Spring 5-2017

Detection and Analysis of Online Extremist Communities

Matthew Curran Benigni
Carnegie Mellon University

Detection and Analysis of Online Extremist Communities

Matthew Curran Benigni

CMU-ISR-17-108

May 2017

Institute for Software Research
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Dr. Kathleen M. Carley
Dr. Zico Kolter
Dr. Daniel Neil
Dr. Randy Garrett

*Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.*

This work was supported by the Office of Naval Research (ONR) through MINERVA grant N000141310835 on State Stability, MURI grant N000140811186 on Structural Approach to the Incorporation of Cultural Knowledge in Adaptive Adversary Models, and grant N000141512564 on Tracking Covert Groups in Twitter.

The views ond conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Office of Naval Research or the U.S. Government.

Keywords: Covert Network Detection, Community Detection, Annotated Networks, Multilayer Networks, Heterogeneous Networks, Spectral Clustering, Socialbots, Botnets

Contents

List of Figures

List of Tables

Acknowledgements

I am extremely grateful to my advisor, Kathleen Carley, for affording me the opportunity to pursue research that has been challenging, consuming, and relevant. Her wisdom and guidance have not only contributed to this work, but to my growth as a scholar and warrior. I would also like to thank each of the members of my committee in shaping this thesis with recommendations and critique. Specifically I would like to thank Dr. Randy Garret whose work motivated my desire to serve the intelligence community.

I also would like to acknowledge important scholastic mentors like Reinhard Furrer, my advisor at the Colorado School of Mines, and Howard Seltman, a senior research statistician at Carnegie Mellon University. In both cases, they offered time and advice far beyond what their duty positions required of them. I cannot over emphasize the contribution both of these men to my development as a scholar.

I would also like to thank the United States Army for affording me this opportunity. I specifically would like to acknowledge Major General John Ferrari and Colonel Douglas Matty for their efforts to provide such incredible opportunities to those within the Operations Research branch. I would also like to thank the countless superiors, peers, and subordinates who's example and selfless service motivate this work.

Learning how to receive mentorship from those often 15 years my junior has been a blessing in its own right. This 42 year old soldier could not have transitioned to scholar without the patient support of my colleagues within CASOS: Geoff Dobson, Will Frankenstein, Binxuan Huang, Kenny Joseph, Mike Kowalchuck, Sumeet Kumar, Geoff Morgan, Jeff Reminga, and Wei Wei.

I also would like to thank my parents, in-laws and brother for their unquestioning support. Above all, I thank my wife Tammy whose sacrifices only I will truly know. Without her support none of this would have been possible.

Abstract

Online social networks have become a powerful venue for political activism. In many cases large, insular online communities form that have been shown to be powerful diffusion mechanisms of both misinformation and propaganda. In some cases these groups users advocate actions or policies that could be construed as extreme along nearly any distribution of opinion, and are thus called Online Extremist Communities (OECs). Although these communities appear increasingly common, little is known about how these groups form or the methods used to influence them. The work in this thesis provides researchers a methodological framework to study these groups by answering three critical research questions:

- How can we detect large dynamic online activist or extremist communities?
- What automated tools are used to build, isolate, and influence these communities?
- What methods can be used to gain novel insight into large online activist or extremist communities?

These group members social ties can be inferred based on the various affordances offered by OSNs for group curation. By developing heterogeneous, annotated graph representations of user behavior I can efficiently extract online activist discussion cores using an ensemble of unsupervised machine learning methods. I call this technique Ensemble Agreement Clustering. Through manual inspection, these discussion cores can then often be used as training data to detect the larger community. I present a novel supervised learning algorithm called Multiplex Vertex Classification for network bipartition on heterogeneous, annotated graphs. This methodological pipeline has also proven useful for social botnet detection, and a study of large, complex social botnets used for propaganda dissemination is provided as well.

Throughout this thesis I provide Twitter case studies including communities focused on the Islamic State of Iraq and al-Sham (ISIS), the ongoing Syrian Revolution, the Euromaidan Movement in Ukraine, as well as the alt-Right.

Chapter 1: Introduction

The emergence of Online Social Networks (OSNs) as publication and news delivery platforms Chu et al. (2010) now offers an open and powerful marketing domain with a global, multi-billion user audience Statista (2016a,b). These large OSNs like Facebook, Twitter, and VKontakte often share URLs linking users to an array of online media sources so vast as to exhaust fact-checking resources **?**. The result is a powerful marketing domain where users' ability to evaluate trustworthiness is extremely difficult Ferrara et al. (2016a). Deceptive use of these platforms for geopolitical gain has been observed in traditional political campaigns Ratkiewicz et al. (b) as well as populist revolutions in Eastern Europe Diuk (2014) and the Middle East **?**. The growth of online communities who supporting extremist has emerged in a variety of areas as well, most notably being the Islamic State of Iraq and ash-Sham's (ISIS). Furthermore, the emergence of nationalist populist movements like Brexit (Mangold, 2016) and the "alt-right" **?** have clearly leveraged OSNs as well. These observations mark a trend toward destabilizing, and at times inhumane, activist movements that are formed and fomented online. The role of social interaction in these online spaces and this tendency towards extremism motivates this work.

OSNs provide a unique social structure due to the relatively low cost of social ties Girvan and Newman (2002), and little is known about how these structures cognitively influence individual opinion. This topology appears in some cases to be not only favorable for the spread of propaganda, but these communities also appear to become insular as observed in the 2016 United States Presidential Election (Benkler et al.). Twitter proved to be a powerful conduit of ISIS propaganda as well, giving the group a global recruiting platform Berger and Morgan; Berger, JM; Lahoud et al. (2014). Other extremist groups have followed suit. Gaining understanding of these online communities requires a means to efficiently identify them at scale which is the focus of this thesis.

Online activism can be thought of on a continuum. Bloggers often promote awareness of an identifiable cause. Then that cause is either political or religious, we refer to this behavior as online activism. When the opinions expressed or groups endorsed would fall on the tails of nearly any distribution of opinion, that online activism meets the definition of online extremism. This thesis is particularly interested in the online communities which contribute tot he diffusion of onlnie extremism. I call these groups online extremist communities (OEC) and define them as follows:

> **Online Extremist Community (OEC):** A social network of users who collectively engage in online activism with identifiable discussion cores engaging in online extremism.

Governmental efforts to effectively counter extremist propaganda within OSNs have not been successful. Chapter 2 presents a case study of a large activist community on Twitter that shared content related to the ongoing Syrian Revolution. We refer to this community as the Syrian Revolution Twitter Community (SRTC) and it contains just over 15,000 Twitter users. Each member's timeline exhibits some level of interest or support for ongoing terrorist operations in Syria and Iraq. This chapter provides illustrative intelligence extractions highlighting the ability to gain high-level insight through social media. The chapter also highlights methodological challenges which motivate many of the subsequent chapters of this work.

These groups appear to self organize by utilizing many of the affordances offered by Twitter to include following, mentioning, and using hash tags. The result is a complex network of social behaviors that can be mined. Chapters 3 through 6 present a methodological pipeline for OEC detection and analysis as depicted by Figure 1.1. Throughout this thesis I will detect OECs of interest by modeling user behavior as a heterogeneous graph. Each dataset in this work is instantiated using an n-hop snowball sampling strategy Goodman (1961) with manually identified members of an OEC of interest as 'seed agents.' The union of seed agents' social ties define the search. This technique is then iterated in steps, which has been done in each datasets presented in this work. Although this technique is not random and prone to bias, it is often used when trying to sample hidden populations Berger and Morgan (2015b). We discuss this search technique in detail in Appendix A as well as each of the datasets used within this work. Given a large snowball sample, unsupervised methods are used to develop a training set, and OEC detection is performed as a classification task.

In Chapter 3 I present two methodological extensions for OEC detection. The first Iterative Vertex Clustering and Classification (IVCC) discusses the iterative approach illustrated by the unsupervised OEC detection and supervised OEC detection boxes in Figure 1.1. The classification algorithm, Multiplex Vertex Classification, is novel as well. I use spectral methods to develop graph-based features for each graph within a heterogeneous representation of users' following, mentioning, and hash tag behaviors. We evaluate performance with a case study of the ISIS supporting OEC on Twitter. IVCC detected over 20,000 ISIS supporters with approximately 5000 heuristically generated positive case training examples with F1 scores over 95%. Positive case training instances within the ISIS case study were identified heuristically by combining unsupervised methods and Twitter suspension patterns to infer ISIS support. This chapter also provides a detailed overview of the societal implications of research methods similar to those presented in this work.

As accurate metadata is not usually available for supervised learning, unsupervised alternatives to identify OEC discussion cores can provide a useful alternative. In Chapter 4 I generalize the findings of Chapter3 to the case where informative meta-data is unavailable. I introduce to methods for detecting OEC discussion cores using dense subgraph methods. The dense subgraph problem has been studied extensively, but I know of no study which applies these methods on a heterogeneous graph. The first, heterogeneous dense subgraph detection (HDSD) extends the work presented by Chen and Saad (2012). We develop a user similarity graph based on our heterogeneous representation of our OSN data and use it to develop a bottom-up user hierarchy. Like (Chen and Saad, 2012) we then search the hierarchy top down based on an a priori minimum density threshold. The second methodology, ensemble agreement clustering (EAC), takes a different approach entirely. EAC develops unique user clusterings for undirected and bipartite user graphs and returns only groups where users are co-clustered across all edge types. I present both case studies and partially simulated results using Twitter data.

Extracting information and mining knowledge from large OECs is a challenging problem as well. In fact it is arguably a research area unto itself. Many traditional methods prove useful in this regard, and Chapter 5 presents three existing methods to mine novel insight from OECs. The first, bispectral graph partitioning Dhillon (2001a), is a useful method to cluster user behavior based on hash tag use. We also present the utility of hash tag co-occurrence graph clustering to summarize community discussion over time. Finally we illustrate the utility of mining Twitter data for specific user behaviors by using URL sharing graphs to identify online recruiting Berger (2014).

As OSNs have become a major venue for news consumption, techniques to use automated social actors or "bots" to manipulate public opinion have become increasingly sophisticated. While developing Chapters 2 through 5 I have consistently observed socialbot networks designed for this purpose. In Chapter 6, I define two specific classes of socialbot networks called Mention Community Socialbot Networks and Cyborg Socialbot Networks. These structures capitalize on the robust application programmer interface provided by Twitter's and have the ability to artificially manipulate many network science measures used to estimate the credibility or social influence of specific users and posts. These artificial promotion methods appear able to not only promote users and content, but also could increase diffusion within or across online communities. It is becoming increasingly important to understand how these structures promote the diffusion of propaganda and misinformation online. In addition to defining both classes of socialbot network, I present a novel, graph-based socialbot network detection methodology. I also provide several case studies of detected socialbot networks used in United States political discourse, the Syrian Revolution, and the Euromaidan Movement.

This thesis can be thought of as a pipeline of methods used to gain insight into online activist and extremist communities. Figure 1.1 summarizes a sequence of methods that will enable researchers to start with a small set of online activists, identify the larger online community they belong to, and extract novel insights into the detected community. Appendix A discusses our collection methods in detail as well as the datasets utilized in this work. Because my collection methods often return large numbers of users who are not members of the community of interest, I find that one must frame community detection as a supervised learning problem. The first challenge then is to develop an sufficiently large training set. Chapter 4 presents an unsupervised method that can be used to identify OEC discussion cores which can be used to quickly develop large training sets. The output from those methods are positive case and negative case instances of our community of interest which then can be used to detect the community at scale as a supervised learning task. Chapter 3 provides the detection framework able to accurately detect these communities as scale. Given a large, detected online extremist community, several existing methods may be used to mine novel insight. Several of these methods are presented in Chapter 5 and a study of Socialbot Network behavior is provided in Chapter 6.

In addition to these contributions, I provide an in depth discussion of limitations and propose future research in Chapter 7. In summary this thesis presents a framework for detection and analysis of OECs at scale and provides an important contribution in this important emergent area of community detection. My hope is that it not only motivates future research, but also emphasizes the need for ongoing collaboration among computational and social scientists.

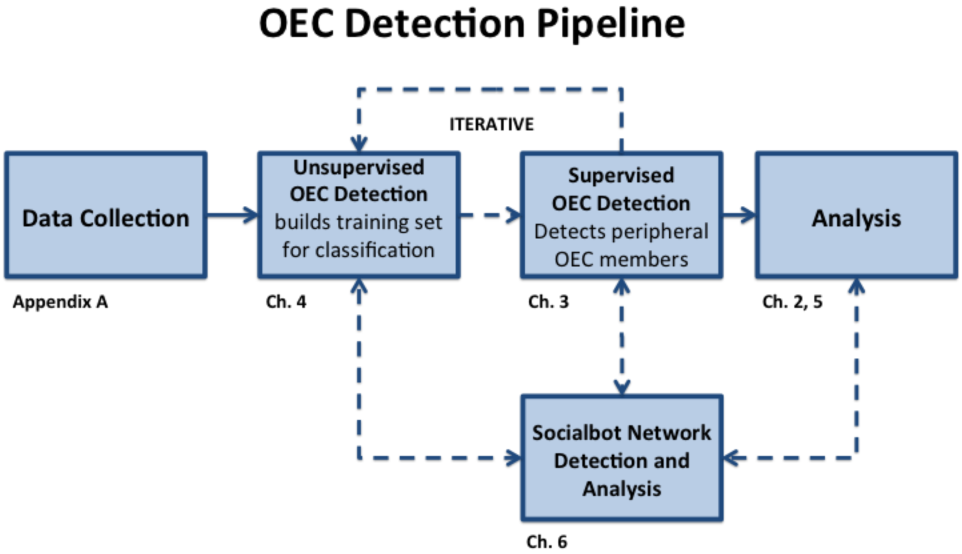

OEC Detection Pipeline
ITERATIVE
Data Collection
Appendix A
Unsupervised OEC Detection builds training set for classification
Ch. 4
Supervised OEC Detection Detects peripheral OEC members
Ch. 3
Analysis
Ch. 2, 5
Socialbot Network Detection and Analysis
Ch. 6

Figure 1.1: This thesis presents OEC detection as a methodological pipeline. Within each chapter I collect data using snowball sampling which is discussed in detail in Appendix A. Then given a large snowball sample, I use unsupervised methods to develop a training set and detect the OEC of interest as a classification task.

Chapter 2: From Tweets to Intelligence: Understanding the Islamic Jihad Supporting Community on Twitter

2.1 Introduction

Extremist groups' powerful use of online social networks (OSNs) to disseminate propaganda and garner support has motivated intervention strategies from industry as well as governments however early efforts to provide effective counter-narratives have not produced the results desired. Mr. Michael Lumpkin, the director of the United States Department of State's Center for Global Engagement, is charged with leading efforts to "coordinate, integrate, and synchronize government-wide communications activities directed at foreign audiences in order to counter the messaging and diminish the influence of international terrorist organizations" Dozier (2016). In a recent interview, Mr. Lumpkin expressed the need for a new approach:

> "So we need to, candidly, stop tweeting at terrorists. I think we need to focus on exposing the true nature of what Daesh is."
>
> Mr. Michael Lumpkin, NPR Interview March 3, 2016

A logical follow-up question to Mr. Lumkin's statement would be "Expose to whom?" Recent literature suggests that "unaffiliated sympathizers" who simply retweet or repost propaganda represent a paradigmatic shift that partly explains the unprecedented success of ISIS Berger, JM; Veilleux-Lepage (2015) and could be the audience organizations like the Global Engagement Center need to focus on. Gaining understanding of this large population of unaffiliated sympathizers and the narratives most effective in influencing them motivates methods to detect and extract information from large online extremist communities (OEC). However, detecting, monitoring, and data-mining targeted OTGSs requires novel methods, and development must include both data science and regional expertise. We define data science as a set of fundamental principles that support and guide the principled extraction of information and knowledge from data, and in this paper we present the Syrian Revolution Twitter Community (SRTC), a online community of over 15,000 Twitter users who support one or more of the radical groups engaged in the ongoing conflicts in Northern Iraq and Syria. We describe how large OECs can offer unique insights into the unaffiliated supporters who appear critical to ISIS' success. We then provide an example of one method used to excite and grow these OGTSCs in the form of an active social botnet. The botnet attempts to elevate the social influence of users supportive to Jabhat al-Nusra's agenda, while encouraging following ties amongst botnet followers. Our goal is to present two novel examples of social computing applied to counterterrorism, and motivate the continued interdisciplinary collaboration required to gain understanding of large online communities and effectively counter extremist propaganda .

2.2 The Syrian Revolution Twitter Community (SRTC)

On November 13, 2015 much of the world watched as terrorist launched a series of coordinated attacks in Paris killing 130 people. In near real-time social media erupted with support for the victims of these attacks, but some online communities viewed the attacks as cause for celebration. In fact, passive supporting but unaffiliated social media users have become an essential element of groups like ISIS and Jabhat al-Nusra's recruiting strategy, possibly aid the motivation and resourcing for attacks like those seen in Paris Veilleux-Lepage (2015). Large online social networks like Twitter offer a means to generate large online communities, and many of the members appear to be "unaffiliated supporters." In fact, Twitter has suspended over 125,000 ISIS-supporting accounts from August to December of 2015. As ISIS recruiters identify community members who show increasing levels of radicalization, small teams of social media cadre have been observed lavishing attention on these recruitment targets and subsequently move the conversation to more secure online platforms Berger, JM. Less secure but large open platforms like Twitter enable extremist groups propaganda to gain broad reach. Denying this *key terrain* requires novel methods designed specifically to identify and analyze extremist communities embedded in OSNs. Information like key users, powerful narratives, and advanced dissemination methods can all be extracted from OECs to inform messaging and intervention strategies. Benigni et. al. present Iterative Vertex Clustering and Classification Benigni, Matthew et al., a novel method to detect large, ideologically organized online communities, using both agent level attributes and network structure. We briefly present the methodology, introduce the SRTC, provide illustrative analysis of the network, and share ongoing research goals in this section.

2.2.1 Background: From Community Detection to Threat Network Detection

The application of network science to counter-terrorism has a long historyCarley (2006); Krebs (2002a); however, the rise of social media and online social networks (OSNs) has motivated methods to apply network science theory to networks at much larger scale. Community detection attempts to identify groups of vertices more densely connected to one another than to other vertices in a network, but networks extracted from OSNs present unique challenges due to their size and high clustering coefficients. Furthermore, an individual's social network also often reflects his or her membership to many different social groups. Thus in many instances algorithms that use only network structure do not provide the precision needed to identify OECs Benigni, Matthew et al.. A sub-class of community detection methods has emerged that attempts to leverage node attributes and network structure called community detection in annotated networks. These methods have been shown to perform well with OSNs because of their ability to account for a great variety of vertex features like user account attributes while still capitalizing on the

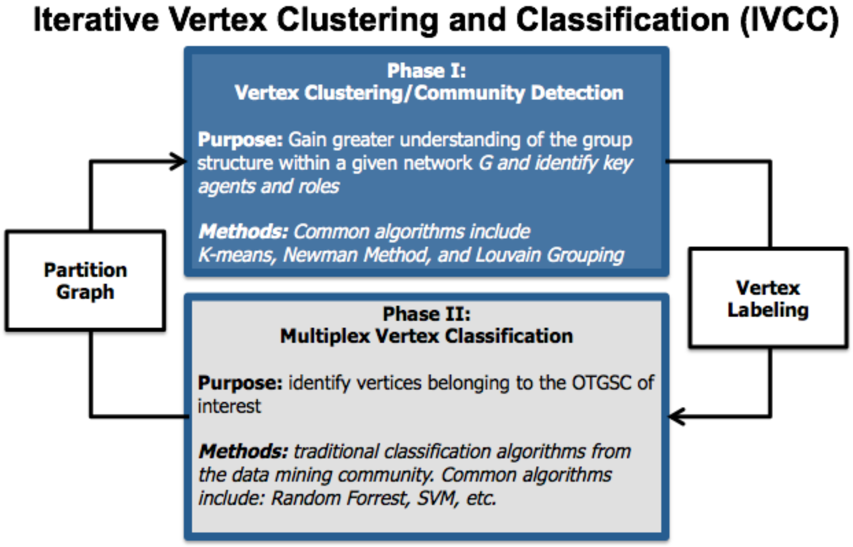

Iterative Vertex Clustering and Classification (IVCC)
Phase I:
Vertex Clustering/Community Detection
Purpose: Gain greater understanding of the group structure within a given network G and identify key agents and roles
Methods: Common algorithms include K-means, Newman Method, and Louvain Grouping
Partition Graph
Vertex Labeling
Phase II:
Multiplex Vertex Classification
Purpose: identify vertices belonging to the OTGSC of interest
Methods: traditional classification algorithms from the data mining community. Common algorithms include: Random Forrest, SVM, etc.

Figure 2.1: IVCC is an online extremist community (OEC) detection methodology conducted in two phases. In Phase I either community optimization or vertex clustering algorithms are used to identify positive and negative case examples to facilitate supervised detection in Phase II.

information provided by the structure of the graph; they also perform well at scale Binkiewicz et al. (2014); Tang and Liu (2011). However, we find that effective OEC detection requires information from users' following, mention, and hashtag behaviors as well. Benigni et. al. present IVCC, an community detection method designed to extract OECs by modeling users within a heterogeneous graph structure with annotated nodesBenigni, Matthew et al..

2.2.2 Iterative Vertex Clustering and Classification

Iterative Vertex Clustering and Classification (IVCC) is conducted two phases, and often iteratively. In Phase I, unsupervised clustering methods like Newman and Louvain grouping are used to both identify positive cases labels and remove noise. This pre-clustering facilitates supervised classification of OEC members in Phase II. At the core of the methodology is the use of both user level features and rich multiplex network structures offered by OSNs. First the authors construct $U_{u\times a}$ consisting of a numeric user attributes where u is the total number of users or nodes in the network. Examples of such attributes are follower count, number of posts, or creation date. Node attributes could also be developed from other sources of intelligence. Spectral methods are used to dimensionally reduce network data like following, mention, or hashtag behaviors. By constructing symmetric graphs of users' following F and mention M relationships, and a weighted bipartite graph H of hash tags in a user's timeline, lead eigenvectors can then be extracted from each graph and concatenated with U to form a feature space for classification. Although IVCC is presented using Twitter data Benigni, Matthew et al., similar methods could be used more generally with large heterogeneous networks.

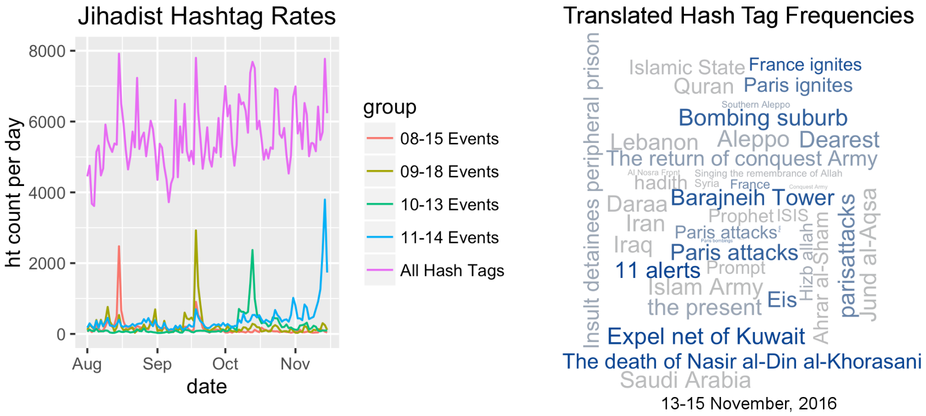


Figure 2.2: The left panel depicts the volume of hashtags used within the SRTC from AUG-NOV 2015. The right panel highlights the hashtags most explanatory of the increased activiy on November 14, 2015.

Benigni et al. collected a two-hop snowball sample of five popular ISIS propagandists presented in Carter et al. (2014), resulting in approximately 120,000 Twitter users. With two iterations of IVCC, they removed accounts with high following counts (i.e. politicians, news media members, celebrities, etc.), and extracted a network of nearly 23,000 *ISIS supporters*. The results of this initial work form the seed accounts for the SRTC.

2.2.3 Threat Network Analysis: The SRTC

CASOS is currently extending IVCC to dynamically monitor extremist online communities. By using historical results and active learning, we update the SRTC based on the recent community activity. Currently the community contains just over 15,000 supporters, where we define a supporter as a Twitter user who positively affirms the leadership, ideology, fighters, or call to Jihad of any of the known Jihadist groups engaged in ongoing operations in Northern Iraq and Syria. The majority of tweeters voice support for ISIS or Jabhat al-Nusra though other groups are present. The size of this community offers insights not easily gleaned from randomly sampled Twitter data or manually developed datasets as will be highlighted in the remainder of this section.

Though many demographical analyses could be useful, for conciseness we will use temporal network activity patterns to illustrate information extraction from OECs. The Twitter REST API limits collection to a tweeter's last 3,200 posts which forces us to normalize daily volume. Some tweeters have more than 3,200 posts in the past 6 months, and quite a few of our tweeters have not posted in over 90 days. Identification of dormant users could provide insight into the radicalization process, but will not be analyzed or discussed in this work. We estimate the SRTC's daily volume by normalizing based on the number of tweeters in our dataset who have a collected tweet before and after any given day which often highlights current events that stimulate this community. A simple news search of events on days of increased activity often reveals operational events in Syria, Northern Iraq, or large scale terror attacks. Similar analysis of hash tag trends often provides richer insight. Figure 2.2 highlights temporal analysis of SRTC hash tag use. The left panel of depicts hash tag frequencies over time, while the right panel depicts trending hashtags on 13-14 November, 2015. Size in the word cloud connotes frequency, and color denotes how anomalous a particular tag's frequency was when compared to a 6 months average. The community's reaction to the 13 November, 2015 Paris attacks is illustrated with both increased volume and trending hashtags. Increased hash tag volume depicted in the left panel of Figure 2.2, coupled with the corresponding hash tag trends in the right panel give startling insight into the unique nature of this online community. Ongoing operations in Syria provide another example. The hash tag ـسـمكـمـع مو مو ـوـق, translated Zabadani, increases tenfold in terms of daily frequency on 15 August and 18 September, 2015. Both dates refer the breakdown of ceasefire agreements in the region Perry (2015). With proper subject matter and language expertise, similar analysis can identify changes in popularity of leaders, organizations, or narratives over time.

2.2.4 Moving Forward

As a supervised learning methodology, IVCC lends itself to leveraging regional expertise by learning patterns based on examples. Active-learning refers to supervised algorithms that iteratively select examples to be labelled by experts, and have been found substantially increase performance with far fewer labelled instances. Such methods could enable regional expertise to be incorporated into the classifier at minimal cost. Furthermore, a user-oriented, server-based interface could enable the regional expert to contribute to the set of annotated instances while conducting his or her own exploratory data analysis. As the set of annotated examples or "training set" grows new , more nuanced classifiers could be trained. Due to the size and diversity of these online communities, exploration and interpretation of results is likely a research area unto itself. One could identify the news sources or propagandists most influential within these communities, and develop more-informed counter-narratives and strategic communications strategies. The challenge in developing tools and methods to facilitate OEC analysis lies in the novelty of the analytical task. Regional experts cannot yet articulate exactly what they want methods to provide, and researchers are challenged to understand what information extractions are most useful to senior leader information requirements. Establishing online tools that provide illustrative analyses and capture feedback while end users to explore large communities would likely accelerate research efforts aimed at countering groups like ISIS.

2.3 The FiribiNome Social Botnet: sophisticated promotion of propaganda to excite a community

While analyzing the SRTC, as well as a similar dataset focused on online dialogue focused on the Russian occupation of Crimea, we observe accounts that tweet with high daily volume, but each tweet or retweet simply contains a string of @mentions. In this section we analyze a network of social bots used to promote specific online activists or propagandists.

Social bots, software automated social media accounts, have become increasingly common in OSNs. Though some provide useful services, like news aggregating bots, others can be used to shape online discourse Abokhodair et al. (2015). ISIS' use of bots has been well documented Berger (2014), and their competitors are following suit. Social botnets are teams of software controlled online social network accounts designed to mimic human users and manipulate discussion by increasing the likelihood of a supported account's content going viral. The use of

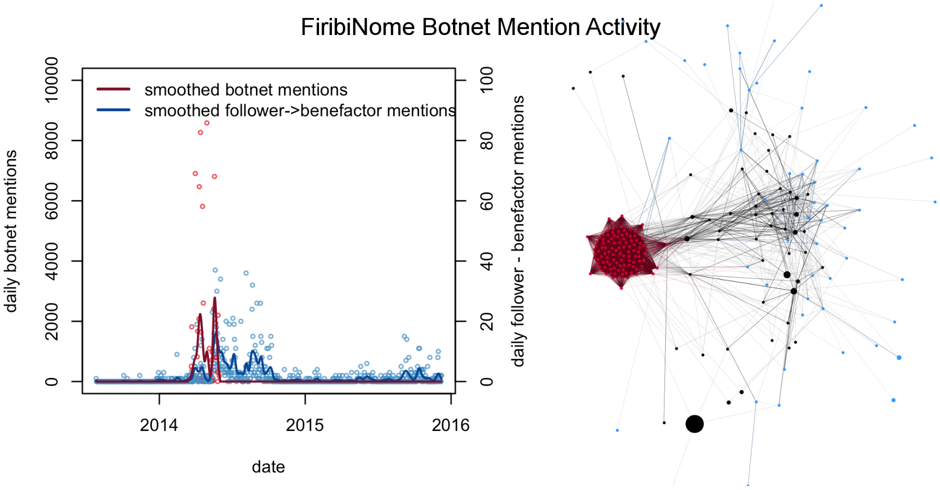


Figure 2.3: Depicts mention behaviors and their effects within the FiribiNome Social Botnet. The left panel depicts two scaled time series. The red circles and smoothed trend line depict the number of daily mentions by botnet members. The blue circles and corresponding trend line depict botnet followers' mentions of benefactor accounts. The association between the two series implies the botnet was able to generate discussion about benefactor accounts among its followers. The right panel depicts the mention network of the FiribiNome social botnet. The vertices are user accounts. The plot depicts how *botnet members*, red vertices, are used to increase the social influence of *benefactors*, black vertices, by promoting them to *botnet followers*, blue vertices. Vertices are scaled by follower count.

bots to influence political opinion has been observed in both domestically Ferrara et al. (2014) and abroad Forelle et al. (2015), the use of social bots has been documented in the MENA region Abokhodair et al. (2015), and ISIS use of them motivated a DARPA challenge to develop detection methods Subrahmanian et al. (2016). In isolation, these accounts appear to be producing spam and relatively harmless, however they are examples of a sophisticated strategy to promote specific accounts while remaining undetected by Twitter.

2.3.1 SRTC Botnet Analysis

Figure 2.3 depicts the mention activity associated with the a Jabhat al-Nusra supporting social botnet designed to increase the social influence of a specific set of accounts and encourage following connections between Jabhat al-Nusra supporting tweeters. The botnet consists of two types of accounts. *Botnet members*, are depicted by red vertices in the right panel of Figure 2.3, and consist of 74 accounts exhibiting near identical behavior. Each account follows between 116 and 134 accounts, most of which are *botnet members*. Their following counts vary from 142 to 322 accounts of which many appear to be real tweeters. They come online for 38-58 days, tweet between 71 to 170 times, then go dormant. This behavior can clearly be seen by the red trend line in Figure 2.3. Their tweets consist of original posts or retweets containing strings of @mentions of other *botnet members*, but occasionally mention or retweet content from what we call *benefactor accounts* (depicted by black vertices in the right panel of Figure 2.3). The *botnet* account FiribiNome20 illustrates this behavior. In isolation, these accounts appear to be producing spam and relatively harmless, however our analysis indicates the network of *botnet members* increases the social influence of *benefactor accounts*. The blue series in the left panel of Figure 2.3 and corresponding blue vertices in the right panel depict the mention activity of the 843 active botnet followers as of February 2016. The left panel depicts *follower* accounts' mentions of *benefactor* accounts and the temporal relationship between the activity associated with each account type implying the botnet effectively promotes discussion of *benefactor* accounts. How much discussion is generated remains an open question. Due to the large number of extremist accounts suspended by Twitter, the number of *botnet followers* active in the summer of 2014 was likely much larger. This mention behavior exhibited by *botnet members* could also trigger Twitter's recommendation system to recommend following ties between *botnet followers*,or encourage *botnet followers* to follow *benefactors*.

Examples of *benefactor accounts* are depicted in Table 2.1; each representing a slightly different style and type of messaging commonly observed in the SRTC. Dr. Hani al-Sibai is a London-based radical Islamic Scholar cited by Ansar al-Sharia as one of five influential motivators of Tunisian terroristsTunisias and Game (2013). *@ba8yaa* or "Daesh are the Enemy " attempts to discredit ISIS through satire and counter-propaganda and could prove informative in development of counter-narratives. There are also many accounts that present the appearance of reporting near-real-time news like *@Ghshmarjhy*, while other accounts promote third party applications like @Almokhtsar and @FiribiNome12. We have found some of these applications request permission to tweet or follow users on the tweeter's behalf. These highly followed and highly mentioned accounts each could offer insight into the sophisticated methods used to leverage social media.

Account	Follower Count	Messaging Type
@Hanisibu	104K	Islamic Scholar
@ba8yaa	1,272	anti-ISIS satire/propaganda
@Ghshmarjhy	6,644	Syrian revolution updates
@Almokhtsar	164K	app: MENA news feed

Table 2.1: Depicts four account promoted by the FiribiNome social botnet. Each account represents a slightly different style and type of messaging.

2.3.2 Moving Forward

It is possible that botnet structures with similar mention behavior could be developed in a more sophisticated manner. Larger networks with more human-like behavior would be much more challenging to detect. The FiribiNome botnet could simply represent a proof of concept explaining its lack of activity since 2014. Although simple heuristics like average mentions per tweet enabled us to detect the botnet, more advanced detection strategies are needed to determine if more sophisticated botnets are influencing the SRTC. Methods of operationalizing this type of intelligence are worth exploring as well. It is possible that similar mention behaviors could be used to target specific online communities with counter-narratives. Again, the need for an interdisciplinary collaboration between the data scientist, regional expert, and decision maker is needed to identify opportunities for useful intelligence extraction.

2.4 Conclusion

We have highlighted the potential of extracting intelligence from large online extremist communities (OECs) and presented illustrative examples with a goal of motivating continued interdisciplinary collaboration. We have also presented the SRTC dataset as an example of an OEC to emphasize how detecting and monitoring extremists can be an important tool in understanding the passive support structure essential to the distribution of extremist propaganda. Furthermore, these methods could facilitate identification of sophisticated dissemination techniques used in these communities and inform our own information operations. Our goal is to refine these methods and grow a consortium of data scientists, regional experts, and strategic decision makers by hosting, curating, and reporting on datasets like the SRTC.

Chapter 3: Online Extremism and the Communities that Sustain It: Detecting the ISIS Supporting Community on Twitter

Introduction

Through an effective social media campaign, the Islamic State of Iraq and ash-Sham (ISIS) has issued a powerful, global call to arms. On Youtube, Twitter and a host of other social media platforms, an ethnically diverse set of Jihadists issue a similar call:

> IFLm calling on all the Muslims living in the West, America, Europe, and everywhere else, to come, to make hijra with your families to the land of Khilafah....Here, you go for fighting and afterwards you come back to your families. And if you get killed, then ... youFLll enter heaven, God willing, and Allah will take care of those youFLve left behind. So here, the caliphate will take care of you. (Stern, Jessica and Berger, JM (2015))

Online Extremism can be defined as advocating support of groups or causes that in any distribution of opinion would lie on one of the "tails" Lake (2002). With respect to ISIS' barbarous online marketing campaign, the amount of online activity generated by their activism has been shocking, and its effect in the offline world has been significant. As of January, 2015, United States intelligence sources estimate ISIS had between 9,000 and 18,000 fighters in Iraq and Syria Starr (2015). Although the majority of ISIS' fighters are from the Middle East and North Africa (MENA), a surprising number of fighters have arrived from the Western world. ISIS' message has global reach and has even motivated lone wolf attacks in Canada Logan (2014), France Wikipedia (2015), and the United States Yan (2015).

Not all members of ISIS' online community display the same levels of online extremism. Some claim unaffiliated sympathizers who simply retweet or repost propaganda represent a paradigmatic shift explaining ISIS' unprecedented online success Berger and Morgan (2015b); Berger, JM; Veilleux-Lepage (2014, 2015). In many cases these unaffiliated users' activity, although offensive to many, is not in clear violation of law or "The Twitter Rules twi." However, this large body of "passive supporters" contribute to the volume of ISIS related content proliferated on Twitter and appears to be a vital component of ISIS social media campaign. These individuals are therefore of interest to any effort to counter online extremism. Some of these passive sympathizers become recruiting targets. ISIS uses small teams of social media users to lavish attention on the potential recruits and move the conversation to more secure online platforms Berger, JM. Thus, while Twitter may not be the place where recruitment ends, growing evidence suggests that identifiable patterns of recruitment *begin* on Twitter.

The primary goal of this work is to provide methods allowing researchers to gain insight into this online social network of unaffiliated sympathizers, propagandists, fighters and recruiters, and how these users interact to create a thriving *online extremist community* (OEC). We argue that such understanding is needed to create counter-narratives tailored to the online populations most vulnerable to this type of online extremism. To do so, we must first solve another problem - identifying an OEC on Twitter. This task is difficult for three reasons. First, the size of OECs varies and is often unknown. With respect to ISIS, it has been estimated that the OEC is between 46,000 and 70,000 strong Berger and Morgan (2015b). However, the relatively small intersection between existing datasets maintained by activists and researchers indicates the group could in fact be much larger. Second, current social media community detection methods require a great deal of manual intervention, or provide unacceptable precision via automated methods - there is thus an existing tradeoff between manual coding of the data and highly inaccurate classification tools in the existing literature.

As ISIS' popularity has grown, so too has its opposition; thus the ISIS OEC and extremist groups in general tend to be *covert* in that they actively attempt to avoid some form of detection. Twitter now systematically identifies and suspends user accounts associated with the group Ross et al. (2015). In fact, Twitter has initiated a systematic campaign to neutralize ISIS' use of the site and announced in March of 2016 the suspension of over 125,000 ISIS supporting accounts in a six month period Calamur (2016). Furthermore, activist groups like Anonymous and Lucky Troll Club have used crowd sourcing to identify and expose ISIS OEC members on Twitter Gladstone (2015a,b); Poe. These attempts to limit ISIS' use of social media platforms has resulted in a predator-prey-like system where the ISIS OEC on Twitter has begun show systematic attempts to make the network anonymous and resilient.

Our work makes three major contributions to the literature. First, we present Iterative Vertex Clustering and Classification (IVCC), a novel approach to detect and extract knowledge from OECs. Our approach utilizes community optimization methods in conjunction with *multiplex vertex classification (MVC)*, a classification method used on heterogeneous graphs that leverages the rich data structures common to many OSNs like user meta-data, mentioning, following, and hash tag use.Capitalizing on this rich structure enables us to outperform existing methods with respect to recall and precision which will be shown in Section 6.5.

After considering the merits of our approach, we then turn to the second major contribution of this work, an illustrative case study of the ISIS OEC on Twitter. By searching known members' following ties and partitioning the resultant network, we identify a community of over 22,000 Twitter users whose online behavior contributes to the online proliferation of ISIS propaganda. We leverage clustering and Twitter suspensions to infer positive case instances with our classifier which is able to partition our training set with 96% accuracy. This offers significant improvement over existing methods, and we claim this makes our output uniquely valid for the study of online radicalization. A sample de-identified dataset Benigni and an R tutorial Benigni (2017) are available as well.

Finally, we discuss an ethical framework for the implementation of methods similar to IVCC. We highlight the framework presented in Walsh and Miller (2016) of: methods, context, and target, and we draw distinctions in context between diplomatic and intelligence applications of social media mining.

We structure this article as follows: In Section 3.1 we discuss related work and highlight the limitations of common community detection methodologies with respect to OEC detection. Section 3.2 provides a detailed overview of our proposed community detection methodology, followed by an illustrative case study of the ISIS OEC on Twitter in Section 3.3. Section 6.5 provides a detailed discussion of the relative performance of IVCC, and Section 3.5 provides a case study of the ISIS supporting OEC on Twitter and illustrative knowledge extractions useful for counter-messaging or intelligence purposes. We then discuss the societal implications and limitations associated with the potential uses of our methods in Section 3.6, and propose future research in Section 5.6.

3.1 Background

Krebs Krebs (2002a,b) was the first to cast large-scale attention on network science-based counter-terrorism analysis with his application of network science techniques to gain insight into the September 11, 2001 World Trade Center Bombings. Although similar methods were presented years earlier Carley et al. (1998), the timeliness of Krebs' work caught the attention of the Western world and motivated a great deal of further researchCarley (2006); Carley et al. (2003); Diesner and Carley (2004); Koschade (2006); Latora and Marchiori (2004); Ressler (2006); Top (2009). Much of this work focused on constructing networks based on intelligence and using the network's topology to identify key individuals and evaluate intervention strategies. The rise of social media has introduced new opportunities for network science-based counter-terrorism, and some foresee social media intelligence *(SOCMINT)* as being a major intelligence source in the future Harman (2015). This presents a fundamentally different counter-terrorism network science problem. Roughly, as opposed to using information about individuals to build networks, we now use networks to gain insight into individuals. Typically, we are also trying to identify a relatively small and possibly covert community within a much larger network. Such a change requires methodologies optimized to detect covert networks embedded in social media.

The problem of community detection has been widely studied within the context of large-scale social networks Papadopoulos et al. (2011). Community detection algorithms attempt to identify groups of vertices more densely connected to one another than to the rest of the network. Social networks extracted from social media, however, present unique challenges due to their size and high clustering coefficients Girvan and Newman (2002). Furthermore, ties in online social networks like Twitter are widely recognized to represent different types of relationships Boccaletti et al. (2006); Joseph and Carley (2015); Miller et al. (2011a); Wang et al. (2010).

The algorithms of Newman Newman (2006) and Blondel Blondel et al. (2008) are recognized as a standard for comparison for community detection within network science. Within the broad landscape of all community detection algorithms, the work of both Newman and Blondel fall under the umbrella of what is more accurately referred to as community optimization algorithms. In community optimization algorithms, the graph is partitioned into k communities based on an optimization problem that centers around minimizing inter-community connections are minimized and k is unspecified. Surprisingly, both Newman and Blondel operationalize this minimization problem as a maximization one, where they maximize *modularity*. The modularity of a graph is defined in Equation C.1. In Equation C.1, the variable $A_{i,j}$ represents the weight of the edge between nodes i and j, $k_i = \sum_j A_{i,j}$ is the sum of the weights of the edges attached to vertex i, c_i is the community to which vertex i is assigned, $\delta(u, v)$ is the inverse identity function, and $m = \frac{1}{2} \sum_{i,j} A_{i,j}$.

$$Q = \frac{1}{2m} \sum_{i,j} = [A_{i,j} - \frac{k_i k_j}{2m}] \delta(c_i, c_j), \tag{3.1}$$

Eaton and Mansbach Eaton and Mansbach (2012) have introduced methods from constrained clustering literature to enable semi-supervised community optimization where a subset of vertices have known memberships as well. While such algorithms work well for certain classes of problems, community optimization algorithms have shown limited ability to detect threat networks embedded in social media when the group maintains connections with non-group members Miller et al. (2011a). Community optimization is also unable to effectively account for multiplex graphs or graphs with multiple connection types. Like community optimization, graph partitioning finds partitions by minimizing intra-group connections; however, the number of groups, k, is fixed Papadopoulos et al. (2011). Covert network detection is then best described as a special case of graph partitioning where the partition is binary (or in other words, where $k = 1$) Smith et al. (2013). Smith et al. Smith et al. (2013) effectively use this viewpoint to model spatiotemporal threat propagation using Bayesian inference, however their method does not extent to multiplex or multimode graphs when applied to social media. To do so, other methods must be used.

In recent years, another sub-class of community detection methods has emerged, community detection in annotated networks. This body of work attempts to effectively incorporate node level attributes into clustering algorithms to account for noisiness of social networks embedded in social media. Vertex clustering originates from traditional data clustering methods and embeds graph vertices in a vector space where pairwise, Euclidian distances can be calculated Papadopoulos et al. (2011). In such approaches, a variety of eigenspace graph representations are used with conventional data clustering and classification techniques such as K-means or hierarchical agglomerative clustering, and support vector machines. These methods offer the practitioner great flexibility with respect to the types of information used as features. Vertex clustering and classification methods have been shown to perform well with social media because of their ability to account for a great variety of vertex features like user account attributes while still capitalizing on the information embedded in the graph; they also perform well at scale Tang and Liu (2011); Wang et al. (2010). Wang et al. (2010) introduces a vertex clustering framework, *SocioDim*, which detects communities embedded in social media by performing vertex clustering where network features are represented spectrally and paired with user account features. Very similar methods are also presented in Binkiewicz et al. (2014). Tang and Liu (2011) then applies *SocioDim* to classification, which is analogous to a binary partition of the graph.

These methods show clear promise with respect to covert network detection in social media as illustrated by Miller et al. (2011a). Eigenspace methods have been shown to adequately model multiplex representations of various types of social ties in social media Tang et al. (2009), and early studies of simulated networks indicate they would perform well on threat detection in social media Miller et al. (2011a). We hypothesize that eigenspace representations of multiplex social networks embedded in social media, when paired with user account features and node level features will provide a more powerful means to detect extremist communities embedded in social media. Our work utilizes community optimization across multiple graphs in an annotated heterogeneous network to facilitate vertex classification and detect a targeted covert community. In sum, we have found that each of the methods listed above offer useful information for classification, but a combination of these techniques must be used to effectively detect covert networks embedded in social media.

3.2 Methods:Iterative Vertex Clustering and Classification

The goal of finding an OEC within a larger dataset can be formalized as attempting to find a relatively small subgraph within a large, annotated, heterogeneous network, $G = (V_1, V_2, .., V_n, E_1, E_2, .., E_m)$. The full network G is a directed, weighted graph with vertex sets $V_1 ... V_n$. Each vertex set contains vertices $v_{n,1} .. v_{n,j}$ with one or more edge types $E_1, E_2, .., E_m$. We define a subset of targeted vertices $A_t \subseteq V_t$ and denote its complement as $\tilde{A}_t$. Our goal is to accurately classify each vertex in V_t as members of either A_t or $\tilde{A}_t$. For example, in our case study we define A_t as our set of *ISIS OEC members* and $\tilde{A}_t$ as the union of both *non-members* and *Official Accounts*, which will be defined below.

In practice, we will often have partial knowledge of the group and its members, and need to leverage as much information as possible to identify vertices in A_t. Our methodology leverages a priori knowledge to search for and detect a covert subgraph in social media by iteratively utilizing community optimization and vertex classification. Our approach is thus conducted in two phases. In Phase I, community optimization algorithms and a priori knowledge are used to gain insight into the larger social network and facilitate supervised machine learning in Phase II. Phase II partitions vertices, retaining only those in A_t, thus finding the targeted covert community. A diagram of the process can be seen in Figure 1.

3.2.1 Phase I: Vertex Clustering and Community Optimization

Although community optimization and vertex clustering methods will often fail to accurately partition our networks into A_t and $\tilde{A}_t$ Miller et al. (2011a), we can often look for community structure within the network to gain insight into the set of vertices in A_t. For example, if a subset of vertices from A_t is known, community optimization can identify clusters containing a large proportion of those known vertices belonging to A_t. Community optimization can also identify groups of vertices that are clearly members of $\tilde{A}_t$. The insights gained from community optimization help provide necessary context with respect to algorithm selection and case labels for vertex classification in Phase II of our methodology.

3.2.2 Phase II: Multiplex Vertex Classification

Like Tang and Liu (2011) we classify $v_{t,1} ... v_{t,j}$ using a set of features extracted from the users' social media profiles and spectral representations of the multiplex ties between V_t. We denote these spectral representations as $U_{V_t \times V_t; E_i}$, where $i = 1, ..., m$. To develop spectral representations of our heterogeneous network, we symmetrize the graphs $W = G_{V_n \times V_n; E_m}$ for $\forall E_m$. These symmetric graphs also leverage the strength of reciprocal ties, which have been shown to better

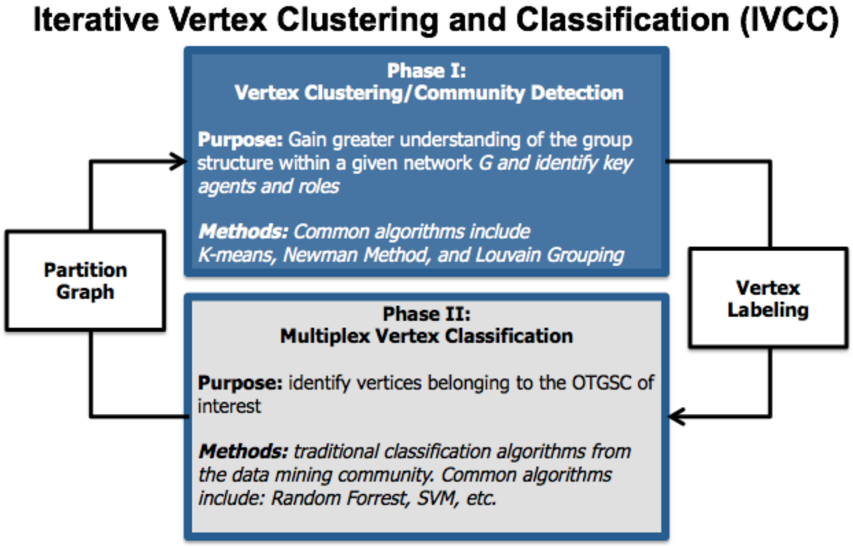

Iterative Vertex Clustering and Classification (IVCC)
Phase I:
Vertex Clustering/Community Detection
Purpose: Gain greater understanding of the group structure within a given network G and identify key agents and roles
Methods: Common algorithms include K-means, Newman Method, and Louvain Grouping
Partition Graph
Vertex Labeling
Phase II:
Multiplex Vertex Classification
Purpose: identify vertices belonging to the OTGSC of interest
Methods: traditional classification algorithms from the data mining community. Common algorithms include: Random Forrest, SVM, etc.

Figure 3.1: We present an iterative methodology conducted in two phases. In Phase I either community optimization or vertex clustering algorithms are used to remove noise and facilitate supervised machine learning to partition vertices in Phase II.

indicate connection in social networks embedded in social media Chiu et al. (2006); Gilbert and Karahalios (2009); Mislove et al. (2007). In our case study we refer to the symmetrized network of following ties as F_{rec}, and the symmetrized network of mention ties as M_{rec}. We then extract the eigenvectors of the graph Laplacian associated with the smallest two eigenvalues as highlighted in Von Luxburg (2007), and we concatenate these matrices as presented in Tang et al. (2009). This enables us to effectively capture the distinct ties represented in many types of social media, as well as node level metrics of each graph and user account features.

Users can often use topical markers like hash tags in Twitter, and these can be used to cluster users with similar topical interests. This results in bipartite graphs, $G_{V_t \times V_n, E_m}$, where users and topical markers represent differing node sets, however we with to use these links to find similarities with respect to topical interests among users. To do so we implement bispectral clustering as introduced by Dhillon (2001a) as a document clustering method. In our case, instead of co-clustering documents based on word frequency, we co-cluster users based on hashtag frequency within their tweets. To do so we develop $W_{V_t \times V_n}$, where $w_{i,j} \in W_{V_t \times V_n}$ represents the number of time vertex $v_{n,j}$ appears in the twitter stream of $v_{t,i}$. To co-cluster $v_{t,1}...vt, n$ we follow the biparitioning algorithm provided in Dhillon (2001a) , which results in eigenvector features similar to those we defined in the previous paragraph.

The combination of user account attributes, node level metrics from the larger network G, and spectral features explained above provide a rich feature space. Paired with a reasonably sized set of labeled vertices, we can detect an extremist community embedded in social media with supervised classification. If labeling vertices is impractical and node attributes appear informative, vertex clustering methods can be used as in Wang et al. (2010). Although we implement two different binary classifiers in Section 3.3, specific algorithms selected for either phase of this methodology are the decision of the researcher. The end result of IVCC, an accurate extraction of vertices A_t, facilitates a social network analysis of the OEC of interest.

3.3 Case Study: The ISIS OEC on Twitter

To illustrate the utility of our methodology we offer a case study of the ISIS OEC on Twitter. This case study aims to validate our proposed methodology, present its limitations in terms of ethical use, and provide illustrative examples of intelligence that can be mined from OECs. Although the results of our case study provide strong results in terms of accuracy, and we have provided both traditional and sampling based methods for performance evaluation, we stress that we see these methods primarily as a means to understand the interests and behaviors of this OEC. As with any classification technique, false identification of ISIS OEC members must be considered by the practitioner, and using IVCC to support any type of intervention should be used within the context of multiple sources of intelligence. We discuss intended use and the societal implications of similar methodologies in detail in Section 6.5.

3.3.1 ISIS Data

In this section we describe both our collection methods and dataset, but before doing so we would like to clearly state that we have complied with all of Twitter's terms of service and privacy policies Twitter (2016). To develop our dataset, we instantiate our sampling strategy with five known, influential ISIS propagandists highlighted in Carter et al. (2014). In November, 2014 we conducted a two step *snowball sample* Goodman (1961) of these users' following ties. Snowball sampling is a non-random sampling technique where a set of individuals is chosen as "seed agents." The k most frequent accounts followed by each seed agent are taken as members of the sample. This technique can be iterated in steps, as we have done in our search. Although this technique is not random and prone to bias, it is often used when trying to sample hidden populations Berger and Morgan (2015b).

Step one of our search collected user account data for our 5 seed agents' 1345 unique following ties. Step 2 resulted in account information for all users followed by the 1345 accounts captured in step 1. Our search resulted in 119,156 user account profiles and roughly 862 million tweets. This network is multimodal, meaning that it has two types of vertices, and multiplex, because it has multiple edge types. We represent this set of networks, as a heterogeneous social network with annotated nodes Steinfield et al. (2008), G with two node classes: users and hashtags, and four types of links: following relationships, mention relationships, and user-hashtag links. Summary statistics of each network are provided in Table 3.2.

The snowball method of sampling presents unique and important challenges within social media. Users' social ties often represent their membership in many communities simultaneously Papadopoulos et al. (2012) . At each step of our sample, this results in a large number of accounts that have little or no affiliation with ISIS. The core problem of the present work is to identify the set of users within the 119,156 accounts collected that support ISIS in varying degrees. In order to do so, we required a rigid definition of what it means to support ISIS. We define the following three user types of interest:

- *ISIS OEC member:* Similar to Berger and Morgan (2015b), we code users who unambiguously support ISIS as OEC members. For example, if the user positively affirmed ISIS leadership or ideology, glorified its fighters as martyrs, affirmed ISIS' call to Jihad as a duty for all Muslims, or used pro-ISIS images in their profile (i.e. the ISIS flag or images of key figures like Abu Musab Al-Zarqawi or Abu Bakr al Baghdadi), we coded them as OEC members. Furthermore, in light of the growing emphasis placed on "passive observers"Veilleux-Lepage (2015), we infer retweets as endorsement. Therefore, a member's *support* is relative and in many cases not in violation of local law or Twitter's terms of use. However, including this broad continuum of support facilitates the study of populations that could be more susceptible to radicalization.

- *non-member:* A user whose tweets were either clearly against ISIS or showed no Jihadist content.
- *official account:* We label vertices as *official accounts* if they meet any of the following criteria: the user's account identifies itself as a news correspondent for a validated news source; the account is attributed to a politician, government, or medium sized company or larger; or, following Berger and Morgan (2015b), if the account has more than 50,000 followers. This third categorization was deemed necessary as in the process of our case study, we identified dense following and mention ties between ISIS OEC members and news media, politicians, celebrities, and other official accounts. Such accounts are interesting in that there higher follower counts and mention rates tend to make them appear highly central even though they do not exhibit any ISIS supporting behaviors. *Official Accounts* must be identified and removed for accurate classification of ISIS-supporting, thus illustrating the utility of an iterative methodology.

3.3.2 IVCC Implementation

By sampling user accounts from G it is clear that the preponderance of accounts collected have no visible affiliation with ISIS, but we, like Berger and Morgan (2015b), expect an ISIS supporting community to be captured by our sampling strategy. However, community optimization results of the mention, M, and following, F, networks highlight an interesting phenomenon. We used the Louvain Grouping method presented in Blondel et al. Blondel et al. (2008) to cluster M and F. In each case we found that our 5 seed agents were assigned to one of two clusters. For example, clusters 4 and 6 of the mention network contained all 5 of our seed agents. During the time period between our data collection and analysis, November of 2014 to March of 2015, Twitter has initiated an aggressive campaign to suspend ISIS supporting users Gladstone (2015c), and we found the clusters containing our seed agents to have excessively high suspension rates. For example clusters 4 and 6 of the M network had suspension rates of 41% and 21% respectively as shown in Figure 2. No other cluster had suspension rates above 5%. Figure 2 depicts the size, suspension/deletion rates, and number of users classified as ISIS OEC members within the 10 largest Louvain groups Blondel et al. (2008) in our weighted, directed network M. We determined excessively high suspension rates within clusters 4 and 6 to be consistent with ISIS support. Although these clusters contained ISIS OEC members, modularity based clustering algorithms like Blondel et al.Blondel et al. (2008), did not provide enough information to distinguish between ISIS OEC members and other user types. There were still many official and non-ISIS supporting accounts in each of the clusters with elevated suspension/deletion levels, and manual sampling indicated that ISIS OEC members existed in clusters without high suspension

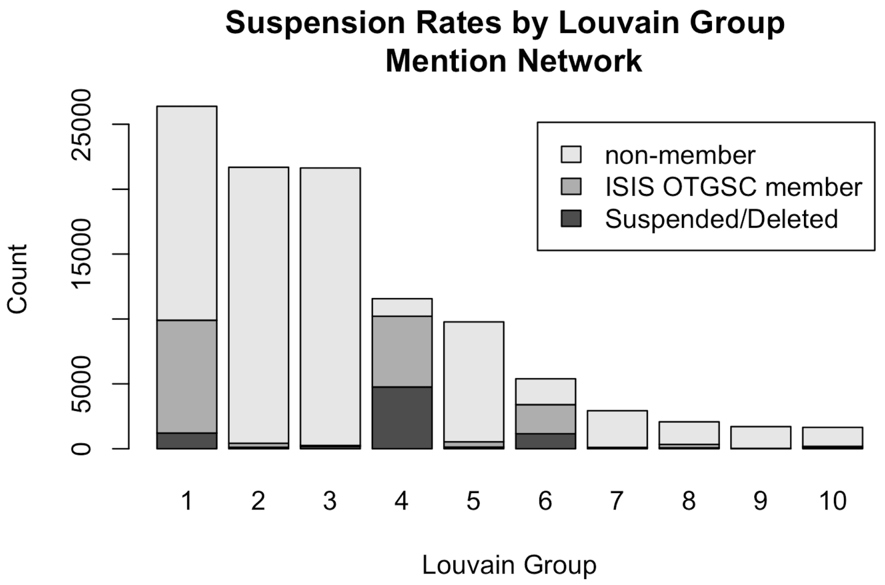


Figure 3.2: depicts the size, suspension/deletion rates, and number of users classified as ISIS OEC members within the 10 largest Louvain groups Blondel et al. (2008) in our weighted, directed network M where edges are defined as the number of times user a mentions user b in his/her Twitter timeline. Our 5 seed agents were assigned to clusters 4 and 6 which had Twitter suspension rates of 41% and 21% respectively. No other cluster had a suspension rate above 5%. Accounts were either deleted by users or suspended by Twitter between the dates of 24 November, 2014 and 12 April, 2015, which coincided with Twitter's aggressive ISIS related account suspension campaign ongoing in the same time period Gladstone (2015c). We used this combination of factors to select suspended/deleted accounts in groups four and six as training examples of *ISIS OEC members* for classification. It is worth noting that our classifier did not simply find accounts contained in clusters 4 and 6 as is highlighted by the figure as well.
.

rates as well. However, community optimization provided enough context for us to reasonable use the union of suspended/deleted users in Louvain clusters 4 and 6 in M, as labelled *ISIS OEC member* cases for vertex classification. Community optimization also helped us identify the need to systematically remove *official accounts*.

We constructed a feature set using spectral representations of the F_{rec}, M_{rec}, and $H_{user \times user;sharedHashTag}$ networks as described in Section 3.2. A full list and description of our feature set is included in Table 3.3. As will be highlighted in Section 6.5, the ISIS OEC is highly interested in the ongoing operations in Northern Iraq and Syria. As such, they discuss political figures and news sources extensively. Initial attempts to detect the ISIS OEC contained many *official accounts* as previously defined. Therefore, in our first iteration of multiplex vertex classification (MVC) the task was to remove all official, celebrity, and news media accounts. To do so, we conduct an iteration of IVCC by developing a training set of positive and negative examples of *official accounts* to apply to the rest of our dataset. Our positive case labels for official accounts consisted of 2,144 known celebrities, politicians, and journalists as well as an additional 873 accounts with more than 150,000 followers. We labelled the 8,356 suspended/deleted accounts in our dataset as non-official accounts, and trained a *Random Forest* classifier Liaw and Wiener (2002) The Random Forest classifier is an ensemble method that constructs a multitude of decision trees and uses the mode of these classes to correct for the problem of overfitting associated with many tree based classifiers. We found its performance to be significantly better than SVM with respect to accuracy when identifying *official* accounts to remove from our dataset. The classifier's superior performance was likely due to the various types of *official* accounts creating contingencies better captured by a tree based classifier. It is worth mentioning that we are not interested in using this classifier on accounts not contained in G; so we conduct use a train/test split, but also use random sampling to assess accuracy.

The resultant classifier yielded accuracy of 91.3 % and an F1 score of 75.8% on these heuristically labeled examples. Our post prediction sampling yielded no significant difference with blind classification of 50 randomly selected accounts. The classifier identified an additional 7,140 news/celebrity/official accounts which we removed from G to form $G^{(-)}$.

Once we were confident that a high percentage of *official accounts* were removed, we conduct an iteration of MVC to identify ISIS OEC members. For this task we train a Support Vector Machine classifier similar to those presented in Tang and Liu (2011). Again, we labeled the 5,126 accounts marked as suspended/deleted and grouped in Louvain clusters 4 and 6 of the M network *ISIS OEC members*. We then randomly sampled 10,000 active accounts in Louvain groups 3,4, and 7 in the F network and labelled them as *non-ISIS supporting*. The resultant classifier identified 18,335 *ISIS OEC members*. We then combine the classified 18,335 vertices with our 5,126 labelled vertices and construct A_t. With our network of suspected ISIS OEC members,A_t, we conduct community optimization and network analysis in Section 6.5. Summary statistics of A_t are provided in Table 3.4. We acknowledge that our positive case training instances contain uncertainty, as Twitter suspends accounts for a variety of reasons. We will address this issue and discuss our efforts to validate model output in detail if the following section.

3.4 Performance and Validation

In this section we will present our results, first for the model's performance on our training data set and then we will discuss additional manual validation efforts using our predictions.

Multiplex vertex classification (MVC) extends current methods by applying a combination of the findings developed in Tang et al. (2009) and Tang and Liu (2011). Given a large multiplex network with annotated vertices, we are able to accurately identify our targeted community, ISIS OEC members. We compare MVC to Tang et al. (2009) and Tang and Liu (2011) by constructing three feature sets:

- θ_{MNVC}: represents the present work and consists of user account features and spectral and node metric representations of the following, mention, and user by user (shared hashtag) networks.
- $\theta_{SocioDim}$: represents Tang and Liu (2011) and consists of user account features and a spectral representation of the mention network.

- θ_{PMM}: represents Principal Modularity Maximization (PMM) as presented in Tang et al. (2009). PMM utilizes eigenspace representations of the following, mention, and user by user (shared hashtag) networks. For this feature set we used the largest two eigenvectors of each of the respective networks and subsequently performed canonical correlations to maximize the correlations between each network's respective eigenspaces.

A detailed description of each feature set is provided in Table 3.3.

Table 3.1 illustrates *MNVC's* superior performance across all performance metrics. Accuracy is simply defined as the proportion of correctly classified cases in our test set. Precision is the percentage of positively classified cases that were actually positive. Recall measures the percentage of positive cases that were classified positive. Finally, the F1 Score Powers (2011) estimates accuracy by adjusting for bias associated with skewed class distribution. It is important for us to reiterate that our measures of performance in this section quantify how well our classifier was able to differentiate classes in our training data. We acknowledge that we have made assumptions to develop our positive case training instances that could reduce precision when applied to unlabeled data. Therefore, an F1 score of 96% does not necessarily imply that approximately 96% of the users we predict are "true" ISIS-supporting OEC members. However, we have taken measures to validate model output manually as will be explained at the end of this section.

We se that *MNVC* outperforms both *SocioDim* and *PMM* with respect to each metric. Although our classifier's performance is relatively high, with approximately 22,000 accounts classified as ISIS OEC members we would expect more than 900 accounts to be falsely labeled as ISIS OEC members. We will discuss the application of these methods in detail in Section 3.6. However, a 4% false positive rate and the varying degrees of "support" observed among passive sympathizers again imply these methods would best serve as a means to study online populations that appear vulnerable to online extremism.

With respect to our *official account* classifier, *MNVC* and *SocioDim* performed almost identically. We hypothesize that this is likely due to the heterogeneous nature of *official* accounts. We used this classifier to remove accounts belonging to celebrities, news media, corporations, NGOs, and governmental organizations. Thus, the positive class likely had many contingencies associated with it and would be more well suited to a tree based classifier like the Random Forest algorithm explained in Section 3.2.

Our use of Twitter suspension rates within specific user groups as positive case labels introduces uncertainty as there are many reasons for Twitter to suspend accounts. To address these limitations, we took several steps to assess the accuracy of our heuristics. This included discussions with native language speakers and blind sampling of accounts predicted as ISIS OEC members. Further, our analysis indicated the ISIS classifier generalizes to unlabeled data in ways that would not suggest biases from our network-based and suspension/deletion-based heuristics. Many of the accounts labeled by our classifier post content that is barbaric and in clear violation of The Twitter Rules precluding the use of the service to promote violence twi. There are other predicted ISIS OEC members whose content does not clearly violate Twitter's policies and would generally be considered free speech. However, these users' content is still consistent with the description of "passive supporters" presented in Berger, JM; Veilleux-Lepage (2014, 2015). Finally, in light of Twitter's continued aggressive program to remove extremist content from its site Balakrishnan, we performed an additional check of suspension rates in January, 2017. We found suspension rates of 39%, 7%, and .4% for our predicted classes of ISIS-supporting, non-ISIS-supporting, and official accounts respectively. Although these suspension rates do not conclusively prove any account predicted as an OEC member to be an ISIS-supporter, they do imply that our methodology identifies communities containing sizable pockets of extremism.

We have, in this section, therefore performed a variety of checks to ensure that our classifier is able to identify members of the ISIS OEC in ways that outperform other relevant approaches. As we have noted, there is no way good to assess "ground truth" with pure certainty in our setting, thus leading to some uncertainty in our validation efforts. However, this uncertainty should be considered in the context of many other related tasks in social media mining and natural language processing where the quality of annotation has recently been questioned Blodgett et al. (2016); Joseph and Carley (2016), even on tasks as seemingly straightforward as dependency parsing Berzak et al. (2016). While analyses of performance are imperfect here, we have tried in various ways to address them (e.g. through analyzing suspension rates and qualitative analysis of results), making our efforts as stringent if not more so than much related work. Future efforts are needed across the field as a whole in order to better understand how to address these outstanding issues.

3.5 Case Study: The ISIS-Supporting OEC

The challenge of drawing useful intelligence analyses from social media remains an open research problem, but OEC detection offers new opportunities for intelligence and strategic communications experts to gain needed understanding into large populations susceptible to extremism. The following subsection is intended to provide illustrative intelligence analyses offered by OEC detection.

The left panel in Figure 3 depicts the ISIS supporting reciprocal mention network, $A_{M,rec}$, where color indicates Louvain Grouping. Language drives the most clear division among internetwork communities and is highlighted in the middle panel. We used *LangID* as introduced in **?** to identify language at the user level. Blue vertices indicate users whose tweet streams identified as Arabic with probability in excess of 90%, while green vertices depict users whose tweet streams identified as English with probability in excess of 90%. Yellow vertices indicate users whose tweets contain a mixture of English and Arabic. A small portion of those users contained mixed language patterns to include Turkish and Russian. For the most part however, these users form a bridge between the Arabic speaking and non-Arabic speaking communities in the ISIS supporting network.

Interesting structure also exists within the Arabic speaking portion of the community. The relatively small cluster to the far right of the Arabic speaking portion of the community, represented by yellow vertices in the left panel, consisted of accounts sharing lectures and videos on Muslim theology. While the majority of these accounts did not overtly promote jihad or support ISIS, it is interesting to highlight that their follower counts often contained hundreds or thousands of ISIS OEC members. An example of one such account belongs to Dr. Hani al-Sibai who has been cited by Ansar al-Sharia as one of five influential thinkers from whom the terrorists in Tunisia obtain their encouragement Tunisias and Game (2013). At this time we are unable to determine to what degree these accounts provide active support, or if their followers simply

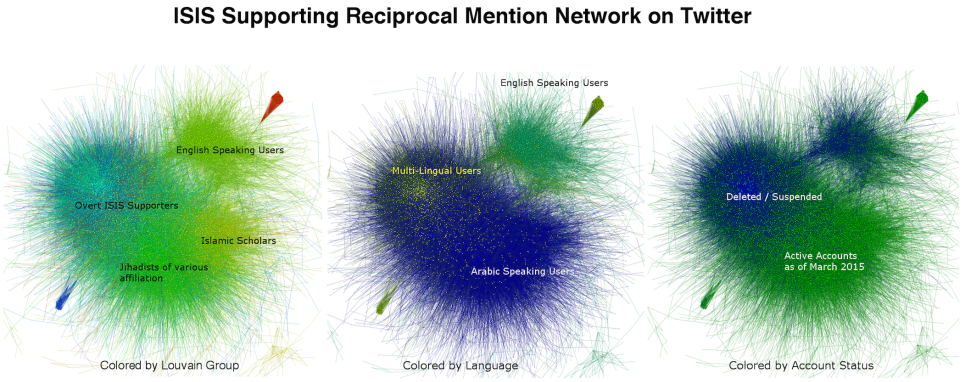


Figure 3.3: each panel depicts the ISIS supporting reciprocal mention network, $A_{M,rec}$. The left panel is colored by Louvain Group, the center panel by user language patterns as detected by his or her tweets, and the right panel depicts each user accounts status as of March 22, 2015.

present a fertile recruiting landscape for ISIS propagandists.

It also appears that some propagandist accounts use bots to gain stronger influence. The red and blue groups depicted in the left panel of Figure 4 are visible examples of what we believe to be bots in our dataset. We believe these to be bots because in each case the groups represent a fully connected sub-group where each account repeatedly mentions all other members of the group, as well as a 'parent account' or accounts. Although relatively few accounts exhibit this group structure, we hypothesize they are used to elevate the relative popularity of the associated "parent accounts" and remove them for subsequent analysis.

Figure 4 highlights changes in user activity with respect to time. The left panel depicts ISIS supporting users where the *x-axis* details the account creation date and the *y-axis* gives the average number of tweets per day for the life of the account. Color indicates the suspension status of the account, where a black circle indicates the account remains active, while red indicates the account has been deleted or suspended. The right panel depicts a time series of the tweet stream of 10,000 randomly sampled ISIS supporting users (black lines). Each time series has a high level of transparency to illustrate the distribution of daily user activity over time. The red line depicts the cumulative distribution function of account creation dates within the ISIS supporting network. The plot highlights the creation of many ISIS supporting accounts providing a high volume of tweets in the fall of 2014. In particular, the large number of high tweet volume accounts introduced in early October 2014 were likely bots. Though the left panel clearly highlights Twitter's ability to identify and suspend these accounts, their effect is clearly seen in the right panel, and this highlights the group's use of bots to possibly generate recruits and/or inflate the perception of their appeal.

Beyond understandings of the group structure and tweet time series, the role and relative importance of users within the observed social network network are of interest. To gain insight into this, we rely on two link types within our dataset: mention and following ties. Reciprocity has been shown to be a strong indicator of trust within online social networks Chiu et al. (2006); Gilbert and Karahalios (2009); Mislove et al. (2007), and reciprocal mention ties provided the most information gain with respect to our ISIS supporting classifier. Co-mention ties also provide strong indicators of core membership within our ISIS supporting network. Both betweenness and

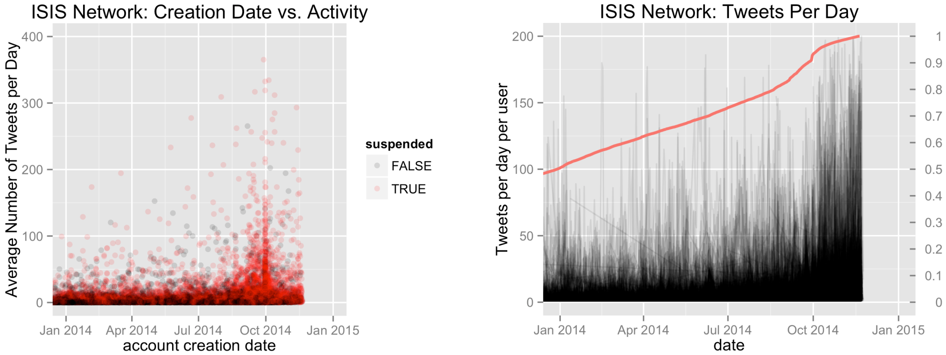


Figure 3.4: highlights changes in user activity with respect to time. The left panel depicts ISIS supporting users where the *x-axis* depicts account creation date, and the *y-axis* depicts the average number of tweets per day for the life of the account. Color indicates the suspension status of the account where a black circle indicates the account remains active, while red indicates the account has been deleted or suspended. The right panel depicts 10,000 randomly sampled, ISIS supporting tweet streams in black. Each time series has a high level of transparency to illustrate the distribution of daily user activity over time. The red line depicts the cumulative distribution function of account creation dates within the ISIS supporting network.

degree centrality quantify how "trusted" a user is among other members of the network, but trust alone does not identify core members or help distinguish roles. To account for this we construct the following metric, which quantifies the proportion of a user's following ties that are members of our ISIS supporting network, A. We refer to this metric as *ISIS Focus*, and use it as a proxy for the user's ideological affiliation with ISIS.

$$\text{ISIS Focus} = \frac{f_{\text{ISIS Supporting}}}{f_{total}} \tag{3.2}$$

Figure 5 depicts the bivariate distribution of users classified as *ISIS supporting* with respect to degree centrality within the reciprocal mention network (*x-axis*) and ISIS focus (*y-axis*). The dashed black lines depict the median values of the two respective metrics, dividing the plot into four quadrants. Though the quadrants depicted in Figure 5 do not represent finite delineations with respect to user role type, we find that both metrics provide useful information when identifying core members

Users with high degree centrality and high ISIS focus (quadrant I in Figure 5) are powerful disseminators of ISIS' message. These are often accounts of popular fighters, accounts designed to look like legitimate news media, or simply popular ISIS propagandists. Those with high ISIS focus and low degree centrality (quadrant II) represent similar accounts, but with less popularity. They appear to have ideals almost identical to those in quadrant I, but are either less skilled at generating a following or relatively new to the network. We also expect recruits to be more likely identified in quadrant II. Accounts with high degree centrality and low ISIS focus are highly trusted but not as highly affiliated with ISIS. This quadrant contained accounts that did not overtly support ISIS but provide information highly relevant to core members like regional news media and Islamic sermons and educational material. Additionally, there were users who

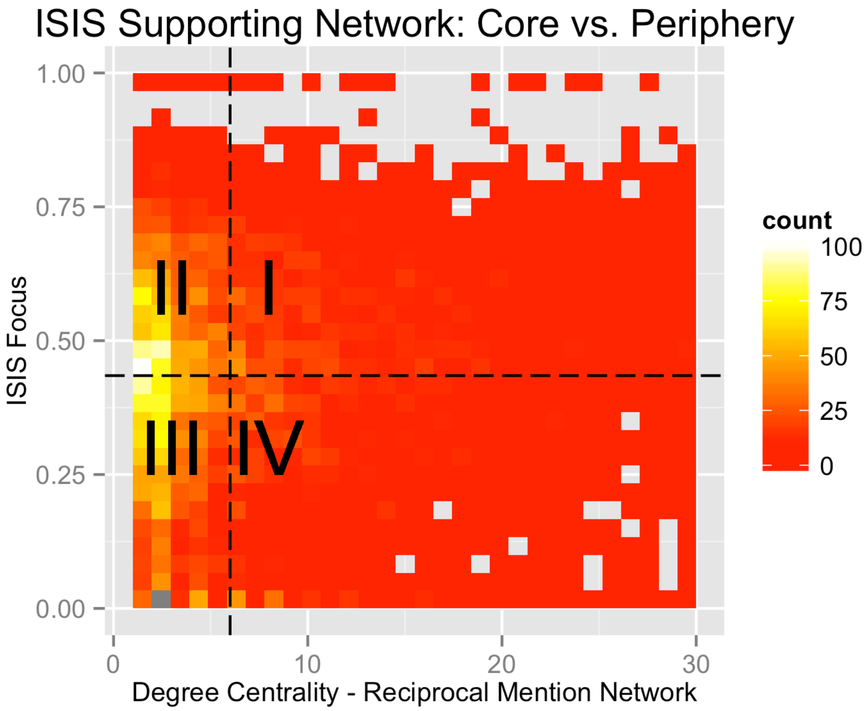


Figure 3.5: depicts the distribution of ISIS supporting accounts with respect to degree centrality in the reciprocal mention network (*x-axis*) and ISIS focus (*y-axis*). ISIS focus refers to the proportion of an individual user's following ties that are classified as ISIS supporting. The dashed white lines depict the median values their respective metrics.

appeared loyal to other jihadist groups such as Jabhat al Nusra or Ahrar al Sham or other popular causes in the region such as charities associated with Gaza. Finally, users with relatively low scores in both metrics (quadrant III) represent passive observers.

These measures are important in that we can use these measures to prioritize additional searches of the Twitter API. That is, for those users we identified in Step 2 of our sample, we have not collected accounts from all of their following ties, and can now use a combination of these metrics to prioritize which accounts to download.

Removing non-ISIS supporting accounts also enables us to understand the topical interests of ISIS OEC members and how they evolve over time. Such analysis is critical to gain understanding and counter ISIS' narrative and its ability to generates resources. We quantify both the frequency of various hashtags, as well as the number of distinct actors using them. This enables us to identify what topics have the broadest appeal, as well as topics that might be the result of a small set of highly active users. Figure 6 depicts the 369,603 unique hashtags used by ISIS OEC members in our dataset. Blue points depict Arabic hash tags, and red points depict hashtags in other languages. Generally, a hashtags frequency and the number of unique users tagging with it are proportional; however some hashtags, like the three labelled in the figure, seem to have frequencies inflated by a relatively small, highly active group of users. A closer look at hashtag 1 is translated "Tweet mentions of Allah" and is associated with a Twitter application that offers to mention God every hour on a user's timeline. The hashtag is used over 100,000 times but by a relatively small set of 1648 users. Of these tweets, 75,382 are posted by only 100 users who all seem to retweet one another's verses from the Quran and Hadith as well as unique ISIS related content from the battlefield. In other words these hashtags are used by high volume tweeting users to systematically link the groups theology with battlefield exploits. We postulate that this type of analysis could also identify key mouthpieces or propagandists in the network.

More broadly, we can identify the most unifying and energizing topics of the network by looking at how the most broadly used hashtags change over time. Figure 7 depicts the top 100 non-Arabic hashtags in terms of number of unique ISIS supporting users. The y-axis depicts the seven day moving average of the respective hashtags frequency over time. Non-Arabic hashtags with a moving average that reach above 500 tweets per day at any given time period are labelled.

Many of the popular hashtags confirm things we already know about the ISIS supporting movement. ISIS OEC members focus on events relating to Sunni conflict in the greater MENA region, and the temporal peaks in Figure 7 reflect those interests. However, some of these hashtags offer novel insight. For example, the popularity of *#helparakan*, referring to a state in Burma, is consistent with the ISIS Study Group's assertion that expansion into South Eastern Asia is one of ISIS' strategic objectives Strategy and Group. The trending hashtag *#EI* refers to 'l'etats Islamic' and highlights the networks interest in Mehdi Nenmouche, a French jihadist's arrest and pending extradition to Belgium in June of 2014 Dickey (2014). Identifying these topics of interest and the influential users tweeting about them could provide useful understanding of the group's 'marketing' objectives and help drive intervention strategies.

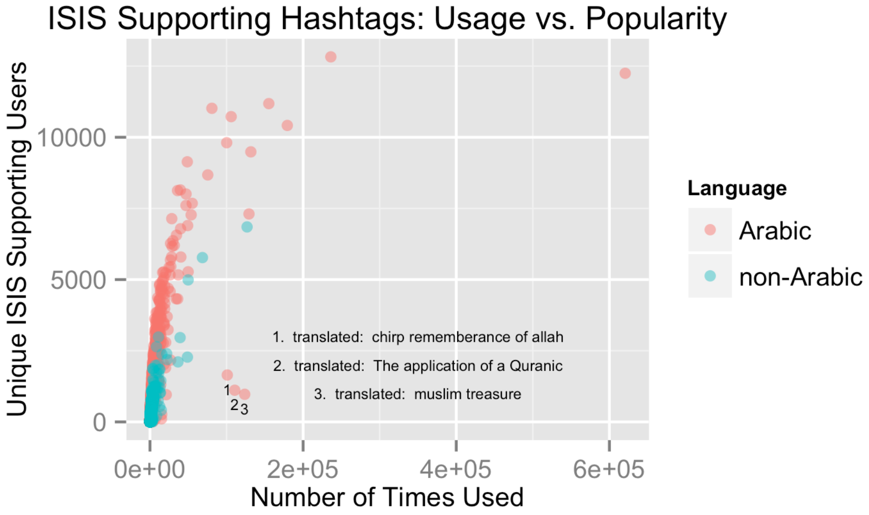


Figure 3.6: depicts the 369,603 unique hashtags used by ISIS OEC members in our dataset. Black points depict Arabic hash tags, and red points depict hashtags in other languages. Generally, a hashtags frequency and the number of unique users tagging with it are proportional; however some hashtags, like the three labelled in the figure, seem to have frequencies inflated by a relatively small, highly active group of users.

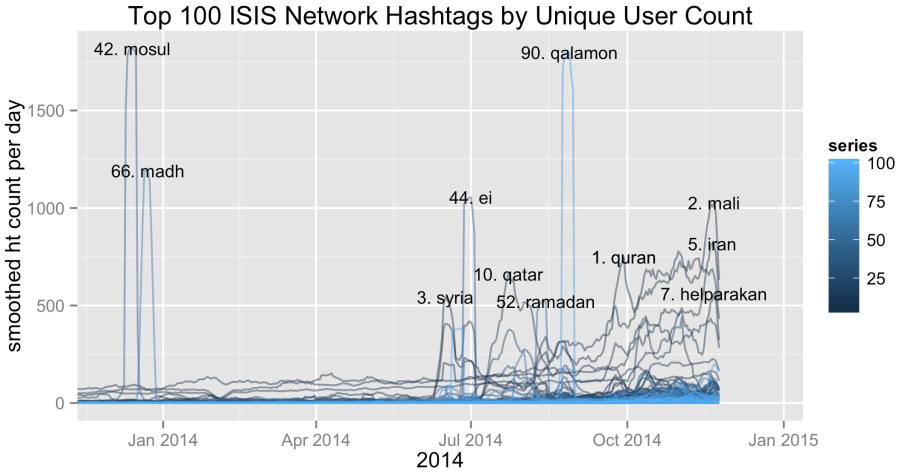


Figure 3.7: depicts the smoothed time series of the top 100 ASCII character hashtags in terms of number of unique users. The series are calculated using a 7 day moving average of each respective hashtags frequency in our ISIS supporting network. All hashtags whose average is greater than 500 at any given time are labelled.

3.6 Societal Implications and Methodological Limitations

The responsible use of social media intelligence and its relationship to individual privacy in democratic states is an important, open question for policy makers Bartlett (2016); Miller et al. (2011b); Walsh and Miller (2016). To this end, we acknowledge that our methods could be unethically employed to identify political opposition or dissidents. Indeed, our classifiers that did not incorporate analysis of hashtags routinely identified online activism related to a variety of causes.

Consequently, we join Walsh et al. in their advocation of patient, nuanced political dialog with respect to developing open source intelligence policy in Western democracies Walsh and Miller (2016). This policy debate centers around both social media users' reasonable expectation of privacy and the ethical implications of mining their online content.

With respect to the latter, the ethical implications of mining online content using our method vary based on the intended use of the method. We have envisioned here two use cases for IVCC. First, and most importantly, as Western governments have started to search for diplomatic means to counter extremist propaganda, IVCC can be as a means to gain understanding of online populations vulnerable to extremism. We believe this to be an ethical use of the method, as the primary intention is to reduce the likelihood of an individual being deceptively coerced into an extreme ideology. A second use case of IVCC would be for intelligence collection. This use case certainly could require more restrictive policy depending to intelligence category.

With respect to the former element of policy debate, it is without question that users' reasonable expectation of privacy must be kept in mind at all times. A common argument against doing so is that social media users have the ability to privatize their accounts, or to not use the media at all. However, these options are often not tenable. Further, although many users understand their online behavior is used for marketing purposes, they may not be comfortable with their behavior being used to inform diplomacy or military operations. Indeed, one could assume users would not consent to the use of their information for intelligence collection.

This distinction between marketing versus intelligence objectives in an important one, particularly in light of the mission statement for the newly formed United States Department of State's Center for Global Engagement:

> *The State Department is revamping its counter-violent-extremist communications efforts through a new Global Engagement Center. This center will more effectively coordinate, integrate and synchronize messaging to foreign audiences that undermines the disinformation espoused by violent extremist groups, including ISIL and al-Qaeda, and that offers positive alternatives. The center will focus more on empowering and enabling partners, governmental and non-governmental, who are able to speak out against these groups and provide an alternative to ISILs nihilistic vision. To that end, the center will offer services ranging from planning thematic social media campaigns to providing factual information that counters-disinformation to building capacity for third parties to effectively utilize social media to research and evaluation.*of State (2016)

For objectives similar to those listed above, the use of IVCC by government agencies would therefore be subject to similar protocols to those used for behavioral research by institutional review boards (IRBs). These protocols include a strong push for de-identification - our methods make no attempt to bind online and offline identities, and agencies using these methods to inform messaging efforts could do so with de-identified data. While we acknowledge that the use of bulk de-identified meta-data has been the subject of concern Walsh and Miller (2016), this issue is routinely encountered by IRBs in academia as well.

Further, within a context of informed diplomatic messaging, the use of IVCC is thus proximal to academic research and further, analogous to individually tailored online marketing. Ethical employment of our methods could be carried out to understand vulnerable online populations and ultimately preserve civil liberties. Peacetime military information operations aimed at messaging to specific populations could be viewed similarly of Staff (2014), and implemented with de-identified data.

The complexity of these issues requires a substantive theoretical framework under which to characterize these various ethnical concerns. Walsh et al. (2016), who provide a framework with which to balance the importance of civil liberties with national security in an intelligence context Walsh and Miller (2016). Their framework is based on the collection method, context, and target. In our case, social media mining would be our method; however the increasingly complex combinations of context and target imply the need for nuanced policy. Currently, policy has started to address the varying expectations of privacy in wartime, peacetime, and counter-terrorism contexts. However, the onset hybrid warfare that is conducted by state and non-state actors purposely beneath the threshold of Western military intervention Hoffman (2009) further complicates policy development.

The intelligence target also has policy implications. Specifically, the purpose and category of the desired intelligence product needs to be considered. For the purpose of describing a commander's operating environment or assessing ongoing operations, authorities could be quite liberal. Intelligence used to develop military targets or bind online and offline identity would imply more restrictive policy. As stated by Walsh et al, the increasingly complex nature of conflict call for patient political dialogue, and policy makers need to 'take their citizens with them' when making arguments for new policy and authorities Walsh and Miller (2016).

In sum, implementation of IVCC for social media intelligence does, on the one hand, require a more formal framework and more nuanced discussion. On the other hand, however, it is clear that the method can also be used in many ethical fashions and to improve efforts of national security.

3.7 Conclusion

The present work makes two major contributions to the literature. First, we develop iterative vertex clustering and classification (IVCC), a scalable, annotated network analytic approach for extremist community detection in social media. Our approach outperforms two existing approaches on a classification task of identifying ISIS supporting users by a significant margin. Second, we provided an illustrative case study of the ISIS supporting network on Twitter. To the best of our knowledge, it is the most comprehensive study of this network, and it provides a variety of important insights that may prove important in better understanding the incredible proliferation of ISIS propaganda on Twitter. Most notably, we find that:

- Leveraging the multiplex and multinode structures available in Twitter data significantly improved our algorithm's ability to accurately identify ISIS OEC members on Twitter.
- Identifying and isolating large portions of an online extremist community offers unique

insights into the group's ideology and influence, and helps identify key users and roles.

- IVCC offers promise for making online extremist community detection in social media a practical reality to inform both diplomacy and defense initiatives.

This case study offers a unique opportunity to infer positively labelled cases based on Twitter suspensions and clustering techniques. However, it is unlikely that such a large number of labeled cases would always be available. Thus, implementations using semi-supervised algorithms or active learning Settles (2010a) would make IVCC more generalizable, and should be a topic for future research. IVCC is also limited in that it does not account for simultaneous group membership of users. It is likely that there are jihadists that support various terrorist groups and allegiances can be dynamic. IVCC does not provide probabilistic clustering or account for changes in group dynamics over time. Similar to Yang et al. (2013), we would like to extend this methodology to an overlapping group framework to account for these types of users and also explore methods to identify temporal change points. Finally, though preliminary results for IVCC as a methodology are encouraging, they are limited in that we do not provide an empirical analysis of IVCC with respect to benchmark. We will leave this analysis to future work, due to the emphasis of this paper being the ISIS case study.

Extremist community detection is an important need in processing social media, and with such approaches like IVCC, we hope that the influence of groups like ISIS can be counteracted in the near future.

Tables

Model	Performance Metric	
	Accuracy, 95% CI: Accuracy	F1
θ_{MNVC}	0.96, (0.95, 0.96)	0.93
$\theta_{SocioDim}$	0.87, (0.86, 0.88)	0.80
θ_{PMM}	0.84, (0.83, 0.84)	0.74

Table 3.1: Performance estimates for the ISIS classifier for feature sets: θ_{MNVC}, $\theta_{SocioDim}$, and θ_{PMM}. The left column depicts both the point estimates and 95% confidence intervals for accuracy. The right column depicts the F1 score Powers (2011) associated with each feature set.

Metric	**Network**				
	F Following	F_{rec} Reciprocal Following	M Mention	M_{rec} Reciprocal Mention	$H_{userxuser}$ User by Hashtag
From Node To Node	User User	User User	User User	User User	User Hash Tag
Link Type	directed, binary	undirected, binary	directed, weighted	undirected, weighted	undirected, weighted
Nodes	119 k	119 k	109 k	109 k	106 k x 4 M
Links	23.1M	3 M	14.6 M	1.1 M	27.4 M
Density	0.00163	0.000425	0.00123	0.00018	0.000065
Isolates	0	10888	291	30,047	0
Dyads	0	104	6	425	188
Triads	0	19	0	50	33
Larger	1	8	2	7	6

Table 3.2: Depicts G_{full}, the resultant heterogeneous network from our 2-step snowball search of known ISIS OEC members. The search yielded 400 G of data containing 119,156 Twitter user accounts' following ties, account profiles, and tweets.

Feature:	**Source, Description**	θ_{MNVC}	$\theta_{SocioDim}$	θ_{PMM}
Creation Date	Twitter User Profile	✓	✓	
Tweet Count	Twitter User Profile	✓	✓	
Follower Count	Twitter User Profile	✓	✓	
Following Count	Twitter User Profile	✓	✓	
Unique Hashtags	Twitter User Profile	✓		
In-Degree Centrality	Follower x Follower Network	✓		
Out-Degree Centrality	Follower x Follower Network	✓		
In-Degree Centrality	Mention x Mention Network	✓		
Out-Degree Centrality	Mention x Mention Network	✓		
Total-Degree Centrality	Follower x Follower Network, Reciprocal Ties	✓		
Total-Degree Centrality	Mention x Mention Network, Reciprocal Ties	✓		
Search Step	Twitter API Script	✓		
U_{RF}	a user x 2 matrix with columns consisting of the eigenvectors associated with the 2 largest eigen values extracted from the graph Laplacian of our Following x Following Network with Reciprocal Ties.	✓		✓
U_{RM}	a user x 2 matrix with columns consisting of the eigen vectors associated with the 2 largest eigen values extracted from the graph Laplacian of our Mention x Mention Network with Reciprocal Ties.	✓	✓	✓
U_{UxHT}	a user x 2 matrix with columns consisting of the eigen vectors associated with the 2 largest eigen values extracted from the graph Laplacian of our User x User (Shared Hash Tag) Network.	✓		✓

Table 3.3: lists and describes features used in each classifier

Metric	**Network**			
	A_F	$A_{F,rec}$	A_M	$A_{M,rec}$
Description	Following	Reciprocal Following	Mention	Reciprocal Mention
From Node	User	User	User	User
To Node	User	User	User	User
Link Type	directed,	undirected,	directed,	undirected,
	binary	binary	weighted	weighted
Nodes	21,343	21,343	23,031	22,456
Links	1,254,529	94,583	1.6M	220,597
Density	.0052	.0008	.003	.0004
Isolates	15	1687	269	0
Dyads	2	58	26	0
Triads	0	10	1	0
Larger	1	6	1	0
Reciprocity	.082	1	.016	1
Char. Path Length	3.432	4.44	4.723	15.76
Clustering Coeff.	.129	.154	.111	.065
Network Diameter	11	13	1521	2213

Table 3.4: Depicts A_t, the suspected ISIS OEC member network identified in Section 6.5. Each network is more dense than its parent network in G.

Chapter 4: Unsupervised Detection of Online Activism and Extremism

4.1 Introduction

Online social networs (OSNs) are increasingly viewed as publication platforms and have become one one of the top delivery platforms of news in the world Kwak et al. (2010). However, the proliferation of inflammatory misinformation has forced companies like Twitter, Facebook, and Google to take measures to counter this type of content (Wakabayashi and Isaac, 2017). This union of inflammatory content and the dense social structures provided by OSNs provide an unprecedented environment for propaganda and appears to be contributing to large geopolitical movements like populism and Islamic terrorism.

Online activism and extremism is emerging as in important component of geopolitics. McCaughey and Ayers (2013) define online activism as ”a politically motivated movement relying on the Internet”, and powerful political campaigns are now being waged on OSNs as well (Ferrara et al., 2014; Forelle et al., 2015; Ratkiewicz et al., a,b). OSNs have curated political discussion associated with relatively unenexpected electoral outcomes like Brexit (Mangold, 2016) and the 2016 United States Presidential Election (Ferrara et al., 2016a). Effective social-media-aided political activism has been observed in the Middle East (Abokhodair et al., 2015), as well as in Eastern Europe (Mungiu-Pippidi and Munteanu, 2009; Szostek). In some cases these powerful campaigns have organized large numbers of users whos views could be viewed as “extreme.” Lake (2002) define political extremism using two attributes. Extremists' political beliefs and objectives are such that few would find them acceptable. Secondly, extremists lack the means to achieve their objectives through traditional political means. Our goal in this work is to present unsupervised methods to detect social media users that promote extremism. Clearly, the Isalamic State of Iraq al-Sham's (ISIS) rise to power provides an example of how an extremist organization can effectively use OSNs for strategic messaging and recruitment (Berger and Morgan, 2015b; Veilleux-Lepage, 2015), but many of the competing factions in Syria currently weild similar online campaigns. Propaganda dissemination within OSNs has become commonplace among many activist and extremist causes.

In each of these political movements, online communities appear to play a vital role in the dissemination of news and the shaping of opinion. Insular online communities have been observed in the 2016 United States Presidential Election and proven to be prone to misinformation Benkler et al.. The results observed with respect to terrorism appear consistent with this phenomenon as well (Berger and Morgan, 2015a; Veilleux-Lepage, 2014). The ability to quickly identify and understand these communities will be essential to informing effective intervention strategies in the future.

Community detection refers to the task of identifying subgroups within a graph where nodes are more densely connected to one another than to the rest of the graph, and has been studied extensively. Fortunato (2010) provide an extensive survey, and Papadopoulos et al. (2012) provide a survey specifically oriented on community detection in social media. Our ultimate goal is to detect and study online activist and extremist communities. We presented Iterative Vertex Clustering and Classification in Chapter 3 and illustrated the ability to detect large online extremist communities (OECs) as a supervised learning task, but to make the framework generalize more broadly tailored unsupervised methods are needed. In Chapter 3 we were able to leverage metadata to infer positive case labels which will not often be the case. Unsupervised methods capable of identifying sets of users who form highly active dense online communities. These are often referred to as disussion cores (Pei et al., 2014), and could be studied directly or used as positive case instances for supervised learning. These groups often use tools like following, sharing hash tags and direct messaging to self organize, their social networks are described by complex graph structure with many edge types. Such graphs are often referred to as heterogeneous graphs Sun et al. (2010).

In this paper we present two unsupervised methodologies to identify online discussion cores. In both cases, the OSN is modeled as a heterogeneous graph, and only groups with interconnectedness across multiple edge types are returned. We first present ensemble agreement clustering (EAC), an ensemble-based methodology that returns only users who are co-clustered across all defined edge types in the methodology. We then present Heterogeneous Dense Subgraph Detection, a hierarchical clustering method designed to return only clusters that meet a specific density threshold. We present case studies of detected subgraphs posting content related to ongoing conflicts in Syria and Eastern Ukraine.

This manuscript is structured as follows. Section 4.2 will present relevant research related to community detection in heterogeneous networks as well dense subgraph based community detection methods. In section 4.5 we present EAC and HDSD and compare the algorithm' performance when subjected to varying levels of noise. We then present case studies of Twitter communities focused on the Syrian Revolution and Euromaidan Movement in Section 4.5. Finally we discuss the strengths and limitations of our findings and summarize our results in Sections 4.6 and 4.7.

4.2 Background

The problem of community detection has been widely studied within the context of large-scale social networks and is well documented in works like Fortunato (2010); Papadopoulos et al. (2012). Community detection algorithms attempt to identify groups of vertices more densely connected to one another than to the rest of the network. Social networks extracted from social media however present unique challenges due to their size and high clustering coefficients (Girvan and Newman, 2002). Furthermore, ties in online social networks like Twitter are widely recognized as having high social dimension, in that users ties represent different types of relationships (Boccaletti et al., 2006; Miller et al., 2011a; Wang et al., 2010). Often times simultaneous membership is of interest and Zhang et al. (2007) presents a methodology using a similar features to those presented in Chapter 3. Our interest however is to detect the core members of a specific online community and therefore are interested in a hard clustering where users are assigned to one and only one group.

The Louvain Grouping algorithm presented in Blondel et al. (2008) is widely used for community optimization within the network science community. Louvain grouping uses a similar objective function as the Newman-Girvan algorithm Newman and Girvan (2004), but is more computationally efficient. In community optimization algorithms, the graph is partitioned into k communities based on an optimization problem that centers around minimizing inter-community connections where k is unspecified. Both Newman and Blondel find these communities by maximizing *modularity*. The modularity of a graph is defined in Equation C.1. In Equation C.1, the variable $A_{i,j}$ represents the weight of the edge between nodes i and j, $k_i = \sum_j A_{i,j}$ is the sum of the weights of the edges attached to vertex i, c_i is the community to which vertex i is assigned, $\delta(u, v)$ is the inverse identity function, and $m = \frac{1}{2}\sum_{i,j} A_{i,j}$.

$$Q = \frac{1}{2m}\sum_{i,j} = [A_{i,j} - \frac{k_i k_j}{2m}]\delta(c_i, c_j), \tag{4.1}$$

Louvain grouping maximizes modularity (Eq. C.1) by first sequentially calculating the modularity gain associated with adding vertex i to its nearest neighbor j's community, and always selects the individual assignment which provides the greatest gain. In its second step communities are replaced by super-vertices, and two super-vertices are connected if there is at least an edge between vertices of the corresponding communities. These steps are repeated recursively until modularity no longer increases. In our ensemble we will develop pairs of community assignments for each user by running this algorithm on the reciprocal mention and reciprocal following networks. However, Blondel et al. had not been extended to heterogeneous graphs, and clusters based only on co-mention or co-friendship ties return clusters that fail to isolate OEC members sufficiently Tang and Liu (2011).

One alternative is to develop graph representations that account for a greater number of the user behaviors used for group organization within OSNs. Mucha et al. (2010) present a method to detect communities in time-dependent multiplex networks by extending community quality functions, like Equation C.1, to account for null models within each graph representation within the heterogeneous graph. Alternatively, Sun et al. (2010) presents an iterative approach that finds similarity-based separations based on multimode graphs. However, the scale associated with OEC-related searches make these methods intractable. The results presented in Chapter 3 imply that a multiplex and multi-node or heterogeneous graph representation could help overcome problems with high social dimension.

Another area of research which could prove useful to our task is that of dense subgraph detection. Some have argued that highly active sets of core users account for the proliferation of content within online communities (Pei et al., 2014). At the node level these users can be identified using a k-core based approach, where users' centrality is measured by their membership in the most dense k-cores across social media platforms. We propose that unsupervised methods able to find highly connected sets of users could define "discussion cores" within online communities, and dense subgraph detection would be an appropriate framework. Dense subgraph detection can be preferable when a complete clustering of the data is not desired. Instead the researcher is merely interested in highly connected communities which may represent a communities most active members. Chen and Saad (2012) present a scalable hierarchical clustering technique community detection technique which returns only clusters which meet an a priori density threshold. Although the method scales well and returns useful results, it has not been extended to a heterogeneous graph representation.

Ideally, the feature space presented in Chapter 3, Φ_{IVCC}, could be clustered in a manner where sets of positive case instances could be easily identified. To estimate the performance of an unsupervised clustering of Φ_{IVCC} we first assume positive case training instances as well as accounts predicted as ISIS supporting and subsequently suspended as ground truth. This assumption will be discussed in greater detail in Section 4.4. Using kmeans, as is common in spectral clustering, with $k = 100$ yields clusters with precision no greater than 67%. Such results could not be effectively used as positive training instances. The ROC curve associated with Φ_{IVCC} is depicted in green in Figure 4.3. Due to the goal of quickly building positive case training instances only clusters of size greater than 100 are inspected. A similar curve for Louvain Grouping of the undirected mention network is depicted in red. The plot illustrates the limitations of each method with respect to "discussion core" detection and motivates the remainder of this chapter.

4.3 Methods

Each method we present is used on similarly collected datasets. In each case we collect users' activity using snowball sampling Goodman (1961), a non-random sampling technique where a set of individuals is chosen as "seed agents", and the k most frequent accounts followed by each seed agent are taken as members of the sample. Although this technique is not random and prone to bias, it is often used when trying to sample hidden populations and provides social structure that hash tag or geographically fences searches do not. We seed searches with sets of users that are known members of a larger OEC of interest. We then collect additional users' activity based on seed users' social ties. In some cases we iterate this process in "hops." These searches often result in hundreds of thousands of users, many of whom will not be of interest. The subsequent task is to efficiently find structures worthy of manual inspection.

Our task is to detect discussion cores of users who's activist cause is easily identifiable. To do so we will attempt to find groups who have organized using multiple edge types. We define our heterogeneous graph as $G = (V_1, V_2, .., V_n, E_1, E_2, .., E_m)$. Where G is an undirected, weighted graph with vertex classes $V_1...V_n$. Each contains vertices $v_{n,1}..v_{n,j}$ with one or more edge types $E_1, E_2, .., E_m$. We define a subset of targeted vertices $A_t \subseteq V_t$ and denote its complement as $\tilde{A}_t$. Our goal is to use unsupervised methods to identify a subset of A_t with high precision in order to subsequently train a classifier to partition the network.

4.3.1 Ensemble Agreement Clustering (EAC)

To detect these core communities we develop an ensemble of clustering techniques that leverage users' commonality across following ties, mention ties, and hash tag use. Like Mucha et al. (2010) we are interested in modularity maximization across a multitude of graph representations. However, instead of assigning all users to clusters, we are interested in sets of users that are co-clustered across R different graph representations. Specifically we are looking for sets of users where clustering algorithms are in "agreement" across multiple edge types, thus returning a subset or users we hope are more likely to be organized around an identifiable activist cause.

To present our method we introduce the following notation. Given a set of vertices $V = \{V_1, V_2, ..., V_n\}$, we look to develop a cluster ensemble for vertex type V_1. A cluster ensemble is a set of clustering solutions, represented as $C = \{C^1, C^2, ..., C^R\}$ where R is the ensemble size. Each solution within the set, C^T, is simply a partition of V into k_T disjoint clusters as $C^T = \{C_1^T, C_2^T, ..., C_{k_T}^T\}$ where T is defined by edge-type and clustering method. Each solution could also be represented as a vector, $\vec{c^T}$, of length n where $\vec{c^T}_i \in 1, 2, .., k_T$. We can then define our solution C^S by assigning each vertex to the clusters defined by its vector of assignments $\vec{c_i^S} = c_i^1, c_i^2, ..., c_n^R$. This leaves a strict subset of vertices that are co-clustered across all R routines within the ensemble.

When applied to Twitter data we develop community assignments based on users' mention and friend ties using the Louvain Grouping algorithm (Blondel et al., 2008), as it is commonly used for community detection in large graphs (Fortunato, 2010; Traud et al., 2012). We also develop community assignments for each users' hash tag behaviors using bipartite graph partitioning (Dhillon, 2001b) which is presented in greater detail in Chapter 5. We set the parameter k for the number of distinct user/hashtag co-clusters based on the characteristics of the cluster results of the co-mention and co-following graphs. We count the number of clusters greater than size m in both cases and take the maximum. Given the results of all three clustering methods, we extract these core communities where sets of users are co-clustered across all 3 graphs.

4.3.2 Heterogeneous Dense Subgraph Detection (HDSD)

Since discussion cores are often characterized by high levels of activity among a smaller number of users and sometimes detected as k-cores (Bessi and Ferrara, 2016), dense subgraph detection is an intuitive approach for OEC discussion core detection. Chen and Saad (2012) present dense subgraph detection as a community detection method, and we will extend there work to a heterogeneous graph.

Chen and Saad (2012) explain their approach as follows. Given a sparse undirected graph G and a density threshold d_min, they construct G' with weighted adjacency matrix A and construct an adjacency matrix M:

$$M(i,j) = \frac{\langle A(:,i), A(:,j) \rangle}{||A(:,i)||||A(:,j)||} \tag{4.2}$$

They then construct an array C of the tuples $(i, j, M(i,j))$, for all nonzero edges $M(i,j)$ where $i > j$, sorted in ascending order with respect to $M(i,j)$. They then incrementally delete edges in G' based the tuple order in M and calculate density of the induced components of G'. The authors are able to show increased speed and algorithmic equivalence however by constructing hierarchy T according to the sorted vertex pairs of C. They subsequently calculate density by traversing T from the top down and calculating the density of sub-graph G' consisting of all child nodes $v \in V$ of a parent node $p \in T$. When one of the subgraphs G' meets threshold d_{min} it is returned.

We use an approach almost identical to Chen and Saad (2012) except that we define M as a weighted combination of similarity graphs within G. Therefore for each of our R edge types associated with V_1 we define M_R in the same manner as Equation 4.2 but with the corresponding adjacency matrix A_R.

$$M_R(i,j) = \frac{\langle A_R(:,i), A_R(:,j)\rangle}{||A_R(:,i)||||A_R(:,j)||} \tag{4.3}$$

Because M_R can become dense when applied to large sparse graphs, we retain only the largest $t_r = 2 \times |E_r|$ edges in each M_r. We then define a weight vector $\vec{w}$ of length R and construct our heterogeneous similarity matrix M_H in the following manner:

$$M_H(i,j) = \sqrt{w_1 M_1(i,j)^2 + w_2 M_2(i,j)^2 + ... + w_R M_R(i,j)^2} \tag{4.4}$$

and construct C by sorting the t largest $t_H = max(t_1, t_2, ..., t_R)$ edge tuples of M_H in descending order. T is then build as a binary tree based on the ordered tuples of C. Unlike Chen and Saad (2012), we search T and set our minimum density threshold using weighted combination of subgraph densities. Again we use our weight vector, $\vec{w}$. In each of our undirected, symmetric user graphs we calculate density as follows:

$$d_R = \frac{|E_R|}{V_R(V_R-1)/2} \tag{4.5}$$

In the case of bimodal or bipartite graphs we simply binerize the similarity graph and calculate density as depicted in Equation 4.5. We finally define heterogeneous density as:

$$d_H(i,j) = w_1 d_1 + w_2 d_2 + ... + w_R d_R \tag{4.6}$$

We then remove parent vertices from the top of the hierarchy T until G' meets our density threshold or breaks into components. If those components meet our density threshold, they are returned as search results. If they do not, they are added to a queue, and the algorithm is repeated. Our approach is summarized in Algorithm 2. For a detailed discussion of the performance and scalability of this algorithm we refer readers to Chen and Saad (2012) as our extension with respect to heterogeneous graphs does not significantly alter computational complexity.

When specifically applied to Twitter data we define undirected unimodal graphs of users' co-mention, G_m , and co-following, G_f, ties. We also define G_h a bipartite graph of users and hash tags where each edge corresponds to the number of times a user i posted with hash tag j. We also select d_min based on values of d_M associated with Louvain Groups Blondel et al. (2008) associated with G_m and G_f. We will discuss parameter selection and sensitivity further in Section 6.5.

4.4 Evaluation

As is often when evaluating performance of community detection algorithms on real world graphs, in our case ground truth in unknown. However, we are not interested in estimating the performance of EAC and HDSD across all graph topologies; we simply want to evaluate these

```
Algorithm 1: Detecting dense cores in a heterogeneous graph
  Input : Given a large sparse, weighted, undirected heterogeneous graph $G$ with $R$ distinct edge
          types corresponding to node class $V_1$ and density threshold $d_{min}$
1 Compute Matrix $M$ as defined in 4.4.
2 Sort the largest $t$ non-zero entries of $M$ in ascending order, where $t = nz(A)$. Denote $C$ the sorted
  array.
3 Construct the hierarchy $T$ according to the sorted vertex pairs designated by $C$.
4 Extract Subgraphs of $G$ where $d_M \geq d_{min}$ as defined in 4.6 using the same algorithmic approach
  defined in Chen and Saad (2012).
```

methods' ability to find activist and extremist communities embedded in large snowball samples within Twitter. We are interested as to wether these methods generalize across multiple types of groups within this specific type of search within this specific OSN.

Evaluating community detection has typically been done using either synthetic graphs as in Chen and Saad (2012) or inferring ground truth from metadata (Berger and Morgan, 2015b; Zachary, 1977). Both methods' shortcomings have been well documented. Synthetic graphs often fail to reproduce complex structures in real world graphs because the graph generating process is often not fully understood (Peel et al., 2016). Our heteregeneous graphs are clearly the result of a highly complex generation process as observed by their complex structures (Girvan and Newman, 2002) and not adequately modelled by existing synthetic graph models. Inferring ground truth from metadata can confound results as well. If the metadata is irrelevant to the structure of the network, or algorithms and metadata "capture different aspects of the networks structure" results can be highly misleading (Peel et al., 2016). We will use partially synthetic graphs to highlight the complexity of our graphs, and subsequently approach evaluation by leveraging unique metadata available in the ISIS NOV14 Twitter dataset. We will subsequently use feedback from intelligence experts to validate EAC results in Section 4.5.

In our case even partially synthetic graphs alter graph structure too much to evaluate performance. Modularity-Based Stochastic Block Modeling as a means to partially alter existing real world graphs while maintaing more conplex structure than more naive methods. For a detailed explanation of this method is provided in Appendix C. The method uses Louvain grouping (Blondel et al., 2008) to identify a block model, and adds, removes, or permutes an apriori percentage of edges in off-diagonal blocks and blocks that meet a maximum size threshold. Large blocks along the diagonal are returned, and the algorithm is executed resursively.

To create partially synthetic graphs with known ground truth, the 31,327 users returned from the ISIS NOV14 dataset by either EAC or HDSD are assumed to be members of a community of interest and thus ground truth. We create three partially synthetic datasets by increasing the edge count within the community of interest by $\xi = \{.05, .10, .25\}$ percent in each using Modularity-Based Stochastic Block Modeling. We augment the bipartite user and hashtag graph by randomly adding user-hashtag ties from within the community of interest. Finally, we reduce variability by binerizing each of the graphs since we do not have a responsible means to select edge weights. Figure 4.1 illustrates the change in graph structure imposed by this method. The x-axis depicts the percentage of edges added to the community, ξ, and the y-axis depicts recall. The figure illustrates how adding signal infact reduces recall in both algorithms. However, comparison of

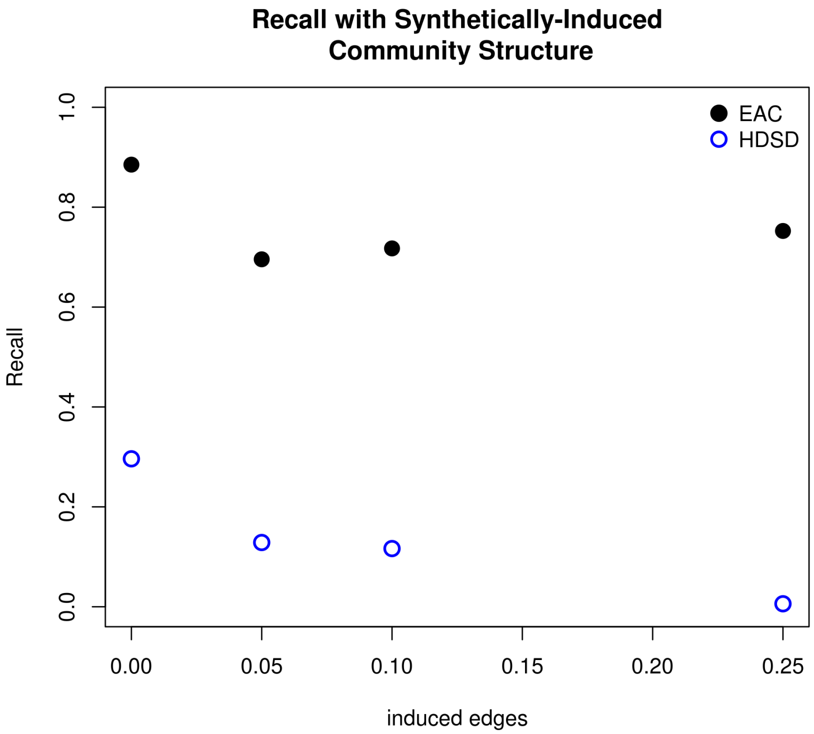


Figure 4.1: Depicts recall when augmentation of community is imposed on $\Theta_{\cup}$ at varying percentages using Modularity-Based Recursive Stochastic Block Modeling as described in Appendix C.

EAC and HDSD highlights interesting differences between the algorithms. EAC recall increases from $\xi = (.05, .25)$ while HDSD recall continues to decrease. It is possible that the way we have parameterized HDSD, results are focused on finding smaller dense subgraphs and the algorithm is unable to identify increased modularity at in large scale communities, and though we have not conducted an study with respect to parametric sensitivity results in Section **??** are consistent with this hypothesis.

To responsibly infer ground truth from metadata, the metadata must be relevant to the clustering task and not capture different aspects of network structure. The ISIS NOV14 provides a unique opportunity to validate both requirements. In November of 2014 we seeded a two-hop snowball sample of influential ISIS propagandists' Carter et al. (2014) following ties in hopes of detecting the ISIS-supporting community on Twitter. The search resulted in 118,879 user account profiles and roughly 186 million tweets. As discussed in Chapter 3, we were able to responsibly infer ISIS sympathizers based on a heuristic which included manual validation, soft clustering, and Twitter suspension rates. This heuristic enabled us to identify 5,126 positive case instances of ISIS sympathizers and subsequently predict over 18,000 additional sypathizers at high accuracy. As of March 21, 2017 over 18,000 accounts, 7,823 of which had been predicted as ISIS-supporting, had been suspended by Twitter. These numbers are not surprising given Twitter's aggressive campaign which had suspended over 360,000 accounts for supporting violent extremism as of August of 2016 (Benner, 2016). We argue this metadata is relevant and captures at least some of the network structure we are interested in as many of these suspensions were likely a result of user participation in online extremist communities. However, we also acknowledge that Twitter suspends accounts for spam as well as abusive behaviors twi. To account for this we develop two ground truth datasets. The first, Θ_S refers to all users within the ISIS NOV14 dataset that have subsequently been suspended. We recognize that Θ_S likely contains some users who were suspended for other reasons, and therefore construct $\Theta_\cap$ consisting of accounts both suspended and predicted as ISIS-sympathizers in Chapter 3.

Figure 4.2 highlights suspension rates within the ISIS NOV14 data set as well as the suspension rates associated with each clustering routine's results. The plot depicts a venn diagram illustrating the relative size of the dataset (118,648 users), the accounts returned by EAC (22,086 users , 123 groups), the accounts returned by HDSD (17,542 users, 26 groups), and the intersection of the two algorithms (8,301 users). Twitter suspension rates associated with each subset are depicted with Φ_R. As can be seen, accounts returned by EAC have suspension rates of 28%, nearly 9% higher than the results returned by HDSD, and we have found that many of the groups returned by HEAC have suspension rates exceeding 90%.

Specific clusters with high rates of positive case instances could be used to rapibly build training sets. Figure 4.3 depicts curves similar to receiver operator curves (ROC) of EAC (black), HDSD (blue), Louvain Grouping of the undirected mention graph (red), and a kmeans clustering of Φ_{IVCC} (green) applied to the ISIS NOV14 dataset with two different ground truth data sets. The left panel depicts the 12,415 accounts predicted as ISIS Supporting in Chapter 3 and suspended by Twitter as of March 21, 2017. We will refer to this set of positive cases as $\Theta_\cap$. The right panel assumes the 29,674 accounts that were either suspended by Twitter or predicted as ISIS supporting to be positive cases. We will refer to this set of positive cases as $\Theta_\cup$. We are interested in finding discussion cores or groups with consisting of high percentages of OEC members. These curves are different from ROC curves in that we inspect only clusters with 100

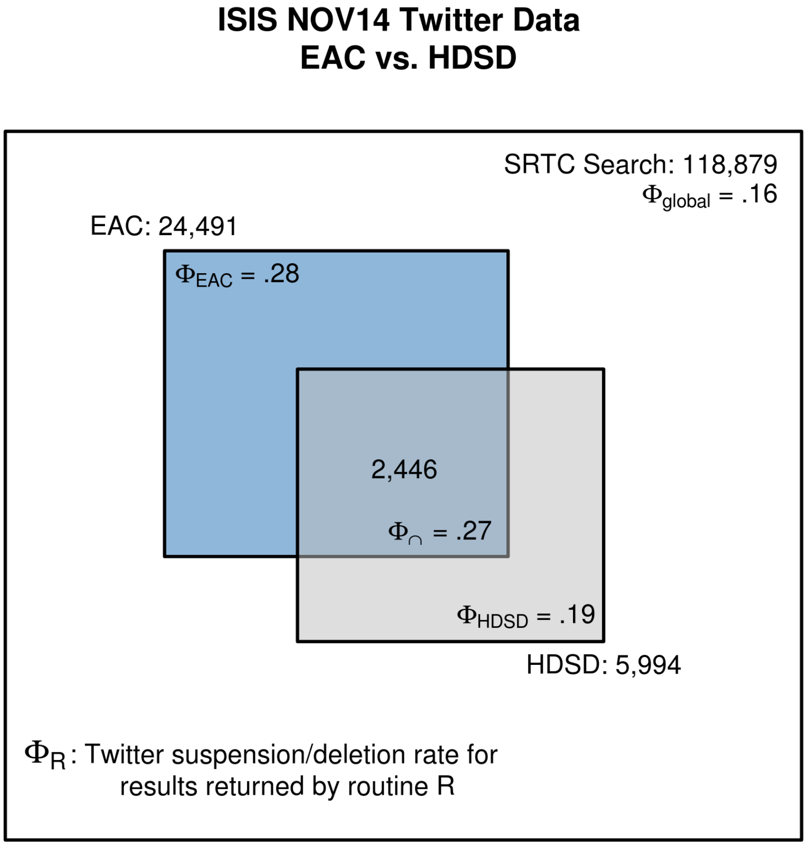


Figure 4.2: Depicts EAC and HDSD results when performed on the ISIS NOV14 Twitter search. Both size in terms of number of users and Twitter suspension rate as of March 21, 2017 are depicted. The plot highlights EAC's ability to rediscover suspended accounts at higher rates than HDSD.

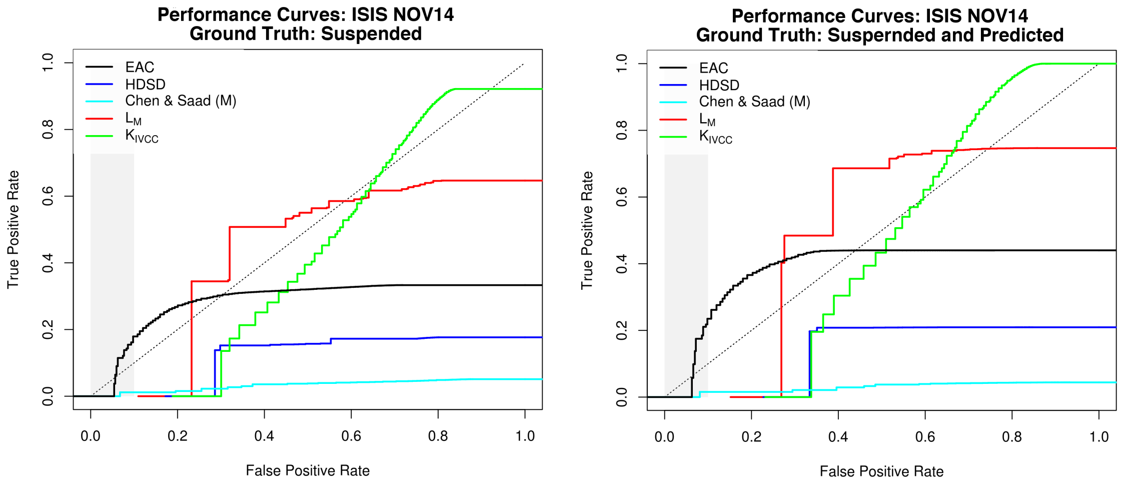


Figure 4.3: depicts performance curves of various unsupervised methods for OEC discussion core detection using two different assumptions of ground truth. The left panel assumes the 18,6302 accounts within the ISIS NOV14 dataset that have been suspended or deleted as ground truth, and right panel depicts the 12,415 accounts predicted as ISIS Supporting in Chapter 3 and suspended by Twitter as of March 21, 2017 as ground truth. Results of kmeans clustering with $k = 100$ of the IVCC feature space presented in Chapter 3 are depicted in green, EAC in black, HDSD in blue, Chen and Saad's dense subgraph detection of the undirected mention graph in cyan, and Louvain groups in red. With each algorithm, only clusters with greater than 100 users are returned. The plot highlights both the limitations of existing methods like Louvain grouping and the use of IVCC as an unsupervised approach and the superior performance of EAC.

or more users in an attempt to evaluate these algorithms ability to enable analysts to efficiently build training sets through manual inspection. If a high enough percentage of users associated with a given cluster meet the definition of an OEC member the entire cluster could be used as positive training instances. As can be seen EAC clearly outperforms all other methods at this task. EAC returns 123 large clusters. The $\Theta_\cap$ set of assumed positive case instances yield X clusters where over 90% of the users belong to the ground truth user set resulting in 1794 positive case instances for training. Furthermore, 13,544 negative case training instances could be identified at over 95% precision. The $\Theta_\cup$ set of assumed positive case instances yields 19 clusters with over 90% precision resulting in 4850 positive case instances. None of the alternative algorithms provide return any groups with greater than 90% precision in either dataset. The closest alternative would be Louvain Grouping 4823 instances at 73% precision using the $\Theta_\cap$ set as ground truth. For the purpose of identifying discussion cores, EAC performance appears superior.

It is possible that these estimates of performance are confounded by biases in Twitter's suspension campaign. Furthermore, Twitter does not provide us with a reason for their decision to suspend specific accounts. If in fact Twitter suspended these accounts for because of spam distribution, we would not expect to see the rich community structure our algorithms are designed

Syrian Revolution Twitter Community (SRTC)

Date	Seed Accounts	Search Return	+ Cases	Detected OEC
NOV15	16,538	91,256	3,572	8,126
MAR16	3,295	87,724	2,529	9,086
DEC16	4,258	118,879	4,567	NA

Euromaidan Twitter Community (ETC)

Date	Seed Accounts	Search Return	+ Cases	Detected OEC
AUG15	8	92,295	1,221	4,307
MAR17	1,175	92,076	4107	NA

Table 4.1: Describes historical performance of EAC, and multiplex vertex classification for two updates the Syrian Revolution Twitter Community (top panel) and the Euromaidan Twitter Community (bottom panel).

to detect. Moreover, these results are consistent with qualitative analysis provided by trained intelligence analysts in Section 4.5. Although HDSD failed to precisely detect discussion cores, we observe some unique qualities with respect to returned clusters which will be discussed in in detail in Section 4.5.

4.5 Case Studies

In this section we will present case studies of dynamic communities organized around common activist causes. The first case study involves Twitter users in the Middle East who actively share content associated with the Syrian Revolution. The second case study focuses on the ongoing Euromaidan movement in Ukraine. In both cases nearly all users meet the definition of online activism, with some meeting our definition of online extremism. Our case studies will be largely consistent with the results presented in Section 4.4 and confirm EAC as the more effective methodology for building positive case training instances of an OEC of interest. In both cases we incorporate feedback from intelligence analysts with regional expertise and language proficiency.

Both case studies the communities have been periodically updated multiple times over the past two years. With each update, we execute a 1 hop snowball sample of OEC members' mention or following ties and use EAC to identify OEC discussion cores through manual inspection. In each case we define the user behaviors of an OEC member and manually inspect randomly sampled users from clusters. Clusters with estimated positive case instance rates over 90% we claim to be useful for follow on supervised OEC detection with multiplex vertex classification. Within each case study we will describe the OEC of interest and the dynamics of the community over time. We have not studies HDSD until recently, and therefore will only compare HDSD and EAC using the most recent update of each case study.

Syrian Revolution Twitter Community

The Syrian Revolution Twitter Community is a periodically updated set of Twitter users based originally on the ISIS-supporting OEC presented Chapter 3. We define a member as a Twitter user who positively affirms the leadership, ideology, fighters, or call to Jihad of any of the

SRTC DEC16 search data and results

Method	Large Clusters	Users	Positive Case Instances
HDSD	12	5,994	NA
EAC	17	15,614	4,125
Louvain	18	80,038	None

Table 4.2: Describes HDSD and EAC results applied to the SRTC DEC16 search data. For purpose of comparison between Louvain clusters and EAC and limited access to regional expertise within the US department of defense only clusters with more than 500 users were inspected. Because HDSD clusters are typically smaller, we inspect clusters larger than 100 users. Returned clusters were manually inspected using random sampling.

known Jihadist groups engaged in ongoing operations in Northern Iraq and Syria, and activity of members spans a continuum that ranges from online activism to extremism.

The community has been updated in November 2015, March 2016, and December 2016; throughout each update, EAC enabled us to identify discussion cores with over 90% of users meeting our criteria for OEC membership. The community was originally detected using a 1-hop snowball sample of the 16,538 members of the ISIS-supporting OEC presented in Section 4.4. The November 2015 SRTC Twitter search yielded 91,256 users. EAC output from the search returned 62 groups consisting of 22,949 users. Manual inspection of clusters identified 6 clusters where over 90% of sampled users met the criteria for OEC membership resulting in 3,572 positive case training instances which we subsequently used to detect the November 2015 instance of the SRTC through multiplex vertex classification. We conducted similar updates in March and December of 2016 with similar results. In each case EAC was effective for identification of OEC discussion cores and useful for efficiently building large training sets. For detailed figures see Table 4.1.

For the purpose of quantifying the utility of EAC, we gained access to intelligence analysts with deployment experience in the Middle East from the Secretary of the Air Force, Administrative Assistants Staff. The December 2016 update SRTC search was collected based on 4,258 SRTC members from the March 2016 SRTC update. The December 2016 search yielded 118,879 users based on seed agents' mention activity. We then ran EAC as well as Louvain grouping Blondel et al. (2008) of the co-mention graph. Due to our limited access to analysts, we limited Louvain and EAC clusters to those containing more than 500 users. The number of groups and size of each cluster are provided in Table 4.2.

To measure levels of activism and extremism, as well as the uniformity and specificity of clusters we provided a web survey. The survey randomly selected ten users from each of the 17 and 18 clusters returned by EAC and Louvain respectively. Analysts were blind to each clusters algorithmic approach and size, and each analysts cluster order and randomly selected accounts were unique. For each user, the analyst viewed the users last 15 tweets and labelled them as exhibiting:

- **No promotion (0):**
- **Online Awareness Promotion (1):** online activity promoting awareness of an identifiable cause, interest, or product McCaughey and Ayers (2013)
- **Online Activism (2):** online activity promoting awareness of an identifiable political

cause often advocating agreement with specific groups or outcomes McCaughey and Ayers (2013)

- **Online Extremism (3):** online activism advocating support of groups or causes that in any distribution of opinion would lie on one of the“tails” (Lake, 2002)

Once the analysts scored 10 randomly selected users within a given cluster, they provided an assessment of the *specificity* and *uniformity* of the group. We define both terms as follows: **Uniformity:** the extent to which users' activist agendas appear uniform

- None (0): no visible uniformity of cause among group members
- Weak (1): 3-4 users display similar interests and intensity
- Moderate (2): 5-6 users display similar interests and intensity
- High (3): 7-10 users display similar interests and intensity

Specificity:

- None (0): no visible specificity of cause among group members
- Weak (1): regional or generally accepted within either Sunni or Shia Islam
- Moderate (2): Country level geospatial similarity or denominational specificity within Sunni or Shia Islam
- High (3): City, Party, or Organizational Specificity

Analysts survey results were consistent with the results we observed in Section 4.4. Analysts feedback implies EAC groups were more uniform with respect to activism and extremism; furthermore, their feedback indicates that EAC clusters have higher specificity. The left panel of Figure 4.4 depicts box plots of the within cluster standard deviation of user scores. As the figure depicts, EAC clusters appear to be more uniform than those provided by Louvain grouping of the undirected mention network. Although quantifying statistical significance is challenging due to the inability to assume independence between users or groups, the box plot is consistent with our assertion of increased uniformity within EAC clusters. The right panel depicts mean group specificity and uniformity scores depicted for EAC (black dots) and Louvain (red crosses) clusters. The means for each cluster type are depicted with dotted lines. It also appears that analysts found EAC groups to show higher specificity, which is consistent with our observations throughout each update of the SRTC and Euromaidan datasets.

Only two clusters had mean specificity scores above 2.5 and discussed content related to ongoing operations in Syria and Northern Iraq. Both clusters were returned from EAC and resulted in a set of 4,125 users that could be used as positive case training instances to update the SRTC. Although we did not have analyst support to evaluate HDSD groups, we randomly sampled output and found 3 groups totaling 433 users adequate for positive case training instance use. Again, analyst feedback is consistent with our results from Section 4.4 implying EACs superior performance for training set development.

Based on observations within our case study, both algorithms provide useful results apart from positive case training instances within our OEC of interest. Both algorithms return activist or extremist clusters unique from our OEC of interest. For example, HDSD returned a few highly specific clusters focused on Islamic Extremism in Libya as well as a highly uniform group in Qatar. Groups supporting the Muslim Brotherhood in Egypt and ongoing operations in Yemen were identified by analysts within EAC results. The November 2015 update search of the SRTC

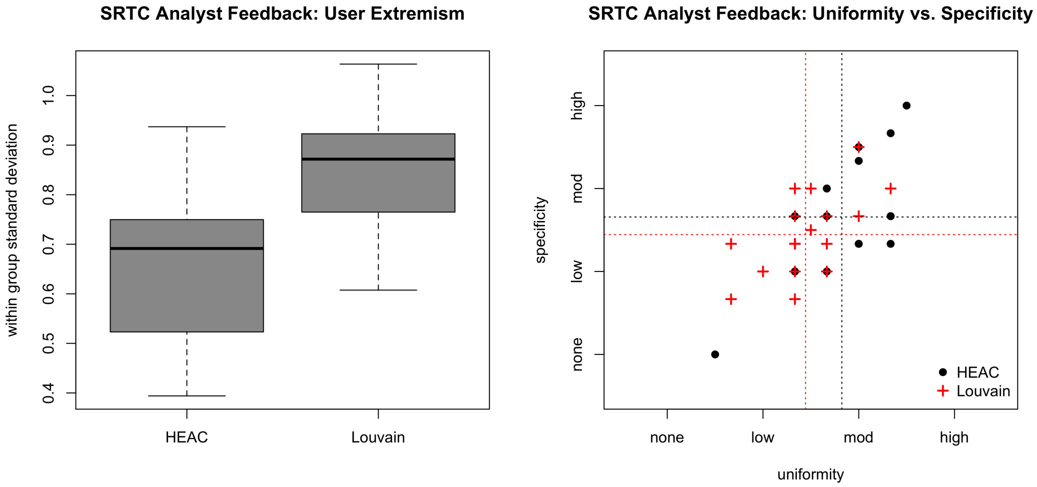


Figure 4.4: Summarizes Air Force analyst feedback concerning EAC and Louvain clustering results of the SRTC DEC16 Twitter Search.

contained a large cluster of Scots blogging for independence from the United Kingdom. HDSD also returns clusters that could be influenced by botnet activity. We observe one image sharing community that could very well organized around a social botnet. We identify similar clusters in the ETC case study as well.

Euromaidan Twitter Search (MAR17)

The Euromaidan movement started as a series of protests in November 2013, where large numbers began to call for the removal of then President Viktor Yanukovych. These protests reached their peak in February 2015, ultimately leading to the removal of many of Ynukovych's senior officials, and were a precursor to Russia's subsequent occupation of Crimea. Despite the installation of a new government, a substantial online activist community continues to oppose Russian influence in the Ukraine and are often described as part of the Euromaidan Movement (Szostek). We will refer to this community as the Euromaidan Twitter Community (ETC). Here, although strong negative sentiment toward the current Ukrainian government is observed, the online activism seen largely advocates change through legitimate government processes. Thus, while we acknowledge that little "extremism" exists in this community, this community is of strategic interest to organizations like the North American Treaty Organization (NATO) due to its relevance to ongoing geopolitical events in the region. This community was extracted originally from a two-hop snowball sample of 8 known Euromaidan movement members' mention ties in March 2014. The search resulted in 92,295 Twitter accounts, and manual inspection of EAC output yielded a community of 1,221 accounts actively supporting the movement. Using methods similar to those described in the last case study, we updated the ETC again in March of 2017 yielding a set of 4,083 community members. Details with respect to searches and results are provided in Table 4.3.

To identify community members we again manually inspected EAC and HDSD results and observe similar algorithmic performance. Manual inspection of 4 of the 6 largest EAC clusters

Euromaidan MAR17 search data and results

Method	Large Clusters	Users	Positive Case Instances
HDSD	27	10,552	1,192
EAC	15	8,287	3,285
$\cup_{HDSD,EAC}$			4,083
Louvain	21	66,695	None

Table 4.3: Describes HDSD and EAC results applied to the ETC MAR17 search data. For purpose of comparison between Louvain clusters and EAC clusters with more than 300 users were inspected. Because HDSD clusters are typically smaller, we inspect clusters larger than 100 users. Returned clusters were manually inspected using random sampling.

yields 3,285 instances of ETC members. These clusters were subsequently validated by advisors from the United States Army Asymmetric Warfare Group with extensive experience working with the Ukrainian Army. HDSD yielded 1,192 community members, 792 of which had not been identified in EAC results. Again EAC outperforms HDSD with respect to training set development, but we also again observe useful and unique characteristics in HDSD clusters.

Again both algorithms return clusters that are not useful for developing training instances within our OEC of interest, but useful for other tasks. Within the March 2017 ETC search data large clusters of apparently American users blogging about the 2016 United States Presidential Election can be observed. Similar right-leaning clusters appear to contain users in the United Kingdom and Germany. HDSD again returned a handful of highly localized clusters. Several groups contained users who's profile descriptions had over 80% agreement at the city level with groups located in Moscow, Ivanovo, and St. Petersburg. We also observed 2 clusters that appear to be botnets. One botnet consisted of over 100 users who post URLs promoting law services in Moscow; another appeared to consist of low level chatbots sharing pro-Russian content. Such results could be quite useful for other tasks and will be discussed in Section 4.6.

4.6 Strengths and Limitations

In Sections 4.4 and 4.6 we have shown the utility of using dense heterogeneous subgraph methods to rapidly build training sets for OEC detection. As the results of our sampling technique commonly return many users who are not of interest, a means to quickly remove users unlikely to be of interest and return clusters whose promotional causes are highly similar is quite useful. In the cases presented in this work, EAC appears to outperform HDSD, but other use cases or could very well lead to different results.

We also find the identification of activist clusters unrelated to our OEC of interest to be useful. The potential to identify competing discussion cores through these methods offers great potential in gaining understanding of the dynamic war of ideas currently observed in online social networks. It is possible that activism has a graph structure within online social networks and methods like HDSD and EAC could be used to better understand how the development of online extremism. The ability to efficiently identify related, competing discussion cores could offer valuable insight for strategic messaging efforts as well. In fact we will discuss applicable methods in Chapter 5.

Finally, the observation of possible botnets could also offer novel insight into tools used to

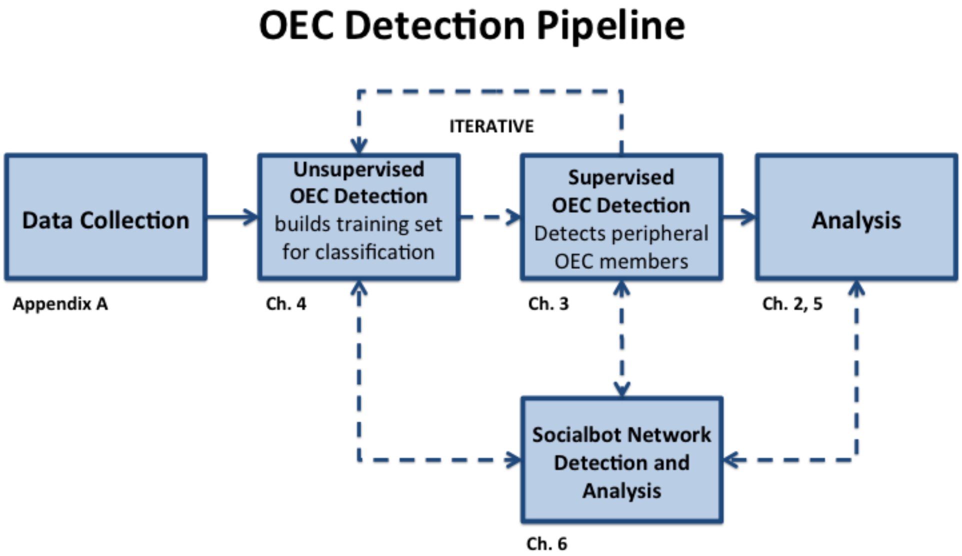

OEC Detection Pipeline
ITERATIVE
Data Collection
Appendix A
Unsupervised OEC Detection builds training set for classification
Ch. 4
Supervised OEC Detection Detects peripheral OEC members
Ch. 3
Analysis
Ch. 2, 5
Socialbot Network Detection and Analysis
Ch. 6

Figure 4.5: Depicts OEC detection as a methodological pipeline. Data Collection is conducted using snowball sampling then training sets are developed using unsupervised methods like HDSD and EAC. Larger portions of the OEC can then be detected using supervised learning.

manipulate these dynamic communities. We have often observed sub-group structure in OECs that could possibly be caused by social botnets. The structures promotional clusters observed in HDSD results could very well be examples of automated means to promote specific brands of activism as well. It is possible that cores exist within many of these online communities that are social botnets designed to deceptively influence online opinion. This will be discussed in detail in Chapter 6.

Although the results presented in this work are encouraging, they represent preliminary work in this important application of community detection. In both cases our results are applied to a very specific methodological pipeline. Our collection method of snowball sampling and our Twitter case study data limit the generality of our results. It is possible that these methods could be effective when applied to other heterogeneous networks within other applications. Furthermore, we have done very little to explore parametric sensitivity within both algorithms. It could be useful to combine the strengths of both algorithmic approaches. It is possible that the hierarchy T constructed in HDSD could be done using modularity as is done in Louvain grouping. Such a method could offer more parametric flexibility than EAC, and provide better results than HDSD.

The work presented in this chapter must be viewed in light of a larger methodological pipeline as depicted in Figure 4.5. In essence, we are developing a active learning pipeline. Active learning refers to a type of supervised learning where data is relatively cheap, but labels are expensive. If the latent structure can be leveraged in the dataset, than instances can be selected for labeling that offer more discriminatory value for the classifier (Settles, 2010b). EAC and HDSD offer the researcher an efficient means to identify informative positive case instances. It is possible that dense clusters that are unrelated to the OEC of interest could provide highly discriminatory information as well, and future work exploring this possibility could be quite fruitful. Again, we recognize that the results presented herein represent preliminary work in this important and emergent application of community detection.

4.7 Conclusion

In this chapter we have presented to methodologies to detect activist or extremist discussion cores on Twitter. We extend methods used in dense subgraph detection to return only clusters where users are similar in their following, mentioning, and hashtag patters. Both methods, ensemble agreement clustering (EAC) and Heterogeneous Dense Subgraph Detection (HDSD), provide useful results. However, we show that EAC provides superior performance when used to detect discussion cores within online extremist communities on Twitter. Furthermore, we illustrate with two case studies the ability to incorporate dense subgraph methods in an active learning framework for OEC detection. Finally, propose other possible applications of both methods and future research toward this important emerging application of community detection.

Chapter 5: Mining Online Communities to Inform Strategic Messaging: practical methods to identify community-level insights

5.1 Introduction

Social media's growing role in the shaping of public opinion has been observed in a variety of political Loader and Mercea and geo-political Herrick; Juris settings. Although the true significance of social media's role in actual political change resulting from this rise in use remains in questionDewey et al. (2012); Howard and Parks (2012); Hussain and Howard (2013); Nanabhay and Farmanfarmaian (2011), the emergence of social media as a means to at least motivate and expose desire for change has been recognized by scholars. The Arab Spring Lotan et al. (2011); Starbird and Palen (2012); Tufekci (2014); Wei et al.; Wolfsfeld et al. and the ongoing conflict in Crimea Pablo Barbera; Szostek, have both highlighted the emergent role of social media, and online social networks (OSN) more specifically, as facilitators of social activism.

Initially many viewed social media's role in the Arab Spring and Euromaidan Movement as positive examples of free speech; however, there also exists a downside to the ability of OSNs to act as platforms of mobilization. More specifically, the rise of ISIS has been largely propagated and highly publicized through OSNs Veilleux-Lepage (2014, 2015). Noting this, Western governments have begun attempts to mitigate the impacts of ISIS' propaganda approaches. However, they have found it challenging to participate in and influence online communities which show signs of extremism. In the United States, the recently-formed Global Engagement Center leads the State Department's effort to "coordinate, integrate, and synchronize government-wide communications activities directed at foreign audiences in order to counter the messaging and diminish the influence of international terrorist organizations" Dozier (2016). Mr. Michael Lumpkin, the group's director, recently spoke to the need for new approaches:

> *"So we need to, candidly, stop tweeting at terrorists. I think we need to focus on exposing the true nature of what Daesh is."*
>
> Mr. Michael Lumpkin
> NPR Interview March 3, 2016

A logical follow-up question to Mr. Lumpkin's statement would be "Expose to whom, and how?" We propose that quantitative analysis of large online extremist communities (OECs) could offer insight into the populations most susceptible to radicalization and could be used to inform strategic messaging or assess ongoing diplomatic or military efforts. Although methods to detect large online extremist communities have emerged in literature Benigni and Carley (2016); Benigni, Matthew et al.; Johnson et al., the ability to summarize community content in meaningful ways remains an open research question.

Online social networks now play in important role in helping people share information, particularly in times of unrest. As seen during the London riots of 2011 Glasgow and Fink as well as post-earthquake information dissemination in Japan Sakaki et al., online communities often organize around trending hash tags with short half lives. Political campaigns have shown similar patterns Weber et al.. Furthermore, organization around these hash tags often coalesces over time and is an important factor influencing information diffusion through social networks Chang. Url sharing and the ability to mention other users have become common attributes of many online social networks as well. In this paper we introduce three applications of existing methods to mine relevant content from large, online communities by taking advantage of tokens like hash tags, urls, and mentions. We discuss the following methods:

- Ideological User Clustering with Bipartite Spectral Graph Partitioning
- Narrative Mining with Hash Tag Co-occurrence Graph Clustering
- Identifying Radicalization with Directed url Sharing Networks

In each instance, we describe the data mining method in detail, present illustrative examples from online communities that exhibit varying levels of extremism, and subsequently discuss limitations and recommend future research. Our manuscript is organized as follows: we first describe the online communities we will use in Section 6.3, we then introduce the aforementioned methods in Sections 5.3, 5.4, and 5.5. Finally we summarize our findings in Section 5.6.

5.2 Data

As case studies we will present two *online extremist communities* , which are composed of one or more *OEC members*. We define these terms as follows:

> **online extremist community (OEC):** a social network of users who interact within social media in support of causes or goals posing a threat to state stability or human rights.
>
> **OEC member:** a Twitter user who's timeline shows unambiguous support to the OEC of interest. For example, if the user positively affirms the OEC's leadership or ideology, glorifies its fighters, or advocates its talking points.

It is important to note that a member's *support* is relative and in many cases not in violation of local law or Twitter's terms of use. In fact, these "passive supporters" appear to be an essential to the diffusion of online propaganda and therefore represent an important element of radicalization efforts Veilleux-Lepage (2015). In each presented case study we instantiate an n-hop snowball sampling strategy Goodman (1961) with known members of a desired community. We then remove non-OEC members via supervised learning as presented in Benigni and Carley (2016); Benigni, Matthew et al.. For this work we present two detected communities as described below.

Case Study 1: The Euromaidan Twitter Community

The Euromaidan movement started as a series of protests in November 2013, where large numbers began to call for the removal of then President Viktor Yanukovych. These protests reached their peak in February 2015, ultimately leading to the removal of many of Ynukovych's senior officials. These events were soon followed by the Russian occupation of Crimea, and despite the installation of a new government, a substantial online activist community continues to oppose Russian influence in the Ukraine. We will refer to this community as the Euromaidan Twitter Community (ETC). Here, although strong negative sentiment toward the current Ukrainian government is observed, the online activism seen largely advocates change through legitimate government processes. Thus, while we acknowledge that little "extremism" exists in this community, we choose to examine this community due to its relevance to ongoing geopolitical events in the region. This community was extracted from a two step snowball sample of 8 known Euromaidan movement members' mention ties from March 2014 to September 2015. The search resulted in 92,295 Twitter accounts, and subsequent OEC detection returned 1,221 accounts actively supporting the movement. We have two collections from this community, one in March, 2016 and one in October, 2016.

Case Study 2: The Syrian Revolution Twitter Community

The Syrian Revolution Twitter Community is an updated set of users based on the ISIS-supporting OEC presented by Benigni et al.Benigni, Matthew et al.. By using mention activity of non-suspended, previously detected users and active learning, we update the SRTC based on the recent community activity. The instance presented in this work was collected in March of 2016 and contains 8718 members. We define a member as a Twitter user who positively affirms the leadership, ideology, fighters, or call to Jihad of any of the known Jihadist groups engaged in ongoing operations in Northern Iraq and Syria. The majority of tweeters voice support for ISIS or Jabhat al-Nusra though nearly all other anti-Assad factions are present.

Both the ETC and SRTC present a large community of Twitter users who's collective activity is of interest. In each case, group substructure, current interests, and information operations are all of interest, yet methods for mining such information remain immature.

5.3 Ideological User Clustering with Bipartite Spectral Graph Partitioning

Algorithms designed to find clusters of highly connected users within real world social networks is often referred to as community detection and has been studied extensively Fortunato (2010). In fact, Papadopoulos et. al. address the topic specifically with respect to social media Papadopoulos et al. (2012). Though many relevant methods exist to identify highly connected sets of users, it has been shown that users' simultaneous membership in multiple social groups often makes community detection methods based exclusively on social ties imprecise Benigni, Matthew et al.; Tang et al. (2009). We and others DeMasi et al. (2016) have found that identifying user clusters based on shared beliefs can be a useful alternative to traditional methods when searching for ideologically homogeneous user groups. Within detected OECs, hash tag use often provides user clusters consistent with ideological substructure within the community. One could view our method as a means to efficiently find community structure based on hash-tag-inferred social ties. For example, within a large Sunni extremist community like the SRTC, one can identify distinct clusters of support for Jabhat Al-Nusra, the Free Syrian Army, and many other competing groups. Distinct regional interests are observed as well. We have found distinct news sharing communities focused on operations in Syria, Northern Iraq, Yemen and Palestine. To identify these user groups we apply bipartite spectral graph partitioning to co-cluster Twitter users and hash tags. Although we have applied these methods exclusively to investigate large online extremist communities, it is possible they could be used for other online settings such as targeted advertising.

Often in network science bipartite graphs can be transformed into a one mode projection by multiplying the bipartite adjacency matrix by its transpose Zweig and Kaufmann (2011). However, in the case of online social networks and bipartite graphs of users and hash tags specifically, we have found these projections often become dense. In the case of users and hashtags, sets of approximately 100 thousand users often generate millions of unique hashtags. A one mode projection can result in a large, dense adjacency matrix which proves costly with respect to memory. For example we trimmed the user by hash tag bipartite graph extracted from the Euromaidan Twitter Community search and retained only hash tags used by five or more unique users. The resultant graph consisted of 92,295 users, 352,120 unique hash tags, and just over 16 million edges requiring 219 Mb of memory in sparse matrix format. The one mode projection requires over 60G of memory. We often find this type of increase in terms of edges with this type of matrix making one mode projections somewhat inconvenient. Furthermore a great number of these edges are quite close to zero and of little value to our task of clustering users. On option would be to simply retain the n largets edges, however ”duality” between users and hash tags exists. Hash tags are used to expose content to specific groups, and groups generate their own hash tags. Dhillon et al. present bipartite spectral graph partitioning as a means to co-cluster words and documents Dhillon (2001a) and argue the method obtains more interpretable clusters than one mode projections because of the “duality of word and document clustering”. In other words, they assert that word clustering induces document clustering and document clustering induces word clustering. We assert the same duality holds true for users and hash tags. To co-cluster words and documents, the authors generate a word-document matrix, and use left and right singular vectors to project words and documents into the same euclidian space. They subsequently use k-means MacQueen to find relevant clusters of documents and words. In our case, we claim duality in user and hash tag clustering. Such a method is notably similar to the most recent advancements in word embedding approaches, which focus on matrix decomposition of term matrices rather than focusing on developing neural models for embedding Hamilton et al. (2016); Pennington et al. (2014). In our model, communities influence hash tag popularity, and hash tag popularity helps organize user communities.

5.3.1 Bipartite Spectral Multi-Partitioning

To explain Dhillon’s Bipartite Spectral Multi-Partitioning algorithm we introduce the following notation. Lower case letters will represent column vectors. Capitol letters will denote adjacency matrices. We construct the graph $A_{m\times n}$ where an edge (or matrix cell) $e_{i,j}$ represents the number of times user i tweeted hash tag j where $i \in 1, 2, ..., m$ and $j \in 1, 2, ..., n$. We can represent this bipartite graph as a square undirected graph as follows:

$$M = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix} \tag{5.1}$$

The authors begin by explaining a spectral bipartition algorithm based on their proof illustrating the second eigenvector of the generalized eigenvalue problem $Lz = \lambda Dz$ provides a relaxation to the minimum normalized cut problem. Where L is the graph Laplacian defined as the nxn symmetric matrix, with one row and column for each vertex, such that:

$$L_{i,j} = \begin{cases} \sum_k E_{ik},\ i = j \\ -E_{ij},\ i \neq j \text{ and there is an edge } \{i, j\} \\ 0,\ otherwise \end{cases} \tag{5.2}$$

Furthermore, $L = D - M$ where D is the diagonal "degree" matrix of adjacency matrix M. This allows us to express L as follows:

$$L = \begin{bmatrix} D_1 & A \\ A^T & D2 \end{bmatrix} \tag{5.3}$$

We can then express the second eigen vector z_2 of L in terms of the second eigen vectors u_2 and $v_2$of the left and right matrices of the singular vector decomposition of A as follows:

$$z_2 = \begin{bmatrix} D_1^{1/2} u_2 \\ D_2^{1/2} v_2 \end{bmatrix} \tag{5.4}$$

One can then approximate the optimal bipartition by assigning the elements of z_2 to bimodal values m_j $(j = 1, 2)$ based on the following minimization:

$$\sum_{j=1}^{2} \sum_{z_2(i) \in m_j} (z_2(i) - m_j)^2 \tag{5.5}$$

Which corresponds to the same objective function minimized by the k-means algorithm Lloyd (1982). The authors then present the following bipartitioning algorithm:

Bipartite Spectral Bipartitioning

1. Given A form $A_n = D_1^{1/2} A D_2^{1/2}$
2. Compute the second singular vectors of A_n, u_2 and v_2 and form the vector z_2 as in (4).
3. Run the k-means algorithm on the z_2 to obtain the bipartitioning.

The authors then generalize this to the multipartitioning case by using the $l = \lceil log(k) \rceil$ sigular vectors of of $A_n, u_2, ..., u_{l+1}$ and $v_2, ..., v_{l+1}$ to obtain a $k - wise$ partition. To do so, they form the matrix:

$$Z = \begin{bmatrix} D_1^{1/2} U \\ D_2^{1/2} V \end{bmatrix} \tag{5.6}$$

Where $U = [u_2, u_3, ..., u_l]$ and $V = [v_2, v3, ..., v_l]$. The $k - wise$ partition can be minimized by the following equation:

$$\sum_{j=1}^{2} \sum_{z_2(i) \in m_j} ||Z(i) - m_j||^2 \tag{5.7}$$

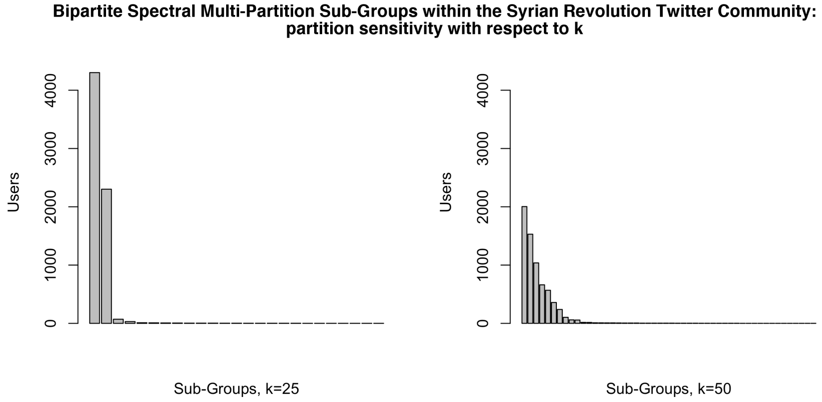


Figure 5.1: depicts the size of user groups, sorted from largest to smallest, from the SRTC data with $k = 25$ and $k = 50$ in the left and right panels respectively. The left highlights the algorithm's tendency to partition small sub-groups leaving one or two relatively large groups when k is not sufficiently large. The right panel corresponds to a selection of $k = 50$ which in this case provides interesting sub-structure.

Like equation (5), equation (7) can be minimized by classical k-means. The algorithm can be described as follows:

Bipartite Spectral Multi-partitioning

1. Given A form $A_n = D_1^{1/2} A D_2^{1/2}$
2. Compute the $l = \lceil log(k) \rceil$ singular vectors of $A_n, u_2, ..., u_{l+1}$ and $v_2, ..., v_{l+1}$ and concatenate them row wise to form Z.
3. Run the k-means algorithm on the $l - dimensional$ data Z to obtain the desired k-way multipartitioning.

5.3.2 Ideological User Clustering

We find that clustering the bipartite graph H of users and hash tags where an edge $e_{i,j}$ is defined as the number of times user i posts hashtag j within our corpus of tweets. To do so we cluster based on algorithm [REFERENCE].

To illustrate the utility of bipartite spectral partitioning we co-cluster users and hash tags within the SRTC. Due to the prevalence social bots within this community Benigni, Matthew et al.; Berger and Morgan (2015a), we set a threshold with respect to the minimum number of unique users who posted a specific hash tag. This enables us to remove hash tags with high frequency that are not necessarily indicative of a sub-group's interests. In this case we set the minimum unique user threshold $\gamma_u = 5$ roughly reducing the number of unique hash tags observed within the community by 80 %. We then co-cluster the 8718 users and 39,137 hashtags using k-means. In this case we choose $k = 50$ sub-groups, but acknowledge that selecting k requires some trial and error by the user. If k is too small, the algorithm returns few large groups; however, when k becomes 'large enough' more interesting sub-groups of users can be found. Figure 5.1 depicts the size of user groups, sorted from largest to smallest, from the SRTC data with $k = 25$ and $k = 50$ in the left and right panels respectively. The plot highlights the method's

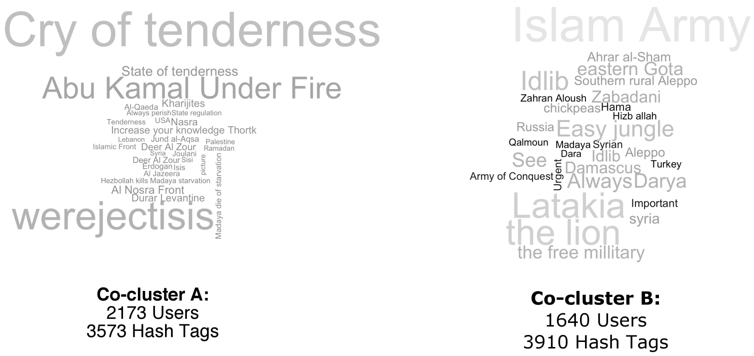


Figure 5.2: depicts the 30 most frequent hash tags co-clustered with two sub-groups within the SRTC. In each panel the hash tags are translated using Google Translate. Color depicts the relative frequency of the hash tag within the subgroup when compared to the rest of the community. Darker font connotes higher relative frequency within the subgroup when compared to the entire community.

propensity to first partition small groups of hash tags and users and highlights how nearly 75% of the users in our dataset are clustered into one of two groups when $k = 25$. Only when k is sufficiently large do we observe interesting clusters. In the case of the SRTC we start to identify clusters talking about distinct conflict zones in the middle east. For example we find distinct clusters discussing ongoing conflict in Syria, Iraq, Egypt and Yemen.

The hash tags co-clustered with the two largest user groups depicted in the right panel of Figure 5.1 are summarized in Figure 5.2. Co-cluster A (left panel) consists of 2173 users and 3573 hash tags, while co-cluster B (right panel) consists of 1640 users and 3910 hash tags. We calculate hash tag uniqueness within the cluster by comparing the relative frequency of each hashtag within the group to the hash tag's frequency within the entire community. Hash tags are colored by relative frequency in Figure 5.2 where darker font indicates higher within group frequency. For example *Army of Conquest* (translated from : مقـہ مك موسس معمف) has a much higher relative frequency within co-cluster B. The term refers to Jaish al-Fatah, an alliance of Islamist rebel faction active in Idlib and supported by Saudi Arabia and Turkey. The relative frequency and count of these terms within the sub-group indicate a common interest among users with respect to ongoing operations in Idlib and Jaish al-Fateh's role in them. Co-cluster A appears focused on denouncing ISIS which is highlighted by the hash tags *werejectisis* and *Abu Kamal Under Fire*. Abu Kamal is a Syrian town once held by ISIS that was highly targeted by US Coalition air strikes in February of 2016 Wood. Identification of such sub-structure can provide novel insight to inform strategic messaging or operational assessments. For example, manual inspection of users' Twitter timelines in co-cluster A highlighted an effective messaging theme employed by Jabhat al-Nusra that highlights ISIS' killing of Muslims. Mining these communities for effective propaganda themes could be used to inform strategic messaging, and be used to identify users with high social influence within specific topics of discussion.

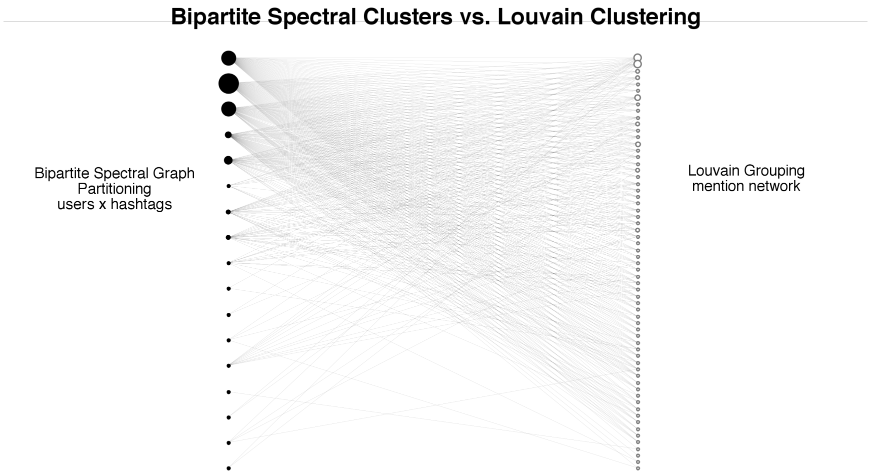


Figure 5.3: depicts the relationship between clusters generated with bipartite spectral partitioning and Louvain grouping using the SRTC dataset. The black nodes on the left hand side of the plot depict clusters derived by bipartite spectral partitioning, while the white nodes depicted on the right side of the plot depict clusters derived by Louvain grouping. Node size depicts cluster size. Edges depict the number of users shared between the two cluster types.

We highlight the distinct difference between bipartite spectral partitions and standard user grouping algorithms like Louvain grouping Blondel et al. (2008). We do not claim that bipartite spectral clustering produces better groups as ground truth in real world networks can rarely be inferred Peel et al. (2016). However, Figure 5.3 highlights the difference between clusters generated with bipartite spectral partitioning and Louvain grouping using the SRTC dataset. The black nodes on the right hand side of the plot depict clusters derived by bipartite spectral partitioning, while the white nodes depicted on the right side of the plot depict clusters derived by Louvain grouping. Node size depicts cluster size. Edges depict the number of users shared between the two cluster types. The figure highlights the differences between the two clustering methods. In practice we observe clusters that appear more homogeneous with respect to user content using bipartite spectral partitioning, but we acknowledge these observations are largely qualitative. Using both methods as an ensemble to find users who mention one another and user similar hash tags could prove interesting as well, though we will leave such questions for future research.

Although bipartite spectral multi-partitioning of users and hash tags offers unique insight into shared user activity, the method is not without limitations. The relative size of groups is highly sensitive to the researcher's selection of k, and an exploration of alternative clustering techniques is worthy of research Steinbach et al.. Specifically, k-means implementations which incorporate a priori knowledge could be useful to cluster more complex graph representations of large social networks Wagstaff et al.. It is also likely that additional information in user timelines could be used to cluster users. Keywords within tweets or urls for example could be incorporated for more informed clusters. Finally, large sets of hash tags are difficult to interpret. Word clouds like the ones depicted in Figure 5.2 do not necessarily imply sentiment. However, sentiment mining techniques could be applied to provide greater understanding, and as those methods become more mature with respect to non-English text we foresee them being highly useful.

5.4 Narrative Mining with Hash Tag Co-occurrence Graph Clustering

10-15 January	10-15 March	10-15 May
ukraine	freesavchenko	eurovision
russia	ukraine	ukraine
little	russia	**rukiprochotmirotvortsa**
freesavchenko	savchenko	**jamala**
russia	savchenko	freesavchenko
saveuatwi	donetsk	russia
ukraine	little	news
donetsk	russia	donetsk
ato	news	eurovision
ukraine	**syria**	**crimea**

Table 5.1: depicts the top 10 translated hash tags with respect to frequency in the Euromaidan Twitter Community from 10-15 January, 10-15 March, and 10-15 May 2017. Hashtags that occur only once are in bold. The table highlights the limitations of naive methods like frequencies to summarize community discussion.

In this section we would like to extract trending narratives from online communities in order to gain understanding of interests and topical connections. Again, we find this particularly informative in large extremist communities, but acknowledge it could be useful in other large online communities as well. Currently, many tools summarize social media content by using naive methods like frequency. However, in online communities participating in political activism frequency alone often leads to predictable results. Table 5.1 highlights the redundancy in trending hashtags over different time periods within the Euromaidan Twitter Community. Of the top 10 translated hashtags with respect to frequency, only 10% occur uniquely and are highlighted in bold font. We define a narrative as a subset of online discussion organized around an identifiable event or set of events within an online community. We use hash tag co-occurrence in tweets to identify clusters of terms which are often quite interpretable to an end user. To do so, we construct a temporally-constrained hash tag co-occurrence graph and use community detection to extract community narratives.

We are interested in characterizing community narratives within an arbitrarily selected time period T, and thus start by identifying the set of hash tags which appear more frequently within T. Twitter limits collection of a user's timeline to their last 3200 tweets Twitter (2016), therefore the number of active users on any given day can vary significantly. Many users' accounts go dormant as well. This forces us to normalize hash tag rates based on active users within our dataset. To do so, we construct a vector $\vec{u}_a$ of length T where element i of $\vec{u}_a$ is the number of active users within our dataset at time interval i. We define an active user collected tweets

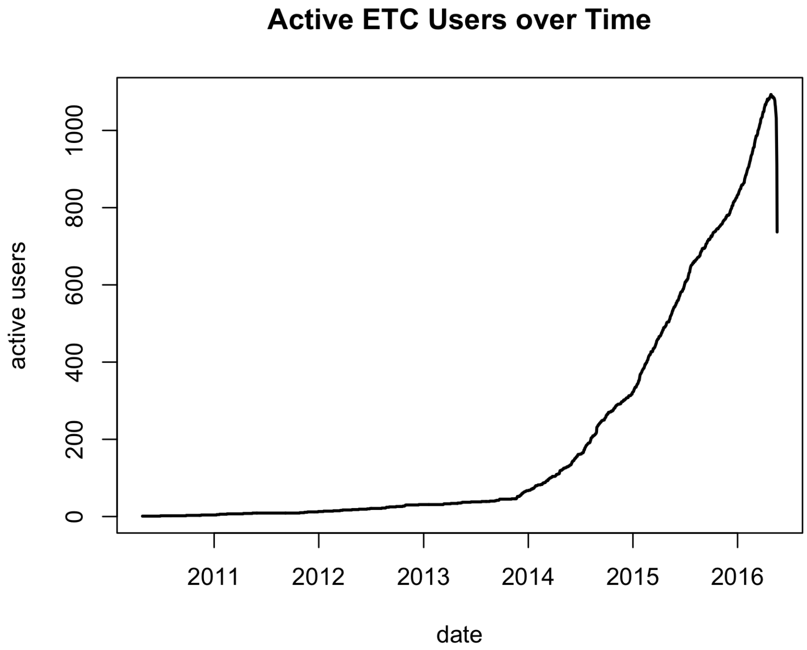


Figure 5.4: The amount of time spanned by a given user's last 3200 tweets varies greatly resulting in a non-uniform number of active user's tweets captured within our dataset. To evaluate trends we need to normalize by active users per day. The figure above depicts active users per day in the SRTC dataset.

span time i. Figure 5.4 depicts $\vec{u}_a$ for the ETC. We then normalize hashtag rates for a given time interval $T = [t_1, t_2]$ as follows

$$d_T = \sum_{i=t_1}^{t_2} \vec{u}_a$$

where d_T is the number of active user days associated with time interval T. We define d_g as the total active user days within our sample where

$$d_g = \sum \vec{u}_a$$

We then construct $\vec{h}_T$ and $\vec{h}_G$. Both are vectors of length n, where n is the number of unique hash tags collected within the community of users U. Each entry in $\vec{h}_T$ and $\vec{h}_G$ represents the normalized counts of each hashtag j divided by the d_T and d_G respectively. $\vec{\lambda}_T$, a vector of length n, represents the change in rate of hash tag j when compared to the global rate and is defined as follows:

$$\vec{\lambda}_T = \vec{h}_T \odot \vec{h}_G^{-1}$$

We then define our set of trending hash tags h as those having $\lambda_T > \phi$. In our case we set $\phi = 2$ or hash tags who's rate was twice as high as their global rate. It is worth mentioning that this parameter needs to be selected with careful consideration as T gets large. Furthermore, we only select hashtags with γ_u or more unique users posting them to ensure we are capturing community narratives and account for bots as discussed in Section 5.3.

We then construct the network H_T where an edge is defined as the number of times hashtag i and hashtag j co-occur within user tweets posted within time interval T. Finally we cluster H_T using the Louvain Grouping algorithm and extract the resultant clusters to identify narratives Blondel et al. (2008). It is worth mentioning that any graph clustering approach suitable to large, weighted, undirected graphs could be used for this step. For an extensive discussion of suitable alternatives we refer researchers to Fortunato et al. Fortunato (2010).

Figures 6.7 and 5.6 depict H_T for the Euromaidan Twitter Community from 10-15 May, 2016 and 9-16 October, 2016 respectively. In each plot the left panel depicts H_T where nodes are sized by $\vec{h}_T$ and colored based on their membership to large Louvain clusters. The right panel summarizes the top 25 terms with respect to λ_T within a specific cluster or narratives. Each word is sized by $\vec{h}_T$, and colored by λ_T where gray font indicates values close to one and increased rates are colored. The four narratives depicted in the right panel of Figure 6.7 center around major news story lines in Eastern Europe in early May, 2016. Narrative one, in red, centers around actions taken by the pro-Ukrainian hacker groups Falcons Flame and Trinity, who defaced the official websites of 9, Russian-backed militant groups involved in the Crimean Conflict Shamanska (2016). Narrative two, in blue, centers around the Ukrainian government's the appointment of Yuriy Lutsenko as Ukraine's attorney general UNIAN (2016). To appoint Lutsenko the government passed a special amendment removing the requirement for the country's attorney general possess a law degree, and was perceived as corruption within the ETC. Narratives two and three discuss the annual Lennart Meri conference in Tallinn Jensen, and Susana Alimivna

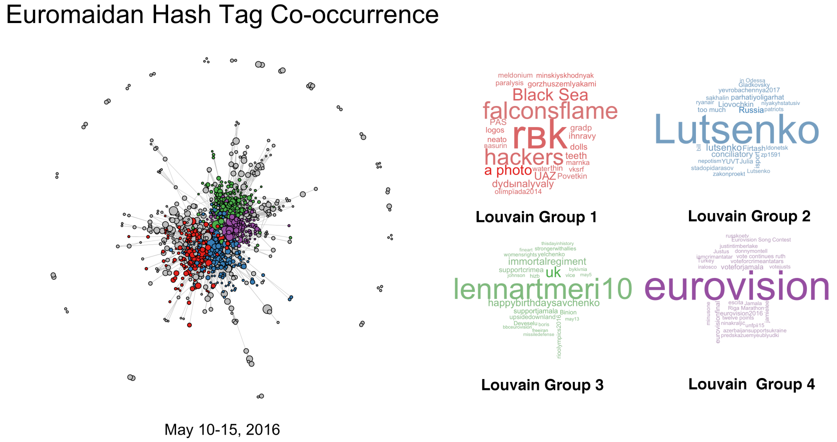


Figure 5.5: depicts community narrative extraction within the Euromaidan Twitter Community from 10-15 May, 2016.

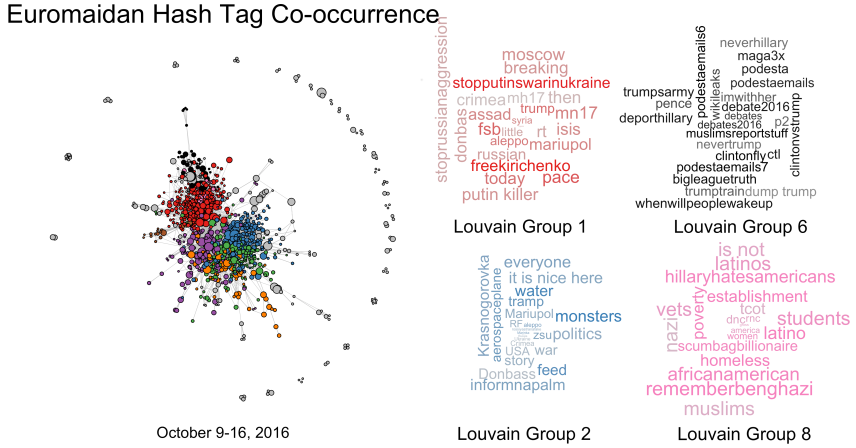


Figure 5.6: depicts community narrative extraction within the Euromaidan Twitter Community from 9-15 October, 2016.

Jamaladinova's Eurovision 2016 victory Roxburgh, Gordon respectively. We have shared similar figures with members of the United States Army Asymmetric Warfare Group with extensive operational advisory experience in Ukraine which they found helpful in interpreting community interests over time. If we think of the identified narratives as components within H_T, the off diagonal densities between narratives could prove interesting as well. For example the connection between narratives associated with the 2016 United States Presidential Election and Russian aggression can clearly be seen in Figure 5.6. Louvain Groups six and eight are densely connected to ongoing narrative of Russian actions in Syria and Ukraine highlighted in Louvain groups 1 and 2.

We assert that clustering and visualizing hash tag co-occurrence matrices offer researchers and analysts a quick means to distill community-level discussion in online social networks. The ability to mine quickly evolving hash tags within these communities offers insight into the complex interests driving activist discussion. Although these methods are limited to fixed time periods of interest, it is likely that they could be analyzed as dynamic networks and offer insights into changing interests over time Carley (2006). Additionally, these methods could be extended to account for urls and hash tags to quickly find the external sources most influential within specific topic areas. Similar to the points made in Section 5.3, incorporation of sentiment analysis could provide additional value as well.

5.5 Identifying Radicalization with Directed url Sharing Networks

The methods introduced in Sections 5.3 and 5.4 illustrate how large online communities can be mined for high level understanding of community interests. In this section we provide an example where we can mine communities for social media intelligence (SOCMINT). Detecting extremist communities at scale enables information extractions that highlight tactics and techniques used for the radicalization process. For example ISIS uses Twitter for broad, general recruiting and typically transitions to more secure messaging platforms as they identify individuals worth personally targeting for radicalization Berger and Morgan (2015a); Berger, JM; Callimachi (2015). This "direct messaging" is typically done on Twitter through the $@mention$, where $user_a$ can ensure his content is included on $user_b$'s timeline by including $@user_b$ in the body of his or her tweet.

These hypothesized recruitment behavior patterns can be extracted and investigated by constructing graphs based on specific types of tweets. The Twitter REST API Twitter (2016) provides structure to easily extract tweets containing urls and $@mentions$, and with this subset of content we form the following graphs:

P, a weighted, directed graph, where we define an edge $e_{i,j}$ as the number of tweets containing a url where $user_i$ mentions $user_j$.

U_p, a weighted, bipartite graph, where we define an edge $e_{i,j}$ as the number of tweets posted by $user_i$ containing url_j. We hypothesis that propagandists and recruiters would be a subset of the users within U_p.

R_u, a weighted, bipartite graph, where we define an edge $e_{i,j}$ as the number of tweets posted containing url_i and mentioning $user_j$. We hypothesize that recruits would be a

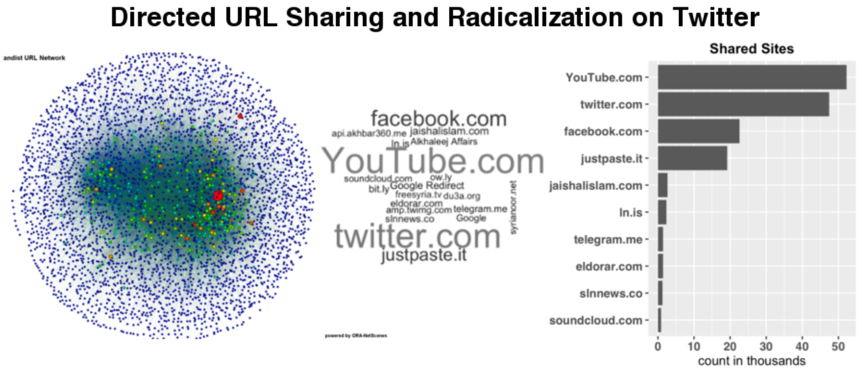


Figure 5.7: The left panel depicts directed URL sharing network P within the SRTC. Nodes are both colored and sized by out degree. The middle panel depicts the top 25 shortened URLs posted in U_P with respect to frequency as does the right panel.

subset of the users within R_u.

A simple analysis of node centrality of users within P provides inference into possible roles. Users who send many messages with $@mentions$ and urls are potential propagandists or recruiters and would have relatively high out-degree Wasserman and Faust (1994) within P. Naturally, users with high in-degree would be potential recruiting targets. Identifying both role types offers insight into more nuances analysis to recruiting techniques, materials, and could potentially help identify the user behaviors used to identify potential recruits. Both U_P and R_u can be used to further infer user roles. For example, if the urls shared by a user in U_p often contain links to peer-to-peer messaging services like Telegram or WhatsApp, they would possibly be recruiters. If they share inflammatory news sources and videos they could be propagandists. Furthermore, the types of links received by users in R_U could offer similar insight to identify potential recruits.

Figure 5.7 depicts the the relationship between recruiters and their recruiting targets as well as the sites they reference in the radicalization process. To develop the plot we extract the 184,225 tweets where SRTC members use the $@mention$ to share URL content between May, 2015 and May, 2016. The left panel depicts the directed mention network, where nodes are SRTC members and edges depict the number of tweets posted by user a that contain both an url and mention of user b. Both the color and size of nodes denote the number of messages sent. Thus, small blue nodes would be likely recruiting targets, and large red nodes would likely be recruiters or propagandists. The center panel summarizes the top 35 sites shared within messages of this type, as does the bar plot in the right panel. For privacy reasons we have chosen not to publish identified recruiters in this forum, but we find dyads within P with relatively small follower counts and edge weights between 20 and 100 to often identify user to user exchanges consistent with targeted recruiting. These online dialogues often facilitate transition to more secure platforms over time as well. Conveniently, as these recruiters move discussion to peer to peer messaging platforms, they typically provide their user identification number. For example, this analysis yielded over 200 unique telegram.me accounts of likely recruiters.

The methods described above to identify recruitment and propaganda dissemination within online communities unfortunately still require a significant amount of manual exploration in order to confidently extract users of specific role types. Moreover, the example provided above merely highlights the ability to develop useful network representations based on hypothesized behaviors. Although these techniques still require a significant level of manual inspection, it is likely that more detailed NLP analysis of users' tweets could further inform automated detection methods, and active learning would provide the technical framework needed to gain adequate performance in an efficient manner Settles (2009). Furthermore, we hypothesize that similar strategies could be used to identify other user types with identifiable communication patterns.

5.6 Conclusion

In this paper we have provided researchers and practitioners three novel applications of existing methods in network science in order to facilitate improved data analysis of large communities in online social networks. We introduced:

- ideologically clustering users by hashtag behavior with bipartite spectral graph partitioning
- narrative extraction through hash tag co-occurence graph clustering
- user role inference from directed url sharing networks

In each instance, we presented the method in detail and illustrated meaningful insights from two activist online communities. We also offered useful extensions to these methods for future work. As stated in Section 5.3, we cannot claim the performance of bipartite spectral partitioning is better than that of other clustering algorithms. However, we do show that clustering users by hash tag use provides a different perspective. We see great potential in developing ensemble methods to cluster users across multiple dimensions such as following ties, mention ties, and shared hash tag use. Hash tag co-clustering also has limitations. Selection of time interval greatly influences the relative frequency of hash tags over time, and understanding the correct T to extract narratives needs further research. It is likely that some temporal decay function on link weights could be used or other methods from dynamic network analysis. Moreover, we highlight that the methods presented in this work almost exclusively mine patterns based on graph structure. This affords these methods to perform independent of the language use of community members, and it is likely that they would complement NLP-based methods to mine intelligence from OECs. Our hope is that this work enables both researchers and practitioners to draw novel insights from large online communities.

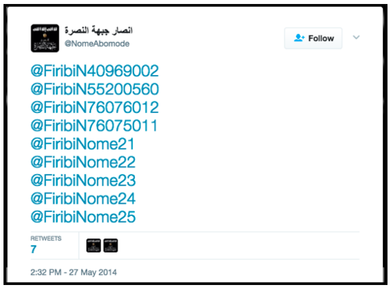
Chapter 6: The Spread of Fake Online Credibility: Detecting Socialbot-Network-Promote Agendas and Users in Ukraine, the Middle East, and the United States

6.1 Introduction

Social influence occurs when a person's emotions, opinions, or behaviors are affected by others Kelman (1958), and the emergence of Online Social Networks (OSNs) as publication and news delivery platforms Chu et al. (2010) has provided a powerful marketing venue accessible to anyone. In fact, as of June, 2016, Facebook and Twitter claim to have nearly 2 Billion active users combined Statista (2016a,b). These large OSNs have become powerful venues to shape emotions, opinions, and behaviors, and it comes as no surprise that we now observed sophisticated technical means used to manipulate them. A common tool used to manipulate perception within OSNs is the automated social actor, or "bot" Ruths and Pfeffer (2014). Initially used to spread malware Zhang et al. (2013), a substantial amount of literature now documents the use of bots to influence geopolotics by creating artificial online personas and content creating an illusion of grass-roots support to a political agenda Ratkiewicz et al. (a,b); Woolley (2016). Socialbots have also been used to spread fake news and manipulate large scale financial disruptions. In April, 2013 hack of the Associated Press Twitter account that spread a fake news story of a terror attack at the White House that resulted in a 147 point drop in the Dow Jones stock index Ferrara (2015).

As understanding of these promotion methods improves, an increasing number of studies describe socialbots deployed as networks. Socialbot Networks (SbNs) have been used in political revolutions like the Arab Spring Abokhodair et al. (2015), and by terrorist groups like the Islamic State in Iraq and Syria (ISIS) Al-khateeb and Agarwal (2015); Berger and Morgan (2015a) and Jabhat al-Nusra [citation removed for blind review]. SbNs have also been shown to be both pervasive and highly active in online conversation within the 2016 United States Presidential Election Bessi and Ferrara (2016). Unmitigated, SbNs have the ability to undermine the very foundation of our information society.

In this paper we introduce two specific classes of SbN used for both promotion of users



انصار جبهة النصرة
@NomeAbomode

@FiribiN40969002
@FiribiN55200560
@FiribiN76076012
@FiribiN76075011
@FiribiNome21
@FiribiNome22
@FiribiNome23
@FiribiNome24
@FiribiNome25

RETWEETS
7

2:32 PM - 27 May 2014

Figure 6.1: Depicts core bot behavior in the Firibinome Mention-core Socialbot Network.

and content, as well as influencing political opinion through propaganda dissemination. Our detection method and analysis originates from the field of community detection which looks for sub-groups of highly connected users within a social space. While studying large online communities of political activism, we have begun to frequently observe two specific types of botnet structure: Mention Community Socialbot Networks (MCSbNs) and Cyborg Socialbot Networks (CSbNs); we know of no comprehensive study of either. MCSbNs are SbNs whose socialbots construct large, mention networks designed to promote specific users, groups, or narratives. CSbNs have socialbot accounts consisting of real users who authorize socialbot access to post from their accounts. These two classes of botnet are not mutually exclusive, and we will provide an example of a botnet meeting the criteria for both.

Accounts within these SbNs exhibit anomalous mention behavior in that they post strings of @mentions, as depicted in Figure 6.1. These posts often can be associated with the #FollowFriday movement in which users share lists of other users to recommend following ties; however, we observe anomalously dense reciprocity in some cases forming subcommunities with clear promotional objectives. Our increased observation MCSbNs and CSbNs for marketing or geopolitical influence motivates this work. Our goal is to provide an understanding of these emergent structures in hopes of shaping future research into the manipulation of online opinion. We define each class of SbN and illustrate how they attempt to promote users, influence online discussion, and bridge seperate online communities by presenting several novel case studies from Twitter. Additionally, we present a dense-subgraph-based MCSbN detection strategy that will likely augment many existing instance-based detection methods. Our case studies span online communities focused on Middle Eastern Affairs, the Ukrainian Euromaidan Movement, and the 2016 United States Presidential Election. The summary of our case studies is provided in Table 6.1. We also discuss the the limitations of our findings and propose future research.

6.2 Related Work

Socialbots are Automated Social Actors (ASA), or software designed to replicate human activity in an online social space Abokhodair et al. (2015). When social bots are deployed as a network they are referred to as socialbot networks (SbN) or social botnets and have been used for a variety of ends. Boshmaf, Muslukhov, Beznosov, and Ripeanu provide extensive analysis of SbNs and define an SbN as a set of socialbots with three components: the botherder who controls the socialbots, the socialbots that carry out tasks assigned by the botherder, and a Command and Control (C&C) channel used to facilitate task assignment Boshmaf et al. (2011, 2013). Although

SbN Name, Size	Data	MCSbN / CSbN
#InfluenceMarketer • 93 core bots • 1.26M tweets • 9.44 mentions/tweet	ALT16	yes / yes
$@Du3a\_org$ • over 125K users • 12M tweets	ISIS14	no / yes
Firibinome • 72 core bots • 12,254 tweets • 5.54 mentions/tweet	ISIS14	yes / no
Euromaidan Images • 8 core bots • 385,715 tweets • 4.53 mentions/tweet	EUR15	yes / no

Table 6.1: describes the four case studies included in our manuscript and their inclusion in the two botnet classes presented therein: Social Cyborg Socialbot, and Mention Community Socialbot Networks.

Boshmaf et al. primarily present SbNs as a means to spread maleware and harvest user data, a growing body of literature details SbN promotion of political agendas. Abokhodair, Yoo, and McDonald (2015) describe the 35-week lifespan of a SbN on Twitter designed to express opinion, testimony of ongoing events, and engage in preliminary conversation associated with the Syrian Revolution Abokhodair et al. (2015). Similar techniques have been observed in the case of ISIS' online video dissemination as well Al-khateeb and Agarwal (2015)[citation removed for blind review]. In some cases partially automated social actors or "cyborgs" have been observed. Chu et al. (2010) discusses the distinction between humans, bots, and *cyborgs* on Twitter Chu et al. (2010) where they define cyborgs as either socialbot-assisted humans or human-assisted socialbots. They further discuss the advantages and hazards of Twitter's facilitation of third party applications through its API. Our work builds upon the aforementioned literature in that we draw distinctions in role with respect to socialbots within the SbNs used for various forms of promotion. Furthermore we introduce the first instances we know of a SbN that employs cyborg users to carry out bot tasks.

6.3 Data

Each SbN presented within our manuscript is drawn from a similarly collected dataset. In each case these datasets were collected originally for the study of online political activism, but in each case we have frequently observed MCSbN and CSbNs. Each dataset has been collected using snowball sampling Goodman (1961), a non-random sampling technique where a set of individuals is chosen as “seed agents”, and the k most frequent accounts followed by each seed agent are taken as members of the sample. This technique can be iterated in steps, as we have done in our searches. Although this technique is not random and prone to bias, it is often used when trying to sample hidden populations [citation removed for blind review]. We seed searches with sets of users that are known members of a larger community of interest we would like to detect. We then collect additional users based on seed seed users’ social ties. In some cases we iterate this process in “hops”, such as in the ISIS dataset explained in the following section.

6.3.1 ISIS Twitter Search (ISIS14)

In November of 2014 we seeded a two-hop snowball sample of influential ISIS propagandists’Carter et al. (2014) following ties to test methods for online extremist community detection. The search resulted in 119,156 user account profiles and roughly 186 million tweets. Within this dataset we have detected two MCSbNs and nine CSbNs.

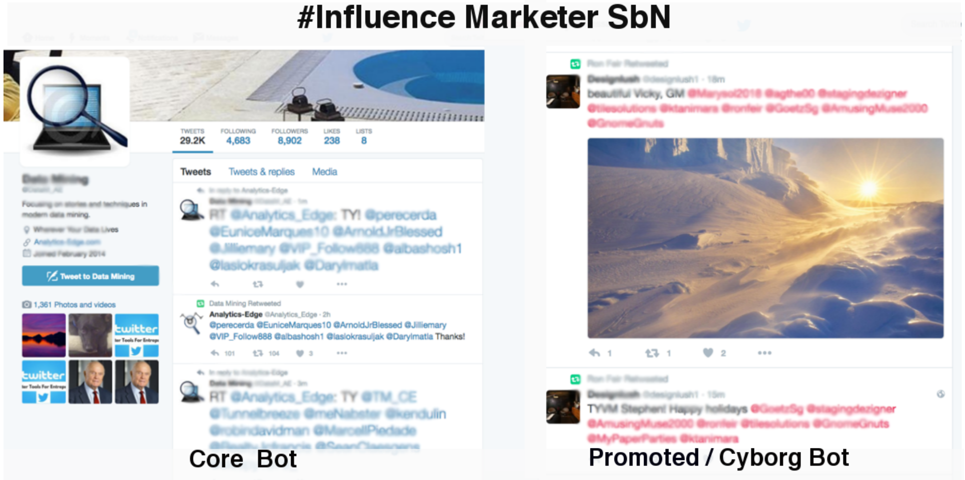
6.3.2 Euromaidan Twitter Search (EUR15)

The Euromaidan Revolution occurred as a wave of demonstrations starting in Ukraine in November 2013 and resulted in the removal of Ukrainian President Viktor Yanukovych from power. In an attempt to study messaging themes used within the Euromaidan movement we conducted a two-hop snowball sample of 8 known members’ mention ties from March 2014 to September 2015. The search resulted in 92,295 Twitter users and 215 million tweets. Within this dataset we have detected two MCSbNs.

6.3.3 Alt-Right Twitter Search (ALT16)

In October of 2016 we seeded a one-hop snowball sample of 2482 users who each followed eight influential Twitter users associated with the Alt-Right political movement. The search resulted in 106K users and collected 272 million tweets. Within this dataset we have detected five MCSbNs and two CSbNs.

For each data set we construct two graphs. First we define $M(V, E)$, a weighted directed graph, with vertices $V : \{v_1, ..., v_n\}$ consisting the users returned form our search. Edges $E_m : \{e_{M,1}, ..., e_{M,m}\}$ are defined as the number of unique tweets where $user_i$ mentions $user_j$. We then define our reciprocal mention network $R(V, E)$ with the same vertices $V : \{v_1, ..., v_n\}$. However, we remove directionality by defining our edges to connote reciprocity as $e_R(j, i) = argmin(e_M(i, j), e_M(j, i))$. We will refer to both graphs M and R throughout Sections 6.4 and 6.5. When referring to examples within our case studies, we will introduce subscript referring to each graph’s respective dataset. It is important to mention that many of the users captured with this sampling technique will not share beliefs or opinions similar to seed accounts and typically community detection methods are used to identify sub-groups of similar opinion [citation removed for blind review].



Data Mining Retweeted

Analytics-Edge @Analytics_Edge · 2h

@perecerda @EuniceMarques10 @ArnoldJrBlessed @Jilliemary @VIP_Follow888 @albashosh1 @lasiokrasuljak @Darylmatia Thanks!

101 104 3

1 1 2

Figure 6.2: examples of a *core bot* (left panel) and promoted cyborg bot (right panel) within the #InfluenceMarketer MCSbN. The mention behaviors exhibited by both are consistent with this core bot behavior of frequently posting strings of mentions. However promoted bots more often post retweets of other promoted bots as well as human-like content. In both cases lists of mentions are posted by the botherder to promote users or content.

6.4 Methods

In this section we will introduce two special classes of SbN: Mention Community Socialbot Networks (MCSbNs), and Cyborg Socialbot Networks (CSbNs). In each case we will define the SbN, discuss the potential promotional objectives of each; we will also introduce a graph-based detection strategy for MCSbNs. Examples of each class will be provided in Section 6.5 including one instance that is both a MCSbN and CSbN.

6.4.1 Mention Community Socialbot Networks (MCSbNs)

MCSbNs are a special case of SbN where a subset of the socialbots are used to create a dense community of users who mention one another with high volume and reciprocity. We believe this class of SbN promotes users and/or narratives by inflating longstanding social network measures of importance. Like other SbNs, they consist of socialbots, a botherder, and have a C&C channel; however, MCSbN have two types of socialbot:

- **Core Bots:** bot accounts make little to no effort to appear as human users, but instead use the content of their tweets to form a dense user community with lists of $@mentions$.
- **Promoted Bots:** accounts mentioned by core bots that attempt to influence a specific online community of interest. These accounts can be bots or cyborgs.

Core bots predominantly mention and retweet each other creating a dense communities with high reciprocity ($user_a$ mentions $user_b$, and $user_b$ mentions $user_a$. However, the core bots also mention highly followed accounts within a specific community of interest, as well as *promoted bots*. An example of core bot behavior is depicted in the left panel of Figure 6.2, and the mentioning behaviors are visualized in Figure 6.4. In some cases promoted bots can also add edges to the MCSbN by retweeting other promoted bots posts, as depicted in the right panel of Figure 6.2 as is done in user promotion MCSbNs that appear to be CSbNs as well.

Figure 6.4 depicts the relationship between accounts mentioned by core bots within the #InfluenceMarketer SbN (x-axis) and mentions by non-core bot accounts within the Alt-Right OCT16 dataset. As will be explained in Section 6.5, the #InfluenceMarketer SbN is both a MCSbN and a CSbN. Points depicted with squares are core bot accounts, and points outlined in black are assumed to be SbN members based on their shared posting method via a specific third party application. Points are sized and filled based on follower count. The plot highlights the dense mention behaviors exhibited by core bots and the relationship between core bot mentions and non-core bot mentions. This same structure is seen in each of the MCSbNs discussed in Section 6.5.

Although we do not have access to the algorithms used by Twitter to prioritize content within users' feeds, we believe the behavior exhibited by MCSbNs promotes individuals and narratives by artificially inflating node, edge, and graph level centrality measures. A user's influence would likely be measured based on node level metrics like degree centrality Wasserman and Faust (1994), eigenvector centrality Bonacich (2007), or Page Rank Brin and Page (2012); each of which would be artificially inflated by core bot behavior. In addition to inflating node-level metrics, the quantity and types of edges formed by core bots creates misleading graph structures as well. The research area of community detection searches for subsets of users that have a greater density of ties within the group then when compared to the rest of the social graph, and reciprocity is often used as a metric to indicate trust among users within a social network. Both concepts have been used to understand the diffusion of news media within Twitter Kwak et al. (2010) and find "centroids of discussion" Bessi and Ferrara (2016). Networked core bot behavior creates fairly large, dense, communities of artificial users with high reciprocity, and it appears that this core structure can be used to attract real users as well.

In many cases the density of core bots within an MCSbN's mention network is also represented in follower and following ties, and it is possible that MCSbN behavior promotes following ties as well. Triadic closure has been shown to be a strong predictor of homophily in social networks and refers to the concept of friendship being likely between individuals with common social ties Kossinets and Watts (2006). If Twitter uses graph structure based concepts like triadic closure to recommend followers, their recommendations would be influenced by MCSbN behavior as well. In fact, we often see common followers across multiple core bot accounts. This could also explain why each MCSbN within presented shows a similar pattern of mentioning highly followed accounts within the MCSbN's community of interest. This can be seen by the highly followed and mentioned accounts depicted in Figure 6.4. Figure 6.3 depicts summary of the most dense, distinct subgraphs consisting of between 30 and 200 users in the Alt-Right Twitter Search. The $x - axis$ connotes subgraph density, while the $y - axis$ connotes size in terms of users. Black circles indicate obvious MCSbNs where more than 50% of users whose mention per tweet ratio was greater than 4.34, the 99.5 percentile of all 287K users collected. However, many users within each of the 20 most dense subgraphs use lists of mentions when posting.

Because MCSbNs artificially create highly dense communities, dense subgraph detection offers a logical means to detect them. Dense subgraph detection can be preferable when a complete clustering of the data is not desired. We are not interested in assigning every user to a cluster, but simply interested in highly connected communities exhibiting with multiple users exhibiting core bot behavior. We have detected 10 overt MCSbNs using dense subgraph detection as presented in Chen and Saad (2012), with one caveat. Due to the excessively high reciprocity among

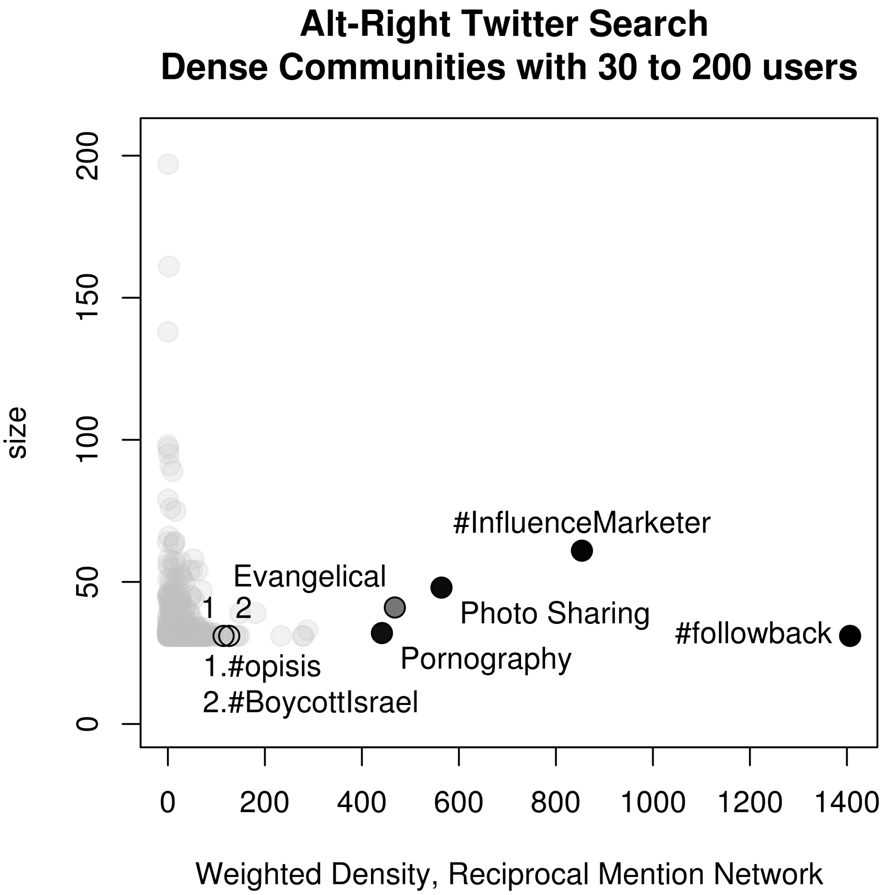


Figure 6.3: depicts summary of the most dense, distinct subgraphs consisting of between 30 and 200 users within the Alt-Right Twitter Search. The $x - axis$ connotes subgraph density, while the $y - axis$ connotes size in terms of users. Black circles obvious MCSbNs, but many of the subgraphs showing high graph density exhibit core-bot-like behavior and could represent more sophisticated MCSbNs.

MCSbN core bots, we define subgraph density of our undirected graph R with n vertices as:

$$d_R = \frac{\sum_{i=1}^{n} \sum_{j>i}^{n} e_R(i,j)}{n(n-1)/2} \tag{6.1}$$

Just as presented in Chen and Saad (2012) we search for MCSbNs by constructing A_R, a weighted adjacency matrix of R and define C_R, the cosine similarity matrix, as:

$$C_R(i,j) = \frac{\langle A_R(:,i), A_R(:,j)\rangle}{||A(:,i)||||A(:,j)||} \tag{6.2}$$

We then set $t = 2 \times |E_R|$ and sort the largest t non zero entries of C_R in ascending order. We denote this sorted array as Q, and construct a hierarchy T based on its sorted vertex pairs. Finally we extract the largest distinct subgraphs from T whose size falls within the interval (s_{min}, s_{max}) and meets a minimum density threshold, d_{min}. We manually inspect this set of subgraphs, but off that other node level bot detection methods could be used to further filter subgraphs. We will discuss this research topic further in Section 6.5. For a detailed discussion of the algorithm we refer readers to Chen and Saad (2012), but we provide a summary of our approach in Algorithm 2.

```
Algorithm 2: Find core botnet members of an FMSbN
Input  : Given a large sparse, weighted, reciprocal mention graph $R$, and density threshold $d_{min}$
Output: Set $D : d_1(v,e),...,d_n(v,e)$ such that each subgraph contains a subset of users exhibiting
        behavior that meets the definition of a FMSbN core bot member.
Compute Matrix $C_R$ as defined in (2)
Sort the largest $t$ non-zero entries of $C_R$ in ascending order, where $t = nz(A)$. Denote $Q$ the
sorted array.
Construct the hierarchy $T$ according to the sorted vertex pairs designated by $Q$.
Extract Subgraphs of where $d_G \geq d_{min}$
Manually inspect subgraphs for FMSbN like behavior
```

Once we have identified a subgraph exhibiting core bot behavior, V_t, we then construct a one-hop mention snowball sample of users within V_t to form graph R_t, a weighted, undirected mention network. V_t and R_t serve as initial conditions for an iterative process where we define V_{t+1} as the vertices described by the largest $1 - \alpha$ percentile of edges in M_t by weight. We then iterate this algorithm until $V_t = V_{t+1)}$ to find additional core bots. A summary of our algorithm is provided in Algorithm 3.

```
Algorithm 3: Find core and peripheral FMSbN accounts
Input  : Given $V_t$ a set of core bot FMSbN accounts and threshold $\alpha$
Output: A set of core and peripheral FMSbN accounts.
1 while $V_t \neq V_{t+1}$ do
2     build graph $R_t$, a 1-hop snowball sample of the $V_t$ mention network.
3     define $V_{t+1}$ by the largest $\alpha$ edges in $M_t$
4     if $V_t = V_{t+1}$ then
5         return $V_t$
6     else
7         $V_t \leftarrow V_{t+1}$;
8         return to line 2;
```

As stated earlier, we have used this process to detect MCSbNs core bots in a variety of datasets, and have manually verified 5, 2, and 3 distinct MCSbNs in the Alt-Right, ISIS NOV14, and Euromaidan Twitter Searches respectively. We present 4 cases in Section 6.5.

Conclusively detecting promoted bots is more challenging, particularly in MCSbNs that are also CSbNs. However, dense subgraph detection of core bots could provide all that is necessary for MCSbN mitigation as removal of core bots would most likely remove MCSbN effects. However, we see promise in combining graph level information like dense subgraphs, with node or user level features. Most bot detection methods currently classify based on characteristics at the node or instance level. Binary classification on annotated graphs offers a strong framework to combine the two methods and will be discussed in greater detail in Section 6.6

Rank	Top ALT16 Users (co-mention degree)
1	@nayami_rescue
2	@Socialfave
3	@sundoghigh
4	@TheMisterFavor
5	@SarCatStyX
6	@saravastiares
7	@KoichicCheryl
8	@realDonaldTrump
9	@Easy_Branches
10	@lupash7

Table 6.2: Depicts top users in the ALT16 dataset co-mention graph with respect to weighted degree centrality.

6.4.2 Cyborg Socialbot Networks (CSbNs)

Like MCSbNs, Cyborg Socialbot Networks (CSbNs) are special class of SbN where the socialbots are bot-assisted humans as defined in Chu et al. (2010). Twitter's API offers botherders the ability to provide efficient C&C to their socialbot accounts, and in the case of CSbNs real users become part of the social botnet by granting the botherder permission to perform tasks on their behalf. This turns the user's account into a "cyborg account." Many Twitter users authorize third party applications to automate a variety of actions from their account to include posting, following accounts, or advertising. Tweets posted from these third party applications can sometimes be identified by the "source" field provided within the tweet class Developers (2017), which has been identified as a strong feature for bot prediction Lin and Huang (2013). In the case of the CSbNs, it appears that users authorize the CSbN to perform tasks on their behalf, and the botherder can then use cyborg accounts as socialbots within his/her SbN. In some cases the cyborg users appear to explicitly authorize the applications service, but in other cases they may not realize what permissions they have granted the botherder. In either case, these CSbNs with large user populations have a significant ability to manipulate online following networks or promote content. We will present two CSbNs in detail in Section 6.5.

6.5 Results

6.5.1 MCSbN and CSbN User Promotion

The SbNs in this work appear to be designed to counfoud measures of influence in the Twitter mention and comention graphs. As it is well known that bots can influence simple network metrics like degree centrality, we are not surprised to find top accounts depicted in Table 6.2 to exhibit behavior similar to the core bot behavior depicted in Figure 6.1. We then turn to more complex network metrics that are better able to withstand spam and bot-like behavior. Specifically, we PageRank **?** and coreness Liu et al. (2014), two common, more complex metrics for measuring influence in complex networks have been shown to be more robust to bot activity. Both metrics are drawn from the family of *radial-volume centralities* Bonacich (1987), which

Rank	PageRank	Coreness
1	@realDonaldTrump	@2020sahara
2	@HillaryClinton	@JuliaZek
3	@YouTube	@Jerz_Gal
4	@POTUS	@DilrubaLees
5	@FoxNews	@7artistai
6	@CNN	@nayami_rescue
7	@nytimes	@wanderingstarz1
8	@timkaine	@JulezPooh
9	@NASA	@Dollhouse
10	@wikileaks	@Edward733

Table 6.3: Depicts top users in the ALT16 dataset co-mention graph with respect to PageRank (left column) and Coreness.

attempt to quantify an account (or more generally, a network vertex) x's influence based on the influence of the other vertices to which x is linked. These measures can be viewed on a continuum based on the length of walk considered in the neighborhood of a given vertex. Coreness Kitsak et al. (2010) is calculated based on the concept of *K-shells* within a graph. A K-shell is defined as the maximal subgraph of a given graph G, where all vertices are of degree greater than or equal to k. A vertex's coreness, K_s, indicates the greatest value k for with the node remains in the corresponding k-shell. In addition to coreness, we also look at PageRank, or eigenvector centrality, which is roughly defined as "accounts who are popular with other accounts who are popular" Kwak et al. (2010). Each of these metrics can be confounded by the SbNs we define in this work. For example top accounts with respect to coreness depicted in Table 6.3 again show high levels of core bot behavior.

We identified the #InfluenceMarketer SbN while studying the 2016 United States Presidential Election using the methodology presented in Section 6.4 with $\alpha = .05$, but the #followback MCSbN depicted in Figure 6.3 is quite similar. Two overt MFSbNs of similar type and scale were detected in the EUR15 dataset as well. The #InfluenceMarketer SbN is both a CSbN and a MCSbN. Figure 6.4 depicts the degree with which the #InfluenceMarketer SbN confounds coreness in the mention and comention graphs. We detected 93 core bots that exclusively post $@mention$ sets using two different third partly applications: *Hootesuite* and *Android Follow Friday Assistant*. Core bot accou Another 1074 promoted bots post messages from one or more of the applications, and are mentioned by core bots. Both account types are depicted in red in Figure 6.4. The gray semi-transparent circles in both panels depict the remaining 106K users in the ALT16 dataset. In both panels the x-axis depicts in-degree coreness in the mention graph, or "what users are highly mentioned by accounts that are highly mentioned." The y-axis in the left panel depicts follower count in log scale, and the y-axis in the right panel depicts coreness in the comention graph. In each case it is clear that measures of influence are inflated by McSbN activity. The behavior and its effect on the neighborhood mention network are depicted in Figures 6.1 and 6.4 respectively.

The promoted bots appear to be cyborg bots because they post what appears to be human content from standard sources as well like Android, Iphone, or the Twitter web application,

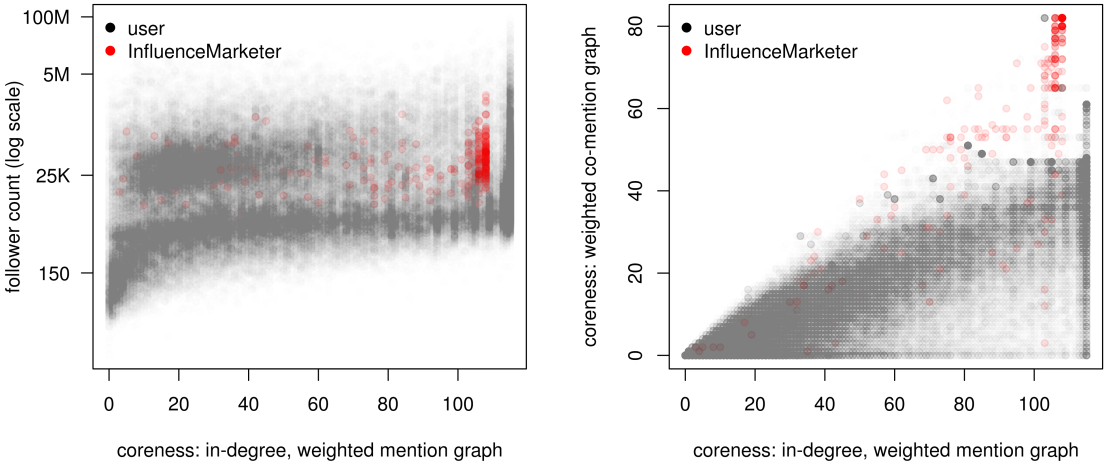
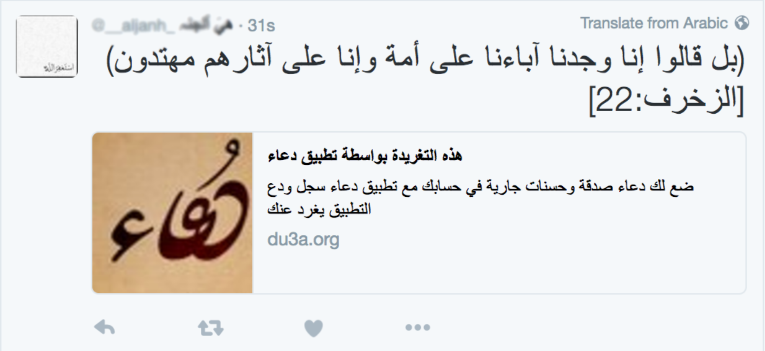


Figure 6.4: Red circles depict members of the #InfluenceMarketer McSbN. The gray semi-transparent circles in both panels depict the remaining 106K users in the ALT16 dataset. In both panels the x-axis depicts in-degree coreness in the mention graph, or ”what users are highly mentioned by accounts that are highly mentioned.” The y-axis in the left panel depicts follower count in log scale, and the y-axis in the right panel depicts coreness in the comention graph.

which leads us to believe that they are cyborg socialbots. In the case of the #InfluenceMarketer SbN we do not know exactly what permissions users authorize the botherder's C&C channel, but they could include adding following ties, as we have seen in the Tweet Remembrance Family of CSbNs discussed in the next section. This class of promotion MCSbN could allow the application developer to sell retweets and followers directly, as well as artificially promote promoted users' graph centrality as discussed in Section 6.4. We have found similar MCSbNs in Hindi, English, Japanese, and Russian, and based on the pervasiveness of this SbN type we hypothesize that they are a profitable endeavor for the botherder. As highlighted by Figure 6.3 we also frequently observe MCSbNs used to promote the distribution of pornography and non-pornographic images. These could be examples more in line with early work hypothesizing the use of social botnets for malware dissemination Boshmaf et al. (2013), but the principles remain the same. Core bot behavior appears to stimulate diffusion and sharing within an online community, and users could subscribe to the MCSbN to promote a product or agenda of their choosing.

Not all CSbNs we have observed are marketed toward promoting users explicitly; we have also observed much larger CSbNs that gain almost full control of large user populations. The Tweet Remembrances Family of CSbNs have become highly popular in the Middle East. An example of content posted using the $@Du3a\_org$ CSbN can be observed in Figure 6.5. This user subscribes to the application $du3a.org$ that tweets Koranic verses hourly from his/her account. Each verse contains the url $du3a.org$, which directs a reader to sign up for the application as well. Upon authorization the application requests permission to to post, retweet, and follow, and the terms of service authorize the application to change these permissions at any time without notifying users. The applications also gains permission to advertise once daily from users'



[redacted] · 31s

Translate from Arabic

(بل قالوا إنا وجدنا آباءنا على أمة وإنا على آثارهم مهتدون)
[الزخرف:22]

هذه التغريدة بواسطة تطبيق دعاء

ضع لك دعاء صدقة وحسنات جارية في حسابك مع تطبيق دعاء سجل ودع
التطبيق يغرد عنك

du3a.org

Figure 6.5: depicts cyborg botnet behavior common to the Tweet Remembrance family of Cyborg Socialbot Networks. These applications tweet 'Islamic blessings' from users' accounts hourly, but some also assume the ability update following ties and retweet as well. @$Du3a\_org$, the most popular example we know of, had over 125K active users as of August 2016.

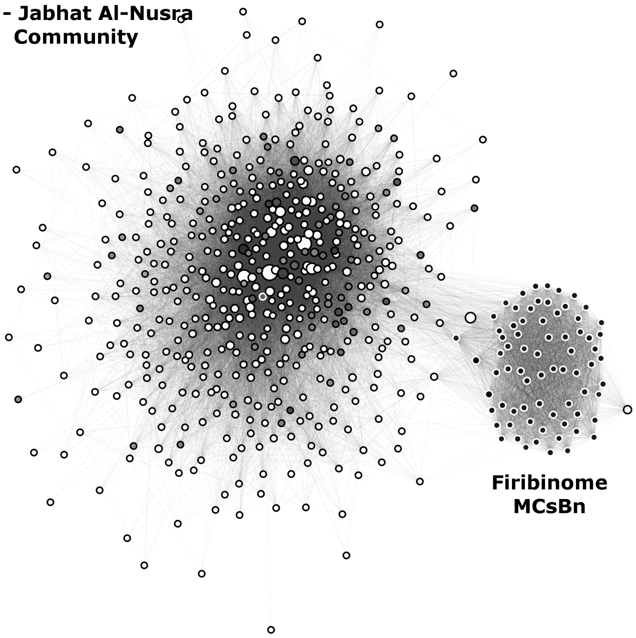
accounts, which could serve as the primary objective of the CSbN. As of August 2016 we collected posts from over 125,000 unique accounts using the $du3a.org$ application. The success of We have found nine nearly identical CSbNs within our data which we believe is consistent with this hypothesis. Although we do not imply that $du3a.org$ violates any of Twitter's terms of use, we highlight that the @$Du3a\_org$ botherder has the ability to adjust retweet and following networks using over 125,000 accounts which could be monetized as well. Additionally, if the MCSbN/CSbNs observed like #InfluenceMarketer have these same privileges they would have a large base from which to sell retweets and followers as well.

It is important to note that the #InfluenceMarketer and @$Du3a\_org$ SbNs do not appear to violate Twitter's terms of use, and the cyborg users who join them give the SbN explicit permission to perform these tasks. Many of the promoted bots we observe in the #InfluenceMarketer MCSbN have clear objectives. Some of professionals using Twitter as a means to market their businesses, others are bloggers seeking to increase their following. In each case, subscribers gain a form of artificial trustworthiness from the SbNs service Ferrara et al. (2016a) that is misleading and could be used for harmful ends. One concern as the ability to use similar methods to generate artificial trustworthiness for geopolitical ends.

6.5.2 MCSbN Propaganda Dissemination

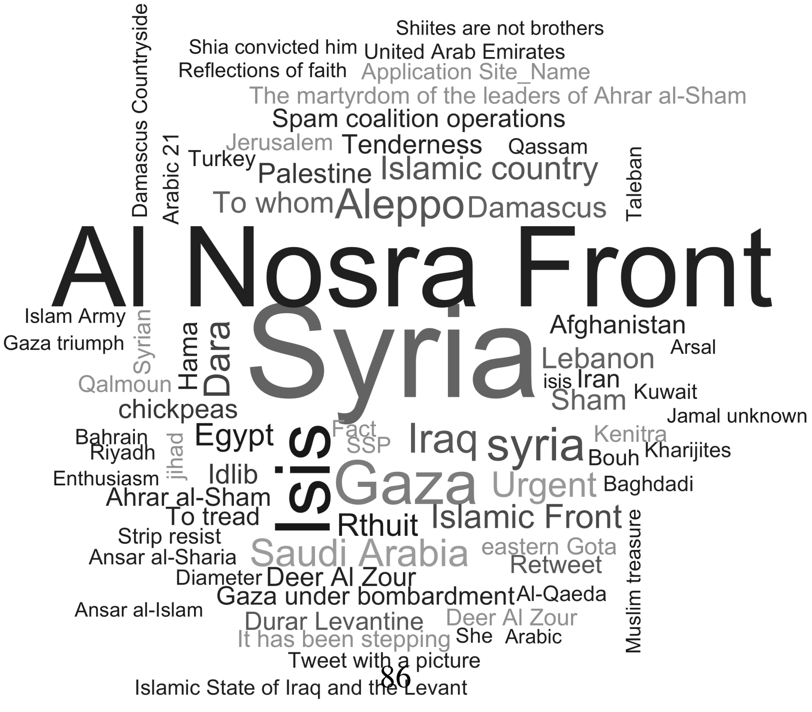
As stated in Section 6.2, evidence of SbNs designed to influence geopolitical discussion are becoming increasingly common, and we observe MCSbNs designed for these ends as well. Our earliest and most overt example of a propaganda MCSbN is the Firibinome MCSbN. This MCSbN consisted of 62 core bots that all shared the same profile image, the Jabhat al-Nusra flag, and similar naming convention consisting of alpha-numeric strings after one of two prefixes translated: firibinome, jishalba. The accounts each followed between 116 and 134 accounts (variability could have been due to account suspensions), most of which were other core bots. Their following counts varied from 142 to 322 accounts, many of whom appeared to be real Twitter users. Each core bot posted 71 to 170 tweets over 38-58 days with each post exemplifying MCSbN behaviors. Furthermore, each of the tweets were posted using the same third party application, "tweetbots.com", which is consistent with automated control from a single botherder. [citation

Firibinome Social Botnet



Directed Mention Network

Hashtag Frequency

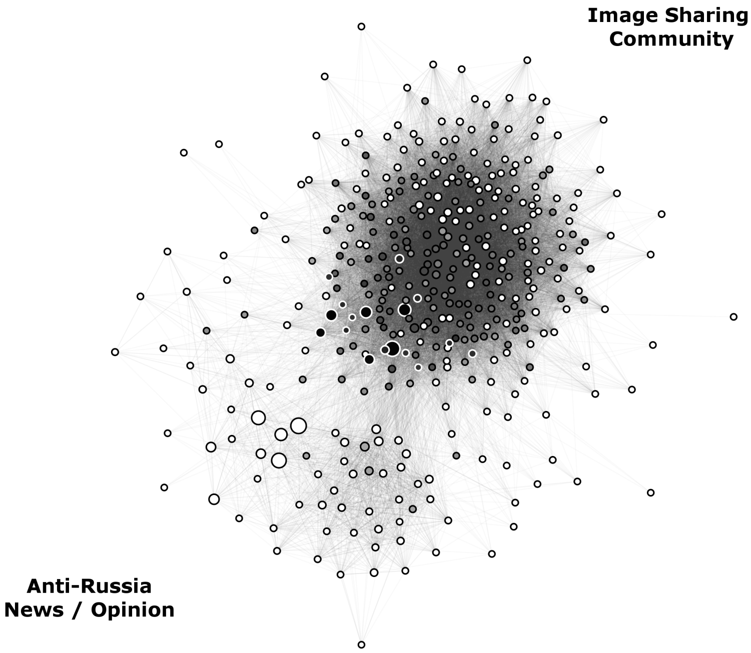


removed for blind review] also show evidence of the botnet's ability to generate discussion of promoted accounts. All 62 core bots of the Firibinome MCSbN were collected as part of the ISIS NOV14 dataset, and we develop an undirected network of all accounts mentioned by or mentioning the SbN. The network is depicted in the top panel of Figure 6.6. Nodes are colored by degree in the reciprocal mention network, R as defined in Section 6.3, from gray (low) to black (high). Nodes with white fill are mentioned by core bots, but do not mention them back. Nodes are sized based on follower count, and framed in white were classified as core bots as described in Section 6.4 with $\alpha = .1$. We summarize the content of users in R based hash tag frequency and visualize the top 75 terms in the lower panel of Figure 6.6. The structure of network plot is consistent with the hypothesis that MCSbN was designed to enhance the botherder's influence within the Pro-Jabhat al-Nusra Twitter community, specifically the large, white nodes linking the two clusters within the figure. Firibinome represents a propaganda MCSbN is its most primitive, overt form.

Propaganda MCSbNs could be used to bridge online communities. Posting tweets that mention users within distinct online communities could cause those users to follow one another. Additionally, if Twitter uses triadic closure to recommend followers MCSbN behavior could achieve this end as well. We detected the a MCSbN possibly designed to bridge Ukrainian users who share artistic nude images with communities sharing news content supporting the Euromaidan movement. We refer to this MCSbN as the Euromaidan Image Sharing (EIS) MCSbN, and detected it using the same parameters used to detect the Firibinome MCSbN. The EIS MCSbN consists of 8 core bots, most of which post hundreds of artistic nude images per day, each of which contains a list of $@mentions$. The accounts they mention however belong to two relatively distinct communities. Some of these mentioned accounts appear to be users interested in similar pictures with similar posting behavior. Others are accounts used for anti-Russian political activism. Figure 6.7 depicts the EIS MCSbN and is formatted similarly to Figure 6.6. However, hash tags in the bottom panel are colored based on their affiliation with the Euromaidan-supporting community (gray) or image sharing community (black). K betweenness centrality is often used on large graphs to identify nodes that connect distinct communities Ediger et al. (2010). The measure quantifies how often a vertix lies on a shortest path of length no longer than k between two other vertices in a graph. Because the core bots in the EIS MCSbN function at the control of one botherder, we collapsed all of their edges into one node and calculated the MCSbN's k-betweenness centrality with k=5. The result was the the EIS MCSbN had the highest k-betweenness centrality of all 92k users within the dataset which highlights the power of MCSbN behavior. Such methods could serve as a means to young men to geopolitically charged news as an initial phase of radicalization. Although different content is used in this case, similar grooming methods have been observed in ISIS' recruiting strategy Berger and Morgan (2015a).

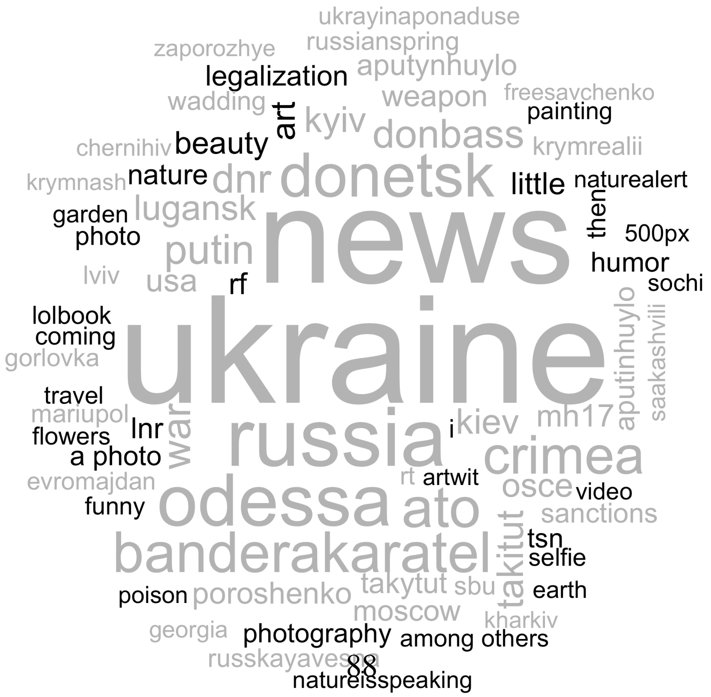
Within each of our datasets we have found a variety of overt instances of MCSbNs and CSbNs. We discuss the challenge of confirming an MCSbNs when core bot behavior is more human-like in Section 6.6, but we have found core-bot-like behavior frequently in highly dense subgraphs in each of our datsets. In some cases the core-bot-like users posts hashtags and mentions, in other cases the user posts some mentions lists, and often mentions multiple users within human like tweets. Figure 6.3 highlights several dense subgraphs that some would interpret as discussion cores Bessi and Ferrara (2016). However, they may very well be MCSbN-assisted. The figure highlights three examples where the core-bot-like behavior is more nuanced, but the

Euromaidan Image Sharing Botnet



Directed Mention Network

Hashtag Frequency



discussion core appears centered around a geopolitical objective. The “Evangelical” discussion core consists of users who appear to be middle aged American women who overtly identify themselves as evangelical Christians through their profile descriptions, usernames, and profile pictures. However, the majority of their content is political in nature. Although we are continuing to study this botnet, it could be used to bridge the evangelical voting community with a particular candidate or it could simply be another instance of simulating a fake grass roots movement. The #opisis discussion core appears to be a self-organized news community designed to fight terrorism. Although we have not verified the content shared by this community or the presence of fake news within it, we highlight the dangers of charging such a topic area with artificial trustworthiness. Finally, the #BoycotIsrael discussion core appears to be an attempt promote a counter-narrative within a broader community. In each case further study is warranted to understand the possible role of bots in their promotion, as well as the effectiveness of their promotion. We recognize the need for additional work with respect to this important topic.

6.6 Limitations and Future Work

Even with the growing body of socialbot detection literature social bots remain a pervasive presence within social media. We find the performance of dense subgraph detection as a means to detect MCSbN core bots drops when core bot mention behavior becomes more human like. In both the Firibinome MCSbN and #InfluenceMarketer SbN, we find that each core bot post is delivered using an identifiable third party application. In the case of the #InfluenceMarketer SbN all core bot activity and promoted bot retweets are posted using one of two applications: *Hootesuite* and *Android Follow Friday Assistant*, the *tapbots* application is used in the Firibinome MCSbN. In both cases we assume the respective applications used to post core bot mention lists as the botherder’s C&C channel, and that all users posting with the C&C channel and showing reciprocity with two or more verified core bots to be core bots as well. In the case of Firibinome, we originally detect 46 core bots and find an additional 16 through the aformentioned criteria. The #InfluenceMarketer bot, as described earlier, incorporates users to assign core bot like behavior. Dense subgraph detection identified 61 core bots, and tweet analysis identified another 32. Another 1979 users are most likely promoted bots in that they are mentioned by core bots and post using the C&C channel. We acknowledge that these assumptions might positively bias are estimates of recall if additional C&C channels are used, but we see no alternative. Figure 6.8 depicts estimated ROC curves associated with Algorithm 3 with α values ranging between $(0, 1)$. The #InfluenceMarketer MCSbN is depicted by the black solid line and the Firibinome MCSbN is depicted by the gray dashed line. Again, we manually inspect dense communities returned from Algorithm 2 which explains why the curve depicted in Figure 6.8 starts with recall of 75.2% and precision of 97.2% for the #InfluenceMarketer MCSbN and recall of 74.2% and precision of 100% for the FiribiNome MCSbNs respectively. The figure highlights highlights the highly detectable community structure exhibited by MCSbNs.

Further study into how these bots manipulate radial-volume measures of centrality merits further study as well. For example, we have not fully explored each botnet’s abilty to confound metrics like betweenness centrality and PageRank. Furthermore, because these SbNs are deployed in some cases as one structure consisting of many users, it would be usefull to explore their influence in the graph when collapsed into one node. The results shown with respect to

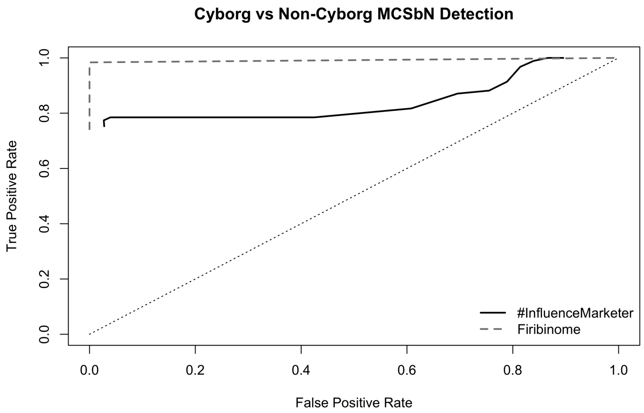


Figure 6.8: depicts estimated ROC curves associated with α values ranging between $(0, 1)$. The #InfluenceMarketer SbN is depicted by the black solid line and the Firibinome MCSbN is depicted by the gray dashed line.

the Euromaidan Image McSbN imply that these structures can prove more influential than single users. However, an in depth study of these questions is needed.

Although dense subgraph detection identifies the core bot network of MCSbNs well, it does not appear to adequately identify promoted bots. We see potential in augmenting a dense subgraph based approach with instance based or node centric approaches like those mentioned in Section 6.2. In fact the binary classification on annotated graphs offers a means to use both social graph location, and node level features, and we have successfully implemented this strategy to detect Online Extremist Communities [under review PLOS ONE, citation removed for blind review]. This would enable practitioners to use both dense subgraph detection and traditional bot detection approaches in concert with one another, and we hope to apply these methods for MCSbN detection in future work. Such advances will be necessary in order to conclusively identify artificial promotion within geopolitical discussion cores.

As stated by Boshmaf, Muslukhov,Beznosov, and Ripeanu (2012), “defending against malicious socialbots is an arms race”, and it is unlikely that detection efforts alone will offer adequate mitigation. Boshmaf et al. rightly propose a framework of socio-technical challenges for mitigating the effect of SbNs organized in three lines of effort: web automation, online-offline identity binding, and usable securityBoshmaf et al. (2012). With the growing body of literature that highlights the impact of large-scale manipulation of OSNs, responsible incorporation of safeguards along all three lines of effort are likely needed. We hope that this work helps motivate future research in this regard.

6.7 Conclusion

In this paper we have discussed the dangers of unmitigated proliferation of propaganda promoting SbNs in Twitter and defined to specific classes of them: MCSbNs, and CSbNs. We have also shown their increased frequency and their application to both individual user promotion and propaganda dissemination. We have also presented 4 novel instances of these SbNs that highlight the scale with which they are employed, and discussed other possible MCSbNs embedded within online discussion cores. We also have introduced the concept of applying dense subgraph detection to identify MCSbN cores, and propose future research to further automate the detection of artificially promoted users.

Chapter 7: Discussion

Understanding social media's emergent role in the shaping of geopolitical opinion is critical. From the Arab Spring (Gerbaudo, 2012; Howard et al., 2011; Wolfsfeld et al.) to the Color Revolutions in Eastern Europe (Jurgenson, 2012; Szostek), to the rise of populist movements like Brexit (Engesser et al., 2016) and extreme-right fake news associated with the 2016 United States presidential election (Allcott and Gentzkow, 2017; Benkler et al.) many ascribe online social networks as being a conduit to energize offline behavior. Having operated in areas of the world now affected by these OECs, I have great interest in applying quantitative methods to assist social scientists and practitioners in understanding these communities and movements. The work presented in this thesis represents an initial step toward understanding of these powerful phenomenon.

Contributions

Although the body of work in this area of research continues to grow, there appears to be a needless bifurcation in the field. The findings of researchers studying the use of social media in the spread of violent extremism Al-khateeb and Agarwal (2015); Berger and Morgan; Ferrara et al. (2016b) observe similar phenomenon to those who discuss social media use for political activism. Furthermore, our observation of both types of activism appears to indicate that these groups self organize using a the affordances provided by the OSN. Communities are formed and curated in Twitter using hash tags, direct mentioning, as well as following relationships. This complex, dynamic environment forces us to use complex, heterogeneous graph representations to precisely identify these groups of users.

The work in this thesis provides a new ways for researchers to answer three critical research questions with respect to online marketing and its role in geopolitical opinion:

- How can we detect large dynamic online activist or extremist communities?
- What automated tools are used to build, isolate, and influence these communities?
- How can we gain novel insight into large online activist or extremist communities?

The methodological framework and associated algorithms presented in Chapters 3 and 4 (depicted in Figure 7.1) will enable researchers to quickly isolate specific online communities for study. While the methodologies presented in Chapter 5 will enable researchers to mine these large online communities for novel insight. Furthermore, the detection methods and description of social influence botnets in Chapter 6 is an important contribution toward understanding the tools used to manipulate online communities. The specific algorithms presented throughout this thesis merely represent what I have found useful for these tasks, and it is likely that other algorithms could be useful when applied to other types of communities or within other OSNs.

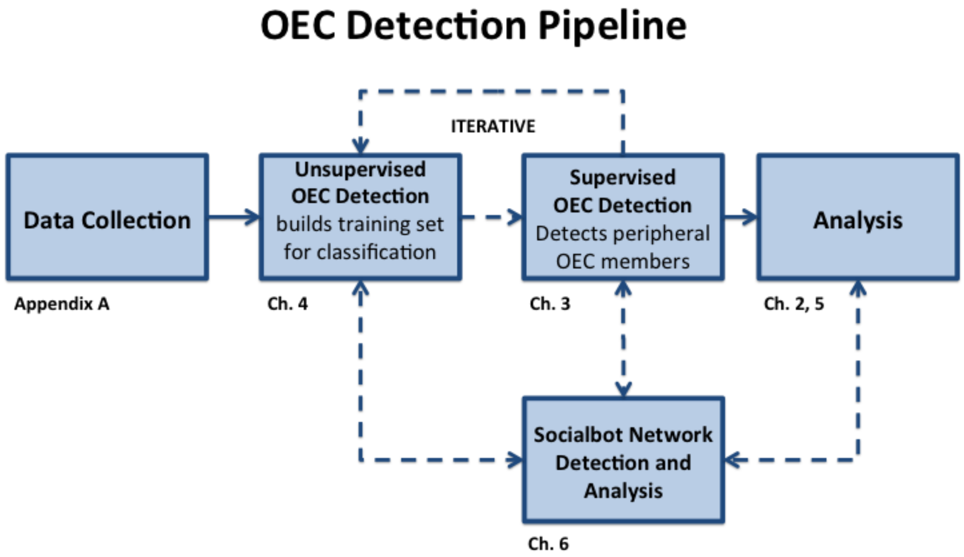

OEC Detection Pipeline
ITERATIVE
Data Collection
Appendix A
Unsupervised OEC Detection builds training set for classification
Ch. 4
Supervised OEC Detection Detects peripheral OEC members
Ch. 3
Analysis
Ch. 2, 5
Socialbot Network Detection and Analysis
Ch. 6

Figure 7.1: Depicts OEC detection as a methodological pipeline. Data Collection is conducted using snowball sampling then training sets are developed using unsupervised methods like HDSD and HEAC. Larger portions of the OEC can then be detected using supervised learning.

Moreover, the methods and data developed are both re-usable and can be made available upon request. A tutorial covering the methods introduced in Chapters 2,3,4, and 6 is provided in Appendix **??**. My hope is that this work both equips and motivates ongoing collaborative research between computational and social scientists.

Moving Forward

As many before me have found scholarship often leaves one with more questions than at the start. The contributions presented in this work could be extended in a number of fruitful areas. One such area would be a more formal framework to incorporate latent data structure to prioritize manual labeling of users. This could be done by gaining a better understanding of the latent substructure within large OECs like the SRTC. Applying clustering algorithms like DBSCAN (Ester et al., 1996) or others(Sajana et al., 2016) to MVC feature spaces could prove useful. It is also likely that specific clustering algorithms will outperform based on the amount of substructure within OECs. Furthermore a great deal of latent information remains in user profiles. For example, URL sharing has become increasingly popular as Twitter has become primarily a publication platform. The techniques used to leverage hash tags in this work could easily be extended to URLs. A great deal of linguistic information remains untouched in users' tweets as well. Topic modeling could offer useful discriminatory value as well and could be incorporated in a relatively straight forward manner in MVC feature spaces.

Another equally important area for continued work would be better incorporation of temporal dynamics. Many of the weighted graphs utilized in each of the chapters have temporal information which we have not incorporated. For example each edge within an undirected mention graph represents a time series of tweets between users a and b. However, the complex relationships between users and content with detected communities preclude us from leveraging nearly all statistical methods that address auto-correlation. Simple extentions like incorporation of temporal decay when developing edge weights would likely be beneficial and could likely be done with little computational expense. However, this problem is complicated by the limitations imposed by the Twitter API. Currently, the API only allows a researcher to download up to a users last 3200 tweets. This dramatically effects the time interval defined by a given users first and last tweet collected. These differences from user to user influence the time potential time interval between any two users for there to be interaction. Thus out weighted edges within such a graph could be normalized with respect to time. Furthermore, we have not utilized the time stamps on tweets to research diffusion patterns with activist communities. Incorporation of the rich temporal information that exists within our data would likely improve findings.

It is also likely that practitioners or follow-on researchers will find the relative cost of data low, but the cost of manual labeling high. It is also possible that initial supervised modeling efforts may prove to have insufficient accuracy. A formalized active-learning framework for updating Multiplex Vertex Classification results could prove highly beneficial. In some cases active learning techniques have proven to exponentially reduce training data needs (Settles, 2010b). One such class of active learning is called uncertainty sampling and prioritizes unlabeled instances that in some respects are “closest” to our classifier’s decision boundary. In my experience the speed and performance of decision forests have been preferable for this task and some literature exists that could enable active learning strategies. Criminisi et al. (2012) introduce the concept of density forests that assuming the unlabeled dataset was created via a probabilistic density function which is to be learned via the datasets latent structure by minimizing an entropy function with each tree. This framework then provides an understanding of where areas of high uncertainty are within the dataset to prioritize manual labeling. The challenge associated with this framework lies in the authors assumption of a Gaussian Mixture Model for their probabilistic generating function. Such models typically assume relatively simple covariance structures which are likely invalid when applied social networks where our users’ interdependence of primary interest.

In general, an active learning framework would enable the methods presented in this work to empower the researcher or analyst employing them. Often the end user of these methods would likely have unique regional expertise that could be incorporated into training sets. The ability of an end user to interact with this data and efficiently provide feedback (or labelled data) to the training set offers great potential. In fact, a well user-oriented graphical interface would enable the end user to improve his algorithm while exploring/interacting with the data. Such a system could identify small sets of seed agents for follow on exploration. For example, an analyst interacting with the Syrian Revolution Twitter Community might find a set of users sharing photographs and new stories in Yemen. My experience in developing these methods is that often times information in user timelines is not necessarily embedded in text. Often times images or shared videos are what indicate the individuals support of specific groups or ideologies, and effective image detection or video detection are unlikely to be quickly employed in this context. An interface that efficiently capitalizes on the regional experts domain knowledge to empower him or her with more data is a worthy human-computer interaction endeavor.

The proliferation of social influence botnets in a variety of marketing applications implies the need for a significant research efforts, and as presented in Chapter 6 there is very little research in this important area. Although a significant contribution of this work is our detection methodology, we have not found an effective means to measure the effect of these network structures on individual perception. The prevalence of them within marketing areas like pornography imply their effectiveness; however we have been unable to quantify this formally. Effort to do so could present a critical step forward in understanding how online activist communities are formed and manipulated.

All of these shortcomings could be compounded as efforts to remove groups and narratives become more advanced. Just as the rich affordances offered by online social networks can be used to curate groups, they can be used to obscure observation and intervention. For example, we have started to observe community managment tools like *commune.it* that could be used to hide relevant traffic within a community through spam. These software tools offer the ability to post messages at scale as well as filter content at scale. They could easily be used to hide relevant OEC communication within a sea of what appears to be irrelevent content. By simply assigning a hash tag or term that can be filtered on, group members can remove the content that obscures relevant tweets.

Community management software applications are just one observation from within these dynamic information ecosystems. As methods to influence online communities and intervene in OECs mature, groups will adapt. We have already observed predator-prey like evolution within this work, and expect this behavior to continue. Thus the methods presented in this work must be contiually refined to address these changes in group curation methods.

Finally and most importantly, this research is about people. What is the effect on people who's activity draws them into one of these communities? What are the cognitive effects of online community activism? The structures I have observed often appear to have their own news sources and in many cases appear to be composed of individuals highly susceptible to confirmation bias. Where the true contribution of this work will be, is if it can help us understand how individuals become extremists and what can be done to mitigate these processes.

Appendices

Appendix A: Data

Each of the datasets used within this dissertation was collected in a similar fashion, and I will describe my collection methods and each dataset within this appendix.

To develop each of my datasets, I instantiate an n-hop snowball sampling strategy Goodman (1961) with known members of my desired network. Snowball sampling is a non-random sampling technique where a set of individuals is chosen as "seed agents." The k most frequent friends of each seed agent are taken as members of the sample. This technique can be iterated in steps, as I have done in my search. Although this technique is not random and prone to bias, it is often used when trying to sample hidden populations Berger and Morgan (2015b).

The snowball method of sampling presents unique and important challenges within OSNs. Users' social ties often represent their membership in many communities simultaneously (Tang and Liu, 2010). At each step of my sample, this results in a large number of accounts that have little or no affiliation with a OEC of interest. The core problem of then involves extracting a relatively small OEC embedded in a much larger graph. In order to do so, I require rigid definitions of account types which will be used for the remainder of this proposal. I define three types of user:

OEC member: A Twitter user who's timeline shows unambiguous support to the OEC of interest. For example, if the user positively affirmed the OEC's leadership or ideology, glorifies its fighters, or affirms its talking points. It is important to mention that a member's *support* is relative and in many cases not in violation of local law or Twitter's terms of use. However, the volume of these "passive members" appears to be an essential element of OECs ability to reach populations prone to radicalization Veilleux-Lepage (2015)

non-member: A user whose tweets are either clearly against or show no interest in the OEC of interest.

official user: I label vertices as *official users* if they meet any of the following criteria: the user's account identifies itself as a news correspondent for a validated news source; the account is attributed to a politician, government, or medium sized company or larger, or accounts with greater than k followers. This third is necessary to account for OEC members' dense ties to news media, politicians, celebrities, and other official accounts. Such accounts are interesting in that there higher follower counts and mention rates tend to make them appear highly central even though they do not exhibit any ISIS supporting behaviors. *Official users* must be identified and removed for accurate classification of ISIS-supporting, thus illustrating the utility of an iterative methodology. This will be discussed in detail in Chapter **??**

Syrian Revolution Community Twitter Searches

Date	Seed Accounts	Search Return	+ Cases	Detected OEC
NOV15	16,538	91,256	3,572	8,126
MAR16	3,295	87,724	2,529	9,086
DEC16	4,258	118,879	4,567	NA

Euromaidan Community Twitter Searches

Date	Seed Accounts	Search Return	+ Cases	Detected OEC
AUG15	8	92,295	1,221	4,307
MAR17	1,175	92,076	4107	NA

Alt-Right Twitter Search

Date	Seed Accounts	Search Return	+ Cases	Detected OEC
AUG15	8	92,295	1,221	4,307

Table A.1: Describes the Syrian Revolution (top panel) and Euromaidan (bottom panel) Twitter searches as well as OEC detection results.

A.1 Islamic Terrorism Datasets

I develop three distinct datasets with respect to discussion centered around ongoing conflict in the Middle Easte. The first instance is the ISIS NOV14 dataset discussed in Chapters 3 and 4, but I have used IVCC to update this dataset twice. Each instace is described below.

- **ISIS NOV14 Twitter Search (ISIS NOV14) (Search Date: November 2014)** This dataset consists of a 2-hop snowball sample of 5 influential ISIS propagandists'Carter et al. (2014) following ties conducted in November of 2014. The search resulted in 119,156 user account profiles and roughly 862 million tweets. Approximately 18,000 accounts have been deleted or suspended by Twitter as of April 2017, and many of these appear to deleted due to support of violent extremism. This offers a unique set of meta-data that can be inferred as ground truth. For details see Chapters 3 and 4.
- **Syrian Revolution Twitter Community Search (SRTC NOV15)** Using 16,538 active (as of NOV15) accounts predicted as ISIS-supporting in Chapter 3, I seeded a 1-hop snowball sample of users following and mention ties resulting in 91,256 accounts and 179 million tweets.
- **Syrian Revolution Twitter Community Search (SRTC MAR16)** Similarly, I collected another update of the SRTC in March of 2016 using 3,295 accounts' following ties resulting in 87,724 users and 185 million tweets.
- **Syrian Revolution Twitter Community Search (SRTC DEC16)** I again updated the SRTC in December of 2016 using 4,258 accounts' following ties resulting in 118,879 users and 230 million tweets.

A.2 Euromaidan Datasets

The Euromaidan movement started as a series of protests in November 2013, where large numbers began to call for the removal of then President Viktor Yanukovych. These protests reached their peak in February 2015, ultimately leading to the removal of many of Ynukovych's senior officials, and were a precursor to Russia's subsequent occupation of Crimea. Despite the installation of a new government, a substantial online activist community continues to oppose Russian influence in the Ukraine and are often described as part of the Euromaidan Movement (Szostek). We will refer to this community as the Euromaidan Twitter Community (ETC). Here, although strong negative sentiment toward the current Ukrainian government is observed, the online activism seen largely advocates change through legitimate government processes. Thus, while we acknowledge that little "extremism" exists in this community, this community is of strategic interest to organizations like the North American Treaty Organization (NATO) due to its relevance to ongoing geopolitical events in the region.

- **Euromaidan Twitter Community Search (ETC AUG15)** This community was extracted originally from a two-hop snowball sample of 8 known Euromaidan movement members' mention ties in March 2014. The search resulted in 92,295 Twitter accounts, and manual inspection of EAC output yielded a community of 1,221 accounts actively supporting the movement.
- **Euromaidan Twitter Community Search (ETC MAR17)** Similar to SRTC updates, the 1,209 remaining active members of ETC were used as seed users to update the community. A 1-hop snowball search of following ties in March of 2017 resulted in 92,706 Twitter users and 212 million tweets.

A.3 Alt-Right Twitter Search

alt-right community (ALT16) dataset. In October, 2016 we seeded a one-hop snowball sample of 2482 users who each followed 5 influential Twitter users associated with the Alt-Right political movement: Richard Spencer, Jared Taylor, American Renaissance, Milo Yiannopoulos, and Pax Dickinson. The search resulted in 106K users and 268 million tweets.

Appendix B: OEC Detect Tutorial

Tutorial: Detecting Pro-Maidan and Anti-Maidan Communities on Twitter

Authors: "Dr. Matthew Benigni and Dr. Kathleen Carley"

Introduction

Online social networks have become a powerful venue for political activism. In many cases large, insular online communities form that have been shown to be powerful diffusion mechanisms of both misinformation and propaganda. We see these methods primarily as a means to study online communities and their role in political activism. To do so we will use online discussion centered around the Euromaidan Movement on Twitter. The Euromaiden Revolution occurred as a wave of demonstrations starting in Ukraine in November 2013 and resulted in the removal of Ukrainian President Viktor Yanukovych from power. A large online community centered around the Euromaidan Movement still exists on Twitter and shares content that opposes Russian influence within Ukraine. However, a sizeable counter movement is present as well. We will refer to these two online activist communities as the Pro-Maidan and Anti-Maidan communities respectively. We will use these two groups to illustrate Iterative Vertex Clustering and Classification, a methodological pipeline developed my Matthew Benigni as part of his doctoral thesis work at the Center for Computational Analysis of Social and Organizational Systems (CASOS), and hope that these methods enable researchers to better understand how online communities form and influence political outcomes.

In this tutorial we will start with a samle of over 90,000 Twitter users and identify 7500 Pro-Maidan users and 6000 Anti-Maidan as a supervised learning task with evaluated accuracy over 92%. We use an unsupervised learning method called Ensemble Agreement Clustering to develop a sufficiently large number of positive case examples of each group. Once we have labelled data, we develop feature space that accounts for social structure described by following and mention ties, as well as hash tag use and user profile characteristics. To do so, we extract spectral features from each graph independently, and train a random forest classifier to detect each user type. Descriptions of this methodological pipeline are are presented in detail in the following works:

1. Benigni, Matthew. " Detection and Analysis of Online Extremist Communities (Unpublished doctoral thesis)." Carnegie Mellon University School of Computer Science, Pittsburgh, PA.
2. Benigni, Matthew, Joseph, Kenneth, and Carley, Kathleen. n.d. " Online Extremism and the Communities that Sustain It: Detecting the ISIS Supporting Community on Twitter." To appear PLOS One.
3. Benigni, Matthew, Joseph, Kenneth, and Carley, Kathleen. n.d. " Mining Online Communities to Inform Strategic Messaging: practical methods to identify community-level insights." To appear Computational & Mathematical Organization Theory.
4. Benigni, Matthew and Carley, Kathleen. "From Tweets to Intelligence: Understanding the Islamic Jihad Supporting Community on Twitter." Springer, Social Computing, Behavioral-Cultural Modeling and Prediction,To appear Spring of 2016.

Tutorial Materials and System Requirements

Source code and data is available on at **foundation.casos.cs.cmu.edu:/usr1/mbenigni/euromaidan/output/tutorial/** all files referenced in this folder are available at the provided paths. The tutorial is designed to be executed in RStudio by executing the various "chunks" of code annotated in this document, which are included in the tutorial zip file as **tutorial_script.R**.

We have executed each of the code blocks on a MacBook Pro with 8GB of RAM and a 2.8 GHz Intel Core i5 processor running OS X Yosemite Version 10.10.5. We have not debugged this work in Windows or Linux and welcome feedback.

Data

In previous work we have used many different sampling strategies, but find that snowball sampling known members' following ties typically returns useful results. Snowball sampling is a non-random sampling technique where a set of individuals is chosen as "seed agents." The k most frequent accounts followed by each seed agent are taken as members of the sample. This technique can be iterated in steps. For example, in a two-hop snowball sample of users' following ties we would take the union of our seed agents' following ties to define the seed agents for hop 2. Although this technique is not random and prone to bias, it is often used when trying to sample hidden populations. It is also worth noting that it tends to return very large, noisy data sets due to users simultaneous membership in many online communities. In fact we have recently found that snowball samples need to be executed one hop at a time. Following networks have become quite dense due to cyborg and bot activity, and often times rules need to be applied to trim returns in the second hop of the sample. The majority of those accounts returned by this search technique are typically not of interest, and standard network clustering techniques fail to extract our communities of interest with adequate precision. Therefore we need to partition this set of users in two groups: community members, and non-members. This data set consists of a one-hop snowball sample of 1,209 previously detected Euromaidan-supporting users' following ties.

Overview

We will detect these communities using a pipeline of methodologies. It is useful to think of these methods in the following phases:

1. **Collection:** (assumed complete for this tutorial) This can be done using twitteR or Tweepy or any of the related libraries/packages designed to collect data from Twitter's API. We use (twitter_dm)[https://github.com/kennyjoseph/twitter_dm].
2. **Cleaning and Graph Construction:** (assumed complete for this tutorial) In this chunk of code we will conduct some minor cleaning of our edgelists to form an annotated, heterogeneous graph. The graph is heterogeneous because it has multiple edge types (mention ties, following ties, hash tags, etc.) and multiple node classes (users, and hash tags); it is annotated in that the nodes of the graph have useful information associated with them like profile attributes.
3. **Removal of Official Accounts:** In this section we will extract an instance-based learning feature space (i.e. rows are users and columns are numerical features) that accounts for each of our graphs and profile information. We then use a list of known "official" accounts to train a classifier and identify similar accounts for removal.
4. **Training Set Development:** training a classifier requires large sets of labelled users. We have developed a method that detects sets of active users who align around an identifiable cause. This method is similar to dense subgraph methods in that we are not interested in a complete clustering of the data, but only users who fall into one of these "discussion cores." We run this clustering algorithm and then manually inspect output to find positive case instances to train our classifiers.
5. **Detect The Pro-Maidan Twitter Community:** using the same methods used to remove "celebrity-like" accounts in Phase 3, we build a new feature space (with celebrity-like accounts removed) and train a classifier to detect the Pro-Maidan Community.
6. **Detect The anti-Maidan Twitter Community:** using the same feature space, and anti-Maidan training examples, detect the Anti-Maidan Cummunity.

Load functions for tutorial

```
#Chunk 1: Load OEC Detect Functions
source('sourceCode/main.R',chdir=TRUE)
```

Phase 1. Collection

1213 seeds (see **foundation.casos.cs.cmu.edu:/usr1/mbenigni/euromaidan/euromaidanUpdate.csv**) following ties resulting in 97,354 files (see ~/**snowballEuromaidan.csv**). The search returned 92,706 accounts. Often times some accounts have protected their tweets resulting in fewer returns than the initial search list. To execute the collection we use the Python library twitter_dm. We use the script

```
#Chunk 1a
nohup python /usr0/home/mbenigni/twitter_dm/examples/collect_user_data.py api_key_path
output_path search_id_file_path
```

The search writes two files per user id to the directories **obj** and **json**. The files in the directory *obj* contain profile information for each user, and the files in the **json** directory contain each users tweets.

Phase 2. Cleaning and Graph Construction

I this phase we transform twitter_dm output into a variety of edgelists and a node table. We then transform the edgelists into an annotated heterogeneous graph in R.

In Chunk 2a we again call a Python script to generate the required edgelists. The script can be found in **twitter_dm/examples/netBuilder_serial.py** and requires only one argument the output path.

```
#Chunk 2a
nohup python /usr0/home/mbenigni/twitter_dm/examples/netBuilder_serial.py tutorial/
```

The scripts outputs the following files:

- attribute.tsv - contains user profile attributes like follower count, creation date, etc. Additional attributes associated with the user's timeline are summarized as well like how many @mentions are included in the tweets returned from the user's tweets.
- friend_edge file.tsv - a directed graph edge list of the following or "friend" ties associated with the userID in the Source column of the edge list.
- mention_edge file.tsv - a directed graph edge list of the mention ties associated with the userID in the Source column of the edge list. The epoch time of the post containing the specific @mention is provided as well as the tweet ID.
- user_ht_edge file.tsv - a bipartite graph edge list where the Source node class is users and Target node class are hash tags. Each instance refers to a hash tag used in a given users tweet. Again, the time of the post is provided in epoch time.
- langfile.tsv - Twitter conducts language detection on each post and provides their label in the json of each tweet. This file provides the language prediction for each tweet in our corpus as well as the associated userID. We summarize this information to express a vector of language preference for each user.
- officialIDs.csv - This is a list of accounts that are either journalists, celebrities, or official government accounts. We find that this type of account must be removed through machine learning for accurate community detection, and this list serves as positive case instances for this task.

To build our heterogeneous graph H we start with a summary of information at the user level by uploading a table of user attributes, **attribute.tsv**. We clean this user set by removing two user types: dormant users, and accounts that are likely Mention-core SocialBot Networks. Mention-core SocialBot Networks are artificial social structures used to influence online communities. They typically post long strings of @mentions to confound measures of social influence, so we remove accounts with an anomalously high mention:tweet ratio. In this case the top 99.5 percentile. For more detail see Thesis Chapter 6. Dormant users are defined as any user who has not posted within 12 months of the profile collection date, and we simply remove accounts based on the time stamp associated with the users' last tweet.

We then use this trimmed set of users to build directed, weighted graphs. The mention graph M consisits of users as nodes and an edge is defined as the number of times user i mentions user j. The following graph F is a directed, binary graph where an edge indicates user i follows user j. We also construct a bipartite graph H where source nodes are users and target nodes are hash tags. An edge is defined as the number of times that user i used hash tag j in his last 3200 tweets.

Finally, we normalize each users language behaviors into a vector in langDF. The result is a rich graph structure that allows us to detect communities of interest based on the many affordances offered by Twitter for group curation. *Expect this chunk to take roughly 13 minutes to execute.* In this chunk we introduce four functions: upLoad(), buildDirectedGraph(), buildBipartiteAdjacency(), and languageCast(). Documentation for each function is provided at the end of the tutorial. **This chunk is optional, but can be executed with the following downloaded data file:maidan_tutorial_edgelists.zip**.

```r
# Chunk 2: Build Annotated Heterogeneous Graph
a=upLoad('attribute.tsv')
a=a[order(a$userID),]

mentionRatio=a$mentionCount/a$tweetsCollected
a=a[mentionRatio<=quantile(mentionRatio,.995),]

lastTweet=as.Date(a$lastTweet,origin='1970-01-01')
a=a[lastTweet>=(range(lastTweet)[2]-365),]

F=buildDirectedGraph(nodes=a$userID,'friend_edgefile.tsv')
M=buildDirectedGraph(nodes=a$userID,'mention_edgefile.tsv')
Ah=buildBipartiteAdjacency(sourceNodes=a$userID,
                           edge_path='user_ht_edgefile.tsv',
                           minUserCount=5)
langDF=languageCast(nodes=a$userID,lang_path='langfile.tsv')
```

As a result, we have reduced our dataset from 92,706 users to 83,481 users and constructed a directed following graph, F, mention graph, M, and an adjacency matrix for the bipartite user, hash tag graph, Ah. Each row in the node table a as well as each graph refer to the same user. This graph representation allows us to incorporate a great deal of community behavior for classification which will be shown in the next phase.

Phase 3: Identify and Remove Celebrity-like Accounts

Differentiating between users who are important locally (i.e. within the community of interest), and globally within Twitter is challenging to do. We find that one must first remove “celebrity-like”" accounts as a supervised learning task. In this phase, we will develop a rich feature space by extracting the k lead eigen vectors associated with undirected versions of F and M. We will also use svd decomposition to extract the first k vectors of the “left matrix” associated with our bipartite graph Ah. *This is the most computationally expensive step in this pipeline, and you can expect this section of code to take approximately 10 minutes to run.* Documentation for the classifierFeatures() function is provided at the end of the document.

```
# Chunk 3a: Remove Official / Celebrity Accounts as a Supervised Learning Task
load('RData/tutorial_graphs.RData')
A=classifierFeatures(a,langDF,F,M,Ah,k=5)
rm(langDF,F,M,Ah)
```

We then use a list of manually verified Twitter user ids of journalists, celebrities, and government officials to train a classifier. Anomalously high page rank consistently returns celebrity accounts in large Twitter samples like the one we are working with, so additional positive case inferences are used based on the 99.9% percentile of PageRank centrality in the directed mention graph. The result is a list of 1225 examples of "official accounts" with which to train a classifier. We then use the function trainTestSplit() to randomly select negative case examples and separate our data into a training set, testing set, and set of unlabeled data with which to apply our classifier.

```
# Chunk 3b: Build Positive Case Training Instances
celebLikeIDs=upLoad('celebrityLikeIDs.csv')$V1
celebLikeIDs=unique(c(celebLikeIDs,A$table$userID[A$table$m_pageRank>=quantile(A$table$m_pageRank,.999)]
celebLikeIDs=celebLikeIDs[celebLikeIDs %in% a$userID]


# Develop list of training data, testing data, and data for classification
T=trainTestSplit(nodeIDs=A$table$userID,
                 feature_set=A$features,
                 posIDs=celebLikeIDs,
                 negIDs=NULL,
                 randomNegCount=2000,
                 p.test.split=.4)
```

With our train, test, and classification data now output in list T. We can begin model development. The function classifier can be implemented in evaluation mode, where simply model performance on test data is returned in the console. Additionally, the csv file **featureSet.csv** can be used to select specific features while developing the model. In Chunk 2c we train a random forest classifier to identify and remove official accounts.

```
# Chunk 3c: develop model
classifier(data_list=T,
           algorithm='randomForest',
           evaluation=TRUE,
           metric='Kappa',
           ratio=.5,
           feature_import=TRUE,
           label='celebLike')
```

As can be seen, both accuracy (.9233) and Cohen's Kappa (.8348) are sufficient for implementation. We now apply the classifier to our unlabeled data and inspect results with two html files which are written to the working directory in the folder "pages."

```
# Chunk 3d: apply classifier
predicted=classifier(data_list=T,
                     algorithm='randomForest',
                     evaluation=FALSE,
                     metric='Kappa',
                     ratio=.5,
                     feature_import=TRUE)

pageIt(data.frame(link=hyperlink(a$ScreenName[a$userID %in% sample(predicted,20)])),'predicted.celebLik
not_predicted=T$predictIDs[!(T$predictIDs %in% predicted)]
pageIt(data.frame(link=hyperlink(a$ScreenName[a$userID %in% as.numeric(sample(not_predicted,20))])),'no
```

When satisfied with classifier output, we can remove official accounts and rebuild our feature space. When we are done calculating Chunk 2e Phase 2 is complete.

```r
# Chunk 3e: Remove celebrity-like accounts and rebuild feature space
a=a[!(a$userID %in% as.numeric(c(celebLikeIDs,predicted))),]
v.remove=as.character(c(celebLikeIDs,predicted))
F=delete_vertices(F,v=v.remove)
M=delete_vertices(M,v=v.remove)
Ah=Ah[!(row.names(Ah) %in% v.remove),]
langDF=langDF[langDF$userID %in% a$userID,]
A=classifierFeatures(a,langDF,F,M,Ah,k=5)
```

Phase 4: Develop Training Data

In this phase we will use unsupervised, dense subgraph clustering methods to detect large sets of users who's behavior is of interest. We start by defining two user types of interest:

- pro-Maidan Community Member: a user who displays content that supports ongoing operations to repel Russian influence/forces from Ukraine. This content could be in the form of direct solicitation of support for forces. Sharing sharing of news associated with Russian corruption inside or outside Ukraine is also consistent with the Euromaidan Community.
- anti-Maidan Community Member: a user who displays overt support for Russian influence in Ukraine. These users brand Euromaidan forces as terrorists, denounce the current Ukrainian government, view Putin's influence in the region in a positive manner.

We run ensemble agreement clustering to detect sets of users who organize around an identifiable cause. Through sampling and manual inspection, we will identify clusters that can be useful either as sets of positive case training examples or negative case training examples. The algorithm looks for users in our heterogeneous graph who's behavior displays social similarity in the co-mention, co-following, and user x hashtag graphs. It then returns only users who are co-clustered across each of these three graphs. Often times the groups that are returned display a clear activist cause.

```r
# Chunk 4a: find EAC clusters
eacObj=EAC(nodes=a$userID,F,M,A=Ah,min.user=100)
```

The function checkEACOutput() enables the user to sample user clusters via html output. Each time the user is prompted for feedback, an html file is written to facilitate labeling. As you work through this output, the accounts you inspect will highlight the need for regional expertise and ultimately the need for an analyst-oriented user interface see principles of user-centered design. The user can click on the hyperlink and inspect each detected user's Twitter profile and timeline. For this exercise, if more than 80% of users within a given cluster appear to be pro-Maidan we would label that entire cluster as "pos" for positive case. Similarly, if more than 80% appear to be anti-Maidan we will label the cluster "neg." Both positive and negative case instances are informative for our classifier. For the remainder of the tutorial we will use pre-inspected sets of each group, but feel free to develop your own using code Chunk 3b.

```r
#Chunk 3b: Inspect EAC Clusters and Develop Training Data
L=checkEACOutput(a=a,eacObj=eacObj)
proMaidan=L$pos
antiMaidan=L$neg
proMaidanClusters=unique(eacObj$users$eac[eacObj$users$userID %in% posIDs])
antiMaidanClusters=unique(eacObj$users$eac[eacObj$users$userID %in% negIDs])
```

Phase 5: Detect the Euromaidan Supporting Community

We now develop and train a classifier to detect pro-Maidan community members in the same manner we identified official accounts. In this case we will use positive examples detected with EAC clustering and a combination of negative instance EAC output and randomly selected accounts. It is important to have both positive case and negative case EAC output in your training set, because it enables the classifier to distinguish between dense subgraphs based on content. Again we build our train, test, and classification data sets using our proMaidan users as positive case training examples. Our negative case instances will consist of 1000 randomly selected accounts and accounts we labelled as anti-Maidan.

```r
# Chunk 5a:Develop list of training data, testing data, and data for classification
load('RData/euromaidan_featuresSpace.RData')
load('RData/euromaidan_eacObject.RData')
T=trainTestSplit(nodeIDs=A$table$userID,
                 feature_set=A$features,
                 posIDs=proMaidan,
                 negIDs=antiMaidan,
                 randomNegCount=1000,
                 p.test.split=.4)
```

Again, you can control what features are incorporated using the file featureSet.csv. Now we develop our classifier and evaluate performance on 40% of our labelled data.

```r
# Chunk 5b: Model Development
classifier(data_list=T,
           algorithm='randomForest',
           evaluation=TRUE,
           metric='Kappa',
           ratio=.5,
           feature_import=TRUE,
           label='proMaidan')
```

Once we are sufficiently pleased with performance of the classifier, we train based on all of our labelled data and apply the classifier to our unlabeled data. Again we inspect the results via html pages is the directory “pages”.

```r
# Chunk 5c: Train and Apply Classifier, Detect the Euromaidan Supporting Community, and inspect results
predicted=classifier(data_list=T,
                     algorithm='randomForest',
                     evaluation=FALSE,
                     metric='Kappa',
                     ratio=.5,
                     feature_import=TRUE)

#inspect results
pageIt(data.frame(link=hyperlink(a$ScreenName[a$userID %in% sample(predicted,20)])),'predicted.proMaidan
not_predicted=T$predictIDs[!(T$predictIDs %in% predicted)]
pageIt(data.frame(link=hyperlink(a$ScreenName[a$userID %in% as.numeric(sample(not_predicted,20))])),'no
proMaidanIDs=c(proMaidan,predicted)
write.csv(data.frame(userIDs=proMaidanIDs),row.names=FALSE,file='proMaidanIDs.csv')
```

Phase 6: Detect the Anti-Maidan Community

Here we use the same procedure but simply change which examples we use as negative case and positive case instances to detect anti-Maidan community members.

```
# Chunk 6a:Develop list of training data, testing data, and data for classification
T=trainTestSplit(nodeIDs=A$table$userID,
                 feature_set=A$features,
                 negIDs=proMaidan,
                 posIDs=antiMaidan,
                 randomNegCount=1000,
                 p.test.split=.4)

# Chunk 6b: Model Development
classifier(data_list=T,
           algorithm='randomForest',
           evaluation=TRUE,
           metric='Kappa',
           ratio=.5,
           feature_import=TRUE,
           label='antiMaidan')
```

```
# Chunk 6c: Train and Apply Classifier, Detect the Euromaidan Supporting Community, and inspect results
predicted=classifier(data_list=T,
                     algorithm='randomForest',
                     evaluation=FALSE,
                     metric='Kappa',
                     ratio=.5,
                     feature_import=TRUE)
#inspect results
pageIt(data.frame(link=hyperlink(a$ScreenName[a$userID %in% sample(predicted,20)])),'predicted.antiMaida
not_predicted=T$predictIDs[!(T$predictIDs %in% predicted)]
pageIt(data.frame(link=hyperlink(a$ScreenName[a$userID %in% as.numeric(sample(not_predicted,20))])),'no
antiMaidanIDs=c(antiMaidan,predicted)
write.csv(data.frame(userIDs=antiMaidanIDs),row.names=FALSE,file='antiMaidanIDs.csv')
```

Conclusion

We started with a set of over 92,000 Twitter users and in less than two hours were able to detect distinct communities of 7500 Euromaidan Supporters and 6000 anti-Euromaidan users which can be used for many unique information extractions. What news sources or fake news sources dominate each community? Who are the key voices in each community? What type of bot activity is observed in each community? We have used this pipeline to study Islamic-jihad-supporting communities as well as far-right political communities in the United States with similar performance. The commonality across each detected group seems to be their insularity. Although little is understood about how and why these groups form, our hope is that this methodological pipeline will facilitate research devoted to these important questions. In a subsequent tuturial we will conduct information extractions from the pro-Maidan community to highlight how detected activist communities can inform strategic messaging.

Function Documentation

function: buildBipratiteAdjacency()

Description builds an adjacency matrix from a bipartite edge list. This function, like *buildDirectedGraph()* takes an ordered set of node names for the “Source” node class. However, the “Target” field in this edge list is assumed to be of a different node class resulting in a bipartite graph. The output is a weighted adjacency matrix in sparse matrix format.

Arguments nodes - a vector of ordered node names

edge_path - a file name / path pointing to the graph edge list. The graph edge list is assumed to have field headers “Source” and “Target.” Source nodes must be listed in the *nodes* vector supplied in argument 1. Node names in the “Target” column will be assumed to be of a different node class. In the case of this tutorial these are hash tags; however URLs or some other object could be used.

minUserCount - the minimum number of unique users to have used a unique node of node class B. In the case of this tutorial, the number of unique users who posted a particular hash tag. Because we are interested in clustering users based on hashtags, the graph can be reduced significantly by trimming a large number of edges that do not help provide community structure.

Value a weighted adjacency matrix where rows as users (node class A) and columns are node class B (hash tags in this case)

function: buildDirectedGraph()

Description builds a directed weighted igraph graph with ordered node names. This function ensures that subsequent calculations to fuse graph and node attributes have consistent naming convention enabling the development of our heterogeneous annotated graph.

Arguments nodes - a vector of ordered node names

edge_path - a file name / path pointing to the graph edge list. The graph edge list is assumed to have field headers “Source” and “Target” which contain node names listed in the *nodes* vector supplied in argument 1.

weighted - TRUE/FALSE

Value a weighted, directed igraph graph

function: classifier()

Description facilitates feature selection and training of a Multiplex Vertex Classification classifier. The function recieves a trainTestObject (**see trainTestSplit()**) as input, and when in eval=FALSE mode, returns performance data when the classifier is applied to test data. Once the model development is complete, set EVAL=FALSE and the classifier will be trained on train and test data then applied to unlabelled data. In this mode the function returns a list of userIDs predicted as positive case.

Arguments

data_list - a trainTestSplit object returned from the function trainTestSplit(). This object consists of 3 feature sets “train”, “test”, and “classify”. The first two have “class” labels, while the third does not.
algorithm - the classifier to be trained. The function calls either the packages carat or randomForest based on the classifier selected. ‘randomForest’ generally returns the best results. evaluation - determines which mode to run the function in. (see description for details) metric - defines which metric to minimize when training the classifier. Because class distribution is often skewed in OEC detection, “Kappa” is the default. ratio - the positive class ration required in leafs when using a Random Forrest classifier feature_import - the

function classifierFeatures() outputs a file featureSet.csv that enables the user to easily remove features when training the classifier. When set to true, the function uses only features selcted in the csv file.

Value When eval=FALSE model performance output is returned. When EVAL=FALSE a vector of positve case userIDs is returned.

function: classifierFeatures()

Description

builds list containing two data frames. The first element "features" provides a set of numeric features where rows refer to users and columns are each normalized numeric features. Standard node level features from the node attribute table are normalized. Each directed graph is replicated as an undirected graph where ties are based on minimum reciprocity (i.e. mode='mutual' in as.undirected()). Centrality measures are extracted in each graph and the *k* lead eigen vectors are extracted from both undirected graphs. The *k* lead vectors associated with the left matrix of the SVD decomposition of the bipartite adjacency matrix Ah are extracted as well. This provides a rich feature space which accounts for each graph within our heterogeneous graph as well as information at the node level. A .csv table is also written to the working directory which can be read externally by the function classifier() during model development.

Arguments

a - a data frame of node attributes

langDF - a data frame of language frequencies

F - the directed following graph (igraph graph)

M - the directed, weighted mention graph (igraph graph)

Ah - a weighted, bipartite adjacency matrix (users x hashtags in this case) in sparse matrix format

k - the number of eigen vectors to extract for spectral features

Value

a data frame with the following fields:

User profile characteristics at the time of collection:

followerCount, followingCount, tweetCount, tweetsCollected, firstTweet (date), and lastTweet (date)

Summarized User Behavior

The tags f, m, rf, rm, and uht refer to the following, mention, co-following, co-mention, and user by hash tag graphs respectively.

mentionRatio, urlRatio - the average number of mentions/urls per collected tweet (useful for simple bot detection)

insularity_f,insularity_m - defines the proportion of following and mention ties that are represented in node table a. This can be thought of as a metric of the users "topical isolation" in our data.

graph centrality measures - in each graph we calculate degree centralities, coreness, and PageRank.

graph spectral features - the k lead eigen vectors associated with each graph

language frequencies - ISO codes for respective languages as provided by the Twitter API.

function: languageCast()

Description develops a data frame where rows are users and columns refer to unique languages detected in our corpus of tweets. Each entry is the frequency with which user i tweets in language j.

Arguments

path - a file name / path to an edge list containing the fields *userID*, *lang*, and *count*

Value a data frame

function: trainTestSplit()

Description returns a trainTestSplit object which consists of a list of three feature spaces. The first two provide a training and testing data for model evaluation and are labeled "train" and "test". The last feature space is labeled "classify" and consists of the unlabeled data to which a trained classifier will be applied.

Arguments

nodeIDs - The specific nodes within the feature set to be included in ouput

feature_set - an MVC feature set returned by the function classifierFeatures()

posIDs - a vector of positve case userIDs for training

negIDs - a vector of negative case userIDs for training

randomNegCount - the number of negative case instances to be selected randomly

p.test.split - the percentage of labelled data to be held out for evaluation

Value a data frame A list of 3 feature spaces. The first two provide a training and testing data for model evaluation and are labeled "train" and "test". The last feature space is labeled "classify" and consists of the unlabeled data to which a trained classifier will be applied.

function: upLoad()

Description a specific implementation of the function *fread* from the package *data.table.* This function reads in various delimited file structures and builds a data frame where strings are never converted to factors.

Arguments

path - a file name / path

Value a data frame

Appendix C: Simulating OSN Data: Partially Synthetic Graph Generation Through Modularity-Based Recursive Stochastic Block Modeling

C.1 Introduction

1.1 The problem of community detection has been widely studied within the context of large-scale social networks (Fortunato; Papadopoulos et al., 2011). Community detection algorithms attempt to identify groups of vertices more densely connected to one another than to the rest of the network. Online social networks however present unique challenges due to their size and high clustering coefficients Girvan and Newman (2002). Graph representations of online social networks are further complicated by API rate limits Twitter, and users' high social dimensionBoccaletti et al. (2006); Wang et al. (2010). Although a great deal of literature has been devoted to finding cohesive groups within social media, evaluating community detection methods remains a largely qualitative endeavor Peel et al..

Synthetic graphs often fail to preserve the topological qualities of OSNs making their use for evaluation problematic. The performance of community detection algorithms is likely to be different on graphs with different structure. We present modularity-based recursive stochastic block modeling as a means to alter the community structure of real world graphs in a way that preserves the graph's topological qualities. This method is the first step towards a means to evaluate performance of community detection algorithms on real world graphs when ground truth is unknown. We recursively mine the graph's substructure and randomly permute edges until the recursion no longer returns sub-graphs larger than an established threshold. In this work we show the degree with which the Louvain algorithm's clusters change as the larger proportions of the original graph are permuted. Additionally we investigate changes in topological features as edges are randomly added or removed from the graph. Although we use the Louvain Grouping algorithm Blondel et al. (2008), the same framework could be used to evaluate other community detection algorithms or compare algorithms.

C.2 Background

Community Detection

The problem of community detection has been widely studied within the context of large-scale social networks and is well documented in works like Fortunato (2010); Papadopoulos et al. (2012). Community detection algorithms attempt to identify groups of vertices more densely connected to one another than to the rest of the network. Social networks extracted from social media however present unique challenges due to their size and high clustering coefficients (Girvan and Newman, 2002). Furthermore, ties in online social networks like Twitter are widely recognized as having high social dimension, in that users ties represent different types of relationships (Boccaletti et al., 2006; Miller et al., 2011a; Wang et al., 2010). There are many classes of community detection algorithms though we will discuss two in detail in this work: modularity-based algorithms, and statistical inference based methods. Although this work merely presents two classes of community detection algorithms, the problem of generating realistic ground-truth data persists in all.

The Louvain Grouping algorithm presented in Blondel et al. (2008) is widely used for community optimization within the network science community. Louvain grouping uses a similar objective function as the Newman-Girvan algorithm Newman and Girvan (2004), but is more computationally efficient. In community optimization algorithms, the graph is partitioned into k communities based on an optimization problem that centers on minimizing inter-community connections where k is unspecified. Both Newman and Blondel find these communities by maximizing *modularity*. The modularity of a graph is defined in Equation C.1. In Equation C.1, the variable $A_{i,j}$ represents the weight of the edge between nodes i and j, $k_i = \sum_j A_{i,j}$ is the sum of the weights of the edges attached to vertex i, c_i is the community to which vertex i is assigned, $\delta(u, v)$ is the inverse identity function, and $m = \frac{1}{2}\sum_{i,j} A_{i,j}$.

$$Q = \frac{1}{2m}\sum_{i,j} = [A_{i,j} - \frac{k_i k_j}{2m}]\delta(c_i, c_j), \tag{C.1}$$

An alternative to modularity based methods is statistical inference based approaches. Stochastic block modeling attempts to decompose graphs into vertices with common properties through statistical inference. In our case we will discuss methods to find groups of vertices with structural equivalence or shared neighbors Lorrain and White. The stochastic block model can be learned from real world data as a community detection method, and then used to recover a synthetic representation of the original graph. In a detection scheme, the model decomposes the orignial graph in k groups and calculates sub-graph density within each of the k defined groups, as well as the $(k \times k) - k$ off-diagonal, bipartite subsets of the graph. These parameters can be used along with an Erdos-Renyi model to recover a stochastic version of the original graph Mossel et al..

Simulating OSN Data

Generating large synthetic networks is commonly done to test community detection algorithms. In many cases simple stochastic block models are used as in Chen and Saad (2012). Alternatively, researchers often use other stylized graphs like scale-free modelsBarabsi as in Danezis and Mittal; Eubank et al.. Other methods infer graphs based on real world data such as the methods presented in Barrett et al. Barrett et al.. In this case the authors generate large synthetic social contact networks using real data to form synthetic contact networks based on physical co-location of interacting people. They argue the result if a far superior representation than existing random graph methods. However, in each of these cases a great deal of the real world graph's topology is lost.

C.3 Methods

Simulation Approach

Again, in this simulation we attempt the goal is to modify a real-world online social network's community structure while maintaining the graph's topology. Such a method would enable researchers to test algorithms for their ability to detect changes in subgraph modulatiry where the degree of change is known. Formally we define the problem in two phases.

First, given a sparse, real world graph $G(V, E)$ where, our goal is to detect clusters based on structural equivalence Lorrain and White. We construct the graph M as follows:

- M, a $u \times u$ graph where and edge $e_{i,j,M}$ is defined as the $argmin$ of the numbers of times u_i mentions u_j and the numbers of times u_j mentions u_i. This is often referred to a reciprocal mention network

In this work we use Louvain grouping to discover latent community structure within M Blondel et al. (2008). Our second task is then to alter community structure in M in a manner where styalistic qualities of the graph are retained. To do so we use the k groups defined by the Louvain grouping algorithm and construct K_M, a $k \times k$ matrix where $K_{i,j,M}$ is defined as the density of the off diagonal block betweet groups i and j. We can then use K to develop a stochastic block model S_M as defined in Faust and Wasserman. Each block larger than u_{min} is permuted by using Algorithm **??**.

We extend Algorithm **??** by allowing for recursion within Louvain groups greater than size u_{min}. In theory, by recursively selecting edges for permutation within blocks we hope to retain styalistic structure within the graph. To do so, we continue to exucute the Louvain Grouping algorithm for all clusters larger than u_{min}. Each cluster found that is less than or equal to u_{min} has α percent of its edges replaced with synthetic edges, as well as each off diagonal block within the sub-graph. We define the recursion in Algorithm **??**

INSERT ALGORITHM 2

C.4 Results

In this section we evaluate the sensitivity of the Louvain grouping algorithm Blondel et al. (2008) to changes in network structure as re For the final paper I would like to conduct two virtual experiments to answer the following questions:

> Virtual Experiment 1: How well does Modularity-Based Recursive Stochastic Block Modeling retain OSN graph characteristics as edges are randomly added or removed from the original graph?
>
> Virtual Experiment 2: How well does Modularity-Based Recursive Stochastic Block Modeling retain OSN graph characteristics as the proportion of randomized edges increases?

In both experiments we will use graph M, an undirected mention graph consisting of 8718 twitter users who actively blog about the ongoing conflict in Syria. Network characteristics of M are listed in Table C.1. In each experiment we will vary α, the percentage of synthetic edges generated and measure changes in clustering output.

Metric	**Value**
Nodes	8718
Edges	22,066
Density	5.8×10^{-4}
Transitivity	.099
Triad Count	793,579
Dyad Count	4068
Non-isolate Louvain Clusters	125 clusters / 5573 users

Table C.1: Summarizes M, an undirected mention graph consisting of 8718 twitter users who actively blog about the ongoing conflict in Syria.

C.4.1 Virtual Experiment 1: Recursively Selected Synthetic Edges

In this experiment we will define a sequence of α values ranging from 0 to 1. In each case we will develop 100 replicates, and record the transitivity, dyad count, and triad count, we will also measure the change in clustering output. To do so we define co-clustered users as the numer of users who are clustered together in both the original graph and the permuted graph. Figure C.1 depicts the results of our experiment. The left panel depicts changes in output of the Louvain group clustering algorithm Blondel et al. (2008) as the percentage of recursively permuted edges increases. The x-axis depicts α. The y-axis depics the number of vertices that are clustered together when compared to the Louvain output of the original graph. The plot shows that even when the entire permuted graph consists of synthetic edges, over 4000 of the original 5000 co-clustered users are retained. The source of this error appears to be related to increased graph transitivity as α increases as depicted in the right panel of Figure C.1. It is likely that the reason for this is the algorithm's tendency to increase triads at the expense of dyads as α increases. Figure C.2 depicts the relationship between dyad count and triad count. The large, gray circle depicts $\alpha = 0$, the original graph. Although retension of original community structure here appears strong, additional structure could be retained by ensuring triad counts are not dramatically increased, however such extensions would likely cause significant computational expense and be difficult to scale.

C.4.2 Virtual Experiment 2: Adding or removing group modularity recursively

In this section we quantify changes in community structure when recursive stochastic block modeling is used to add or remove modularity within a sub-graph. For this experiment we will

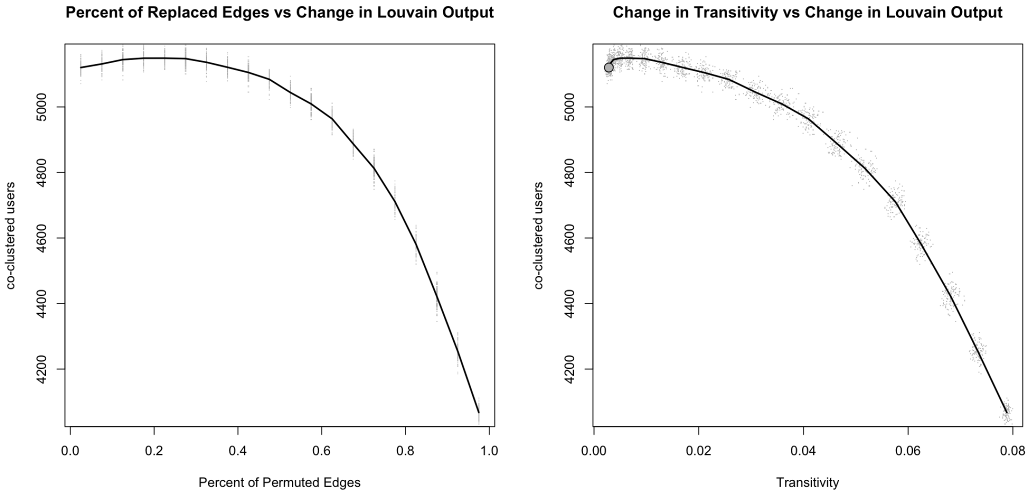


Figure C.1: depicts the results of 100 replicates at 20 different values for α, the proportion of synthetic edges recursively generated. The left panel depicts changes in output of the Louvain group clustering algorithm Blondel et al. (2008) as the percentage of recursively permuted edges increase. The x-axis depicts α. The y-axis depics the number of vertices that are clustered together when compared to the Louvain output of the original graph. The plot shows that 80% of the clustering results of the original graph are retained even when the graph is completely comprised of recursively selected synthetic edges. The right panel depicts the algorithm's tendency to increase graph transitivity as a larger proportion of edges become synthetic. The x-axis depicts graph transitivity and the y-axis is the same as described in the left panel.

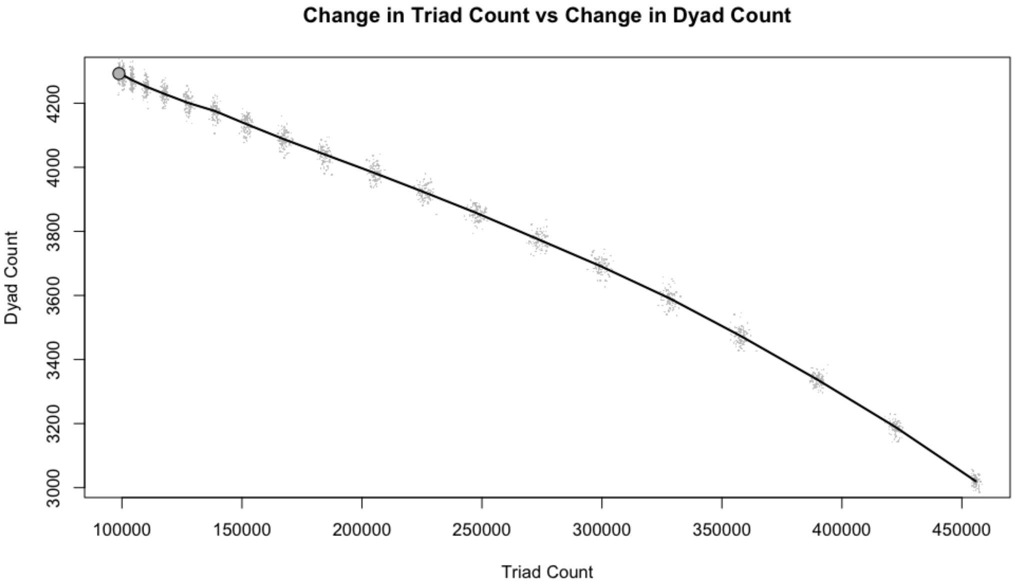


Figure C.2: depicts the results of 100 replicates at 20 different values of α. The x-axis depict triad count while the y-axis depicts dyad count. The large, gray circle depicts $\alpha = 0$, the original graph. As can be seen recursive stochastic block modeling tends to generate more triads and fewer dyads that the original graph, which could be the reason for changes in clustering output as α increases.

use the same original graph and add or remove α edges. In this experiment we select 40 values of α ranging from $-.5$ to $.5$. We again generate 100 replicates for each α value and record the transitivity, dyad count, and triad count, we will also measure the change in clustering output. Figure C.4 depicts the results of 100 replicates at 40 different values for α. The left panel depicts changes in output of the Louvain group clustering algorithm Blondel et al. (2008) as the percentage of recursively added edges increases. The x-axis depicts α, the percentage of edges added to the original graph. The y-axis depics the number of vertices that are clustered together when compared to the Louvain output of the original graph. The right panel depicts levels of α where edges are removed. The the plots show, the algorithms alters community structure most when removing edges. This result implies that any large subgraph could have its modularity increased while maintaining graph structure. Therefore, a graphs sub-structure could be altered in a manner where the change is known. This enables a researcher to detect known changes from a real world graph with unknown ground truth. Although further research is needed to compare algorithms, and to understand what drives the change in each algorithm's changes in performance as synthetic edges are added, this method offers a step towards ground truth evaluation of real world graphs at scale.

C.5 Conclusion

In this work we have presented modularity-based recursive stochastic graph permutation as a means to alter the community structure of real world graphs in a way that preserves the graph's topological qualities. We have shown that this algorithm retains a great deal of the community

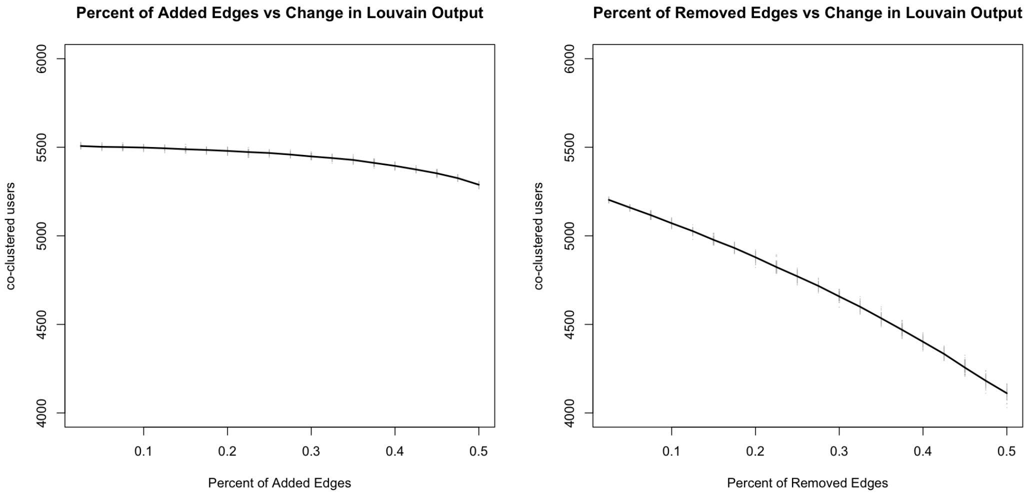


Figure C.3: depicts the results of 100 replicates at 40 different values for α. The left panel depicts changes in output of the Louvain group clustering algorithm Blondel et al. (2008) as the percentage of recursively added edges increases. The x-axis depicts α, the percentage of edges added to the original graph. The y-axis depics the number of vertices that are clustered together when compared to the Louvain output of the original graph. The right panel depicts levels of α where edges are removed. The the plots show, the algorithms alters community structure most when removing edges.

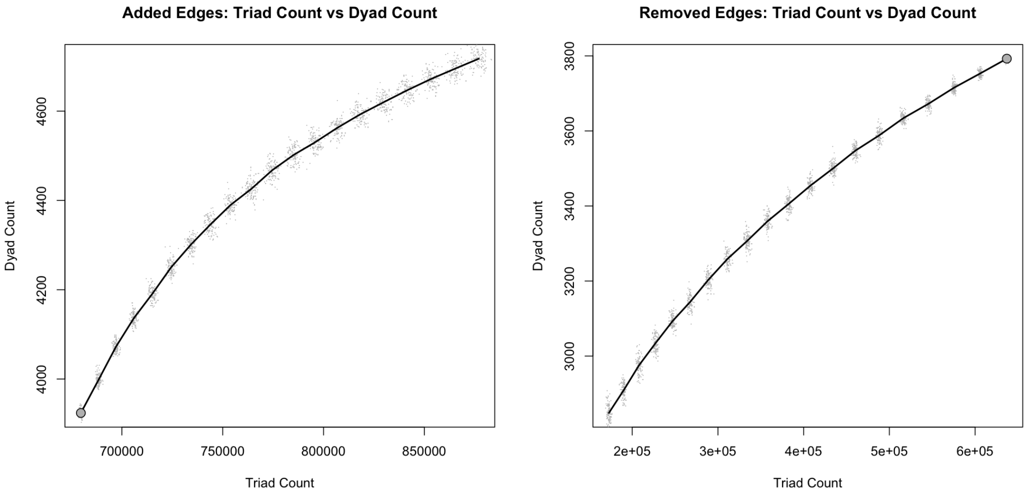


Figure C.4: depicts the results of 100 replicates at 20 different values of α in both panels. The left panel depicts $\alpha \in (0, .5)$ and the right panel depicts $\alpha \in (-.5, 0)$. The x-axes depict triad count while the y-axes depicts dyad count. The large, gray circle depicts $\alpha = 0$, the original graph.

structure found in the real world graph when replacing all of the graphs edges with synthetically generated edges. Furthermore, we show the algorithms ability to maintain structure while adding and subtracting edges to specific sub-graphs. This provides researchers a means to alter real world graphs in order to evaluate community detection algorithms. Although the results presented are encouraging, more study is needed to adjust for the algorithm's inability to retain dyadic and triadic closure. A formal method to compare detection algorithms is needed as well. Finally, additional case studies are needed to see how well this method generalizes to other real world graphs. However, this method represents a step forward in the evaluation of community detection algorithnms.

Bibliography

The Twitter Rules. 3, 3.4, 4.4

Abokhodair, N., Yoo, D., and McDonald, D. W. (2015). Dissecting a Social Botnet: Growth, Content and Influence in Twitter. pages 839–851. ACM Press. 2.3, 4.1, 6.1, 6.2

Al-khateeb, S. and Agarwal, N. (2015). Examining botnet behaviors for propaganda dissemination: A case study of ISIL's beheading videos-based propaganda. In *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pages 51–57. 6.1, 6.2, 7

Allcott, H. and Gentzkow, M. (2017). Social media and fake news in the 2016 election. Technical report, National Bureau of Economic Research. 7

Balakrishnan, A. Twitter suspends hundreds of thousands of accounts for promoting terrorism. 3.4

Barabsi, A.-L. Scale-free networks: A decade and beyond. 325(5939):412–413. C.2

Barrett, C. L., Beckman, R. J., Khan, M., Anil Kumar, V. S., Marathe, M. V., Stretz, P. E., Dutta, T., and Lewis, B. Generation and analysis of large synthetic social contact networks. In *Winter Simulation Conference*, WSC '09, pages 1003–1014. Winter Simulation Conference. C.2

Bartlett, J. (2016). Orwell versus the Terrorists: Crypto Wars and the Future of Surveillance. 3.6

Benigni, M. PLOS one data - accompanies "the broad reach of online extremism: Understanding the ISIS supporting community on twitter". DOI:10.6084/m9.figshare.3166798.v2. 3

Benigni, M. (2017). Algorithms for online extremist community detection. 3

Benigni, M. and Carley, K. M. (2016). From Tweets to Intelligence: Understanding the Islamic Jihad Supporting Community on Twitter. In *Social Computing, Behavioral-Cultural Modeling, and Prediction*, page to appear. Springer. 5.1, 5.2

Benigni, Matthew, Joseph, Kenneth, and Carley, Kathleen. The Broad Reach of Online Extremism: Uncovering the ISIS Supporting Network on Twitter. *Submitted to Plos One*. 2.2, 2.2.1, 2.2.2, 5.1, 5.2, 5.3, 5.3.2

Benkler, Y., Faris, R., Roberts, H., and Zuckerman, E. Study: Breitbart-led right-wing media ecosystem altered broader media agenda. *Columbia Journalism Review*. 1, 4.1, 7

Benner, K. (2016). Twitter Suspends 235,000 More Accounts Over Extremism. *The New York Times*. 4.4

Berger, J. and Morgan, J. (2015a). The isis twitter census: Defining and describing the population of isis supporters on twitter. *The Brookings Project on US Relations with the Islamic World*, 3(20). 4.1, 5.3.2, 5.5, 6.1, 6.5.2

Berger, J. M. (2014). How ISIS Games Twitter. *The Atlantic*. 1, 2.3

Berger, J. M. and Morgan, J. Defining and describing the population of ISIS supporters on

Twitter. 1, 7

Berger, J. M. and Morgan, J. (2015b). Defining and describing the population of ISIS supporters on Twitter. 1, 3, 3.3.1, 3.3.2, 4.1, 4.4, A

Berger, JM. Tailored Online Interventions: The Islamic States Recruitment Strategy. *Combating Terrorism Center Sentinel*. 1, 2.1, 2.2, 3, 3.4, 5.5

Berzak, Y., Huang, Y., Barbu, A., Korhonen, A., and Katz, B. (2016). Anchoring and Agreement in Syntactic Annotations. *arXiv preprint arXiv:1605.04481*. 3.4

Bessi, A. and Ferrara, E. (2016). Social bots distort the 2016 u.s. presidential election online discussion. *First Monday*, 21(11). 4.3.2, 6.1, 6.4.1, 6.5.2

Binkiewicz, N., Vogelstein, J. T., and Rohe, K. (2014). Covariate Assisted Spectral Clustering. *arXiv preprint arXiv:1411.2158*. 2.2.1, 3.1

Blodgett, S. L., Green, L., and O'Connor, B. (2016). Demographic dialectal variation in social media: A case study of African-American English. *EMNLP'16*. 3.4

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008. arXiv: 0803.0476. (document), 3.1, 3.3.2, 3.2, 4.2, 4.3.1, 4, 4.4, 4.5, 5.3.2, 5.4, C.1, C.2, C.3, C.4, C.4.1, C.1, C.4.2, C.3

Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., and Hwang, D.-U. (2006). Complex networks: Structure and dynamics. *Physics reports*, 424(4):175–308. 3.1, 4.2, C.1, C.2

Bonacich, P. (1987). Power and centrality: A family of measures. *American journal of sociology*, 92(5):1170–1182. 6.5.1

Bonacich, P. (2007). Some unique properties of eigenvector centrality. *Social networks*, 29(4):555–564. 6.4.1

Boshmaf, Y., Muslukhov, I., Beznosov, K., and Ripeanu, M. (2011). The socialbot network: When bots socialize for fame and money. In *Proceedings of the 27th Annual Computer Security Applications Conference*, ACSAC '11, pages 93–102. ACM. 6.2

Boshmaf, Y., Muslukhov, I., Beznosov, K., and Ripeanu, M. (2012). Key challenges in defending against malicious socialbots. In *Presented as part of the 5th USENIX Workshop on Large-Scale Exploits and Emergent Threats*. 6.6

Boshmaf, Y., Muslukhov, I., Beznosov, K., and Ripeanu, M. (2013). Design and analysis of a social botnet. *Computer Networks*, 57(2):556–578. 6.2, 6.5.1

Brin, S. and Page, L. (2012). Reprint of: The anatomy of a large-scale hypertextual web search engine. *Computer networks*, 56(18):3825–3833. 6.4.1

Calamur, K. (2016). Twitters New ISIS Policy. *The Atlantic*. 3

Callimachi, R. (2015). ISIS and the Lonely Young American. *The New York Times*. 5.5

Carley, K. M. (2006). A Dynamic Network Approach to the Assessment of Terrorist Groups and the Impact of Alternative Courses of Action. Technical report. 2.2.1, 3.1, 5.4

Carley, K. M., Dombroski, M., Tsvetovat, M., Reminga, J., Kamneva, N., and others (2003). Destabilizing dynamic covert networks. In *Proceedings of the 8th international command and control research and technology symposium*. 3.1

Carley, K. M., Reminga, J., and Kamneva, N. (1998). Destabilizing terrorist networks. *Institute* *for Software Research*, page 45. 3.1

Carter, J. A., Maher, S., and Neumann, P. R. (2014). #Greenbirds Measuring Importance and Influence in Syrian Foreign Fighter Networks. *International Centre for the Study of Radicalization Report*. 2.2.2, 3.3.1, 4.4, 6.3.1, A.1

Chang, H.-C. A new perspective on twitter hashtag use: Diffusion of innovation theory. 47(1):1–4. 5.1

Chen, J. and Saad, Y. (2012). Dense subgraph extraction with application to community detection. *IEEE Transactions on Knowledge and Data Engineering*, 24(7):1216–1230. 1, 4.2, 4.3.2, 4.3.2, 4.3.2, 4, 4.4, 6.4.1, C.2

Chiu, C.-M., Hsu, M.-H., and Wang, E. T. (2006). Understanding knowledge sharing in virtual communities: An integration of social capital and social cognitive theories. *Decision support systems*, 42(3):1872–1888. 3.2.2, 3.5

Chu, Z., Gianvecchio, S., Wang, H., and Jajodia, S. (2010). Who is tweeting on twitter: human, bot, or cyborg? In *Proceedings of the 26th annual computer security applications conference*, pages 21–30. ACM. 1, 6.1, 6.2, 6.4.2

Criminisi, A., Shotton, J., Konukoglu, E., et al. (2012). Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends® in Computer Graphics and Vision*, 7(2–3):81–227. 7

Danezis, G. and Mittal, P. SybilInfer: Detecting sybil nodes using social networks. In *NDSS*. San Diego, CA. C.2

DeMasi, O., Mason, D., and Ma, J. (2016). Understanding Communities via Hashtag Engagement: A Clustering Based Approach. In *Tenth International AAAI Conference on Web and Social Media*. 5.3

Developers, T. (2017). Twitter developers. 6.4.2

Dewey, T., Kaden, J., Marks, M., Matsushima, S., and Zhu, B. (2012). The impact of social media on social unrest in the Arab Spring. *International Policy Program*. 5.1

Dhillon, I. S. (2001a). Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 269–274. ACM. 1, 3.2.2, 5.3

Dhillon, I. S. (2001b). Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 269–274. ACM. 4.3.1

Dickey, C. (2014). French Jihadi Mehdi Nemmouche Is the Shape of Terror to Come. 3.5

Diesner, J. and Carley, K. M. (2004). Using network text analysis to detect the organizational structure of covert networks. In *Proceedings of the North American Association for Computational Social and Organizational Science (NAACSOS) Conference*. 3.1

Diuk, N. (2014). Euromaidan: Ukraine's self-organizing revolution. *World Affairs*, 176(6):9. 1

Dozier, K. (2016). Anti-ISIS-Propaganda Czars Ninja War Plan: We Were Never Here. 2.1, 5.1

Eaton, E. and Mansbach, R. (2012). A Spin-Glass Model for Semi-Supervised Community Detection. In *AAAI*. Citeseer. 3.1

Ediger, D., Jiang, K., Riedy, J., Bader, D. A., and Corley, C. (2010). Massive social network analysis: Mining twitter for social good. In *2010 39th International Conference on Parallel Processing*, pages 583–593. IEEE. 6.5.2

Engesser, S., Ernst, N., Esser, F., and Büchel, F. (2016). Populism and social media: how politicians spread a fragmented ideology. *Information, Communication & Society*, pages 1–18. 7

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231. 7

Eubank, S., Guclu, H., Anil Kumar, V. S., Marathe, M. V., Srinivasan, A., Toroczkai, Z., and Wang, N. Modelling disease outbreaks in realistic urban social networks. 429(6988):180–184. C.2

Faust, K. and Wasserman, S. Blockmodels: Interpretation and evaluation. 14(1):5–61. C.3

Ferrara, E. (2015). Manipulation and abuse on social media by emilio ferrara with ching-man au yeung as coordinator. page 4. 6.1

Ferrara, E., Varol, O., Davis, C., Menczer, F., and Flammini, A. (2014). The rise of social bots. *arXiv preprint arXiv:1407.5225*. 2.3, 4.1

Ferrara, E., Varol, O., Menczer, F., and Flammini, A. (2016a). Detection of promoted social media campaigns. In *Tenth International AAAI Conference on Web and Social Media*. 1, 4.1, 6.5.1

Ferrara, E., Wang, W.-Q., Varol, O., Flammini, A., and Galstyan, A. (2016b). Predicting online extremism, content adopters, and interaction reciprocity. In *International Conference on Social Informatics*, pages 22–39. Springer. 7

Forelle, M., Howard, P., Monroy-Hernndez, A., and Savage, S. (2015). Political Bots and the Manipulation of Public Opinion in Venezuela. *arXiv:1507.07109 [physics]*. arXiv: 1507.07109. 2.3, 4.1

Fortunato, S. Community detection in graphs. 486(3):75–174. C.1

Fortunato, S. (2010). Community detection in graphs. *Physics reports*, 486(3):75–174. 4.1, 4.2, 4.3.1, 5.3, 5.4, C.2

Gerbaudo, P. (2012). *Tweets and the streets: Social media and contemporary activism*. Pluto Press. 7

Gilbert, E. and Karahalios, K. (2009). Predicting Tie Strength with Social Media. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 211–220, New York, NY, USA. ACM. 3.2.2, 3.5

Girvan, M. and Newman, M. E. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826. 1, 3.1, 4.2, 4.4, C.1, C.2

Gladstone, R. (2015a). Activist Links More Than 26,000 Twitter Accounts to ISIS. *The New York Times*. 3

Gladstone, R. (2015b). Behind a Veil of Anonymity, Online Vigilantes Battle the Islamic State. *The New York Times*. 3

Gladstone, R. (2015c). Twitter Says It Suspended 10,000 ISIS-Linked Accounts in One Day.

The New York Times. (document), 3.3.2, 3.2

Glasgow, K. and Fink, C. Hashtag lifespan and social networks during the london riots. In Greenberg, A. M., Kennedy, W. G., and Bos, N. D., editors, *Social Computing, Behavioral-Cultural Modeling and Prediction*, number 7812 in Lecture Notes in Computer Science, pages 311–320. Springer Berlin Heidelberg. DOI: 10.1007/978-3-642-37210-0_34. 5.1

Goodman, L. A. (1961). Snowball Sampling. *The Annals of Mathematical Statistics*, 32(1):148–170. 1, 3.3.1, 4.3, 5.2, 6.3, A

Hamilton, W. L., Clark, K., Leskovec, J., and Jurafsky, D. (2016). Inducing Domain-Specific Sentiment Lexicons from Unlabeled Corpora. *arXiv preprint arXiv:1606.02820*. 5.3

Harman, J. (2015). Disrupting the Intelligence Community. *Foreign Affairs*, (March/April 2015). 3.1

Herrick, D. The social side of cyber power? social media and cyber operations. In *Cyber Conflict (CyCon), 2016 8th International Conference on*, pages 99–111. NATO CCD COE. 5.1

Hoffman, F. G. (2009). Hybrid warfare and challenges. Technical report, DTIC Document. 3.6

Howard, P. N., Duffy, A., Freelon, D., Hussain, M. M., Mari, W., and Maziad, M. (2011). Opening closed regimes: what was the role of social media during the arab spring? 7

Howard, P. N. and Parks, M. R. (2012). Social media and political change: Capacity, constraint, and consequence. *Journal of Communication*, 62(2):359–362. 5.1

Hussain, M. M. and Howard, P. N. (2013). What best explains successful protest cascades? ICTs and the fuzzy causes of the Arab Spring. *International Studies Review*, 15(1):48–66. 5.1

Jensen, D. D. N. Lennart meri and the 'new normal' | huffington post. 5.4

Johnson, N. F., Zheng, M., Vorobyeva, Y., Gabriel, A., Qi, H., Velasquez, N., Manrique, P., Johnson, D., Restrepo, E., Song, C., and Wuchty, S. New online ecology of adversarial aggregates: ISIS and beyond. 352(6292):1459–1463. 5.1

Joseph, K. and Carley, K. M. (2015). Culture, Networks, Twitter and foursquare: Testing a Model of Cultural Conversion with Social Media Data. 3.1

Joseph, K. and Carley, K. M. (2016). Relating semantic similarity and semantic association to how humans label other people. *NLP+ CSS 2016*, page 1. 3.4

Jurgenson, N. (2012). When atoms meet bits: Social media, the mobile web and augmented revolution. *Future Internet*, 4(1):83–91. 7

Juris, J. S. Reflections on #occupy everywhere: Social media, public space, and emerging logics of aggregation. 39(2):259–279. 5.1

Kelman, H. C. (1958). Compliance, identification, and internalization: Three processes of attitude change. *Journal of conflict resolution*, pages 51–60. 6.1

Kitsak, M., Gallos, L. K., Havlin, S., Liljeros, F., Muchnik, L., Stanley, H. E., and Makse, H. A. (2010). Identification of influential spreaders in complex networks. *Nature physics*, 6(11):888–893. 6.5.1

Koschade, S. (2006). A social network analysis of Jemaah Islamiyah: The applications to counterterrorism and intelligence. *Studies in Conflict & Terrorism*, 29(6):559–575. 3.1

Kossinets, G. and Watts, D. J. (2006). Empirical analysis of an evolving social network. *science*, 311(5757):88–90. 6.4.1

Krebs, V. (2002a). Uncloaking terrorist networks. *First Monday*, 7(4). 2.2.1, 3.1

Krebs, V. E. (2002b). Mapping networks of terrorist cells. *Connections*, 24(3):43–52. 3.1

Kwak, H., Lee, C., Park, H., and Moon, S. (2010). What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, pages 591–600. ACM. 4.1, 6.4.1, 6.5.1

Lahoud, N., Milton, D., Price, B., and others (2014). The Group That Calls Itself a State: Understanding the Evolution and Challenges of the Islamic State. Technical report, DTIC Document. 1

Lake, D. A. (2002). Rational extremism: Understanding terrorism in the twenty-first century. *Dialogue IO*, 1(01):15–29. 3, 4.1, 4.5

Latora, V. and Marchiori, M. (2004). How the science of complex networks can help developing strategies against terrorism. *Chaos, solitons & fractals*, 20(1):69–75. 3.1

Liaw, A. and Wiener, M. (2002). Classification and regression by randomforest. *R news*, 2(3):18–22. 3.3.2

Lin, P.-C. and Huang, P.-M. (2013). A study of effective features for detecting long-surviving twitter spam accounts. In *Advanced Communication Technology (ICACT), 2013 15th International Conference on*, pages 841–846. IEEE. 6.4.2

Liu, Y., Tang, M., Zhou, T., and Do, Y. (2014). Core-like groups result in invalidation of identifying super-spreader by k-shell decomposition. *arXiv preprint arXiv:1409.5187*. 6.5.1

Lloyd, S. (1982). Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137. 5.3.1

Loader, B. D. and Mercea, D. Networking democracy? 14(6):757–769. 5.1

Logan, M. H. (2014). Lone Wolf Killers: A Perspective on Overvalued Ideas. *Violence and Gender*, 1(4):159–160. 3

Lorrain, F. and White, H. C. Structural equivalence of individuals in social networks. 1(1):49–80. C.2, C.3

Lotan, G., Graeff, E., Ananny, M., Gaffney, D., Pearce, I., and Boyd, D. (2011). The revolutions were tweeted: Information flows during the 2011 Tunisian and Egyptian revolutions. *International Journal of Communication*, 5:1375–1405. 5.1

MacQueen, J. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. University of California Press. 5.3

Mangold, L. (2016). should i stay or should i go: Clash of opinions in the brexit twitter debate. *Computing*. 1, 4.1

McCaughey, M. and Ayers, M. D. (2013). *Cyberactivism: Online activism in theory and practice*. Routledge. 4.1, 4.5

Miller, B. A., Beard, M. S., and Bliss, N. T. (2011a). Eigenspace analysis for threat detection in social networks. In *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*, pages 1–7. IEEE. 3.1, 3.1, 3.2.1, 4.2, C.2

Miller, S., Mameli, P., Kleinig, J., Salane, D., and Schwartz, A. (2011b). *Security and privacy: global standards for ethical identity management in contemporary liberal democratic states*.

ANU Press. 3.6

Mislove, A., Marcon, M., Gummadi, K. P., Druschel, P., and Bhattacharjee, B. (2007). Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 29–42. ACM. 3.2.2, 3.5

Mossel, E., Neeman, J., and Sly, A. Stochastic block models and reconstruction. C.2

Mucha, P. J., Richardson, T., Macon, K., Porter, M. A., and Onnela, J.-P. (2010). Community structure in time-dependent, multiscale, and multiplex networks. *science*, 328(5980):876–878. 4.2, 4.3.1

Mungiu-Pippidi, A. and Munteanu, I. (2009). Moldova's twitter revolution. *Journal of Democracy*, 20(3):136–142. 4.1

Nanabhay, M. and Farmanfarmaian, R. (2011). From spectacle to spectacular: How physical space, social media and mainstream broadcast amplified the public sphere in Egypt's Revolution'. *The Journal of North African Studies*, 16(4):573–603. 5.1

Newman, M. E. (2006). Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582. 3.1

Newman, M. E. J. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113. 4.2, C.2

of Staff, J. C. (2014). Joint pub 3-13 joint doctrine for information operations. *United States Department of Defense*, pages 1–35. 3.6

of State, U. D. (2016). *Global Engagement Center*. 3.6

Pablo Barbera. Tweeting the revolution: Social media use and the #euromaidan protests | huffington post. 5.1

Papadopoulos, S., Kompatsiaris, Y., Vakali, A., and Spyridonos, P. (2011). Community detection in Social Media. *Data Mining and Knowledge Discovery*, 24(3):515–554. 3.1, 3.1, C.1

Papadopoulos, S., Kompatsiaris, Y., Vakali, A., and Spyridonos, P. (2012). Community detection in social media. *Data Mining and Knowledge Discovery*, 24(3):515–554. 3.3.1, 4.1, 4.2, 5.3, C.2

Peel, L., Larremore, D. B., and Clauset, A. The ground truth about metadata and community detection in networks. C.1

Peel, L., Larremore, D. B., and Clauset, A. (2016). The ground truth about metadata and community detection in networks. *arXiv preprint arXiv:1608.05878*. 4.4, 5.3.2

Pei, S., Muchnik, L., Andrade Jr, J. S., Zheng, Z., and Makse, H. A. (2014). Searching for superspreaders of information in real-world social media. *arXiv preprint arXiv:1405.1790*. 4.1, 4.2

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, 12:1532–1543. bibtex: pennington_glove:_2014. 5.3

Perry, T. (2015). Syria ceasefire ends, fighting resumes. *Reuters*. 2.2.3

Poe, T. ISIS getting social media megaphone (Opinion) - CNN.com. 3

Powers, D. M. (2011). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. (document), 3.4, 3.1

Ratkiewicz, J., Conover, M., Meiss, M., Goncalves, B., Patil, S., Flammini, A., and Menczer, F. Truthy: mapping the spread of astroturf in microblog streams. In *Proceedings of the 20th international conference companion on World wide web*, pages 249–252. ACM. 4.1, 6.1

Ratkiewicz, J., Conover, M., Meiss, M., Gonalves, B., Flammini, A., and Menczer, F. Detecting and tracking political abuse in social media. *ICWSM*, 11:297–304. 1, 4.1, 6.1

Ressler, S. (2006). Social network analysis as an approach to combat terrorism: Past, present, and future research. *Homeland Security Affairs*, 2(2):1–10. 3.1

Ross, B., Meek, J. G., and Ferran, L. (2015). Twitter escalates isis battle: 2,000 accounts suspended. 3

Roxburgh, Gordon. Ukraine wins the 2016 eurovision song contest | news | eurovision song contest. 5.4

Ruths, D. and Pfeffer, J. (2014). Social media for large studies of behavior. *Science*, 346(6213):1063–1064. 6.1

Sajana, T., Rani, C. S., and Narayana, K. (2016). A survey on clustering techniques for big data mining. *Indian Journal of Science and Technology*, 9(3). 7

Sakaki, T., Okazaki, M., and Matsuo, Y. Earthquake shakes twitter users: Real-time event detection by social sensors. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 851–860. ACM. 5.1

Settles, B. (2009). Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison. 5.5

Settles, B. (2010a). Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11. 3.7

Settles, B. (2010b). Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11. 4.6, 7

Shamanska, A. (2016). Hackers in ukraine deface separatist websites to mark victory day. [Online; posted 23-October-2016]. 5.4

Smith, S. T., Senne, K. D., Philips, S., Kao, E. K., and Bernstein, G. (2013). Covert Network Detection. *Lincoln Laboratory Journal*, 20(1). 3.1

Starbird, K. and Palen, L. (2012). (How) will the revolution be retweeted?: information diffusion and the 2011 Egyptian uprising. In *Proceedings of the acm 2012 conference on computer supported cooperative work*, pages 7–16. ACM. 5.1

Starr, B. (2015). U.S. officials say 6,000 ISIS fighters killed in battles - CNNPolitics.com. 3

Statista (2016a). Facebook users worldwide 2016. 1, 6.1

Statista (2016b). Twitter: number of active users 2010-2016. 1, 6.1

Steinbach, M., Karypis, G., Kumar, V., and others. A comparison of document clustering techniques. In *KDD workshop on text mining*, volume 400, pages 525–526. Boston. 5.3.2

Steinfield, C., Ellison, N. B., and Lampe, C. (2008). Social capital, self-esteem, and use of online social network sites: A longitudinal analysis. *Journal of Applied Developmental Psychology*, 29(6):434–445. 3.3.1

Stern, Jessica and Berger, JM (2015). ISIS and the Foreign-Fighter Phenomenon. *The Atlantic*. 3

Strategy, I. and Group, I. S. Abdulgani Pagao's Capture and the Rising ISIS Threat to the Philippines. 3.5

Subrahmanian, V. S., Azaria, A., Durst, S., Kagan, V., Galstyan, A., Lerman, K., Zhu, L., Ferrara, E., Flammini, A., Menczer, F., Waltzman, R., Stevens, A., Dekhtyar, A., Gao, S., Hogg, T., Kooti, F., Liu, Y., Varol, O., Shiralkar, P., Vydiswaran, V., Mei, Q., and Huang, T. (2016). The DARPA Twitter Bot Challenge. *arXiv:1601.05140 [physics]*. arXiv: 1601.05140. 2.3

Sun, Y., Tang, J., Han, J., Gupta, M., and Zhao, B. (2010). Community evolution detection in dynamic heterogeneous information networks. In *Proceedings of the Eighth Workshop on Mining and Learning with Graphs*, pages 137–146. ACM. 4.1, 4.2

Szostek, J. The media battles of ukraines EuroMaidan. 4.1, 4.5, 5.1, 7, A.2

Tang, L. and Liu, H. (2010). Community Detection and Mining in Social Media. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 2(1):1–137. A

Tang, L. and Liu, H. (2011). Leveraging social media networks for classification. *Data Mining and Knowledge Discovery*, 23(3):447–478. 2.2.1, 3.1, 3.2.2, 3.3.2, 3.4, 4.2

Tang, L., Wang, X., and Liu, H. (2009). Uncoverning groups via heterogeneous interaction analysis. In *Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on*, pages 503–512. IEEE. 3.1, 3.2.2, 3.4, 5.3

Top, N. M. (2009). Counterterrorism?s new tool:?metanetwork?analysis. 3.1

Traud, A. L., Mucha, P. J., and Porter, M. A. (2012). Social structure of facebook networks. *Physica A: Statistical Mechanics and its Applications*, 391(16):4165–4180. 4.3.1

Tufekci, Z. (2014). Big Questions for Social Media Big Data: Representativeness, Validity and Other Methodological Pitfalls. In *ICWSM '14: Proceedings of the 8th International AAAI Conference on Weblogs and Social Media*. 5.1

Tunisias, A. a.-S. and Game, L. (2013). Dawa, hisba, and jihad. 2.3.1, 3.5

Twitter, R. *API*. C.1

Twitter, R. (2016). API. 3.3.1, 5.4, 5.5

UNIAN (2016). Rada adopts bill allowing lutsenko to be nominated prosecutor general. [Online; retrieved 20-October-2016]. 5.4

Veilleux-Lepage, Y. (2014). Retweeting the Caliphate: The Role of Soft-Sympathizers in the Islamic States Social Media Strategy. In *2014 6th International Terrorism and Transnational Crime Conference*. 3, 3.4, 4.1, 5.1

Veilleux-Lepage, Y. (2015). Paradigmatic Shifts in Jihadism in Cyberspace: The Emerging Role of Unaffiliated Sympathizers in the Islamic State&#39;s Social Media Strategy. 2.1, 2.2, 3, 3.3.1, 3.4, 4.1, 5.1, 5.2, A

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416. 3.2.2

Wagstaff, K., Cardie, C., Rogers, S., Schrdl, S., and others. Constrained k-means clustering with background knowledge. In *ICML*, volume 1, pages 577–584. 5.3.2

Wakabayashi, D. and Isaac, M. (2017). In Race Against Fake News, Google and Facebook Stroll to the Starting Line. *The New York Times*. 4.1

Walsh, P. F. and Miller, S. (2016). Rethinking Five Eyes Security Intelligence Collection Policies and Practice Post Snowden. *Intelligence and National Security*, 31(3):345–368. 3, 3.6

Wang, X., Tang, L., Gao, H., and Liu, H. (2010). Discovering overlapping groups in social media. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 569–578. IEEE. 3.1, 3.1, 3.2.2, 4.2, C.1, C.2

Wasserman, S. and Faust, K. (1994). *Social Network Analysis: Methods and Applications*. Cambridge University Press. 5.5, 6.4.1

Weber, I., Garimella, V. R. K., and Teka, A. Political hashtag trends. In Serdyukov, P., Braslavski, P., Kuznetsov, S. O., Kamps, J., Rger, S., Agichtein, E., Segalovich, I., and Yilmaz, E., editors, *Advances in Information Retrieval*, number 7814 in Lecture Notes in Computer Science, pages 857–860. Springer Berlin Heidelberg. DOI: 10.1007/978-3-642-36973-5_102. 5.1

Wei, W., Joseph, K., Liu, H., and Carley, K. M. The fragility of twitter social networks against suspended users. In *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 9–16. IEEE. 5.1

Wikipedia (2015). *Charlie Hebdo* shooting. Page Version ID: 660950143. 3

Wolfsfeld, G., Segev, E., and Sheafer, T. Social media and the arab spring politics comes first. 18(2):115–137. 5.1, 7

Wood, P. IS conflict: Counting the civilian cost of US-led air strikes. 5.3.2

Woolley, S. C. (2016). Automating power: Social bot interference in global politics. *First Monday*, 21(4). 6.1

Yan, H. (2015). ISIS claims responsibility for Garland, Texas, shooting - CNN.com. 3

Yang, J., McAuley, J., and Leskovec, J. (2013). Community detection in networks with node attributes. In *Data Mining (ICDM), 2013 IEEE 13th international conference on*, pages 1151–1156. IEEE. 3.7

Zachary, W. W. (1977). An information flow model for conflict and fission in small groups. *Journal of anthropological research*, 33(4):452–473. 4.4

Zhang, J., Zhang, R., Zhang, Y., and Yan, G. (2013). On the impact of social botnets for spam distribution and digital-influence manipulation. In *2013 IEEE Conference on Communications and Network Security (CNS)*, pages 46–54. 6.1

Zhang, S., Wang, R.-S., and Zhang, X.-S. (2007). Identification of overlapping community structure in complex networks using fuzzy c-means clustering. *Physica A: Statistical Mechanics and its Applications*, 374(1):483–490. 4.2

Zweig, K. A. and Kaufmann, M. (2011). A systematic approach to the one-mode projection of bipartite graphs. *Social Network Analysis and Mining*, 1(3):187–218. 5.3