UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL

DISEÑO, DESARROLLO Y EVALUACIÓN DE UN ALGORITMO PARA DETECTAR SUB-COMUNIDADES TRASLAPADAS USANDO ANÁLISIS DE REDES SOCIALES Y MINERÍA DE DATOS.

TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN GESTIÓN DE OPERACIONES

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL INDUSTRIAL

RICARDO LUIS MUÑOZ CANCINO

PROFESOR GUÍA:
SEBASTÍAN ALEJANDRO RÍOS PÉREZ

MIEMBROS DE LA COMISIÓN:
FELIPE IGNACIO AGUILERA VALENZUELA
MARCELO GABRIEL MENDOZA ROCHA
CAROLINA ALEJANDRA BONACIC CASTRO

Este trabajo ha sido parcialmente financiado por CONICYT, Beca FOLIO: 22121473

SANTIAGO DE CHILE
ENERO 2013

RESUMEN DE LA TESIS
PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL INDUSTRIAL
POR: RICARDO MUÑOZ C.
FECHA: 09/01/2013
PROF. GUÍA: SEBASTIÁN A. RÍOS

DISEÑO, DESARROLLO Y EVALUACIÓN DE UN ALGORITMO PARA DETECTAR SUB-COMUNIDADES TRASLAPADAS USANDO ANÁLISIS DE REDES SOCIALES Y MINERÍA DE DATOS

Los sitios de redes sociales virtuales han tenido un enorme crecimiento en la última década. Su principal objetivo es facilitar la creación de vínculos entre personas que, por ejemplo, comparten intereses, actividades, conocimientos, o conexiones en la vida real. La interacción entre los usuarios genera una comunidad en la red social.

Existen varios tipos de comunidades, se distinguen las comunidades de interés y práctica. Una comunidad de interés es un grupo de personas interesadas en compartir y discutir un tema de interés particular. En cambio, en una comunidad de práctica las personas comparten una preocupación o pasión por algo que ellos hacen y aprenden cómo hacerlo mejor. Si las interacciones se realizan por internet, se les llama comunidades virtuales (VCoP/VCoI por sus siglas en inglés). Es común que los miembros compartan solo con algunos usuarios formando así subcomunidades, pudiendo pertenecer a más de una. Identificar estas subestructuras es necesario, pues allí se generan las interacciones para la creación y desarrollo del conocimiento de la comunidad. Se han diseñado muchos algoritmos para detectar subcomunidades. Sin embargo, la mayoría de ellos detecta subcomunidades disjuntas y además, no consideran el contenido generado por los miembros de la comunidad. El objetivo principal de este trabajo es diseñar, desarrollar y evaluar un algoritmo para detectar subcomunidades traslapadas mediante el uso de análisis de redes sociales (SNA) y Text Mining.

Para ello se utiliza la metodología SNA-KDD propuesta por Ríos et al. [79] que combina Knowledge Discovery in Databases (KDD) y SNA. Ésta fue aplicada sobre dos comunidades virtuales, Plexilandia (VCoP) y The Dark Web Portal (VCoI). En la etapa de KDD se efectuó el preprocesamiento de los posts de los usuarios, para luego aplicar Latent Dirichlet Allocation (LDA), que permite describir cada post en términos de tópicos. En la etapa SNA se construyeron redes filtradas con la información obtenida en la etapa anterior. A continuación se utilizaron dos algoritmos desarrollados en esta tesis, SLTA y TPA, para encontrar subcomunidades traslapadas.

Los resultados muestran que SLTA logra un desempeño, en promedio, un 5% superior que el mejor algoritmo existente cuando es aplicado sobre una VCoP. Además, se encontró que la calidad de la estructura de sub-comunidades detectadas aumenta, en promedio, un 64% cuando el filtro semántico es aumentado. Con respecto a TPA, este algoritmo logra, en promedio, una medida de modularidad de 0.33 mientras que el mejor algoritmo existente 0.043 cuando es aplicado sobre una VCoI. Además la aplicación conjunta de nuestros algoritmos parece mostrar una forma de determinar el tipo de comunidad que se está analizando. Sin embargo, esto debe ser comprobado analizando más comunidades virtuales.

DESIGN, DEVELOPMENT AND EVALUATION OF AN OVERLAPPING COMMUNITY DETECTION ALGORITHM USING TEXT MINING AND SOCIAL NETWORK ANALYSIS.

The social networking sites have grown tremendously in the last decade. Its main objective is to facilitate the creation of links among people who, for example, share interests, activities, backgrounds, or connections in real life. The interaction among users generates a Community in the network itself.

There are several types of communities, among them are distinguished communities of interest and communities of practice. A community of interest is a group of people interested in sharing information and discussing a particular topic that interests them. Instead, a community of practice is a group of people who share a concern or a passion for something they do and learn how to do it better as they interact regularly. If interactions are conducted online, are called virtual communities of practice/interest (VCoP/VCoI). It is usual that members only share with some network's users building subcommunities, they may belong to several subcommunities. Identify these substructures is necessary because the interactions to create and to develop the knowledge of the community are generated there. Several algorithms have been designed to detect overlapping communities. However, most of the work has been done on disjoint community detection and also, doesn't include the content generated by community's members. The main objective of this work is to design, to develop and to evaluate an overlapping community detection algorithm using text mining and social network analysis (SNA).

To accomplish the main objective of this thesis, SNA-KDD methodology proposed by Ríos et al. [79] is used. This is an hybrid approach which combines Knowledge Discovery in Databases (KDD) and SNA. It was applied over two virtual communities, Plexilandia (VCoP) and The Dark Web Portal (VCoI). In the KDD step, users' posts were preprocessed. Then, Latent Dirichlet Allocation (LDA) was applied, it allows us to described each post in topic terms. In SNA step, filtered networks were built using the information obtained previously. Finally, two novel algorithms developed in this thesis, SLTA and TPA, were used to detect overlapping subcommunities.

Results shows that SLTA achieves, in average, a 5% higher performance than the best state of the art algorithm when it is applied over a VCoP. Also, it was found that the quality of the subcommunity structure detected is, in average, 64% higher when the semantic filter applied on networks is increased. With regard to TPA, this algorithms achieves, in average, a modularity measure of 0.33 while the best state of the art algorithm only achieves 0.043 when it is applied over a VCoI. Moreover, applying both algorithms over a virtual community would let us to detect the type of community analyzed (VCoP or VCoI). This hypothesis has to be validated applying the proposed methodology in this thesis over new virtual communities.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this chapter, a general background of this thesis' purpose is presented, followed by this thesis general and specific objectives. Then the methodology used for the development of the work is discussed. Finally, the structure of this work with a brief introduction for all chapters is presented.

1.1 The Overlapping Community Detection Problem

The recent explosion of social on-line networking has created a variety of social-media services with many different purposes: connecting with friends, sharing multimedia content, entertaining, blogging, bookmarking, etc. For example, the popular social network site facebook[1], which primary focus is creating and maintaining a network of friends or acquaintances, reports having more than 955 million active users as of July 2012 [14]; the photo-sharing program and social network Instagram[2] claimed to host 1 billion images in May 2012 [29]; and Twitter[3], the popular microblogging system, has recently revealed having over 500 million registered users (but just 140 million active users)[17], who send an average total of 144 million messages a day [16]. If we examine every minute of the day it's possible to see some interesting facts [16]:

- 684,478 pieces of content are shared on Facebook
- 100,000 tweets are sent on Twitter
- 2,000,000 search queries are made on Google
- 48 hours of video are uploaded to YouTube
- 47,000 apps are downloaded from the App Store
- 3,600 photos are shared on Instagram
- 3,125 photos are add on Flickr

[1] http://www.facebook.com

[2] http://instagram.com/

[3] http://www.twitter.com

- 571 new websites are created

There is a general request for methods and tools for analysing this wealth of social information:

"From retailing to counter-terrorism, the ability to analyze social connections is proving increasingly useful.".[33]

Understanding the viral spread of information in social media, modelling how information propagation relates to the underlying community structure, and identifying influential users, are some of the important challenges in social-network analysis. To address these issues, data mining has long being used to perform text mining based knowledge discovery process on web systems. However, only few years ago these techniques have being combined with Social Network Analysis (SNA) techniques to perform a better understanding of social structures.

SNA uses a graph representation to model people, organizations, etc. and its relations, interactions, among others. this graphs are commonly called networks. Networks are a natural representation to various types of complex system in: society, biology, linguistics, and other fields. One of the most interesting properties of human networks have is community structure; the existence of more densely connected/linked vertices groups which create a cohesive group of people. Communities often represent groups of individuals which may have common interests, motivations, tastes, preferences, among other in the real world [41].

Community structure is an important characteristic of social networks and has long been studied in sociology. The classic paper of Luce and Perry in 1949 -which introduced the term "clique" to graph theory -described a community as subsets of individuals every pair of whom are acquainted. The text of Scott [84] equates communities with objects such as cliques or other dense subgraphs. Another seminal 1974 paper, Breiger [21] develops a theory of communities in terms of affiliation networks, which in graph theoretic terms consist of using a bipartite graph with people on one side and communities on the other.

It is well understood that people in a social network are naturally characterized by multiple community memberships (i.e., the existence of overlap). For instance, a person usually has connections to several social groups like family, friends, and colleagues; a researcher may be active in several areas. Thus there is growing interest in finding communities that are allowed to overlap, as they do in most real-life social networks. Different motivations to investigate the overlapping community structure of a network exist. For example, it is possible to put into evidence interesting properties or hidden information about the network itself. Moreover, individuals that belong to a same community may share some similarities and possibly have common interests or are connected by a specific relationship in the real-world. These aspects arise a lot of commercial and scientific applications; in the first category we count, for example, marketing and competitive intelligence investigations and recommender systems. In fact, users belonging to a same community could share tastes or interests in similar products.

Many algorithms have been designed to discover networks' community structure. Random walks, spectral clustering, modularity maximization, differential equations, and statistical mechanics have been used previously. Most of these algorithms detect disjoint communities

[30, 36, 54, 61], which means that every community member belongs to a single community. These models do not consider that a person may have more than one interest. Thus, lately, a few methods have been designed to find overlapping communities. But most researchers have either emphasize to solve this problem on computing network's structural properties, or using graphical models for the community extraction process, where structural properties of networks are not considered. However, when end users are connected with each other by documents, posts or comments it is not possible to ignore underlying informations' semantics from these texts.

For this reason, the main contribution of this work, is a community finding strategy which considers both structural properties of posted messages and their content semantics. By using both traditional overlapping community detection algorithms and topic models, the interaction between members of a virtual community and their interests are gathered. This way, we make easier to perform the analysis phase of the knowledge discovery process.

Of course, our proposal was experimentally tested on real data from two virtual communities' datasets. To show we are capable to discover better quality information based on modularity. This data was selected according the classification proposed by Wenger [90], he has proposed the following types of communities: communities of interest, communities of purpose, and communities of practice. This Thesis is focused on the problem of the overlapping community detection on Virtual Communities, specifically we will focus on Communities of Practice (CoP) and Communities of Interest (CoI) which have been studied by many researchers like [52, 76]. According the underlying sociological theory, it's expected to have different results when a community detection algorithm is applied over a VCoP or a VCoI. Two recently-collected samples of Virtual Communities, Plexilandia and Dark Web Portal, are described and analyzed in detail. Plexilandia corresponds to a Virtual Community of practice (VCoP) [90] and The Dark Web corresponds to a Virtual Communities of Interests (VCoI) [52, 76].

1.2 Objectives

1.2.1 General Objective

The main objective of this thesis is to extend the proposed methodology in [28] and to design, to develop and to evaluate an algorithm, by using SNA and Text Mining, to detect overlapping sub-communities on Virtual Community Systems.

1.2.2 Specific Objectives

1. A research about the state of the art in overlapping community detection algorithms and methodologies. Understanding how they work, which is the best algorithm and their constraints.

2. To design and develop a new overlapping community detection algorithm that uses semantic information from texts on posts made by community's members.
3. To prove and evaluate our algorithm with the state of the art over real-world virtual communities.
4. To characterize the detected overlapping sub-communities using Text Mining and Association Rules.

1.3 Expected Results

1. A benchmark study of the state of the art overlapping community detection algorithms.
2. Use the state of the art research and benchmark study for at least one publication (conference or journal).
3. A new overlapping community detection algorithm which includes semantic information and that over perform other similar algorithms.
4. Algorithms implemented and added to benchmark framework.
5. Report of new algorithms' detection quality and performance.
6. A characterization of the detected overlapping sub communities using Text Mining and Association Rules.

1.4 Methodology

The methodology of this thesis is based on Knowledge Discovery in Data Bases (KKD) combined with SNA developed by [8, 28]. Two data sources are used in this thesis. Firstly, data from a VCoP, which was previously selected, cleaned and pre-processed by Álvarez [8]. Second, some data from a VCoI, which was selected, cleaned and pre-processed according to a SNA-KDD process. Also, a graphical representation of these data is built using text mining techniques. Finally, SNA techniques are used to detect overlapping communities. This methodology is showed in Figure 1.1.

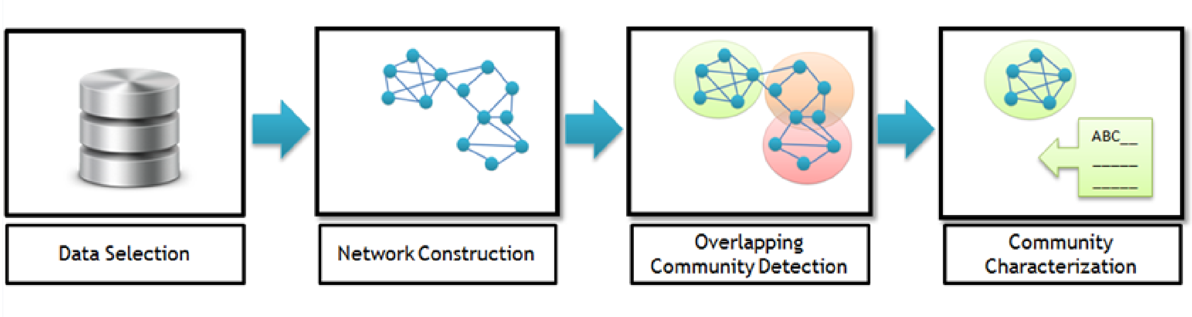


Figure 1.1: Overlapping Community Detection

The methodology used for the development of this thesis is structured in the following steps:

- **Previous Work and overlapping community detection**
  For the accomplishment of this thesis, it is necessary to have a previous background about Social Network Analysis and their applications over Virtual Communities. Also, Text Mining methods for content extraction are required in order to understand the relevance of the content generated by community members. For that reason, state of the art of both, SNA and TM will be reviewed to establish the most appropriate methods for this thesis. Then, a review of methods for overlapping community detection and community characterization are developed as well.
- **Graph Representation**
  The community graph representation will depend on their definition, in other words, the definition of both, nodes and arcs. Also, the configuration of the graph, in terms of links between nodes is relevant for further experiments, because it will represent the interaction among the community members.
- **Overlapping Community Detection Algorithm selection**
  The overlapping community detection problem will be solved using state of the art algorithms. These algorithms must have two properties. First, a quality measure over algorithms' output has to be applied. Second, the algorithms have to be applied over a graph.
- **Proposed Methodology Application**
  Overlapping community detection algorithms and text mining techniques are both applied over a real VCoP and a real VCoI. Both results are shown independently to detect if there are different effects when the same methodology is applied over different kinds of VC. Moreover, two novel algorithms to incorporate a semantic approach in the overlapping community detection problem are shown.
- **Analysis and Conclusions**
  After the presented methodology is applied, the algorithms designed for this thesis' purpose are compared with state of the art algorithms and their performance over different kinds of VC's are shown. Finally, a conclusion for each step described above are shown.

1.5 Thesis Structure

In the next chapter, a description about Virtual Communities of Practice and Interest, related work about the state of the art in Social Network Analysis, their applications in overlapping community detection, Text Mining for topic extraction and content reduction, Techniques to mine association rules are presented. The main idea of this chapter is establish that actual approach it is not considering the content which community members develops.

On chapter 3, the methodology used in this thesis is presented, following the SNA-KDD process: Community structure and data required for this work, Text Mining approach for community content reduction, Network configuration, according to how the replies in the community are defined; network filtering, considering the results of Text Mining methods; network construction algorithm, which explain how to build the graph with the filter and without it; an approach to detect overlapping communities in the different network configuration using both state of the art algorithms and algorithms presented in this thesis, and finally how results will be analysed, evaluated and characterized.

Chapter 4, presents the main contribution of this thesis, two novel algorithms to detect overlapping communities which include semantic information from users' posts to enhance the quality of sub-community structure detected.

Then, on chapter 5, an experiment on a real life VCoP is presented. Here, the VCoP is described in terms of content, users, and main topics. Also, the text processing method for the needed content representation and evaluation method are presented. Then, main results for both traditional and proposed system are presented and analysed. These results are presented according to the evaluation criteria previously introduced. Chapter 5 has the same structure of chapter 6, but it shows the experiment on a real life VCoI.

Finally, the main conclusions are presented, including our main findings and contributions, as well as the future work and lines for research.

Chapter 2

Previous Work

The following remarks are intended to give a brief introduction into the necessary topics to understand this thesis. To do so, it is essential to start with some preliminary explanations of the virtual communities theory, social network analysis and graph theory. After that, topics models for content extraction and the state of the art algorithms for overlapping community detection are described. The performance of these algorithms is measure with two quality measures which are described as well. Finally, association rules theory and algorithms to mine them are presented with the aim of to characterize overlapped communities detected.

2.1 Virtual Communities

A virtual community is a social network of individuals who interact through specific social media, potentially crossing geographical and political boundaries in order to pursue mutual interests or goals. There are different kinds of Virtual Communities. Kim et al. [50] organizes Social Web Communities describing the kind of users, uses, and needed features for every type of community. Wenger [90] identified three different Virtual Communities (VC), depending on the objectives pursued: access to information (VC of Interest), to complete a particular objective (VC of Purpose) or knowledge about an specific topic, skill or profession (VC of Practice). Specifically, it defines Virtual Communities of Practice (VCoP) as a group of people who share about specific topics and depth their own knowledge and expertise interacting on a friendly interface. People become members of a VCoP through shared practices and they are linked to each other through their involvement in common activities. It is this mutual engagement that binds the members of a VCoP together in a single social entity. Otherwise, Community of Interest (VCoI) is a community of people who share a common interest or passion, where ideas and thoughts are exchanged about the given passion, but may know (or care) a little about each other outside this area. Participation in a community of interest can be compelling, entertaining and create a “sticky” community where people return frequently and remain for extended periods.

Many authors has studied CoP. According Wenger [90] and Bobrow [20] we can notice there is three crucial characteristics to define a CoP:

- **The domain or knowledge:** The community's identity is defined by a shared domain of interest. It's possible to recognize a member for their commitment to the domain. This domain constitutes valuable knowledge for the community.
- **The community:** Members build relationships while they pursuit their domain of interest. They engage in joint activities and discussions, help each other, and share information. Theses relationships enable them to learn from each other.
- **The practice or sharing:** It's not enough to share the same interest to constitute a community of practice. Members of a community of practice interacts to each other to develop experiences, tools, stories, etc.

2.2 Social Network Analysis

The basis of social network analysis (also known as network science or network sociology) is that individual nodes (which, depending on the type of network, can be people, events, etc.) are connected by complex yet understandable relationships that form networks [10, 88] The origin of contemporary social network analysis can be traced back to the work of Stanley Milgram [39]. In his famous 1967 experiment, Milgram conducted a test to understand how people are connected to others by asking random people to forward a package to any of their acquaintances who they thought might be able to reach the specific target individual[65] . In his research, Milgram found that most people were connected by six acquaintances. This research led to the famous phrase *six degrees of separation*, which is still widely used in popular culture. Lately, experiments conducted by Lars et al. [9] using the entire facebook network of active users showed that most of people were connected by four acquaintances.

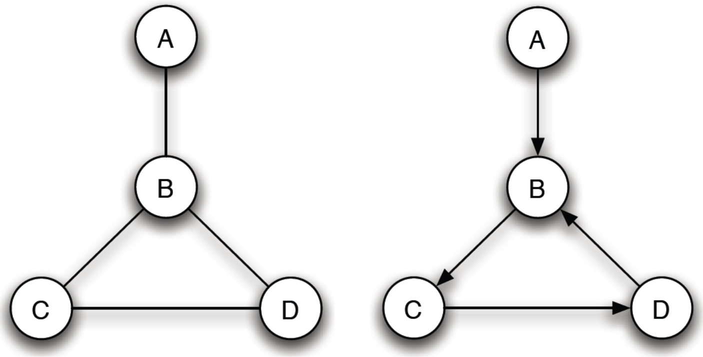
Another important step in the development of social network analysis was the work of Mark Granovetter on network structures. In his widely-cited 1973 article "The Strength of Weak Ties", Granovetter argues that "weak ties" -your relationships with acquaintances- are more important than "strong ties" -your relationships with family and close friends- when trying to find employment [81]. Granovetter's article and subsequent research extended this argument by positing that more disperse, non-redundant, open networks have greater access to information and power than smaller, denser, and more interconnected networks because they supply more diversity of knowledge and information.

Research on networks consists in empirical observations about the structure of networks and the models giving rise to such structures. The empirical analysis of networks aims to discover common structural properties or patterns [25], such as heavy-tailed degree distributions [23, 34], local clustering of edges [60, 89], small diameters [6, 58] , navigability [51, 65], emergence of network community structure [36, 40, 59], and so on. In parallel, there have been efforts to develop the network-formation mechanisms that naturally generate networks with the observed structural features. In these network-formation mechanisms, there have been two relatively dichotomous modelling approaches.

2.2.1 Network representation using Graph Theory

In this section some of the basic ideas behind graph theory are developed, the study of network structure. This allows to formulate basic network properties in a unifying language. **Graphs: Nodes and Edges.** A graph is a way of specifying relationships among a collection of items. A graph consists of a set of objects, called nodes, with certain pairs of these objects connected by links called edges. For example, the graph in Figure 2.1(a) consists of 4 nodes labelled A, B, C, and D, with B connected to each of the other three nodes by edges, and C and D connected by an edge as well. We say that two nodes are neighbors if they are connected by an edge.

Figure 2.1: Two graphs: (a) an undirected graph, and (b) a directed graph.



(a) A graph on 4 nodes.

(b) A directed graph on 4 nodes.

In Figure 2.1(a), it's possible to think of the relationship between the two ends of an edge as being symmetric; the edge simply connects them to each other. In many settings, however, asymmetric relationships are useful -for example, that A points to B but not vice versa. For this purpose, a *directed graph* is defined to consist of a set of nodes, as before, together with a set of directed edges; each directed edge is a link from one node to another, with the direction being important. Directed graphs are generally drawn as in Figure 2.1(b), with edges represented by arrows. When it wants to emphasize that a graph is not directed, we can refer to it as an *undirected graph.*

Mathematically, we represent a network by a graph (V, g) which consists of a set of nodes $V = \{1, \ldots, n\}$ and a $n \times n$ matrix $g = [g_{\mathrm{i}j}]_{\mathrm{i},j \in V}$ (referred to as an adjacency matrix), where $g_{\mathrm{i}j} \in \{0, 1\}$ represents the availability of an edge from node i to node j. The edge weight $g_{\mathrm{i}j} > 0$ can also take on non-binary values, representing the intensity of the interaction, in which case we refer to (V, g) as a weighted graph. We refer to a graph as a *directed graph* if $g_{\mathrm{i}j} \neq g_{j\mathrm{i}}$ and an *undirected graph* if $g_{\mathrm{i}j} = g_{j\mathrm{i}}$ $\forall \mathrm{i}, j \in V$.

Another representation of a graph is given by (V, E), where E is the set of edges in the network. For *directed graphs*, E is the set of directed edges, i.e., $(\mathrm{i}, j) \in E$ and for *undirected graphs* E is the set of undirected edges, i.e., $\{\mathrm{i}, j\} \in E$.

Also, two nodes are said to be *adjacent*, *neighbors*, or *connected* if there exist an edge between them. If all k nodes in the graph are adjacent, the graph is said to be *k-complete.* A graph is simple, i.e. loop-less and lacks multiple edges, if there is at most one edge between each pair of nodes and no node is neighbor with itself.

A *walk* is an ordered set of alternating nodes and edges that starts in one node i and ends in another node j. If the walk only transverses each node at most once, it is called a *path*. A k-*cycle* is a path where the first and last nodes are the same, and the path contains k edges. A graph is *connected*, if there exists a path between any given pair of nodes. The *shortest path* between two nodes is the *geodesic* and the longest geodesic is the diameter of the graph.

A graph without cycles is called a *tree* (or a forest if unconnected). A *subgraph*, G', of a graph G contains all edges that connect a subset of the node set, i.e. $V'(G) \subset V(G)$ such that $E'(G) \subset E(G)$ contains all edges connecting the nodes in $V'(G)$. One says that the edge set is spanned by the set of nodes. Two subgraphs are therefore disjoint and not connected. A k-*clique* is a k-complete sub-graph.

2.2.2 Metrics in social network analysis

Within graph theory and network analysis, there are various measures of the centrality of a vertex within a graph that determine the relative importance of a vertex within the graph. There are four measures of centrality that are widely used in network analysis: degree centrality, closeness, betweenness, and eigenvector centrality.

Degree centrality

The most intuitive measure of centrality of a vertex into a network is called degree centrality. Given a graph $G = (V, E)$ represented by means of its adjacency matrix A, in which a given entry $A_{\mathrm{i}j} = 1$ if and only if i and j are connected by an edge, and $A_{\mathrm{i}j} = 0$ otherwise, the *degree centrality* $C_D(v_\mathrm{i})$ of a vertex $v_\mathrm{i} \in V$ is defined as:

$$C_D(v_\mathrm{i}) = \mathrm{d}(v_\mathrm{i}) = \sum_j A_{\mathrm{i}j} \tag{2.1}$$

The idea behind the degree centrality is that the importance of a vertex is determined by the number of vertices adjacent to it, i.e. the larger their degree, the more important the vertex is.

Even though, in real world networks only a small number of vertices have high degrees, the degree centrality is a rough measure but it is adopted very often because of the low computational cost required for its computation. There exists a *normalized* version of the degree centrality, defined as follows

$$C'_D(v_\mathrm{i}) = \frac{\mathrm{d}(v_\mathrm{i})}{n-1} \tag{2.2}$$

where n represents the number of the vertices in the network.

Closeness centrality

A more accurate measure of centrality of a vertex is represented by the *closeness centrality* [82]. The closeness centrality relies on the concept of *average distance*, defined as:

$$D_{avg}(v_\mathrm{i}) = \frac{1}{n-1} \sum_{j \neq \mathrm{i}}^{\mathrm{i}} g(v_\mathrm{i}, v_j) \tag{2.3}$$

where $g(v_\mathrm{i}, v_j)$ represents the geodesic distance between vertices v_i and v_j.

The closeness centrality $C_C(v_\mathrm{i})$ of a vertex v_i is defined as

$$C_C(v_\mathrm{i}) = \frac{1}{n-1} \sum_{j \neq \mathrm{i}}^{\mathrm{i}} g(v_\mathrm{i}, v_j) \tag{2.4}$$

In practice, the closeness centrality calculates the importance of a vertex on how close the give vertex is to the other vertices. Central vertices, with respect to this measure, are important as they can reach the whole network more quickly than non-central vertices. Different generalizations of this measures for weighted and disconnected graphs have been proposed in [74].

Betweenness centrality

A more complex measure of centrality is the betweenness centrality [37, 38]. It relies on the concept of shortest paths. In detail, in order to compute the betweenness centrality of a vertex, it is necessary to count the number of shortest paths that pass across the given vertex.

The betweenness centrality $C_B(v_\mathrm{i})$ of a vertex v_i is computed as

$$C_B(v_\mathrm{i}) = \sum_{v_s \neq v_\mathrm{i} \neq v_t \in V} \frac{\sigma_{st}(v_\mathrm{i})}{\sigma_{st}} \tag{2.5}$$

where σ_{st} is the number of shortest paths between vertices v_s and v_t and $\sigma_{st}(v_\mathrm{i})$ is the number of shortest paths between v_s and v_t that pass through v_i. Vertices with high values of betweenness centrality are important because maintain an efficient way of communication inside a network and foster the information diffusion.

Eigenvector centrality

Another way to assign the centrality to a vertex is based of the idea that if a vertex has many central neighbors, it should be central as well. This measure is called eigenvector centrality and establishes that the importance of a vertex is determined by the importance of its neighbors. The eigenvector centrality $C_E(v_\mathrm{i})$ of a given vertex v_i is

$$C_E(v_\mathrm{i}) \propto \sum_{v_j \in N_\mathrm{i}} A_{\mathrm{i}j} C_E(v_j) \tag{2.6}$$

where N_i is the neighborhood of the vertex v_i, being $x \propto Ax$ that implies $Ax = \lambda x$. The centrality corresponds to the top eigenvector of the adjacency matrix A.

For simplicity, we will compute Degree, Betweenness and Closeness Centrality for all networks in this work. However, results for every node of these networks are not shown in this thesis. Results are available in `http://alturl.com/75b69` [1].

2.3 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) by Blei et al. [18] is a probabilistic generative model that can be used to estimate properties of multinomial observations by unsupervised learning. With respect to text modelling, LDA is a method to perform so-called latent semantic analysis (LSA). The intuition behind LDA is to find the latent structure of *topics* or *concepts* in a text corpus, which captures the meaning if the text that is imagined to be obscured by *word choice* noise. Then term latent semantic analysis has been coined in [31], who empirically showed that the co-occurrence structure of terms in text documents can be used to recover this latent topic structure. In turn, latent topic representation of text allow modelling of linguistic phenomena like synonymy and polysemy. This allows information retrieval systems to represent text in a way suitable for matching user needs (queries) with content items on a meaning level rather than by lexical congruence. LDA is a model closely linked to the probabilistic latent semantic analysis (PLSA) [47], an application of the latent aspect method to the latent semantic analysis task. More specifically, LDA extends PLSA method by defining a complete generative model[18].

The idea behind LDA is to model documents as arising from multiple topics, where a topic is defined to be a distribution over a fixed vocabulary of terms. Specifically, we assume that K topics are associated with a collection, and that each document exhibits these topics with different proportions. This is often a natural assumption to make because documents in a corpus tend to be heterogeneous, combining a subset of main ideas or themes that permeate the collection as a whole.

[1] https://www.dropbox.com/s/7un49rf572xlbsz/Graph%20Stat.xlsx

In LDA, the observed data are the words of each document and the hidden variables represent the latent topical structure, i.e., the topics themselves and how each document exhibits them. Given a collection, the posterior distribution of the hidden variables given the observed documents determines a hidden topical decomposition of the collection. Applications of topic modelling use posterior estimates of these hidden variables to perform tasks such as information retrieval and document browsing. The interaction between the observed documents and hidden topic structure is manifest in the probabilistic generative process associated with LDA, the imaginary random process that is assumed to have produced the observed data.

2.3.1 The model

2.3.1.1 Mixture Modelling

LDA is a mixture model, i.e., it uses a convex combination of a set of component distributions to model observations. A convex combination is a weighted sum whose weighting proportion coefficients sum to one. In LDA, a word w is generated from a convex combination of topics z. In such a mixture model, the probability that a word w instantiates term t is:

$$p(w = t) = \sum_k p(w = t|z = k)p(z = k) \tag{2.7}$$

$$\sum_k p(z = k) = 1 \tag{2.8}$$

where each mixture component $p(w = t|z = k)$ is a multinomial distribution over terms that corresponds to one of the latent topics $z = k$ of the text corpus. The mixture proportion consists of the topic probabilities $p(z = k)$. However, LDA goes a step beyond a global topic proportion and conditions the topic probabilities on the document a word belongs to. Based on this, we can formulate the main objectives of LDA inference: to find (1) the term distribution $p(t|z = k) = \vec{\beta_k}$ for each topic k and (2) the topic distribution $p(z|\mathrm{d} = \mathrm{d}) = \vec{\theta}_\mathrm{d}$ for each document d. The estimated parameter sets $\Phi = \{\vec{\beta_k}\}_{k=1}^{K}$ and $\Theta = \{\vec{\theta_\mathrm{d}}\}_{\mathrm{d}=1}^{D}$ are the basis for latent-semantic representation of words and documents.

2.3.1.2 Generative Model

Let K be a specified number of topics, V the size of the vocabulary, $\vec{\alpha}$ a positive K-vector, and η a scalar. We let $Dir_V(\vec{\alpha})$ denote a V-dimensional Dirichlet with vector parameter$\vec{\alpha}$ and $Dir_K(\eta)$ denote a K dimensional symmetric Dirichlet with scalar parameter η.

1. For each topic $k \in [1, K]$,
   (a) Draw a distribution over words $\vec{\beta_k} \sim Dir_K(\eta)$
2. For each document $\mathrm{d} \in [1, D]$,

(a) Draw a vector of topic proportions $\vec{\theta_{\text{d}}} \sim Dir_V(\vec{\alpha})$

(b) For each word $n \in [1, N_{\text{d}}]$ in document d,

i. Draw a topic assignment $Z_{\text{d},n} \sim Mult(\vec{\theta_{\text{d}}}), Z_{\text{d},n} \in \{1, \ldots, K\}$

ii. Draw a word $W_{\text{d},n} \sim Mult(\vec{\beta_{Z_{\text{d},n}}}), W_{\text{d},n} \in \{1, \ldots, V\}$

This is illustrated as a directed graphical model in 2.2. Nodes denote random variables; edges denote dependence between random variables. Shaded nodes denote observed random variables; unshaded nodes denote hidden random variables. The rectangular boxes are *plate notation*, which denote replication.

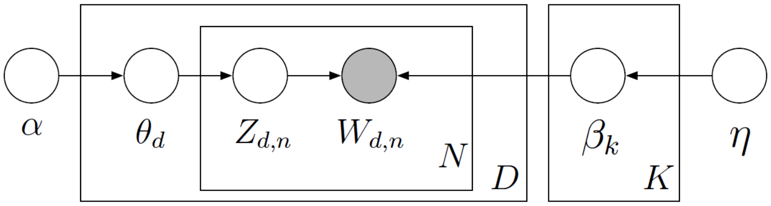


Figure 2.2: A graphical model representation of the latent Dirichlet allocation (LDA).

The hidden topical structure of a collection is represented in the hidden random variables: the topics $\vec{\beta}_{1:K}$, the per-document topic proportions $\vec{\theta}_{1:D}$, and the per-word topic assignments $Z_{1:D,1:N}$. With these variables, LDA is a type of mixed-membership model.

2.3.1.3 Likelihoods

Given the parameters α and η, the joint distribution of a topic mixture θ, a set of N topics z, and a set of N words w is given by:

$$p(\theta, z, w|\alpha, \eta) = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta) p(w_n|z_n, \eta) \tag{2.9}$$

where $p(z_n|\theta)$ is simplyθ_{i} for the unique i such that $z_n^{\text{i}} = 1$ Integrating over θ and summing over z, we obtain the marginal distribution of a document:

$$p(w|\alpha, \eta) = \int p(\theta|\alpha) \left(\prod_{n=1}^{N} \sum_{z_n} p(z_n|\theta) p(w_n|z_n, \eta) \right) \mathrm{d}\theta \tag{2.10}$$

Finally, taking the product of the marginal probabilities of single documents, we obtain the probability of a corpus:

$$p(D|\alpha,\eta) = \prod_{\mathrm{d}=1}^{M} \int p(\theta_\mathrm{d}|\alpha) \left(\prod_{n=1}^{N_\mathrm{d}} \sum_{z_{\mathrm{d}n}} p(z_{\mathrm{d}n}|\theta_\mathrm{d}) p(w_{\mathrm{d}n}|z_{\mathrm{d}n},\eta) \right) \mathrm{d}\theta_\mathrm{d} \tag{2.11}$$

The LDA model is represented as a probabilistic graphical model in Figure 2.2. As the figure makes clear, there are three levels to the LDA representation. The parameters α and η are corpus level parameters, assumed to be sampled once in the process of generating a corpus. The variables θ_d are document-level variables, sampled once per document. Finally, the variables $z_{\mathrm{d}n}$ and $w_{\mathrm{d}n}$ are word-level variables and are sampled once for each word in each document.

2.3.1.4 Inference via Gibbs sampling

The Inference problem wants to ask the question: To which topics does a given document belong to? Thus want to compute the posterior distribution of the hidden variables of the hidden variables given a document:

$$p(\theta, z|w,\alpha,\eta) = \frac{p(\theta, z, w|\alpha,\eta)}{p(w|\alpha,\eta)} \tag{2.12}$$

Although, Latent Dirichlet Allocation is still a relatively simple model, exact inference is generally intractable. The solution to this is to use approximate inference algorithms, such as mean-field variational expectation maximisation [18], expectation propagation [66], and Gibbs sampling [44]. Gibbs sampling is a special case of Markov-chain Monte Carlo (MCMC) simulation [63] and often yields relatively simple algorithms for approximate inference in high-dimensional models such as LDA.

2.4 Overlapping community detection

Detecting clusters or communities in real-world graphs such as large social networks, web graphs, and biological networks is a problem of considerable practical interest that has received a great deal of attention [26, 35, 40, 45, 49].

The first problem in graph clustering is to look for a quantitative definition of community. According to Fortunato [36] there is no universally accepted definition. As a matter of fact, the definition often depends on the specific system at hand and/or application one has in mind. From intuition we get the notion that there must be more edges *inside* the community than edges linking vertices of the community with the rest of the graph [40, 78]. This is the reference guideline at the basis of most community definitions. But many alternative recipes are compatible with it. Moreover, in most cases, communities are algorithmically defined, i. e. they are just the final product of the algorithm, without a precise a priori definition.

Mathematically we can define this problem as follows:

Given a network or graph $G = \{E, V\}$, V is a set of n nodes and E is a set of m edges. For dense graphs $m = O(n^2)$, but for sparse networks $m = O(n)$. The network structure is determined by the $n \times n$ adjacency matrix A for unweighted networks and weight matrix W for weighted networks. Each element $A_{\mathrm{i}j}$ of A is equal to 1 if there is an edge connecting nodes i and j; and it is 0 otherwise. Each element $w_{\mathrm{i}j}$ of W takes a non-negative real value representing strength of connection between nodes i and j.

In the case of overlapping community detection, the set of clusters found is called a *cover* $C = \{c_1, c_2, \ldots, c_k\}$ [53], in which a node may belong to more than one cluster. Each node i associates with a community according to a belonging factor (i.e., soft assignment or membership) $[a_{\mathrm{i}1}, a_{\mathrm{i}2}, \ldots, a_{\mathrm{i}k}]$ [68], in which $a_{\mathrm{i}c}$ is a measure of the strength of association between node i and cluster c. Without loss of generality, the following constraints are assumed to be satisfied

$$0 \leq a_{\mathrm{i}c} \leq 1 \, \forall \mathrm{i} \in V, \, \forall c \in C, \tag{2.13}$$

$$\sum_{k=1}^{|C|} a_{\mathrm{i}c} = 1, \tag{2.14}$$

where $|C|$ is the number of clusters.

In general, algorithms produce results that are composed of one of two types of assignments, *crisp* (non-fuzzy) assignment or *fuzzy* assignment [43]. With *crisp* assignment, each node belongs to one or more communities with *equal* strength. The relationship between a node and a cluster is *binary*. That is, a node i either belongs to cluster c or does not. With fuzzy assignment, each node is associated with communities in proportion to a belonging factor. With a threshold, a *fuzzy* assignment can be easily converted to a *crisp* assignment. Typically, a detection algorithm outputs *crisp* community assignments.

2.4.1 Overlapping Community Detection Algorithms

Huge number of algorithms have been developed using a variety of methods; these vary in their effectiveness and time performance for different types of network. In this section, algorithms for overlapping community detection are reviewed and categorized into five classes which reflect how communities are identified.

2.4.1.1 Clique Percolation

The clique percolation algorithm [86] (CPM) detects communities based on the assumption that a community or k-community is a set of nodes which can be reached through a series of adjacent k-cliques (a k-clique is set of k nodes, which are all connected to each other), where two k-cliques are adjacent if they share $(k-1)$ nodes. The algorithms begins by identifying all cliques of size k in a network. Afterward, the method build k-communities from k-cliques found. Since a vertex can be in multiple k-cliques simultaneously, overlap between communities is possible. Empirically, $k = 3$ or 4 has been show to give the best results. CFinder [2]is an implementation of CPM, whose time complexity is polynomial in many applications. Despite conceptual simplicity, one may argue that CPM-like algorithms are more like pattern matching rather than finding communities since they aim to find specific, localized structures in a network.

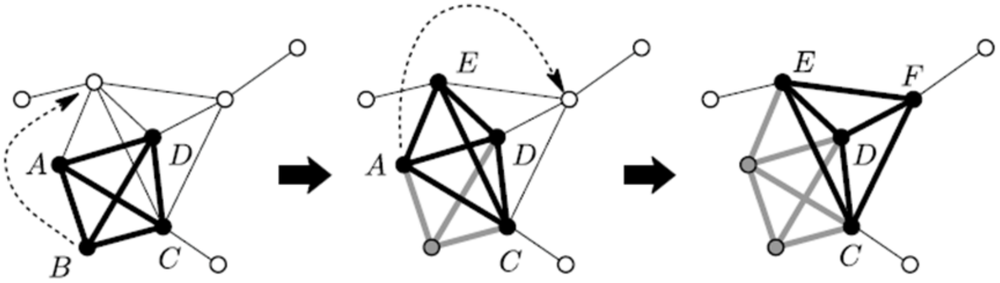


Figure 2.3: Clique Percolation [85]

Figure 2.3 shows the iterative procedure to detected overlapping communities using Clique Percolation methods.

2.4.1.2 Line Graph and Link Partitioning

The idea of partitioning links instead of nodes to discover community structure has also been explored. A node in the original graph is called overlapping if links connected to it are put in more than one cluster. In [4], links are partitioned via hierarchical clustering of edge similarity. Given a pair of links $\mathrm{e}_{\mathrm{i}k}$ and e_{kj} incident on a node k, a similarity can be computed via the Jaccard Index defined in Equation 2.15.

$$S(\mathrm{e}_{\mathrm{i}k}, \mathrm{e}_{kj}) = \frac{|N_{\mathrm{i}} \cap N_j|}{|N_{\mathrm{i}} \cup N_j|} \tag{2.15}$$

Where N_{i} is the neighborhood of node i including i. Single-linkage hierarchical clustering is then used to build a link dendrogram. Cutting this dendrogram at some threshold yields to linked communities. The time complexity is $O(nk_{max}^2)$, where k_{max} is the maximum node degree in the network. Although, the link partitioning for overlapping detection seems conceptually natural, there is no guarantee that it provides higher quality detection than node based detection does [36] because these algorithms also rely on an ambiguous definition of community.

2.4.1.3 Local Expansion and Optimization

Algorithms utilizing local expansion and optimization are based on growing a natural community [53] or a partial community. Most of them rely on a local benefit function that

[2] `www.cfinder.org` [online: accessed 22-02-2012]

characterizes the quality of a densely connected group of nodes. Baumes et al. [12, 13] proposed a two-step process. First, the algorithm *RankRemoval* is used to rank nodes according to some criterion. Then, the process iteratively removes highly ranked nodes until small, disjoint cluster cores are formed. These cores serve as seed communities for the second step of the process, *Iterative Scan* (IS), that expands the cores by adding or removing nodes until a local density function cannot be improved. The proposed density function can be formally given as

$$f(c) = \frac{w_{\text{in}}^{c}}{w_{\text{in}}^{c} + w_{out}^{c}} \tag{2.16}$$

where w_{in}^{c} and w_{out}^{c} are the total internal and external weight of the community c. The worst-case running time is $O(n^2)$. The quality of discovered communities depends on the quality of seeds.

OSLOM [3] [57] is based on the local optimization of a fitness function. This function expresses the statistical significance of clusters with regard to global null model (i.e., the random graph generated by the configuration model). OSLOM can be used alone or as a refinement procedure of partitions/covers delivered by other techniques.

OSLOM consists of three phases:

- First, it looks for significant clusters, until convergence is reached
- Second, it analyses the resulting set of clusters, trying to detect their internal structure or possible unions
- Third, it detects found clusters' hierarchical structure

iLCD [4] [24] is capable to detect communities taking network dynamics into account. Given a set of edges created at some time step, iLCD updates the existing communities by adding a new node if the following rules are satisfied:

- The node is able to access easily (in two step or lesser) to most of the community (at least as much as the other nodes of the community)
- The node has a *robust access* to other nodes; which means a node can be reached by at least two paths of length two or less.

The complexity of iLCD is $O(nk^2)$ in general, whose precise quantity depends on community structures and its parameters.

First, we need to define the input of the iLCD (intrinsic Longitudinal Community Detection) algorithm. Due to the longitudinal analysis, we will use a list of edges, ordered by their creation time, those edges could correspond to links creation among existing nodes or could also imply the creation of a new node. As some edges creations can be simultaneous (think

[3] `www.oslom.org` [online: accessed 22-02-2012]

[4] `http://cazabetremy.fr/Cazabetremy/iLCD.html` [online: accessed 22-02-2012]

about the publication of several articles in a given journal issue), we will use ordered sets of edges, where edges of a given set are created at the same time.

More formally, let's note $G = (V, E)$ the graph that is dynamically built and $C =< C_k >$the set of communities that is dynamically built. Initially, G and C are empty. We then define E_{in} the set of edges in input as $E_{in} =< E_t >$ i.e. composed by ordered time-stamped sets of edges. (see Algorithm 1 for pseudo-code)

Algorithm 1 iLCD

```
for each time-stamped set $E_t$ do
  for each edge $(u, v)$ of the set $E_t$ do
    Add $(u, v)$ to $E$. If $u$ or $v$ is not in $V$, add it to $V$
    Determine the updates of existing communities. For each community $C_k$ to
    which $u$ (respectively $v$) belongs, try to integrate $v$ (resp. $u$ ) to $C_k$
  end for
  Update of previous communities
  If $u$ y $v$ do not already belong to the same community, Try to create a new community.
  Merge similar communities.
end for
```

2.4.1.4 Fuzzy Detection

Fuzzy community detection algorithms quantify the strength of association between all pairs of nodes and communities. In these algorithms, a soft membership vector, or belonging factor [42], is calculated for each node. A drawback of such algorithms is the need to determine the dimensionality k of the membership vector. This value can be either provided as a parameter to the algorithm or calculated from the data.

2.4.1.5 Agent Based and Dynamical Algorithms

COPRA[5] [42], is an algorithm based on the label propagation technique of Raghavan, Albert and Kumara; which is able to detect communities that overlap. Like the original algorithm, vertices have labels that propagate between neighbouring vertices so that community members reach a consensus on their community membership, each node updates its belonging coefficients by averaging the coefficients from all its neighbors at each time step in a synchronous fashion.

COPRA algorithm(Community Overlap Propagation Algorithm) keeps two vectors of vertex labels: *old* y *new*; *old.x* (resp. *new.x*) denotes the previous (resp. updated) label for vertex x.

[5] `www.cs.bris.ac.uk/~steve/networks/software/copra.html` [online: accessed 22-02-2012]

Each vertex label is a set of pairs (c, b), where c is a community identifier and b is the belonging coefficient. $N(x)$ is the set of neighbours of vertex x (see Algorithm 2 for pseudocode).

Algorithm 2 COPRA

```
For each vertex $x$
  old.$x$ ← $\{(x, 1)\}$.
For each vertex $x$
  Propagate($x$, old, new).
If id(old)=id(new):
  min ← mc(min, count(new)).
Else:
  min ← count(new).
If min ≠ oldmin:
  old ← new.
  oldmin ← min.
  Repeat from step 2.
For each vertex $x$
  ids ← id(old.$x$).
  For each $c$ in ids:
    If, for some $g$, $(c, g)$ is in coms, $(c, \mathrm{i})$ in sub:
      coms ← coms $-\{(c, g)\} \cup \{(c, g \cup \{x\})\}$.
      sub ← sub $-\{(c, \mathrm{i})\} \cup \{(c, \mathrm{i} \cap \mathrm{id}s)\}$.
    Else:
      coms ← coms $\cup\{(c, \{x\})\}$.
      sub ← sub $\cup\{(c, \mathrm{id}s)\}$.
For each $(c, \mathrm{i})$ in sub:
  If $\mathrm{i} \neq \{\}$ : coms ← coms $-(c, g)$.
Split disconnected communities in coms.
```

SLPA[6] [93] is a general speaker-listener based information propagation process. It spreads labels between nodes according to pairwise interaction rules. Unlike others algorithms, where a node forgets knowledge gained in the previous iterations, SLPA provides each node with a memory to store received information (labels). The membership strength is interpreted as the probability of observing a label in a node's memory. One advantage of SLPA is that it does not require any knowledge about the number of communities. The time complexity is $O(tm)$, linear in the number of edges m, where t is a predefined maximum number of iterations.

SLPA is an extension of the Label Propagation Algorithm (LPA) proposed by Raghavan, Albet and Kumara. In LPA, each node holds only a single label that is iteratively updated by adopting the majority label in the neighborhood. Disjoint communities are discovered when the algorithm converges. One way to account for overlap is to allow each node to possess multiple labels. SLPA follows this idea but applies different dynamics with more general features.

In the dynamic process, we need to determine 1) how to spread nodes' information to others; 2) how to process the information received from others. The critical issue related to both questions is how information should be maintained. A speaker-listener based information propagation process (SLPA) is proposed to mimic human communication behavior.

In SLPA, each node can be a listener or a speaker. The roles are switched depending on whether a node serves as an information provider or information consumer. Typically, a node can hold as many labels as it likes, depending on what it has experienced in the stochastic

[6] https://sites.google.com/site/communitydetectionslpa

processes driven by the underlying network structure. A node accumulates knowledge of repeatedly observed labels instead of erasing all but one of them. Moreover, the more a node observes a label, the more likely it will spread this label to other nodes (mimicking people's preference of spreading most frequently discussed opinions).

This algorithm will be the base to develop new algorithms to include semantic information, according the objectives stated in Section 3.8.

2.4.2 Evaluation Criteria

Evaluating the quality of a detected partitioning or cover is nontrivial, and extending evaluation measures from disjoint to overlapping communities is rarely straightforward.

In this section we present two well-know quality measures *Normalized Mutual Information* and *Modularity*. In this thesis we will use *Normalized Mutual Information* to compare two covers and it will be useful when the ground-truth is known. If the ground-truth is unknown we use *Modularity* as clustering quality measure.

2.4.2.1 Normalized Mutual Information

There are many evaluation criteria in the literature (see [47]), but most of them can be used to compare partitions: a *partition* is a union of subsets which are non-overlapping and which cover the whole set; a *cover* is just a collection of(overlapping) subsets.

Although there is currently no consensus on which is the best measure, information theoretic based measures have received increasing attention for their strong theoretical background. Let us first review some of the very fundamental concepts of information theory [27] and then see how those concepts might be used toward assessing clusterings agreement.

Definition 1 The information entropy of a discrete random variable X, that can take on possible values in its domain $\chi = \{x_1, x_2, \ldots, x_n\}$ is defined by:

$$H(X) = -\sum_{x \in \chi} p(x) \; log(p(x)) \tag{2.17}$$

Definition 2 The mutual information between two random variables X and Y with respective domains χ and Υ is defined by:

$$I(Y, X) = \sum_{x \in \chi} \sum_{y \in \Upsilon} p(y, x) \; log \frac{p(y, x)}{p(x)p(y)} \tag{2.18}$$

The mutual information (see Figure 2.4) is a symmetric measure that quantifies the mutual dependence between two random variables, or the information that X and Y share. It measures how much knowing one of these variables reduces our uncertainty about the other.

This property suggests that the mutual information can be used to measure the information shared by two clusterings, and thus, assess their similarity. Lancichinetti et al. [56] has extended the notion of normalized mutual information to account for overlap between communities.

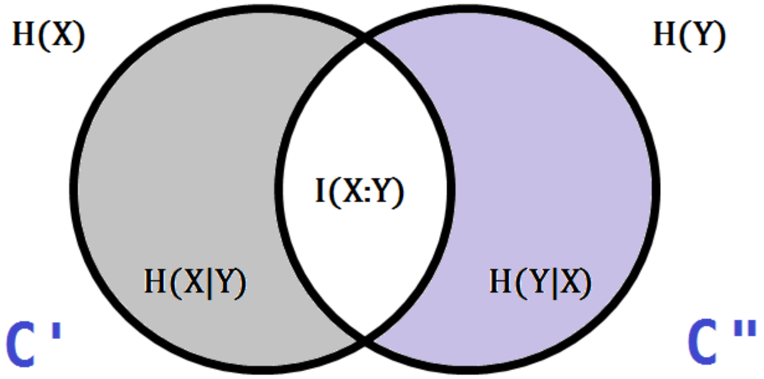


Figure 2.4: Mutual Information

The normalized mutual information is defined as:

$$NMI(X|Y) = \frac{H(X) + H(Y) - H(X,Y)}{(H(X) + H(Y))/2} \tag{2.19}$$

where $H(X)$ $(H(Y))$ is the entropy of the random variable X (Y) associated to the partition C' (C''), whereas $H(X,Y)$ is the joint entropy. This variable is in the range $[0,1]$ and equals 1 only when the two partitions C' and C'' are exactly coincident.

For each node i in cover C', its community membership can be expressed as a binary vector of length $|C'|$ (i.e., the number of clusters in C'). $(x_{\mathrm{i}})_k = 1$ if node i belongs to the k^{th} cluster C'_k; $(x_{\mathrm{i}})_k = 0$ otherwise. The k^{th} entry of this vector can be viewed as a random variable X_k, whose probability distribution is given by $P(X_k = 1) = n_k/n$, $P(X_k = 0) = 1 - P(X_k = 1)$, where $n_k = |C'_k|$ is the number of nodes in the cluster C'_k and n is the total number of nodes. The same holds for the random variable Y_l associated with the l^{th} cluster in cover C''. The joint probability distribution $P(X_k, Y_l)$ is defined as:

$$\begin{aligned}
P(X_k = 1, Y_l = 1) &= \frac{|C'_k \cap C''_l|}{n} \\
P(X_k = 1, Y_l = 0) &= \frac{|C'_k| - |C'_k \cap C''_l|}{n} \\
P(X_k = 0, Y_l = 1) &= \frac{|C''_l| - |C'_k \cap C''_l|}{n} \\
P(X_k = 0, Y_l = 0) &= \frac{n - |C'_k \cup C''_l|}{n}
\end{aligned}$$

The conditional entropy of a cluster X_k given Y_l is defined as $H(X_k|Y_l) = H(X_k, Y_l) - H(Y_l)$. The entropy of X_k with respect to the entire vector Y is based on the best matching between X_k and any component of Y given by

$$H(X_k|Y) = min_{l \in \{1,2,\ldots,|C''|\}} H(X_k|Y_l)$$

The normalized conditional entropy of a cover X with respect to Y is:

$$H(X|Y) = \frac{1}{|C'|} \sum_k \frac{H(X_k|Y)}{H(X_k)}$$

In the same way, one can define $H(X|Y)$. Finally the NMI for two covers C' and C'' is given by:

$$NMI(X|Y) = 1 - [H(X|Y) + H(Y|X)]/2$$

The extended NMI is between 0 and 1, with 1 corresponding to a perfect matching.

2.4.2.2 Modularity

To develop a method for community identification, one needs an evaluation criteria to judge the quality of the detected community structure. One of such measures was proposed by Newman and Girvan in [72] and is based on the intuitive idea that random networks do not exhibit (strong) community structure. To measure the quality of a cover produced by overlapping detection algorithms on real-world social networks where the ground truth is usually unknown, most measures extend the framework of modularity Q for a disjoint partition, which is given as:

$$Q = \frac{1}{2m} \sum_c \sum_{\mathrm{i},j \in c} \left[A_{\mathrm{i}j} - \frac{k_\mathrm{i} k_j}{2m} \right],$$

where c is a community, $A_{\mathrm{i}j}$ is the element of the adjacency matrix for nodes i and j, $m = \frac{1}{2} \sum_{\mathrm{i}j} A_{\mathrm{i}j}$ is the total number of edges, and k_i is the degree of node i.

The idea behind Newman's modularity is simple: a subgraph is a community if the number of links among nodes in the subgraph is higher than what would be expected if links were randomly placed. This is exactly what happens in real-world communities, where the number and density of links among people belonging to groups (families, clubs, user groups etc) is higher than expected in a random graph of the same size [69, 71, 75]. This definition of modularity implies the choice of a so-called *null model* [70]; i.e. graph model to which any other graph can be compared in order to assert the existence of any degree of modularity. When testing for modularity of a complex network, the null model used has so far been a random graph with the same number of nodes, the same number of edges and the same degree distribution as in the original graph, but with links among nodes randomly placed.

2.4.2.3 Link Based Modularity

Nicosia et al. [73] proposed an extension to modularity measure based on the belonging coefficients of links to account for overlap between communities. Since each node has a belonging coefficient for each community, it is possible to define this coefficient for incoming or outgoing edges from a node. We can intuitively suppose that the community belonging coefficient c of an edge $l = (\mathrm{i}, j)$ which starts at node i and ends at node j can be represented by a certain function of the corresponding belonging coefficients of i and j to community c, in the following equation:

$$\beta_{l(\mathrm{i},j),c} = F(a_{\mathrm{i}c}, a_{jc}) \tag{2.20}$$

The expected belonging coefficient of any possible link $l(\mathrm{i}, j)$ from node i to a node j in community c can be defined as $\beta^{out}_{l(\mathrm{i},j),c} = \frac{1}{|V|}\sum_{j\in V} F(a_{\mathrm{i}c}, a_{jc})$. Accordingly, the expected belonging coefficient of any link $l(\mathrm{i}, j)$ pointing to node j in community c is defined as $\beta^{\mathrm{in}}_{l(\mathrm{i},j),c} = \frac{1}{|V|}\sum_{\mathrm{i}\in V} F(a_{\mathrm{i}c}, a_{jc})$. Latter coefficients are used as weights for the probability of an observed link and the probability of a link starting from i to j in the null model. These are used in the new modularity defined as:

$$Q^{\mathrm{Ni}}_{ov} = \frac{1}{m}\sum_{c}\sum_{\mathrm{i},j\in V}\left[\beta_{l(\mathrm{i},j),c}A_{\mathrm{i}j} - \beta^{out}_{l(\mathrm{i},j),c}\beta^{in}_{l(\mathrm{i},j),c}\frac{k^{out}_{\mathrm{i}}k^{\mathrm{in}}_{j}}{m}\right] \tag{2.21}$$

2.5 Topic Association Rules

The problem of mining association rules over basket data was introduced in [3]. In this thesis a similar approach is used to mine association rules over post data.

The following is a formal statement of the problem [4]: Let $T = \{t_1, t_2, \ldots, t_m\}$ be a set of topics. Let D be a set of posts, where each post P is a set of topics such that $P \subseteq T$. Associated with each post is a unique identifier, called its PID. We say that a post P *contains* X, a set of some topics in T, if $X \subseteq P$.

An *association rule* is an implication of the form $X \Rightarrow Y$, where $X \subseteq T$, $Y \subseteq T$, and $X \cap Y = \emptyset$. The rule $X \Rightarrow Y$ holds in the post set D with *confidence* c if $c\%$ of posts in D that contain X also contain Y . The rule $X \Rightarrow Y$ has *support* s in the post set D if $s\%$ of posts in D contain $X \cup Y$.

2.5.1 FP-Growth Algorithm

As shown in [46], the main bottleneck of the Aprioi-like methods is at the candidate set generation and test. FP-growth algorithm [5] is an efficient method of mining all frequent topicsets without candidate's generation. The algorithm mine the frequent topicsets by using a divide-and-conquer strategy as follows: FP-growth first compresses the database representing frequent topicset into a frequent-pattern tree, or FP-tree, which retains the topicset association information as well. The next step is to divide a compressed database into set of conditional databases (a special kind of projected database), each associated with one frequent topic. Finally, mine each such database separately. Particularly, the construction of FP-tree and the mining of FP-tree are the main steps in FP-growth algorithm.

The definition, according to [46] is as follows.

Definition 3 (FP-tree) A frequent pattern tree is a tree structure defined below.

1. It consists of one root labeled as *root*, a set of topic prefix sub-trees as the children of the root, and a frequent-topic header table.
2. Each node in the topic prefix sub-tree consists of three fields: topic-name, count, and node-link, where topic-name registers which topic this node represents, count registers the number of topics represented by the portion of the path reaching this node, and node-link links to the next node in the FP-tree carrying the same topic-name, or null if there is none.
3. Each entry in the frequent-topic header table consists of two fields, (1) topic-name and (2) head of node-link, which points to the first node in the FP-tree carrying the topic-name.

The actual algorithm, according also to [46] is:

The FP-growth [46] algorithm for mining frequent patterns with FP-tree by pattern fragment growth is:

Algorithm 3 FP-tree construction

INPUT: A forum database DB and a minimum support threshold ξ

OUTPUT: Its frequent pattern tree, **FP-tree**

METHOD: The FP-tree is constructed in the following steps:

1. Scan the forum database DB once. Collect the set of frequent topics F and their supports. Sort F in support descending order as L, the *list* of frequent topics.
2. Create the root of an **FP-tree**, T, and label it as "root". For each post $Post$ in DB do the following.
   - a. Select and sort the frequent topics in $Post$ according to the order of L. Let the sorted frequent topic list in $Post$ be $[p|P]$, where p is the first element and P is the remaining list. Call $insert-tree([p|P],T)$.
   - b. The function $insert-tree([p|P],T)$ is performed as follows. If T has a child N such that $N.topic-name = p.topic-name$, then increment N's count by 1; else create a new node N, and let its count be 1, its parent link be linked to T, and its node-link be linked to the nodes with the same topic-name via the node-link structure. If P is nonempty, call $insert-tree(P,N)$ recursively.

Algorithm 4 FP-Growth algorithm

INPUT:

- a FP-tree constructed with the above mentioned algorithm.
- D forum database.
- s minimum support threshold.

OUTPUT: The complete set of frequent patterns.

METHOD:

```
  call FP-growth(FP-tree, null)
Procedure FP-growth(Tree, A)
if $Tree$ contains a simple path $P$ then
  for each combination (denoted as $B$) of the nodes in the path $P$ do
    generate pattern $B \cup A$ with support= minimun support of nodes in$B$
  end for
else
  for each ai in the header of the Tree do
    generate pattern $B = ai \cup A$ with support = ai.support
    construct B's conditional pattern base and B's conditional FP-tree $TreeB$
    if $TreeB \neq \emptyset$ then
      call FP-Growth(TreeB, B)
    end if
  end for
end if
```

Chapter 3

Methodology

In this thesis we used SNA-KDD methodology developed by Ríos & Aguilera [79]. Other applications of SNA-KDD methodology can be found in [8, 64]. This is an hybrid approach which combines *Knowledge Discovery in Databases* (KDD) and *Social Network Analysis*(SNA). We customized SNA-KDD methodology to detect overlapping communities. The main idea is to use KDD steps and incorporate Text Processing and SNA to the process. Figure 3.1 illustrates the SNA-KDD approach.

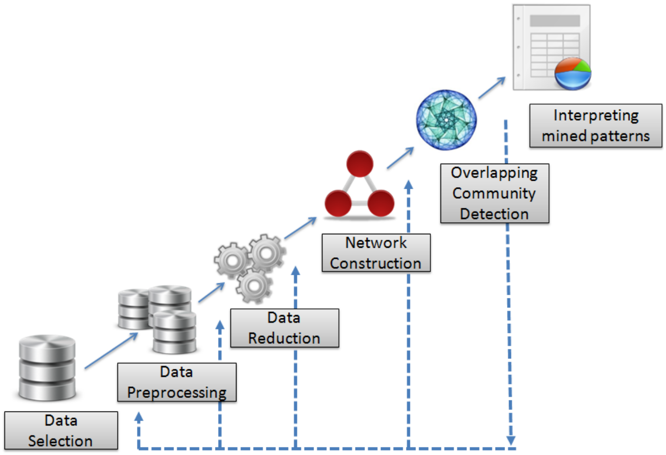


Figure 3.1: Overview of the steps constituting the SNA-KDD process

The overall process of finding and interpreting patterns from data involves the repeated application of the following steps:

- **Learning the application domain** includes relevant prior knowledge and the goals of the application.
- **Data selection** includes selecting a dataset or focusing on a subset of variables or data samples on which discovery is to be performed
- **Data cleaning and preprocessing** includes basic operations, such as removing noise or outliers if appropriate, collecting the necessary information to model or account for noise, deciding on strategies for handling missing data fields, and accounting for time

sequence information and known changes, as well as deciding DBMS issues, such as data types, schema, and mapping of missing and unknown values.

- **Data reduction** includes finding useful features to represent the data, depending on the goal of the task, and using dimensionality reduction or transformation methods to reduce the effective number of variables under consideration or to find invariant representations for the data. Considering data selected's characteristic text mining techniques are used for data reduction.
- **Network construction** includes using forum's structure to represent forum's interactions as a network. This step includes: network configuration, network filtering and network construction.
- **Overlapping community detection and characterization** includes selecting method(s) to be used for detecting overlapping communities, such as deciding which models and parameters may be appropriate.
- **Interpreting mined patterns** includes interpreting the discovered patterns and possibly returning to any of the previous steps, as well as possible visualization of the extracted patterns, removing redundant or irrelevant patterns, and translating the useful ones into terms understandable by users.
- **Consolidating discovered knowledge** includes incorporating this knowledge into the performance system, taking actions based on the knowledge, or simply documenting it and reporting it to interested parties, as well as checking for and resolving potential conflicts with previously believed (or extracted) knowledge.

This section is organized as follows: first, basic notation and document's representation is introduced; then, how every step of SNA-KDD methodology was customized in order to discover overlapping communities is presented.

3.1 Basic Notation

Let us introduce some concepts. In the following, let $\mathcal{V}$ a vector of words that defines the vocabulary to be used. We will refer to a word w, as a basic unit of discrete data, indexed by $\{1, ..., |\mathcal{V}|\}$. A post message is a sequence of S words defined by $\mathbf{w} = (w^1, ..., w^S)$, where w^s represents the s^{th} word in the message. Finally, a corpus is defined by a collection of $\mathcal{P}$ post messages denoted by $\mathcal{C} = (\mathbf{w}_1, ..., \mathbf{w}_{|\mathcal{P}|})$.

A vectorial representation of the posts corpus is given by `TF-IDF`$= (m_{\mathrm{i}j}), \mathrm{i} \in \{1, \ldots, |\mathcal{V}|\}$ and $j \in \{1, \ldots, |\mathcal{P}|\}$, where $m_{\mathrm{i}j}$ is the weight associated to whether a given word is more important than another in a document. The $m_{\mathrm{i}j}$ weights considered in this research is defined as an improvement of the *tf-idf* term [83] (*term frequency times inverse document frequency*), defined by:

$$m_{\mathrm{i}j} = \frac{n_{\mathrm{i}j}}{\sum_{k=1}^{|\mathcal{V}|} n_{kj}} \times log\left(\frac{|\mathcal{C}|}{n_{\mathrm{i}}}\right) \tag{3.1}$$

Where $n_{\mathrm{i}j}$ is the frequency of the ith word in the j^{th} document and n_{i} is the number of documents containing word i. The *tf-idf* term is a weighted representation of the importance of a given word in a document that belongs to a collection of documents. The *term frequency* (TF) indicates the weight of each word in a document, while the *inverse document frequency* (IDF) states whether the word is frequent or uncommon in the document, setting a lower or higher weight respectively.

A *Temporal Virtual Community* (TVC) is a list of following time frames (time windows) T. Each time frame is in fact a single Virtual Community $VC(V, E)$, where V is a set of vertices and E is a set of undirected edges $\langle x, y \rangle : x, y \in V$.

$$\begin{aligned} TVC &= \langle T_1, T_2, \ldots, T_m \rangle, m \in N \\ T_{\mathrm{i}} &= VC_{\mathrm{i}}(V_{\mathrm{i}}, E_{\mathrm{i}}), \mathrm{i} = 1, 2, \ldots, m \\ E_{\mathrm{i}} &= \langle x, y \rangle : x, y \in V_{\mathrm{i}}, \mathrm{i} = 1, 2, \ldots m \end{aligned}$$

An example of a *Temporal Virtual Community* (TVC) is presented in Figure 3.2. it consists of three time frames, and each time frame is a separate virtual community created from data gathered in the particular interval of time. In this work, disjoint intervals will be used, it means one interval starts when the previous interval ends.

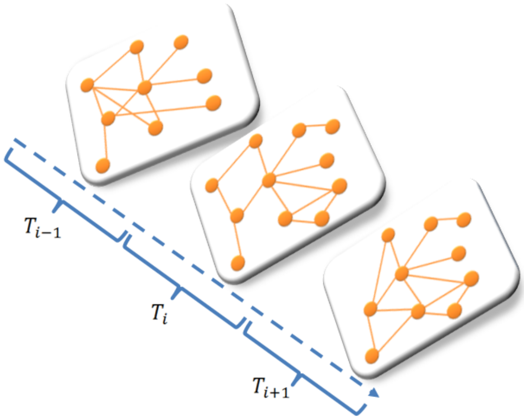


Figure 3.2: The example of Temporal Virtual Community consisting of three time frames

3.2 Data Selection

VCs' usually are supported by forum systems (like VBuletin, PHPbb, etc.). The forum is the virtual place in which members interact each other and generate knowledge. Then, the forum has categories where different topics are discussed. For example, a forum have categories like Sports, Movies, Music, etc., which are not related one to each other. In VCoPs' and VCoIs', on the contrary, the categories are related with a main purposes that members are interested to develop. Each conversation in the VCoP/VCoI is arranged in threads; generally started by a member question, and every member can participate in a thread by replying with a post. In other words, the members interaction in a VCoP/VCoI is represented by posts in the different threads.

As the object of this thesis is to detect overlapping communities, members data is necessary. Data like nicknames or user ID will be used to identify them, know with whom is interacting, and associate the content to the correct member. Another relevant data is the community content, representing by the members' posts. Like others web features, the content of the post could be text, images, hyper links, videos, etc. For the purpose of present work, the content used will be texts of the community posts. All the data related with the post is necessary, such as the thread and category it belongs, date of the post, who posted and the text of it.

3.3 Preprocessing Data

In order to use content for overlapping community detection, members' posts will be used, but it could not be possible to use it directly. In terms of the message itself, forums sometimes include quotes, so members' content would be replicated other members' post. In this case, a member would be assigned to a community only for the content that he is quoting. So a first filter approach is to identify the quotes and deleted from post, keeping only the new generated content. Also, there are post which not represent a contribution for the community, such as spam, trolling or flood posts.

This kind of messages have to be detected and ignored for the latter analysis to compare members replies. From the point of view of the words of a post, misspelling and acronyms difficult the comparison between a pair of post. Also, there are terms which not correspond to words that are used in forums, such as emoticons or terms like *laughs*, hahaha, LOL, ROFL or XD. To solve this, a preprocessing step has to be applied. A preprocessing step may also contain some transformations applied to the text, depending on the problem at hand. Examples of such transformations include stop word removal and stemming. In this section we explain some of these concepts and how various techniques can be combined to pre-process text:

3.3.1 Stop words

A stop word is a predefined word by the user that should be filtered out, i.e. removed from documents. A large proportion of the words in a document are function words such as articles and conjunctions. Examples of such words in English are and, is, of and so on. In the bag of words model, these words don't tell us anything about the content of a text, so they are usually added to the stop word list. Also numerics can be removed from the text in some applications.

The function words need to be manually identified for each language. Another way of deciding which stop words to use is to find them using word frequency when constructing the vocabulary list of the corpus. A word occurring in most of the documents in a corpus will not help discriminate the documents from each other, hence it may be removed from the corpus by adding it to the stop word list.

There are however cases where functional words are of importance, for example phrase search. Consider the phrase *to be or not to be*. By using functional words as stop words, this phrase would be completely removed.

3.3.2 Stemming

Many words in a text is not in its lemma form. For example, *dog* and *dogs* refer to the same concept if you ask a human, but a computer treat them as completely different words. This means that a document containing only the form *dog* will not be seen as similar to a document containing *dogs*. There are a couple of known ways to cope with this problem.

One technique is to use a dictionary which maps all known words and their inflections to their lemma form. This is called lemmatization. Although efficient, this technique requires access to such a dictionary which may not be available. It also requires that the dictionary always is up to date since new words are regularly added to languages.

Another approach is suffix stripping algorithms, which was first examined by Porter for English [77] but now exist for many languages [15], such as Spanish, French, Portuguese, among other. These algorithms simply follow a small set of rules which removes the suffixes. This approach is called stemming since it leaves only the stem of the word, for example brows for browse and browsing. Although the results may not be real words, it maps words with standard inflections into the same stems, and thus reduces the number of word types. A potential problem with this approach is that words with different semantic meaning (which should be separate words in the analysis) can be stripped to the same stem. Below is the example document with Porters stemming algorithm applied to it.

3.4 Data Reduction: Topic Modelling

A topic model, e.g. Latent Dirichlet Allocation (LDA) [19], can be considered a probabilistic model that relates documents and words through variables which represent the main topics inferred from the text itself. In this context, a document can be considered as a mixture of topics, represented by probability distributions which can generate the words in a document given these topics. The inferring process of the latent variables, or topics, is the key component of this model, whose main objective is to learn from text data the distribution of the underlying topics in a given corpus of text documents.

To LDA, given the smoothing parameters η and α and a joint distribution of a topic mixture θ, the idea is to determine the probability distribution to generate – from a set of topics $\mathcal{K}$ – a message composed by a set of N words w ($\mathbf{w} = (w^1, \ldots, w^N)$),

$$p(\theta, z, \mathbf{w}|\alpha, \eta) = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta) p(w^n|z_n, \eta) \tag{3.2}$$

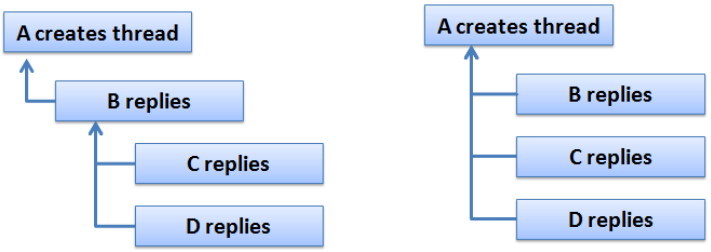
Where $p(z_n|\theta)$ can be represented by the random variable θ_{i}; such topic z_n is presented in document i ($z_n^{\mathrm{i}} = 1$). A final expression can be deduced by integrating Equation 3.2 over the random variable θ and summing over topics $z \in \mathcal{K}$.

3.5 Network Configuration

To build the social network graph, the members' interaction must be taken into consideration. In general, members' activity is followed according to its participation on the forum. Likewise, participation appears when a member post a comment in the community. Because, VCs' activity is described according members' participation, the network will be configured according to the following: *nodes* will represent VCs' members and *arcs* will represent interaction between them. How to link the members and how to measure their interactions to complete the network is our main concern.

There are two kinds of forums. *Directed Forums*, which shows clearly to whom is replying a member, and post are aligned according to which member is replying and the time when it was posted, and *Undirected Forums*, where it is not possible to identify to whom is replying, posts are aligned only according their time in which was posted. Figure 3.3 illustrate both forum classifications.

Figure 3.3: An example of thread post sequence.



(a) Directed Forum (b) Undirected Forum

For Undirected Forums, it is necessary to take assumptions about to which members is replying. In this work we used a all-previous-reply network[7, 8, 62] to represent the VCs' network. This means, when a member creates a post in a thread, every following reply will be related to all people who replied before on thread. In other words, we assume that last reply is a broadcast to all members who post a comment before in that specific thread.

In Figure 3.4 the all previous reply approach is presented. Arcs represents members' replies and nodes represent the members who made the posts.

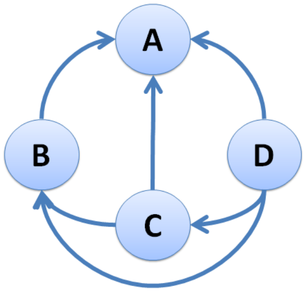


Figure 3.4: All previous reply approach

3.6 Topic-based Network Filtering

The idea is to compare with euclidean distance two members' posts semantic information. The semantic is extracted or represented by topics, which are not keywords. If the similarity is over a certain threshold θ, an interaction will be considered between them. We support the idea that this will help us to avoid irrelevant interactions. For example, in a VC with k topics, let TB_j a post of user j that is a reply to post of user i (TB_i). The distance between them will be calculated with Equation (3.3).

$$\mathrm{d}_m(TB_\mathrm{i}, TB_j) = \frac{\sum_k g_{\mathrm{i}k} g_{jk}}{\sqrt{\sum_k g_{\mathrm{i}k}^2 \sum_k g_{jk}^2}} \tag{3.3}$$

Where $g_{\mathrm{i}k}$ is the score of topic k in post of user i. It is clear that the distance exists only if TB_j is a reply to TB_i. After that, the weight of arc $a_{\mathrm{i},j}$ is calculated according to Equation (3.4).

$$a_{\mathrm{i},j} = \sum_{\substack{\mathrm{i},j \\ \mathrm{d}_m(TB_\mathrm{i},TB_j)\geq\theta}} \mathrm{d}(TB_\mathrm{i}, TB_j) \tag{3.4}$$

3.7 Network Construction

Considering all posts $\mathcal{P}$, the network is built following the structure described in Section 3.5. In other words, for each post of user i in a thread, the arc $a_{\mathrm{i},j}$ is added for each user j who post a comment on that thread. But, we only consider the arcs if their messages' distance is greater or equal than the threshold θ in Equation 3.4. This way, we are able to filter arcs by topic similarity to a specific threads' topics. Algorithm 6 presents the pseudo-code on how a the graph $\mathcal{G} = (V, E)$ is build by using the All-Previous Reply network.

Algorithm 5 Network Filtering

```
Require: $\mathcal{P}$ (Posts)
Require: $\mathcal{V}$ (Vocabulary)
Require: $k$ (Number of Topics)
Ensure: Network $\mathcal{G} = (V, E)$
  Initialize $V = \{\}, E = \{\}$
  for each i $\in \mathcal{P}$ do
    $V \leftarrow V \cup \mathrm{i}$
  end for
  for each i $\in \mathcal{P}.creator$ do
    for each $j \in \{\mathrm{i}.replies\}, \mathrm{i} \neq j$ do
      if $\mathrm{d}_m(P_\mathrm{i}, P_j) \geq \theta$ then
        $a_{\mathrm{i},j} \leftarrow a_{\mathrm{i},j} + 1$
        $E \leftarrow E \cup a_{\mathrm{i},j}$
      end if
    end for
  end for
```

Algorithm 6 All Previous Reply Topic-based Network

```
Require: $\mathcal{P}$ (Posts)
Require: $\mathcal{V}$ (Vocabulary)
Require: $k$ (Number of Topics)
Ensure: Network $\mathcal{G} = (V, E)$
  Initialize $V = \{\}, E = \{\}, Prev = \{\}$
  for each thread $t \in \mathcal{P}$ do
    $Prev = \{\}$
    $\mathrm{i} \leftarrow t.creator$
    $V \leftarrow V \cup \mathrm{i}$
    $Prev \leftarrow Prev \cup \mathrm{i}$
    for each $j \in \{t.replies\}, \mathrm{i} \neq j$ do
      for each $k \in Prev$ do
        if $\mathrm{d}_m(P^t_k, P^t_j) \geq \theta$ then
          $a_{\mathrm{i},j} \leftarrow 1$
          $E \leftarrow E \cup a_{k,j}$
        end if
      end for
      $Prev \leftarrow Prev \cup j$
    end for
  end for
```

3.8 Overlapping Community Detection in Topic-Based Networks

The overlapping community detection problem can be solved using a topology-based approach (See section 2.4.1), but these algorithms ignore the content generated for users of a VC, where every post can let us improve the detection of the communities which a node belongs. Approaches to incorporate content information in the overlapping community detection are presented in [32, 94]. However, these methodologies don't present an algorithm able to use semantic information as an input. In Chapter 4 we present two novel algorithms which incorporate both a topology-based approach and a topic-based approach for the overlapping community detection.

3.9 Community Characterization

A community is characterized for replying to two main questions. Firstly, we would like to find out what the community is talking about, and secondly, which topics could be of interest for the community members.

To find out what the community is talking about we will use the following methodology. Let $T = VC(V, E) \in TVC$ a virtual community and $C_1, C_2, \ldots, C_{|C|}$ a set of cover for virtual community T. For each cover C_j we define the content community vector CV_j of length K, where K is number of topics extracted for corpus $\mathcal{C}$. The component CV_j^k can be expressed as follows:

$$CV_j^k = \frac{|cv_j^k|}{\sum_{j \in C} |cv_j^k|} \tag{3.5}$$

where $cv_j^k = \{P_{\text{il}} \in TB \mid q_{ikl} \geq \kappa \; and \, \text{i} \in C_j\}$, cv_j^k is the set of posts written by members of community j who has a topic score q_{ikl} for topic k greater than a threshold $\kappa \in [0, 1]$.

Finally, to determine which topics could be of interest for the community members, topics association rules will be used (See 2.5). For this, the next procedure is used:

1. Select the topic set for the same time frame used to build the analyzed network. Then, a thresholding procedure is applied to transform the current topic set to a dummy topic set.
2. For each community select the post published for members who belong to the community. Finally, FP-Growth algorithm is applied to that topic set.

Mathematically, let $T = VC(V, E) \in TVC$ a virtual community, $C_1, C_2, \ldots, C_{|C|}$ a set of cover for virtual community T and $\mathcal{P}$ the set of posts over the same time frame of T. For each post $p \in \mathcal{P}$ we define the dummy post vector $\bar{p} \in \bar{\mathcal{P}}$ of length K, where K is number of topics extracted for corpus $\mathcal{C}$, as follows:

$$\bar{p}_k = \begin{cases} 1 & : g_k \geq \tau \\ 0 & : g_k < \tau \end{cases}$$

where g_k is the topic score for topic k. Then, for each cover o community C_j the problem of mining association rules will be solve with FP-Growth Algorithm (see Section 2.5) over the set $\bar{\mathcal{P}}_j$, where $\bar{\mathcal{P}}_j = \{\bar{p} \in \bar{\mathcal{P}} \mid \bar{p}\ was\ post\text{ed}\ for\ us\text{er}\ \text{i}\ \in C_j\}$

Chapter 4

Design and Development of algorithms for overlapping community detection

This chapter presents a performance evaluation of state of the art algorithms for overlapping community detection. Then, two novel algorithms are presented. These algorithms are based on the algorithm which obtained the best performance. Unlike this algorithm, our approach include semantic information to enhance overlapping community detection.

4.1 Benchmark and test in synthetic networks

It is necessary to have good benchmarks to both study the behavior of a proposed community detection algorithm and to compare the performance across various algorithms. In order to accurately perform these two analyses, networks in which the ground truth is known are needed. This requirement implies that real-world networks, which are often collected from online or observed interactions, do not paint a clear enough picture due to their lack of *ground truth*. In light of this requirement, we begin our discussion with synthetic networks. In the Girvan-Newman benchmark [40], equal size communities are embedded into a network for a given expected degree and a given mixing parameter μ that measures the ratio of internal connections to outgoing connections. One drawback of this benchmark is that it fails to account for the heterogeneity in complex networks; Another is that it does not allow embedded communities to overlap. A few benchmarks have been proposed for testing overlapping community detection algorithms, the LFR[1] benchmark proposed in [54] introduces heterogeneity into degree and community size distributions of a network. These distributions are governed by power laws with exponents τ_1 and τ_2, respectively. To generate overlapping communities, O_n, the fraction of overlapping nodes is specified and each node is assigned to $O_m \geq 1$ communities. The generating procedure is equivalent to generating a bipartite network where the two classes are the communities and nodes subject to the requirement that the sum of community sizes equals the sum of node memberships. LFR also provides a rich set of parameters to control the network topology, including the mixing

[1] `https://sites.google.com/site/andrealancichinetti/files` [online: accessed 14-11-2012]

parameter μ, the average degree $\bar{k}$, the maximum degree k_{max}, the maximum community size c_{max}, and the minimum community size c_{min}. The LFR model brings benchmarks closer to the features observed in real-world networks. However, requiring that overlapping nodes interact with the same number of embedded communities is unrealistic in practice. A simple generalization, where each overlapping node may belong to different number of communities has been considered in [1].

We empirically compare the performance of different algorithms on LFR networks. We focus on algorithms which produce a crisp assignment of vertices to communities. In total, five algorithms that we were able to collect and test are listed in Table 4.1. Note that the time complexity given is for the worst case.

Table 4.1: Algorithms Included in the Experiments.

Algorithm	Complexity	Imp.
CFinder[2]	-	C++
iLCD[24]	nk^2	Java
COPRA[42]	$O(vmlog(vm/n))$	Java
OSLOM[57]	$O(n^2)$	C++
SLPA[93]	$O(tm)$	C++

For algorithms with tunable parameters, the performance with the optimal parameter is reported. For CFinder, k varies from 3 to 10; for COPRA, v varies from 1 to 10; For SLPA, the number of iterations t is set to 100 and r varies from 0.01 to 0.1. As in [92], the average performance over ten repetitions are reported for SLPA and COPRA.

We used networks with sizes $n \in \{1000, 5000\}$. The average degree is kept at $\bar{k} = 10$, which is of the same order as most large real-world social networks[2] The rest of the parameters of LFR generator are set similar to those in [54, 91, 92]: node degrees and community sizes are governed by power law distributions with exponents $\tau_1 = 2$ and $\tau_2 = 1$ respectively, the maximum degree is $k_{max} = 50$, and community sizes vary between $c_{min} = 20$ and $c_{max} = 100$. The mixing parameter μ is from $\{0.1, 0.3\}$, which is the expected fraction links through which a node connecting to other nodes in the same community.

The degree of overlap is determined by two parameters. O_n is the number of overlapping nodes, and O_m is the number of communities to which each overlapping node belongs. We fixed the former to be 10% of the total number of nodes. Instead of fixing the O_m, we allow it to vary from 2 to 8 indicating the diversity of overlapping nodes. By increasing the value of O_m, we create harder detection tasks in an intuitive way.

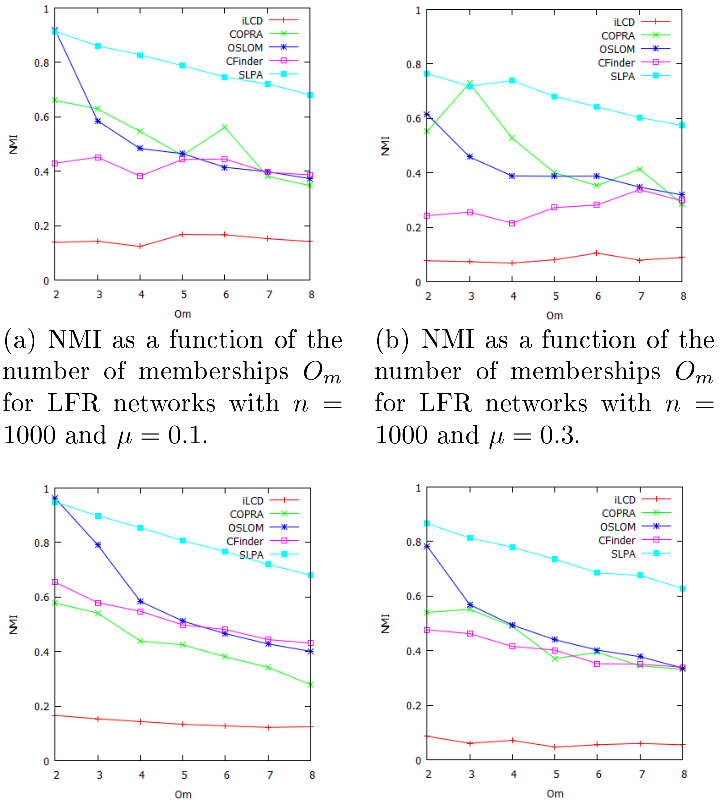
To evaluate the efficiency of the state of the art algorithms, we use supervised evaluation with Normalized Mutual Information (NMI) [55] to compare the obtained solution with the correct community structure specified by LFR-Benchmark's output. The NMI is used to quantify the quality of communities discovered by an algorithm. NMI measures the fraction of nodes in agreement in two covers, and it yields values between 0 and 1. The closer this value is to 1, the better the performance is.

We examine how the performance changes as the number of memberships O_m varies from small to large values (i.e., 2 to 8). The results for $n = \{1000, 5000\}$ are shown in Figure

[2] `http://snap.stanford.edu/data.` [online: accessed 14-11-2012]

4.1. In general, changes in the network topology, especially the mixing value μ. That is, the larger the value of μ, the poorer the results produced by detection algorithms. However, increasing network size from 1000 to 5000, results in slightly better performance. On the contrary, performance decays as the degree of overlapping increases (i.e., O_m getting larger) for almost all algorithms.

Figure 4.1: Results for benchmark evaluation.



(a) NMI as a function of the number of memberships O_m for LFR networks with $n = 1000$ and $\mu = 0.1$.

(b) NMI as a function of the number of memberships O_m for LFR networks with $n = 1000$ and $\mu = 0.3$.

(c) NMI as a function of the number of memberships O_m for LFR networks with $n = 5000$ and $\mu = 0.1$.

(d) NMI as a function of the number of memberships O_m for LFR networks with $n = 5000$ and $\mu = 0.3$.

It is worth noticing that SLPA achieves by far the best performance over all tested networks. With these results in the next section we presents two SLPA-based algorithms which include semantic information in order to enhance the overlapping community detection.

4.2 Proposed Algorithms for Overlapping Community Detection in Topic-Based Networks

Community or modular structure is considered to be a significant property of real-world social networks. Thus, numerous techniques have been developed for community detection. However, most of the work has been done on *disjoint* community detection. It has been well understood that people in a real social network are naturally characterized by *multiple* community memberships. For example, a person usually has connections to several social groups like family, friends and colleges; a researcher may be active in several areas; in the Internet, a person can simultaneously subscribe to an arbitrary number of groups. For this reason, discovering overlapping structures is necessary for realistic social analysis. Overlapping community detection algorithms aim to discover a cover [55], defined as a set of clusters in which each node belongs to at least one cluster.

In this section, we present SLTA [67, 80] (Speaker-Listener Topic Propagation Algorithm) and TPA (Topic Propagation Algorithm). These algorithms are an hybrid between two different overlapping community detection approaches, the first one considers the graph structure of the network (topology-based community detection approach); the second one takes the textual information of the network nodes into consideration (topic-based community detection approach)

4.2.1 SLTA: Speaker-Listener Topic Propagation Algorithm

The algorithm proposed in this thesis is a modification of the Speaker-Listener Propagation Algorithm (SLPA) [93]. In SLPA, the memory of each node is initialized with the node's id. Our algorithm follows this idea but applies different initialization process. SLTA mimics human pairwise communication behavior. At each communication step, each node serves as both a speaker (information provider) and a listener (information consumer). Specifically, each node broadcasts an interest topic to neighbors and at the same time receives an interest from each neighbor.

In summery, the proposed algorithm consists of the following three stages (see algorithm 7 for pseudo-code):

1. The memory of each node is initialized with his node's topic-label. The topic-label is computed using text mining techniques, specifically LDA which was applied in the data reduction step 3.4.
2. Then, the following steps are repeated until the stopping criterion is satisfied:
   a. One node is selected as a listener.
   b. Each neighbor of the selected node sends out a single label following certain *speaking rul*e, such as selecting a random label from its memory with probability proportional to the occurrence frequency of this label in the memory.
   c. The listener accepts one label from the collection of labels received from neighbors following certain *listening rul*e, such as selecting the most popular label from what it observed in the current step.
3. Finally, the post-processing based on the labels in the memories of nodes is applied to output the communities.

In the initialization process, the memory of a node i is initialized with his topic-label T_i unlike SLPA presented in section 2.4.1.5 where the memory of a node is initialized with his id, T_i is the topic which has the highest average score over all post messages from user i.

Mathematically:

$$T_{\mathrm{i}} = \arg\max_{k} \frac{1}{|TB|} \sum_{l \in TB} q_{\mathrm{i}kl} \tag{4.1}$$

Where TB is the set of user is' post messages, $q_{\mathrm{i}kl}$ is the score of topic k in post l of user i

Algorithm 7 SLTA($Topic, T, r$)

```
[n, Nodes]=loadnetwork();
[n, Topic]=loadtopic();
Stage 1: initialization
for i = 1 : $n$ do
  Nodes(i).Mem=Topic(i);
end for
Stage 2: evolution
for $t = 1 : T$ do
  Nodes.ShuffleOrder();
  for i = 1 : $n$ do
    Listener=Nodes(i);
    Speakers=Nodes(i).getNbs();
  end for
  for $j = 1 : Speakers.len$ do
    LabelList(j)= Speakers(j).speakerRule();
  end for
  w=Listener.listenerRule(LabelList);
  Listener.Mem.add(w);
end for
Stage 3: post-processing
for i = 1 : $n$ do
  remove Nodes(i) labels seen with probability $< r$;
end for
```

a. **Stopping Criterion:** Like SLPA, we can stop at any time as long as we collect sufficient information for post-processing. In the current implementation we simply stop when the predefined maximum number of iterations T is reached. In general, the algorithm produces relatively stable outputs, independent of network size or structure, when T is greater than 20. Although SLTA is non-deterministic due to the random selection, it performs well on average as shown in following sections.

b. **Post-processing and Community Detection:** Given the memory of a node, SLTA converts it into a probability distribution of labels. This distribution defines the association strength to communities to which the node belongs. A simple thresholding procedure is performed to produce a crisp assignment from nodes to communities. If the probability of seeing a particular label during the whole process is less than a given threshold $r \in [0, 1]$, this label is deleted from a node's memory. After thresholding, nodes having a particular label are grouped together and form a community. If a node contains multiple labels, it belongs to more than one community and is therefore called an overlapping node.

c. **Complexity:** The initialization of labels requires $O(n)$, where n is the total number of nodes. The outer loop is controlled by the user defined maximum iteration T, which is a small constant[3]. The inner loop is controlled by n. Each operation of the inner loop executes one speaking rule and one listening rule. For the speaking rule, selecting a label from the memory proportionally to the frequencies is, in principle, equivalent to randomly selecting an element from the array which is $O(1)$ operation. For listening rule, since the listener needs to check all the labels from its neighbors, it takes $O(\bar{K})$ on average, where $\bar{K}$ is the average degree. The complexity of the dynamic evolution (i.e., stage 1 and 2) for the asynchronous update is $O(Tm)$ on an arbitrary network and $O(Tn)$ on a sparse network, when m is the total number of edges. In the post-processing, the thresholding operation requires $O(Tn)$ operations since each node has a memory of size T. Therefore, the time complexity of the entire algorithm is $O(Tn)$ in sparse networks. It is worth to say that SLTA and SLPA, which was the best algorithm tested in our benchmark, have the same complexity.

4.2.2 TPA:Topic Propagation Algorithm

This Algorithm extends the idea presented in 4.2.1 where the topic more used for a node is propagated in the network. In this algorithm, nodes interact between them following certain interaction rule, this interaction rules updates the membership vector of each node in a asynchronous process. The membership vector for each node is initialized with his topic score's vector (see Equation 4.2). In summery, the proposed algorithm consists of the following three stages (see algorithm 8 for pseudo-code):

1. The membership vector of each node is initialized with his topic score's vector. The topic score's vector is computed using text mining techniques, specifically LDA which was applied in the data reduction step 3.4.
2. Then, the following steps are repeated until the stop criterion is satisfied:
   a. One node is selected as a candidate.
   b. The average membership vector of all neighbors of the selected node is calculated.
   c. The candidate updates his membership vector following certain interaction rule between his membership vector and the average membership vector of his neighbors. Then, the candidate's membership vector is normalized.
3. Finally, the post-processing based on the belonging vectors of nodes is applied to output the communities.

In the initialization process, the membership vector of a node i is initialized with his topic score's vector Ψ^{i} (see algorithm 8 for pseudo-code) ,Ψ^{i} is a vector where every component is the average score over all post messages from user i. Mathematically:

$$\Psi^{\mathrm{i}}_k = \frac{1}{|TB|} \sum_{l \in TB} q_{\mathrm{i}kl} \tag{4.2}$$

$$\Psi^{\mathrm{i}} = \frac{1}{\|\Psi^{\mathrm{i}}\|} \Psi^{\mathrm{i}} \tag{4.3}$$

[3] In our experiments, we used T set to 100

Where TB is the set of user is' post messages, $q_{\mathrm{i}kl}$ is the score of topic k in post l of user i.

Let Ψ^{i}_n be the normalized average membership vector of the node i's neighborhood. Then, the interaction rule between Ψ^{i} and Ψ^{i}_n at iteration t updates the belonging vector of node i as follows:

$$\Psi^{\mathrm{i},t} = \Psi^{\mathrm{i},t-1} + \varphi(\Psi^{\mathrm{i},t-1}, \Psi_n^{\mathrm{i},t-1})[\Psi_n^{\mathrm{i},t-1} - \Psi^{\mathrm{i},t-1}] \tag{4.4}$$

$$\Psi^{\mathrm{i},t} = \frac{1}{\|\Psi^{\mathrm{i},t}\|}\Psi^{\mathrm{i},t} \tag{4.5}$$

In this thesis we will use the function $\varphi(\Psi^{\mathrm{i}}, \Psi^{\mathrm{i}}_n)$ as:

$$\varphi(\Psi^{\mathrm{i}}, \Psi^{\mathrm{i}}_n) = \exp\left(-\frac{t}{4\log(k)}\right) \tag{4.6}$$

Algorithm 8 TPA($Topic, T, r$)

```
[n, Nodes]=loadnetwork();
[n, Topic]=loadtopic();
Stage 1: initialization
for i = 1 : $n$ do
  Nodes(i).MemV=Topic(i);
end for
Stage 2: evolution
for $t = 1 : T$ do
  Nodes.ShuffleOrder();
  for i = 1 : $n$ do
    Neighbors=Nodes(i).getNbs();
    AvgNeighborMemV=Neighbors.MemV.AVG();
    AvgNeighborMemV.Normalize();
    Node(i).MemV=IntRule(Node(i).MemV, AvgNeighborMemV);
    Node(i).MemV.Normalized();
  end for
end for
Stage 3: post-processing
for i = 1 : $n$ do
  for $j = 1 : No$de(i).$length$ do
    if If Nodes(i).MemV.Component(j) > $r$ then
      Nodes(i) belongs to community $j$
    end if
  end for
end for
```

a. **Stopping Criterion:** Like SLTA, we can stop at any time as long as we collect sufficient information for post-processing. In the current implementation we simply stop when the predefined maximum number of iterations T is reached. Although, TPA is non-deterministic due to the random selection, it performs well on average as shown in later sections; as well as SLTA.

b. **Post-processing and Community Detection:** Given the membership vector of a node, a simple thresholding procedure is performed to produce a crisp assignment. If the topic score of certain component is greater than a given threshold $r \in [0, 1]$, this node belong to a community characterized for the respective topic. If a node belongs to more than one community and is therefore called an overlapping node.

c. **Complexity:** The initialization of labels requires $O(nK)$, where n is the total number of nodes and K is the dimension if the topic score's vector. The outer loop is controlled by the user defined maximum iteration T, which is a small constant[4]. The inner loop is controlled by n. Each operation of the inner loop updates the membership vector for each node. For calculate the average membership vector of all neighbours is needed to operate with all neighbours' membership vectors, it takes $O(\bar{K})$ on average, where $\bar{K}$ is the average degree. Updates the membership vector according the interaction rule is $O(K)$ operation. The complexity of the dynamic evolution (i.e., stage 1 and 2) for the asynchronous update is $O(TKm)$ on an arbitrary network and $O(TKn)$ on a sparse network, when m is the total number of edges. In the post-processing, the thresholding operation requires $O(Kn)$ operations since each node has a membership vector of size K. Therefore, the time complexity of the entire algorithm is $O(TKn)$ in sparse networks. Despite of the fact that TPA, SLTA and SLPA have a similar structure they differ in the complexity of the dynamic evolution for asynchronous update, where SLTA and SLTA have the same complexity. With these algorithms there is a trade-off between complexity and how much semantic information is included to detect overlapping communities. TPA includes more semantic information than SLTA, but his complexity is higher.

[4] In our experiments, we used T set to 30

Chapter 5

Overlapping Community Detection on a real VCoP

The method presented in Section 4.2 was evaluated over Plexilandia forum[1], a virtual community of practice formed by a group of people who have met towards the building of music effects, amplifiers and audio equipment (like "Do it yourself" style). In the beginning was born as a community for share common experiences in the construction of plexies[2]. Today, plexilandia count more than 2,500 members and 10 years of activity. All these years they have been sharing and discussing their knowledge about building their own plexies, effects. Besides, there are other related topics such as luthier, professional audio, buy/sell parts. Although, they have a basic community information web page, most of their members' interactions are produced on the discussion forum. Table 5.1 presents the activity in the different categories of the forum since the beginning of the community until 2010 is shown.

Table 5.1: VCoP's Activities

Forum	2002	2003	2004	2005	2006	2007	2008	2009	2010	TOTAL
Amplifiers	392	2165	2884	3940	3444	3361	2398	1252	525	20362
Effects	184	1432	3362	3718	4268	5995	4738	2317	731	26745
Luthier	34	388	849	1373	1340	2140	926	699	452	8201
General	76	403	855	1200	2880	5472	3737	1655	666	16944
Pro Audio	—	—	—	—	—	342	624	396	132	1494
Synthetizers	—	—	—	—	—	—	—	104	65	169
TOTAL	686	4388	7950	10231	11932	17310	12423	6423	2571	73914

To illustrate the results, the temporal virtual community of practice consisted of a three-month TVC (See section 3.1 for *Temporal Virtual Community* description) extracted from plexilandia forum data, the first time frame begins in April 2009. In order to validate the proposed method, it was applied using two different LDA filtered networks. We compared SLTA with five well-know algorithms (see Table 5.2).

Like the experiments reported in [92], we used default parameters setting for most algo-

[1] `www.plexilandia.cl` [online: accessed 25-05-2012]

[2] "Plexi" is the nickname given to Marshall amp heads model 1959 that have the clear perspex (a.k.a plexiglass) fascia to the control panel with a gold backing sheet showing through as opposed to the metal plates of the later models.

Table 5.2: Algorithms Included in the Experiments.

Algorithm	Complexity	Imp.
CFinder[2]	-	C++
iLCD[24]	$O(nk^2)$	Java
COPRA[42]	$O(vmlog(vm/n))$	Java
OSLOM[57]	$O(n^2)$	C++
SLPA[93]	$O(Tm)$	C++
SLTA	$O(Tm)$	Java
TPA	$O(TKm)$	Java

rithms if applicable. For SLPA and SLTA, the maximum number if iterations T was set to 100 and r takes values in set $\{0.01, 0.05, 0.1, 0.2, 0.3, 0.4\}$ and for TPA, parameter r varies from $1/k$ to $5/k$ with an interval $1/2k$ where k is the number of extracted topics. The average performance over ten repetitions are reported for SLPA, SLTA and COPRA, since the algorithms can be rather non-deterministic.

We selected overlapping link based modularities Q_{ov}^{Ni} in Eq. (2.21) as quality measure. Although, modularity is a function of a cover and a network; we sometimes refer to the *modularity of algorithm A on network N*, meaning the modularity of the cover produced by algorithm A when run over network N.

5.1 Topic Extraction

The application of LDA over text content resulted in 50 topics with 100 words and their respectively probabilities. Table 5.3 presents the first 10 words of five topics obtained by LDA.

Table 5.3: Topics obtained from Plexilandia Forum.

Topic 7 **"Cable Connections"**	Topic 44 **"Components buy/sell"**	Topic 4 **"Classic questions"**	Topic 6 **"Weld Techniques"**	Topic 18 **"Amplifiers"**
cable (0.1149)	mand (0.0717)	cos (0.0424)	cort (0.0551)	tubo (0.2154)
jack (0.0472)	vend (0.0404)	electronica (0.0389)	com (0.0415)	pre (0.0591)
pata (0.0365)	precio (0.0365)	hac (0.0389)	pas (0.0413)	power (0.0556)
entr (0.0326)	mail (0.0349)	com (0.0326)	cal (0.0264)	par (0.0458)
negar (0.0237)	pag (0.0349)	bas (0.0313)	placa (0.0262)	potencia (0.0242)
conectado (0.0184)	cos (0.0212)	aprend (0.0298)	par (0.0244)	bia (0.0194)
bateria (0.0161)	pedir (0.0195)	gust (0.0217)	soldadura (0.0212)	fender (0.0189)
sal (0.0159)	envio (0.0187)	conocer (0.0202)	soldar (0.0189)	sonar (0.0176)
conectar (0.0147)	avis (0.0161)	construir (0.0161)	pista (0.0176)	rulz (0.0156)
conecta (0.0139)	traer (0.0155)	empezar (0.0153)	qued (0.0157)	preamp (0.0154)

Results of LDA were presented to administrators[3], which provided meaning to many of the topics obtained. Only 9 of the 50 topics were considered *unhelpful* to understand the community. The words of each resulted topic were used for the construction of the graphs filtered by LDA.

[3] Work developed by Sebastián Ríos

5.2 Topic-Based Social Network Visualization

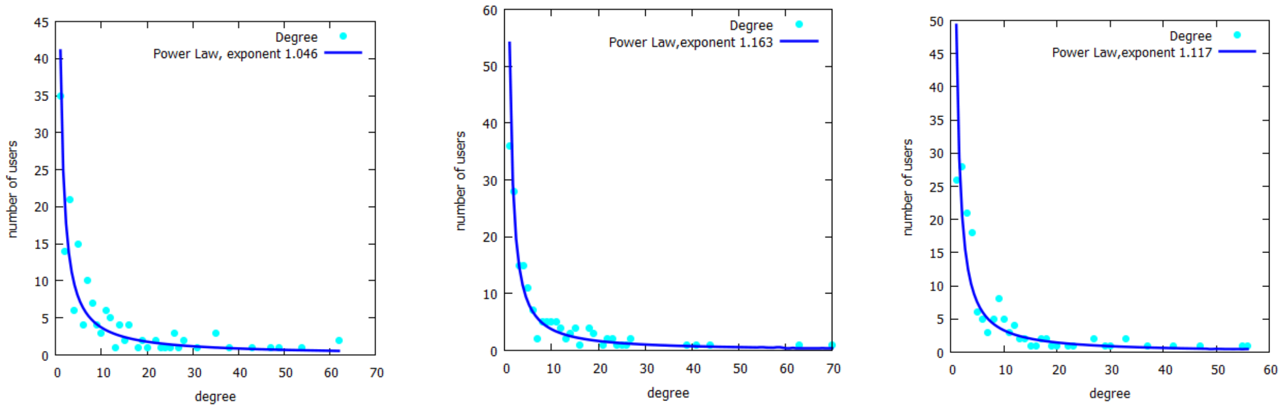
To configure the network and have the graph representation, we consider a three-month TVC extracted from plexilandia forum data, the first time frame begins in April 2009. Interaction between forum's users was modelled according all-previous-reply scheme [8] and two configurations using different graph filters by topics, according to section 3.6. Table 5.4 shows TVC properties.

Table 5.4: Plexi - Network Properties (3-Months TVC).

Period	LDA-Filter	Nodes	Edges	Density	Diameter	AVG degree
May2009 - Jun2009	F0	167	740	0.053	6.0	8.86
	F1	115	249	0.038	7.0	4.33
	F2	76	88	0.031	6.0	2.31
Jul2009 - Sept2009	F0	170	629	0.044	3.0	7.40
	F1	117	220	0.032	4.0	3.76
	F2	76	88	0.031	6.0	2.31
Oct2009 - Dec2009	F0	157	573	0.047	5.0	7.3
	F1	110	230	0.038	6.0	4.18
	F2	77	103	0.035	6.0	2.67
Jan2010 - Mar2010	F0	168	769	0.055	3.0	9.10
	F1	123	317	0.042	3.0	5.15
	F2	82	123	0.037	5.0	3.00

Figure 5.1 illustrates the distribution of the node degree in three Plexilandia's virtual communities. Similar to the observations in [11], we find that the node degree in these virtual communities also follow a power-law distribution. This means that is possible to find members with very large numbers of links, a social network's property shown in [48]. Hence, our methodology to built networks from forum data produces networks which also keep social network's properties.

Figure 5.1: A power law distribution (such as this one for the number of Web page in-links, from [22]) shows up as a blue line



(a) May09-Jun09 VC(F0)

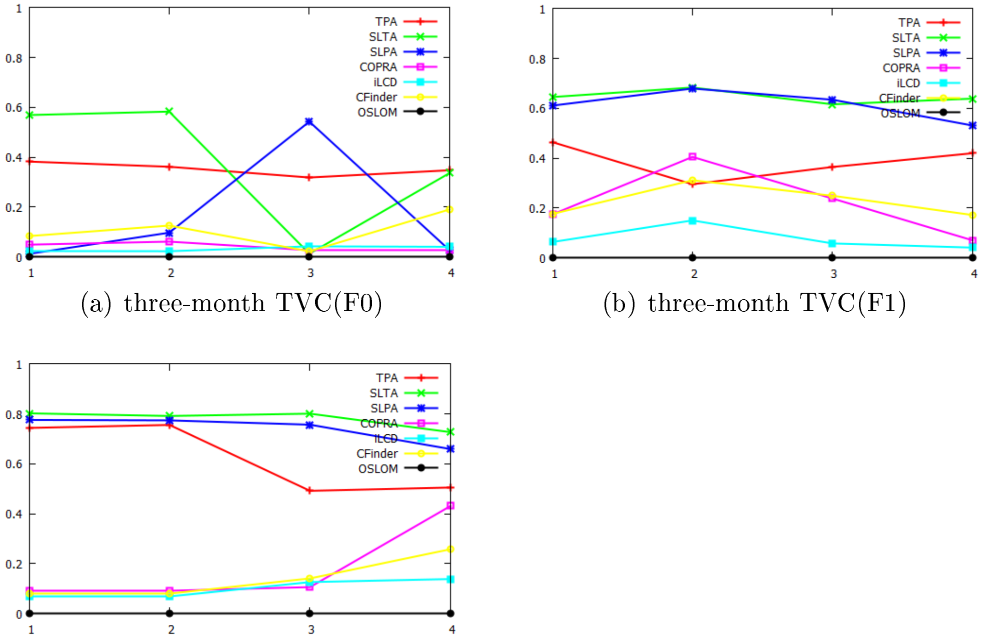
(b) Jul09-Sep09 VC(F0)

(c) Oct09-Dec09 VC(F0)

5.3 Overlapping Community Detection

Results are shown in Figure 5.2, SLTA achieves the highest Q_{ov}^{Ni} in almost all tested networks excluding the Oct 2009-Dec 2009 period. It is important to remark that most algorithms obtain similar results independently of semantic filter applied to the VCoP. However, SLTA performance is higher when the semantic filter threshold is higher, i.e, irrelevant interactions are deleted, therefore, lesser noise is processed. SLTA achieves, in average, a 5% higher performance than the best state of the art algorithm when it is applied over a VCoP. Also, it was found that the quality of the sub-community structure detected is, in average, 64% higher when the semantic filter applied on networks is increased. We excluded from this analysis results from non-filtered networks for Oct-2009 to Mar-2010 period, because of the distortion that occurs when the average was calculated. SLTA and SLPA achieve similar results because they are intrinsically related, they only differ in the initialization process. Moreover, they have similar behaviour when the semantic filter is increased, we can state that the quality of the detected community structure is higher when the semantic filter is increased for both SLTA and SLPA. With respect to TPA, this algorithm achieves a better performance when the semantic filter is increased when is applied over a VCoP. Nonetheless, his results are not as good enough as SLTA or SLPA.

Figure 5.2: Average Link Based Modularity for LDA-filtered networks. F0(0.0;0.0), F1(0.2;0.0) and F2(0.3;0.01) indicates the LDA-filter applied, F2 has a higher semantic filter than F1, F0 is a non-filtered network



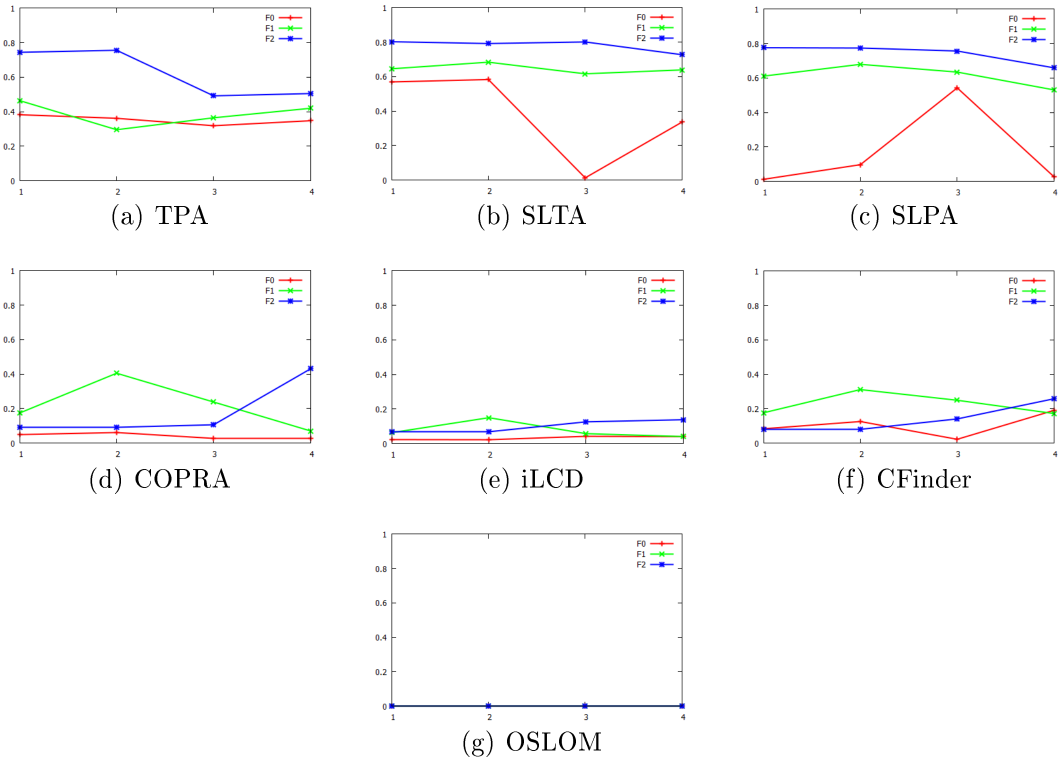
(a) three-month TVC(F0)

(b) three-month TVC(F1)

(c) three-month TVC(F2)

Table 5.5 shows the average number of detected communities with SLTA algorithm over ten iterations. Moreover, we tested if there is statistical difference in the average number of communities detected in a network with different LDA-Filters. Results show that there statistical differences in the average number of communities when we apply a higher semantic

Figure 5.3: Average Link Based Modularity for LDA-filtered networks. F0(0.0;0.0), F1(0.2;0.0) and F2(0.3;0.01) indicates the LDA-filter applied, F2 has a higher semantic filter than F1



(a) TPA

(b) SLTA

(c) SLPA

(d) COPRA

(e) iLCD

(f) CFinder

(g) OSLOM

filter. Therefore, it's possible to detect more disaggregated communities when the semantic filter is higher.

Table 5.5: PLEXI - Detected Communities.

SLTA-threshold (r)	Period	com-num-F1	com-num-F2	F-stat	P-value
r0.05	May 2009 - Jun 2009	3.668	8.395	4898.14	0
r0.05	Jul 2009 - Sept 2009	5.392	8.378	1565.922	0
r0.05	Oct 2009 - Dec 2009	2.717	5.452	4322.639	0
r0.05	Jan 2010 - Mar 2010	3.308	6.359	3070.462	0
r0.2	May 2009 - Jun 2009	3.655	7.697	3473.834	0
r0.2	Jul 2009 - Sept 2009	4.772	7.796	2101.653	0
r0.2	Oct 2009 - Dec 2009	2.738	5.137	2140.359	0
r0.2	Jan 2010 - Mar 2010	2.681	5.828	3017.122	0
r0.4	May 2009 - Jun 2009	3.47	7.028	3190.624	0
r0.4	Jul 2009 - Sept 2009	4.616	7.061	1410.131	0
r0.4	Oct 2009 - Dec 2009	2.57	5.273	2856.475	0
r0.4	Jan 2010 - Mar 2010	2.899	5.626	2436.208	0

Findings

As a final conclusion, we can state that we have changed the actual idea of community where two nodes, at least, must have an edge connecting to each other to belong the same community. In our case, we extended that that notion since – beside a connected graph – we also used the members' interaction semantic information; which means that two nodes belong to the same community if they are interested in the same topics. This allow to identify better groups to find much better communities. In Fig. 5.4, we show communities detected through SLTA for (Jul 2009-Sept 2009) TVC (LDA-filter F2(0.3,0.01)).

5.4 Community Characterization

In this section overlapping detected communities are characterized according to the methodology proposed in section 3.9. Table 5.6 shows a summary for the overlapped communities detected on networks produced after semantic filter F2(0.3,0.1) application on 2009-02-VC. For this filtered network, the node and edge number are shown. Also, the modularity after SLTA application and the percentage of overlapped nodes detected are illustrated as well, these number correspond to the cover set with the highest modularity after ten iterations of SLTA algorithm. #Com indicates the number of detected communities with 6 or more members, *Largest Com* and *Second Com* are the members' number for the largest community detected and the second largest community detected respectively. Finally, %[Lar+Sec] indicates the percentage of VC's members who belongs to the two largest communities. This percentage is calculated as follows: Let LC be the set of nodes who belong to the largest community and SC the set of nodes who belongs to the second largest community. $\%[Lar + Sec]$ can be defined as:

$$\%[Lar + Sec] = \frac{|LC \cup SC| - |LC \cap SC|}{|V|} \tag{5.1}$$

where $|V|$ is the number of nodes.

Table 5.6: 2009-02-VC, with filter F2(0.3, 0.01)

LDA-Filter	Nodes	Edges	Overlapped Nodes	Modularity	#Com	Largest Com	Second Com	%[Lar+Sec]
F2(0.3,0.1)	76	90	9,2%	0.8027	3	35	21	71%

As an example we applied the methodology described in Section 3.9 to the 2009-02-VC with a semantic filter F2(0.3, 0.01), we characterized the SLPA's output who obtained the highest modularity after ten iterations. Figure 6.8 illustrates the selected cover to describe the community structure present in 2009-02-VC F2(0.3, 0.01). Communities are detected through SLPA, multicolor nodes indicate overlap nodes. In this case we can see the new community concept where two nodes can belong to the same community even if there isn't a edge

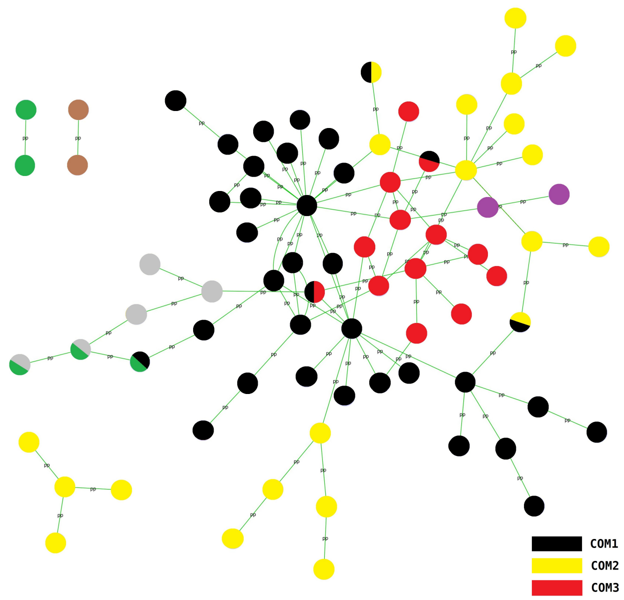
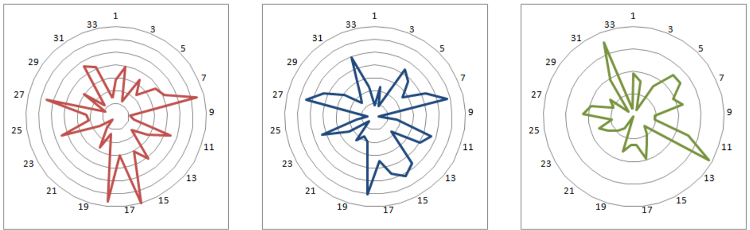


Figure 5.4: Plexilandia 2009-02 LDA-filtered network, with filter F2(0.3, 0.01). Communities are detected through SLTA, nodes with two colors indicate overlap nodes.

Figure 5.5 shows the content vector for the three largest communities detected in filtered 2000-02-VC with LDA-Filter F2(0.3, 0.01). Each content vector shows the topic distribution over each community. In this case, it's possible to see how different content vectors are when communities for the same filtered VC are analyzed. This property is expected of any semantic-based community description. Where, it expects that detected communities in the same VC are described for different topic distributions.

In figure 5.5 we already presented the semantic content vector for the three largest communities (Com1, Com2, Com3) for 2005-VC LDA-Filter F2(0.3, 0.01). Table 5.7 shows the three largest communities detected based on the 2005-VC with LDA-Filter 0.1 by TPA community detection approach. For each community, the five topics with highest score are shown. It is possible to see how different the topic which describe each community are. For this example, Com1's members post more frequently about *1) places where to buy, 2) failure*

Figure 5.5: Content community vector of the three largest communities detected through SLTA on 2009-02 VC.



(a) F2(0.3, 0.01) **Members:** 35

(b) F2(0.3, 0.01) **Members:** 21

(c) F2(0.3, 0.01) **Members:** 14

detection process, 3) electronic supplies stores, 4) components buy and sell solicitudes and 5) brands vs. prices opinion and recommendation, Com2's members about *1) places where to buy, 2) electronic supplies, 3)components buy and sell solicitudes, 4) modulation effects and 5)couple problems in distortion effects* and finally Com'3 is characterized by *1) tube amplifier, 2) couple problems in distortion effects, 3) boxes connections and construction, 4) failure explanation and 5) electronic components.* Therefore, we can observe that each community is talking about a different topic.

Table 5.7: Three largest communities for 2009-2-VC

Community	Topic	Top 10 words
Largest Community	topic16 (0,07)	vend, don, com, hay, tender, san, cos, precio, encontrar, peso
$Members = 35$	topic18 (0,066)	probar, deb, creer, pas, caso, sea, cuenta, haya, mal, nad
	topic8 (0,064)	encontrar, casa, busc, don, com, andar, royal, vend, victronic, dato
	topic27 (0,055)	mand, vend, precio, mail, pag, cos, pedir, envio, avis, traer
	topic31 (0,046)	com, sal, creer, cos, sea, barato, pen, opcion, caro, plata
Second Community	topic16 (0,061)	vend, don, com, hay, tender, san, cos, precio, encontrar, peso
$Members = 21$	topic8 (0,058)	encontrar, casa, busc, don, com, andar, royal, vend, victronic, dato
	topic27 (0,055)	mand, vend, precio, mail, pag, cos, pedir, envio, avis, traer
	topic15 (0,052)	efecto, par, delay, sonar, rever, com, chip, choru, modul, digital
	topic32 (0,049)	ruido, volumen, pote, probar, pot, cambi, sonar, potencia, baj, tono
Third Community	topic12 (0,077)	tubo, pre, power, par, potencia, bia, fender, sonar, rulz, preamp
$Members = 14$	topic32 (0,071)	ruido, volumen, pote, probar, pot, cambi, sonar, potencia, baj, tono
	topic5 (0,051)	com, habia, tenia, dia, pen, dijo, hoy, haber, hora, vez
	topic6 (0,051)	probar, pas, nad, cambi, puse, hice, tenia, despue, quem, luego
	topic11 (0,051)	conden, resistencia, poner, cambi, sonar, par, cap, diodo, com, hay

In tables 5.8,5.9 ,5.10 topic association rules extracted are presented for Com1, Com2 and Com3 respectively. The method to mine topic association rules was presented in Section 3.9. We used a rapidminer [4] implementation for FP-Growth Algorithm with default parameters for setting this algorithm. The dummy post vector was built using $\tau = 0.01$. We can see that the mined topic association rules are related to similar topics within a community, but they are different among communities.

[4] http://rapid-i.com [last accessed 17-10-2012]

Table 5.8: Association Rules extracted from 2009-02 LDA-filtered network, with filter F2(0.3, 0.01), COM1 (35 members)

Premises	**Conclusion**	**Support**	**Confidence**	**Lift**
classical new users questions	help asking	0.667	0.762	0.962
transformers for tube amplifiers	modifications to JCM Marshall amplifier	0.667	0.800	1.011
transformers for tube amplifiers	failures explanation	0.667	0.800	1.129
sound differences factors AND tube amplifier	failure detection process	0.667	0.800	1.129

For example, the rule

$$\text{topic(sound differences factors)} \wedge \text{topic(tube amplifier)} \implies \text{topic(failure detection process)}$$
$$\text{support} = 66.7\%, \ \text{confidence} = 80.0\%$$

showed in Table 5.8 indicates that of the all Plexilandia Com1's members under study, 66.7% posted about *sound differences factors* and *tube amplifier*, and have posted about *failure detection process.* There is a 80.0% probability that a Com1's member who post about *sound differences factors* and *tube amplifier* will post about *failure detection process.*

Table 5.9: Association Rules extracted from 2009-02 LDA-filtered network, with filter F2(0.3, 0.01), COM2 (21 members)

Premises	**Conclusion**	**Support**	**Confidence**	**Lift**
electronic supplies stores	places where to buy	0.727	0.800	0.978
cable connections AND electronic supplies stores	guitar construction woods	0.682	0.833	1.019
electronic supplies stores AND electronic components	handmade brands	0.682	0.938	1.146

Table 5.10: Association Rules extracted from 2009-02 LDA-filtered network, with filter F2(0.3, 0.01), COM3 (14 members)

Premises	**Conclusion**	**Support**	**Confidence**	**Lift**
distortion	effects sheets impression	0.591	0.743	1.090
transistors adjustment	failure detection process	0.614	0.750	0.971
weld techniques	cable connections	0.614	0.794	0.971
cable connections AND transistors adjustment	transformers for tube amplifiers	0.636	0.933	1.081

Chapter 6

Overlapping Community Detection on The Dark Web Portal

The method presented in Section 4.2 was evaluated over a Dark Web Portal dataset. The Dark Web Forum Portal [95] is a Web-based knowledge portal which was created based on a general framework for Web forum data integration. The portal incorporates the data collected from different international Jihadist forums. These online discussion sites are dedicated to topics relating primarily to Islamic ideology and theology. The Dark Web can be considered as a Virtual Communities of Interests (VCoI) whose members are extremists who share and comment their feelings and interests with others that support their cause. Our proposed methodology for overlapping community detection was applied in the `IslamicAwakening` English language based forum, available on ISI-KDD 2012 Website[1].

Next, an analysis of topics extracted using LDA (described in Section 3.4) is presented. Then, the network topology construction by using All-Previous-Reply-oriented structures for the whole period is described. Finally, Overlapping community detection was applied, and its results were compared against different LDA-Filtered networks.

In order to validate the proposed method (described in Section 3), it was applied using five different LDA filtered networks. To understand better the performance of the proposed algorithm, we compared SLTA and TPA with two well-know algorithms (see Table 6.1).

Table 6.1: Algorithms Included in the Experiments.

Algorithm	Complexity	Imp.
COPRA[42]	$O(vmlog(vm/n))$	Java
SLPA[93]	$O(Tm)$	C++
SLTA	$O(Tm)$	Java
TPA	$O(TKm)$	Java

We used default parameters setting for most algorithms if applicable. For TPA the maximum number of iterations T was set to 30 and for SLPA and SLTA, this parameter was set

[1] `http://www.ischool.drexel.edu/isi-kdd2012/challenge.html` [last accessed 23-04-2012]

to 100. The threshold r for SLPA and SLTA takes values in set $\{0.01, 0.05, 0.1, 0.2, 0.3, 0.4\}$ and for TPA, parameter r varies from $1/k$ to $5/k$ with an interval $1/2k$ where k is the number of extracted topics. For the extracted communities we measure and we report the maximum performance over ten repetitions for TPA, SLPA, SLTA and Copra.

We selected overlapping link based modularities $Q_{ov}^{N\mathrm{i}}$ in Eq. (2.21) as quality measure. Although modularity is a function of a cover and a network, we sometimes refer to the *modularity of algorithm A on network N*, meaning the modularity of the cover produced by algorithm A when run on network N.

6.1 Topic Extraction

There are 7 years (Apr2004 - May2010) of data available. Posts were created by 2792 members and extracted topics where realized over 127.216 posts $\mathcal{P}$ and 244.200 words in the vocabulary $\mathcal{V}$ by using a `Java` Gibbs sampling-based implementation of LDA[2] previously described in Section 3.4.

Table 6.2: Ten most relevant words with their respective conditional probabilities for five most relevant topics for all data from the `IslamicAwakening` forum.

Topic 19 **“Iraq War”**	Topic 15 **“Terrorist Attacks” (General)**	Topic 33 **“Political/Terrorist Trials”**	Topic 54 **“Religion (Allah)”**	Topic 51 **“Islamic Religion”**
government (0.0140)	kill (0.0255)	court (0.0149)	allah (0.0141)	islamic (0.0254)
war (0.0113)	police (0.0180)	guantanamo (0.0099)	prophet (0.0163)	muslim (0.0176)
military (0.0104)	soldier (0.0168)	trail (0.0080)	messenger (0.0141)	islam (0.0159)
country (0.0091)	attack (0.0139)	prison (0.0073)	peace (0.0118)	world (0.0126)
united (0.0089)	force (0.0137)	judge (0.0066)	people (0.0113)	religious (0.0107)
security (0.0083)	military (0.0109)	torture (0.0065)	lord (0.0107)	society (0.0104)
force (0.0079)	official (0.0105)	rights (0.0061)	day (0.0099)	people (0.0086)
international (0.0059)	security (0.0089)	charges (0.0057)	believer (0.0096)	law(0.0083)
official (0.0059)	report (0.0086)	government (0.0056)	bless (0.0084)	political (0.0078)
american (0.0056)	army (0.0075)	arrested (0.0056)	quran (0.0077)	western (0.0070)

The application of LDA over text content resulted in $\{10, 50, 100\}$ topics with 20 words and their respectively probabilities. In Table 6.2, the most popular topics extracted from the `IslamicAwakening` forum is presented, theses topics represent the most popular ideas posted in the forum when 100 topics are extracted.

6.2 Topic-Based Social Network Visualization

The graph has many variables which modify its configuration [8]:

1. **Time:** One dimension that was not mentioned before is time. Depending the time period of that it is wanted to be analysed; it could be possible to have monthly, annual or historic networks.
2. **Graph Filtering:** Including the traditional non-filtered graph, the other four configurations correspond to graphs filtered by topics, according to section 3.6.

[2] `http://jgibblda.sourceforge.net/` [Last accessed 23-04-2012]

Table 6.3: Jihad time virtual community of interest

Time frame	Frame Number	Fist Time Frame	Last Time Frame
One-year	7	2004	2010
Four-month	20	Jan2004-Apr2004	May2010-Aug2010
One-month	74	Apr2004	May2010

3. **Interaction topology:** According to the assumption of to *whom is replying?*. The all-previous-reply network is considered, in this configuration every reply of a thread will be a response to all posts which are already in a specific thread.

To configure the network and have the graph representation, all of these three variables have to be decided. In this thesis, three temporal virtual communities of interest are used, they are extracted from the same forum data, but they have different time frame. Table 6.3 shows the different time frames used.

Networks have been built from 2004 to 2010 using three different topic sets, and to be compared, the threshold θ takes values in set {0.0, 0.1, 0.2, 0.3, 0.4} for LDA-Filter, as explained in section 3.6. Then, for each interaction representation, the result is a graph with the members who posts in a specific period of time and has an interaction greater or equal to the filter threshold, we chose the LDA threshold to eliminate a high number of irrelevant interactions but without exclude many members from the network. For example, after applying the highest threshold over the one-year TCV, the networks have, in average, 25% less interactions but we excluded a 5% of members. Table 6.4 shows network properties after applying LDA-filter to the one-year TCV, in this case we defined the density of a network as the quotient between the edge number and all possible edges in the network. Figure 6.1 illustrates five LDA-filtered networks for 2009 using the network construction methodology proposed in 3.7, θ, $|V|$ and $|E|$ represent the LDA-Filter threshold, user's number (nodes) and interaction's number (edges) respectively, the graph density D is defined as $D = \frac{2|E|}{|V|(|V|-1)}$. We obtained similar networks using different topics sets {10,50,100}.

Figure 6.2 illustrates the distribution of the node degree in 2009-VC. Similar to the observations in [11], we find that the node degree in these virtual communities also follow a power-law distribution. This means that is possible to find members with very large numbers of links, a social network's property shown in [48]. Hence, our methodology to built networks from forum data produces networks with social network's properties.

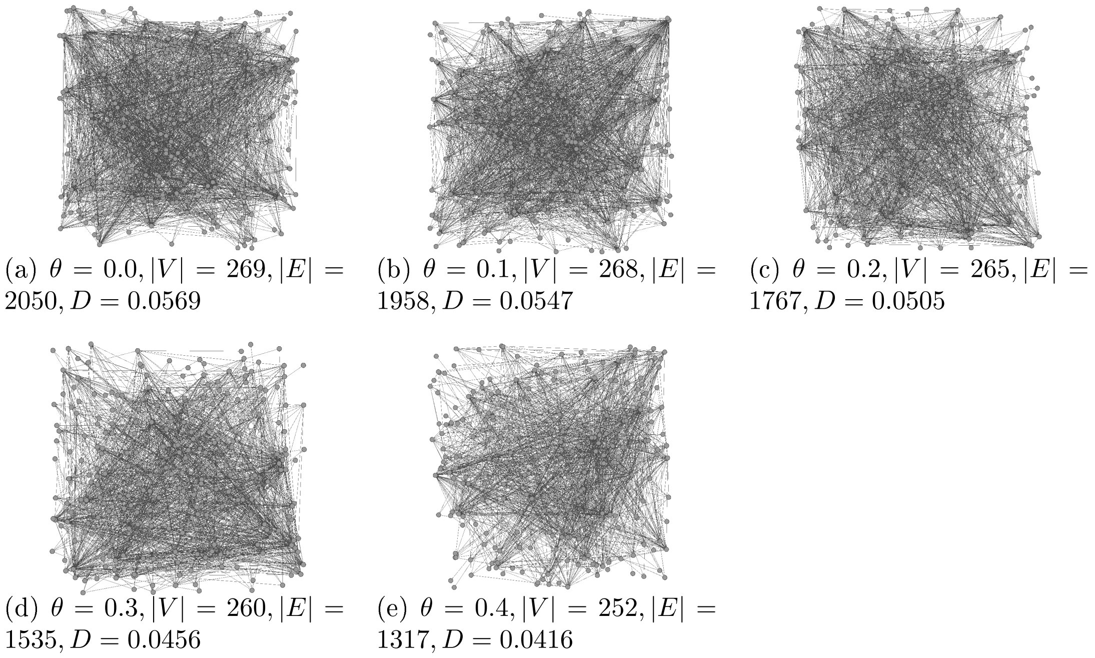
6.3 Overlapping Community Detection

Our results are shown in Figure 6.3 and Figure 6.4. Figure 6.3 shows results obtained after applying the methodology described in Chapter 3 over all four-month TVC available and Figure 6.3 only shows results for one-month TVC for the period between Jan-2008 and May-2010, all previous VCs are not considered, despite of the fact we obtained high modularity in theses VCs, because their size is not big enough and it's possible to detect communities manually, therefore the information provided by these networks is irrelevant for the purpose

Table 6.4: Network Properties (10T)

Network	Filter (θ)	Nodes	Edges	Density	AVG Degree
2008	0.0	929	22598	0.052	48.7
	0.1	925	22029	0.052	47.6
	0.2	919	20784	0.049	45.2
	0.3	904	19102	0.047	42.3
	0.4	890	17094	0.043	38.4
2009	0.0	1235	29168	0.038	47.2
	0.1	1232	28246	0.037	45.9
	0.2	1216	26399	0.036	43.4
	0.3	1203	24019	0.033	39.9
	0.4	1184	21306	0.030	36.0
2010	0.0	819	11380	0.034	27.8
	0.1	814	10959	0.033	26.9
	0.2	801	10097	0.032	25.2
	0.3	787	9029	0.029	22.9
	0.4	769	7877	0.027	20.5

Figure 6.1: Network reduction with LDA-Filter methods for May 2010.



(a) $\theta = 0.0, |V| = 269, |E| = 2050, D = 0.0569$

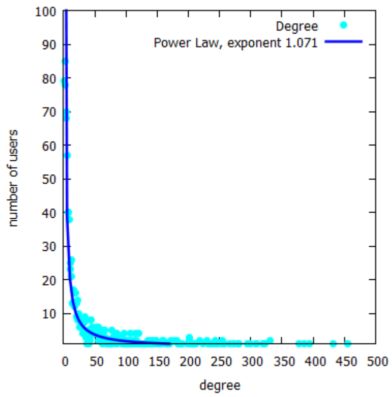
(b) $\theta = 0.1, |V| = 268, |E| = 1958, D = 0.0547$

(c) $\theta = 0.2, |V| = 265, |E| = 1767, D = 0.0505$

(d) $\theta = 0.3, |V| = 260, |E| = 1535, D = 0.0456$

(e) $\theta = 0.4, |V| = 252, |E| = 1317, D = 0.0416$

Figure 6.2: A power law distribution (such as this one for the number of Web page in-links, from [22]) shows up as a blue line



(a) 2009 VC

of this thesis. Both figures show the maximum link based modularity ($Q_{ov}^{N\text{i}}$, see 2.4.2.3) over ten repetitions for algorithms presented in Table 6.1. Despite of the fact we used three different topic sets to detect overlapping communities, we only present those results with the highest average modularity which were obtained using 50 topics.

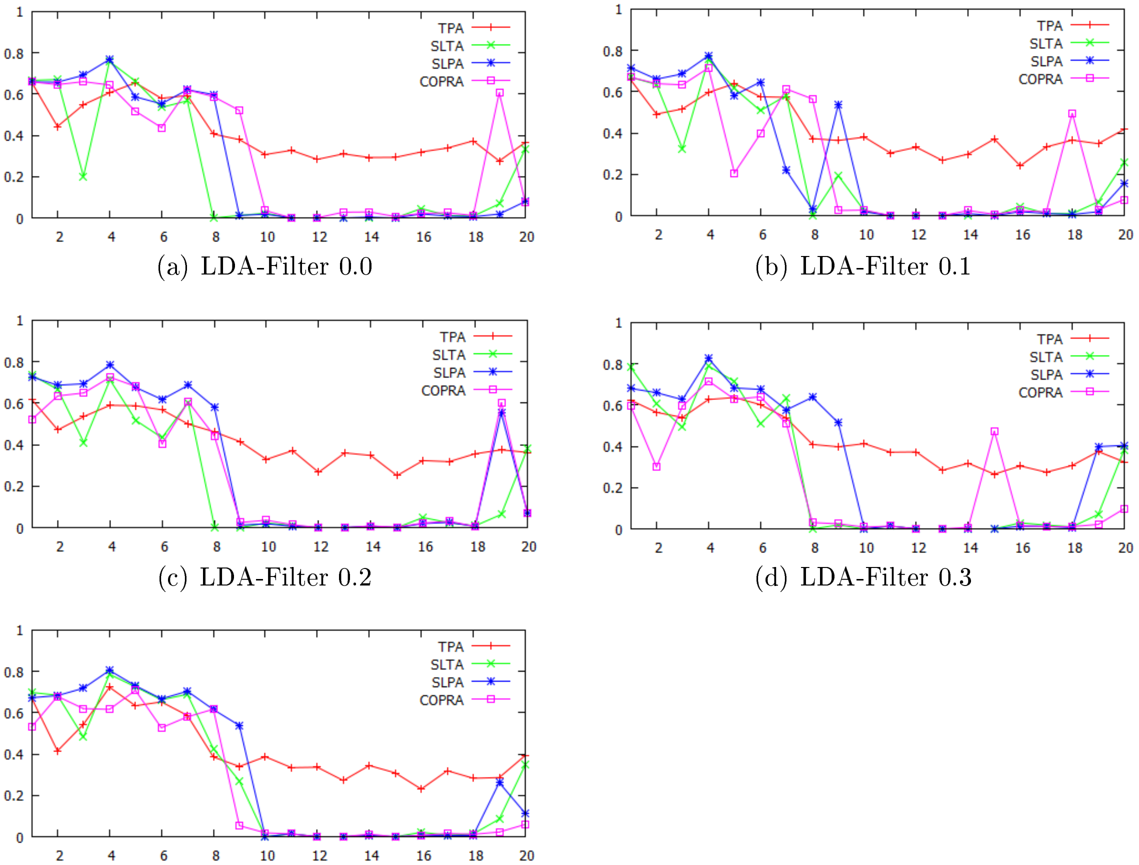
Results shows that TPA achieves, in average, a modularity measure of 0.33 while the best state of the art algorithm only achieves 0.043 when it is applied over a VCoI. Nonetheless, there is no different results when the semantic filter applied over the networks is increased. We excluded from this analysis results with a modularity measure equals zero, because of the distortion that occurs when the average was calculated.

According to the sociological theory, it is expected that VCoI's members are all related to each other because they share the same interests. Therefore, all users should belong to the same community. Most of algorithms captures this, they only find a cover which contains all members. Nonetheless, TPA detects a community structure, we will see ahead that the detected community structure defines communities which share almost all the content generated within them, and they differ only in a few topics which let us characterize each community according different topics. Most of algorithms can't detect this groups because they don't include semantic information as TPA.

In figure 6.3 we can see that any algorithm achieves a clear advantage over the others for the period between 2004 and 2006 (see VC 1 to 9 in the figure). However, TPA achieves a clear advantage for the period between 2007 and 2010 (see VC 10 to 20 in figure) except in certain cases; and we obtained similar behavior regardless of the semantic filter (θ) applied. It is worth to say that if modularity of some algorithm on a VC equals zero, it implies that the algorithm didn't find a community structure on these VC and his output is a cluster containing all VC's nodes.

When the time frame is changed for a one-month period to analyze in a more disaggregated way, a similar behavior is detected. TPA achieves the higher modularity $Q_{ov}^{N\text{i}}$ over all tested networks, theses results are shown in Figure 6.4. As previously stated, results are shown only

Figure 6.3: Algorithm comparison for four-month TVC, figures include the period between Jan-2004 and Aug-2010



(a) LDA-Filter 0.0

(b) LDA-Filter 0.1

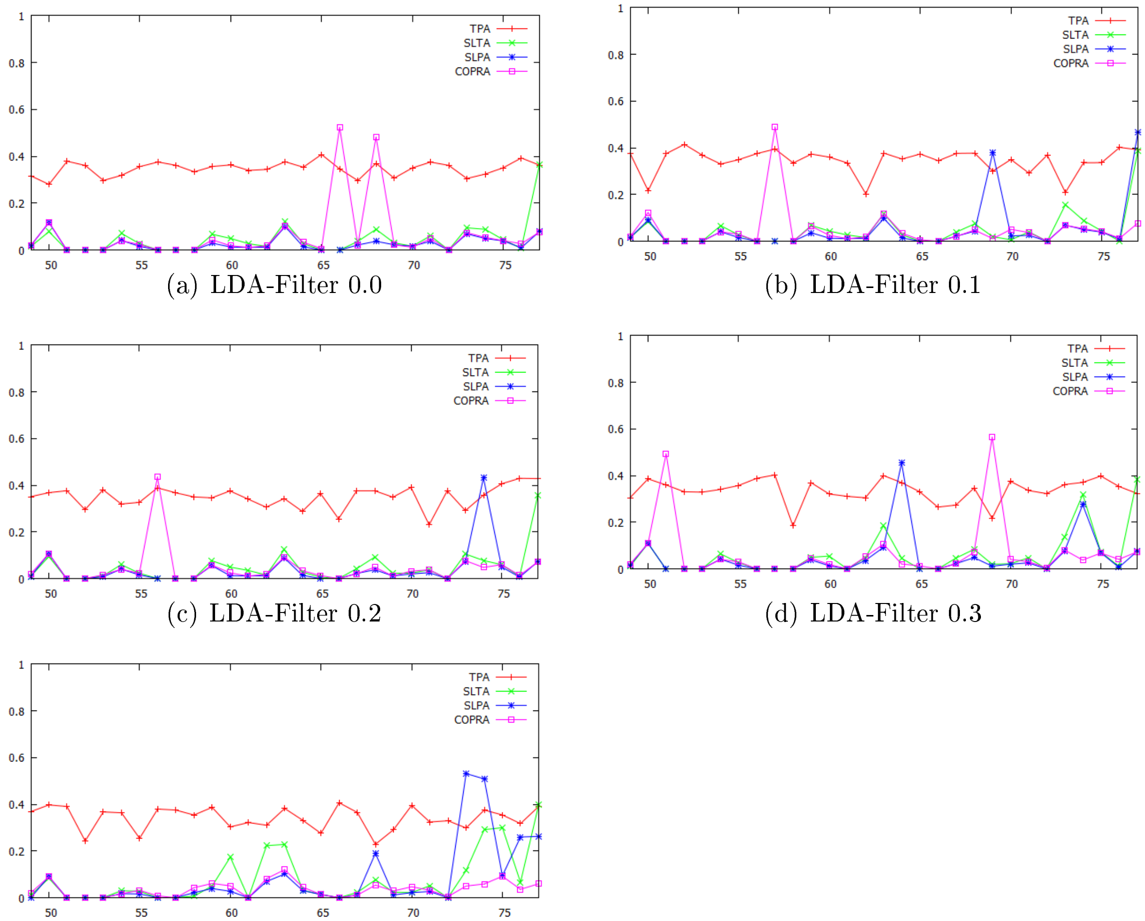
(c) LDA-Filter 0.2

(d) LDA-Filter 0.3

(e) LDA-Filter 0.4

for the period between 2008 and 2010 (see VC 49 to 77 in figure) because previous one-month VCs only has, in average, 30 nodes and 67 edges.

Figure 6.4: Algorithm comparison for one-month TVC, figures include the period between Jan-2008 and May-2010



(a) LDA-Filter 0.0

(b) LDA-Filter 0.1

(c) LDA-Filter 0.2

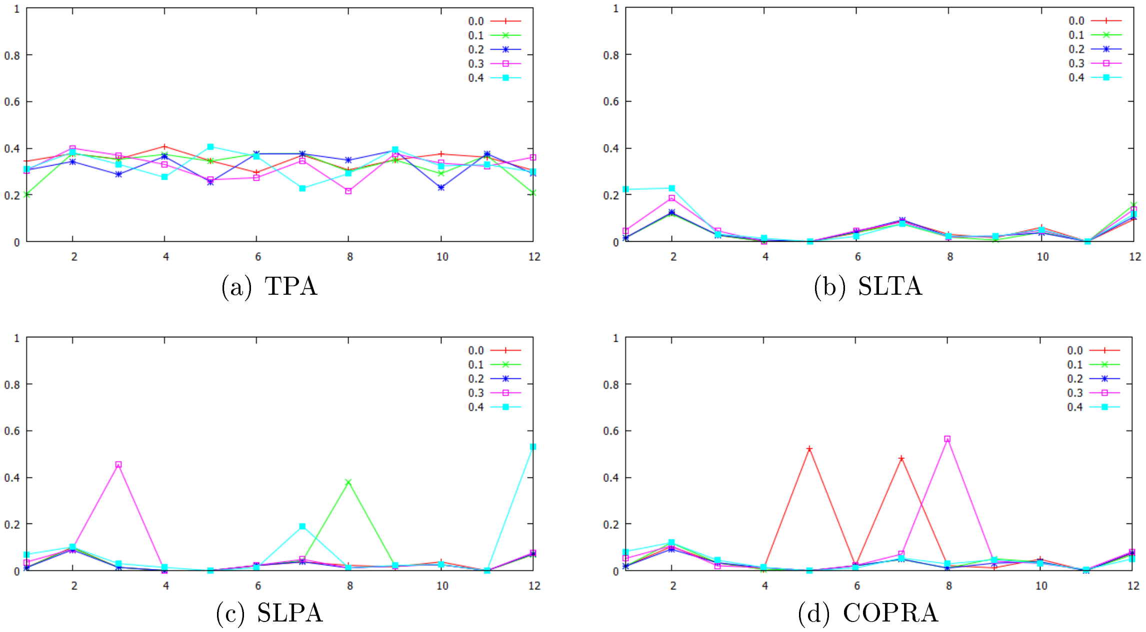
(d) LDA-Filter 0.3

(e) LDA-Filter 0.4

After analyzing the effect of different semantic filters over the quality of overlapping community detection on VCoI, it is possible to see that there are no significant differences. Unlike overlapping community detection on VCoP (see Chapter 5) where was founded that the performance of SLTA and SLPA is higher when the semantic threshold is higher, i.e, irrelevant interactions are deleted, therefore, lesser noise is processed. Figure 6.5 illustrates the effect of LDA-Filters in performance of overlapping community detection for one-month TCV for year 2009.

Table 6.5 shows the average number of detected communities with TPA algorithm over ten iterations. Unlike results obtained in a VCoP, the increment of the semantic filter doesn't increase the number of communities detected.

Figure 6.5: Comparison of LDA-Filter effects for one-month TVC, figures include the results for year 2009



(a) TPA

(b) SLTA

(c) SLPA

(d) COPRA

Table 6.5: Dark Web Portal - Detected Communities

Network	TPA-threshold	$\theta = 0.0$	$\theta = 0.1$	$\theta = 0.2$	$\theta = 0.3$	$\theta = 0.4$
Jan07-Apr07	r0.02	8.3	7.3	11.0	3.0	2.9
Jan07-Apr07	r0.03	6.3	6.2	7.5	3.3	3.4
May07-Aug07	r0.02	5.4	6.9	3.9	4.7	6.8
May07-Aug07	r0.03	6.8	12.4	4.4	18.1	11.0

Findings

As a final conclusion, we can state that we have changed the actual idea of community where two nodes, at least, must have an edge connecting to each other to belong to the same community. In our case, we extended that notion since – beside a connected graph – we also used the members' interaction semantic information; which means that two nodes belong to the same community if they are interested in the same topics. This allows us to identify better groups to find much better communities, where every community is characterized according to the methodology developed in this thesis (see 3.9) . In Figure 6.8, we show communities detected through TPA for 2005 data network (LDA-filter 0.1), multicolor nodes indicate overlap nodes. This figure illustrates the new concept of community where two nodes don't need to be connected to belong to the same community, while they share the same interest.

6.4 Community Characterization

In this section overlapping detected communities are characterized according to the methodology proposed in section 3.9. Table 6.6 shows a summary for the overlapped communities detected on networks produced after semantic filter application on 2005 VC. For each filtered network, the node and edge number are shown. Also, the modularity after TPA application and the percentage of overlapped nodes detected are illustrated as well, these numbers correspond to the cover set with the highest modularity after ten iterations of TPA algorithm. #Com indicates the number of detected communities with 6 or more members, *Largest Com* and *Second Com* are the members' number for the largest community detected and the second largest community detected respectively. Finally, %[Lar+Sec] indicates the percentage of VC's members who belong to the two largest communities. This percentage is calculated as follows: Let LC be the set of nodes who belongs to the largest community and SC the set of nodes who belongs to the second largest community. $\%[Lar + S\mathrm{ec}]$ can be defined as:

$$\%[Lar + S\mathrm{ec}] = \frac{|LC \cup SC| - |LC \cap SC|}{|V|} \tag{6.1}$$

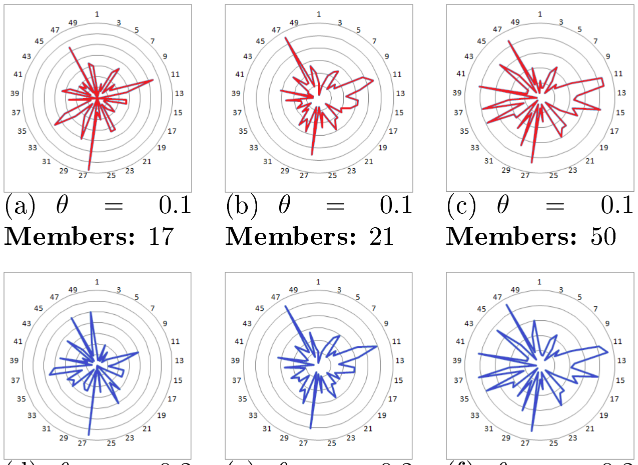
where $|V|$ is the number of nodes.

Table 6.6: 2005-VC Community Analysis

LDA-Filter	Nodes	Edges	Overlapped Nodes	Modularity	#Com	Largest Com	Second Com	%Lar+Sec
0.0	99	295	4.0%	0.50	2	51	26	74.8%
0.1	99	288	4.0%	0.57	3	50	21	71.7%
0.2	95	266	36.8%	0.52	10	53	30	63.8%
0.3	90	238	15.6%	0.57	8	48	30	75.0%
0.4	86	204	4.7%	0.51	6	61	13	82.2%

Figure 6.6 shows the content vector for the three largest communities detected in filtered 2005-VC with LDA-Filter $\theta = 0.1$ and $\theta = 0.2$. Each content vector shows the topic distribution over each community. In this case, it's possible to see how different content vectors

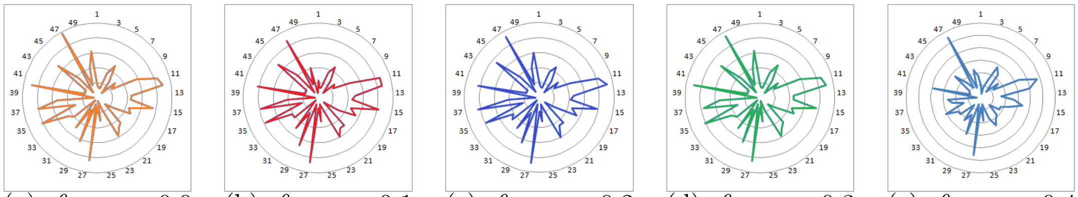
Figure 6.6: Content community vector of the three largest communities detected through TPA on 2005-VC.



(a) $\theta = 0.1$ **Members:** 17
(b) $\theta = 0.1$ **Members:** 21
(c) $\theta = 0.1$ **Members:** 50
(d) $\theta = 0.2$ **Members:** 23
(e) $\theta = 0.2$ **Members:** 30
(f) $\theta = 0.2$ **Members:** 53

are when communities for the same filtered VC are analysed, this is likely to happen because the communities were formed towards different interest topics. On the contrary, when communities with similar size are analysed across different filtered 2005-VC is possible to see that the content for each doesn't change when we apply different semantic filters. These properties are expected of any semantic-based community description. First, it is expected that detected communities in the same VC are described by different topic distributions. Finally, the content, i.e topic distribution, within a community is independent to irrelevant interactions. Therefore, topic distribution for a community doesn't change when a semantic filter is applied, since its topics are always its topics.

Figure 6.7: Content community vector of the largest communities detected through TPA on 2005-VC with several semantic filters.



(a) $\theta = 0.0$ **Members:** 51
(b) $\theta = 0.1$ **Members:** 50
(c) $\theta = 0.2$ **Members:** 53
(d) $\theta = 0.3$ **Members:** 48
(e) $\theta = 0.4$ **Members:** 61

Figure 6.7 shows that content community vector for the largest community detected after applied TPA algorithm for each LDA-Filtered network remains the same, it is possible to see that there is no change in community content vector despite of the fact we applied different semantic filters. This can be corroborated in Table 6.7 where cosine similarity between community content vectors is calculated. This example corroborates the idea explained above; the community content vector is constant even if a semantic filter is applied. This can be explained for the methodology used to characterize a community where we only included in our analysis components over a threshold. This let us to avoid the inclusion of irrelevant topics inside a community leaving the core meaning of every post.

Table 6.7: Cosine similarity for content vectors

	0.0	0.1	0.2	0.3	0.4
0.0	1	0.9819	0.9967	0.9998	0.9765
0.1	0.9819	1	0.9851	0.9814	0.9700
0.2	0.9967	0.9851	1	0.9966	0.9708
0.3	0.9998	0.9814	0.9966	1	0.9766
0.4	0.9765	0.9700	0.9708	0.9766	1

In Table 6.6 we can see that the semantic filter doesn't affect the presented measures. Regarding the two highest communities, we can see that there is not a significant change in these communities' size when we applied different semantic filters. Moreover, these two communities explain the entire VC.

As an example we applied the methodology described in Section3.9 to 2005-VC with a semantic filter $\theta = 0.1$, we characterized the TPA's output which obtained the highest modularity after ten iterations. Figure 6.8 illustrates the selected cover to describe the community structure present in 2005-VC($\theta = 0.1$). Communities are detected through TPA, multicolor nodes indicate overlap nodes. In this case we can see the new community concept where two nodes can belong to the same community even if there isn't an edge between them.

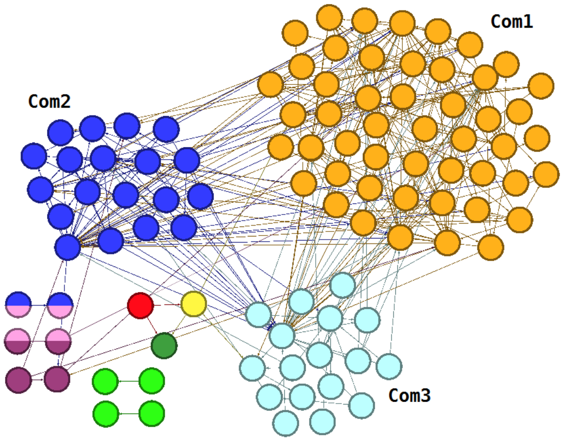


Figure 6.8: Dark Web Portal 2005 LDA-filtered network, with LDA-filter 0.1.

In figure 6.1 we already presented the semantic content vector for the three largest communities (Com1, Com2, Com3) for 2005-VC($\theta = 0.1$). Table 6.8 shows the three largest communities detected based on the 2005-VC with LDA-Filter 0.1 by TPA community detection approach. For each community, the five topics with highest score are shown. In this example, we can see that within the five most relevant topics for each community, three of them are shared for all communities *(1) Live Life Quotes, 2) Profiles (forum) and 3) Islam in General*) and just two of them let us characterize the content generated by community's members.

In tables 6.9, 6.10 ,6.11 topic association rules extracted are presented for Com1, Com2 and Com3 respectively. The method to mine topic association rules was presented in Section 3.9. We used a rapidminer [3] implementation for FP-Growth Algorithm with default parameters

[3] http://rapid-i.com [last accessed 17-10-2012]

Table 6.8: Three largest communities for 2005-VC(0.1)

Community	Topic	Top 10 words
Largest Community	topic12 (0.043)	ibn, hadith, imam, abu, book, narration, ahmad, al, weak, muhammad
$Members = 50$	topic11 (0.043)	people, time, feel, talk, person, bad, life, understand, live, start
	topic27 (0.043)	allah, people, lord, heart, love, believer, day, person, life, prophet
	topic47 (0.043)	quote, posted, originally, bro, akhi, abumuwahid, ahmed, brothermujahid, waziri, wild
	topic40 (0.041)	sheikh, al, ibn, muhammad, scholar, bin, abdullah, book, lecture, anwar
Second Community	topic47 (0.055)	quote, posted, originally, bro, akhi, abumuwahid, ahmed, brothermujahid, waziri, wild
$Members = 21$	topic11 (0.045)	people, time, feel, talk, person, bad, life, understand, live, start
	topic27 (0.045)	allah, people, lord, heart, love, believer, day, person, life, prophet,
	topic10 (0.038)	salafi, people, call, sunnah, dawah, issue, aqeedah, scholar, qutb, manhaj
	topic12 (0.033)	ibn, hadith, imam, abu, book, narration, ahmad, al, weak, muhammad
Third Communinty	topic27 (0.067)	allah, people, lord, heart, love, believer, day, person, life, prophet
$Members = 17$	topic47 (0.054)	quote, posted, originally, bro, akhi, abumuwahid, ahmed, brothermujahid, waziri, wild
	topic11 (0.054)	people, time, feel, talk, person, bad, life, understand, live, start
	topic34 (0.047)	sufi, music, sheikh, tasawwuf, love, listen, people, sound, call, singing
	topic29 (0.040)	alaykum, assalam, wa, salam, inshallah, assalamu, rahmatullah, hope, forum, false

for setting this algorithm. The dummy post vector was built using $\tau = 0.01$

Table 6.9: Association Rules extracted from 2005-VC with LDA-Filter 0.1, COM1 (50 members)

Premises	**Conclusion**	**Support**	**Confidence**	**Lift**
Parents - Life & Death, Ibn Taymiyyah -school	Muhammad the prophet	0.350	0.946	2.029
Forum topic, Islamic Law	Live life Quotes	0.350	0.940	1.738
Forum topic, Imaam Muhammad Ibn Abdul Wahhab Discussion	Jurisprudence (general)	0.355	0.935	1.922
Parents - Life & Death, Muhammad the prophet	Ibn Taymiyyah -school	0.350	0.922	2.008
Feeding behaviour	Live life Quotes	0.360	0.912	1.686
Live life Quotes, Islam in General	Parents - Life & Death	0.350	0.898	1.699
Live life Quotes, Christianism	Islamic Law	0.350	0.898	1.766

For example, the rule

$$\begin{aligned} \text{topic(Parents - Life \& Death)} \;\wedge\; \text{topic(Ibn Taymiyyah -school)} \\ \Longrightarrow\; \text{topic(Muhammad the prophet)} \\ \text{support} = 35.0\%,\ \text{confidence} = 94.6\% \end{aligned}$$

showed in Table 6.9 indicates that of the all The Dark Web Portal Com1's members under study, 35.0% post about *Parents - Life & Death* and *Ibn Taymiyyah -school*, and have posted about *Muhammad the prophet*. There is a 94.6% probability that a Com1's member who post about *Parents - Life & Death* and *Ibn Taymiyyah -school* will post about *Muhammad the prophet*.

Table 6.10: Association Rules extracted from 2005-VC with LDA-Filter 0.1, COM2 (21 members)

Premises	**Conclusion**	**Support**	**Confidence**	**Lift**
Islamic Law, Islam in General	Salafism	0.314	0.917	1.816
Profiles (forum), Disbelievers	Jurisprudence (general)	0.314	0.891	1.898
Islamic Law, Live life Quotes	Salafism	0.320	0.886	1.753
Islamic Law, Jurisprudence (general)	Salafism	0.314	0.884	1.750
Salafism, Islam in General	Islamic Law	0.314	0.865	1.713
Christianism	Islamic Law	0.322	0.850	1.683
Terrorist trial	Islamic Law	0.332	0.843	1.669
British Islamic communities	Islamic Law	0.348	0.839	1.660

Table 6.11: Association Rules extracted from 2005-VC with LDA-Filter 0.1, COM3 (17 members)

Premises	**Conclusion**	**Support**	**Confidence**	**Lift**
India	Shiite - Sunnis	0.396	0.908	1.853
India	Islam in General	0.389	0.892	1.621
Terrorist Investigations	Live life Quotes	0.396	0.881	1.562
Prohibited or not? Discussion	Islam in General	0.396	0.881	1.600
Prohibited or not? Discussion	Parents - Life & Death	0.396	0.881	1.749
Islamic blessing	Islam in General	0.389	0.866	1.573
Terrorist Investigations	British Islamic communities	0.389	0.866	1.767
U.S Presidential elections	Live life Quotes	0.403	0.857	1.520
Disbelievers	Allah (Doctrinal Issue)	0.416	0.849	1.710

Conclusion and Future Work

Overlapping community detection problem is commonly solved with SNA techniques. Many algorithms have been designed to discover networks' community structure. However, these algorithm only consider topological characteristic, ignoring the semantic information.

We have presented a novel way of finding communities that it is not only efficient for handling large graphs, but also takes into account the fact that nodes in a graph might belong to several communities at the same time. We have seen that it is possible to include semantic information to enhance overlapping community detection.

In this thesis, a hybrid approach was proposed which integrates both topic identification using the advantages of LDA and topology-based community detection techniques. Two layers of enriched information are constructed based on the forums' data: one is the topology of VCs and the other is the topic model of the communities overlaid on the VCs. This hybrid approach uses the advantages of LDA to build improved and filtered networks. Then, two novel algorithms were developed to include two different overlapping community detection approaches, the first one considers the graph structure of the network (topology-based overlapping community detection approach); the other one takes the textual information of the network nodes into consideration (topic-based overlapping community detection approach). The main objective of this work was to use this hybrid approach, text mining techniques and social network analysis, in order to improve the detection of overlapping communities using an algorithm based on this approach. This algorithm was designed and developed in this thesis. As presented in sections 5.3, 6.3, the main objective was accomplish, following the specific objectives stated in section 1.2.2. Each objective is fulfilled and their contribution to this thesis is presented in the following list.

1. In section 2.4.1. A research about the state of the art in overlapping community detection was presented. We categorized existing algorithms into five major classes, summarized as follows: (I) clique percolation based algorithms, which are based on the assumption that a community consists of fully connected subgraphs and detecting overlapping communities by searching for adjacent cliques; (II) line graph and link partitioning based algorithms, whose idea is to partition links instead of nodes to allow multiple memberships; (III) local expansion and optimization based algorithms are based on growing a natural community by maximizing a benefit function that characterizes the quality of a densely connected group of nodes; (IV) fuzzy algorithms, which construct a soft membership vector or belonging factor explicitly; (V) dynamical algorithms, which are based on principles from statistical mechanism or simulations.

2. In section 4.1 we tested the state of the art algorithms in synthetic networks using LFR-Benchmark. Results showed that that SLPA achieves the best performance over all tested networks. This motivated us to design an SLPA-based algorithm to detect overlapping communities.

3. In chapter 4 we designed and developed two novel algorithms to solve the overlapping community detection problem. First, we introduced a dynamic interaction process, SLTA (Speaker-Listener Topic Propagation Algorithm) as a modified version of SLPA algorithm which includes semantic information to enhance the overlapping community detection. This is an efficient (linear time) and effective overlapping community detection algorithm. Secondly, TPA (Topic Propagation Algorithm) is presented as a dynamic process which updates the membership vector of each node asynchronously. Both Algorithms allow us to analyze different kinds of community structures, such as disjoint communities and overlapping communities. The application of these algorithm over virtual communities shows that SLTA and TPA achieves higher modularity in almost all tested networks. Results shows that SLTA achieves, in average, a 5% higher performance than the best state of the art algorithm when it is applied over a VCoP. Also, it was found that the quality of the subcommunity structure detected is, in average, 64% higher when the semantic filter applied on networks is increased. With regard to TPA, this algorithms achieves, in average, a 660% higher performance that the best state of the art algorithm when it is applied over a VCoI. Nonetheless, there is no different results when the semantic filter applied over the networks is increased.

   According to the underlying sociological theory, we expected to detect a high modularity sub-community structure in a VCoP, where each sub-community represents a group of people who share about specific topics. Besides, when our methodology is applied in order to detect overlapping communities over a VCoI, it is expectable to detect no sub-community structure because all VCoI's members only share ideas and thoughts about a common interest or passion but they don't share their knowledge and expertise to learn more about an specific topic. After applied our methodology we made valid the underlying theory finding that our algorithms detect better a sub-community structure when the semantic filter is increased over a VCoP and the algorithms don't detect a sub-community structure when they are applied on a VCoI. Only TPA is capable to find a sub-community structure on a VCoI, this is because TPA includes semantic information to detect overlapping communities and it is able to capture groups which differ only in a few topics. Results seem to show us a new way to detect the type of community analyzed according the performance of both algorithms, TPA and SLTA.

4. Chapter 3 presents a SNA-KDD based methodology to detect overlapping communities which include data selection, preprocessing data, topic extraction and overlapping community detection with the algorithms presented in this thesis.

5. In section 3.9, a methodology to characterize overlapping sub-communities detected was presented, this methodology let to characterize a sub-community according what the community is talking about, and which topics could be of interest for the community members.

Future Work

There are several interesting directions for future work. In this work, the data sets to which the LDA model was applied do not have *static* topics; they are instead *dynamic*. The data was collected over time, and generally patters present in the early part of the collection are not in effect later. Topics rise and fall in prominence; they split apart; they merge to form new topics; words change their correlations. LDA model isn't aware of these dependencies on document timestamps. Therefore, a model that explicitly models time jointly with word co-occurrence patterns is needed. Appendix C presents a suggested model to incorporate time in topic analysis.

In this thesis, we changed the actual idea of community where two nodes must have an edge connecting to each other to belong to the same community. This idea impact in the quality measure used in this thesis, a preliminary study (see Appendix B) indicates that, despite of the fact that our algorithms achieves the best performance among all the algorithms evaluated, we are underestimating our results. Therefore, a new version of modularity has to be formulated to include this new concept of community.

In Chapter 4, this thesis contributes with two fast algorithm for overlapping community detection in large-scale networks. The results are also mainly on undirected unweighted networks. However, for many real-world applications, the weight and direction associated with each edge bear very important information and should be taken into account.

In order to enhance the performance of TPA algorithm, new interaction rules has to be proven.

For network filtering, SLTA and TPA post-processing, and community characterization a thresholding procedure was performed. In this work, several thresholds were used for each one. A parameter estimation method which optimizes the output in every step where a threshold is used has to be developed.

Applying both algorithms over a new virtual community would let us to detect the type of community analyzed (VCoP or VCoI). This hypothesis has to be validated applying the proposed methodology in this thesis over new virtual communities.

Detecting when a member will leave a community and it will be part of another one is an interesting topic for future work. This can be studied through the analysis of similarity between a user's content vector and the communities' content vector. Changes in topic distribution of the content generated by an user will be useful to detect when a member is leaving a community.

From the computational point of view, it would be interesting to design a system to keep and update the community information once it has been calculated, in order to respond to successive queries more efficiently, and to explore the *hierarchy* of the community structure in the graph.

Another interesting research line for the next future would be studying strategies for the efficient parallelization of our algorithms, making it possible to handle even larger graphs.

A first approach to parallelize these algorithms is to split the iterative process in different threads and then, merge the memories/membership vector of each node. Another approach is to split the analized network in overlapping connected sub-networks and then, run TPA/SLTA over these sub-networks and then merge the memories/membership vector of each node.

Acknowledgment

Authors would like to thank the continuous support of Instituto Sistemas Complejos de Ingeniería (ICM: P-05-004- F, CONICYT: FBO16); Initiation into Research Founding (FONDECYT), project code 11090188, entitled “Semantic Web Mining Techniques to Study Enhancements of Virtual Communities” ; and the Web Intelligence Research Group (wi.dii.uchile.cl).

Moreover, this thesis would not have been accomplished without funds supported of CONICYT, scholarship code 22121473.

Conferences and Workshops

[1] R. Muñoz and S. Ríos. Overlapping community detection in VCoP using topic models. *16th International Conference on Knowledge-Based and Intelligent Information Engineering Systems, KES2012,* 2012.

[2] S. Ríos and R. Muñoz. Dark web portal overlapping community detection based on topic model. In *Proceedings of the ACM SIGKDD Workshop on Intelligence and Security Informatics*, ISI-KDD'12, pages 2:1-2:7, New York, NY, USA, 2012. ACM.

Bibliography

[1] Neil Hurley Aaron McDaid. Detecting highly overlapping communities with model-based overlapping seed expansion. Overlapping, scalable, community finding algorithm, March 2010.

[2] Balazs Adamcsek, Gergely Palla, Illés J. Farkas, Imre Dérenyi, and Tamás Vicsek. Cfinder: Locating cliques and overlapping modules in biological networks. *BIOINFORMATICS*, 22:1021, 2006.

[3] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *ACM SIGMOD Intl. Conf. Management of Data*, May 1993.

[4] Yong-Yeol Ahn, James P. Bagrow, and Sune Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466:761–764, August 2010.

[5] S. Aiman Moyaid, D. Pdd, and A. Azween. A Comparative Study of FP-growth Variations. *international journal of computer science and network security*, 9(5):266–272, 2009.

[6] Réka Albert, Hawoong Jeong, and Albert-László Barabási. The diameter of the world wide web. *CoRR*, cond-mat/9907038, 1999.

[7] H Alvarez, SA Ríos, E Merlo, F Aguilera, and L Guerrero. Enhancing sna with a concept-based text mining approach to discover key members on a vcop. page to appear, Mar 2010.

[8] Héctor Alvarez. Detección de miembros clave en una comunidad virtual de prática mediante análisis de redes sociales y minería de datos avanzada. Master's thesis, University of Chile, 2010.

[9] Lars Backstrom, Paolo Boldi, Marco Rosa, Johan Ugander, and Sebastiano Vigna. Four degrees of separation. *CoRR*, abs/1111.4570, 2011.

[10] Albert-László Barabási. *Linked: The New Science of Networks*. Basic Books, 1st edition, May 2002.

[11] Albert-László Barabási and Eric Bonabeau. Scale-free networks. *Sci. Am.*, 288(5):50–59, 2003.

[12] Jeffrey Baumes, Mark Goldberg, Mukkai Krishnamoorthy, Malik Magdon-Ismail, and Nathan Preston. Finding communities by clustering a graph into overlapping subgraphs. pages 97–104, 02 2005.

[13] Jeffrey Baumes, Mark Goldberg, and Malik Magdon-ismail. Efficient identification of overlapping communities. In *In IEEE International Conference on Intelligence and Security Informatics (ISI*, pages 27–36, 2005.

[14] Shea Bennett. Facebook, twitter, youtube, pinterest and the social media revolution [infographic] [online: accessed 12-11-2012]. `http://www.mediabistro.com/alltwitter/social-media-revolution_b30974`, November 2012.

[15] Shea Bennett. The snowball project [online: accessed 19-11-2012]. `http://snowball.tartarus.org/`, November 2012.

[16] Shea Bennett. Twitter, facebook, google, youtube -what happens on the internet every 60 seconds? [infographic] [online: accessed 12-11-2012]. `http://www.mediabistro.com/alltwitter/data-never-sleeps_b24551`, June 2012.

[17] Shea Bennett. Twitter facts and figures 2012 [infographic] [online: accessed 12-11-2012]. `http://www.mediabistro.com/alltwitter/twitter-stats-2012_b30967/`, November 2012.

[18] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[19] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[20] D.G. Bobrow and J. Whalen. Community knowledge sharing in practice: the eureka story. *Reflections*, 4(2):47–59, 2002.

[21] Ronald L. Breiger. The duality of persons and groups. *Social Forces*, 53(2):pp. 181–190, 1974.

[22] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. *Comput. Netw.*, (1-6):309–320, June.

[23] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the web: Experiments and models. In *Proceedings of the Ninth Conference on World Wide Web*, pages 309–320, Amsterdam, Netherlands, May 2000. ACM Press.

[24] Rémy Cazabet, Frédéric Amblard, and Chihab Hanachi. Detection of overlapping communities in dynamical social networks. In *Proceedings of the 2010 IEEE Second International Conference on Social Computing*, SOCIALCOM '10, pages 309–314, Washington, DC, USA, 2010. IEEE Computer Society.

[25] Deepayan Chakrabarti and Christos Faloutsos. Graph mining: Laws, generators, and algorithms. *ACM Computing Surveys CSUR*, 38(1), 2006.

[26] A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Physical Review E*, 70:066111, 2004.

[27] Thomas M. Cover and Joy A. Thomas. *Elements of information theory.* Wiley-Interscience, New York, NY, USA, 1991.

[28] Lautaro Cuadra. Metodología de búsqueda de sub-comunidades mediante análisis de redes sociales y minería de datos. Master's thesis, University of Chile, 2011.

[29] Katy Daniells. Infographic: Instagram statistics 2012 [online: accessed 12-11-2012]. `http://www.digitalbuzzblog.com/infographic-instagram-stats/`, May 2012.

[30] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005:P09008, 2005.

[31] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407, 1990.

[32] Ying Ding. Community detection: Topological vs. topical. *J. Informetrics*, 5(4):498–514, 2011.

[33] The Economist. Untangling the social web software [online: accessed 12-11-2012]. `http://www.economist.com/node/16910031`, September 2010.

[34] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, SIGCOMM '99, pages 251–262, New York, NY, USA, 1999. ACM.

[35] S. Fortunato and M. Barthélemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36, 2007.

[36] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75–174, 2010.

[37] L. C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41, 1977.

[38] Linton C. Freeman. Centrality in social networks conceptual clarification. *Social Networks*, 1(3):215 – 239, 1978-1979.

[39] Linton C. Freeman. *The Development of Social Network Analysis: A Study in the Sociology of Science.* Empirical Press, 2004.

[40] M Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proc Natl Acad Sci USA*, 99(12):7821–6, June 2002.

[41] Steve Gregory. Finding overlapping communities using disjoint community detection algorithms. In Santo Fortunato, Giuseppe Mangioni, Ronaldo Menezes, and Vincenzo Nicosia, editors, *Complex Networks*, volume 207 of *Studies in Computational Intelligence*, pages 47–61. Springer, Berlin / Heidelberg, 2009.

[42] Steve Gregory. Finding overlapping communities in networks by label propagation. *New Journal of Physics*, 2010.

[43] Steve Gregory. Fuzzy overlapping communities in networks. *Journal of Statistical Mechanics: Theory and Experiment 2011*, 2011.

[44] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Suppl. 1):5228–5235, April 2004.

[45] R. Guimera, M. Sales-Pardo, and L.A.N. Amaral. Modularity from fluctuations in random graphs and complex networks. *Physical Review E*, 70(2):025101, 2004.

[46] Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. *SIGMOD Rec.*, 29(2):1–12, May 2000.

[47] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 50–57, New York, NY, USA, 1999. ACM.

[48] David Easley Jon Kleinberg. *Networks, Crowds, and Markets ; Reasoning about a Highly Connected World*. Cambridge University Press, [S.l.], 2010.

[49] Brian Karrer, Elizaveta Levina, and M. E. J. Newman. Robustness of community structure in networks. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 77(4):046119+, 2008.

[50] Won Kim, Ok-Ran Jeong, and Sang-Won Lee. On social web sites. *Inf. Syst.*, 35(2):215–236, 2010.

[51] Jon Kleinberg. Navigation in a small world. 46:845, August 2000.

[52] Miia Kosonen. Knowledge sharing in virtual communities – a review of the empirical research. *Int. J. Web Based Communities*, 5(2):144–163, 2009.

[53] Andrea Lancichinetti and Santo Fortunato. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Phys. Rev. E*, 80(1):016118, July 2009.

[54] Andrea Lancichinetti and Santo Fortunato. Community detection algorithms: A comparative analysis. *Physical Review E*, 80(5):056117, November 2009.

[55] Andrea Lancichinetti, Santo Fortunato, and Janos Kertesz. Detecting the overlapping and hierarchical community structure of complex networks, 2008. cite arxiv:0802.1218 Comment: 20 pages, 8 figures. Final version published on New Journal of Physics.

[56] Andrea Lancichinetti, Santo Fortunato, and János Kertesz. Detecting the overlapping and hierarchical community structure in complex network. *New Journal of Physics*, 2009.

[57] Andrea Lancichinetti, Filippo Radicchi, José J. Ramasco, and Santo Fortunato. Finding statistically significant communities in networks. *PLoS ONE*, 6(4):e18961, 04 2011.

[58] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, KDD '05, pages 177–187, New York, NY, USA, 2005. ACM.

[59] Jure Leskovec, Kevin J. Lang, Anirban Dasgupta, and Michael W. Mahoney. Statistical properties of community structure in large social and information networks. In Jinpeng Huai, Robin Chen, Hsiao-Wuen Hon, Yunhao Liu, Wei-Ying Ma, Andrew Tomkins, and Xiaodong Zhang, editors, *WWW*, pages 695–704. ACM, 2008.

[60] Jure Leskovec, Kevin J. Lang, Anirban Dasgupta, and Michael W. Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1):29–123, 2009.

[61] Jure Leskovec, Kevin J. Lang, and Michael Mahoney. Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 631–640, New York, NY, USA, 2010. ACM.

[62] Gaston L'Huillier, Sebastián A. Ríos, Héctor Alvarez, and Felipe Aguilera. Topic-based social network analysis for virtual communities of interests in the dark web. In *ACM SIGKDD Workshop on Intelligence and Security Informatics*, ISI-KDD '10, pages 9:1–9:9, New York, NY, USA, 2010. ACM.

[63] D. J. C. Mackay. *Information theory, inference, and learning algorithms.* Cambridge University Press, Cambridge, 2003.

[64] Eduardo Merlo. Identificación de comunidades de copia en instituciones educacionales mediante el análisis de redes sociales sobre documentos digitales. Master's thesis, University of Chile, 2010.

[65] S. Milgram. The small world problem. *Psychology Today*, 61:60–67, 1967.

[66] T.P. Minka and John Lafferty. Expectation-Propagation for the Generative Aspect Model. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*, pages 352–359, 2002.

[67] R. Muñoz and S. Ríos. Overlapping community detection in vcop using topic mod-

els. *16th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems, KES2012*, 2012.

[68] Tamás Nepusz, Andrea Petróczi, László Négyessy, and Fülöp Bazsó. Fuzzy communities and the concept of bridgeness in complex networks. *Phys. Rev. E*, 77:016107, Jan 2008.

[69] M. E. J. Newman. The structure and function of complex networks. *SIAM REVIEW*, 45:167–256, 2003.

[70] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69:026113, Feb 2004.

[71] M. E. J. Newman and Juyong Park. Why social networks are different from other types of networks. *Phys. Rev. E*, 68:036122, September 2003.

[72] M.E.J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69, September 2003.

[73] Vicenzo Nicosia, Giuseppe Mangioni, Vicenza Carchiolo, and Michele Malgeri. Extending the definition of modularity to directed graphs with overlapping communities. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(03):P03024, 2009.

[74] Tore Opsahl, Filip Agneessens, and John Skvoretz. Node centrality in weighted networks: Generalizing degree and shortest paths. *Social Networks*, 32(3):245 – 251, 2010.

[75] Juyong Park1 and M. E. J. Newman. The origin of degree correlations in the internet and other networks. 68(026112), 2003.

[76] Constance E. Porter. A typology of virtual communities: A multi-disciplinary foundation for future research. *Journal of Computer-Mediated Communication*, 10(1):00, 2004.

[77] M. F. Porter. Readings in information retrieval. chapter An algorithm for suffix stripping, pages 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

[78] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi. Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences*, 101(9):2658, 2004.

[79] Sebastián A Ríos, Felipe Aguilera, and Luis Guerrero. Virtual communities of practice's purpose evolution analysis using a concept-based mining approach. *Knowledge-Based Intelligent Information and Engineering Systems - Part II; Lecture Notes in Computer Science*, 5712:480–489, 2009.

[80] Sebastián A. Ríos and Ricardo Muñoz. Dark web portal overlapping community detection based on topic models. In *Proceedings of the ACM SIGKDD Workshop on Intelligence and Security Informatics*, ISI-KDD '12, pages 2:1–2:7, New York, NY, USA, 2012. ACM.

[81] Granovetter Mark S. The strength of weak ties. 1973.

[82] Gert Sabidussi. The centrality index of a graph. *Psychometrika*, 31(4):581–603, December 1966.

[83] G Salton, A Wong, and C S Yang. A vector space model for automatic indexing. *Commun. ACM*, Vol. 18(11):613–620, 1975.

[84] John Scott. *Social network analysis: A handbook.* SAGE, London, 2nd edition, 2000.

[85] Gergely Palla; Imre Derényi; Illés Farkas; Tamás Vicsek. Handbook of large-scale random networks, chapter 9.

[86] Gergely Palla; Imre Derényi; Illés Farkas; Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435, 2005.

[87] Xuerui Wang and Andrew McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 424–433, New York, NY, USA, 2006. ACM.

[88] Duncan J. Watts. Networks, dynamics, and the small-world phenomenon. *The American Journal of Sociology*, 105(2):493–527, 1999.

[89] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393(6684):440–442, June 1998.

[90] Etienne Wenger, Richard Arnold McDermott, and William Snyder. *Cultivating communities of practice: a guide to managing knowledge.* Harvard Business Press, 2002.

[91] Jierui Xie. *Agent-based dynamics models for opinion spreading and community detection in large-scale social networks.* PhD thesis, Rensselaer Polytechnic Institute, Troy, New York, May 2012.

[92] Jierui Xie, Stephen Kelley, and Boleslaw K. Szymanski. Overlapping community detection in networks: the state of the art and comparative study. *CoRR*, abs/1110.5813, 2011.

[93] Jierui Xie, B. K. Szymanski, and Xiaoming Liu. Slpa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. In *ICDM 2011 Workshop on DMCCI*, 2011.

[94] Erjia Yan, Ying Ding, Stasa Milojevic, and Cassidy R. Sugimoto. Topics in dynamic research communities: An exploratory study for the field of information retrieval. *Journal of Informetrics*, 6(1):140 – 153, 2012.

[95] Yulei Zhang, Shuo Zeng, Li Fan, Yan Dang, Catherine A. Larson, and Hsinchun Chen. Dark web forums portal: searching and analyzing jihadist forums. In *Proceedings of the 2009 IEEE international conference on Intelligence and security informatics*, ISI'09, pages 71–76, Piscataway, NJ, USA, 2009. IEEE Press.

Appendix A

Topics extracted

A.1 Plexilandia topics extracted

Table A.1: Plexilandia topic extracted

Topic_id	Meaning
1	distortion
2	not helpful
3	effects sheets impression
4	classical new users questions
5	help asking
6	weld techniques
7	cable connections
8	not helpful
9	boxes connection and construction
10	not helpful
11	failures explanation
12	congrats
13	guitar construction woods
14	electronic supplies stores
15	references to english books and general texts
16	modifies to JCM Marshall amplifier
17	electronic components
18	tube amplifier
19	previous posts replies conversations
20	not helpful
21	handmade brands
22	not helpful
23	plexilandia website
24	images and videos of construction advances
25	modulation effects
26	places where to buy

27	sound differences factors
28	not helpful
29	failure detection process
30	transistors adjustment
31	band sound appreciation
32	transformers for tube amplifiers
33	tube amplifiers rectification
34	not helpful
35	many words bad written
36	community coexistence norms
37	effects boxes and amplifiers plate-holder
38	effects schemes
39	different effects
40	effects interrupters
41	not helpful
42	plexi-meeting
43	not helpful
44	components buy and sale solicitudes
45	guitars brand and models
46	distortion stages
47	software and hardware for sound applications
48	brands vs prices opinion and recommendation
49	couple problems in distortion effects
50	acoustic isolation

A.2 Dark Web Portal topic extracted

Table A.2: Dark Web Portal topic extracted

Topic_id	Meaning
1	U.S Presidential elections
2	Somalia
3	Contact in social life
4	Qur'aan lecture
5	India
6	Islamic acknowledgements
7	Earth Creation (Science vs Theology)
8	Shiite - Sunnis
9	World business (politics)
10	Salafism
11	Live life Quotes
12	Ibn Taymiyyah -school
13	Attacks & deaths (Jihad)
14	Islamic Law
15	Allah (Doctrinal Issue)
16	Family behaviour

17	Terrorist Investigations
18	Jokes
19	Forum topic
20	Christianism
21	Islamic blessing
22	Al qaeda - Attacks
23	Gaza Strip
24	Afghanistan-Pakistan troops
25	Jihad - killing
26	European countries
27	Islam in General
28	Jihad in the Quran and Hadith
29	Islamic greetings
30	Glory be to Allah
31	Schools
32	Feeding behaviour
33	Muhammad the prophet
34	Sufism (mystical Islam)
35	British Islamic communities
36	Abuz Zubair Discussion
37	links to content (forum topic)
38	Disbelievers
39	Hamza Yusuf Discussion
40	Imaam Muhammad Ibn Abdul Wahhab Discussion
41	Ramadan
42	World celebrations
43	Terrorist trial
44	Irak war
45	The Revival of Islam
46	Prohibited or not? Discussion
47	Profiles (forum)
48	Saudi Arabia
49	Jurisprudence (general)
50	Parents - Life & Death

Appendix B

Modularity analysis

Because we changed the current idea of community is necessary to estimate how much this new idea affects to the modularity measures. First, we have to remember the modularity formula (See Equation B.1)

$$Q = \frac{1}{2m} \sum_{c} \sum_{\mathrm{i},j \in c} \left[A_{\mathrm{i}j} - \frac{k_{\mathrm{i}} k_j}{2m} \right], \tag{B.1}$$

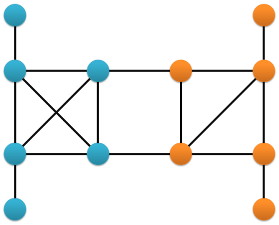
Under the traditional approach, we always have that if $\mathrm{i}, j \in C_k \implies A_{\mathrm{i}j} = 1$, but this is not always true under this new idea of community. Hence, for each isolated node within a community we will underestimate the modularity value because every time we consider this node in the first summation, at least once, we will add 0 instead 1 to that summation. Then for each isolated node $\mathrm{i} \in C_k$ we have (the worst case):

$$\Delta Q_{\mathrm{i}} = \frac{|C_k| - 1}{2m} \approx \frac{n-1}{2m} \approx 1/n \;\; for\; \mathrm{d}ense\; networks\; where\; m = O(n^2) \tag{B.2}$$

Where ΔQ_{i} is the modularity underestimation caused because the node i is an isolated node within a community

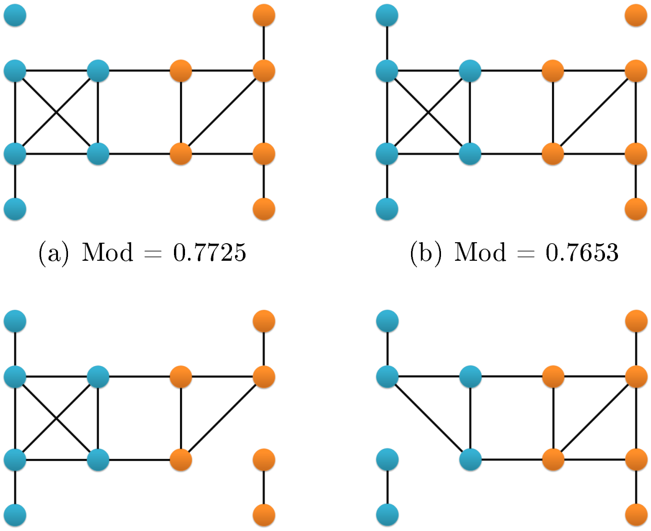
To exemplify this result, figure B.1 shows a synthetic community structure, each color means a different community. If we isolate different nodes we obtain the results shows in figure B.2. For all cases examined we obtained a modularity underestimation less than 3.5% which is according the theoretical result.

Figure B.1: Modularity Analysis



(a) Mod = 0.7787

Figure B.2: Modularity Analysis



(a) Mod = 0.7725

(b) Mod = 0.7653

(c) Mod = 0.7592

(d) Mod = 0.7517

Appendix C

Topics over Time

C.1 A Non-Markov Continuous-Time Model of Topical Trends

Topics over Time (TOT) [87] is an LDA-style method for topic modelling that explicitly models word co-occurrences jointly with time. In other words, TOT captures both the low-dimensional structure of data and how the structure changes over time. TOT parametrizes a continuous distribution over time associated with each topic, and topics are responsible for generating both observed timestamps as well as words.

When a strong word co-occurrence pattern appears for a brief moment in time then disappears, TOT will create a topic with a narrow time distribution. When a pattern of word co-occurrence remains consistent across a long time span, TOT will create a topic with a broad time distribution.

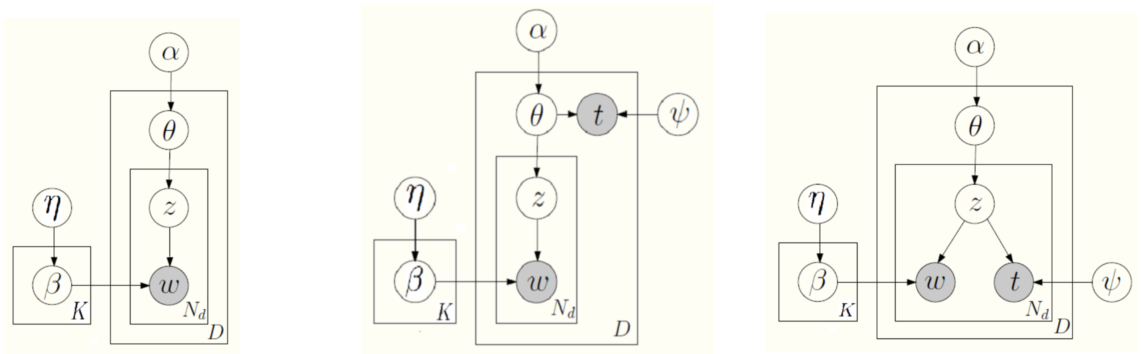
C.1.1 Procedure

Topics over Time is a generative model of time stamps and the words in the time stamped documents. There are two ways of describing its generative process. The first, which corresponds to the process used in Gibbs sampling for parameter estimation, is as follows:

1. For each topic $k \in [1, K]$,
   (a) Draw a distribution over words $\vec{\beta_k} \sim Dir_K(\eta)$
2. For each document $\mathrm{d} \in [1, D]$,
   (a) Draw a vector of topic proportions $\vec{\theta_{\mathrm{d}}} \sim Dir_V(\vec{\alpha})$
   (b) For each word $n \in [1, N_{\mathrm{d}}]$ in document d,
      i. Draw a topic assignment $Z_{\mathrm{d},n} \sim Mult(\vec{\theta_{\mathrm{d}}}), Z_{\mathrm{d},n} \in \{1, \ldots, K\}$
      ii. Draw a word $W_{\mathrm{d},n} \sim Mult(\vec{\beta_{Z_{\mathrm{d},n}}}), W_{\mathrm{d},n} \in \{1, \ldots, V\}$

iii. Draw a timestamp $t_{\mathrm{d},n} \sim Beta(\psi_{\vec{z}_{\mathrm{d},n}})$

Figure C.1: Three topic models: LDA and two perspectives on TOT



(a) LDA

(b) TOT model, alternate view

(c) TOT model, for Gibbs sampling

In this generative process, a time stamp is associated with each word, which is perfect when different parts of the document are discussing different time periods (See Figure C.1(c)). However, common documents typically have only one time stamp per document. Therefore, an alternative model to TOT is one where a single time stamp is associated with each document (See Figure C.1(b)).