Unsupervised Attributed Graph Learning: Models and Applications

by

Amin Salehi

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved September 2019 by the
Graduate Supervisory Committee:

Hasan Davulcu, Chair
Huan Liu
Baoxin Li
Hanghang Tong

ARIZONA STATE UNIVERSITY

December 2019

ABSTRACT

Graph is a ubiquitous data structure, which appears in a broad range of real-world scenarios. Accordingly, there has been a surge of research to represent and learn from graphs in order to accomplish various machine learning and graph analysis tasks. However, most of these efforts only utilize the graph structure while nodes in real-world graphs usually come with a rich set of attributes. Typical examples of such nodes and their attributes are users and their profiles in social networks, scientific articles and their content in citation networks, protein molecules and their gene sets in biological networks as well as web pages and their content on the Web. Utilizing node features in such graphs—attributed graphs—can alleviate the graph sparsity problem and help explain various phenomena (e.g., the motives behind the formation of communities in social networks). Therefore, further study of attributed graphs is required to take full advantage of node attributes.

In the wild, attributed graphs are usually unlabeled. Moreover, annotating data is an expensive and time-consuming process, which suffers from many limitations such as annotators' subjectivity, reproducibility, and consistency. The challenges of data annotation and the growing increase of unlabeled attributed graphs in various real-world applications significantly demand unsupervised learning for attributed graphs.

In this dissertation, I propose a set of novel models to learn from attributed graphs in an unsupervised manner. To better understand and represent nodes and communities in attributed graphs, I present different models in node and community levels. In node level, I utilize node features as well as the graph structure in attributed graphs to learn distributed representations of nodes, which can be useful in a variety of downstream machine learning applications. In community level, with a focus on social media, I take advantage of both node attributes and the graph structure to discover not only communities but also their sentiment-driven profiles and inter-community

relations (i.e., alliance, antagonism, or no relation). The discovered community profiles and relations help to better understand the structure and dynamics of social media.

To my parents and fiancée for their love and support

ACKNOWLEDGMENTS

I am grateful of my lovely parents, Mahnaz and Fathollah, and my beloved fiancée, Maryam, who have always been supportive through my journey. Without your love and support, none of my ambitions could come true.

I would like to express my gratitude to my advisor, Dr. Hasan Davulcu, for his support and open-mindedness. He created an environment in our lab in which I could focus on my research without any stress. I would also like to thank my committee members, Dr. Huan Liu, Dr. Baoxin Li, and Dr. Hanghang Tong for their insightful comments and helpful suggestions. I took social media mining and advanced social media mining with Dr. Huan Liu. In addition to learning interesting topics, I was impressed by his passion to help students engage and learn more. With Dr. Baoxin Li, I had statistical machine learning and deep learning in which I learned many state-of-the-art machine learning techniques. Dr. Hanghang Tong thought me many traditional machine learning techniques in statistical machine learning course. It was an honor to have all of you in my committee.

Finally, I am happy to give my thanks to my group members for their support and friendship; especially Mert Ozer who was a true friend during the hardest times. It was a pleasure to work alongside all of you for several years.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Chapter 1

INTRODUCTION

In a broad range of real-world applications, data can be represented as graphs. Social networks, the Wide World Web, biological networks, computer networks are some examples, which can be modeled as graphs. Accordingly, there has been a surge of research to learn from graphs in order to accomplish various machine learning and graph analysis tasks. However, most of these efforts only utilize the graph structure while nodes in real-world graphs usually come with a rich set of attributes (i.e. features). Typical examples of such nodes and their attributes are users and their profiles in social networks, scientific articles and their text in citation networks, protein molecules and their gene sets in biological networks as well as web pages and their content on the Web. Graphs in which their nodes come with attributes are called attributed graphs [1] . Node attributes in attributed graphs have the potential to alleviate the graph sparsity problem and explain various phenomena in graphs (e.g., the motives behind the formation of communities in social network graphs).

In the wild, attributed graphs are usually unlabeled. Moreover, annotating data is an expensive and time-consuming process, which suffers from many limitations such as annotators' subjectivity, reproducibility, and consistency. The challenges of data annotation and the growing increase of unlabeled attributed graphs in various real-world applications significantly demand unsupervised learning for attributed graphs.

In this dissertation, I propose a set of novel models for unsupervised attributed graph learning in node and community levels in order to better understand and rep-

[1]The terms "attributed graphs", "attributed networks", and "graph-structured data" are exchangeably used in this dissertation.

resent nodes and communities in attributed graphs. In node level, I utilize node attributes as well as the graph structure to learn distributed representations of nodes in graphs, which can be useful for many downstream machine learning tasks. In community level, with a focus on social media, I take advantage of both node attributes and the graph structure to not only detect communities but also their sentiment-driven profiles and inter-community relations (i.e., alliance, antagonism, or no relation). The discovered community profiles and inter-community relations help us to better understand the structure and dynamics of social media.

1.1 Research Challenges

The research challenges I face in this dissertation are as follows:

- How can I utilize node attributes as well as the graph structure to learn low-dimensional vector representations of nodes?
- How can I utilize node attributes as well as the graph structure in social media to detect and profile communities in a way that community profiles represent the collective opinions of community members?
- How can I utilize node attributes as well as the graph structure in social media to detect communities and their inter-community relations (i.e., alliance, antagonism, or no relation)?

1.2 Contributions

The contributions of this dissertation are summarized as follows:

- Proposing a novel neural network architecture to embed nodes in attributed graphs in such a way that their learned node representations encode the graph structure and node attributes.

- Proposing a novel framework to detect and profile communities in a way that a community profile reflects the collective opinions of community members.
- Presenting a novel framework to detect communities and their inter-community relations.
- Conducting experiments on real-world datasets to verify the efficacy of the proposed frameworks.

1.3 Organization

The rest of this dissertation is organized as follows. In Chapter 2, I propose a graph auto-encoder equipped with self-attention mechanism to learn the representations of nodes in attributed graphs. In Chapter 3, I present a novel framework to detect and profile communities in a way that a community profile reflects the collective opinions of community members. In Chapter 4, I present a novel framework to detect communities and their inter-community relations.

Chapter 2

UTILIZING NODE ATTRIBUTES FOR GRAPH REPRESENTATION LEARNING

2.1 Overview

Auto-encoders have emerged as a successful framework for unsupervised learning. However, conventional auto-encoders are incapable of utilizing explicit relations in structured data. To take advantage of relations in graph-structured data, several graph auto-encoders have recently been proposed, but they neglect to reconstruct either the graph structure or node attributes. This hinders their capability to learn rich node representations. In this chapter, I present the graph attention auto-encoder (GATE), a neural network architecture for unsupervised representation learning on graph-structured data (i.e., attributed graphs). The proposed architecture is able to reconstruct graph-structured inputs, including both node attributes and the graph structure, through stacked encoder/decoder layers equipped with self-attention mechanisms. In the encoder, by considering node attributes as initial node representations, each layer generates new representations of nodes by attending over their neighbors' representations. In the decoder, I attempt to reverse the encoding process to reconstruct node attributes. Moreover, node representations are regularized to reconstruct the graph structure. The proposed architecture does not need to know the graph structure upfront, and thus it can be utilized for inductive learning. My experiments demonstrate competitive performance on several node classification benchmark datasets for transductive and inductive tasks.

2.2 Introduction

Low-dimensional vector representations of nodes in graphs have demonstrated their utility in a broad range of machine learning tasks. Such tasks include node classification (Grover and Leskovec, 2016), recommender systems (Ying *et al.*, 2018), community detection (Wang *et al.*, 2017b), graph visualization (Perozzi *et al.*, 2014; Tang *et al.*, 2015), link prediction (Wei *et al.*, 2017) and relational modeling (Schlichtkrull *et al.*, 2018). Accordingly, there has been a surge of research to learn better node representations. However, most of the proposed methods (Grover and Leskovec, 2016; Belkin and Niyogi, 2002; He and Niyogi, 2004; Ahmed *et al.*, 2013; Cao *et al.*, 2015; Ou *et al.*, 2016; Perozzi *et al.*, 2014; Grover and Leskovec, 2016; Perozzi *et al.*, 2016; Chamberlain *et al.*, 2017; Tian *et al.*, 2014; Wang *et al.*, 2016; Tang *et al.*, 2015; Cao *et al.*, 2016; Chen *et al.*, 2018a) only utilize the graph structure while nodes in real-world graphs usually come with a rich set of attributes (i.e. features). Typical examples are users in social networks, scientific articles in citation networks, protein molecules in biological networks and web pages on the Internet.

Significant efforts have been made (Huang *et al.*, 2017b; Yang *et al.*, 2016; Defferrard *et al.*, 2016; Monti *et al.*, 2017; Kipf and Welling, 2017; Velickovic *et al.*, 2018; Hamilton *et al.*, 2017) to utilize node attributes for graph representation learning. Nevertheless, the most successful methods, notably graph convolutional networks (Kipf and Welling, 2017) and graph attention networks (Velickovic *et al.*, 2018), depend on label information, which is not available in many real-world applications. Moreover, the process of annotating data suffers from many limitations, such as annotators' subjectivity, reproducibility, and consistency.

To avoid the challenges of annotating data, several unsupervised graph embedding methods (Kipf and Welling, 2016; Duran and Niepert, 2017; Pan *et al.*, 2018;

Veličković *et al.*, 2019; Gao and Huang, 2018; Huang *et al.*, 2017a; Yang *et al.*, 2015) have been proposed, but these methods suffer from at least one of the three following problems. First, despite utilizing node features, some of these models (Kipf and Welling, 2016; Pan *et al.*, 2018; Gao and Huang, 2018) heavily depend on the graph structure. This hinders their capability to fully exploit node features. Second, many (Gao and Huang, 2018; Huang *et al.*, 2017a; Yang *et al.*, 2015) are not capable of inductive learning, which is crucial to encounter unseen nodes (e.g., new users in social networks, recently published scientific articles and new web pages on the Internet). Third, even though some efforts have been made (Veličković *et al.*, 2019; Hamilton *et al.*, 2017) to address inductive learning tasks, they are not unified architectures for both transductive and inductive tasks.

Auto-encoders have recently become popular for unsupervised learning due to their ability to capture complex relationships between input's attributes through stacked non-linear layers (Baldi, 2012; Bengio *et al.*, 2013). However, conventional auto-encoders are not able to take advantage of explicit relations in structured data. To utilize relations in graph-structured data, several graph auto-encoders (Kipf and Welling, 2016; Wang *et al.*, 2017a; Pan *et al.*, 2018) have been proposed. Although the encoders in these models fully utilize graph-structured inputs, the decoders neglect to reconstruct either the graph structure or node attributes. This hinders their capability to learn rich node representations.

Another successful neural network paradigm is the attention mechanism (Bahdanau *et al.*, 2014), which has been extremely useful in tackling many machine learning tasks (Chorowski *et al.*, 2015; Chen *et al.*, 2016; Wang *et al.*, 2018), particularly sequence-based tasks (Vaswani *et al.*, 2017; Luong *et al.*, 2015; Dehghani *et al.*, 2018; Rush *et al.*, 2015). The state-of-the-art attention mechanism is self-attention or intra-attention, which computes the representation of an input (e.g., a set or sequence) by

focusing on its most relevant parts. Self-attention has been successfully applied to a variety of tasks including machine translation (Vaswani *et al.*, 2017), video classification (Wang *et al.*, 2018) and question answering (Dehghani *et al.*, 2018). Nonetheless, the majority of these efforts target supervised learning tasks, and few efforts (Devlin *et al.*, 2018; He *et al.*, 2017) are made to tackle unsupervised learning tasks.

In this chapter, I present a novel graph auto-encoder to learn node representations within graph-structured data (i.e., attributed graphs) in an *unsupervised manner*. Our auto-encoder takes in and reconstructs node features by utilizing the graph structure through stacked encoder/decoder layers. In the encoder, node attributes are fed into stacked layers to generate node representations. By considering node features as initial node representations, each encoder layer generates new representations of nodes by utilizing neighbors' representations according to their relevance, which is determined by a graph attention mechanism. In the decoder, the architecture aims to reverse the entire encoding process to reconstruct node attributes. To this end, each decoder layer attempts to reverse the process of its corresponding encoder layer. Moreover, node representations are regularized to reconstruct the graph structure. To our knowledge, no auto-encoder is capable of reconstructing both node attributes and the graph structure. Our architecture can also be applied to inductive learning tasks since it doesn't need to know the graph structure upfront.

Our key contributions are summarized as follows:

- I propose a novel graph auto-encoder for unsupervised representation learning on graph-structured data by reconstructing both node features and the graph structure.
- I utilize self-attention for unsupervised attributed graph representation learning.
- I present a unified neural architecture capable of both transductive and induc-

tive learning.

The rest of the chapter is organized as follows. I review related work in Section 2.3. In Section 2.4, I formally define the problem of unsupervised representation learning on graph-structured data. Section 2.5 presents the architecture of our proposed graph auto-encoder. In Section 2.6, I quantitatively and qualitatively evaluate GATE using several benchmark datasets for both transductive and inductive learning tasks. Section 2.7 concludes the chapter.

2.3 Related Work

2.3.1 Graph Representation Learning

Most of the graph embedding methods fall into one of the following three categories: factorization based, random walk based, and auto-encoder based approaches.

Factorization based approaches are inspired by matrix factorization methods, which assume that the data lies in a low dimensional manifold. Laplacian Eigenmaps (Belkin and Niyogi, 2002) and LPP (He and Niyogi, 2004) rely on eigendecomposition to preserve the local manifold structure. Due to expensive eigendecomposition operations, these methods face difficulty to tackle large-scale graphs. To alleviate this problem, several techniques—notably the Graph Factorization (GF) (Ahmed *et al.*, 2013), GraRep (Cao *et al.*, 2015) and HOPE (Ou *et al.*, 2016)—have been proposed. These methods differ mainly in their node similarity calculation. The graph factorization computes node similarity based on the first-order proximities directly extracted from the adjacency matrix. To capture more accurate node similarity, GraRep and HOPE utilize the high-order proximities obtained from different powers of the adjacency matrix and similarity measures (i.e., cosine similarity) respectively.

Random walk based approaches assume a pair of nodes to be similar if they are

close in simulated random walks over the graph. DeepWalk (Perozzi *et al.*, 2014) and node2vec (Grover and Leskovec, 2016) are the most successful methods in this category and differ primarily in their random walk generation. DeepWalk simulates uniform random walks while node2vec relies on a biased random walk generation. Preozzi et al. (Perozzi *et al.*, 2016) extend DeepWalk to encode multiscale node relationships in the graph. In contrast to DeepWalk and node2vec, which embed nodes in the Euclidean space, Chamberlan et al. (Chamberlain *et al.*, 2017) utilize the hyperbolic space.

Factorization based and random walk based approaches adopt shallow models, which are incapable of capturing complex graph structures. To solve this problem, auto-encoder based approaches are proposed to capture non-linear graph structures by using deep neural networks. Tian et al. (Tian *et al.*, 2014) present a stacked sparse auto-encoder to embed nodes by reconstructing the adjacency matrix. Moreover, Wang et al. (Wang *et al.*, 2016) propose a stacked auto-encoder, which reconstructs the second-order proximities by using the first-order proximities as a regularization. Cao et al. (Cao *et al.*, 2016) use stacked denoising auto-encoder to reconstruct the pointwise mutual information matrix.

Although the majority of graph embedding methods fall into one of the aforementioned categories, there are still some exceptions. For instance, LINE (Tang *et al.*, 2015) is a successful shallow embedding method to preserve both the local and global graph structures. Another example is HARP (Chen *et al.*, 2018a), which introduces a graph processing step coarsening a graph into smaller graphs at different levels of granularity, and then it embeds them from the smaller graph to the largest one (i.e., the original graph) by using one of the graph embedding methods (i.e., DeepWalk, node2vec, and LINE).

2.3.2 *Attributed Graph Representation Learning*

The graph embedding methods described in the previous section only utilize the graph structure to learn node representations. However, nodes in real-world graphs usually come with a rich set of attributes. To take advantage of node features, many attributed graph embedding methods have been proposed, which fall into two main categories: supervised and unsupervised approaches.

Supervised attributed graph embedding approaches embed nodes by utilizing label information. For example, Huang et al. (Huang *et al.*, 2017b) propose a supervised method leveraging spectral techniques to project the adjacency matrix, node feature matrix, and node label matrix into a common vector space. Hamilton et al. (Hamilton *et al.*, 2017) present four variants of GraphSAGE, a framework to compute node embeddings in an inductive manner. Many approaches address graphs with partial label information. For example, Graph Convolution Network (GCN) (Kipf and Welling, 2017) incorporates spectral convolutions into neural networks. Graph Attention Network (GAT) (Velickovic *et al.*, 2018) utilizes an attention mechanism to determine the influence of neighboring nodes in final node representations.

The unsupervised attributed graph embedding methods address the lack of label information, which exists in many real-world applications. Yang et al. (Yang *et al.*, 2015) and Huang et al. (Huang *et al.*, 2017a) propose matrix factorization methods to combine the graph structure and node attributes. Moreover, Kipf et al. (Kipf and Welling, 2016) propose two graph auto-encoders utilizing graph convolution networks. Pan et al. (Pan *et al.*, 2018) also introduce a graph-encoder based on an adversarial approach. For graph clustering, Wang et al. (Wang *et al.*, 2017a) present a graph auto-encoder, which is able to reconstruct node features. However, these auto-encoders reconstruct either the graph structure or node attributes instead

of both. To alleviate this limitation, Gao et al. (Gao and Huang, 2018) propose a framework consisting of two conventional auto-encoders, which reconstruct the graph structure and node attributes separately. These two auto-encoders are regularized in a way that their learned representations of neighboring nodes are similar. However, their framework does not fully leverage the graph structure due to the incapability of conventional auto-encoders in utilizing explicit relations in structured data. Most of the aforementioned unsupervised methods are not designed for inductive learning, which is crucial to encounter unseen nodes. Velivckovic et al. (Veličković *et al.*, 2019) and Hamilton et al. (Hamilton *et al.*, 2017) propose unsupervised models for tackling inductive tasks, but their models are not unified frameworks for both transductive and inductive tasks.

2.4 Problem Statement

In this section, I present the notations used in the chapter and formally define the problem of unsupervised node representation learning on graph-structured data. I use bold upper-case letters for matrices (e.g., $\mathbf{X}$), bold lowercase letters for vectors (e.g., $\mathbf{x}$), and calligraphic fonts for sets (e.g., $\mathcal{N}$). Moreover, I represent the transpose of a matrix $\mathbf{X}$ as $\mathbf{X}^T$. The i^{th} element of vector $\mathbf{x}$ is denoted by $\mathbf{x}_i$. $\mathbf{X}_{ij}$ denotes the entry of matrix $\mathbf{X}$ at the i^{th} row and the j^{th} column. Table 2.1 summarizes the main notations used in the chapter.

In the attributed graph representation learning setup, we are provided with the node feature matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N]$, where N is the number of nodes in the graph and $\mathbf{x}_i \in \mathbb{R}^F$ corresponds to the i^{th} column of matrix $\mathbf{X}$, denoting the features of node i. We are also given the adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, representing the relations between nodes. Even though the matrix $\mathbf{A}$ may consist of real numbers, in our experiments, I assume the graph is unweighted and includes self-loops, i.e.,

Table 2.1: The Notations Used in Chapter 2.

Notations	**Definitions**
N	The number of nodes in the graph
E	The number of edges in the graph
L	The number of layers
$d^{(k)}$	The number of node representation dimensions in the k^{th} encoder/decoder layer
F	The number of node features ($d^{(0)} = F$)
P	The number of iterations (i.e., epochs)
$\mathbf{A} \in \mathbb{R}^{N \times N}$	The adjacency matrix
$\mathbf{H}^{(k)} \in \mathbb{R}^{d^{(k)} \times N}$	The node representation matrix generated by the k^{th} encoder layer
$\widehat{\mathbf{H}}^{(k)} \in \mathbb{R}^{d^{(k)} \times N}$	The node representation matrix reconstructed by the k^{th} decoder layer
$\mathbf{H} \in \mathbb{R}^{d^{(L)} \times N}$	The node representation matrix ($\mathbf{H} = \mathbf{H}^{(L)} = \widehat{\mathbf{H}}^{(L)}$)
$\mathbf{X} \in \mathbb{R}^{F \times N}$	The node feature matrix ($\mathbf{H}^{(0)} = \mathbf{X}$)
$\widehat{\mathbf{X}} \in \mathbb{R}^{F \times N}$	The reconstructed node feature matrix ($\widehat{\mathbf{X}} = \widehat{\mathbf{H}}^{(0)}$)
$\mathbf{C}^{(k)} \in \mathbb{R}^{N \times N}$	The attention matrix in the k^{th} encoder layer
$\widehat{\mathbf{C}}^{(k)} \in \mathbb{R}^{N \times N}$	The attention matrix in the k^{th} decoder layer
$\mathbf{h}_i{}^{(k)} \in \mathbb{R}^{d^{(k)}}$	The representation of node i generated by the k^{th} encoder layer
$\widehat{\mathbf{h}}_i^{(k)} \in \mathbb{R}^{d^{(k)}}$	The representation of node i reconstructed by the k^{th} decoder layer
$\mathbf{h}_i \in \mathbb{R}^{d^{(L)}}$	The representation of node i ($\mathbf{h}_i = \mathbf{h}_i{}^{(L)} = \widehat{\mathbf{h}}_i^{(L)}$)
$\mathbf{x}_i \in \mathbb{R}^F$	The features of node i ($\mathbf{h}_i^{(0)} = \mathbf{x}_i$)
$\widehat{\mathbf{x}}_i \in \mathbb{R}^F$	The reconstructed features of node i ($\widehat{\mathbf{x}}_i = \widehat{\mathbf{h}}_i^{(0)}$)
$\alpha_{ij}^{(k)}$	The attention coefficient indicating the relative relevance of neighboring node j to node i in the k^{th} encoder layer
$\widehat{\alpha}_{ij}^{(k)}$	The attention coefficient indicating the relative relevance of neighboring node j to node i in the k^{th} decoder layer
$\mathcal{N}_i$	The neighborhood of node i, including itself

$\mathbf{A}_{ij} = 1$ if there is an edge between node i and node j in the graph or i equals j, and $\mathbf{A}_{ij} = 0$ otherwise.

Given the node feature matrix $\mathbf{X}$ and the adjacency matrix $\mathbf{A}$, our objective is to learn node representations in the form of matrix $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_N]$, where $\mathbf{h}_i \in \mathbb{R}^D$ corresponds to the i^{th} column of matrix $\mathbf{H}$, denoting the representation of node i.

2.5 Architecture

In this section, I illustrate the architecture of the graph attention auto-encoder. First, I present the encoder and decoder to show how the proposed auto-encoder reconstructs node features using the graph structure. Then, I describe the proposed loss function, which learns node representations by minimizing the reconstruction loss of node features and the graph structure. In the end, I present the matrix formulation of GATE as well as its time and space complexities.

2.5.1 Encoder

The encoder in my architecture takes node features and generates node representations by using the graph structure through stacked layers. I use multiple encoder layers for two reasons. First, more layers make the model deeper, and hence increasing the learning capability. Second, they propagate node representations through the graph structure, resulting in richer node embeddings.

Each encoder layer generates new representations of nodes by utilizing their neighbors' representations according to their relevance. To determine the relevance between nodes and their neighbors, I use a self-attention mechanism with shared parameters among nodes, following the work of Velickovic et al. (Velickovic *et al.*, 2018). In the k^{th} encoder layer, the relevance of a neighboring node j to node i is computed as follows:

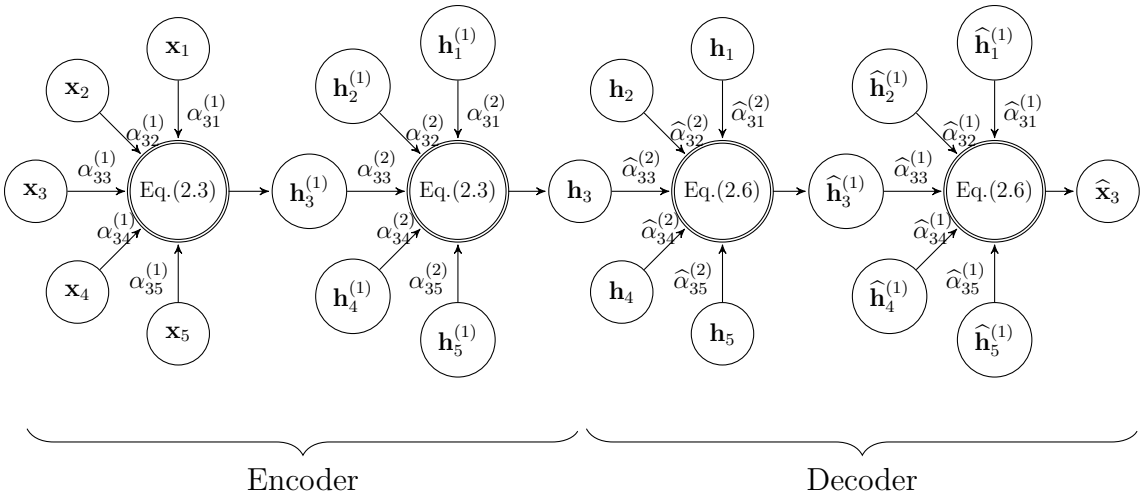


Figure 2.1: The Illustration of Reconstructing the Features of Node 3, with Neighborhood $\mathcal{N}_3 = \{1, 2, 3, 4, 5\}$, Using the Graph Attention Auto-encoder with Two Layers.

$$e_{ij}^{(k)} = \text{Sigmoid}\left(\mathbf{v}_s^{(k)^T}\sigma\left(\mathbf{W}^{(k)}\mathbf{h}_i^{(k-1)}\right) + \mathbf{v}_r^{(k)^T}\sigma\left(\mathbf{W}^{(k)}\mathbf{h}_j^{(k-1)}\right)\right) \tag{2.1}$$

where $\mathbf{W}^{(k)} \in \mathbb{R}^{d^{(k)} \times d^{(k-1)}}$, $\mathbf{v}_s^{(k)} \in \mathbb{R}^{d^{(k)}}$, and $\mathbf{v}_r^{(k)} \in \mathbb{R}^{d^{(k)}}$ are the trainable parameters of the k^{th} encoder layer, σ denotes the activation function and Sigmoid represents the sigmoid function (i.e., Sigmoid $(x) = 1/(1 + \exp^{-x})$).

To make the relevance coefficients of node i's neighbors comparable, I normalize them by using the softmax function as follows:

$$\alpha_{ij}^{(k)} = \frac{\exp\left(e_{ij}^{(k)}\right)}{\sum_{l \in \mathcal{N}_i} \exp\left(e_{il}^{(k)}\right)} \tag{2.2}$$

where $\mathcal{N}_i$ represents the neighborhood of node i (i.e., a set of nodes connected to node i according to the adjacency matrix $\mathbf{A}$, including node i itself).

By considering node features as initial node representations (i.e., $\mathbf{h}_i{}^{(0)} = \mathbf{x}_i$, $\forall i \in \{1, 2, .., N\}$), the k^{th} encoder layer generates the representation of node i in layer k as follows:

$$\mathbf{h}_i^{(k)} = \sum_{j \in \mathcal{N}_i} \alpha_{ij}^{(k)} \sigma\left(\mathbf{W}^{(k)} \mathbf{h}_j^{(k-1)}\right) \tag{2.3}$$

After applying L encoder layers, I consider the output of the last layer as the final node representations (i.e., $\mathbf{h}_i = \mathbf{h}_i^{(L)}$, $\forall i \in \{1, 2, .., N\}$).

2.5.2 Decoder

The encoder is reminiscent of graph attention networks (Velickovic *et al.*, 2018), which use supervised learning to embed nodes. My main contribution is reversing the encoding process in order to learn node representations without any supervision. To this end, I use a decoder with the same number of layers as the encoder. Each decoder layer attempts to reverse the process of its corresponding encoder layer. In other words, each decoder layer reconstructs the representations of nodes by utilizing the representations of their neighbors according to their relevance. The normalized relevance (i.e., attention coefficient) of a neighboring node j to node i in the k^{th} decoder layer is computed as follows:

$$\widehat{\alpha}_{ij}^{(k)} = \frac{\exp\left(\widehat{e}_{ij}^{(k)}\right)}{\sum_{l \in \mathcal{N}_i} \exp\left(\widehat{e}_{il}^{(k)}\right)} \tag{2.4}$$

$$\widehat{e}_{ij}^{(k)} = \text{Sigmoid}\left(\widehat{\mathbf{v}}_s^{(k)^T} \sigma\left(\widehat{\mathbf{W}}^{(k)} \widehat{\mathbf{h}}_i^{(k)}\right) + \widehat{\mathbf{v}}_r^{(k)^T} \sigma\left(\widehat{\mathbf{W}}^{(k)} \widehat{\mathbf{h}}_j^{(k)}\right)\right) \tag{2.5}$$

where $\widehat{\mathbf{W}}^{(k)} \in \mathbb{R}^{d^{(k-1)} \times d^{(k)}}$, $\widehat{\mathbf{v}}_s^{(k)} \in \mathbb{R}^{d^{(k-1)}}$, and $\widehat{\mathbf{v}}_r^{(k)} \in \mathbb{R}^{d^{(k-1)}}$ are the trainable parameters of the k^{th} decoder layer.

By considering the output of the encoder as the input of the decoder (i.e., $\widehat{\mathbf{h}}_i^{(L)} = \mathbf{h}_i^{(L)}$, $\forall i \in \{1, 2, .., N\}$), the k^{th} decoder layer reconstructs the representation of node i in layer $k-1$ as follows:

$$\widehat{\mathbf{h}}_i^{(k-1)} = \sum_{j \in \mathcal{N}_i} \widehat{\alpha}_{ij}^{(k)} \sigma \left(\widehat{\mathbf{W}}^{(k)} \widehat{\mathbf{h}}_j^{(k)} \right) \tag{2.6}$$

After applying L decoder layers, I consider the output of the last layer as the reconstructed node features (i.e., $\widehat{\mathbf{x}}_i = \widehat{\mathbf{h}}_i^{(0)}$, $\forall i \in \{1, 2, .., N\}$). Figure 2.1 illustrates the process of reconstructing node features in GATE through an example. Note that $\mathbf{h}_i^{(0)} = \mathbf{x}_i$, $\mathbf{h}_i = \mathbf{h}_i^{(2)} = \widehat{\mathbf{h}}_i^{(2)}$, and $\widehat{\mathbf{x}}_i = \widehat{\mathbf{h}}_i^{(0)}$, $\forall i \in \{1, 2, .., N\}$.

2.5.3 Loss Function

Graph-structured data include node features and the graph structure, and both should be encoded by high-quality node representations. I minimize the reconstruction loss of node features as follows:

$$\sum_{i=1}^{N} ||\mathbf{x}_i - \widehat{\mathbf{x}}_i||_2 \tag{2.7}$$

The absence of an edge between two nodes in the graph does not necessarily imply dissimilarity due to the possibility of feature similarity. Thus, I minimize the reconstruction loss of the graph structure by making the representations of neighboring nodes similar. I accomplish this by minimizing the following equation:

$$-\sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \log \left(\frac{1}{1 + \exp(-\mathbf{h}_i^T \mathbf{h}_j)} \right) \tag{2.8}$$

By merging Eq. (2.7) and Eq. (2.8), I minimize the reconstruction loss of node features and the graph structure as follows:

$$\text{Loss} = \sum_{i=1}^{N} ||\mathbf{x}_i - \widehat{\mathbf{x}}_i||_2 - \lambda \sum_{j \in \mathcal{N}_i} \log \left(\frac{1}{1 + \exp(-\mathbf{h}_i^T \mathbf{h}_j)} \right) \tag{2.9}$$

where λ controls the contribution of the graph structure reconstruction loss.

2.5.4 Matrix Formulation

Since the adjacency matrix $\mathbf{A}$ is usually very sparse in practice, I can leverage sparse matrix operations (e.g., sparse softmax) to tackle large graphs. Therefore, I present the corresponding matrix formulas for the aforementioned encoder and decoder equations.

Let us begin with obtaining the attention matrix $\mathbf{C}^{(k)} \in \mathbb{R}^{N \times N}$ in the k^{th} encoder layer, where $\mathbf{C}_{ij}^{(k)} = \alpha_{ij}^{(k)}$ if there is an edge between node i and node j, and $\mathbf{C}_{ij}^{(k)} = 0$ otherwise. I compute $\mathbf{C}^{(k)}$as follows:

$$\mathbf{C}^{(k)} = \text{Softmax}\left(\text{Sigmoid}\left(\mathbf{M}_s^{(k)} + \mathbf{M}_r^{(k)}\right)\right) \tag{2.10}$$

$$\mathbf{M}_s^{(k)} = \mathbf{A} \odot \left(\mathbf{v}_s^{(k)^T} \sigma\left(\mathbf{W}^{(k)}\mathbf{H}^{(k-1)}\right)\right) \tag{2.11}$$

$$\mathbf{M}_r^{(k)} = \mathbf{A} \odot \left(\mathbf{v}_r^{(k)^T} \sigma\left(\mathbf{W}^{(k)}\mathbf{H}^{(k-1)}\right)\right)^T \tag{2.12}$$

where $\odot$ is element-wise multiplication with broadcasting capability and σ denotes the activation function.

By considering $\mathbf{H}^{(0)} = \mathbf{X}$, the k^{th} encoder layer generates node representations in layer k as follows:

$$\mathbf{H}^{(k)} = \sigma\left(\mathbf{W}^{(k)}\mathbf{H}^{(k-1)}\right)\mathbf{C}^{(k)} \tag{2.13}$$

After applying L encoder layers, I consider $\mathbf{H}^{(L)}$ as the final node representation matrix (i.e., $\mathbf{H} = \mathbf{H}^{(L)}$).

The attention matrix $\widehat{\mathbf{C}}^{(k)} \in \mathbb{R}^{N \times N}$ in the k^{th} decoder layer, where $\widehat{\mathbf{C}}_{ij}^{(k)} = \widehat{\alpha}_{ij}^{(k)}$ if there is an edge between node i and node j, and $\widehat{\mathbf{C}}_{ij}^{(k)} = 0$ otherwise, is computed as follows:

$$\widehat{\mathbf{C}}^{(k)} = \text{Softmax}\left(\text{Sigmoid}\left(\widehat{\mathbf{M}}_s^{(k)} + \widehat{\mathbf{M}}_r^{(k)}\right)\right) \tag{2.14}$$

$$\widehat{\mathbf{M}}_s^{(k)} = \mathbf{A} \odot \left(\widehat{\mathbf{v}}_s^{(k)^T} \sigma \left(\widehat{\mathbf{W}}^{(k)} \widehat{\mathbf{H}}^{(k)} \right) \right) \tag{2.15}$$

$$\widehat{\mathbf{M}}_r^{(k)} = \mathbf{A} \odot \left(\widehat{\mathbf{v}}_r^{(k)^T} \sigma \left(\widehat{\mathbf{W}}^{(k)} \widehat{\mathbf{H}}^{(k)} \right) \right)^T \tag{2.16}$$

By considering $\widehat{\mathbf{H}}^{(L)} = \mathbf{H}^{(L)}$, the k^{th} decoder layer reconstructs node representations in layer $k-1$ as follows:

$$\widehat{\mathbf{H}}^{(k-1)} = \sigma \left(\widehat{\mathbf{W}}^{(k)} \widehat{\mathbf{H}}^{(k)} \right) \widehat{\mathbf{C}}^{(k)} \tag{2.17}$$

After applying L decoder layers, I consider $\widehat{\mathbf{H}}^{(0)}$ as the reconstructed node feature matrix (i.e., $\widehat{\mathbf{X}} = \widehat{\mathbf{H}}^{(0)}$).

Algorithm 1 shows the forward propagation of the proposed architecture using matrix formulation.

2.5.5 Complexity

The proposed auto-encoder is highly efficient because the operations involved in the graph attention mechanisms can be parallelized across edges, and the rest of the operations in the encoder and decoder can be parallelized across nodes. Theoretically, the time complexity of the architecture for one iteration can be expressed as follows:

$$O(NFD + ED) \tag{2.18}$$

where N and E are respectively the number of nodes and edges in the graph, F is the number of node features and D is the maximum $d^{(k)}$ in all layers (i.e., $D = \max_{k \in \{1,2,..,L\}} d^{(k)}$).

By taking advantage of sparse matrix operations, the space complexity of the proposed auto-encoder is linear in terms of the number of nodes and edges.

Algorithm 1 GATE Forward Propagation Algorithm Using Matrix Formulation.

Input: The node feature matrix $\mathbf{X}$ and the adjacency matrix $\mathbf{A}$

output: The node representation matrix $\mathbf{H}$ and the reconstructed node feature matrix $\widehat{\mathbf{X}}$

Initialize $\mathbf{W}^{(k)}$, $\widehat{\mathbf{W}}^{(k)}$, $\mathbf{v}_s^{(k)}$, $\widehat{\mathbf{v}}_s^{(k)}$, $\mathbf{v}_r^{(k)}$ and $\widehat{\mathbf{v}}_r^{(k)}$, $\forall k \in \{1, 2, .., L\}$

$\mathbf{H}^{(0)} = \mathbf{X}$

for *epoch* $\leftarrow 1$ **to** P **do**

  for $k \leftarrow 1$ **to** L **do**

    Compute $\mathbf{C}^{(k)}$ according to Eq. (2.10)

    $\mathbf{H}^{(k)} = \sigma\left(\mathbf{W}^{(k)}\mathbf{H}^{(k-1)}\right)\mathbf{C}^{(k)}$

  end for

  $\widehat{\mathbf{H}}^{(L)} = \mathbf{H}^{(L)}$

  for $k \leftarrow L$ **to** 1 **do**

    Compute $\widehat{\mathbf{C}}^{(k)}$ according to Eq. (2.14)

    $\widehat{\mathbf{H}}^{(k-1)} = \sigma\left(\widehat{\mathbf{W}}^{(k)}\widehat{\mathbf{H}}^{(k)}\right)\widehat{\mathbf{C}}^{(k)}$

  end for

  $\mathbf{H} = \mathbf{H}^{(L)}$

  $\widehat{\mathbf{X}} = \widehat{\mathbf{H}}^{(0)}$

end for

2.6 Evaluation

In this section, I quantitatively and qualitatively evaluate the proposed GATE architecture using several benchmark datasets. Section 2.6.1, 2.6.2, and 2.6.3 respectively describe the datasets, baselines, and experimental setup used in the experiments. In Section 2.6.4, I quantitatively evaluate the efficacy of the architecture. Section 2.6.5 investigates the impact of the three main components used in the proposed architecture, namely the self-attention mechanism, graph structure reconstruc-

Table 2.2: The Statistics of the Benchmark Datasets Used in Chapter 2.

Dataset	Nodes	Edges	Features	Classes	Train/Val/Test Nodes
Cora	2,708	5,429	1,433	7	140/500/1,000
Citeseer	3,327	4,732	3,703	6	120/500/1,000
Pubmed	19,717	44,338	500	3	60/500/1,000

tion, and node feature reconstruction. Finally, I investigate the quality of the node representations learned by GATE in Section 2.6.6.

2.6.1 Datasets

For transductive tasks, I use three benchmark datasets—Cora, Citeseer and Pubmed (Sen *et al.*, 2008)—that are widely used to evaluate attributed graph embedding methods. In all datasets, each node belongs to one class. I follow the experimental setup of Yang et al. (Yang *et al.*, 2016), where 20 nodes per class are used for training. In the transductive setup, I have access to the graph structure and all nodes' feature vectors during training. I evaluate the predictive performance of each method on 1000 test nodes. The statistics of the datasets are presented in Table 2.2.

For inductive tasks, I also use the same datasets and experimental setup in order to evaluate the generalization power of different methods to unseen nodes by comparing the difference between their performance in transductive and inductive tasks for the same dataset. As required by inductive learning, any information related to (unseen) test nodes, including features and edges, are completely unobserved during training.

2.6.2 Baselines

I compare my proposed auto-encoder against the following state-of-the-art unsupervised methods:

- **DeepWalk** (Perozzi *et al.*, 2014): DeepWalk is a graph embedding method, which trains Skipgram model (Mikolov *et al.*, 2013) on simulated random walks over the graph.

- **Enhanced DeepWalk (DeepWalk + features)**: This baseline is a variant of DeepWalk concatenating raw node features and DeepWalk embeddings to take advantage of node features.

- **Graph Auto-Encoder (GAE)** (Kipf and Welling, 2016): GAE uses graph convolutional networks as the encoder and reconstruct the graph structure in the encoder.

- **Variational Graph Auto-Encoder (VGAE)** (Kipf and Welling, 2016): VGAE is the variational version of GAE.

- **GraphSAGE** (Hamilton *et al.*, 2017): GraphSAGE has four unsupervised variants, which differ in their feature aggregator as follows: GraphSAGE-GCN (applying a convolution-style aggregator), GraphSAGE-mean (taking the element-wise mean of feature vectors), GraphSAGE-LSTM (aggregating by providing neighboring nodes' features into a LSTM), and GraphSAGE-pool (performing an element-wise max-pooling operation after applying a fully-connected neural network).

- **Deep Graph Infomax (DGI)** (Veličković *et al.*, 2019): DGI is an unsupervised attributed graph embedding, which simultaneously estimates and maxi-

mizes the mutual information between the graph-structured input and learned high-level graph summaries. stochastic algorithm for graph convolutional networks, which uses neighborhood sampling and historical hidden representations to reduce the receptive field of the graph convolution.

For transductive tasks, I compare my proposed auto-encoder against unsupervised approaches, which are DeepWalk, enhanced DeepWalk, VGAE, GAE, and DGI. For inductive tasks, I similarly compare GATE against unsupervised approaches, which are VGAE, GAE, and four unsupervised variants of GraphSAGE.

2.6.3 Experimental Setup

In the experiments, Adam optimizer (Kingma and Ba, 2014) is used to learn model parameters with an initial learning rate of 10^{-4}. For all datasets, I use two layers with 512 node representation dimensions (i.e., $d^{(1)} = d^{(2)} = 512$). I set the number of epochs to 100 for Cora and Citeseer, and 500 for Pubmed. I also set λ to 0.5 for Cora and Pubmed, and 20 for Citeseer. I use only half of the trainable parameters by setting $\widehat{\mathbf{W}}^{(k)} = \mathbf{W}^{(k)^T}$ and $\widehat{\mathbf{C}}^{(k)} = \mathbf{C}^{(k)}$. Moreover, σ is set to the identity function, empirically resulting in better performance compared to other activation functions. I have used Tensorflow (Abadi *et al.*, 2016) to implement GATE [1] .

For the baselines to which I directly compare GATE, I use their default hyperparameter settings as well as the following settings. I perform a hyperparameter sweep on initial learning rates $\{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}\}$. I also swept over the number epochs in the set $\{50, 100, 200, 300\}$ for VGAE and GAE due to their sensitivity to this hyperparameter. I also set the number of node representation dimensions to 512 for all baselines.

[1]The implementation of the proposed architecture may be found at: `https://github.com/amin-salehi/GATE`

Table 2.3: Transductive Node Classification Accuracies on the Cora, Citeseer and Pubmed Datasets.

Available Data	**Method**	**Cora**	**Citeseer**
Pubmed			
Raw features	$47.9 \pm 0.4\%$	$49.4 \pm 0.2\%$	$69.1 \pm 0.3\%$
DeepWalk (Perozzi et al. (Perozzi *et al.*, 2014))	67.2%	43.2%	65.3%
DeepWalk + features	$70.7 \pm 0.6\%$	$51.4 \pm 0.5\%$	$74.3 \pm 0.9\%$
VGAE (Kipf & Welling (Kipf and Welling, 2016))	$72.4 \pm 0.2\%$	$55.7 \pm 0.2\%$	71.6 ± 0.4 %
GAE (Kipf & Welling (Kipf and Welling, 2016))	$81.8 \pm 0.1\%$	$69.2 \pm 0.9\%$	$78.2 \pm 0.1\%$
DGI (Velickovic et al. (Veličković *et al.*, 2019))	$82.3 \pm 0.6\%$	$\mathbf{71.8} \pm 0.7\%$	$76.8 \pm 0.6\%$
GATE (ours)	$\mathbf{83.2} \pm 0.6\%$	$\mathbf{71.8} \pm 0.8\%$	$\mathbf{80.9} \pm 0.3\%$

Table 2.4: Inductive Node Classification Accuracies on the Cora, Citeseer and Pubmed Datasets.

Available Data	**Method**	**Cora**	**Citeseer**
Pubmed			
GraphSAGE-LSTM (Hamilton et al. (Hamilton *et al.*, 2017))	$50.1 \pm 0.2\%$	$40.3 \pm 0.2\%$	$77.1 \pm 0.1\%$
GraphSAGE-pool (Hamilton et al. (Hamilton *et al.*, 2017))	$57.5 \pm 0.2\%$	$45.9 \pm 0.2\%$	$79.9 \pm 0.1\%$
VGAE (Kipf & Welling (Kipf and Welling, 2016))	$58.4 \pm 0.4\%$	$55.4 \pm 0.2\%$	$71.1 \pm 0.2\%$
GraphSAGE-mean (Hamilton et al. (Hamilton *et al.*, 2017))	$67.0 \pm 0.2\%$	$52.8 \pm 0.1\%$	$79.3 \pm 0.1\%$
GraphSAGE-GCN (Hamilton et al. (Hamilton *et al.*, 2017))	$74.3 \pm 0.1\%$	$54.5 \pm 0.1\%$	$77.5 \pm 0.1\%$
GAE (Kipf & Welling (Kipf and Welling, 2016))	$80.5 \pm 0.1\%$	$69.1 \pm 0.9\%$	$78.1 \pm 0.2\%$
GATE (ours)	$\mathbf{82.5} \pm 0.5\%$	$\mathbf{71.5} \pm 0.7\%$	$\mathbf{80.8} \pm 0.3\%$

2.6.4 Comparison

In this section, I compare out proposed method with the aforementioned state-of-the-art baselines based on transductive and inductive node classifications. For transductive node classification, I report the mean classification accuracy (with stan-

dard deviation) of the proposed method on the test nodes after 100 runs of training (followed by logistic regression). The accuracies for DeepWalk are retrieved from Kipf & Welling (Kipf and Welling, 2017). I also reuse the metrics reported in Velickovic et al. (Veličković *et al.*, 2019) for the performance of enhanced DeepWalk, DGI, and logistic regression with raw features. Moreover, I directly compare my method against GAE and VGAE.

Table 2.3 shows the transductive node classification accuracies for the Cora, Citeseer, and Pubmed datasets. Accordingly, I make the following observations:

- GATE achieves strong performance across all three datasets. Particularly, GATE outperforms all baselines on the Cora and Pubmed datasets.

- GATE outperforms or matches all baselines across all datasets. I observe an improvement of 2.7% and 0.9% over the second best baseline for Pubmed and Cora respectively.

- The reconstruction of node features by GATE results in a considerable improvement compared to the graph auto-encoder baselines reconstructing only the graph structure. Compared to the best graph auto-encoder baseline (i.e., GAE), my method achieves an improvement gain of 2.7%, 2.6%, and 1.4% on Pubmed, Citeseer, and Cora respectively.

For inductive node classification, I utilize the same datasets used for the transductive tasks. This enables us to compare the performance of GATE between transductive and inductive tasks for the same dataset in order to evaluate the generalization power of the proposed auto-encoder to unseen nodes. I report the mean classification accuracy (with standard deviation) of my method on the (unseen) test nodes after 100 runs of training (followed by logistic regression). I directly compare my method against VGAE, GAE, and four variants of GraphSAGE.

Table 2.4 shows the inductive node classification accuracies for the Cora, Citeseer, and Pubmed datasets. Accordingly, I make the following observations:

- GATE exceeds the performance of the baselines across all three datasets. I are able to improve upon the best baselines by a margin of 2.4%, 2%, and 0.9% on Citeseer, Cora, and Pubmed respectively.

- I can observe that GATE achieves similar accuracies for inductive and transductive tasks with regard to the same dataset. For example, the accuracy difference between inductive and transductive tasks is 0.1%, 0.3%, and 0.7% on Pubmed, Citeseer, and Cora respectively. This suggests that GATE naturally generalizes to unseen nodes.

2.6.5 In-depth Analysis

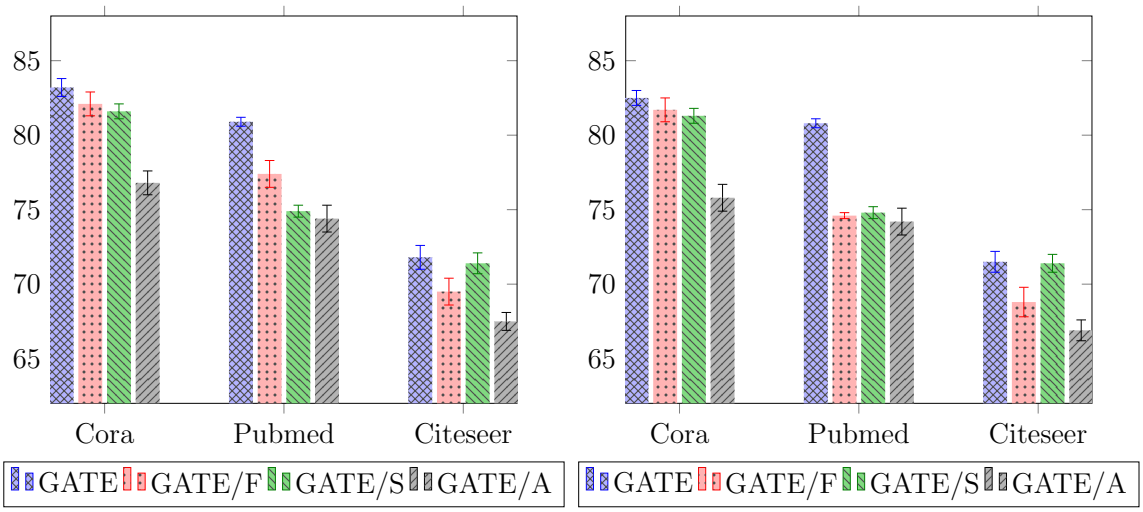
In this section, I investigate the impact of the three main components used in the proposed architecture, namely the self-attention mechanism, graph structure reconstruction and node feature reconstruction. In the experiments, I use the following variants of my architecture:

- **GATE**: The full version of the proposed auto-encoder which includes all three components.

- **GATE/A**: A variant of the architecture which includes all components except the self-attention mechanism. In other words, I assign the same importance to each neighbor.

- **GATE/S**: A variant of the architecture which includes all components except the graph structure reconstruction.

- **GATE/F**: A variant of the architecture which includes all components except the node feature reconstruction.



(a) Trandsuctive (b) Inductive

Figure 2.2: Node Classification Accuracies on the Cora, Citeseer and Pubmed Datasets for the Different Variants of the proposed Architecture GATE.

I first compare the four variants of the proposed architecture based on transductive node classification. Figure 2.2a shows the mean classification accuracy (with standard deviation) of all four variants on the test nodes after 100 runs of training (followed by logistic regression). Accordingly, I make the following observations:

- GATE outperforms other variants in all datasets. Therefore, each component contributes to the overall performance of my architecture.
- GATE/A performs worse than other variants. This suggests that the self-attention mechanism contributes the most in my architecture compared to the graph structure and node feature reconstructions.

- In Cora and Pubmed which have higher average node degree (i.e., 2 and 2.5 respectively), GATE/F outweighs the performance of GATE/S. On the other hand, GATE/S exceeds the performance of GATE/F in Citeseer which has the lowest average node degree (i.e., 1.4) and the highest number of features.

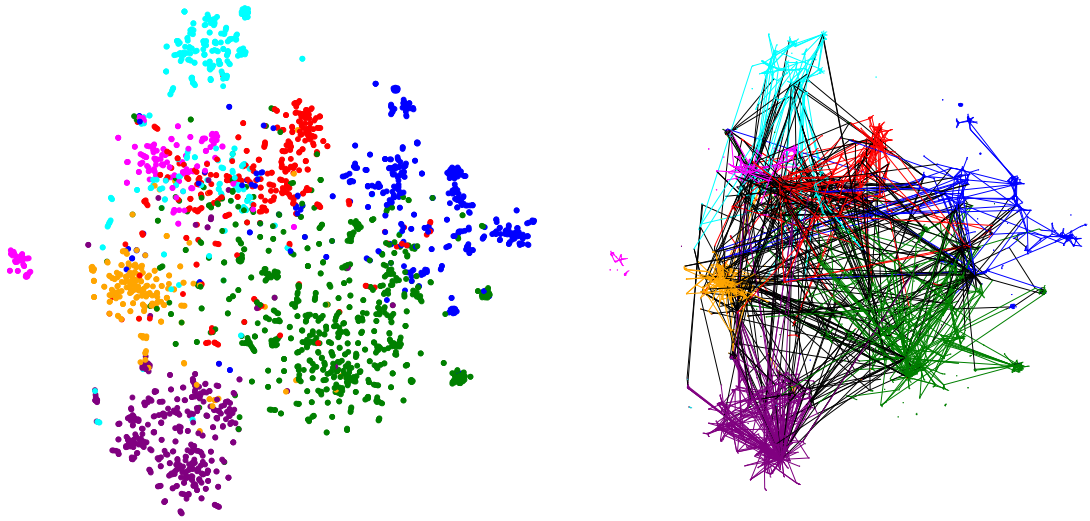
Now I compare all variants of my architecture based on inductive node classification. Figure 2.2b shows the mean classification accuracy (with standard deviation) of all four variants on the (unseen) test nodes after 100 runs of training (followed by logistic regression). Accordingly, I make the following observations:

- Like the transductive node classification experiments, GATE and GATE/A are respectively the best and the worst variants of my architecture in all datasets.

- I observe that the performances of GATE/F and GATE/S in Cora and Citeseer are similar to those of transductive node classification experiments. However, I notice a huge drop in the performance of GATE/F in Pubmed even though the performance of GATE/S has not undergone such a decrease. This can be attributed to both the low number of features and high average node degree of Pubmed compared to those of Cora and Citeseer, which hugely benefit GATE/F in transductive learning over inductive learning.

2.6.6 Qualitative Analysis

In this section, I qualitatively investigate the effectiveness of the node representations and attention coefficients learned by GATE. To this end, I utilize t-SNE (Maaten and Hinton, 2008) to project the learned node representations into a two-dimensional space. Due to space limitation, I only show the visualization for the Cora dataset. Figure 2.3a shows the t-SNE visualization of the learned node representa-



(a) The t-SNE visualization of nodes. (b) The t-SNE visualization of edges.

Figure 2.3: The t-SNE Visualizations of the Node Representations Learned by GATE on the Cora Dataset in Node and Edge perspectives.

Note: In Figure 2.3a, node colors denote classes. In Figure 2.3b, the edges with source and target nodes belonging to the same class are colored with the corresponding color of the class, and the others are colored black. Moreover, edge thickness indicates the averaged attention coefficients between node i and j across all layers (i.e., $\sum_{k=1}^{L}\left(\alpha_{ij}^{(k)}+\widehat{\alpha}_{ij}^{(k)}\right)/2L$)

tions for Cora, where node colors denote classes. I can observe that the learned node representations result in discernible clusters.

Figure 2.3b shows the t-SNE visualization of the edges, in the Cora dataset, thickened by their attention coefficients averaged across all layers. In this figure, the edges with source and target nodes belonging to the same class are colored with the color of the class, and the others are colored black. Accordingly, I expect high-quality node representations to result in thicker colorful edges. In Figure 2.3b, I can observe that the colorful edges are usually thicker than the black edges. However,

in few spots where GATE faces difficulty in separating nodes belonging to different classes, I can notice the presence of some thick black edges.

2.7 Conclusion

In this chapter, I introduced the graph attention auto-encoder (GATE), a novel neural architecture for unsupervised representation learning on graph-structured data. By stacking multiple encoder/decoder layers equipped with graph attention mechanisms, GATE is the first graph auto-encoder, which reconstructs both node features and the graph structure.

Experiments on both transductive and inductive tasks using three benchmark datasets demonstrate the efficacy of GATE, which learns high-quality node representations. In most experiments, the proposed auto-encoder outweighs state-of-the-art unsupervised baselines. Moreover, the experiments show that GATE naturally generalizes to unseen nodes.

Chapter 3

UTILIZING NODE ATTRIBUTES FOR COMMUNITY PROFILING

3.1 Overview

Web 2.0 helps to expand the range and depth of conversation on many issues and facilitates the formation of online communities. Online communities draw various individuals together based on their common opinions on a core set of issues. Most existing community detection methods merely focus on discovering communities without providing any insight regarding the collective opinions of community members and the motives behind the formation of communities. Several efforts have been made to tackle this problem by presenting a set of keywords as a community profile. However, they neglect the positions of community members towards keywords, which play an important role in understanding communities in the highly polarized atmosphere of social media. To this end, I present a sentiment-driven community profiling and detection framework which aims to discover community profiles presenting positive and negative collective opinions of community members separately. With this regard, the proposed framework initially extracts key expressions in users' messages as representative of issues and then identifies users' positive/negative attitudes towards these key expressions. Next, it uncovers a low-dimensional latent space in order to cluster users according to their opinions and social interactions (i.e., retweets). I demonstrate the effectiveness of the proposed framework through quantitative and qualitative evaluations.

3.2 Introduction

With the advent of social media platforms, individuals are able to express their opinions on a variety of issues online. Like-minded users forge online communities by interacting with each other and expressing similar attitudes towards a set of issues. While many methods (Papadopoulos *et al.*, 2012) have been proposed to detect online communities, most of them do not provide insights into the collective opinions of community members. To shed light on such opinions, few efforts have focused on profiling communities, but a large body of work has been devoted to user profiling (Mislove *et al.*, 2010; Harvey *et al.*, 2013; Ikeda *et al.*, 2013). Indeed, "the founders of sociology claimed that the causes of social phenomena were to be found by studying groups rather than individuals" (Hechter, 1988).

Turner et al. (Turner *et al.*, 1987) suggest that individuals come together and form communities by developing a shared social categorization of themselves in contrast to others . Therefore, to profile a community, we need to uncover the collective opinions of its members which make them distinguishable from the members of other communities. Tajfel (Tajfel, 2010) suggests focusing on unit-forming factors (e.g., similarities, shared threats, or common fate) which function as cognitive criteria for segmentation of the social world into discrete categories. Accordingly, the controversial issues on which users have different opinions can be taken into account in order to discover the motives driving the segmentation of social media and the formation of communities. As a result, the profile of a community should present its important issues on which its members generally have the same position. Such community profiles can be found useful in a broad range of applications such as recommender systems (Sahebi and Cohen, 2011), community ranking (Chen *et al.*, 2008; Han *et al.*, 2016), online marketing (Kozinets, 2002), interest shift tracking of communities (Zhou *et al.*,

2012), and community visualization (Cruz *et al.*, 2013). For example, a group recommender system (Boratto, 2016) can suggest more relevant items to communities by knowing the collective opinions of their members.

Many community detection methods (Cai *et al.*, 2017; Zhou *et al.*, 2012; Akbari and Chua, 2017; Natarajan *et al.*, 2013; Ozer *et al.*, 2016; Pathak *et al.*, 2008; Pei *et al.*, 2015; Sachan *et al.*, 2012; Zhou *et al.*, 2006) which are capable of community profiling have been proposed. However, these methods usually present a set of frequent keywords used by the members of a community as the community profile. However, it is common in social media that the members of different communities use the same keywords in their messages. Therefore, keywords alone might not be enough to differentiate communities in which their members have similar word usage. For instance, in the course of the US presidential election of 2016, Republicans and Democrats have used many common keywords such as Trump, Clinton, and Obamacare but with different sentiments. To differentiate and understand these two parties, not only keywords but also the collective attitude of community members towards these keywords should be taken into account.

In this chapter, I tackle the aforementioned problem by proposing a sentiment-driven community profiling and detection framework which utilizes user-generated content and social interactions. The proposed framework first captures key expressions in users' messages as representative of issues by utilizing a POS-tagger and built-in features of social media platforms (i.e. hashtags and user accounts). Next, it identifies users' attitudes towards the extracted key expressions. Finally, I employ a novel graph regularized semi-nonnegative matrix factorization (GSNMF) technique to cluster users according to both their opinions and social interactions. GSNMF uncovers not only communities but also their sentiment-driven profiles. The main contributions of the chapter are as follows:

- Providing sentiment-driven community profiles which separately present the positive and negative collective attitudes of the members of each community towards their important key expressions;
- Achieving higher performance in detecting communities compared to several existing state-of-the-art community detection methods.

The rest of the chapter is organized as follows. I review related work in Section 3.3. I also formally define the problem in Section 3.4. In Section 3.5, I propose the sentiment-driven community profiling and detection framework. To demonstrate the efficacy of my framework, I conduct quantitative and qualitative experiments by using real-world social media datasets in Section 3.6. Section 3.7 concludes the paper and discusses future work.

3.3 Related Work

Community detection methods can fall into three broad categories: link-based, content-based and hybrid methods. Most of the existing works belong to the first category and utilize only social interactions (Clauset *et al.*, 2004; Blondel *et al.*, 2008). However, they neglect to utilize valuable user-generated content in which users express their opinions. On the other hand, content-based methods only utilize user-generated content (Lee *et al.*, 2013). Nevertheless, the content on social media is extremely noisy, resulting in the failure in detecting communities effectively. To alleviate these challenges, hybrid community detection methods are proposed. These methods are the most related work to my study since they not only exploit both user-generated content and social interactions but are also capable of profiling communities. These methods roughly fall into two categories: probabilistic graphical models and non-negative matrix factorization (NMF) based methods.

3.3.1 Probabilistic Graphical Models

Community User Topic (CUT) models (Zhou *et al.*, 2006) are one of the earliest works for detecting communities using probabilistic graphical models. The first proposed model (CUT_1) assumes that a community is a distribution over users, while the second one (CUT_2) considers a community as a distribution over topics. To discover communities, CUT_1 and CUT_2 are biased towards social interactions and user-generated content, respectively. Community Author Recipient Topic (CART) (Pathak *et al.*, 2008) is an unbiased model which assumes the members of a community discuss topics of mutual interests and interact with one other based on these topics. CART considers users as both authors and recipients of a message. However, in well-known social networks such as Twitter and Facebook, the number of recipients for a message can be very large. To make community detection scalable, Topic User Community Model (TUCM) (Sachan *et al.*, 2012), considering users as authors not recipients, is proposed. Since CART and TUCM consider users as authors, recipients, or both, they are limited to certain types of social interactions (e.g., retweet and reply-to in Twitter). The link-content model (Natarajan *et al.*, 2013) solves this problem by ignoring the assumption that messages can be related to each other using social interactions. It is also capable of using different types of social interactions (e.g., friendship in Facebook and followership in Twitter). Furthermore, COCOMP (Zhou *et al.*, 2012) is proposed to model each community as a mixture of topics about which a corresponding group of users communicate. (Cai *et al.*, 2017) is another model which detects and profiles communities in the domains having user-user, user-document, and document-document links.

3.3.2 NMF-based Methods

In order to encode graphs as local geometric structures, many methods extending standard NMF are proposed. LLNMF (Gu and Zhou, 2009) introduces a regularizer, imposing the constraint that each data point should be clustered based on the labels of the data points in its neighborhood. GNMF (Cai *et al.*, 2011) further incorporates a graph regularizer to encode the manifold structure. Moreover, DNMF (Shang *et al.*, 2012) is proposed based on the the idea that not only the data, but also the features lie on a manifold. The graph regularizers proposed by the above methods have been utilized by several other works (Pei *et al.*, 2015; Ozer *et al.*, 2016) to detect communities on social media. Moreover, another work (Akbari and Chua, 2017) proposes an NMF-based approach utilizing a graph regularizer to exploit different social views (i.e., different social interactions and user-generated content) as well as prior knowledge in order to detect and profile communities.

3.4 Problem Statement

I first begin with the introduction of the notations used in the paper as summarized in Table 3.1. Let $\mathcal{U} = \{u_1, u_2, ..., u_n\}$ be the set of n users, $\mathcal{C} = \{c_1, c_2, ..., c_k\}$ indicate the set of k communities, and $\mathcal{S} = \{s_1, s_2, ..., s_k\}$ denote the set of m key expressions. $\mathbf{X} \in \mathbb{R}^{m \times n}$ indicates the matrix of users' attitudes towards key expressions, where $\mathbf{X}_{li}$ corresponds to the attitude of user u_i towards key expression s_l. Furthermore, $\mathbf{U} \in \mathbb{R}_+^{n \times k}$ indicates the community membership matrix, in which $\mathbf{U}_{ik}$ corresponds to the membership strength of user $\mathbf{U}_i$ in community c_k. $\mathbf{V} \in \mathbb{R}^{m \times k}$ further denotes the community profile matrix, where $\mathbf{V}_{lk}$ corresponds to the contribution strength of key expression s_l in the profile of community c_k. Moreover, $\mathbf{W} \in \mathbb{R}_+^{n \times n}$ indicates the social interaction matrix, in which $\mathbf{W}_{ij}$ represents the number of social interactions

Table 3.1: The Notations Used in Chapter 3.

Notation	Explanation
$\mathcal{U}$	The set of users
$\mathcal{C}$	The set of communities
$\mathcal{S}$	The set of key expressions
n	The number of users
m	The number of key expressions
k	The number of communities
$\mathbf{X}$	User opinion matrix
$\mathbf{U}$	Community membership matrix
$\mathbf{V}$	Community profile matrix
$\mathbf{W}$	Social Interaction matrix
$\widetilde{\mathbf{W}}$	Symmetrically normalized matrix $\mathbf{W}$
$\mathbf{D}$	Degree matrix of $\mathbf{W}$

between user u_i and user u_j. I use $\widetilde{\mathbf{W}}$ to denote the symmetric normalization of $\mathbf{W}$ (i.e., $\widetilde{\mathbf{W}} = \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$, where $\mathbf{D}$ is the degree matrix of $\mathbf{W}$).

By using the above notations, the problem of detecting and profiling communities can be defined as: *Given an attributed graph in which node features and the graph structure are represented by user opinion matrix* $\mathbf{X}$ *and social interaction matrix* $\mathbf{W}$ *respectively, I aim to obtain community membership matrix* $\mathbf{U}$ *and community profile matrix* $\mathbf{V}$.

3.5 The Proposed Framework

3.5.1 Extracting Key Expressions as Issues

Social media presents an opportunity to utilize user-generated content in which individuals express their opinions on various issues. The first step towards understanding users' opinions is the extraction of the issues they discuss. To this end, many efforts (Qiu *et al.*, 2011; Mukherjee and Liu, 2012; Zhang and Liu, 2014) have been made to extract issues or related aspects. However, these methods require enough training samples for a specific domain to work accurately. Due to the lack of such dataset for the required experiments, I follow a simple approach to extract key expressions. I utilize the built-in features common among well-known social media platforms. In such social networks, hashtags and user account mentions, which usually indicate issues, are perpended by '#' and '@', respectively. However, the built-in features are not enough to detect all issues. To tackle this problem, I employ a part-of-speech (POS) tagger to extract proper nouns and noun phrases (two or more nouns in a row) as representative of issues. If some proper nouns are in a row, they are considered as a single key expression. I utilize the POS tagger proposed in (Gimpel *et al.*, 2011) proven to perform well for the content on social media.

3.5.2 Capturing Users' Opinions

The position individuals take towards issues reflects their opinions [1] . Many efforts (Pontiki *et al.*, 2016; Tang *et al.*, 2016) have been made to detect users' sentiments towards issues. However, these methods work effectively when enough training samples for a specific domain are given. However, there is no such a dataset for the required experiments so I apply a simple approach although a sophisticated approach

[1] An opinion is defined as an attitude towards an issue (Fishbein and Ajzen, 1977).

can improve the result of the proposed framework. First, a window with a certain size centered at each positive/negative sentiment word is created. Next, the nearest key expression to the sentiment word is selected, and the positivity/negativity of the sentiment word determines the user's positive/negative attitude towards that key expression. For instance, in the message "Conservatives seem angry every time economy adds jobs", I assume the author has a negative sentiment towards key expression "conservatives" because it is the closest key expression to the negative sentiment word "angry" if I consider the window size to be at least two. To generate matrix $\mathbf{X}$, I need to apply the above procedure for all messages. Therefore, for each message if author u_i takes a positive/negative attitude towards key expression s_l, I add the sentiment strength of the corresponding sentiment word to $\mathbf{X}_{li}$, respectively. I utilize SentiStrength (Thelwall *et al.*, 2010) to discover positive and negative words as well as their sentiment strength.

3.5.3 Modeling Users' Opinions

After extracting users' attitudes towards key expressions, the next major objective is sentiment-driven community profiling and detection of like-minded users. To accomplish this, I exploit semi-nonnegative matrix factorization (Ding *et al.*, 2010) as follows:

$$\begin{aligned} \min_{U,V} \quad & ||\mathbf{X} - \mathbf{V}\mathbf{U}^T||_F^2 \\ \text{s.t.} \quad & \mathbf{U} \geq 0. \end{aligned} \tag{3.1}$$

Since the non-negativity constraint in Eq. (3.1) only holds on matrix $\mathbf{U}$, matrix $\mathbf{V}$ can contain both positive and negative values. A positive/negative value of $\mathbf{V}_{lk}$ denotes that the members of community c_k have a collective positive/negative attitude towards key expression s_l. The larger the positive value of $\mathbf{V}_{lk}$ is, the more the members of community c_k have a collective positive attitude towards key expression

s_l. The lower the negative value of $\mathbf{V}_{lk}$ is, the more the members of community c_k have a collective negative attitude towards key expression s_l. This property of matrix $\mathbf{V}$ also results in the categorization of key expressions into positive and negative categories according to the sign of the corresponding elements of key expressions in matrix $\mathbf{V}$. Therefore, key expressions in a community profile are divided into two positive and negative categories. Moreover, the key expressions in each category can also be ranked by their values in matrix $\mathbf{V}$ in order to show how important they are to the members of the corresponding community.

3.5.4 Modeling Social Interactions

Social interactions (e.g., retweets in Twitter and friendships in Facebook) are one of the most effective sources of information to detect communities (Papadopoulos *et al.*, 2012). To utilize social interactions, NMF-based methods exploit graph regularizers. Gu *et al.* (Gu *et al.*, 2011) suggest that graph regularizers used in GNMF (Cai *et al.*, 2011) and DNMF (Shang *et al.*, 2012) suffer from the trivial solution problem and the scale transfer problem. When the graph regularizer parameter is too large, the trivial solution problem occurs and results in similarity among the elements of each row of community membership matrix $\mathbf{U}$. The scale transfer problem, in which $\{\mathbf{V}^*, \mathbf{U}^*\}$ stands as the optimal solution for Eq. (3.1), results in a smaller objective value for the scaled transferred solution $(\frac{\mathbf{V}_*}{\beta}, \beta\mathbf{U}')$, for any real scalar $\beta > 1$.

To avoid these problems, I propose using the following graph regularizer,

$$\begin{aligned} \max_{U} \quad & Tr(\mathbf{U}^T\widetilde{\mathbf{W}}\mathbf{U}) \\ \text{s.t.} \quad & \mathbf{U} \geq 0, \mathbf{U}^T\mathbf{U} = \mathbf{I}. \end{aligned} \tag{3.2}$$

where $\mathbf{I}$ is the identity matrix with the proper size. Eq. (3.2) clusters users into k communities, with the most interactions within each community and the fewest

interactions between communities. In fact, Eq. (3.2) is equivalent to the nonnegative relaxed normalized cut as put forth in (Ding *et al.*, 2005).

3.5.5 The Proposed Framework GSNMF

In the previous sections, I introduced my solutions to exploit and social interactions and users' attitudes toward key expressions. Using these solutions, the proposed framework simultaneously utilizes users' opinions and social interactions to uncover communities and their profiles. The proposed framework requires solving the following optimization problem,

$$\begin{aligned} \min_{U,V} \quad & \mathcal{F} = ||\mathbf{X} - \mathbf{V}\mathbf{U}^T||_F^2 - \lambda Tr(\mathbf{U}^T\widetilde{\mathbf{W}}\mathbf{U}) \\ \text{s.t.} \quad & \mathbf{U} \geq 0, \mathbf{U}^T\mathbf{U} = \mathbf{I}. \end{aligned} \tag{3.3}$$

where λ is a non-negative regularization parameter controlling the contribution of the graph regularizer in the final solution. Since the optimization problem in Eq. (3.3) is not convex with respect to variables $\mathbf{U}$ and $\mathbf{V}$ together, there is no guarantee to find the global optimal solution. As suggested by (Lee and Seung, 2001), I introduce an alternative scheme to find a local optimal solution to the optimization problem. The key idea is optimizing the objective function with respect to one of the variables $\mathbf{U}$ or $\mathbf{V}$, while fixing the other one. The algorithm keeps updating the variables until convergence.

Computation of U

Optimizing the objective function $\mathcal{F}$ in Eq. (3.3) with respect to $\mathbf{U}$ is equivalent to solving

$$\begin{aligned} \min_{\mathbf{U}} \quad & \mathcal{F}_U = ||\mathbf{X} - \mathbf{V}\mathbf{U}^T||_F^2 - \lambda Tr(\mathbf{U}^T\widetilde{\mathbf{W}}\mathbf{U}) \\ \text{s.t.} \quad & \mathbf{U} \geq 0, \mathbf{U}^T\mathbf{U} = \mathbf{I}. \end{aligned} \tag{3.4}$$

Let $\mathbf{\Gamma}$ and $\mathbf{\Lambda}$ be the Lagrange multiplier for constraints $\mathbf{U}^T\mathbf{U} = \mathbf{I}$ and $\mathbf{U} \geq 0$ respectively, and the Lagrange function is defined as follows:

$$\begin{aligned} \min_{\mathbf{U}} \quad \mathcal{L}_{\mathbf{U}} = & ||\mathbf{X} - \mathbf{V}\mathbf{U}^T||_F^2 - \lambda Tr(\mathbf{U}^T\widetilde{\mathbf{W}}\mathbf{U}) \\ & - Tr(\mathbf{\Lambda}\mathbf{U}^T) + Tr(\mathbf{\Gamma}(\mathbf{U}^T\mathbf{U} - \mathbf{I})) \end{aligned} \tag{3.5}$$

The derivative of $\mathcal{L}_{\mathbf{U}}$ with respect to $\mathbf{U}$ is

$$\frac{\partial \mathcal{L}_{\mathbf{U}}}{\partial \mathbf{U}} = -2\mathbf{X}^T\mathbf{V} + 2\mathbf{U}\mathbf{V}^T\mathbf{V} - 2\lambda\widetilde{\mathbf{W}}\mathbf{U} - \mathbf{\Lambda} + 2\mathbf{U}\mathbf{\Gamma} \tag{3.6}$$

By setting $\frac{\partial \mathcal{L}_{\mathbf{U}}}{\partial \mathbf{U}} = 0$, I get

$$\mathbf{\Lambda} = -2\mathbf{X}^T\mathbf{V} + 2\mathbf{U}\mathbf{V}^T\mathbf{V} - 2\lambda\widetilde{\mathbf{W}}\mathbf{U} + 2\mathbf{U}\mathbf{\Gamma} \tag{3.7}$$

With the KKT complementary condition for the nonnegativity of $\mathbf{U}$, I have $\mathbf{\Lambda}_{ij}\mathbf{U}_{ij} = 0$. Therefore, I have

$$(-\mathbf{X}^T\mathbf{V} + \mathbf{U}\mathbf{V}^T\mathbf{V} - \lambda\widetilde{\mathbf{W}}\mathbf{U} + \mathbf{U}\mathbf{\Gamma})_{ij}\mathbf{U}_{ij} = 0 \tag{3.8}$$

where $\mathbf{\Gamma} = \mathbf{U}^T\mathbf{X}^T\mathbf{V} - \mathbf{V}^T\mathbf{V} + \lambda\mathbf{U}^T\widetilde{\mathbf{W}}\mathbf{U}$.

Matrices $\mathbf{\Gamma}$, $\mathbf{X}^T\mathbf{V}$, and $\mathbf{V}^T\mathbf{V}$ take mixed signs. Motivated by (Ding *et al.*, 2010), I separate positive and negative parts of any matrix $\mathbf{A}$ as $\mathbf{A}_{ij}^+ = (|\mathbf{A}_{ij}| + \mathbf{A}_{ij})/2, \quad \mathbf{A}_{ij}^- = (|\mathbf{A}_{ij}| - \mathbf{A}_{ij})/2$.

Thus, I get

$$\begin{aligned} & [-((\mathbf{X}^T\mathbf{V})^+ + [\mathbf{U}(\mathbf{V}^T\mathbf{V})^-] + \lambda\widetilde{\mathbf{W}}\mathbf{U} + \mathbf{U}\mathbf{\Gamma}^-) \\ & +((\mathbf{X}^T\mathbf{V})^- + [\mathbf{U}(\mathbf{V}^T\mathbf{V})^+] + \mathbf{U}\mathbf{\Gamma}^+)]_{ij}\mathbf{U}_{ij} = 0 \end{aligned} \tag{3.9}$$

Therefore, optimizing the objective function $\mathcal{F}$ with respect to $\mathbf{U}$ leads to the following update rule,

Algorithm 2 The Proposed Algorithm for GSNMF

Input: user opinion matrix $\mathbf{X}$ and social interaction matrix $\mathbf{W}$

output: community membership matrix $\mathbf{U}$ and community profile matrix $\mathbf{V}$

1: Initialize $\mathbf{U}$ and $\mathbf{V}$ randomly where $\mathbf{U} \geq 0$

2: **while** not convergent **do**

3: Update $\mathbf{U}$ according to Eq. (3.10)

4: Update $\mathbf{V}$ according to Eq. (3.13)

5: **end while**

$$\mathbf{U} = \mathbf{U} \odot \sqrt{\frac{(\mathbf{X}^T\mathbf{V})^+ + [\mathbf{U}(\mathbf{V}^T\mathbf{V})^-] + \lambda\widetilde{\mathbf{W}}\mathbf{U} + \mathbf{U}\mathbf{\Gamma}^-}{(\mathbf{X}^T\mathbf{V})^- + [\mathbf{U}(\mathbf{V}^T\mathbf{V})^+] + \mathbf{U}\mathbf{\Gamma}^+}} \tag{3.10}$$

where $\odot$ denotes the Hadamard product.

Computation of V

Optimizing the objective function $\mathcal{F}$ in Eq. (3.3) with respect to $\mathbf{V}$ is equivalent to solving

$$\min_{\mathbf{V}} \quad \mathcal{F}_{\mathbf{V}} = ||\mathbf{X} - \mathbf{V}\mathbf{U}^T||_F^2 \tag{3.11}$$

The derivative of $\mathcal{F}_{\mathbf{V}}$ with respect to $\mathbf{V}$ is

$$\frac{\partial \mathcal{F}_{\mathbf{V}}}{\partial \mathbf{V}} = -2\mathbf{X}\mathbf{U} + 2\mathbf{V}\mathbf{U}^T\mathbf{U} \tag{3.12}$$

By setting $\frac{\partial \mathcal{F}_{\mathbf{V}}}{\partial \mathbf{V}} = 0$, I compute the updating rule of $\mathbf{V}$ as follows:

$$\mathbf{V} = \mathbf{X}\mathbf{U}(\mathbf{U}^T\mathbf{U})^{-1} \tag{3.13}$$

3.5.6 Algorithm Complexity

The algorithm for GSNMF is shown in Algorithm 2. In line 1, it randomly initializes $\mathbf{U}$ and $\mathbf{V}$. From lines 2 to 5, it updates $\mathbf{U}$ and $\mathbf{V}$ until convergence is achieved. In Algorithm 2, the most costly operations are the matrix multiplications in update rules Eq. (3.10) and Eq. (3.13). Therefore, I provide the time complexity of these two updating rules as follows:

- The time complexity of Eq. (3.10) is $O(nmk + mk^2 + n^2k + nk^2)$.
- Since the inversion of small matrix $\mathbf{U}^T\mathbf{U}$ is trivial, the time complexity of Eq. (3.13) is $O(mnk + nk^2)$.

Accordingly, the time complexity of Algorithm 2 is $O(ik(nm + mk + n^2 + nk))$ where i is the number of iterations. The proposed framework can be applied to large scale social network platforms by exploiting the distributed approaches outlined in (Liu *et al.*, 2010; Gemulla *et al.*, 2011; Li *et al.*, 2014).

3.6 Experiments

To evaluate the efficacy of my framework, I need to answer the following two questions:

1. How effective is my framework in detecting communities compared to the the-state-of-the-art community detection methods?
2. How effective is my framework in profiling communities according to the collective opinions of community members?

In the next sections, I first describe the datasets used in this study. Next, the performance of GSNMF is compared with several state-of-the-art community detec-

tion methods. Then, I qualitatively evaluate the community profiles uncovered by my framework.

3.6.1 Data Description

I take politics as an example to evaluate my framework. In this regard, I used the Twitter search API to crawl politicians' tweets from three different countries, namely the United States, United Kingdom, and Canada. However, Twitter API imposes the limitation of retrieving only the latest 3200 tweets for each user. To overcome this limitation, I crawled politicians' user accounts several times during the time each dataset covers. The datasets are described as follows,

- **US Dataset** consists of the tweets posted by 404 politicians from two major political parties (Republican party and Democratic party) in the United States from August 26 to November 29, 2016.
- **UK Dataset** consists of the tweets posted by 317 political figures from five major political parties (Conservative Party, Labour Party, Scottish National Party, Liberal Democratic Party, and UK Independence Party) in the United Kingdom from January 1 to September 30, 2015.
- **Canada Dataset** consists of the tweets posted by 102 politicians from three major political parties (Liberal Party, Conservative Party, and New Democratic Party) from January 1 to November 18, 2016.

All users in the datasets have discussed at least 15 key expressions. Moreover, the key expressions used by less than 15 users and stop words are eliminated. As a window size, I experimentally determine the threshold of 3 for the nearest keywords on both sides of each sentiment word. Furthermore, the party to which a user belongs is labeled as ground truth. The statistics for the datasets are shown in Table 3.2.

Table 3.2: The Statistics of the Datasets Used in Chapter 3.

	US	UK	Canada
# of tweets	113,818	236,008	98,899
# of retweets	18,891	6,863	3,104
# of distinct words	5,773	7,653	3,738
# of distinct key expressions	165	349	69
# of users	404	317	102
# of baseline communities	2	5	3

The GSNMF code and users' Twitter accounts as well as their ground truth labels used in this paper are available [2] .

3.6.2 Community Detection Evaluation

Baselines

In order to demonstrate the effectiveness of my framework, I compare GSNMF with the following state-of-the-art community detection methods,

- **GNMF** (Cai *et al.*, 2011) is a hybrid method utilizing both user-generated and social interactions by incorporating a graph regularizer into standard NMF.
- **Louvain** (Blondel *et al.*, 2008) is a link-based method optimizing modularity using a greedy approach.
- **Infomap** (Rosvall and Bergstrom, 2008) is a link-based method built upon information theory to compress the description of random walks in order to find community structure.

[2] `https://github.com/amin-salehi/GSNMF`

- **DNMF** (Shang *et al.*, 2012) is a hybrid method utilizing both user-generated content and social interactions by incorporating two regularizers (i.e, a graph regularizer and a word similarity regularizer) into standard NMF.

- **Soft Clustering** (Yu *et al.*, 2005) is a link-based method that assigns users to communities in a probabilistic way.

- **CNM** (Clauset *et al.*, 2004) is a link-based method based on modularity optimization.

Evaluation Metrics

To evaluate the performance of the methods, I utilize three metrics frequently used for community detection evaluation; namely, Normalized Mutual Information (NMI), Adjusted Rand Index (ARI) and purity.

Experimental Results

For this experiment, I use all three datasets. I also utilize the party membership of each politician as ground truth in the evaluation. For the methods providing soft community membership, like my framework, I select the community with the highest membership value for each user as the community to which she/he belongs. Regularization parameters of NMF-based methods are set to be all powers of 10 from 0 to 9 to find the best configuration for each of these methods. I run each method 10 times with its best configuration and then report the best result. According to the results shown in Table 3.3, I can make the following observations,

- The proposed framework achieves the highest performance in terms of NMI and ARI for all three datasets. In terms of purity, it also achieves the best in the Canada and US datasets. In the UK dataset, Louvain, Infomap, and

Table 3.3: Performance Comparison of Community Detection Methods in Chapter 3.

	US			UK			Canada		
Method	NMI	ARI	Purity	NMI	ARI	Purity	NMI	ARI	Purity
Louvain	0.5083	0.3889	0.9752	0.7077	0.4352	**0.9937**	0.8602	0.8430	0.9902
Infomap	0.5026	0.3755	0.9752	0.8871	0.8874	0.9936	0.8971	0.9299	0.9804
CNM	0.5741	0.4664	0.9752	0.8830	0.8746	0.9905	0.9405	0.9643	0.9902
GNMF	0.8564	0.9126	0.9777	0.8120	0.8291	0.9085	0.9597	0.9794	0.9902
DNMF	0.8599	0.9222	0.9802	0.8308	0.8030	0.8896	0.9574	0.9716	0.9902
Soft Clustering	0.8934	0.9413	0.9851	0.8481	0.8450	0.9495	**1.0000**	**1.0000**	**1.0000**
GSNMF	**0.9069**	**0.9510**	**0.9876**	**0.9298**	**0.9612**	0.9811	**1.0000**	**1.0000**	**1.0000**

CNM obtain higher purity compared to my framework since they generate an artificially large number of communities for sparse graphs such as social media networks. For instance, Louvain detects 21 communities for UK dataset.

- Exploiting both user-generated content and social interactions does not necessarily result in achieving better performance compared to link-based methods. For example, the Soft Clustering method achieves better results compared to GNMF and DNMF in terms of all three used metrics. However, link-based methods do not uncover any community profile.

- All NMF-based methods achieve their highest performance with large values (i.e., from 10^6 to 10^9) for the graph regularizer parameter.

3.6.3 Community Profiling Evaluation

In this section, I evaluate the effectiveness of the proposed framework in profiling communities by using the US and UK datasets. In this regard, I first label each community detected by my framework with the party to which the majority of community members belong. Next, I evaluate how effectively the profile of a community represents its corresponding ground truth party. To this end, two graduate students who

have knowledge of US and UK politics are assigned to label the results of community profiling methods. It is asked that each key expression in a community profile to be assigned to one of the following categories:

- Supported: A key expression is labeled as supported if the majority of community members have a positive attitude towards it or support it.
- Opposed: A key expression is labeled as opposed if the majority of community members have a negative attitude towards it or oppose it.
- Concerned: A key expression is labeled as concerned if the majority of community members are concerned about it.
- Unrelated: A key expression is labeled as unrelated if the annotators cannot find a strong relevance between the community (party) and the key expression.

In the tables representing community profiles, I color (and mark) supported, opposed, and concerned key expressions with green (+), red (−), and blue (±), respectively. I also leave unrelated key expressions uncolored (and unmarked).

In the following experiments, I expect the proposed framework to achieve three goals:

1. Uncovering community profiles which represent the collective opinions of community members into two positive/negative categories;
2. Assigning supported key expressions and opposed/concerned ones to positive/negative categories, respectively;
3. Minimizing the number of unrelated key expressions in community profiles.

In Sections 3.6.3 and 3.6.3, I evaluate the results of GSNMF according to the first and second goals by using US and UK datasets. To evaluate the performance of the

third goal, Section 3.6.3 compares GSNMF with the baselines with regard to their effectiveness in extracting relevant key expressions.

US Politics

The US dataset covers many events such as occurrences of gun violence, police brutality (e.g., the shooting of Terence Crutcher), the Flint water crisis, and the death of Fidel Castro; but the major event is the US presidential election of 2016. To give brief background knowledge, two major US parties during the election are described as follows (Lilleker *et al.*, 2016),

- **Democratic Party:** A liberal party focusing on social justice issues. In 2016, Hillary Clinton was nominated as the presidential candidate of the party with Tim Kaine as her vice president. Moreover, Barrack Obama, the incumbent Democratic President, was a strong advocate for Hillary Clinton.

- **Republican Party:** A conservative party, known as the GOP, which had the majority of congressional seats in 2016 and embraces Judeo-Christian ethics. Moreover, Donald Trump was nominated as the party candidate for the presidency with Mike Pence as his vice president.

During the campaign, Republicans—especially Donald Trump—mainly criticized President Obama and his policies (e.g., Obamacare, tax plans, and Iran deal) in order to discredit Hillary Clinton, whom they claimed was going to continue the Obama legacy and uphold the status quo (Lilleker *et al.*, 2016). On the other hand, Clinton's campaign brought the issue of gun violence into the contest, and also focused on human rights for groups such as women and LGBTQ (Lilleker *et al.*, 2016).

Table 3.4 shows the profiles of two communities detected by my framework in the US dataset as well as their corresponding ground truth political parties and experts'

Table 3.4: The Profiles of Two Communities Detected by GSNMF in the US Dataset.

Democrats				Republicans			
Positive		Negative		Positive		Negative	
+	HillaryClinton	±	Zika	+	America	−	Obamacare
+	POTUS	−	Trump	+	@SpeakerRyan	±	#BetterWay
+	America	−	@HouseGOP	+	Congress	±	Zika
+	#WomensEqualityDay	−	Donald Trump	+	@Mike_Pence	−	Iran
+	#NationalComingOutDay	−	Gun Violence	+	@RepTomPrice	−	Obama
+	Americans	±	#Trans	+	@realDonaldTrump	−	Tax code
+	#LaborDay	−	#GunViolence	+	Texas	±	Breast Cancer
+	TimKaine	±	Climate Change	+	#VeteransDay	−	President Obama
+	Hillary	±	#Trabajadores		ICYMI	±	GITMO
+	American	±	TerenceCrutcher		Senator	−	Islamic
	Cubs	−	GOP	+	#LaborDay	−	State Sponsor
+	Halloween	−	Violence Situations	+	God	−	POTUS
+	Veterans	−	ISIS	+	Constitution Day	−	ISIS
	Florida	±	#FundFlint	+	USMC	−	Hillary
+	#LGBTQ equality	−	Donald	+	Thanksgiving	−	Fidel Castro

Note: All colors, signs, and the name of parties in the table are ground truth.

labels. According to the provided background, the community on the left highly resembles the Democratic Party since its members have generally expressed: (1) positive attitudes towards Hillary Clinton, the U.S. president (i.e., POTUS), Tim Kaine, and human rights issues (e.g., #WomensEqualityDay, #LGBTQ equality, and #NationalComingOutDay), and (2) negative attitudes towards the Republican Party (e.g., @HouseGOP and GOP), Donald Trump, and gun violence, police brutality (e.g., the shooting of Terence Crutcher). On the other hand, the community on the right highly resembles the Republican Party since its members have generally expressed: (1) positive attitudes towards the Republican Party (e.g., @HouseGOP and @SpeakerRyan), Donald Trump, Mike Pence, Congress, and God, and (2) negative attitudes towards President Obama and his policies (i.e., Obamacare, tax code, Iran, Guantanamo Bay detention camp (i.e., GITMO)) as well as Hillary Clinton.

Table 3.5: The Profiles of Two Communities Detected by GNMF and DNMF in the US Dataset.

a. GNMF

Democrats		Republicans	
−	Trump	−	Obamacare
+	Hillary	±	#BetterWay
+	Gov	+	Congress
+	HillaryClinton	±	Zika
−	Donald Trump	+	America
±	#DoYourJob	+	@HouseGOP
	DebateNight	+	American
−	@realDonaldTrump		ICYMI
	China	−	Obama
−	@CoryBooker		Florida
±	Russia	+	Americans
	ElectionDay	−	Iran
±	Climate Change	+	U.S.
	Debate	±	#HurricanMatthew
+	Hillary Clinton	−	POTUS
−	Donald	+	@realDonaldTrump
	Virginia	−	Clinton
+	HRC	±	Hurrican Matthew
	VPDebate	−	Washington
+	America	±	FBI
+	TimKaine	+	Texas
+	#WomenEqualityDay	+	Senate
+	FLOTUS	+	Veterans
+	#IamWithHer	±	Matthew
±	Flint		#DoYourJob
+	POTUS	+	@SpeakerRyan
+	HouseDemocrats		Ohio
−	Steve Bannon	±	#NeverForget
−	Bannon	+	GOP
+	USA		WSJ

b. DNMF

Democrats		Republicans	
−	Congress	+	Congress
+	Obamacare	−	Obamacare
−	Trump	+	Trump
	#BetterWay	±	#BetterWay
+	America	+	America
±	Zika	±	Zika
−	@HouseGOP	+	@HouseGOP
+	American	+	American
+	Gov	−	Gov
±	#DoYourJob		#DoYourJob
	ICYMI		ICYMI
+	Americans	−	HillaryClinton
+	HillaryClinton	+	Americans
+	Hillary	−	Hillary
−	@realDonaldTrump	+	@realDonaldTrump
+	Obama	−	Obama
+	U.S.	+	U.S.
+	POTUS	−	POTUS
+	Iran	−	Iran
+	Clinton	−	Clinton
−	Donald Trump	+	Donald Trump
+	Veterans	+	Veterans
+	Washington	−	Washington
±	HurricanMatthew	±	HurricanMatthew
−	Senate	+	Senate
±	FBI	±	FBI
	Florida		Florida
	Texas	+	Texas
−	GOP	+	GOP
	Oct		Oct

Negative sentiment implies both opposition and concern. If necessary, my framework can differentiate opposition from concern by providing the sentiment words frequently expressed by the members of a community towards each key expression. For example, Democrats' negative sentiment towards Donald Trump mainly comes from the sentiment words "unfit", "low", and "dangerous" which suggest opposition. On the other hand, their negative sentiment towards #Trans (i.e., transgender people) mainly originates from the sentiment words "discrimination" and "murder" which

indicate concern.

To demonstrate the advantage of my community profiling method, I compare the profiles of typical community profiles usually provided by retrospective studies with those uncovered by my framework. Table 3.5 shows the profiles of two communities detected by GNMF and DNMF in the US dataset as well as their corresponding ground truth political parties. As I observe, it is almost impossible for a non-expert individual to recognize the party associated with each profile since the position of the communities towards the key expressions are not taken into account. For example, in profiles corresponding to the Democratic Party and the Republican Party, many key expressions related to Trump, Clinton, and Obama exist, but there is no information regarding collective attitude of community members toward such key expressions. However, Table 3.4 shows that the proposed method correctly divides opposed/concerned key expressions and supported ones into the correct categories. Therefore, my framework makes it easy not only to differentiate and understand communities better but also to associate online communities with their real-world counterparts (if exist).

UK Politics

The UK dataset covers many events such as the rise of terrorism and terrorist attacks (e.g., CharlieHebdo and Tunisia attack), and many natural disasters (e.g., Nepal earthquake and Ebola) that happened in the first nine months of 2015. However, the major event in this period of time is the UK general election. Brief background knowledge about five major UK parties during the general election are provided as follows (Moran, 2015),

- **Conservative Party:** This party is also known as Tory and was led by David Cameron in 2015. David Cameron also led the UK government before and after

the election of 2015. George Osborne, Nicky Morgan, and Jeremy Hunt were some of his secretaries.

- **Labour Party:** Ed Miliband was the leader of the Labour party for the election and selected Tom Watson as his deputy chair and campaign coordinator. Jeremy Corbyn, Yvette Cooper, Liz Kendall, and Andy Burnham were among the prominent members of the party.

- **Liberal Democrat Party:** Nick Clegg led the Liberal Democrat Party in 2015. Norman Lamb, John Leech, Nick Harvey, Tim Farron, and Charles Kennedy were some of the party's parliamentarians.

- **Scottish National Party:** The SNP is a Scottish Nationalist party led by Nicola Sturgeon in 2015. Alan Brown and Neil Gray were some of the party's parliamentarians.

- **UK Independence Party:** UKIP was led by Nigel Farage in 2015. The party embodies opposition to both United Kingdom EU membership and immigration.

Table 3.6 shows the profiles of five communities detected by my framework in the UK dataset as well as their corresponding ground truth political parties. As shown in the table, all parties have a common key expression, the general election of 2015 (e.g, GE2015 and GE15). I can also observe that the members of each party have generally expressed: (1) positive attitudes towards their party and also their prominent members and (2) negative attitudes towards other parties and their prominent members due to election competition (Moran, 2015). Moreover, the government was a coalition between the Conservative Party and the Liberal Democrat Party before the election. This coalition explains why they expressed positive sentiments towards the government related issues (i.e., Govt and Cameron) (Moran, 2015).

Table 3.6: The Profiles of Five Communities Detected by GSNMF in the UK Dataset.

Conservative Party (Tory)		Labour Party (Lab)		Liberal Democrat Party (Lib Dem)		Scottish National Party (SNP)		UK Independence Party (UKIP)	
Positive	Negative	Positive	Negative	Positive	Negative	Positive	Negative	Positive	Negative
+ Conservatives	− Labour	+ Labour	− Tories	+ LibDems	− Labour	+ TheSNP	− Tory	+ UKIP	± Calais
+ @David_Cameron	− Miliband	+ UKLabour	− Tory	+ @Nick_Clegg	± Iraq	GE15	± Trident	+ @Nigel_Farage'	− Labour
+ Cameron	± Tunisia	+ LabourDoorStep	− Cameron	GE2015	±Climate Change	+ SNP	− Tories	+ Nigel Farage	− ISIS
GE2015	± Paris	+@AndyBurnhamMP	A&E	+ NormanLamb	− Tories	+NicolaSturgeon	− Labour	BBCqt	± Greece
+ @George_Osborne	FIFA	+ Ed_Miliband	− David Cameron	Cardiff	− Tory	+ Scotland	+#RefugeesWelcome	GE2015	− BBC
+VoteConservative*	− ISIL	+ Britain	− Bedroom Tax	+ @TimFarron	− UKLabour	MPs	± Iraq	− Cameron	± Britain
+ NickyMorgan01*	± Heathrow	+ @GloriadePiero	− BedroomTax	+ Lib Dem	− Tuition Fees	+ Glasgow	± Paris	Mark	− Government
London	− Charles Kennedy	+ YvetteCooperMP	− Govt	±Mental Health	± HIV	Alan	± CharlieHebdo*	+ Brexit	− Tories
Wales	− Ed_Miliband	+ YvetteForLabour*	± Tax Credits	+ Lib Dems	− SNP	+ Scottish	± Syria	Telegraph	− David Cameron
+ David Cameron	− SNP	+ SteveReedMP	− SNP	+ NHS	PMQs	GE2015	± Mediterranean	BBC5live	± Libya
+ @Jeremy_Hunt	± Syria	+ @LeicesterLiz	− Government	John	± CharlieHebdo	Edinburgh	± Med	+ Queen	− Miliband
+ Team2015*	− Lab	+ LizforLeader*	± France	+ Nick Clegg	− Nigel Farage	Maiden Speech	± French	Andrew	± Tunisia
Chris	− LibDems	+@TristramHuntMP	± Tunisia	LBC	± Ebola	+ NHS	± Charles	+ George's	± Paris
+ @Tracey_Crouch	± Calais	GE2015	± Syria	+ LibDem	± Paris	Neil	± Tunisia	− JeremyCorbyn*	− SNP
+ England	± Nepal	+ VoteLabour*	+ Europe	+ Govt	− Welfare Bill	+Nicola Sturgeon	− Westminster	− Jeremy Corbyn	± Mediterranean

According to the negative attitudes of almost all parties, I can determine that the Conservative party and the Labour party are the ones towards which other parties expressed most of the negative sentiments. Furthermore, these two parties expressed a high negative sentiment towards each other. The reason behind this antagonism is that these parties are the two biggest parties having the highest chance of winning an outright majority in the election (Moran, 2015). Moreover, UKIP's negative view on Calais, the city in France where immigrants enter the UK, and Mediterranean (immigrants/immigration) reflects its anti-immigration stance. In addition, UKIP's positive sentiment on Brexit and its negative sentiment on Greece indicates its anti-EU orientation.

Table 3.7 shows the profiles of five communities detected by GNMF in the UK dataset as well as their corresponding ground truth political parties. Due to space limitation, I do not provide the community profiles detected by DNMF. As I observe from Table 3.7, the same problem which exists in the profiles of communities detected by GNMF and DNMF in the US dataset still exists here. In other words, it is not clear which community represents which party. For instance, the profile which corresponds to the Conservative Party and the Labour Party shared many key expressions such as Labour, Tories, David Cameron (@David_Cameron), @Ed_Miliband, UKLabour, and VoteLabour, but there is no other information to understand the positions of these two parties towards these key expressions in order to differentiate them and also associate the community profiles to the parties. However, as Table 3.6 suggests, the community profiles detected by my framework shows that the community associated to the Conservative Party has a positive attitude towards David Cameron but a negative attitude towards Labour and Miliband. On the other hand, the community associated with the Labour Party has a positive attitude towards Labour, UKLabour, and Miliband but a negative attitude towards Tories and David Cameron. Since this

Table 3.7: The Profiles of Five Communities Detected by GNMF in the UK Dataset.

Conservatives	Labours	Lib dems	SNPs	UKIPs
− Miliband	− UKIP	+ Libdems	+ SNP	− @JessPhillips
+ Conservatives	+ Labour	GE2015	+ Scotland	birmingham
GE2015	− Tories	− Labour	+ VoteSNP	− Labour
+ @David_Cameron	+ NHS	+ @LFeatherstone	GE15	john
− Labour	+ Britain	− @CLeslieMP	+ @TheSNP	− Libdems
+ VoteConservetives	− Cameron	Bradford	− Labour	− NHS
+ Govt	− @Nigel_Frange	+ @Nick_Clegg	− Westminster	− LabourEoin
+ NHS	+ UKLabour	London	− Tory	− Lib dems
LeaderDebates	London	± Budget2015	GE2015	− Lib dem
+ @ZacGoldsmith	+ LabourDoorStep	Wales	+ @NicolaSturgeon	Hansard
− @Ed_Miliband	BBC	+ NHS	LeadersDebate	− @TobyPerkinsMP
MPs	BBCqt	− Miliband	− LaboursDoorStep	− UKLabour
London	+ Europe	LeaderDebate	MPs	MPs
− UKLabour	+ @AndyBurnhamMP	− @George_Osborne	± Trident	@SabelHardman
Wales	+ @ED_Miliband	+ Lib dems	+ Scottish	Jess
+ England	+ TessaJowell	− David Cameron	London	− Labour party
+ @George_Osborne	− David Cameron	+ Lib dem	− UKLabour	− @SimonDanczuk
croydon	− Tory	± Mental Health	PMQS	− Libdem
+ @NickyMorgan01	+ Corbyn	+ @SWilliamsMP	− @David_Cameron	GE2015
+ @NorwichChloe	± Calasis	+ @NormanLamb	Wales	Youtube
+ @RobertBuckland	+ @YvetteCooper	− Conservatives	+ @GradySNP	− Europe
− VoteLabour	+ VoteLabour	− Tories	+ Glasgow	− miliband
− @CLeslieMP	± Greece	+ Nick Clegg	Front Page	− Food Banks
− LabourLeadership	+ England	Cardiff	− VoteLabour	− Housing Benefit
+ Tories	+ Jeremy Corbyn	+ @TimFarron	− ScottishLabour	Google
+ Minister	− Farage	− VoteConservatives	+ Nicola Sturgeon	− @GiselaStuart
− Guardian	+ YvetteCooperLabour	Bristol	− LabourLeadership	− Labour MPs
Leeds	− Telegraph	Croydon	− Lab	+ Britain
+ Government	+ @EmmaReynoldMP	Norwich	Edinburgh	Wales
State	Thurrock	− Chancellor	− @AndyburnhamMP	− @David_Cameron

Note: All colors, signs, and the name of parties in the table are ground truth.

corresponds to the ground truth, I can conclude that sentiment information can play an essential role in providing better community profiles.

Quantitative Results

In this section, I aim to compare GSNMF with GNMF and DNMF in terms of their effectiveness in extracting relevant key expressions for community profiles. Figure 3.1 shows the accuracy of all methods in the US and UK datasets by considering a different number of top key expressions as community profiles. As I observe, GSNMF

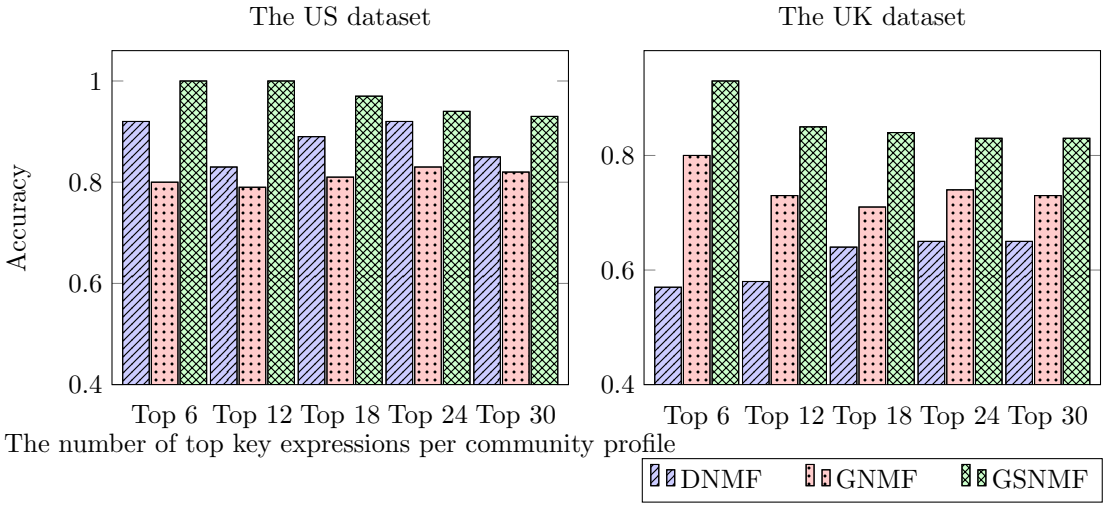


Figure 3.1: The Accuracy of Community Profiling Methods in Extracting Relevant Key Expressions.

outweighs GNMF and DNMF in all experiments. For instance, by considering top 30 key expressions as community profiles, 93% of key expressions extracted by GSNMF in the US dataset are relevant compared to 82% in GNMF and 85% in DNMF. Similarly, 83% of key expressions extracted by GSNMF in the UK dataset are relevant compared to 65% in DNMF and 73% in GNMF. The experiments also suggest that GSNMF achieves better accuracy with a lower number of top key expressions as community profiles. This implies that the higher a key expression is ranked by GSNMF, the more likely it is relevant. Following these observations, sentiment-driven community profiling produces key expressions which are more relevant than its sentiment insensitive counterparts.

3.7 Conclusion

In this chapter, I presented a sentiment-driven community profiling and detection framework uncovering a low-dimensional latent space in order to cluster users according to their opinions and social interactions. It also provides community pro-

files reflecting positive/negative collective opinions of their members. Experimental results on real-world social media datasets demonstrated: (1) my framework obtains significant performance in detecting communities compared to several state-of-the-art community detection methods, and (2) my framework presents a sentiment-driven community profiling approach providing better insights into the collective opinions of community members by dividing key expressions into positive/negative categories.

Chapter 4

UTILIZING NODE ATTRIBUTES FOR INTER-COMMUNITY RELATION DISCOVERY

4.1 Overview

Community detection on social media has attracted considerable attention for many years. However, existing methods do not reveal the relations between communities. Communities can form alliances or engage in antagonisms due to various factors, e.g., shared or conflicting goals and values. Uncovering such relations can provide better insights to understand communities and the structure of social media. According to social science findings, the attitudes that members from different communities express towards each other are largely shaped by their community membership. Hence, I hypothesize that inter-community attitudes expressed among users in social media have the potential to reflect their inter-community relations. Therefore, I first validate this hypothesis in the context of social media. Then, inspired by the hypothesis, I develop a framework to detect communities and their relations by jointly modeling users' attitudes and social interactions. I present experimental results using three real-world social media datasets to demonstrate the efficacy of the proposed framework.

4.2 Introduction

Although community detection plays an important role in providing insights into the structure and function of social media (Papadopoulos *et al.*, 2012), existing community detection methods do not reveal inter-community relations, which are indis-

pensable to deepen our insights. Moreover, to better understand communities, there is a need to uncover their relations. Indeed, social scientists suggest that "the understanding of policies and practices prevailing within groups will be inadequate unless relations among them are brought into the picture" (Sherif and Sherif, 1953). A community, or group in social sciences, is defined as a set of users with many intra-group social interactions and few inter-group ones (Girvan and Newman, 2001), who tend to have mainly positive attitudes towards each other (Festinger *et al.*, 1950; Lott and Lott, 1965).

Several methods (Chu *et al.*, 2016; Gao *et al.*, 2016; Lo *et al.*, 2013, 2011; Zhang *et al.*, 2010, 2013) have been proposed to detect antagonistic communities. There are generally two categories of such methods: (1) those which detect antagonistic communities from signed networks (Chu *et al.*, 2016; Gao *et al.*, 2016; Lo *et al.*, 2013, 2011), and (2) those which mine antagonistic communities by finding frequent patterns in users' ratings (Zhang *et al.*, 2010, 2013). However, these methods suffer from two main limitations. First, they cannot be applied to a majority of popular social network platforms (e.g., Facebook and Twitter) since these platforms do not provide signed links or users' ratings explicitly. Second, inter-community relations are not restricted to antagonisms. Indeed, communities can also form alliances.

According to social science findings, inter-community attitudes that individuals express towards each other are largely shaped by their community membership rather than their characteristics or personal relationships (Tajfel, 1979; Billig and Tajfel, 1973). Moreover, Tajfel (Tajfel, 2010) observed a pair of characteristics in inter-community behavior. First, the members of a community display uniformity in their behavior and attitude towards any other community. Second, they tend to perceive the characteristics and behavior of the members of any other community as undifferentiated. Moreover, social scientists suggest that "the social psychology of intergroup

relations is concerned with intergroup behaviour and attitudes" (Tajfel, 2010). According to these observations, inter-community attitudes that users express towards each other in social media have the potential to reflect inter-community relations.

In this chapter, I propose a framework, namely DAAC, which detects communities and their relations (i.e., antagonism, alliance, or neither) by exploiting users' social interactions (e.g., retweets) and attitudes expressed on social media. My main contributions are:

- Validating the hypothesis suggesting that inter-community attitudes that users express towards each other in social media can reflect the relations of their communities;
- Achieving higher performance in detecting communities compared to several standard community detection methods;
- Uncovering inter-community relations, i.e., antagonism, alliance, or no relation.

The rest of the chapter is organized as follows. In Section 4.3, I review related work. In Section 4.4, I formally define the problem of detecting communities and their relations on social media. Section 4.5 describes three real-world social media datasets used in the experiments. In Section 4.6, I first validate the aforementioned hypothesis and then present the proposed framework. In Section 4.7, I demonstrate the effectiveness of the proposed framework. Section 4.8 concludes the paper and discusses future work.

4.3 Related Work

There has been a lot of efforts to detect communities efficiently and accurately. To this end, a wide variety of approaches have been utilized. Modularity-based methods

are among the most well-known techniques to detect communities. The modularity measure proposed in (Newman and Girvan, 2004) evaluates whether a division is good enough to form communities. Many variants of modularity-based community detection (Clauset *et al.*, 2004; Blondel *et al.*, 2008) have been developed. Another well-known category includes spectral algorithms (Dhillon *et al.*, 2004; Ding *et al.*, 2005; Newman, 2006; Salehi *et al.*, 2018) which aims to divide the network into several communities in which most of the interactions are within communities while the number of interactions across communities is minimized. Probabilist approaches (Yu *et al.*, 2005), in which users are assigned to clusters in a probabilistic way, are also applied to the problem of community discovery. There are a variety of approaches such as information theory based methods (Rosvall and Bergstrom, 2008), random walk techniques (Harel and Koren, 2001; Pons and Latapy, 2006), and model-based methods (Raghavan *et al.*, 2007; Gregory, 2010) to tackle this problem.

Although many efforts are made to detect communities, to the best of my knowledge, no previous work has been proposed to uncover the existence of antagonism and alliance between communities. However, some efforts have been made (Chu *et al.*, 2016; Gao *et al.*, 2016; Lo *et al.*, 2013, 2011; Zhang *et al.*, 2010, 2013) to detect only antagonistic communities. These methods can be roughly divided into two main categories. The irst category includes the methods (Zhang *et al.*, 2010, 2013) utilizing frequent patterns in users' ratings to mine antagonistic communities. The second category includes the methods (Chu *et al.*, 2016; Gao *et al.*, 2016; Lo *et al.*, 2013, 2011) utilizing signed networks, having trust and distrust links, to detect antagonistic communities. A majority of these methods (Gao *et al.*, 2016; Lo *et al.*, 2013, 2011) detect a pair of subgraphs with most trust links preserved between the members of each subgraph and most distrust links remained between the members of different subgraphs. These methods are limited to detecting only a pair of antagonistic com-

munities. To address this limitation, another method (Chu *et al.*, 2016) has been proposed to detect multiple antagonistic communities by finding several dense subgraphs with the mentioned property. However, as experiments in (Chu *et al.*, 2016) show such methods usually end up with a large number of small subgraphs due to high sparsity of users' interactions in social media.

4.4 Problem Statement

I first begin with the introduction of the notations used in this chapter as summarized in Table 4.1. Let $\mathcal{U} = \{u_1, u_2, ..., u_n\}$ be the set of n users and $\mathcal{C} = \{c_1, c_2, ..., c_k\}$ indicate the set of k communities. $\mathbf{R} \in \mathbb{R}_+^{n \times n}$ denotes the social interaction matrix, where $\mathbf{R}_{i,j}$ corresponds to the number of social interactions between user u_i and user u_j. $\mathbf{S} \in \mathbb{R}^{n \times n}$ indicates the attitude matrix, where the positive/negative value of $\mathbf{S}_{i,j}$ corresponds to the positive/negative attitude strength of user u_i towards user u_j. $\mathbf{U} \in \mathbb{R}_+^{n \times k}$ indicates the community membership matrix, in which $\mathbf{U}_{i,l}$ corresponds to the membership strength of user u_i to community c_l. $\mathbf{H} \in \mathbb{R}^{k \times k}$ denotes intra/inter-community relation matrix, where $\mathbf{H}_{i,j}$, if $i \neq j$, corresponds to the strength and type of inter-community relation between community c_i and community c_j; the negative, positive, and zero value of $\mathbf{H}_{i,j}$ indicates antagonism, alliance, or no relation between community c_i and community c_j, respectively. Moreover, $\mathbf{H}_{i,i}$ corresponds to the intra-community attitudes that the members of community c_i have expressed towards each other. I define the symmetric normalization of $\mathbf{R}$ as $\widetilde{\mathbf{R}} = \mathbf{D}^{-1/2}\mathbf{R}\mathbf{D}^{-1/2}$, where $\mathbf{D} = diag(d_1, d_2, ..., d_n)$ is the degree matrix of $\mathbf{R}$ and the degree of user u_i is $d_i = \sum_{j=1}^{n} \mathbf{R}_{i,j}$. I separate positive and negative parts of matrix $\mathbf{A}$ as $\mathbf{A}^+_{i,j} = (|\mathbf{A}_{i,j}| + \mathbf{A}_{i,j})/2$ and $\mathbf{A}^-_{i,j} = (|\mathbf{A}_{i,j}| - \mathbf{A}_{i,j})/2$.

By using the aforementioned notations, the problem of detecting communities and their relations on social media can be defined as follows: *Given an attributed graph*

Table 4.1: The Notations Used in Chapter 4.

Notation	Explanation
$\mathcal{U}$	The set of users
$\mathcal{C}$	The set of communities
n	The number of users
k	The number of communities
$\mathbf{R}$	The social interaction matrix
$\mathbf{S}$	The attitude matrix
$\mathbf{U}$	The community membership matrix
$\mathbf{H}$	The community intra/inter-relation matrix
$\widetilde{\mathbf{R}}$	Symmetrically normalized matrix $\mathbf{R}$
$\mathbf{D}$	Degree matrix of $\mathbf{R}$
$\mathbf{A}^+$	The positive part of matrix $\mathbf{A}$ (i.e., $(\lvert\mathbf{A}\rvert + \mathbf{A})/2$)
$\mathbf{A}^-$	The negative part of matrix $\mathbf{A}$ (i.e., $(\lvert\mathbf{A}\rvert - \mathbf{A})/2$)

in which node features and the graph structure are represented by attitude matrix $\mathbf{S}$ *and social interaction matrix* $\mathbf{R}$ *respectively, I aim to obtain community membership matrix* $\mathbf{U}$ *and intra/inter-community relation matrix* $\mathbf{H}$*.*

4.5 Data Description

Politics is a domain in which it is common among political parties (i.e., communities) to form alliances or engage in antagonisms. To validate the aforementioned hypothesis and evaluate the proposed framework, I use the following political Twitter datasets:

- **US Dataset** consists of the tweets posted by 583 politicians from two major US political parties (the Republican Party and the Democratic Party) from August

26 to November 29, 2016. For the period of time that this dataset covers, there were antagonisms between these parties particularly due to the 2016 presidential election campaigning (Lilleker *et al.*, 2016).

- **Australia Dataset** consists of the tweets posted by 225 user accounts, including politicians and political groups, from five major Australian political parties (the Liberal Party, the National Party, the Liberal National Party, the Greens, and the Labor Party) from January 1 to November 18, 2016. For several decades, there has been a coalition among the Liberal Party, the National Party, and the Liberal National Party (Clune, 2016). In the 2016 federal election, all relations between the parties were antagonistic except the relations between the members of the coalition,.

- **UK Dataset** consists of the tweets posted by 389 user accounts, including politicians and political groups, from five major UK political parties (the Conservative Party, the Labour Party, the Scottish National Party, the Liberal Democrats Party, and the UK Independence Party) from January 1 to October 31, 2015. There was antagonism among five major UK political parties in this period of time, especially due to the 2015 general election campaigning (Moran, 2015).

Pre-processing: For all datasets, I remove the users who do not have any retweet (i.e., social interaction). Table 4.2 shows the statistics of the pre-processed datasets. All users in the datasets have been labeled with their corresponding parties, and these labels are used to evaluate the proposed method.

Although aspect-based sentiment classification techniques (Pontiki *et al.*, 2016) have been proposed to capture users' attitudes towards entities, publicly available training datasets are either too small or domain-oriented, making such techniques

Table 4.2: The Statistics of the Datasets Used in Chapter 4.

	US	Australia	UK
# of tweets	111,743	159,499	267,085
# of retweets	17,724	21,111	14,892
# of mentions	8,470	14,996	33,462
# of user accounts	583	225	389
# of true communities	2	5	5
# of allied relations	0	3	0
# of antagonistic relations	1	7	10

incapable to tackle real-world problems. Therefore, I use the following technique to extract the attitudes that users express towards each other in social media. Given each message in which author u_i has mentioned user u_j, I add the strength of the message's sentiment to the corresponding elements of matrix $\mathbf{S}$ (i.e., $\mathbf{S}_{i,j}$). Even though some messages may carry a negative sentiment, the author may not necessarily have an antagonistic attitude towards a mentioned user. To alleviate this problem, I ignore such messages if there is social interaction (i.e., retweet) between the author and the mentioned user since social interaction indicates the presence of a good relationship (Conover *et al.*, 2011). I utilize SentiStrength (Thelwall *et al.*, 2010) to detect the sentiment polarity and strength of messages. I have made the code and datasets used in this chapter available [1] .

[1] `https://github.com/amin-salehi/DAAC`

4.6 The Proposed Framework

In this section, I first demonstrate the existence of a significant level of correlation between the type of inter-community relation (i.e., alliance or antagonism) between two communities and the type of sentiment (i.e., positive or negative) that members from these communities expressed towards each other. Next, I propose the framework.

4.6.1 Validating the Hypothesis

According to social science findings (Tajfel, 1979; Billig and Tajfel, 1973), the attitudes that members from different communities express towards each other are largely shaped by their community membership. Therefore, I hypothesize that inter-community attitudes expressed among users towards each other in social media have the potential to reflect inter-community relations. However, the findings borrowed from social sciences do not necessarily hold in social media due to many factors, such as the validity and representativeness of available information (Tufekci, 2014; Ruths and Pfeffer, 2014). Moreover, the attitudes that users express towards each other in social media might result from users' personal relationships. Therefore, in this section, I aim to verify my hypothesis by answering the following two questions. With this respect, I utilize the Australia dataset since it is the only dataset containing both allied and antagonistic relations.

- Are the communities of two users who express negative attitudes towards each other more likely to be in antagonism?
- Are the communities of two users who express positive attitudes towards each other more likely to be in alliance?

I first answer the former by using the following procedure inspired by (Beigi *et al.*, 2016). For each pair of users (u_i, u_j) who are from different communities and have

expressed negative attitudes towards each other (i.e., $\mathbf{S}_{i,j} < 0$), I randomly select a user u_k where users u_i and u_k are from different communities and have not expressed negative attitudes towards each other (i.e., $\mathbf{S}_{i,k} \geq 0$). Then, I check whether there is antagonism between the communities of u_i and u_j and between the communities of u_i and u_k. If there is antagonism between the communities of u_i and u_j, I set $t_p = 1$; otherwise $t_p = 0$. Similarly, if there is antagonism between the communities of u_i and u_k, I set $t_r = 1$; otherwise $t_r = 0$. Let vector T_p denote the set of all t_ps for pairs of users from different communities who have expressed negative attitudes towards each other, and vector T_r denote the set of all t_rs for pairs of users from different communities who have not expressed negative attitudes towards each other.

I conduct a two-sample t-test on T_p and T_r. The null hypothesis H_0 and alternative hypothesis H_1 are defined as follows:

$$H_0 : T_p \leq T_r, \quad H_1 : T_p > T_r \tag{4.1}$$

The null hypothesis is rejected at significance level $a = 0.01$ with p-value of 3.56e−105. Therefore, the result of the two-sample t-test demonstrates that *the communities of two users who express negative attitudes towards each other are highly probable to be in antagonism.* I apply a similar procedure to answer the second question. For brevity, I only report the result of the two-sample t-test. The null hypothesis is rejected at significance level $a = 0.01$ with p-value of 1.57e−26. As a result, I conclude that *the communities of two users who express positive attitudes towards each other are highly probable to be in alliance.*

4.6.2 Modeling Users' Attitudes

In the previous section, I demonstrated that inter-community attitudes expressed by users can reflect the relation of their communities in the context of social media.

Inspired by this observation, I propose a model which uncovers intra/inter-community relations by exploiting the attitudes users express towards each other as,

$$\begin{aligned} \min_{\mathbf{U},\mathbf{H}} \quad & ||\mathbf{W} \odot (\mathbf{S} - \mathbf{U}\mathbf{H}\mathbf{U}^T)||_F^2 \\ \text{s.t.} \quad & \mathbf{U} \geq 0. \end{aligned} \tag{4.2}$$

where $\odot$ is Hadamard product, $\mathbf{W}_{i,j}$ controls the contribution of $S_{i,j}$ in the model, and a typical choice of $\mathbf{W} \in \mathbb{R}_+^{n \times n}$ is,

$$\mathbf{W} = \begin{cases} 0, \text{ if } \mathbf{S} = 0 \\ 1, \text{ otherwise} \end{cases} \tag{4.3}$$

Given communities c_i and c_j, Eq. (4.2) aims to uncover their inter-community relation $\mathbf{H}_{i,j}$ by using their attitudes. To this end, $\mathbf{U}_{:,i}\mathbf{H}_{i,j}\mathbf{U}_{:,j}^T$ estimates the inter-community attitudes among the members of these two communities as presented in matrix $\mathbf{S}$. Since the non-negativity constraint only holds on $\mathbf{U}$, $\mathbf{H}_{i,j}$ will be negative, positive, or zero if the members of two communities have generally expressed negative, positive, or no attitudes towards each other, respectively. The lower the negative value of $\mathbf{H}_{i,j}$ is, the more antagonistic communities c_i and c_j are. On the other hand, the larger the positive value of $\mathbf{H}_{i,j}$ is, the more allied communities c_i and c_j are. Moreover, $\mathbf{H}_{i,i}$ indicates the intra-community attitudes that the members of community c_i have expressed towards each other.

4.6.3 Modeling Social Interactions

Social interactions are one of the most effective sources of information to detect communities (Papadopoulos *et al.*, 2012). In this section, I aim to cluster users into k communities with the most social interactions within each community and the fewest

social interactions between communities. To this end, I use the following model,

$$\begin{aligned} \max_{\mathbf{U}} \quad & Tr(\mathbf{U}^T\widetilde{\mathbf{R}}\mathbf{U}) \\ \text{s.t.} \quad & \mathbf{U} \geq 0, \mathbf{U}^T\mathbf{U} = \mathbf{I}. \end{aligned} \tag{4.4}$$

where $\mathbf{I}$ is the identity matrix with the proper size. In fact, Eq. (4.4) is equivalent to the nonnegative relaxed normalized cut as put forth in (Ding *et al.*, 2005).

4.6.4 The Proposed Framework DAAC

I separately introduced the models to utilize users' attitudes and social interactions. In this section, I propose my framework DAAC, which jointly exploits these two models to uncover communities and their relations. The proposed framework requires solving the following optimization problem,

$$\begin{aligned} \min_{\mathbf{U},\mathbf{H}} \quad & \boldsymbol{\mathcal{F}} = ||\mathbf{W} \odot (\mathbf{S} - \mathbf{U}\mathbf{H}\mathbf{U}^T)||_F^2 - \lambda Tr(\mathbf{U}^T\widetilde{\mathbf{R}}\mathbf{U}) \\ \text{s.t.} \quad & \mathbf{U} \geq 0, \mathbf{U}^T\mathbf{U} = \mathbf{I}. \end{aligned} \tag{4.5}$$

where λ is a non-negative regularization parameter controlling the contribution of social interactions in the final solution.

Since the optimization problem in Eq. (4.5) is not convex with respect to variables $\mathbf{U}$ and $\mathbf{H}$ together, there is no guarantee to find the global optimal solution. As suggested by (Lee and Seung, 2001), I introduce an alternative scheme to find a local optimal solution of the optimization problem. The key idea is optimizing the objective function with respect to one of the variables $\mathbf{U}$ or $\mathbf{H}$, while fixing the other one. The algorithm keeps updating the variables until convergence.

Optimizing the objective function $\boldsymbol{\mathcal{F}}$ in Eq. (4.5) with respect to U is equivalent to solving:

$$\begin{aligned} \min_{\mathbf{U}} \quad & \boldsymbol{\mathcal{F}}_{\mathbf{U}} = ||\mathbf{W} \odot (\mathbf{S} - \mathbf{U}\mathbf{H}\mathbf{U}^T)||_F^2 - \lambda Tr(\mathbf{U}^T\widetilde{\mathbf{R}}\mathbf{U}) \\ \text{s.t.} \quad & \mathbf{U} \geq 0, \mathbf{U}^T\mathbf{U} = \mathbf{I}. \end{aligned} \tag{4.6}$$

Let $\boldsymbol{\Gamma}$ and $\boldsymbol{\Lambda}$ be the Lagrange multiplier for constraints $\mathbf{U}^T\mathbf{U} = \mathbf{I}$ and $\mathbf{U} \geq 0$, respectively, and the Lagrange function is defined as follows:

$$\begin{aligned} \min_{\mathbf{U}} \quad \boldsymbol{\mathcal{L}}_{\mathbf{U}} = & ||\mathbf{W} \odot (\mathbf{S} - \mathbf{U}\mathbf{H}\mathbf{U}^T)||_F^2 - \lambda Tr(\mathbf{U}^T\widetilde{\mathbf{R}}\mathbf{U}) \\ & - Tr(\boldsymbol{\Lambda}\mathbf{U}^T) + Tr(\boldsymbol{\Gamma}(\mathbf{U}^T\mathbf{U} - \mathbf{I})) \end{aligned} \tag{4.7}$$

The derivative of $\boldsymbol{\mathcal{L}}_{\mathbf{U}}$ with respect to $\mathbf{U}$ is

$$\begin{aligned} \frac{\partial \boldsymbol{\mathcal{L}}_{\mathbf{U}}}{\partial \mathbf{U}} = & -2(\mathbf{W} \odot \mathbf{W} \odot \mathbf{S})\mathbf{U}\mathbf{H}^T - 2(\mathbf{W} \odot \mathbf{W} \odot \mathbf{S})^T\mathbf{U}\mathbf{H} \\ & + 2(\mathbf{W} \odot \mathbf{W} \odot \mathbf{U}\mathbf{H}\mathbf{U}^T)\mathbf{U}\mathbf{H}^T \\ & + 2(\mathbf{W} \odot \mathbf{W} \odot \mathbf{U}\mathbf{H}\mathbf{U}^T)^T\mathbf{U}\mathbf{H} \\ & - 2\lambda\widetilde{\mathbf{R}}\mathbf{U} - \boldsymbol{\Lambda} + 2\mathbf{U}\boldsymbol{\Gamma} \end{aligned} \tag{4.8}$$

For the sake of simplicity, let us assume that,

$$\mathbf{E}_1 = -(\mathbf{W} \odot \mathbf{W} \odot \mathbf{S})\mathbf{U}\mathbf{H}^T \tag{4.9}$$

$$\mathbf{E}_2 = -(\mathbf{W} \odot \mathbf{W} \odot \mathbf{S})^T\mathbf{U}\mathbf{H} \tag{4.10}$$

$$\mathbf{E}_3 = (\mathbf{W} \odot \mathbf{W} \odot \mathbf{U}\mathbf{H}\mathbf{U}^T)\mathbf{U}\mathbf{H}^T \tag{4.11}$$

$$\mathbf{E}_4 = (\mathbf{W} \odot \mathbf{W} \odot \mathbf{U}\mathbf{H}\mathbf{U}^T)^T\mathbf{U}\mathbf{H} \tag{4.12}$$

By setting $\frac{\partial \boldsymbol{\mathcal{L}}_{\mathbf{U}}}{\partial \mathbf{U}} = 0$, I get

$$\Lambda = -2\mathbf{E}_1 - 2\mathbf{E}_2 + 2\mathbf{E}_3 + 2\mathbf{E}_4 - 2\lambda\widetilde{\mathbf{R}}\mathbf{U} + 2\mathbf{U}\boldsymbol{\Gamma} \tag{4.13}$$

With the KKT complementary condition for the nonnegativity of $\mathbf{U}$, I have

$$\boldsymbol{\Lambda}_{ij}\mathbf{U}_{ij} = 0 \tag{4.14}$$

Therefore, I have

$$(\mathbf{E}_1 + \mathbf{E}_2 + \mathbf{E}_3 + \mathbf{E}_4 - \lambda\widetilde{\mathbf{R}}\mathbf{U} + 2\mathbf{U}\mathbf{\Gamma})_{ij}\mathbf{U}_{ij} = 0 \tag{4.15}$$

where

$$\mathbf{\Gamma} = -\mathbf{U}^T\mathbf{E}_1 - \mathbf{U}^T\mathbf{E}_2 - \mathbf{U}^T\mathbf{E}_3 - \mathbf{U}^T\mathbf{E}_4 + \lambda\mathbf{U}^T\widetilde{\mathbf{R}}\mathbf{U} \tag{4.16}$$

Since $\mathbf{E}_1$, $\mathbf{E}_2$, $\mathbf{E}_3$, $\mathbf{E}_4$, and $\mathbf{\Gamma}$ can take mixed signs. Suggested by (Ding *et al.*, 2010), I separate positive and negative parts of any matrix A as

$$\begin{aligned} \mathbf{A}_{ij}^+ &= (|\mathbf{A}_{ij}| + \mathbf{A}_{ij})/2 \\ \mathbf{A}_{ij}^- &= (|\mathbf{A}_{ij}| - \mathbf{A}_{ij})/2 \end{aligned} \tag{4.17}$$

Then, I get the following update rule of $\mathbf{U}$,

$$\mathbf{U} = \mathbf{U} \odot \sqrt{\frac{\mathbf{E}_1^+ + \mathbf{E}_2^+ + \mathbf{E}_3^- + \mathbf{E}_4^- + \lambda\widetilde{\mathbf{R}}\mathbf{U} + \mathbf{U}\mathbf{\Gamma}^-}{\mathbf{E}_1^- + \mathbf{E}_2^- + \mathbf{E}_3^+ + \mathbf{E}_4^+ + \mathbf{U}\mathbf{\Gamma}^+}} \tag{4.18}$$

The derivative of $\boldsymbol{\mathcal{F}}$ with respect to $\mathbf{H}$ is as follows:

$$\begin{aligned} \frac{\partial \boldsymbol{\mathcal{F}}}{\partial \mathbf{H}} =& -2\mathbf{U}^T(\mathbf{W} \odot \mathbf{W} \odot \mathbf{S})\mathbf{U} \\ & - 2\mathbf{U}^T(\mathbf{W} \odot \mathbf{W} \odot \mathbf{U}\mathbf{H}\mathbf{U}^T)\mathbf{U} \end{aligned} \tag{4.19}$$

Thus, the update rule of $\mathbf{H}$ is as follows:

$$\mathbf{H} = \mathbf{H} - \alpha\frac{\partial \boldsymbol{\mathcal{F}}}{\partial \mathbf{H}} \tag{4.20}$$

where α is the learning rate for updating $\mathbf{H}$.

4.6.5 Time Complexity

The detailed algorithm for DAAC is shown in Algorithm 3. I briefly review Algorithm 3. In line 1, it randomly initializes $\mathbf{U}$ and $\mathbf{H}$. From line 2 to 5, it updates $\mathbf{U}$ and $\mathbf{H}$ until convergence is achieved. In Algorithm 3, the most costly operations are

the matrix multiplications in update rules Eq. (4.18) and Eq. (4.20) on which I focus in this section. $\mathbf{W}$ and $\mathbf{R}$ are usually very sparse matrices, so let N_w and N_r denote the number of non-zero elements of $\mathbf{W}$ and $\mathbf{R}$, respectively. The time complexities of Eq. (4.18) and Eq. (4.20) are described as follows:

- I first focus on the time complexity of Eq. (4.18). Note that $\mathbf{W} \odot \mathbf{W} \odot \mathbf{S}$ needs to be calculated once. Therefore, the time complexities of both $\mathbf{E_1}$ and $\mathbf{E_2}$ are $\mathcal{O}(N_w k + nk^2)$ thanks to the sparsity of matrices $\mathbf{W}$ and $\mathbf{S}$. The time complexity of $\mathbf{W} \odot \mathbf{W} \odot \mathbf{U}\mathbf{H}\mathbf{U}^T$ is $\mathcal{O}(N_w n + nk^2 + n^2 k)$. The number of non-zero values of $\mathbf{W} \odot \mathbf{W} \odot \mathbf{U}\mathbf{H}\mathbf{U}^T$ is the same as $\mathbf{W}$ owing to the sparsity of $\mathbf{W}$. Thus, the time complexities of both $\mathbf{E_3}$ and $\mathbf{E_4}$ are $\mathcal{O}(N_w n + nk^2 + n^2 k)$. Using a similar procedure, the time complexities of $\widetilde{\mathbf{R}}\mathbf{U}$ and $\mathbf{\Gamma}$ are $\mathcal{O}(N_r k)$ and $\mathcal{O}(N_w n + nk^2 + n^2 k + N_r k)$, respectively. As a result, the time complexity of Eq. (4.18) is $\mathcal{O}(N_w(n + k) + N_r k + nk^2 + n^2 k)$.

- Now I provide the time complexity of Eq. (4.20). The cost of $\mathbf{U}^T(\mathbf{W} \odot \mathbf{W} \odot \mathbf{S})$ is $\mathcal{O}(N_w k)$ thanks to the sparsity of $\mathbf{W}$. Thus, the time complexity of $\mathbf{U}^T(\mathbf{W} \odot \mathbf{W} \odot \mathbf{S})\mathbf{U}$ is $\mathcal{O}(N_w k + nk^2)$. Similarly, the cost of $\mathbf{U}^T(\mathbf{W} \odot \mathbf{W} \odot \mathbf{U}\mathbf{H}\mathbf{U}^T)\mathbf{U}$ is $\mathcal{O}(N_w n + nk^2 + n^2 k)$. Therefore, the time complexity of Eq. (4.20) is $\mathcal{O}(N_w(n + k) + nk^2 + n^2 k)$.

Hence, the time complexity of Algorithm 3 is $\mathcal{O}(i(N_w(n + k) + N_r k + nk^2 + n^2 k))$ where i is the number of iterations required for the convergence. My framework can be applied to large scale social network platforms by exploiting distributed approaches outlined in (Liu *et al.*, 2010; Gemulla *et al.*, 2011; Li *et al.*, 2014).

Algorithm 3 The Proposed Algorithm for DAAC

Input: attitude matrix $\mathbf{S}$ and social interaction matrix $\mathbf{R}$

Output: community membership matrix $\mathbf{U}$ and intra/inter-community relation matrix $\mathbf{H}$

Initialize $\mathbf{U}$ and $\mathbf{H}$ randomly where $\mathbf{U} \geq 0$
while not convergent **do**
    Update $\mathbf{U}$ according to Eq. (4.18)
    Update $\mathbf{H}$ according to Eq. (4.20)
end while

4.7 Experiments

To evaluate the proposed framework, I design the required experiments to answer the following two questions.

1. How effective is my framework compared to the standard community detection methods?

2. How effective is my framework in discovering inter-community relations?

In the next section, I first compare the performance of several well-known community detection methods with DAAC. Then, I evaluate the effectiveness of my framework in uncovering inter-community relations. Finally, I study the sensitivity of my framework with respect to regularization parameter λ. For the experiments, I set the number of communities for any method, if it is required, as the true number of communities (i.e., parties) in each dataset.

4.7.1 Evaluation of Community Detection

Baselines

In order to demonstrate the efficacy of DAAC, I compare it with six well-known community detection methods presented as follows:

- **Louvain:** This method (Blondel *et al.*, 2008) greedily maximizes the benefit function known as modularity to detect communities.
- **InfoMap:** This baseline (Rosvall and Bergstrom, 2008) is based on information theory and compresses the description of random walks in order to find communities.
- **Leading eigenvectors:** Newton (Newman, 2006) presents a formulation of modularity in a matrix form, namely modularity matrix. Then, he proposes to use the eigenvectors of modularity matrix to detect communities.
- **CNM:** This method (Clauset *et al.*, 2004) uses a greedy approach to find the divisions of the network which maximizes the modularity.
- **Label propagation:** (Raghavan *et al.*, 2007) This method initially assigns unique labels to users. Then, in each iteration, users adopt the label that most of their neighbors posses. Finally, users with the same label fall into the same community.
- **Soft clustering:** This baseline (Yu *et al.*, 2005) assigns users to communities in a probabilistic way.

Table 4.3: Performance Comparison of Community Detection Methods in Chapter 4.

	US dataset			Australia dataset			UK dataset		
Method	NMI	ARI	Purity	NMI	ARI	Purity	NMI	ARI	Purity
Louvain	0.4311	0.3863	0.9434	0.8252	0.8330	0.9422	0.8581	0.8417	0.9871
InfoMap	0.4314	0.3519	0.9468	0.8319	0.8317	0.9422	0.9097	0.9287	**0.9923**
Leading eigenvectors	0.5801	0.6780	0.9382	0.7799	0.5734	0.6933	0.9137	0.9533	0.9820
CNM	0.5029	0.4876	0.9451	0.8425	0.8483	0.9378	0.9391	0.9716	0.9846
Label propagation	0.6008	0.6556	0.9588	0.8222	0.8267	0.9378	0.9584	0.9790	0.9897
Soft clustering	0.7358	0.8292	0.9554	0.8412	0.8128	0.8444	0.9512	0.9743	0.9872
DAAC	**0.7683**	**0.8545**	**0.9623**	**0.9037**	**0.9083**	**0.9511**	**0.9588**	**0.9788**	0.9897

Performance Measures

To evaluate the performance of the methods, I utilize three following measures which are frequently used for community detection evaluation: Normalized Mutual Information (NMI), Adjusted Rand Index (ARI), and Purity.

Experimental Results

I run all methods with their hyperparameters initialized from $\{10^x | x \in [0, 9]\}$. Table 4.3 shows the best result for each method. According to the table, I can make the following observations:

- The proposed framework achieves the highest performance in terms of NMI and ARI for all three datasets. In terms of Purity, it also achieves the best in US and Australia datasets. In the UK dataset, only InfoMap obtains higher Purity compared to my framework since it generates a large number of communities (e.g., 11 communities for the UK dataset) for sparse graphs such as social media networks.

- My framework achieves its highest performance with large values of regularization parameter λ (e.g., 10^7). This implies that social interactions are more

Table 4.4: The Uncovered Relations Between Detected Communities (i.e., Parties) by DAAC in the US Dataset.

	Republicans	Democrats
Republicans	259	-138
Democrats	-138	112

Note: all values in the table are rounded.

effective in detecting communities compared to users' attitudes. I will study more on the impact of the regularization parameter in Section 4.7.3.

4.7.2 Evaluation of Inter-community Relations

In this section, I evaluate the effectiveness of the proposed framework in uncovering inter-community relations by conducting two experiments. To the best of our knowledge, there is no previous work to discover inter-community antagonistic and allied relations. Therefore, as the first experiment, I compare the inter-community relations which my framework detects with the real-world inter-community relations. Each community detected by my framework is labeled with the party to which the majority of its members belong. Then, I evaluate inter-community relations (i.e., the matrix **H**) detected by my algorithm according to the known ground-truth inter-party relations as previously presented in Section 4.5.

Table 4.4 shows intra/inter-community relation matrix **H** for the US dataset as well as the parties corresponding to the detected communities. In 2016, the Republican Party and the Democratic Party were strongly antagonistic towards each other, especially due to the 2016 presidential election campaigning [2] (Lilleker *et al.*, 2016). As Table 4.4 shows, my framework uncovers the existence of strong antagonism between these two parties. It also discovers that intra-community attitudes among the

[2] `https://en.wikipedia.org/wiki/United_States_presidential_election,_2016`

Table 4.5: The Uncovered Relations Between Detected Communities (i.e., Parties) by DAAC in the Australia Dataset.

	Liberals	Nationalists	Liberal Nationalists	Labors	Greens
Liberals	87	61	34	-21	-32
Nationalists	61	52	46	-4	-22
Liberal Nationalists	34	46	39	-4	-61
Labors	-21	-4	-4	121	-31
Greens	-32	-22	-61	-31	64

Table 4.6: The Uncovered Relations Between Detected Communities (i.e., Parties) by DAAC in the UK Dataset.

	Conservatives	Labours	Lib dems	SNPs	UKIPs
Conservatives	154	-37	-7	-21	-9
Labours	-37	242	-8	-11	-26
Lib dems	-7	-8	63	-3	-14
SNPs	-21	-11	-3	55	-5
UKIPs	-9	-26	-14	-5	30

members of each community are highly positive as expected owing to the election campaign dynamics.

Table 4.5 shows intra/inter-community relation matrix **H** for the Australia dataset as well as the parties corresponding to the detected communities. The Liberal Party, the National Party, and the Liberal National party forged a coalition in the 2016 federal election. Except the relations between the members of the coalition, other relations among all parties were antagonistic [3] . As shown in Table 4.5, my framework uncovers the coalition in which the three involved parties are in alliance with each other. It also discovers antagonism between the members of the coalition and other

[3] https://en.wikipedia.org/wiki/Australian_federal_election,_2016

Table 4.7: Inter-community Detection Performance of DAAC and the Two-step Approach.

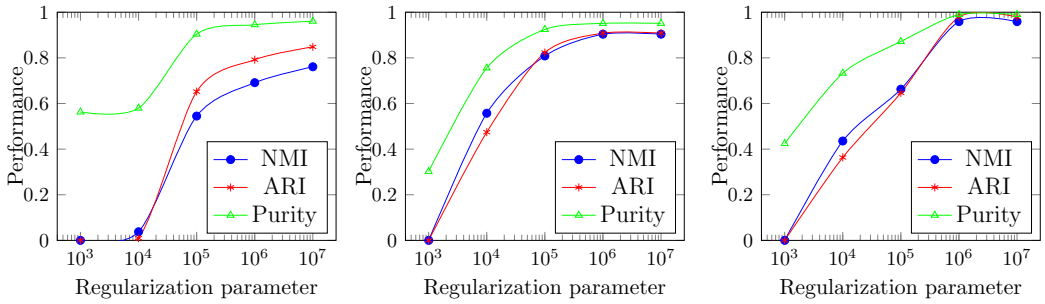
	US	**Australia**	**UK**
Two-step approach	1.0	1.0	0.8
DAAC	1.0	1.0	1.0

parties as well as the antagonism between the Greens and the Labor Party. Moreover, it detects high positive intra-community attitudes among the members of communities as expected.

Table 4.6 shows intra/inter-community relation matrix **H** for the UK dataset as well as the parties corresponding to the detected communities. In 2015, there were antagonisms between all five major UK political parties, especially due to the 2015 general election campaigning [4] (Moran, 2015). As shown in Table 4.6, my framework correctly detects all antagonistic relations between these parties. It also discovers that intra-community attitudes among the members of each community are highly positive as expected.

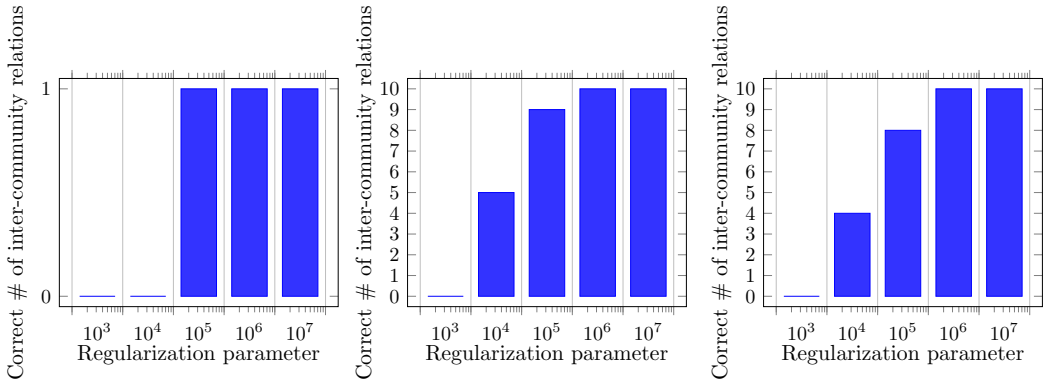
The second experiment compares my framework with a two-step approach described as follows. I first utilize social interactions to detect communities. Then, I aggregate the sentiment expressed among the members of different communities in order to figure out their inter-community relations. To have a fair comparison, I use Eq. (4.4) to detect communities for the two-step approach; which is the main component in DAAC for utilizing social interactions. As Table 4.7 shows, the two-step approach is able to detect correct relations in the US and Australia datasets. However, it fails to detect two out of ten inter-community relations in the UK dataset.

[4] `https://en.wikipedia.org/wiki/United_Kingdom_general_election,_2015`



(a) US dataset (b) Australia dataset (c) UK dataset

Figure 4.1: Community Detection Performance With Regard to λ.



(a) US dataset (b) Australia dataset (c) UK dataset

Figure 4.2: The Correct Number of Inter-community Relations With Regard to λ.

This result shows that the proposed framework can detect inter-community relations more accurately by jointly using and social interactions and attitudes among users compared to an approach which sequentially detects communities and their relations.

4.7.3 Study on the Regularization Parameter

In this section, I investigate the sensitivity of my framework with respect to regularization parameter λ. I vary the value of λ, and plot NMI, ARI and Purity measures in Figure 4.1 for all three datasets used in the study. Similarly, I plot the correct

number of inter-community relations discovered by DAAC in Figure 4.2 for all three datasets with respect to different values of λ.

As I observe from Figure 4.1, very large values of λ (e.g., 10^6 and 10^7) for all datasets result in the highest performance of DAAC in detecting communities. Similarly, Figure 4.2 shows that very large values of λ also result in the highest number of correct inter-community relations discovered by DAAC. The rationale behind this is that inter-community relations cannot be correctly identified unless communities are accurately detected.

4.8 Conclusion

In this chapter, I proposed a framework to discover communities and their relations by exploiting social interactions and user-generated content. I validated the hypothesis that inter-community attitudes that users express towards each other in social media can reflect inter-community relations. As inspired by this hypothesis, the proposed framework DAAC jointly models users' attitudes and social interactions in order to uncover communities and their antagonistic/allied relations. Experimental results on three real-world social media datasets demonstrated that the proposed framework obtains significant performance in detecting communities compared with several baselines and also detects inter-community relations correctly. Moreover, I showed that a two-step approach, which sequentially detect communities and their relations, can fail to detect correct inter-community relations.

REFERENCES

Abadi, M., P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: a system for large-scale machine learning.", in "OSDI", vol. 16, pp. 265–283 (2016).

Ahmed, A., N. Shervashidze, S. Narayanamurthy, V. Josifovski and A. J. Smola, "Distributed large-scale natural graph factorization", in "Proceedings of the 22nd international conference on World Wide Web", pp. 37–48 (ACM, 2013).

Akbari, M. and T.-S. Chua, "Leveraging behavioral factorization and prior knowledge for community discovery and profiling", in "Proceedings of the Tenth ACM International Conference on Web Search and Data Mining", pp. 71–79 (ACM, 2017).

Bahdanau, D., K. Cho and Y. Bengio, "Neural machine translation by jointly learning to align and translate", arXiv preprint arXiv:1409.0473 (2014).

Baldi, P., "Autoencoders, unsupervised learning, and deep architectures", in "Proceedings of ICML workshop on unsupervised and transfer learning", pp. 37–49 (2012).

Beigi, G., J. Tang and H. Liu, "Signed link analysis in social media networks.", in "ICWSM", pp. 539–542 (2016).

Belkin, M. and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering", in "Advances in neural information processing systems", pp. 585–591 (2002).

Bengio, Y., A. Courville and P. Vincent, "Representation learning: A review and new perspectives", IEEE transactions on pattern analysis and machine intelligence **35**, 8, 1798–1828 (2013).

Billig, M. and H. Tajfel, "Social categorization and similarity in intergroup behaviour", European Journal of Social Psychology **3**, 1, 27–52 (1973).

Blondel, V. D., J.-L. Guillaume, R. Lambiotte and E. Lefebvre, "Fast unfolding of communities in large networks", Journal of statistical mechanics: theory and experiment **2008**, 10, P10008 (2008).

Boratto, L., "Group recommender systems", in "Proceedings of the 10th ACM Conference on Recommender Systems", pp. 427–428 (ACM, 2016).

Cai, D., X. He, J. Han and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation", IEEE Transactions on Pattern Analysis and Machine Intelligence **33**, 8, 1548–1560 (2011).

Cai, H., V. W. Zheng, F. Zhu, K. C.-C. Chang and Z. Huang, "From community detection to community profiling", Proceedings of the VLDB Endowment **10**, 7, 817–828 (2017).

Cao, S., W. Lu and Q. Xu, "Grarep: Learning graph representations with global structural information", in "Proceedings of the 24th ACM International on Conference on Information and Knowledge Management", pp. 891–900 (ACM, 2015).

Cao, S., W. Lu and Q. Xu, "Deep neural networks for learning graph representations.", in "AAAI", pp. 1145–1152 (2016).

Chamberlain, B. P., J. Clough and M. P. Deisenroth, "Neural embeddings of graphs in hyperbolic space", arXiv preprint arXiv:1705.10359 (2017).

Chen, H., B. Perozzi, Y. Hu and S. Skiena, "Harp: Hierarchical representation learning for networks", in "Thirty-Second AAAI Conference on Artificial Intelligence", (2018a).

Chen, J., J. Zhu and L. Song, "Stochastic training of graph convolutional networks with variance reduction.", in "ICML", pp. 941–949 (2018b).

Chen, L.-C., Y. Yang, J. Wang, W. Xu and A. L. Yuille, "Attention to scale: Scale-aware semantic image segmentation", in "Proceedings of the IEEE conference on computer vision and pattern recognition", pp. 3640–3649 (2016).

Chen, W.-Y., D. Zhang and E. Y. Chang, "Combinational collaborative filtering for personalized community recommendation", in "Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining", pp. 115–123 (ACM, 2008).

Chorowski, J. K., D. Bahdanau, D. Serdyuk, K. Cho and Y. Bengio, "Attention-based models for speech recognition", in "Advances in neural information processing systems", pp. 577–585 (2015).

Chu, L., Z. Wang, J. Pei, J. Wang, Z. Zhao and E. Chen, "Finding gangs in war from signed networks", in "Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining", pp. 1505–1514 (ACM, 2016).

Clauset, A., M. E. Newman and C. Moore, "Finding community structure in very large networks", Physical review E **70**, 6, 066111 (2004).

Clune, D., "Contemporary australian political party organisations", (2016).

Conover, M., J. Ratkiewicz, M. R. Francisco, B. Gonçalves, F. Menczer and A. Flammini, "Political polarization on twitter.", ICWSM **133**, 89–96 (2011).

Cruz, J. D., C. Bothorel and F. Poulet, "Community detection and visualization in social networks: Integrating structural and semantic information", ACM Transactions on Intelligent Systems and Technology (TIST) **5**, 1, 11 (2013).

Defferrard, M., X. Bresson and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering", in "Advances in Neural Information Processing Systems", pp. 3844–3852 (2016).

Dehghani, M., S. Gouws, O. Vinyals, J. Uszkoreit and Ł. Kaiser, "Universal transformers", arXiv preprint arXiv:1807.03819 (2018).

Devlin, J., M.-W. Chang, K. Lee and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding", arXiv preprint arXiv:1810.04805 (2018).

Dhillon, I. S., Y. Guan and B. Kulis, "Kernel k-means: spectral clustering and normalized cuts", in "Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining", pp. 551–556 (ACM, 2004).

Ding, C. H., X. He and H. D. Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering.", in "SDM", vol. 5, pp. 606–610 (SIAM, 2005).

Ding, C. H., T. Li and M. I. Jordan, "Convex and semi-nonnegative matrix factorizations", IEEE transactions on pattern analysis and machine intelligence **32**, 1, 45–55 (2010).

Duran, A. G. and M. Niepert, "Learning graph representations with embedding propagation", in "Advances in Neural Information Processing Systems", pp. 5119–5130 (2017).

Festinger, L., K. W. Back and S. Schachter, "The spatial ecology of group formation", in "Social pressures in informal groups: A study of human factors in housing", vol. 3, chap. 4 (Stanford University Press, 1950).

Fishbein, M. and I. Ajzen, "Belief, attitude, intention, and behavior: An introduction to theory and research", (1977).

Gao, H. and H. Huang, "Deep attributed network embedding.", in "IJCAI", (2018).

Gao, M., E.-P. Lim, D. Lo and P. K. Prasetyo, "On detecting maximal quasi antagonistic communities in signed graphs", Data Mining and Knowledge Discovery **30**, 1, 99–146 (2016).

Gemulla, R., E. Nijkamp, P. J. Haas and Y. Sismanis, "Large-scale matrix factorization with distributed stochastic gradient descent", in "Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining", pp. 69–77 (ACM, 2011).

Gimpel, K., N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan and N. A. Smith, "Part-of-speech tagging for twitter: Annotation, features, and experiments", in "Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2", pp. 42–47 (Association for Computational Linguistics, 2011).

Girvan, M. and M. Newman, "Community structure in social and biological networks", Proc. Natl. Acad. Sci. USA **99**, cond-mat/0112110, 8271–8276 (2001).

Gregory, S., "Finding overlapping communities in networks by label propagation", New Journal of Physics **12**, 10, 103018 (2010).

Grover, A. and J. Leskovec, "node2vec: Scalable feature learning for networks", in "Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining", pp. 855–864 (ACM, 2016).

Gu, Q., C. Ding and J. Han, "On trivial solution and scale transfer problems in graph regularized nmf", in "IJCAI Proceedings-International Joint Conference on Artificial Intelligence", vol. 22, p. 1288 (2011).

Gu, Q. and J. Zhou, "Local learning regularized nonnegative matrix factorization", in "Proceedings of the 21st international jont conference on Artifical intelligence", pp. 1046–1051 (Morgan Kaufmann Publishers Inc., 2009).

Hamilton, W., Z. Ying and J. Leskovec, "Inductive representation learning on large graphs", in "Advances in Neural Information Processing Systems", pp. 1024–1034 (2017).

Han, X., L. Wang, R. Farahbakhsh, Á. Cuevas, R. Cuevas, N. Crespi and L. He, "Csd: A multi-user similarity metric for community recommendation in online social networks", Expert Systems with Applications **53**, 14–26 (2016).

Harel, D. and Y. Koren, "On clustering using random walks", in "FSTTCS", pp. 18–41 (Springer, 2001).

Harvey, M., F. Crestani and M. J. Carman, "Building user profiles from topic models for personalised search", in "Proceedings of the 22nd ACM international conference on Conference on information & knowledge management", pp. 2309–2314 (ACM, 2013).

He, R., W. S. Lee, H. T. Ng and D. Dahlmeier, "An unsupervised neural attention model for aspect extraction", in "Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)", vol. 1, pp. 388–397 (2017).

He, X. and P. Niyogi, "Locality preserving projections", in "Advances in neural information processing systems", pp. 153–160 (2004).

Hechter, M., *Principles of group solidarity*, vol. 11 (Univ of California Press, 1988).

Huang, X., J. Li and X. Hu, "Accelerated attributed network embedding", in "Proceedings of the 2017 SIAM International Conference on Data Mining", pp. 633–641 (SIAM, 2017a).

Huang, X., J. Li and X. Hu, "Label informed attributed network embedding", in "Proceedings of the Tenth ACM International Conference on Web Search and Data Mining", pp. 731–739 (ACM, 2017b).

Ikeda, K., G. Hattori, C. Ono, H. Asoh and T. Higashino, "Twitter user profiling based on text and community mining for market analysis", Knowledge-Based Systems **51**, 35–47 (2013).

Kingma, D. P. and J. Ba, "Adam: A method for stochastic optimization", arXiv preprint arXiv:1412.6980 (2014).

Kipf, T. N. and M. Welling, "Variational graph auto-encoders", NeurIPS Workshop on Bayesian Deep Learning (2016).

Kipf, T. N. and M. Welling, "Semi-supervised classification with graph convolutional networks", in "International Conference on Learning Representations", (2017).

Kozinets, R. V., "The field behind the screen: Using netnography for marketing research in online communities", Journal of marketing research **39**, 1, 61–72 (2002).

Lee, D. D. and H. S. Seung, "Algorithms for non-negative matrix factorization", in "Advances in neural information processing systems", pp. 556–562 (2001).

Lee, K., J. Caverlee, Z. Cheng and D. Z. Sui, "Campaign extraction from social media", ACM Transactions on Intelligent Systems and Technology (TIST) **5**, 1, 9 (2013).

Li, F., B. Wu, L. Xu, C. Shi and J. Shi, "A fast distributed stochastic gradient descent algorithm for matrix factorization", in "Proceedings of the 3rd International Conference on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications-Volume 36", pp. 77–87 (JMLR. org, 2014).

Lilleker, D., D. Jackson, E. Thorsen and A. Veneti, "Us election analysis 2016: Media, voters and the campaign.", (2016).

Liu, C., H.-c. Yang, J. Fan, L.-W. He and Y.-M. Wang, "Distributed nonnegative matrix factorization for web-scale dyadic data analysis on mapreduce", in "Proceedings of the 19th international conference on World wide web", pp. 681–690 (ACM, 2010).

Lo, D., D. Surian, P. K. Prasetyo, K. Zhang and E.-P. Lim, "Mining direct antagonistic communities in signed social networks", Information Processing & Management **49**, 4, 773–791 (2013).

Lo, D., D. Surian, K. Zhang and E.-P. Lim, "Mining direct antagonistic communities in explicit trust networks", in "Proceedings of the 20th ACM international conference on Information and knowledge management", pp. 1013–1018 (ACM, 2011).

Lott, A. J. and B. E. Lott, "Group cohesiveness as interpersonal attraction: a review of relationships with antecedent and consequent variables.", Psychological bulletin **64**, 4, 259 (1965).

Luong, M.-T., H. Pham and C. D. Manning, "Effective approaches to attention-based neural machine translation", arXiv preprint arXiv:1508.04025 (2015).

Maaten, L. v. d. and G. Hinton, "Visualizing data using t-sne", Journal of machine learning research **9**, Nov, 2579–2605 (2008).

Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado and J. Dean, "Distributed representations of words and phrases and their compositionality", in "Advances in neural information processing systems", pp. 3111–3119 (2013).

Mislove, A., B. Viswanath, K. P. Gummadi and P. Druschel, "You are who you know: inferring user profiles in online social networks", in "Proceedings of the third ACM international conference on Web search and data mining", pp. 251–260 (ACM, 2010).

Monti, F., D. Boscaini, J. Masci, E. Rodola, J. Svoboda and M. M. Bronstein, "Geometric deep learning on graphs and manifolds using mixture model cnns", in "Proc. CVPR", vol. 1, p. 3 (2017).

Moran, M., *Politics and Governance in the UK* (Palgrave Macmillan, 2015).

Mukherjee, A. and B. Liu, "Aspect extraction through semi-supervised modeling", in "Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1", pp. 339–348 (Association for Computational Linguistics, 2012).

Natarajan, N., P. Sen and V. Chaoji, "Community detection in content-sharing social networks", in "Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining", pp. 82–89 (ACM, 2013).

Newman, M. E., "Finding community structure in networks using the eigenvectors of matrices", Physical review E **74**, 3, 036104 (2006).

Newman, M. E. and M. Girvan, "Finding and evaluating community structure in networks", Physical review E **69**, 2, 026113 (2004).

Ou, M., P. Cui, J. Pei, Z. Zhang and W. Zhu, "Asymmetric transitivity preserving graph embedding", in "Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining", pp. 1105–1114 (ACM, 2016).

Ozer, M., N. Kim and H. Davulcu, "Community detection in political twitter networks using nonnegative matrix factorization methods", IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (2016).

Pan, S., R. Hu, G. Long, J. Jiang, L. Yao and C. Zhang, "Adversarially regularized graph autoencoder for graph embedding.", in "IJCAI", pp. 2609–2615 (2018).

Papadopoulos, S., Y. Kompatsiaris, A. Vakali and P. Spyridonos, "Community detection in social media", Data Mining and Knowledge Discovery **24**, 3, 515–554 (2012).

Pathak, N., C. DeLong, A. Banerjee and K. Erickson, "Social topic models for community extraction", in "Proceedings of the 2nd SNA-KDD Workshop", (2008).

Pei, Y., N. Chakraborty and K. Sycara, "Nonnegative matrix tri-factorization with graph regularization for community detection in social networks", in "Proceedings of the 24th International Conference on Artificial Intelligence", pp. 2083–2089 (AAAI Press, 2015).

Perozzi, B., R. Al-Rfou and S. Skiena, "Deepwalk: Online learning of social representations", in "Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining", pp. 701–710 (ACM, 2014).

Perozzi, B., V. Kulkarni and S. Skiena, "Walklets: Multiscale graph embeddings for interpretable network classification", arXiv preprint arXiv:1605.02115 (2016).

Pons, P. and M. Latapy, "Computing communities in large networks using random walks.", J. Graph Algorithms Appl. **10**, 2, 191–218 (2006).

Pontiki, M., D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, A.-S. Mohammad, M. Al-Ayyoub, Y. Zhao, B. Qin, O. De Clercq *et al.*, "Semeval-2016 task 5: Aspect based sentiment analysis", in "Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)", pp. 19–30 (2016).

Qiu, G., B. Liu, J. Bu and C. Chen, "Opinion word expansion and target extraction through double propagation", Computational linguistics **37**, 1, 9–27 (2011).

Raghavan, U. N., R. Albert and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks", Physical review E **76**, 3, 036106 (2007).

Rosvall, M. and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure", Proceedings of the National Academy of Sciences **105**, 4, 1118–1123 (2008).

Rush, A. M., S. Chopra and J. Weston, "A neural attention model for abstractive sentence summarization", arXiv preprint arXiv:1509.00685 (2015).

Ruths, D. and J. Pfeffer, "Social media for large studies of behavior", Science **346**, 6213, 1063–1064 (2014).

Sachan, M., D. Contractor, T. A. Faruquie and L. V. Subramaniam, "Using content and interactions for discovering communities in social networks", in "Proceedings of the 21st international conference on World Wide Web", pp. 331–340 (ACM, 2012).

Sahebi, S. and W. W. Cohen, "Community-based recommendations: a solution to the cold start problem", in "Workshop on recommender systems and the social web, RSWEB", (2011).

Salehi, A., M. Ozer and H. Davulcu, "Sentiment-driven community profiling and detection on social media", in "Proceedings of the 29th ACM Conference on Hypertext and Social Media", (ACM, 2018).

Schlichtkrull, M., T. N. Kipf, P. Bloem, R. van den Berg, I. Titov and M. Welling, "Modeling relational data with graph convolutional networks", in "European Semantic Web Conference", pp. 593–607 (Springer, 2018).

Sen, P., G. Namata, M. Bilgic, L. Getoor, B. Galligher and T. Eliassi-Rad, "Collective classification in network data", AI magazine **29**, 3, 93 (2008).

Shang, F., L. Jiao and F. Wang, "Graph dual regularization non-negative matrix factorization for co-clustering", Pattern Recognition **45**, 6, 2237–2250 (2012).

Sherif, M. and C. W. Sherif, "Groups in harmony and tension; an integration of studies of intergroup relations.", (1953).

Tajfel, H., "Human intergroup conflict: Useful and less useful forms of analysis", Human ethology: Claims and limits of a new discipline pp. 369–422 (1979).

Tajfel, H., *Social identity and intergroup relations* (Cambridge University Press, 2010).

Tang, D., B. Qin and T. Liu, "Aspect level sentiment classification with deep memory network", arXiv preprint arXiv:1605.08900 (2016).

Tang, J., M. Qu, M. Wang, M. Zhang, J. Yan and Q. Mei, "Line: Large-scale information network embedding", in "Proceedings of the 24th International Conference on World Wide Web", pp. 1067–1077 (International World Wide Web Conferences Steering Committee, 2015).

Thelwall, M., K. Buckley, G. Paltoglou, D. Cai and A. Kappas, "Sentiment strength detection in short informal text", Journal of the American Society for Information Science and Technology **61**, 12, 2544–2558 (2010).

Tian, F., B. Gao, Q. Cui, E. Chen and T.-Y. Liu, "Learning deep representations for graph clustering.", in "AAAI", pp. 1293–1299 (2014).

Tufekci, Z., "Big questions for social media big data: Representativeness, validity and other methodological pitfalls", arXiv preprint arXiv:1403.7400 (2014).

Turner, J. C., M. A. Hogg, P. J. Oakes, S. D. Reicher and M. S. Wetherell, *Rediscovering the social group: A self-categorization theory.* (Basil Blackwell, 1987).

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, "Attention is all you need", in "Advances in Neural Information Processing Systems", pp. 5998–6008 (2017).

Velickovic, P., G. Cucurull, A. Casanova, A. Romero, P. Lio and Y. Bengio, "Graph attention networks", in "International Conference on Learning Representations", (2018).

Veličković, P., W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio and R. D. Hjelm, "Deep graph infomax", in "International Conference on Learning Representations", (2019).

Wang, C., S. Pan, G. Long, X. Zhu and J. Jiang, "Mgae: Marginalized graph autoencoder for graph clustering", in "Proceedings of the 2017 ACM on Conference on Information and Knowledge Management", pp. 889–898 (ACM, 2017a).

Wang, D., P. Cui and W. Zhu, "Structural deep network embedding", in "Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining", pp. 1225–1234 (ACM, 2016).

Wang, X., P. Cui, J. Wang, J. Pei, W. Zhu and S. Yang, "Community preserving network embedding.", in "AAAI", pp. 203–209 (2017b).

Wang, X., R. Girshick, A. Gupta and K. He, "Non-local neural networks", in "The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)", vol. 1, p. 4 (2018).

Wei, X., L. Xu, B. Cao and P. S. Yu, "Cross view link prediction by learning noise-resilient representation consensus", in "Proceedings of the 26th International Conference on World Wide Web", pp. 1611–1619 (International World Wide Web Conferences Steering Committee, 2017).

Yang, C., Z. Liu, D. Zhao, M. Sun and E. Y. Chang, "Network representation learning with rich text information.", in "IJCAI", pp. 2111–2117 (2015).

Yang, Z., W. W. Cohen and R. Salakhutdinov, "Revisiting semi-supervised learning with graph embeddings", arXiv preprint arXiv:1603.08861 (2016).

Ying, R., R. He, K. Chen, P. Eksombatchai, W. L. Hamilton and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems", in "Proceedings of the 24th ACM SIGKDD international conference on Knowledge discovery and data mining", (2018).

Yu, K., S. Yu and V. Tresp, "Soft clustering on graphs", in "Advances in neural information processing systems", pp. 1553–1560 (2005).

Zhang, K., D. Lo and E.-P. Lim, "Mining antagonistic communities from social networks", in "Pacific-Asia Conference on Knowledge Discovery and Data Mining", pp. 68–80 (Springer, 2010).

Zhang, K., D. Lo, E.-P. Lim and P. K. Prasetyo, "Mining indirect antagonistic communities from social interactions", Knowledge and information systems **35**, 3, 553–583 (2013).

Zhang, L. and B. Liu, "Aspect and entity extraction for opinion mining", in "Data mining and knowledge discovery for big data", pp. 1–40 (Springer, 2014).

Zhou, D., E. Manavoglu, J. Li, C. L. Giles and H. Zha, "Probabilistic models for discovering e-communities", in "Proceedings of the 15th international conference on World Wide Web", pp. 173–182 (ACM, 2006).

Zhou, W., H. Jin and Y. Liu, "Community discovery and profiling with social messages", in "Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining", pp. 388–396 (ACM, 2012).

Zhu, X. and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation", (2002).