected, meaning that if $(v_i, v_j) \in E$, then $(v_j, v_i) \in E$, for every $v_i, v_j \in V$. For that reason, the edge (v_i, v_j) will be represented as the unordered set $\{v_i, v_j\}$.

A convenient way to represent a graph is through an *adjacency matrix* $A \in \mathbb{R}^{|V|\times|V|}$. Giving a specific order to the graph nodes, one can represent the edges as binary entries in this matrix:

$$A[v_i, v_j] = \begin{cases} 1 & \text{if } \{v_i, v_j\} \in E \\ 0 & \text{otherwise} \end{cases}$$

Let $\mathcal{N}(v_i)$ denote the neighborhood of node v_i, i.e., the set of the adjacent nodes to it. The quantity that represents the number of nodes in $\mathcal{N}(v_i)$ is called the *degree of the vertex* v_i. This is one of the most obvious and informative feature for the structure of the graph and is denoted by

$$d_i = \sum_{j=1}^{n} A[v_j, v_i].$$

Finally, we can summarize that graph's information in the *degree matrix* D which is defined as the diagonal matrix with the degrees $d_1, d_2, ..., d_n$ on the diagonal.

2.2.1 The Unnormalized Laplacian

The unnormalized graph *Laplacian matrix* L is defined as

$$L = D - A$$

Proposition 2.2.1 (Some properties of L). *The matrix L, as defined above, satisfies the following properties:*

1. *For every vector $x = (x_1, x_2, ..., x_n) \in \mathbb{R}$ we have*

$$x^T L x = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \left(x_i - x_j\right)^2$$

2. *L is symmetric and positive semi-definite*

3. *L has n non-negative, real-valued, eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$*

4. *The smallest eigenvalue of* L *is* 0*, the corresponding eigenvector is the constant one vector* $\mathbb{1}$.

2.2.2 Normalized Laplacians

The *symmetric normalized Laplacian* matrix L_{sym} is defined as

$$L_{sym} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$$

while the *random walk Laplacian* is defined as

$$L_{rw} = D^{-1} L$$

2.2.3 The graph partitioning problem

In order to introduce the graph partitioning problem in its different settings, the mathematical definition of the concepts involved are presented below.

Definition 2.2.1. *Given a graph* $G = (V, E)$*, a partition of* G *is a collection of* k *subsets* $P_1, P_2, ..., P_k \subset V$ *such that:*

1. $P_i \cap P_j = \emptyset$ *for* $i \neq j$*, where* $i, j \in \{1, 2, ..., k\}$
2. $\cup_{i=1}^{k} P_k = V$

A partition of a graph can be seen as simply removing edges from the original graph in such way the obtained partitions are subgraphs. There are many ways a graph can be partitioned into subgraphs and the way it gets done depends completely on the application of interest. However, independently of the problem to solve, the objective relies on minimizing the connections between the partitions in the original graph. The following concepts provides a useful notation to turn the problem into an optimization one.

For a collection $S \subset V$ of vertices, we define the *edge boundary* $\partial(S)$ to consist of all edges in E with exactly one endpoint in S, that is,

$$\partial(S) := \{\{u, v\} \in E \mid u \notin S \text{ and } v \in S\}$$

Now the problem turns into finding a partition $P_1, P_2, ..., P_k$ such that minimizes the *cut value* of the partition, usually called just *cut*, which is defined as

$$\textsc{Cut}(P_1, P_2, ..., P_k) := \frac{1}{2} \sum_{i=1}^{k} \partial(P_i) \tag{2.1}$$

The notion of cut allows to measure the quality of any partition, nevertheless solving the min cut problem ...

For a collection of vertices $S \subset V$ consider the following quantities related to the edges of

$$\textsc{Vol}(S) := \sum_{v_i \in S} d_i$$

we want to agroupe by similarity so its natural to Cut value of that partition
solve the mincut problem
The next consider two different ways of measuring the size of the partitions

$$\begin{aligned}\textsc{RatioCut}(P_1, P_2, ..., P_k) &:= \frac{1}{2}\sum_{i=1}^{k}\frac{\partial(P_i)}{|P_i|}\\ &= \sum_{i=1}^{k}\frac{\textsc{Cut}(P_i, \overline{P_i})}{|P_i|}\end{aligned}$$

$$\begin{aligned}\textsc{NormCut}(P_1, P_2, ..., P_k) &:= \frac{1}{2}\sum_{i=1}^{k}\frac{\partial(P_i)}{\textsc{Vol}(P_i)}\\ &= \sum_{i=1}^{k}\frac{\textsc{Cut}(P_i, \overline{P_i})}{\textsc{Vol}(P_i)}\end{aligned}$$

The spectral method

1. Let v denote the second smallest eigenvector of $\mathscr{L}$. Sort the vertices i of G in increasing order of v_i. Let the resulting ordering be $v_1 \leq v_2 \leq \cdots v_n$
2. For each i, consider the cut induced by $\{1, 2, ..., i\}$ and its complement. Calculate its conductance.
3. Among these $n-1$ cuts, choose the one with minimum conductance.

Cheeger's inequality

For a graph $G = (V, E)$ the *conductance* or *Cheeger ratio* of a set $S \subset V$ is the ratio of the fraction of edges in the cut $(S, \overline{S})$ o the volume of S,

$$\phi(S) = \frac{E(S, \overline{S})}{\textsc{Vol}(S)}$$

The *conductance* or *Cheeger constant* of a graph G is denoted by

$$\phi(G) = \min_{S} \phi(S)$$

Theorem 2.2.1. *In a graph G, the Cheeger constant $\phi(G)$ and the spectral gap λ_G are related as follows:*

$$2\phi(G) \geq \lambda_G \geq \frac{\alpha_G^2}{2} \geq \frac{\phi(G)^2}{2}$$

where α_G^2 is the minimum Cheeger ratio of subsets S_i consisting of vertices with the largest i values in the eigenvector associated with $lambda_G$, over all $i \in [n]$

Generalization to many partitions

1. Perform eigenvalue decomposition to find the eigenvectors of L_{sym}.
2. Select the k largest eigenvectors $e_1, e_2, ..., e_k$ of L_{sym} associated to the largest eigenvalues $\lambda_1, \lambda_2, ..., \lambda_k$
3. Form the matrix Y from the matrix $X = [e_1, e_2, ..., e_k]$ given by

$$Y_{ij} = \frac{X_{ij}}{\left(\sum_j X_{ij}^2\right)^{\frac{1}{2}}}$$

4. Treating each row of Y as a point in $\mathbb{R}^k$, cluster them into k clusters using $K-means$
5. Finally, assign the original vertex to cluster j if and only if row i of the matrix was assigned to cluster j

[2] [1] [3]

2.2.4 Spectral partitioning and Normalized Cut

2.3 Literature review

2.3.1 Graph Convolutional Neural Networks and GraphSAGE

2.3.2 Generalizable Approximate Graph Partitioning (GAP) Framework

2.3.3 PinSAGE and Markov Chain Negative Sampling (MCNS)

CHAPTER 3

Proposed solution (Graph Partitioning for Large Graphs)

CHAPTER 4

EXPERIMENTAL RESULTS

CHAPTER 5

Conclusion

Las conclusiones y el trabajo a futuro inicia aquí...

5.1 Contributions

5.2 Recommendations and future work

APPENDIX A

ANALISIS

El apéndice inicia aquí.

Bibliography

[1] Alice Gatti, Zhixiong Hu, Tess Smidt, Esmond G. Ng, and Pieter Ghysels.
Deep Learning and Spectral Embedding for Graph Partitioning, pages 25–36.

[2] William L. Hamilton.
Graph representation learning.
Synthesis Lectures on Artificial Intelligence and Machine Learning, 14(3):1–159.

[3] Azade Nazi, Will Hang, Anna Goldie, Sujith Ravi, and Azalia Mirhoseini.
Gap: Generalizable approximate graph partitioning framework.
ArXiv, abs/1903.00614, 2019.