المدرسة الوطنية للعلوم التطبيقية
ⵜⵉⵍⵎⵉ ⵜⴰⵏⴰⵎⵓⵔⵜ ⵏ ⵜⵎⴰⵙⵙⴰⵏⵉⵏ ⵜⵓⵙⵏⵉⵏ
Ecole Nationale des Sciences Appliquées de Fès

جامعة سيدي محمد بن عبد الله
ⵜⴰⵙⴷⴰⵡⵉⵜ ⵙⵉⴷⵉ ⵎⵓⵃⵎⵎⴰⴷ ⴱⵏ ⵄⴱⴷⵍⵍⴰⵀ
Université Sidi Mohamed Ben Abdellah

Master Architectures et Applications des Systèmes d'Information

Master 2ASI

# RAPPORT DE PROJET DE FIN D'ETUDES

**Intitulé:**

# ONLINE SOCIAL NETWORKS: MEASUREMENTS, ANALYSIS AND SOLUTIONS FOR MINING CHALLENGES.

**Réalisé par :**
El-Mehdi CHOUKI

**Encadré par :**
Youness IDRISSI KHAMLICHI

**Présenté le 28/06/2019 devant le jury composé de :**

- Youness IDRISSI KHAMLICHI (PH, ENSA, Fès).
- Adil KENZI (PH, ENSA, Fès).
- Adil JEGHAL (PH, FSDM, Fès).

**Année Universitaire 2018/2019**

# Dedication

**To my dear *parents***

**To my dear *brothers***

**To my dear *friends***

**To all my *teachers***

**...**

# Acknowledgements

The work in this thesis could not exist without the help and support of many people during Master Thesis research project. I am hugely indebted to friends, family and colleagues.

I would like to express my sincere gratitude to my supervisor Dr. Youness Idrissi Khamlichi for his always guidance to provide me with the necessary fundamental to fulfill my research project and for the many hours spent in discussing to point me to many interesting problems in order to pursue my research in the right direction. I am also grateful to everyone in the National Schools of Applied Sciences and all the people working there.

Also, I would like to thank all the researchers and scientists I have referred to in my thesis. I spent many hours reading and studying their resources. They provided me with rich theories and very useful insights about the field.

I would not have accomplished this road without the continuous support, affection and love showered by my parents Mohammed CHOUKI and Mina SAFSAFI. Their patience, understanding and continuous encouragement are greatly appreciated. Thanks my brother Mouhssine, to my sister Meryame for their selfless love and care which contributed a lot for the completion of my thesis. I consider myself a lucky person to have such a lovely and caring family, standing and supporting me unconditionally.

# Summary of Contributions

The aim of this dissertation is to highlight some challenges of analysing social networks and achieve some progress towards a better understanding of these problems to provide solutions to better deal with them. We tried to achieve this by tackling some of these challenges and by proposing solutions, studies and methodologies to how to deal and overcome the impact of these challenges during mining a network.

Through the thesis, we have addressed some of these challenges. In chapter 4, an approach using machine learning was proposed for predicting the performance or the execution time to analyse a social network. This approach presents a way to deal with evolving and scale-free networks problem. The proposed approach provides an easy way to predict the approximate time taken to analyse a new network (or graph) given the number of nodes and edges within the network. A simulation for the evolving graph is achieved by extracting subgraphs of increasing in size from one network. Then, an analysis is held on some popular graph measures using two different tools like SNAP and Gremlin. Finally, we utilized four different machine learning regression models: MARS, Boosting, M5 and Support Vector Machine. The models were trained and tested over 50 samples of graphs having different sizes in order to select the best model using RMSE an evaluation metric. Our results concluded that MARS outperformed the other three models suggesting that it might be the best suited for addressing this problem. We provided multiple computational models with their coefficients for each graph measure in terms of nodes and edges.

For sure, this work can not cover all topics and the challenges related to social networks analysis. But in the last years, this area has got a lot of research attention. In the following section, we suggest some open problems and questions to be explored for further research.

# Résumé des contributions

L'objectif de cette thèse est de mettre en évidence certains défis de l'analyse des réseaux sociaux et de progresser quelque peu vers une meilleure compréhension de ces problèmes afin de proposer des solutions pour mieux les gérer. Nous avons essayé d'y parvenir en abordant certains de ces défis et en proposant des solutions, des études et des méthodologies permettant de gérer et de surmonter l'impact de ces défis lors de l'exploitation d'un réseau.

Au cours de la thèse, nous avons abordé certains de ces défis. Au chapitre 4, une approche utilisant l'apprentissage automatique a été proposée pour prédire la performance ou le temps d'exécution pour analyser un réseau social. Cette approche offre un moyen de traiter le problème des réseaux en évolution et sans échelle. L'approche proposée offre un moyen simple de prédire le temps approximatif nécessaire pour analyser un nouveau réseau (ou graphique) en fonction du nombre de nœuds et d'arêtes dans le réseau. Une simulation du graphe en évolution est réalisée en extrayant des sous-graphe de plus en plus grands à partir d'un réseau. Ensuite, une analyse est effectuée sur certaines mesures graphiques populaires en utilisant deux outils différents, tels que SNAP et Gremlin. Enfin, nous avons utilisé quatre modèles de régression d'apprentissage automatique : MARS, Boosting, M5 et Support Vector Machine. Les modèles ont été formés et testés sur 50 échantillons de graphiques de tailles différentes afin de sélectionner le meilleur modèle en utilisant RMSE, une métrique d'évaluation. Nos résultats ont conclu que MARS surperformait les trois autres modèles, suggérant qu'il pourrait être le mieux adapté pour résoudre ce problème. Nous avons fourni plusieurs modèles de calcul avec leurs coefficients pour chaque mesure de graphique en termes de nœuds et d'arêtes.

Bien sûr, ce travail ne peut pas couvrir tous les sujets et les défis liés à l'analyse des réseaux sociaux. Mais ces dernières années, ce domaine a fait l'objet de nombreuses recherches. Dans la section suivante, nous suggérons quelques problèmes en suspens et des questions à explorer pour des recherches ultérieures.

# ملخص المساهمات

الهدف من هذه الرسالة هو تسليط الضوء على بعض تحديات تحليل الشبكات الاجتماعية وتحقيق بعض التقدم نحو فهم أفضل لهذه المشكلات لتوفير حلول للتعامل معها بشكل أفضل. لقد حاولنا تحقيق ذلك من خلال معالجة بعض هذه التحديات واقتراح حلول ودراسات ومنهجيات حول كيفية التعامل والتغلب على تأثير هذه التحديات أثناء استخراج شبكة.

من خلال الأطروحة ، تعاملنا مع بعض هذه التحديات. في الفصل ٤ ، تم اقتراح طريقة تستخدم التعلم الآلي للتنبؤ بالأداء أو وقت التنفيذ لتحليل شبكة اجتماعية. يقدم هذا النهج طريقة للتعامل مع مشكلة الشبكات المتطورة وخالية من النطاق. يوفر النهج المقترح طريقة سهلة للتنبؤ بالوقت التقريبي الذي يستغرقه تحليل شبكة جديدة (أو رسم بياني) بالنظر إلى عدد العقد والحواف داخل الشبكة. يتم إجراء محاكاة للرسم البياني المتطور من خلال استخراج الرسومات الفرعية ذات الحجم المتزايد من شبكة واحدة. بعد ذلك ، يتم إجراء تحليل لبعض مقاييس الرسم البياني الشائعة باستخدام أداتين مختلفتين مثل SNAP و Gremlin. أخيراً ، استخدمنا أربعة نماذج مختلفة من تعلم الانحدار الآلي: MARS , Boosting , M5 و Support Vector Machine. تم تدريب النماذج واختبارها على أكثر من ٥٠ عينة من الرسوم البيانية ذات أحجام مختلفة من أجل اختيار أفضل نموذج باستخدام مقياس تقييم RMSE.

خلصت نتائجنا إلى أن MARS تفوقت على النماذج الثلاثة الأخرى مما يشير إلى أنها قد تكون الأنسب لمعالجة هذه المشكلة.

بالتأكيد ، لا يمكن لهذا العمل تغطية جميع المواضيع والتحديات المتعلقة بتحليل الشبكات الاجتماعية. ولكن في السنوات الأخيرة ، حظي هذا المجال بالكثير من الاهتمام بالبحث.

# Abstract

In the last decade, online social networks showed enormous growth. With the rise of these networks and the consequent availability of wealth social network data, Social Network Analysis (SNA) led researchers to get the opportunity to access, analyse and mine the social behaviour of millions of people, explore the way they communicate and exchange information.

Despite the growing interest in analysing social networks, there are some challenges and implications accompanying the analysis and mining of these networks. For example, dealing with large-scale and evolving networks is not yet an easy task and still requires a new mining solution. In addition, finding communities within these networks is a challenging task and could open opportunities to see how people behave in groups on a large scale. Also, the challenge of validating and optimizing communities without knowing in advance the structure of the network due to the lack of ground truth is yet another challenging barrier for validating the meaningfulness of the resulting communities.

In this thesis, we started by providing an overview of the necessary background and key concepts required in the area of social networks analysis. Our main focus is to provide solutions to tackle the key challenges in this area. For doing so, we introduce a predictive technique to help in the prediction of the execution time of the analysis tasks for evolving networks through employing predictive modeling techniques to the problem of evolving and large-scale networks.

**Keywords:** Online Social Networks, Social Networks Analysis, Predictive modeling, Data mining, Supervised machine learning, Visualization

# Résumé

Au cours de la dernière décennie, les réseaux sociaux en ligne ont enregistré une croissance énorme. Avec la montée en puissance de ces réseaux et la disponibilité conséquente de données de réseaux sociaux de richesse, l'analyse de réseau social (ARS) a conduit les chercheurs à avoir la possibilité d'accéder, d'analyser et d'explorer le comportement social de millions de personnes, d'explorer leur manière de communiquer et d'échanger des informations .

Malgré l'intérêt croissant porté à l'analyse des réseaux sociaux, l'analyse et l'exploitation de ces réseaux soulèvent des difficultés et ont des implications. Par exemple, gérer des réseaux à grande échelle et en évolution n'est pas encore une tâche facile et nécessite toujours une nouvelle solution d'exploitation minière. De plus, trouver des communautés au sein de ces réseaux est une tâche difficile et pourrait ouvrir la voie à la manière dont les gens se comportent en groupes à grande échelle. En outre, le défi de valider et d'optimiser les communautés sans connaître à l'avance la structure du réseau en raison de l'absence de vérité sur le terrain constitue un obstacle supplémentaire à la validation du sens des communautés résultantes.

Dans cette thèse, nous avons commencé par donner un aperçu du contexte et des concepts clés nécessaires dans le domaine de l'analyse des réseaux sociaux. Notre objectif principal est de fournir des solutions pour relever les principaux défis dans ce domaine. Pour ce faire, nous introduisons une technique prédictive pour aider à prévoir le temps d'exécution des tâches d'analyse pour les réseaux en évolution en utilisant des techniques de modélisation prédictive pour résoudre le problème des réseaux en évolution et à grande échelle.

**Mots clés :** Réseaux sociaux en ligne, Analyse des réseaux sociaux, Modélisation prédictive, Exploration de données, Apprentissage supervisé, Visualisation

# ملخص

في العقد الماضي ، أظهرت الشبكات الاجتماعية على الإنترنت نمواً هائلاً. مع ظهور هذه الشبكات وما يترتب على ذلك من توفر بيانات الشبكة الاجتماعية للثروة ، قاد تحليل الشبكة الاجتماعية الباحثين إلى إتاحة الفرصة للوصول إلى السلوك الاجتماعي لملايين الناس وتحليلهم واستكشافهم ، واستكشاف الطريقة التي يتواصلون بها وتبادل المعلومات .

على الرغم من الاهتمام المتزايد بتحليل الشبكات الاجتماعية ، هناك بعض التحديات والآثار المصاحبة لتحليل هذه الشبكات واستخراجها. على سبيل المثال ، لم يعد التعامل مع الشبكات واسعة النطاق والمتطورة مهمة سهلة ولا يزال يتطلب حلاً جديداً للتعدين. بالإضافة إلى ذلك ، فإن العثور على مجتمعات داخل هذه الشبكات يمثل مهمة صعبة ويمكن أن يفتح الفرص لرؤية كيف يتصرف الناس في مجموعات على نطاق واسع. كذلك ، فإن التحدي المتمثل في التحقق من صحة المجتمعات وتحسينها دون معرفة بنية الشبكة مسبقاًا نظراً لعدم وجود الحقيقة يمثل عائقاًا آخر صعباًا أمام التحقق من صحة معنى المجتمعات الناتجة.

في هذه الأطروحة ، بدأنا بتقديم نظرة عامة على الخلفية الضرورية والمفاهيم الأساسية المطلوبة في مجال تحليل الشبكات الاجتماعية. ينصب تركيزنا الرئيسي على توفير حلول لمواجهة التحديات الرئيسية في هذا المجال. للقيام بذلك ، نقدم تقنية تنبؤية للمساعدة في التنبؤ بوقت تنفيذ مهام التحليل للشبكات المتطورة من خلال استخدام تقنيات النمذجة التنبؤية لمشكلة الشبكات المتطورة والشبكات واسعة النطاق.

**الكلمات الدالة**: الشبكات الاجتماعية عبر الإنترنت ، تحليل الشبكات الاجتماعية ، النمذجة التنبؤية ، استخراج البيانات ، تعلم الآلة الخاضعة للإشراف ، التصور

# Contents

# List of Figures

# List of Tables

Chapter 1

# Introduction

## 1.1 Motivation of the thesis

Online social networks have shown a huge growth in the last decade. With this enormous rise of the online social networks like Facebook[1], Twitter[2] and LinkedIn[3] , researchers have gained the opportunity to have access to online social networks data and to track the behavior of people and their interactions. These online networks started with only hundreds of people but with the drastic growth of such networks nowadays, this sheds the light for researchers to gather deeper insights about people and see how they behave in the network. They also provide information about communities, roles and user actions. Social graphs are popular structures for modeling relationships, interactions and communication between users, organizations or even groups. They are the commonly used way of modeling the shared information and ideas over the Internet [16]. Network analysis is a collection of techniques to calculate various metrics for a social graph.

There is a positive look on the field of mining, analyzing social networks and their vast applications in terms of research and impact on business. But still the area is not that mature and there are a lot of challenges which require novel solutions or techniques from different disciplines. Some of these challenges are technical challenges and others are social and human challenges.

An example of technical challenges is the networks dynamics, as most of the networks turned from a static to dynamic due to the evolving nature of these networks [59]. Being able to deal with this evolution is becoming crucial and a lot of research need to be done to create models to deal and understand this growing of time-stamped networks either for research or business goals. Also, data preparation can create technical challenges; it requires developing techniques to facilitate managing, cleaning, documenting and anonymizing data. Other challenges include evaluation and the difficulty of having a reference metric either due to the difficulty of collecting and sharing data or due to the difficulty of formulating a ground truth even if the data is available.

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

[2] https://twitter.com

[3] https://www.linkedin.com

Hence, the generalization of results of one performed experiment became a challenging task.

As for the social and human challenges, these include privacy problems, which requires defining the right balance whether if the information is related to the situation, individual or organization. Other ethical and legal issues place limitations on making use of the data even in research purposes. In addition to the aforementioned challenges, defining the community structure involves defining the appropriate level of communities and subcommunities that need to be targeted and how the existing links can be interpreted.

In this thesis, we consider some of the challenges in social network analysis like networks dynamics, community structure, evaluation, etc. We contribute and provide some techniques, methods and frameworks to deal with these challenges, this will be discussed in detail in the following sections.

## 1.2 Contribution of the dissertation

The main contribution of the thesis is to provide and introduce some solutions for the analysis and mining challenges in social networks which may remove some of the obstacles that obstructs the analysis process. We have provided a prediction approach that can be used to predict the execution time and the performance of the analysis specifically to deal with evolution of networks phenomena and solve the challenge of dealing with huge networks. This also helps in defining the required hardware and reflects an estimate about the expected performance before performing the analysis. It is argued that this approach could be used further across any network size or any type of hardware.

## 1.3 Structure of the thesis

The thesis is organized as follows: Chapter 2 gives an overview of graph theory and the mathematical background that is required for core contribution chapters. Chapter 3 provides an overview of the concepts and the techniques for analysing graphs. Also, it presents an overview on the state-of-the-art work and highlights the challenges that can obstruct the network analysis process. Chapter 4 presents our contribution regarding dealing with the evolution of networks and predicting the performance in advance.

**Chapter 2** introduces an overview of the mathematical concepts of graph theory which provide the basic concepts and applications used for network analytics. Several definitions and theorems are reviewed to give a theoretical overview of the hidden mathematics that provides the way to analyse a social network using social graphs.

**Chapter 3** aims to present background of the social network analysis area and its fundamentals. Different analysis techniques and methods are proposed. A literature review provides the commonly used applications and the work done in this area. Finally, a categorization for the main challenges that have gained the attention of many researchers in this field is presented.

**Chapter 4** introduces an approach based on predictive modeling that can tackle one of the most popular challenges in social networks field, which is the dynamic nature of the real-world social networks that dramatically change over time. Incorporating features of dynamic or evolving networks requires repetitive computations and so performance of the calculations is essential. Thus, we introduce a performance prediction approach that can be used as a tool to provide an approximate execution time needed to analyse a given network. The approach is based on applying supervised machine learning models by training the performance of calculating some popular network measures with different mining tools. Our approach aims to predict the execution time for a given network before even starting the analysis process. Predicting the performance can help in choosing an appropriate mining tool that can suit the size of the given network and can assist in deciding if there is a need for additional hardware (e.g. High performance computing clusters). The work in this chapter addresses the fertile area of how to deal efficiently with the evolving networks.

**Chapter 5** highlights the conclusions of the achievements and the contribution of the thesis. It also suggests the possible future work that can be further researched in the area of social network analysis

# Part 1

# THE STATE OF THE ART

Chapter 2

# The Power of Graph Theory

## Contents

The purpose of this chapter to touch lightly on the topic of graph theory and familiarize the reader with the basic concepts, and, consequently, a list of useful definitions and notations. All of these definitions constitute the basic structure of social networks analysis. At first we will provide an overview of the history of graph theory and how the concept of modern graph theory arose. Then we will try to summarize the most relevant graph theoretical concepts, definitions and formulas that are considered the mathematical background for network analytics.

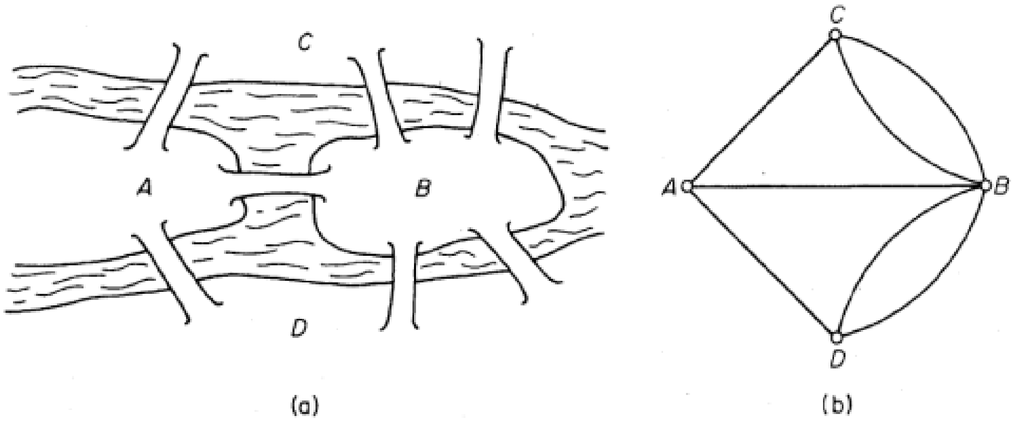


Figure 2.1: The seven bridges of Köningsberg and their graph [10].

## 2.1 Graph History

The origin of the theory of graphs started with the problem of the seven Köningsberg bridges as shown in Figure 2.1, when Euler was asked to find a nice path across the seven Köningsberg bridges and this path should cross over each of the seven bridges exactly once [30]. Köningsberg is situated at the sides of the river Pegel which comprises two big islands. Euler thought that the only clue to solve this problem is to derive a sequence for the bridges to cross. Euler mapped the problem into mathematical graph structure. He mapped land masses to vertices and the bridges to edges. He pointed out that during a walk: If the walk starts from a certain bridge, it should then leave from another one, given the fact that every bridge should be crossed only once. This should mean that the number of bridges that touch the land mass should be even. But, there is a contradiction here as all the land masses were touched by an odd number of edges. Therefore, Euler concluded that this walk exist iff the graph is connected with exactly 0 or 2 vertices having an odd degree. By proving this, Euler put the essential base of modern graph theory.

## 2.2 Preliminaries

Graph theory is the main study for graphs where simple data structure are mapped into set of vertices and links between these vertices. Sometimes a vertex is called a node and an edge might be called a link. For an overview on the general theories of graphs, the reader can refer to [24] and [10]. As mentioned before, some fundamental notation and definitions used throughout this thesis are presented in this chapter. Others will be given later in the next chapter. We begin by defining a **simple graph** or **graph** G is a pair of sets (V ,E) where E is a set of two sets of V . Graphs can be represented by diagrams, however these diagrams are not graphs. Graphs can be weighted, labeled and colored which we will describe later.

While a **subgraph** H=(T, R) of a graph G =(V, E) is a graph where $T \leqslant V, R \leqslant E$ .

The subgraph is induced by a set of vertices T is the graph obtained by including the edges between the vertices of T.

## 2.3 Paths, Cycles and Walks

Finding the optimal solution to measure the shortest path within a graph is indeed not a trivial task due to the difficulty of choosing which vertex should be chosen next at each point of the path. Some of the common measures used to tackle this point include: a **path** which represents a sequence of distinct vertices involved while following a sequence of edges through out the network. A path is named as a **geodesic path** when it represents the shortest path between two vertices having the minimum number of edges. A **walk** is an another measure which represents an ordered sequence of vertices including the repeated ones. On the other hand, a connected sequence of distinct edges in the graph with repeated vertices but no repeated edges is named as a **trail**. However, when a trail starts and ends at the same vertex with no vertex repeated, it forms a **cycle**.

## 2.4 Graph Properties

The following are some commonly graph properties and characteristics that deal with the number of vertices and the number of edges for any graph G. These properties can give a high level overview about the structure of any graph.

- The **size** is the number of edges E in the graph while the order is the number of vertices V in the graph.
- The **path length** L is the distance d(i, j) between any vertex of the graph and every other vertex. This distance corresponds to the number of edges that should be crossed to move from vertex i to vertex j, this is what is called by the **shortest path** that is defined in [103] as: a path length L of a graph is the median of the means of the shortest path lengths connecting each vertex $v \in V(G)$ to all other vertices.
- The **diameter** of the graph G is the maximal distance between any two vertices in the graph. It is also known by the maximum *eccentricity* across all vertices such that the eccentricity of a node i is the maximal shortest path distance between i and any other node. While the *radius* is the minimum eccentricity across all vertices in the graph. According to the computational complexity of measuring the radius and the diameter they are always calculated through measuring the eccentricity of a random of vertices in the network.

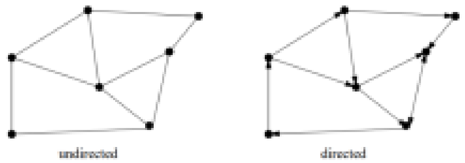


Figure 2.2: Directed graph versus Undirected graph.

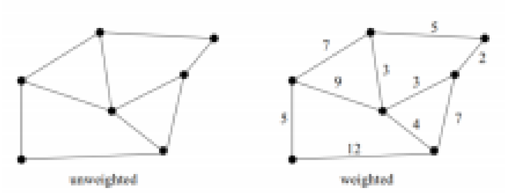


Figure 2.3: Weighted graph versus Unweighted graph.

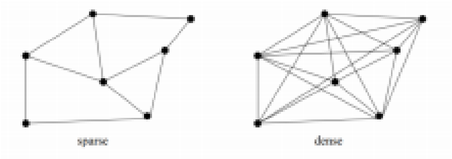


Figure 2.4: Sparse graph versus Dense graph.

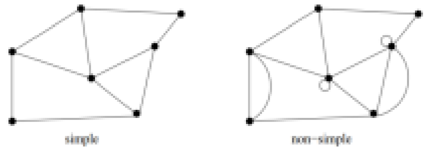


Figure 2.5: Simple graph versus Multigraph (non-simple).

- The **degree** of vertex i is the number of incident edges to this vertex. An isolated vertex has degree 0 while the degree of a self loop counts 2.
- The **neighborhood** of a vertex i is the set of all the vertices that are connected to vertex i. The formal definition of neighborhood is: The neighborhood $\Gamma(i)$ is the subgraph consisting of all vertices adjacent to i (not including i itself).
- The **clustering coefficient** of the vertex is measuring how the neighborhood subgraph of a vertex i is well connected. It is the fraction of actual edges and possible (expected) edges between the neighborhood of vertex i. While the clustering coefficient of the graph G is the average clustering coefficient across all vertices.

## 2.5 Graph Characterization

Graphs differ in several ways based on the types of edges that connect the vertices to each other. The focus in this section will be on the characterizations of different types of graphs as shown in Figures 2.2, 2.3, 2.4 and 2.5.

### 2.5.1 Directed and Undirected

Directed graph or digraph is a graph where each edge has a direction. For $\mathrm{E} = (\mathrm{V_s}, \mathrm{V_t})$ an edge between source vertex $\mathrm{V_s}$ and terminal vertex $\mathrm{V_t}$ . In directed graphs, edges are represented by arrows. Each node here has in-degree $\mathrm{d_{in}(V\ )}$ and out-degree $\mathrm{d_{out}(V\ )}$ where the graph is called balanced graph when $\mathrm{d_{in}(V\ ) = d_{out}(V\ )}$ for all the nodes in the graph. While the Undirected graph is the graph that represents the relationship in a symmetric form, this imply that bond can exist in either (or both) directions .

### 2.5.2 Weighted and Unweighted

A graph in which edges are assigned with weights (i.e. real numbers) to indicate their strength. The importance of an edge for unweighted graphs are merely defined from their relationship with other edges.

### 2.5.3 Simple and Multigraph

A simple graph is a graph that has no self loops from the vertex to itself or multiple edges between the same vertices. While the non-simple graph (multigraph) allows self loops and multiple edges between the same vertices. Thus, the degree of a vertex in simple graph is the number of adjacent vertices to it while in multigraph the degree is the number of times it appears in the edge set.

### 2.5.4 Sparse and Dense

A graph is sparse when a small number of vertices have edges actually defined between them. The sparsity of the graph always depends on its application, for example, road networks have to be sparse due to the constraints by road junctions. Sparse and dense are sometimes used informally, but it can be made formal. For example, sparse graphs are linear in their size, however, dense graphs have a quadratic number of edges.

### 2.5.5 Connected

A connected graph is a graph where every vertex can be reached from any other vertex by traversing an infinite number of edges.

## 2.6 Graph Representation

If we are going to deal with the graph as a data structure, then, we need to define a way to represent it in memory. There are a couple of different methods that we can use, but we must keep in consideration that whatever the method we want to employ, our ultimate goal for all the methods is check the existence of an edge between two vertices. A common task in a graph is iterating over vertices adjacent to each other.

### 2.6.1 Adjacency Matrix

The adjacency matrix is a way to represent the edges of a graph. A graph can be completely determined by its adjacency matrix where the properties of this matrix are closely correspond to the properties of the graph. An adjacency matrix A(G) is a matrix of size $n \times n$ where $A_{i,j}$ = 1 iff vertex i is connected to vertex j and $A_{i,j}$ = 0 if otherwise. This representation could be normally extended to represent digraphs, where $A_{i,j}$ = 1 iff i and j were connected by an edge directed from i to j. The adjacency matrix for an undirected graph is always symmetric across its diagonal. It is usually preferred when the graph is dense and can represents either directed or undirected graphs. Many types of information about a graph can be easily derived with the help of adjacency matrix.

### 2.6.2 Edge List

Another common way to represent graphs is through a list of edges. Each edge consists of two vertices represented by just having one array of two vertices or linked list that store pairs of vertices. This can be useful if the edges are sparse in the graph.

### 2.6.3 Adjacency List

A graph represented with an adjacency list combines both edge list and adjaceny matrix. For each vertex in the graph there is an array of the other vertices adjacent to it, each of which contains a list of all adjacent vertices in an arbitrary order. The way we look for an edge (i,j) within the graph is that we check the $i^{th}$ adjacency list when we go for the $j^{th}$ in the $i^{th}$ list. It is usually used to represent sparse graphs.

## 2.7 Special Graphs

- A **Complete graph** is a simple graph K $_n$ having all possible edges between n vertices (every vertex is adjacent to every other vertex) where n = 2,3,4 ...... and $E = n(n-1) \div 2$.
- A **Bipartite graph** is a graph composed of two disjoint graph vertices sets such that no two vertices in the same set are adjacent. It is a special case of K-bipartite when K=2. A graph G is called bipartite, if $V_G$ has a partition to two subsets X and Y such that each edge $ij \in G$ connects a vertex of X and a vertex of Y . In this case, (X, Y ) is a bipartition of G, and G is (X, Y )-bipartite. There are variety of useful theorems about Bipartite graphs, for example:

  **Theorem 2.7.1.** *A graph G is bipartite iff V (G) has a partition to two stable sets.*

  **Theorem 2.7.2.** *A graph G is bipartite iff every cycle is of even length.*

  **Theorem 2.7.3.** *A graph is bipartite iff it contains no odd cyles.*

- A **complete bipartite graph** $K_{m,n}$ is the graph that has its vertex set partitioned into two sets: X and Y of m, n vertices respectively. There exists an edge between two vertices iff one vertex from set X and the other vertex from set Y

- An **Acyclic graph** is a graph with no cycles. This type of graph contain at minimum one node with zero in-degree.
- A **Tree** is a connected undirected graph having no cycles. There are many results regarding trees. For example, a spanning tree is a subgraph that is a tree which includes all of the vertices of the graph G with the minimum possible number of edges.

  **Theorem 2.7.4.** *Each connected graph has a spanning tree that is a spanning graph that is a tree.*
- Other existing types like **star** and **ring** graphs [74].

## 2.8 Hamiltonian Cycles and Euler Circuits

In this section, we will present some definitions that Euler has used to show his theorem (discussed earlier in this chapter) to solve the problem of how to walk through the town and traverse all the bridges only once.

- **Euler cycle** is a cycle containing every edge in the graph precisely once. A graph has an Euler cycle iff every vertex is of even degree. A graph G is said is Eulerian when it has Euler cycle.
- **Euler trail** is a path through a graph containing each edge precisely once, starting and finishing at different points. A graph has an Euler trail iff it has precisely two vertices of odd degree. A graph of this kind is called traversable graph.

  **Theorem 2.8.1.** *An Eulerian trail exists in a connected graph iff there are exactly two odd vertices.*
- The **Hamiltonian cycle** is a cycle containing every vertex of the graph precisely once.

  Generally as shown in Figure 2.6, the core difference between an Eulerian cycle and Hamiltonian cycle is that, the first traverses every edge in a graph exactly once, but may repeat vertices, while the second visits each vertex in a graph exactly once but may doesn't traverses some edges.

## 2.9 Graph Topological Cases

### 2.9.1 Isomorphism and Homeomorphism

1. Isomorphism

   An isomorphism exists between two graphs G and H if G and H clearly have the same structure and only differ in the names of vertices and edges.

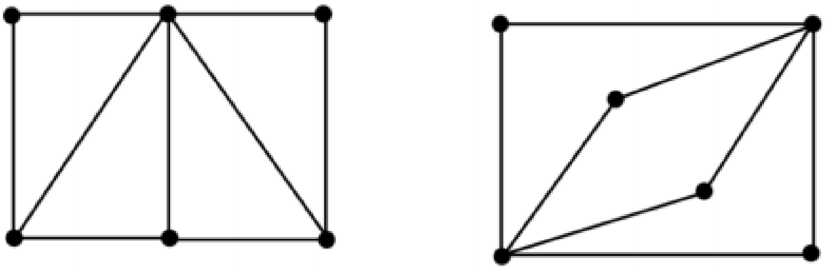


Figure 2.6: The left graph is Hamiltonian but non-Eulerian and the right graph is Eulerian but non-Hamiltonian.

**Theorem 2.9.1.** *Two graphs G and H are isomorphic iff they have a common adjacency matrix and the two isomorphic graphs have exactly the same set of adjacency matrices.*

If two graphs G and H are identical then they can have same diagrams. But, if G and H are non-identical graphs, they can also have identical diagrams. These graphs are not identical one but they are isomorphic graphs. They look exactly the same but their vertices and edges differ in their labels. This means that G and H are acquiring the same structure and the only difference in their names. The following definitions on isomorphism in graphs were presented in [7].

- Two graphs G=(V, E), H=($V'$ , $E'$) are said to be isomorphic if there is a bijection, such that this mapping f: V (G) -> V (H), E(G) -> E (H) is an isomorphism, i.e., iff i , j are adjacent in G the f(i), f(j) in H are also adjacent.
- If two graphs are isomorphic then the subgraphs induced by the sets of vertices of a given degree are also isomorphic.
- If two graphs are isomorphic then they have the same number of cycles of a given length

2. Homeomorphism
A graph F is homeomorphic from a graph G if it can be obtained by subdividing edges of G or if they are isomorphic or they are both homeomorphic from a third graph H.

### 2.9.2 Planarity

The graph that can be drawn in the plane with no edge crossing so that its edges intersect only at their ends. Any planar graph can be colored with four colors. Long time ago a formula 2.1 discovered by Euler in 1736 to test whether a graph is planar or not. This formula plays an observable role in the study of planar graph. After that, Kuratowski discovered a criterion for a graph to be planar 2.9.2 Then, Whitney developed some properties of how to embed the graph into the plane.

- *Euler's Theorem*
  Let G be simple planar graph with V vertices, E edges and R regions then

$$V - E + R = 2 \tag{2.1}$$

- *Kuratowski's Theorem*

  **Theorem 2.9.2.** *The graph is planar iff it contains no subgraph homeomorphic from $K_5$ nor $K_{3,3}$.*

  The $K_5$ is the complete graph of order 5 while the $K_{3,3}$ is the complete bipartite 3 by 3 which is also named T homsen graph. $K_5$ and $K_{3,3}$ are the fundamental blocks in a non planar graph.

  where V is the number of vertices , E is the number of edges and R is the number of regions. Regions should be drawn in a planar form. If G is a simple planar graph then the degree of each region is at least 3. It should be also clear that a graph is planar iff each of its block is planar.

- *Maximal planar*
  The graph in which no more edges can be added without losing neither the simplicity nor the planarity. These graphs are some times called triangulated (every region is of degree 3).

  **Theorem 2.9.3.** *If G is a maximal planar graph then:*

$$E = 3V - 6 \tag{2.2}$$

  **Corollary 2.9.1.** *If G is planar graph then:*

$$E \leq 3V - 6 \tag{2.3}$$

## 2.10 Connectivity and Components

A **strongly connected graph** is a graph whose each node is connected to every other node through a direct path. The **components** of a graph G are the maximal connected subgraphs. A maximal connected subgraph is achieved when no any vertex or edge can be added and have a connected piece.

Sometimes the removal of a vertex or an edge can affect the number of components this is called a *cut-vertex* or *cut-edge* respectively. A cut-vertex is a single vertex whose removal disconnects the graph and the cut-edge is a single edge whose removal disconnect the graph. Sometimes the cut-edge is called a bridge. Anytime a graph contain a cut-edge then there is a cut-vertex but not vice verse.

A **block** is a maximal 2-connected subgraph. A block is 2-connected when it possess no cut-vertex or cut-edge. A **minimal block** is a block such that the removal of any edge e results in a subgraph $G - e$ such that $G - e$ is not a block. Every minimal block contain at least one vertex of degree 2. Hence G is not a minimal block.

## 2.11 Graph Coloring

In this section we define some of the fundamental notions for graph coloring. A coloring of a graph is one of the most essential concepts within graph theory. A proper graph coloring is the assignment of the minimum number of colors to the vertices or the edges of the graph such that no two adjacent vertices or two edges meet at the same vertex have the same color. We denote to this as vertex-coloring and edge-coloring respectively. This minimum color is known as **chromatic number**.

### 2.11.1 Four Color Problem

Around 1850, the four color problem was conjectured, where a country map should be colored given four colors only. This problem involves the association of colors such that any two regions with common side should be given a different color. It has been known that four colors can color some maps and five colors can be sufficient for all maps [34].

### 2.11.2 Vertex Coloring

A K-coloring or k-vertex coloring of a graph G = (V ,E) is mapping f : $\mathrm{V_G}$ to [1,k]. A graph G is K-colorable given that there exists a proper K-coloring.

A **vertex chromatic number** $\chi(G)$ of G is the minimum number K for a graph for which there exists a K coloring. It is defined as:

$$\chi(G) = min(k|exists\ proper\ k\ coloring) \tag{2.4}$$

therefore, if $\chi(G) = k$ then G is k-chromatic.

**Theorem 2.11.1.** *Let G be a graph, then $\chi(G) \leq \Delta(G) + 1$.*
*where $\Delta(G)$ is the maximum degree of the vertices for a graph G.*

### 2.11.3 Edge Coloring

An **edge coloring** of a graph G is the assignment of K colors to edges of G such that no two adjacent edges have the same color. An **edge chromatic number** $\chi'$ (G) of G is the minimum number K for a graph for which there exists a K edge colorable. There are some special cases for edge coloring according to the type of the graph.

## 2.12 Summary

In this chapter we introduce some graph theory concepts which we make use of some of them in the rest of the thesis and provide an overview of graph theory as presented in standard textbooks. The overview in this chapter is the mathematical introduction for the next chapter in which we will discuss in more detail the way of dealing with graphs, graph analytics and their application in social networks.

Chapter 3

# Overview on Social Network Analytics

## Contents

In this chapter, we introduce the foundational aspects of Social Networks Analysis (SNA), provide technical research work, applications in this area and discuss a set of research challenges related to mining data in social networks. The area of social network analysis is evolving fast in different directions so we will try, in this chapter, to cover the main aspects related to areas such as definition, approaches, mining and challenges. We will focus on analysis measures for individuals or the whole network in Section 3.5, as that will be needed for the rest of the thesis. In addition to giving an overview of popular mining areas in Section 3.6. Finally, we will end up highlighting different key problems and challenges of social network analysis which will be considered in the next chapters.

## 3.1 Social and Non-Social networks

There is a key difference between a social network and non-social network [96]. A social network focuses on the structure of relationships between social entities (Individuals) while a non-social network can represents countries, machines, products or any non-social entities. Analyzing a social network is based on an encapsulated concept which not only consider the individuals but also include the relational and actionable links between them. Wasserman [25] highlighted four main features of social network analysis as follows:

1. *Actors and their actions are viewed as interdependent rather than independent, autonomous units.*

2. *Relational ties (linkages) between actors are channels for flow of resources (either material or non-material).*

3. *Network models focusing on individuals view the network structural environment as providing opportunities for or constraints on individual action.*

4. *Network models conceptualize structure (social, economic, political, and so forth) as lasting patterns of relations among actors*

Popular social platforms like Facebook, LinkedIn, etc. contain huge amount of data, information and lots of interactions for people within these networks and this has gained the attention of many researchers. Collecting such a huge amount of data, implied the shift to “computational social science” [55] with the availability of technological techniques that contribute significant advances in the research of social network.

## 3.2 Social Network Analysis

Early social network analysis was mainly based on graph theory, which has been proved to be very efficient on large networks. This basis of graph theory has proven to be a very powerful methodology to study and analyse physical or social networks. The roots of social network theory begin from social science, in addition to statistics and mathematics. Analyzing social network data is used in many fields to discover the social nature and behaviors of people in the network. In the recent years, social networks analysis has acquired huge attention due to the significant increase of social networks, accompanied by the availability of rich sources of social network data. This dramatic growth in social networks has encouraged massive number of users to share information, communicate, create new content and, based on their interactions, they can get useful recommendations. This rapid increase opened new challenges and unlocked the boundaries of studies and research for new analytics solutions.

### 3.2.1 Types of Social Network Analysis

A social network has a set of relations of ties, which can be viewed in two different ways. One approach focus on an individual, called *ego-centered* network, and put it at the centers of the network. Members of the network are defined by the relations with the ego. Ego-centered network analysis can show the range and breadth of connectivity for individuals and identify those who have access to diverse pools of information and resources. The ego-centered approach is useful when the population is large, or the boundaries of the population are hard to define [54] [92]. The second approach , called *socio-centered* network considers the whole network based on some specific criterion of population boundaries such as a formal organization, department, club or kinship group. Whole network analysis can identify those members of the network who emerge as central figures or who act as bridges between different groups. This approach requires responses from all members on their relations with all others in the same environment, such as the extent of email and video communication in a workgroup [43].

### 3.2.2 Social Network Analysis Techniques

The focus on actors, relations and patterns of existing relations requires a set of tools, methods and analytical concepts that are distinct from the methods of traditional statistics and data analysis. Thus, social network analysis is defined as the process of analysing social network and defining key actors, groups and relationships as well as changes in these variables [69]. While social Network analysis tools are a set of techniques, tools, methods and visualization techniques used in social network analysis which include graph theory concepts and statistical techniques [69].

Visualization is also a hot topic of social network analysis, and it is a suitable technique in this area. Through the visualization of social networks, the characters of social networks can be understood easily, such as the structure of networks, the distribution of nodes, the links (relationships) between nodes and the clusters and groups in the social networks [44] [102].

In additional to social network extraction and visualization, there are other measurements that can be used for social network analysis as well [102]. For example, centrality degree of a social network is a measurement that is used to measure the betweenness and closeness of the social network [101]. Betweenness centrality indicates the extent to which a node lies on the shortest path between every other pair of nodes. Closeness centrality analyzes centrality structure of a network based on geodesic distances among nodes in a social network [21]. Cluster coefficient is a measurement to discover the clusters in a social network and to measure the coefficient of the clusters. The density measurement can be used to analyze the connectivity and the degree of nodes and links in a social network [57].

The measurements path length and reachability can be used to analyze how to reach a node from another node in the social networks. Structural hole is also a measurement of social network analysis, which can be used to discover the holes in a social network and by this to fill

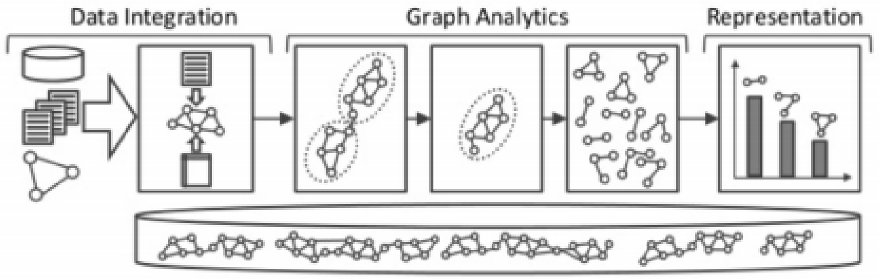


Figure 3.1: A full cycle of Social Network Analytics (SNA) process from importing data to results visualization.

the hole and expand the social network [35]. These sparse regions are structural holes that prove opportunities for brokering information flows among actors. Thus, maximizing the structural holes spanned or minimizing redundancy between actors is an important aspect of constructing an efficient, information-rich network [13].

A full network analytics cycle, from begin to end, can be summarized in three stages that show the sequence of steps required for the analysis process (Figure. 3.1):

1. **Data Integration:** is the process of collection and integration of data from different data sources (e.g. graph databases, files, etc.) into a graph structure.

2. **Graph Analytics:** is the process of analytics which includes path analysis, connectivity analysis, community analysis and centrality analysis to discover the structure of the network (graph) or behavior of people within the network.

3. **Representation:** is the process visualizing the results of the analysis results and might be in from of charts, bars or visualized graphs.

### 3.2.3 Social Networks Analysis in Research Area

In the research area of social networks analysis, it is usually the main task about how to extract social networks from different communication resources [47] [72]. The data that used for building social networks is relational data [91], which can be obtained and transferred from different resources including the web, email communication, internet relay chats, telephone communications, organization and business events, etc [14]. For example, the email communication is a rich source for extracting and constructing social networks. In the issue of email social networks extraction, the relationship between email senders and receivers can be transformed by measuring the frequency of email communication with take the communication behavior (such as reply, forward, etc.) into account [18]. The transformed relational data can then be used for social networks construction.

In the past three decades, social network analysis has developed a range of concepts and methods for detecting structural patterns, identifying patterns of different types of relationship interrelate, analyzing the implications that structural patterns for the behavior of network members, studying the impact on social structures of network members and their social relationships [29] [91] [102].

## 3.3 Social Network Data Integration

Social networks provide for the sharing of huge amount of information between actors. Real examples of social networks can include: a group of employees in a large organization, with links joining people who work on the same project; a set of scientists in a particular field, with links people who co-authored papers; or a set of leaders in a business, with links represent people who worked together within a directors board. Some social network data gathered by researchers has been made available on-line to be used by other researchers. For example, Zachary's karate club dataset [106] is a popular freely available on-line dataset, where nodes correspond to members and edges correspond to interaction between those members. Another example is the Football network [37], where nodes correspond to teams and edges indicate if they were involved in a match against each other or not. Others real world networks such as the Les Miserables [75] or the Dolphins network [68] can be found in [61] or via Newman's home page [79]. As for the on-line social networks like Twitter or Facebook, a crawling process is frequently applied. This is the process to extract the user profiles from the network one by one using crawling techniques like Breadth First Search (BFS) and Depth First Search (DFS) [100]. Other data can be collected from different sources, such as relational databases, or graph databases and can be in different forms, either as structured or unstructured data files.

## 3.4 Social Network Graphs

Different mathematical methods are used to represent the network data for the analysis process. Both matrices and graph theory act as concrete foundations for many concepts in the analysis of social networks. Many researchers use the graph notation as a starting point for the analysis of a network. This plays a substantial role in understanding the data within the network and studying the results of the analysis. Graph representation includes three different types of data: (a) the nodes which can correspond to individuals, organizations or groups, (b) the edges which represent the relationship across the nodes and (c) the attributes which represent the different characteristics or properties either for the nodes or the edges. In addition to, these graphs capture the nature (e.g. density) and the direction of relationships (e.g. directed, undirected). Models based on graphs have been used frequently to do analytics on social networks using various types of graph representation. Different types of graphs such as directed, undirected and weighted depend on the kind of the network itself. For a Facebook network, if a user becomes friend with another, then both agreed to have a relationship, therefore the network is considered undirected. While for a Twitter network, a user can follow another one without having his consent and without being followed back from the other user. This case defines the direction of the edge. The weights in some type of networks play an essential role in assessing the quality of the relation as they reflect the frequency and the level of interaction, for example, the frequency of email contacts between two actors in an email network. Representing social networks as graphs is known as socio-grams in the field of sociology. Combining these graphs with some concepts and measures from graph theory can provide a visualized description for the data and the information within the social network, especially for small size graphs. For large graphs, the high density of edges make it difficult to provide an informative visual picture. Generally, graph theory is essential for analyzing social networks. It is used in the analysis of these networks to determine the most common properties and features of the network, of the nodes and of the links. It clarifies some important insights about actors; influencers, experts and trustworthy people [45] or active, engaged and initiator actors in the network [88], this in return will have a great influence on activities and benefit opinions or even decisions made by decision makers or management.

## 3.5 Network Analytics using Graphs

In this section we clarify how the graph theory concepts, which were presented in the previous chapter, are used as analytical measures for a social network.

We give the definition of the most popular metrics that can be used for analyzing any network and which are used in our experiments in the next chapters. We cover the local properties that corresponds to the individual and some global properties that corresponds to the network as a whole.

### 3.5.1 Preliminaries

Analysis of social networks can be applied in any field that can be modeled as a graph G = (V, E). These graphs represent a rich source of information. In on-line social networks, the vertices (nodes) represent actors and the links (edges) represent interactions or relationships among actors. The edges between the nodes can either be directed, which means that there is a source node and target node for each link, or undirected, which means there is a edge between two nodes without specifying the source and the target. The degree of the node is the number of edges to other nodes in the graph. For directed networks, there may be a differentiation between in-degree (the number of coming edges) and out-degree (the number of leaving edges).

### 3.5.2 Topological Properties

The network properties are the properties that provide information about the structure of the network as a whole not to specific individuals in the network. They aggregate the entire relational and behavioral interactions between the individuals in the network. These measures can describe the size and the overall structure of the network. Other network properties relate to node and link properties and illustrate different connection patterns between people, identify the roles and highlight important people, groups and events. All of the following metrics can help in reporting and giving insights on leadership, health, organization and hierarchy. Here are definitions of some of popular properties which are used in many research works and in our thesis.

- **size.** The size of the network is commonly used to find the way the network can be analyzed and crawled or obtained. It is defined as the number of nodes within the network. For instance, small sized network can easily be gathered and analyzed but on the other side, large sized networks need automated methods. Thus, size is considered an important feature that provide researchers with insight about the network. The type of the network is considered the main factor that affects the number of nodes in the network [100].

- **Diameter.** It is the longest path among all shortest paths calculated between actors. Diameter affects the speed of the diffusion of information within the network.

- **Small world.** One of the common properties in many networks. This property is used to refer to two properties: the *small distance;* when two people are joined through a short chain of reciprocal acquaintances and the *clustering effect;* when two people are likely to know each other when they share a common neighbor.

- **Degree Distribution.** An important feature of networks is how the links are distributed over the nodes in the network.

- **Power law.** The power law confirms that the number of vertices with degree $k$ is proportional to $k^{-\gamma}$
  for some exponent $\gamma > 1$. The power involves using one parameter only (the exponent) to illustrate the distribution of degree for billions of nodes and facilitate applying a comprehensive analysis of these networks It is a first-order estimate and an important basic to understand networks. The power law distribution can be illustrated in the normal scale or logarithmic scale. The precise distribution follow a power-law form [76]. Networks with power-law distribution are called scale-free networks because their degree distribution depends on properties other than the size of the network. The formal definition of the degree distribution P($k$) is the probability for a fraction of nodes in a network having degree $k$. A common property is popular in social networks which is the power-law degree distribution the is denoted as:

  $$P(k) \sim k^{-\gamma} \tag{3.1}$$

  Where $k$ represents the degree , $\gamma$ the exponent and P($k$) is the degree distribution.

- **Triads Count.** The aim of this algorithm is to count all the triangles or in other words the cliques of size 3. Counting the triads can be beneficial for many graph algorithms because they can be used to view similarity between structure of graphs [86] and can also be useful for detecting the communities (groups) in the network [82]. Hence counting triads is considered as a graph metric that is the basis of other analysis algorithms.

- **Centralization.** Centralization has a crucial importance in social networks analysis as it uncovers the persons who have critical positions within the network. For instance, leaders and popular actors occupy central positions in the network. The most commonly used centrality measure in social network analysis community are degree, betweenness, closeness which were proposed by Freeman [33] and eigenvector centrality which was proposed by Bonacich [9].

  1. *Degree Centrality* is the number of direct ties that involve a given node. It is based on the idea that an important node is the one with largest number of links relative to other nodes in the graph. According to Freeman, it is defined as:

     $$C_{\mathrm{d}}(i) = \sum_{j=1}^{N} A_{\mathrm{ij}} \tag{3.2}$$

     Degree centrality indicates the activity of the direct relations to the node i. N is the number of nodes in the network, A is the adjacency matrix where $\mathrm{A_{ij}} = 1$ if there is a link between the nodes i and j and $\mathrm{A_{ij}} = 0$ if there is not a link between these nodes. The measure focuses on the most observable actors in the network. Actors with low degree act as peripherals while actors with high degree act as a main channel of information within the network.

  2. *Closeness Centrality* is a measure which is based on the minimum geodesic distance d(i, j). The geodesic distance is the minimum length of an indirect path from i to j where i $\neq$ j according to Freeman definition distance. According to Freeman, it is defined as:

     $$C_{\mathrm{c}}(i) = \frac{1}{\sum_{j=1}^{N} d(i,j)} \tag{3.3}$$

     Closeness centrality indicates the freedom of the node i to control actions of others. N is the number of nodes in the network and d(i, j) is the distance between node i and other nodes. The measure focuses on how close an actor is to all other actors within the network. The idea of closeness is inversely proportion to distance. As the distance of a node from other nodes increases (more geodesics linking a node to other nodes), the closeness centrality will decreases.

  3. *Betweenness Centrality* Betweenness Centrality is the number of shortest paths between all the pairs of nodes that passes through this edge in the graph. It is based on the concept of minimum geodesic distance. According to Freeman, it is defined as:

     $$C_{\mathrm{b}}(i) = \sum_{j \neq k} \frac{g_{\mathrm{jk}}(i)}{g_{\mathrm{jk}}} \tag{3.4}$$

Where $\mathrm{g_{jk}}(i)$ is the number of shortest paths between $j$ and $k$ passing through $i$ and $\mathrm{g_{jk}}$ is the total number of shortest paths between $j$ and $k$ where $j \neq k$. Betweenness centrality indicates the intermediate location of a node and its ability to limit or facilitate interactions within the nodes it links. The measure focuses that the actor must be between many of the other actors through geodesic distance.

4. *Eigenvector Centrality* is a positive eigen vector of adjacent matrix. According to Bonacich, the eigenvector centrality $\mathrm{C_e}$(i) of a node i is defined as:

$$C_{\mathrm{e}}(i) = \frac{1}{\lambda} \sum_{j=1}^{N} A_{\mathrm{ij}} C_{\mathrm{e}}(j) \;\; \forall i \tag{3.5}$$

Where $\lambda$ is a constant, N is the total number of nodes, $\mathrm{A_{ij}}$ is the adjacency matrix of an undirected (connected) graph and $\mathrm{C_e}(i)$, $\mathrm{C_e}(j)$ are the scores of the $\mathrm{i^{th}}$ and the $\mathrm{j^{th}}$ node respectively.

A node is more central if it is in relation with nodes that are central themselves. Thus, it can be claimed that the centrality of a node not only depends on the number of its adjacent nodes, but also, on their value of centrality.

- **Detecting Communities.** A community or a cluster is defined as a group of nodes that have a better connection within a group and sparsely connected with other groups in the network. There are numerous community detection algorithms for finding communities within the network [63]. These algorithms assign nodes to communities when there is no available ground truth.

- **Modularity.** This metric is proposed by Newman [77]. The modularity metric is used when examining communities with the network to evaluate and asses the quality of dividing the network into communities. Modularity ranges from -1 to 1, and is 0 when no community structure. Practically, a real network with significant community structure can have a modularity from 0.3 or more [77]. This metrric is defined in [56] as:

$$Q = \frac{1}{2m} \sum_{ij} \left( A_{\mathrm{ij}} - \frac{K_{\mathrm{i}} K_{\mathrm{j}}}{2m} \right) \delta(c_{\mathrm{i}}, c_{\mathrm{j}}) \tag{3.6}$$

where $\mathrm{A}_{ij}$ is the adjacency matrix of the network between node $i$ and $j$, $\mathrm{K}_i$ is the degree of node $i$ and $\mathrm{c}_i$ is the community of node $i$ and $\delta(u,\ v) = 1$ if $u = v$ and $\delta(u,\ v) = 0$ if otherwise.

- **Transitivity.** It is the average clustering coefficient of the all the nodes within the graph. It ranges between 0 to 1 where high values correspond to high cliquishness,clustering coefficient with value 1 corresponds to perfect cliques and clustering coefficient with 0 corresponds to no triangles found between connected nodes. It is defined as:

$$C = \frac{1}{V} \sum_{i \in V} C(i) \tag{3.7}$$

While the internal transitivity within a group or community depends on how the direct neighbors of a certain node are connected. It is the actual number of links (edges) between neighbors, divided by all the possible links if they are all connected. The internal transitivity $T$ of a community $C$ is defined as:

$$T = \frac{1}{N_{\mathrm{C}}} \sum_{i \in C} \frac{2 * l(i)}{K_{in}(i)[K_{in}(i) - 1]} \tag{3.8}$$

where $\mathrm{N_C}$ is the number of nodes in community C, l($i$) is the number of actual links between neighbors of the node $i$ and $\mathrm{K}_{in}(i)$ is the indegree of the node $i$.

- **Edge Density.** The edge density $\mathcal{P}$ of a community $C$ of an unweighted graph is the ratio between the actual realized links in the community $\mathrm{E_C}$ to the maximum number of possible links it can contain if all the nodes are well connected to each other $\mathrm{N_C}(\mathrm{N_C}$ - 1)/2 where $\mathrm{N_C}$ is the number of nodes in the community. Communities are supposed to be higher in density than the whole network. The density function is defined as:

$$\mathcal{P} = \frac{2E_{\mathrm{C}}}{N_{\mathrm{C}}(N_{\mathrm{C}} - 1)} \tag{3.9}$$

- **Connected Components.** In the case of undirected graphs, a connected components is a subset of nodes so that there is a path between each pair of nodes. While for directed graph, a differentiation is made between a *strongly connected component* (SCC) and a *weakly connected component* (WCC). Strongly connected component corresponds to set of nodes where a path exists between all the pairs in the set. In contrast, weakly connected component corresponds to a set of nodes where a path exists between all pairs in the set if all the links were viewed as undirected in the network. Studies have shown that there is usually a dominant strongly connected component within the network [12].

## 3.6 Mining Social Network

As discussed before, social networks provide a wealth of information that can be transformed into valuable insights about the dynamic and the structure of the network. There are some typical questions researchers should ask when analyzing a social network. Some of these questions are: Who knows who in the network and how strong is the relationship? How well do people know each other's knowledge, expertise and skills? Who is the source of information and spread the word? What resources do people use to share information? How could the pattern of individual choices derives more holistic patterns, may be using predictive modeling (e.g. correlation and regression) on network data? Answering these types of questions is what has been known as *mining a social network*.

From our preservative, and based on the existing literature, we categorize the mining areas of social network into four popular categories: Node Mining, Link Mining, Content Mining and Community Mining, and provide a literature review of some of the existing work in each category.

### 3.6.1 Node Mining

Node mining is a type of user behavioral analysis to find patterns in the network, predict popularity, actions of actors [97], and find influencers [17]. It studies how actors are embedded and located in the overall network. Several studies have been focused on identifying the role of actors in the network [22]. A quantitative study of the topological characteristics is proposed in [52] to report the results of the participation of users within a Twitter network. The study includes ranking users to find the top influential people within the network based on their number of followers and page rank. Rowe [89] presented an algorithm based on mining an email network of the Enron corporation, one of the well-known publicly available email dataset for real corporation. His analysis focused on analyzing the behavior of users and finding the communications patterns to extract hierarchal levels that reflect organization work chart and define the main players within the network. [26] studied the period of Enron's crises to find the structural features of the network, extract structural properties and find the key players within this period. Varshney [23] analysed the level of response of employees based on email exchanges. Diesner [27] conducted a study to find the communication patterns via their identified hierarchical structure. Marlow [71] identifies authoritative blog authors by investigating to what extent the blogs are inter-connected.

### 3.6.2 Link Mining

Several studies have been focused on the quality of ties between actors [22] There are numerous methods that are based on the number and the density of ties to find influential, reputable, authoritative and central people in the social network. The output of these methods is usually a ranking score which corresponds to a reputation or social prestige from the social network perspective. **PageRank** is one of the popular methods which is best known as it is used by Google to rank web pages [81]. It derives scores not only for specific nodes, but also for those connected to them. Gynogyi [40] proposed a method called **TrustRank**. The method derives a trust score for each node. Another method, introduced by [49] is called **HITS**. This method requires computing two scores for each node, the first is called *hub score* for the node with many outlinks and the second is *authority score* for the node with many inlinks. Generally, methods like PageRank, TrustRank and HITS have been used by many researchers to find "interesting" people in a social network [84]. They are also used to find expert people in the network [15]. Other ranking methods like betweenness centrality [11] and eigenvector centrality [93] are also commonly used.

Links or relationships in social networks often contain patterns that can correspond to properties like the rank or the importance of the object. Sometimes, it is desirable to predict the existence of a link when it is not observed or to predict if the link will come into existence over time as the network is evolving. This is why the link prediction problem has also gained attention in the literature. Link prediction is the problem of inferring missing links based on the observed network. The problem is formalized in [66]. They propose link prediction approach that concludes that future interaction can often be predicted based on the topology of the network and measures for detecting the proximity of nodes within the network.

### 3.6.3 Content Mining

Mining the content within the network is a way to discover useful information, content features and classify content into topics. For example, in [52], tweets are ranked and analysed based on their active duration to find the most trendy topics, where most of the topics appeared to be headline of news. Generally, this work covered several questions like: What topics do they talk about? How is the information diffusing within the network?. Leskovec et al. [58] shows that the dynamics of popular topics in online social networks are made up from succesive focus and defocus on topics and that's result into information diffusion in the networks. Mccallum et al. [73] discovers topics within discussion based on sender-recepient relationship in email network and this combines the connectivety within the network with topic clustering. Gloor et al. [38] aims to improve data quality and discovers insights within an enterprise through mining content in communication archives such as blogs, instant messages and emails.

### 3.6.4 Community Mining

Over the recent years Community detection has attracted attention of researchers enormously in terms of the different proposed community detection algorithms. A community can be group of people who may be interested in same topic and contribute to each other through post or replies. Identifying communities has a crucial value in network analysis. They can reveal a priori unknown information related to topics or cyber-communities. Tsagkias [99] focuses on prediction of community activity using eight different new networks and finds the pattern across each network. Rowe et al. [88] proposed an analysis of community types and based on dynamics of community behavior using one single network. Gibson [36] proposed a technique to derive hyper-linked communities in Web environments, which includes hubs and authorities identification which are discussed above in Subsection 3.6.2.

Generally, the description of network structure for complex networks has been studied in different ways in the field network analytics. Finding the community structure is considered in between two different levels. It can be node-based level which involve finding the properties of nodes (centrality, degree and so on) and can be *network-based* level which involve finding the whole network properties (clustering coefficient, degree distribution). Generally, the basic definition of community structure is introduced in [78] by Girvan and Newman where they defined it as "*The division of network nodes into groups within which the network connections are dense, but between which are sparser*". Based on this definition, nodes should be densely connected to each other within the community while having few links to nodes in different communities. A hierarchal phenomena can even exist such that different levels of sub-communities can exist within each community.

There is no specific requirements for community size but it often follows the power-law distribution [19]. Leskovec [62] observed that in real networks, communities blend more with the rest of the network as the community size increase. This consequently reduces the appearance of communities and their quality. Thus, the focus is good to be mainly on the densely small sized communities. A very useful overview about community structure in networks is proposed by Fortunato in [32]. Besides, the dynamics and the evolution of communities which makes the analysis process of finding communities over time more challenging as it requires applying clustering algorithms at different timesteps which will results in independent communities that can be hard to link over time.

## 3.7 Network Analysis Challenges

Despite the potential benefits of analysing social networks, there are many challenges to mapping and analysing actual real world relationships. There are a wide range of challenges in the literature in this area that can be rich for research. In this section, we tried to categorize the social network challenges from the literature by trying to highlight some common challenges in area of social networks and categorize them from our own perspective. These challenges will mostly relate to our contribution in the rest of the thesis. We will provide in the next chapters an in-detail discussions about these challenges and propose our contribution which we aim to unlock some challenges and can be a way forward in areas related to community structure, evaluation, and networks dynamics in social network analysis.

### 3.7.1 Data Collection and Preparation

Collecting and extraction of data can be done either manually through self-reporting and interviews or using sensors, web-crawling, automated network extraction methods, for example, relation extraction or entity extraction [80]. Collecting personal data is not free, and we want to make sure that we get the best value from the data. There are many challenges to collect activity within the network. Standardizing on methodology, models, and tooling could significantly reduce the effort, risk, and cost of collecting such data. It is a challenging process as it requires having a clear picture about the available data, its format, who have the rights over this data and if there is any missing or poorly documented relevant data. Even for available data there might be complexity in the structure of the data and its records may include many fields that are not relevant from social network analysis perspective. Additional problems might arise like: duplicate nodes, for instance, a single node with two different emails or a person who is no longer in the network or removed his account and his profile remain active. Therefore, data cleaning and pre-processing is a required step before the analysis.

### 3.7.2 Evaluation and Validation

Choosing an evaluation metric in a principled way is difficult, as often collecting and sharing data is not easy even if the data is available as reaching a ground truth for validation is difficult. This problem, in the data mining literature, is known as lack of ground truth. Besides, the novel techniques in social media research, there is a need for evaluation without a standard reference or ground truth.

Many researchers approach ground truth problem through surveys to identify knowledge flows and relationships to people to derive a hypothetical scenarios [45]. However these scenarios might indicates the way people think but not necessarily reflects their actual behavior. These kinds of techniques do not assure validation, but reduce the percentage of error during the analysis process and help in validating the insights to some extent.

### 3.7.3 Community Detection

One of the common problems and challenges in the field are community detection problems and their evaluation. Many researchers devoted their research work to discover communities in social networks (graph clustering). The problem arises in two ways: the first is ensuring the accuracy and the second is ensuring the quality of communities [8]. When ground truth is available, assuring accuracy is achieved when the objective is to check and compare to the actual communities. This can generate an accuracy score which can correspond to accuracy metrics [85]. Assuring both the accuracy and the quality is not an easy task if ground truth is not available [8]. As ensuring quality is achieved through measuring the feasibility of the community structure depends on the connectivity inside the community relative to connectivity to other objects outside the community within the network. Thus, the scores generated by quality metrics are based on the structure of the networks and the community structure within the network [90].

Several community detection approaches/algorithms have been proposed to find communities and groups in the network based on graph theory concepts. However, one of the main challenges is to find the optimal number of communities and the appropriate community structure for structureless networks or unstructured communities. There is challenge of comparing different community results and decide which one is the optimum with hidden community structure. There are various functions proposed with the aim to compare results or find the optimum one within a given network.

### 3.7.4 Evolving and Dynamic Networks

Many researchers have focused their attention on the evolution and the dynamics of social networks. It was observed that most of the networks turned to be much denser across time due to the time stamped datasets [31]. Consequently there is a super linear increase in the number of nodes and the number of edges within these networks [60]. A lot of work focused on this dynamic change in the data which can reveal a new type of information and uncover the interaction between communities. Another issue that is getting more popular is the study of graphs evolving in time. This is now possible due to the availability of timestamped network data sets. Tracking the evolution of community structure in time is very important, to uncover how communities are generated and how they interact with each other. Several steps have been taken in consideration with the evolution of data which will lead consequently to the evolution of graphs to achieve an efficient mining process. However, there is a common problem which is the lack or the high cost of big data technologies and having insufficient computing resources or clusters that can handle the huge size of datasets. Despite the fact that there are many open source and web formats that can help in the preparation of this data but there is always a problem facing the management of large scale datasets.

### 3.7.5 Blinding Decision

Active networks are the networks that represent active engagement of actors in the network (tags, likes, comment, etc). This active participation can shape and influence the network structure or the relation between nodes. The more active ties or engagement in the network, the more information and insights derived. The active relations play a substantial role in breaking down traditional hierarchies and silos as well as informing insights or decisions. While passive relations such as friendship, follow, etc. within the network may result into information loss between its nodes and might lead to a potential problem of deriving insights about people. For instance, an employee in an organization might have a network of friends, but he might be not a fan of using social networks. If decisions will be based on the network structure only, then this will blind having real insights about this person. Trying to friend many people as possible does not reflect the real communications and ties. Social media in some cases does not reflect the actual social interactions and experience. Passive behaviors (browsing, reading other's stuff, etc.) or passive relations (friend, follow, etc.) may not be a real interaction. These types of relations just acknowledge that you share a space with others. Another problem which is the articulation of the right boundaries of the network through filtering and selection, defining the boundaries and the partiality of the network (who is in and who is out) actually matters to reflect a qualitatively informed understanding of the nature and the characteristics of the social network structure. Also, missing relations or missing nodes in the network can affect the derived decisions as they hide information about their effect on the whole structure of the network.

### 3.7.6 Privacy and Ethics

Despite the fact that collecting data from on-line and off-line sources is much easier than before, but, still there are many challenges facing researchers when they need to use data for development or for management the streaming of data. Preparation of data is an essential step as discussed before not only for efficiency, but also, for anonymization process to overcome privacy issues. Some common new challenges like computational complexity, security issues arise when revealing a sensitive and confidential information about people or organization. Data mining paradigms have been proposed to perform mining tasks taking into account data protection to preserve privacy of data or personal information. Networks or graphs can be rich sources of data that discovers and reveals information about personal identity and insights about users. Defining the right balance between hiding data and disclosing data is suggested through many approaches. This approaches include auditing queries [48] and sanitization of [5]. Removing names or identification number from the data is not sufficient, as the structure of these graphs can reveal and reflect information about the individuals themselves [6]. The main challenge is how to anonimize data by hiding personal information or sensitive structure in addition to having a useful data to recover useful insights. Data protection can impact our ability to collect personal data from the network, that's why mechanisms are needed to model, collect, and manage consent of data.

In addition to privacy, there are ethical and legal issues. For example, ethical policies in an enterprise, laws related to data across countries or even from industry to another industry. All of these cases wil have severe impacts and limitations on the analysis process and what exactly can be done on the data.

## 3.8 Summary

After presenting a background on the graph theory concepts in the previous chapter which are used as a mathematical base for the work proposed in this thesis. In this Chapter, we focused on presenting an overview of the social network analysis. First, we introduced the definitions and the concepts of social network analysis. Then, we discussed the type of data presented in social networks and how this data can be represented in social graphs. Also, we reviewed the analytics that can be used for mining social networks and discussed the different areas of mining. Finally, a special focus has been given to the challenges in the social network analysis area which will be the core of our contribution in the next chapters.

# Part 2

# METHODOLOGY AND APPLICATION

Chapter 4

# Analysing and Predicting the runtime of Social Graphs

## Contents

In this chapter, we introduce our predictive technique that can address the evolving graphs problem in the area of social networks. The model provides an estimated execution time for the end user indicating the analysis that can be done on a given network without requiring any information about the network except the number of nodes and edges.

## 4.1 Motivation

The explosion of Social Network Analysis in many different areas and the growing need for powerful data analysis has emphasized the importance of in-memory big data processing in computer systems. Particularly, large-scale graphs are gaining much more attention due to their wide range of application. This rise, accompanied by a massive number of vertices and edges, led computations to become increasingly expensive and time consuming. That is why there is a move towards distributed systems or Big Data cluster(s) to provide the required computational power and memory to handle such demand of huge graphs. Thus, figuring out whether a new social graph dataset can be processed successfully on a personal machine or there is a need for a distributed system or big-memory machine is still an interesting question. In this chapter, we try to address this question by providing a comparative analysis for the performance of two of the most well known SNA tools: Gremlin [87] and SNAP [64] for performing commonly used graph algorithms such as counting Triads, calculating Degree Distribution and finding Clusters which can give an indication of the possibility of carrying out the work on a personal machine. Based on these measurements, we train different supervised machine learning models for predicting the execution time of these algorithms. We compare the accuracy of the different machine learning models and provide the details of the most accurate model that can be exploited by end users to better estimate the execution time expected for processing new social graphs on a personal machine.

## 4.2 Evolving Graphs Problem

In recent years, there has been an observed increase in the number of large on-line social networks, many of them have a massive number of users that can reach hundreds of millions of users [94]. Analysing these social network databases can provide rich information which can be beneficial for a wide range of applications and areas but is sometimes considered expensive and time consuming. This is due to the expected super-linear increase in the computational time and therefore speed and scalability should be key challenges of social network analysis.

In order to handle huge real-world network analysis problems, distributed clusters may be required to accommodate "real-world" graph sizes. Alternatively, big-memory machines that can do a highly interactive analysis that can have advantages over distributed clusters [83]. A long computational time may be needed to handle any graph analytics like community detection, node ranking, computing shortest paths, number of triads, degree distribution and connected components. The need to have an efficient computational tool/model or even a query language to use with this social graphs has been addressed by many researchers e.g. [94], [28], [105].

What if we have an option to run our analytics on different computational platforms? How could we predict which platform is most suitable? This will be the goal of our work in this chapter, to be able to predict the execution time of graph algorithms for unseen graphs using two of the most commonly social network analysis tools as an example, but to reach this predictive execution time we will need first to know how currently available tools perform on a personal machine. There are many social network tools and libraries that can perform a set of operations, features and various algorithms with many functionalities for graph analysis of this rich data and information within the graphs. For example, SNAP[1], Gephi[2], NetwrokX[3] and Gremlin[4] are some

[1] http://snap.stanford.edu/snap

[2] https://gephi.org

[3] https://networkx.github.io

[4] https://github.com/tinkerpop/gremlin/wiki

of the most commonly used tools in the community.

## 4.3 Big Data and Graph Analytics

Perez et al. in [83] proposed Ringo, a big-memory graph analytics tool, which supports interactive graph analytics of millions of edges through merging big-memory machines that can outperform all other distributed systems. The authors showed that a single machine with big-memory can provide an efficient platform for doing graph analytics. Distributed graph systems like Pregel that support parallel graph algorithms on multiple machines and support adoption of a Bulk Synchronous Parallel (BSP) model was proposed by Malewicz et al. in [70] and also GraphLab in [67], a distributed system for data mining and machine learning. In [53] Kyrola et al. presented the performance of GraphChi, a disk-based system on a PC that supports evolving graphs overtime, GraphChi has a low memory requirement which was designed specially for computation on big-scale graphs. The authors performed a comparison between GraphChi and other distributed systems like Spark [107], Hadoop [2], PowerGraph [39] and GraphLab, it was found that PowerGraph can compute graph analytics using large cluster much more faster than using GraphChi on just a single machine. The comparison was performed on PageRank, one of the most popular graph algorithms. It was shown that GraphChi can provide high performance for different purposes.

Seo et al. in [95] claimed that the performance of Datalog, a declarative logic programming language that is usually used as a query language [46], is not competitive with other low-level languages in the past. However, it allows the expression of many graph algorithms and supports recursion and high-level semantics which consequently allow optimization in time and parallelization. A high level query language for graph analytics named SociaLite was proposed by the authors as a Datalog extension for powerful analysis on graphs. The authors performed a comparison for the execution times for running a shortest path graph algorithm on different benchmarks like Giraph [1] and Hadoop and then compared their execution time with SociaLite concluding that the latter outperforms. The authors presented a comparison for the execution time in [94] between Datalog engines like Overlog [20], IRIS [3] and LogicBlox [4] for running shortest path algorithm on single machine, showing a better performance for LogicBlox. However when compared with SociaLite, the latter showed a better performance than LogicBlox. Also, the authors proposed a comparison of SociaLite with other implementations in java of almost 50%.

We have found that tackling the performance issue to predict an estimated time needed for analysing graphs is a new area that can be fruitful for detecting the execution time of evolving graphs. To the best of our knowledge, this is the first comparative analysis that aims to find an estimate prediction for the execution time of graph analytics based on different benchmarks using a PC.

## 4.4 Experimental Components

In this section, we introduce the tools, the measures and the predictive models used in the experiments. We give a brief description of each of them and we highlight the purpose of selecting each of them in our technique.

### 4.4.1 Investigated Tools

As we descibed, the increase of Social Network Analysis is driven by the rise of on-line networks specially human networks [25]. This rise has driven many researchers and developers to create and develop different approaches, algorithms and tools to easily apply graph mining and analytics. This led to having a plentiful supply of publicly available frameworks that have many algorithms supporting the study and manipulation of data for any type of network. Our main concern will be how to select the right tool for the observed large-scale evolving graphs and decide which tool can suit your system design, graph size or even the algorithms that are to be used.

Our experiments will target a comparison between examples of two different types of graph analytics tools. We have chosen to conduct this comparison between QueryLanguage-Based tools like Gremlin and Software-Based tools like SNAP. Query language tools are based on query language which can be used for generic purposes and enable many users to do social network queries in an easy and professional way without having a software background [94]. While, other Software tools like what mentioned before can be C++ and Python-based, so they are supposed to have a better computational time. We concisely summarize the features of both tools as below:

- **Query Based Tools**
  *Gremlin* is an example of a Query-Based tool. It is a domain specific language for working with graphs, a graph based programming language developed for multirelational graphs, named property graphs. The following are the main features of Gremlin:

  1. Supports complex graph traversals.
  2. Works over different frameworks, graph databases and graph processors.
  3. Used within the Java language as a virtual machine that has a direct access to Java based application.
  4. Combines query language, network analysis and manipulation of graphs.
  5. Enables a wide range of users who do not have a software background to do efficient and easy queries.

- **Software Based Tools**
  *SNAP* is an example of a typical Software-Based tool. It is a free general purpose network analysis and graph mining based package tool with the following features:

1. It is written in C++.
2. Provides a Python interface (snap.py) [65] for use with Python and runs on Windows, Mac OS and Linux.
3. Scales to huge networks with hundreds of million of nodes and edges.
4. Calculates the graph's structural properties, provide standard graph algorithms and different network structure measures.

### 4.4.2 Graph Measures

We tested the execution time of three popular graph analysis algorithms which were discussed in detail in chapter 3 for both tools.

- **Triads Count**: The aim of this measure is to count all the triangles or in other words the cliques of size 3. Counting the triads can be beneficial for many graph algorithms because they can be used to view similarity between structure of graphs [86] and can also be useful in community detection [82].
- **Degree Distribution**: This is a simple measure to count the number of edges for each node in the graph, it is based on the concept of neighborhood to find the vertices that have the most direct links to other vertices.
- **Detecting Clusters**: The aim of this measure is to find communities (clusters) or know how many unique clusters and what is the distribution of vertices within each cluster.

The analysis of these graph measures and their execution time are implemented using Python language for SNAP tool and Groovy[5] scripts with Gremlin[6] language for Gremlin tool.

### 4.4.3 Predictive Modeling

With the continuous growth of data in various social graphs, learning how to take decisions based on this data to improve business or provide solutions is an important need. Consequently, learning a model for the performance issues using the number of nodes and number of edges as predictors for the model can be useful for making decisions on where to run a graph analysis job. That is our main reason for choosing to apply supervised machine learning algorithms to learn how to quickly a system can perform graph analytics.

Supervised learning is a useful and popular type of machine learning. There are several types of supervised machine learning algorithms that have been presented within the Machine Learning area like *classification*, *regression*, and *anomaly detection* [51]. Supervised machine learning algorithms are used to make predictions based on a set of features. Theyare used to find

[5] http://groovy-lang.org/

[6] http://gremlindocs.spmallette.documentup.com/

patterns in the data. Each algorithm looks for different types of patterns. After the algorithm has derived the best pattern, then this pattern is used to provide predictions for unlabeled testing data. Supervised learning can be applied on any type of data (e.g. financial, seasonal, geopolitical) and it has several applications [42].

According to [51], it was concluded that the following are the most suitable regression models that can be used to address this type of problem: Support Vector Machine (SVM), Multivariate Adaptive Regression Splines (MARS), M5 and Boosting. These are used in our experiments to train the results from the performance analysis in order to select the best model that will predict the execution time of unseen graph based on two main features that are usually known in any social graph: nodes and edges. Our target is to present an approximate model as a computational technique that provides a relation between the execution time and the structure of the network.

- **SVM**: The SVM model [51] is considered for both regression and classification problems based on detecting whether the data can be categorized or not. It is closely related to robust regression through reducing the effects of outliers in the regression process. Also, SVM is based on the value of the linear combination of the input features and capable for representing non-linear relationships in a linear fashion using a kernel function (RBF), which is a method for using a linear algorithm to solve non-linear problems.

- **MARS**: The MARS model [51] usually uncovers important data patterns effectively, it is more flexible than other regression models and does not require any data preparation. The nature of MARS features is that it splits the predictor into two groups and then start to model the linear relationships between the outcome in each group and the predictor. The MARS model can be interpreted easily through the existence of the hinge function which partitions the input data automatically and works more appropriate for numeric variables make them work efficiently for numeric data. A pair of hinge function is usually written as $h(x - a)$ and $h(a - x)$.

- **M5**: The M5 [51] is a model tree algorithm. It is used for the approximation and modeling complex non-linear problems. M5 is regarded as a promising model for prediction of numerical problems and is popular because of its robustness and efficiency where it can tackle tasks with very high dimensionality. The main implementation of this model is included in the Weka software package [41].

- **Boosting**: The Boosting model [51] is known to be a powerful predicting model. It combines the outputs of multiple weak regression models to obtain a better model for a final prediction. The algorithm is considered as a powerful prediction tool as it usually outperforms other individual models. The algorithm of the boosting model gets initialized at first with a best guess (e.g., the mean value) and then the gradient is calculated, then, a model is fit to minimize what is called as *loss function* and the current model is added to the previous one. This process is repeated according to the number of iterations specified by the user.

R scripts are implemented to apply predective modeling using R language [98] and Caret [50] library to train and test the performance results of the graph measures for SNAP and Gremlin. For tuning the parametres of the four models, we used the *trainControl* function which computes the default parameters for each model. As for the method used in each model, *svmRadial* method is used for SVM with RBF kernel function, *earth* method is used for MARS, *M5* is used for model tree and *gbm* is used for Boosting. Then, the selection of the models is assesed by measuring the **Root Mean Square Error (RMSE)**. It is a commonly used error metric to measure the performance of regression models. A linear regression model is fit with least squares, which means minimizing the RMSE.

## 4.5 Experimental Setup

In this section, we start by outlining our testing setup and environment. Then we give an overview of the dataset used and our terminology in preparing and preprocessing the data for the experiments. We provide a comparative analysis for the results based on the SNAP and Gremlin tools and how their computational times differ for the same graph metrics.

### 4.5.1 Setup

- **Test Machine**: Most of the experiments were done on an Apple Mac Pro computer, 2.4 GHz Intel core 2 Duo (2 cores), 3 MB cache size and 8 GB of main memory.
- **Test Protocol**: Each test is performed under the same setup and configuration. In the tests, we used multiple iterations and the mean execution time was reported. The datasets were held in-memory for each test. All our experiments were based on undirected graphs. It is worth noticing that the computational timing measured for both tools does not include the time taken to load the file into memory or writing the results into a file.

### 4.5.2 Facebook Dataset

The dataset considered in this chapter is a public available Facebook dataset collected by Stanford University [61], a freely available real world graph dataset. This dataset represents a list of friends from Facebook, it presents a political affiliation social graph between members. The data was anonymized be replacing the internal ids for each user by new value. The circles consist of 4039 vertices and 88234 edges. Each vertex represents a user and an edge exists if any two users have same political affiliations. To perform our experiments, we divided this data into subgraphs to compare the results on different sizes of graphs with growing number of nodes and edges. We will explain the subgraphing methodology in Section 4.6.

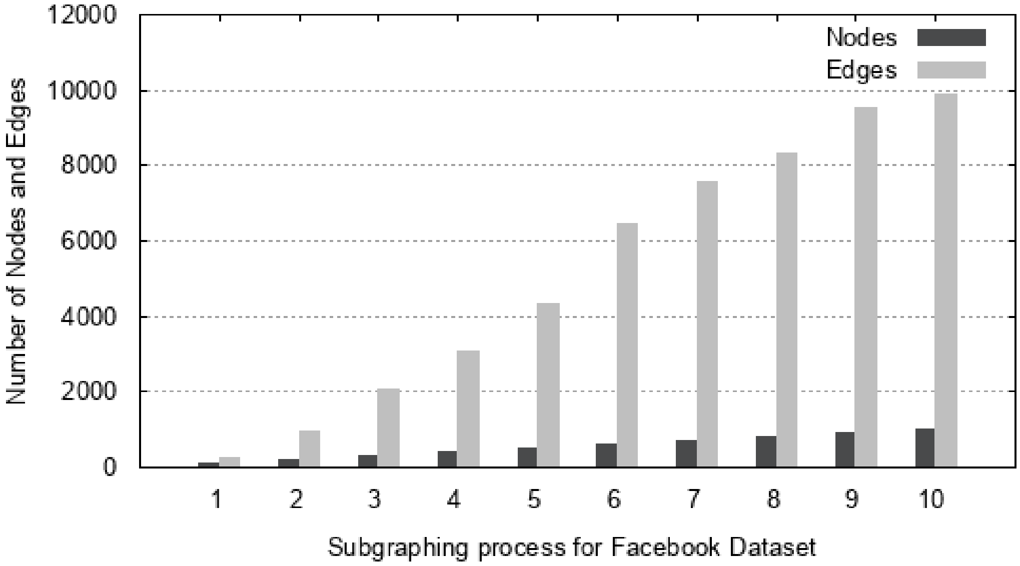


Figure 4.1: Subgraphing process for Facebook Dataset.

## 4.6 Evolving Graph Preparation

In order to test the performance of the tools and to simulate the evolution phenomena using one graph, a subgraphing process took place in order to extract subgraphs of varying sizes from the complete graph. We divided the Facebook dataset into subgraphs, each subgraph was represented as a selected number of nodes and all their associated edges or links between them in the whole network. The selection of the nodes IDs is based on their ID value in the graph. For instance, with a subgraph of 100 nodes, we select the nodes with ID values between 0 and 100. We extracted 10 subgraphs using snap.py library [65], by the main graph and specified nodes IDs in vector form as their parameters and returns a subgraph induced by the nodes specified in this vector and the edges between these nodes. We repeated this process for having different subgraph sizes. The resulting subgraphs are 10 times smaller in edge count and nearly 4 times smaller in node count of the whole graph. Our strategy selected: 100, 200, 300, ...1000 nodes and their associate edges so the network size varies from 100 to 1000 nodes and their edges numbers varies from 275 to 9890 edges as shown in Figure 4.1. Hence, we evaluated the computational time on a graph growing constantly. Given this is a Facebook graph, the graph type is undirected with no multiple edges or self loops. These subgraphs were represented as an edge list in different files ready for analysis.

## 4.7 Results and Discussion

In this section, we provide a discussion of the results for the performance analysis that has been conducted using different graph measures using two tools as discussed above. Then we presented the process of training and testing this performance analysis using four different supervised ma-

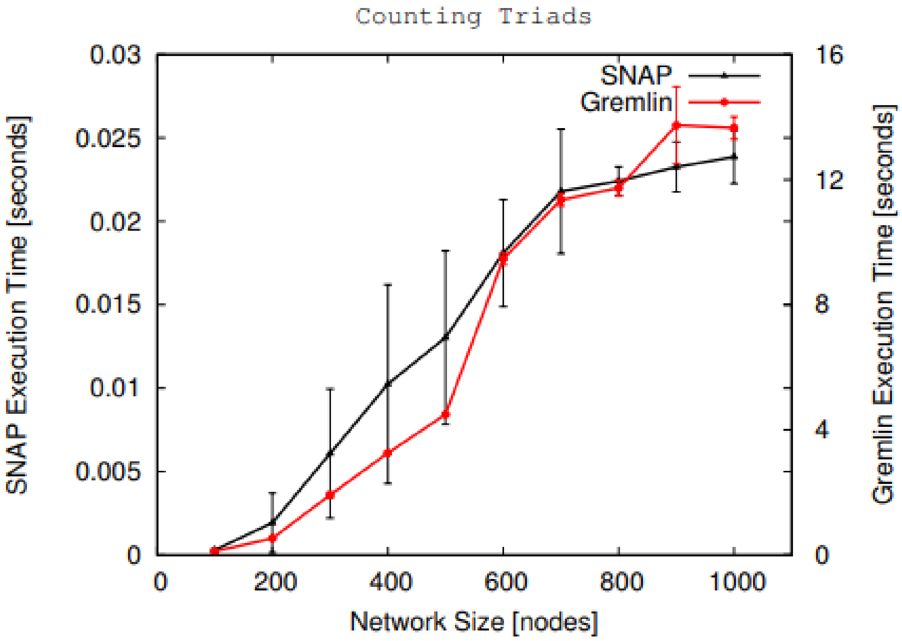


Figure 4.2: Execution time of counting triads for SNAP and Gremlin represented on different scales.

chine learning models as described above. We then asses the output of the model and evaluate them using RMSE measure.

### 4.7.1 Performance Results

This subsection presents the performance results of the tools discussed in the previous section. The comparison is based on the execution time measured by both tools to calculate different metrics on different graph sizes. We expect that the computational time will be affected by the structure and the size of the graph. The elapsed time for running the algorithms will differ based on whether these algorithms access the edge list one or more times.

Referring to Figures 4.2, 4.3 and 4.4, we measured the execution time taken by each tool to measure each of Triad Count, Degree Distribution and Detecting Clusters on different sizes of subgraphs. For the sake of increasing the accuracy of our reported results, we repeated each measurement for the execution time 20 times. The reason for the variance indicated by the error bars is seems to be due to runtime performance variability of the software and hardware. We repeated these tests with the same subgraph size by selecting different nodes to build the subgraph and found the execution time was similar. This indicates that the mean execution time is not affected much by the nodes selected and that the variation arises from runtime issues.

In the first test, for the *Triads Count* as shown in Figure 4.2, we found a major difference in the performance of both tools. SNAP performed much better than Gremlin, the execution time taken by Gremlin was nearly 13 seconds to traverse 1000 nodes and 9800 edges while in SNAP it

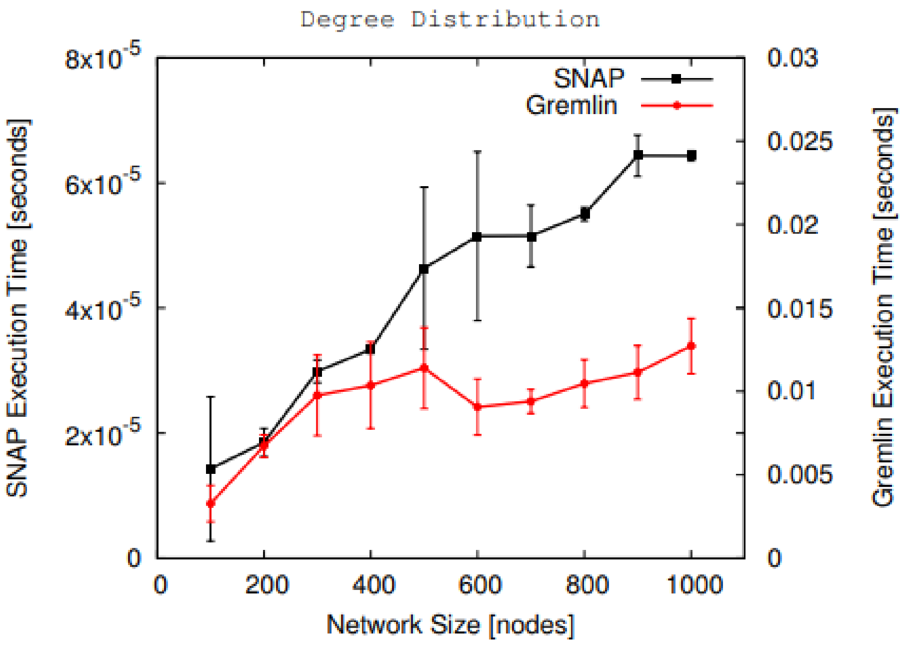


Figure 4.3: Execution time of degree distribution for SNAP and Gremlin represented on different scales.

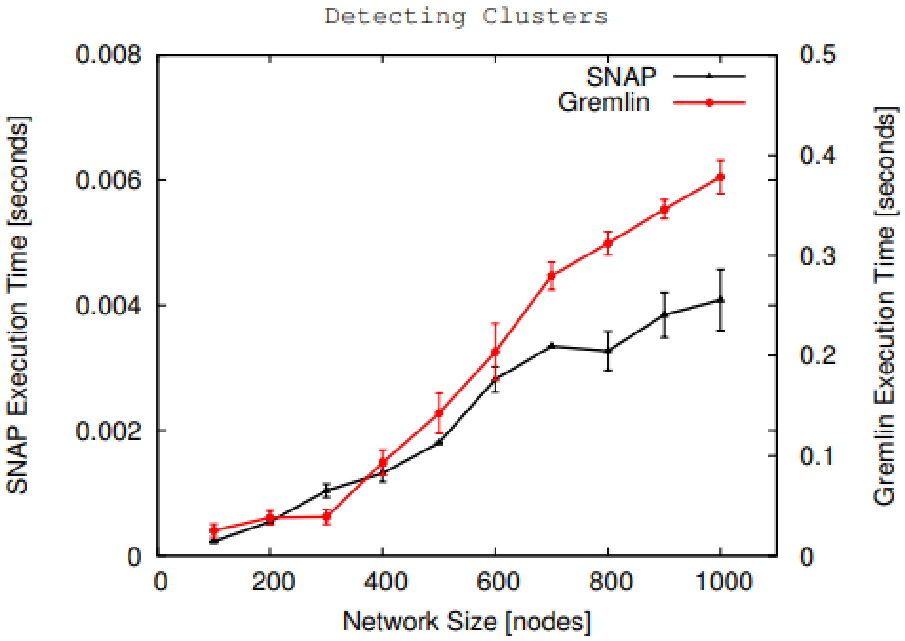


Figure 4.4: Execution time of detecting clusters for SNAP and Gremlin represented on different scales.

took nearly around 0.025 seconds and this is because the query used in Gremlin for counting the triads is likely touches and traverses every vertex in the graph to check their connection with their neighborhood and then check that their neighbors are connected so this led to traversing many vertices more than once. Counting triads of large-scale graphs usually require a fast algorithm, specially for graphs having billions of nodes and edges and it is preferable to be in a parallelized processes.

For the second test, the *Degree Distribution* as shown in Figure 4.3, both tools performed better compared to calculating the triads. Unexpectedly, we observed that initially the execution time of Gremlin was high for the small subgraphs then it started to decrease when approaching a graph size of 500 nodes. It is worth emphasizing that we repeated the same experiment multiple times but while the dip is within the error bars, there does seem to be a trend. We do not have a clear justification of this behavior. The Gremlin query here traversed all the vertices to get their edge count. As for SNAP, it performed normally with an observed linear increase with the size of the network.

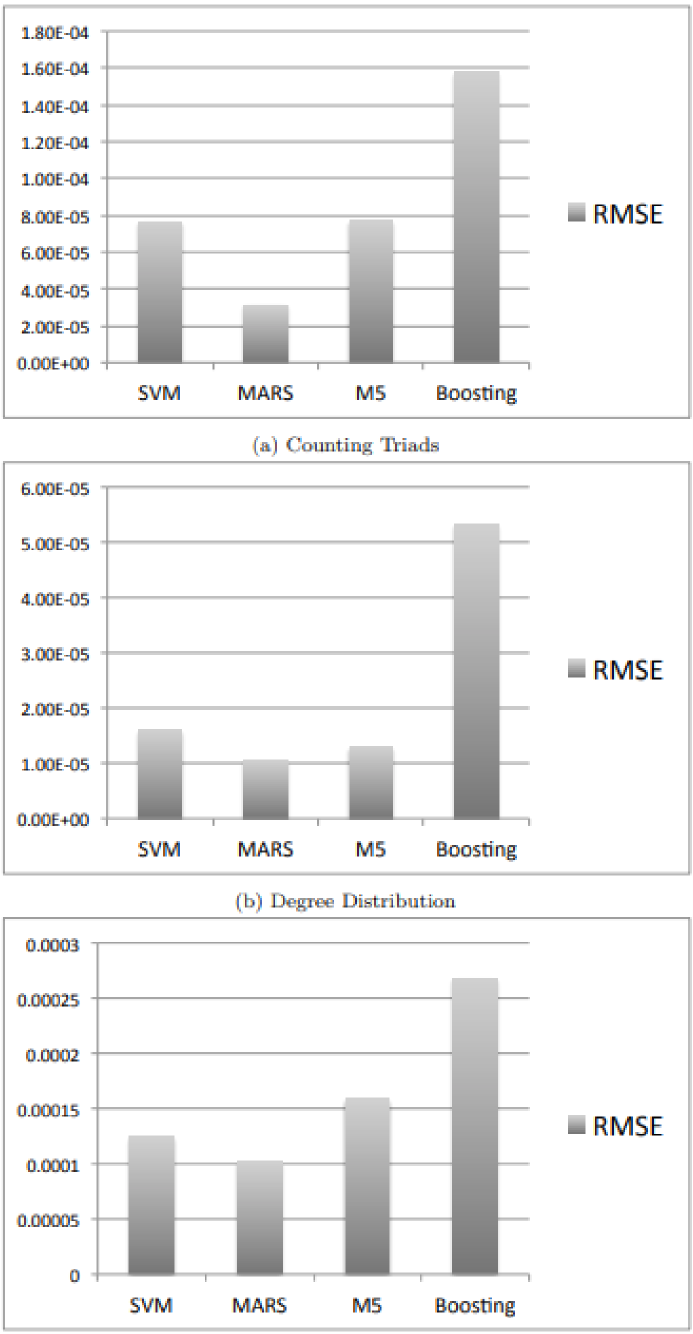
In the last test for *Detecting Clusters* as shown in Figure 4.4, we observed that using SNAP took around 0.004 seconds for a graph with 1000 nodes while Gremlin took 0.4 seconds for the same graph so it is clear that SNAP is 100 times faster than Gremlin. The algorithm used in SNAP is based on computing the average clustering coefficient for smallworld networks as defined by Watts and Strogatz [104]. While, the Gremlin query used in this test is based on the peer pressure vertex program algorithm, where every vertex assigned what is called by nominal value and if two vertices have same value therefore they are in same cluster and acquire the same cluster ID. Overall, SNAP performs much better than Gremlin in the three experiments.

### 4.7.2 Training Performance Results for Prediction

For the sake of training the machine learning models and achieving better accuracy, we extended our measurements to 50 various sizes of the same dataset and calculated the execution time for each of the three algorithms for both tools. In order to train and test the models, a general practice is to split the data into a *training set* (70 % of the dataset) and *test set* (30 % of the dataset). We applied one split method on our dataset formed from the 50 samples to a training set of 35 samples and 15 samples for the test set. We used the training set for estimating the coefficients of the different machine learning models whilst the test set was used as a test data for evaluating their performance. The resulting performance profile appeared to have an observable difference between the four models in terms of Root Mean Square Error (RMSE).

### 4.7.3 Prediction Results

Referring to Figure 4.5 and Figure 4.6 , the graphs represent the RMSE for each model for predicting the execution time for each of the Counting Triads, Degree Distribution and Detecting Clusters using SNAP and Gremlin as shown in the figures. It is clear that for both tools, the

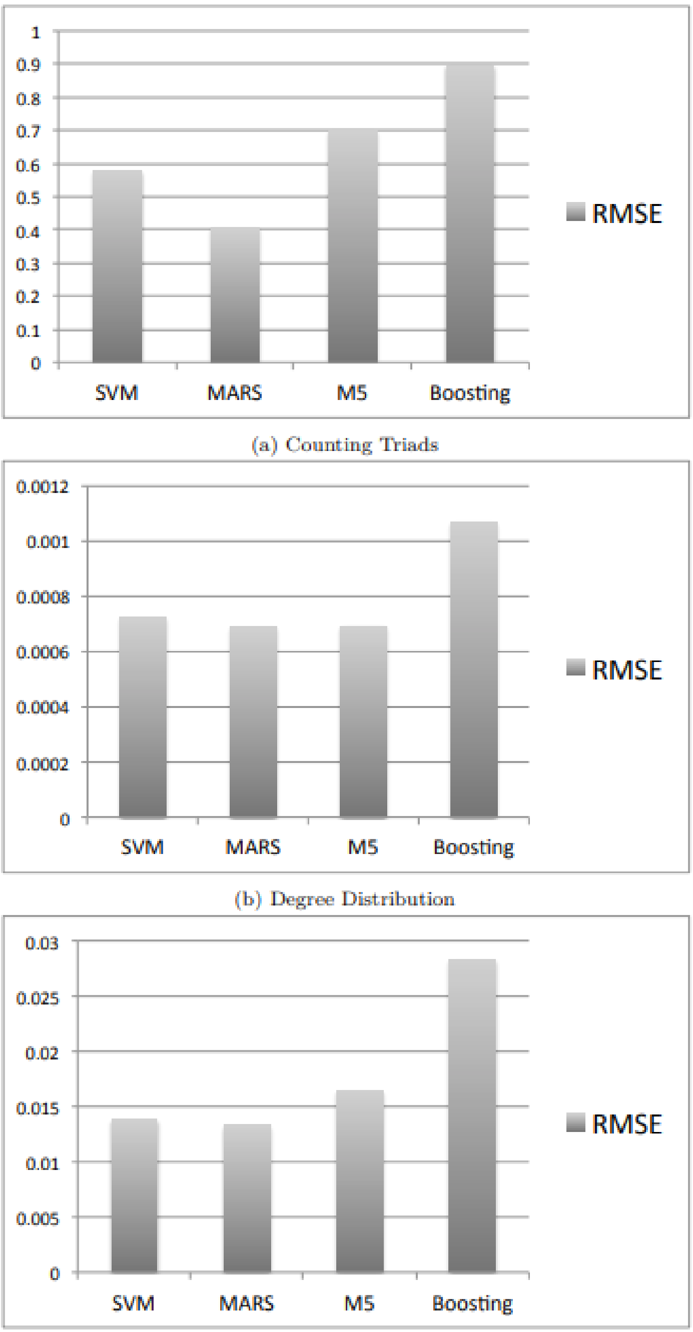


(a) Counting Triads

(b) Degree Distribution

(c) Detecting Clusters

Figure 4.5: RMSE for predicting the execution time (in seconds) for the three metrics using SNAP.



(a) Counting Triads

(b) Degree Distribution

(c) Detecting Clusters

Figure 4.6: RMSE for predicting the execution time (in seconds) for the three metrics using Gremlin.

Boosting model had the highest RMSE for all of the three metrics. On the other hand, we found that the best model with the minimum RMSE for both tools regarding the three graph metrics is MARS. For SVM and M5, the first outperformed the latter for *Triads Count* and *Detecting Clusters* while on the other side M5 outperformed for calculating the *Degree Distribution* for both SNAP and Gremlin. Therefore, our proposed prediction will be based on MARS model since it showed the best performance for all metrics using both tools. Hence, we derived our MARS based model for predicting the execution time based on the number of nodes and edges for measuring each metric illustrated by Equation 4.1 where the coefficients (a, b, c, d, and e) along with the hinge functions (h1, h2, h3, and h4) for each metric for both tools are defined in Table 4.4, Table 4.2 for SNAP and Table 4.3, Table 4.1 for Gremlin. We denoted by **ST**, **SD** and **SC** as the Counting Triads, Degree Distribution and Clusters respectively for SNAP. Similarly, **GT**, **GD** and **GC** for same metrics but for Gremlin.

$$\text{Time} = a + b \times h_1(x) + c \times h_2(x) + d \times h_3(x) + e \times h_4(x)$$

$$\text{where} \quad h_{1,2,3,4}(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \qquad (4.1)$$

| | *a* | *b* | *c* | *d* | *e* |
|---|---|---|---|---|---|
| **GT** | -1.05 | 0.009 | -0.012 | -0.002 | 0.002 |
| **GD** | 2.7 | -6.7 | 1.4 | 0 | 0 |
| **GC** | 0.05 | -0.00009 | 0.0005 | -0.0009 | 0.00008 |

Table 4.1: Coefficients of the MARS model for the execution time using Gremlin.

| | *a* | *b* | *c* | *d* | *e* |
|---|---|---|---|---|---|
| **ST** | $-1.5 * 10 - 4$ | $1.7 * 10 - 6$ | $-1.5 * 10 - 6$ | $-4.3 * 10 - 7$ | 3.8 |
| **SD** | $2.9 * 10 - 4$ | $-2.9 * 10 - 7$ | $2.1 * 10 - 7$ | $-7.5 * 10 - 8$ | $2.7 * 10 - 8$ |
| **SC** | 0.003 | -0.0000004 | 0.0000002 | 0 | 0 |

Table 4.2: Coefficients of the MARS model for the execution time using SNAP.

### 4.7.4 Summary

This chapter first propose a performance comparative analysis between social network analysis tools using a personal machine. Our results related to two different types of tools: Software and Query based tools, SNAP and Gremlin respectively. Gremlin tool showed lower performance

| | $h_1(x)$ | $h_2(x)$ | $h_3(x)$ | $h_4(x)$ |
|---|---|---|---|---|
| **GT** | h(500-N) | h(N-740) | h(1522-E) | h(E-1522) |
| **GD** | h(340-N) | h(N-340) | 0 | 0 |
| **GC** | h(340-N) | h(N-340) | h(N-820) | h(E-8620) |

Table 4.3: Hinge function coefficients of the MARS model for the execution time for Gremlin.

| | $h_1(x)$ | $h_2(x)$ | $h_3(x)$ | $h_4(x)$ |
|---|---|---|---|---|
| **ST** | h(500-N) | h(N-500) | h(1522-E) | h(E-1522) |
| **SD** | h(420-N) | h(N-420) | h(1522-E) | h(E-1522) |
| **SC** | h(7153-E) | h(E-7153) | 0 | 0 |

Table 4.4: Hinge function coefficients of the MARS model for the execution time for SNAP.

than SNAP, especially in calculating the number of triads in the graph. This suggests that using Gremlin should be accompanied by having a cluster to be able to parallelize many computations. On the other hand, SNAP performed efficiently on a personal machine and showed better results for all the metrics, so it can be useful for anyone who is comfortable with Python or C language. While SNAP can provide analytics on massive graphs it may lack features related to compatibility issues with graph databases and graph processors which are supported by Gremlin. Next, in order to provide the end user with a prediction tool for the execution time, we trained different machine learning models: SVM, MARS, M5 and Boosting on our test data to help take good decisions that facilitate timely analysis of the graphs. They report the execution time of different graph sizes using three graph metrics for SNAP and Gremlin. Our experiments concluded that MARS outperformed other models for both tools. Hence, we provided a computational formula based on MARS model for each graph metric for both tools to estimate the execution time needed to analyse a given graph.

# Part 3

# CONCLUSION

Chapter 5

# Conclusion and Future Directions

## Contents

In this chapter, we conclude the thesis by highlighting the main contributions of our work. We will also discuss the future work suggested for further improvements.

## 5.1 Summary of Contributions

The aim of this dissertation is to highlight some challenges of analysing social networks and achieve some progress towards a better understanding of these problems to provide solutions to better deal with them. We tried to achieve this by tackling some of these challenges and by proposing solutions, studies and methodologies to how to deal and overcome the impact of these challenges during mining a network.

Through the thesis, we have addressed some of these challenges. In chapter 4, an approach using machine learning was proposed for predicting the performance or the execution time to analyse a social network. This approach presents a way to deal with evolving and scale-free networks problem. The proposed approach provides an easy way to predict the approximate time taken to analyse a new network (or graph) given the number of nodes and edges within the network. A simulation for the evolving graph is achieved by extracting subgraphs of increasing in size from one network. Then, an analysis is held on some popular graph measures using two different tools like SNAP and Gremlin. Finally, we utilized four different machine learning regression models: MARS, Boosting, M5 and Support Vector Machine. The models were trained and tested over 50 samples of graphs having different sizes in order to select the best model using RMSE an evaluation metric. Our results concluded that MARS outperformed the other three models suggesting that it might be the best suited for addressing this problem. We provided multiple computational models with their coefficients for each graph measure in terms of nodes and edges.

For sure, this work can not cover all topics and the challenges related to social networks analysis. But in the last years, this area has got a lot of research attention. In the following section, we suggest some open problems and questions to be explored for further research.

## 5.2 Future work

The focus of the work and the main contributions in this thesis are in highlighting the challenges of social network mining and proposing research solutions to tackle these challenges. Our explorations have been focused on different social networks and tried to cover different challenges. The proposed approaches are potentially generic and could be applied on other types of networks. However, our research work in this thesis has the potential to be extended in the future work to include the following:

The proposed analysis and approach based on predictive modeling for measuring the performance in Chapter 4 are based on a single hardware with normal specs. However, we suggest that our analysis and our proposed approach can be applied on multiple hardware with higher specs, such as available CPU resources and RAM size that can achieve higher performance for evolving graphs. It is also interesting to explore the execution time of distributed graph systems that are based on memory approaches like Pregel and GraphLab which are proposed in the literature. Also, since Gremlin showed low performance on a personal machine and given that one of its advantages that it can be integrated with Hadoop (Gremlin/Hadoop) to allow parallel execution of Gremlin scripts as map reduce jobs on a Hadoop infrastructure. We believe that studying the impact on the computational time based on the structure of the network whether it is real-world network or random network can be an interesting path to explore.

## 5.3 Closing

We believe that the work presented in this thesis contributed to develop new techniques for dealing with some of the challenges and problems in the social network analysis area. The focus is providing new approaches that unlock some of the analytics problems in social networks. We believe that providing solutions for these challenges still needs more research attention and it is as important as paying attention to the area of mining the network itself. Through these contributions, we try to connect and combine different areas through our experiments such as studying most of the popular graph measures used in mining, comparing different analysis tools, applying predictive modeling techniques, studying different real world networks and deriving insights about them. We hope that the work in this thesis act as good start for researchers to continue working, investigating the challenges introduced at the beginning and trying to improve or extend the approaches presented in this thesis.

# Bibliography

[1] Giraph.

[2] Hadoop: An open-source framework for distributed processing of large datasets.

[3] Iris: An open-source datalog engine.

[4] Logicblox.

[5] Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. In *ACM Sigmod Record*, volume 29, pages 439–450. ACM, 2000.

[6] Lars Backstrom, Cynthia Dwork, and Jon Kleinberg. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *Proceedings of the 16th international conference on World Wide Web*, pages 181–190. ACM, 2007.

[7] Claude Berge. *The theory of graphs.* Courier Corporation, 2001.

[8] Anupam Biswas and Bhaskar Biswas. A framework for analyzing community detection algorithms. In *2016 IEEE Students' Technology Symposium (TechSym)*, pages 61–66. IEEE, 2016.

[9] Phillip Bonacich. Technique for analyzing overlapping memberships. *Sociological methodology*, 4:176–185, 1972.

[10] John Adrian Bondy, Uppaluri Siva Ramachandra Murty, et al. *Graph theory with applications*, volume 290. Citeseer, 1976.

[11] Ulrik Brandes. A faster algorithm for betweenness centrality. *Journal of mathematical sociology*, 25(2):163–177, 2001.

[12] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the web. *Computer networks*, 33(1-6):309–320, 2000.

[13] Ronald Burt. Structural holes: The social structure of competition. *Bibliovault OAI Repository, the University of Chicago Press*, 40, 05 1994.

[14] Deng Cai, Zheng Shao, Xiaofei He, Xifeng Yan, and Jiawei Han. Mining hidden community in heterogeneous social networks. In *Proceedings of the 3rd International Workshop on Link Discovery*, LinkKDD '05, pages 58–65, New York, NY, USA, 2005. ACM.

[15] Christopher S Campbell, Paul P Maglio, Alex Cozzi, and Byron Dom. Expertise identification using email communications. In *Proceedings of the twelfth international conference on Information and knowledge management*, pages 528–531. ACM, 2003.

[16] Peter J Carrington, John Scott, and Stanley Wasserman. *Models and methods in social network analysis*, volume 28. Cambridge university press, 2005.

[17] Meeyoung Cha, Hamed Haddadi, Fabricio Benevenuto, and Krishna P Gummadi. Measuring user influence in twitter: The million follower fallacy. In *fourth international AAAI conference on weblogs and social media*, 2010.

[18] Soumen Chakrabarti. *Mining the Web: Discovering Knowledge from HyperText Data*. Science & Technology Books, 2002.

[19] Aaron Clauset, Mark EJ Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004.

[20] Tyson Condie, David Chu, Joseph M Hellerstein, and Petros Maniatis. Evita raced: metacompilation for declarative networks. *Proceedings of the VLDB Endowment*, 1(1):1153–1165, 2008.

[21] Robert Lee Cross, Robert L Cross, and Andrew Parker. *The hidden power of social networks: Understanding how work really gets done in organizations*. Harvard Business Press, 2004.

[22] Jonathon N Cummings, Brian Butler, and Robert Kraut. The quality of online social relationships. *Communications of the ACM*, 45(7):103–108, 2002.

[23] P Deepak, Dinesh Garg, and V Varshney. Analysis of enron email threads and quantification of employee responsiveness. In *Workshop on Text Mining and Link Analysis (TextLink 2007)*, 2007.

[24] Narsingh Deo. *Graph theory with applications to engineering and computer science*. Courier Dover Publications, 2017.

[25] Mario Diani. Stanley wasserman e katherine faust, social network analysis: Methods and applications, cambridge, cambridge university press, 1994, pp. 825. *Italian Political Science Review/Rivista Italiana di Scienza Politica*, 25(3):582–584, 1995.

[26] Jana Diesner and Kathleen M Carley. Exploration of communication networks from the enron email corpus. In *SIAM International Conference on Data Mining: Workshop on Link Analysis, Counterterrorism and Security, Newport Beach, CA*, pages 3–14. Citeseer, 2005.

[27] Jana Diesner, Terrill L Frantz, and Kathleen M Carley. Communication networks from the enron email corpus "it's always about the people. enron is no different". *Computational & Mathematical Organization Theory*, 11(3):201–228, 2005.

[28] Sameh Elnikety and Yuxiong He. System support for managing large graphs in the cloud. In *Proceedings of the NSF Workshop on Social Networks and Mobility in the Cloud*. Citeseer, 2012.

[29] Heinz Eulau. Social structures: A network approach. edited by barry wellman and s. d. berkowitz. new york: Cambridge university press, 1988. 513p. *American Political Science Review*, 83(4):1404–1405, 1989.

[30] Leonhard Euler. Solutio problematis ad geometriam situs pertinentis. *Commentarii academiae scientiarum Petropolitanae*, pages 128–140, 1741.

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

[32] Santo Fortunato and Claudio Castellano. Community structure in graphs. *Computational Complexity: Theory, Techniques, and Applications*, pages 490–512, 2012.

[33] Linton C Freeman, Douglas Roeder, and Robert R Mulholland. Centrality in social networks: Ii. experimental results. *Social networks*, 2(2):119–141, 1979.

[34] Rudolf Fritsch, Rudolf Fritsch, G Fritsch, and Gerda Fritsch. *Four-Color Theorem*. Springer, 1998.

[35] Feng Fu, Xiaojie Chen, Lianghuan Liu, and Long Wang. Social dilemmas in an online social network: The structure and evolution of cooperation. *Physics Letters A*, 371(1):58 – 64, 2007.

[36] David Gibson, Jon Kleinberg, and Prabhakar Raghavan. Inferring web communities from link topology. In *Proceedings of the ninth ACM conference on Hypertext and hypermedia: links, objects, time and space—structure in hypermedia systems: links, objects, time and space—structure in hypermedia systems*, pages 225–234. Citeseer, 1998.

[37] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

[38] Peter A Gloor and Yan Zhao. Analyzing actors and their discussion topics by semantic social network analysis. In *Tenth International Conference on Information Visualisation (IV'06)*, pages 130–135. IEEE, 2006.

[39] Joseph E Gonzalez, Yucheng Low, Haijie Gu, Danny Bickson, and Carlos Guestrin. Powergraph: Distributed graph-parallel computation on natural graphs. In *Presented as part of the 10th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 12)*, pages 17–30, 2012.

[40] Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. Combating web spam with trustrank. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 576–587. VLDB Endowment, 2004.

[41] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[42] Frank E Harrell Jr. *Regression modeling strategies: with applications to linear models, logistic and ordinal regression, and survival analysis.* Springer, 2015.

[43] Caroline Haythornthwaite, Barry Wellman, and Marilyn Mantei. Work relationships and media use: A social network analysis. *Group Decision and Negotiation*, 4(3):193–211, 1995.

[44] J. Heer and D. Boyd. Vizster: visualizing online social networks. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, pages 32–39, Oct 2005.

[45] Michal Jacovi, Ido Guy, Shiri Kremer-Davidson, Sara Porat, and Netta Aizenbud-Reshef. The perception of others: inferring reputation from social media in the enterprise. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, pages 756–766. ACM, 2014.

[46] D Ullman Jeffrey. Principles of database and knowledge-base systems, 1989.

[47] Yingzi Jin, Yutaka Matsuo, and Mitsuru Ishizuka. Extracting social networks among various entities on the web. In *European Semantic Web Conference*, pages 251–266. Springer, 2007.

[48] Jon Kleinberg, Christos Papadimitriou, and Prabhakar Raghavan. Auditing boolean attributes. *Journal of Computer and System Sciences*, 66(1):244–253, 2003.

[49] Jon M Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

[50] Max Kuhn et al. Building predictive models in r using the caret package. *Journal of statistical software*, 28(5):1–26, 2008.

[51] Max Kuhn and Kjell Johnson. *Applied predictive modeling*, volume 26. Springer, 2013.

[52] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, pages 591–600. AcM, 2010.

[53] Aapo Kyrola, Guy Blelloch, and Carlos Guestrin. Graphchi: Large-scale graph computation on just a {PC}. In *Presented as part of the 10th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 12)*, pages 31–46, 2012.

[54] Edward Laumann, Peter V. Marsden, and David Prensky. The boundary specification problem in network analysis. *Applied Network Analysis: A Methodological Introduction*, 61, 01 1983.

[55] David Lazer, Alex Pentland, Lada Adamic, Sinan Aral, Albert-László Barabási, Devon Brewer, Nicholas Christakis, Noshir Contractor, James Fowler, Myron Gutmann, et al. Computational social science. *Science*, 323(5915):721–723, 2009.

[56] Elizabeth A Leicht and Mark EJ Newman. Community structure in directed networks. *Physical review letters*, 100(11):118703, 2008.

[57] T Lento, Howard Welser, L Gu, and Marc Smith. The ties that blog: Examining the relationship between social ties and continued participation in the wallop weblogging system. *3rd Annual Workshop on the Weblogging Ecosystem*, page 12, 01 2006.

[58] Jure Leskovec, Lars Backstrom, and Jon Kleinberg. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 497–506. ACM, 2009.

[59] Jure Leskovec, Lars Backstrom, Ravi Kumar, and Andrew Tomkins. Microscopic evolution of social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 462–470. ACM, 2008.

[60] 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*, pages 177–187. ACM, 2005.

[61] Jure Leskovec and Andrej Krevl. {SNAP Datasets}:{Stanford} large network dataset collection. 2015.

[62] 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.

[63] 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*, pages 631–640. ACM, 2010.

[64] Jure Leskovec and Rok Sosič. Snap: A general-purpose network analysis and graph-mining library. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(1):1, 2016.

[65] Jure Leskovec and Rok Sosič. Snap: A general-purpose network analysis and graph-mining library. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(1):1, 2016.

[66] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031, 2007.

[67] Yucheng Low, Danny Bickson, Joseph Gonzalez, Carlos Guestrin, Aapo Kyrola, and Joseph M Hellerstein. Distributed graphlab: a framework for machine learning and data mining in the cloud. *Proceedings of the VLDB Endowment*, 5(8):716–727, 2012.

[68] D Lusseau, K Schneider, OJ Boisseau, P Haase, E Slooten, and SM Dawson. An undirected social network of frequent associations between 62 dolphins in a community living off doubtful sound. *Behavioral Ecology & Sociobiology*, 54(4):396–405, 2003.

[69] Sammantha L Magsino. Applications of social network analysis for building community disaster resilience. In *Workshop Summary*, 2009.

[70] Grzegorz Malewicz, Matthew H Austern, Aart JC Bik, James C Dehnert, Ilan Horn, Naty Leiser, and Grzegorz Czajkowski. Pregel: a system for large-scale graph processing. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 135–146. ACM, 2010.

[71] Cameron Marlow. Audience, structure and authority in the weblog community. In *International Communication Association Conference*, volume 27, 2004.

[72] Yutaka Matsuo, Hironori Tomobe, and Takuichi Nishimura. Robust estimation of google counts for social network extraction. In *Proceedings of the 22Nd National Conference on Artificial Intelligence - Volume 2*, AAAI'07, pages 1395–1401. AAAI Press, 2007.

[73] Andrew McCallum, Xuerui Wang, and Andrés Corrada-Emmanuel. Topic and role discovery in social networks with experiments on enron and academic email. *Journal of Artificial Intelligence Research*, 30:249–272, 2007.

[74] Brett Meador. A survey of computer network topology and analysis examples. *Washington University*, 2008.

[75] Kurt Mehlhorn and Stefan Naher. Leda: a platform for combinatorial and geometric computing. *Communications of the ACM*, 38(1):96–103, 1995.

[76] Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[77] Mark EJ Newman. Fast algorithm for detecting community structure in networks. *Physical review E*, 69(6):066133, 2004.

[78] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

[79] MEJ NEWMAN. Network data from mark newman's home page.

[80] Wanda J Orlikowski and Jack J Baroudi. Studying information technology in organizations: Research approaches and assumptions. *Information systems research*, 2(1):1–28, 1991.

[81] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[82] Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *nature*, 435(7043):814, 2005.

[83] Yonathan Perez, Rok Sosič, Arijit Banerjee, Rohan Puttagunta, Martin Raison, Pararth Shah, and Jure Leskovec. Ringo: Interactive graph analytics on big-memory machines. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 1105–1110. ACM, 2015.

[84] Josep M Pujol, Ramon Sangüesa, and Jordi Delgado. Extracting reputation in multi agent systems by means of social network topology. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*, pages 467–474. ACM, 2002.

[85] William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.

[86] John W Raymond, Eleanor J Gardiner, and Peter Willett. Rascal: Calculation of graph similarity using maximum common edge subgraphs. *The Computer Journal*, 45(6):631–644, 2002.

[87] Marko A Rodriguez. The gremlin graph traversal machine and language (invited talk). In *Proceedings of the 15th Symposium on Database Programming Languages*, pages 1–10. ACM, 2015.

[88] Matthew Rowe, Miriam Fernandez, Harith Alani, Inbal Ronen, Conor Hayes, and Marcel Karnstedt. Behaviour analysis across different types of enterprise online communities. In *Proceedings of the 4th Annual ACM Web Science Conference*, pages 255–264. ACM, 2012.

[89] Ryan Rowe, German Creamer, Shlomo Hershkop, and Salvatore Stolfo. Automated social hierarchy detection through email network analysis. 2007.

[90] Satu Elisa Schaeffer. Graph clustering. *Computer science review*, 1(1):27–64, 2007.

[91] John Scott. *Social Network Analysis: A Hand Book (4nd ed.)",*. SAGE publications, 2017.

[92] John P. Scott and Peter J. Carrington. *The SAGE Handbook of Social Network Analysis*. Sage Publications Ltd., 2011.

[93] John R Seeley. The net of reciprocal influence. a problem in treating sociometric data. *Canadian Journal of Experimental Psychology*, 3:234, 1949.

[94] Jiwon Seo, Stephen Guo, and Monica S Lam. Socialite: Datalog extensions for efficient social network analysis. In *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, pages 278–289. IEEE, 2013.

[95] Jiwon Seo, Jongsoo Park, Jaeho Shin, and Monica S Lam. Distributed socialite: a datalog-based language for large-scale graph analysis. *Proceedings of the VLDB Endowment*, 6(14):1906–1917, 2013.

[96] Coach Drs R Smit. The influence of social network structure on the chance of success of open source software project communities.

[97] Gabor Szabo and Bernardo A Huberman. Predicting the popularity of online content. *Available at SSRN 1295610*, 2008.

[98] R Core Team et al. R: A language and environment for statistical computing. 2013.

[99] Manos Tsagkias, Wouter Weerkamp, and Maarten De Rijke. Predicting the volume of comments on online news stories. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1765–1768. ACM, 2009.

[100] S Van Kester. Efficient crawling of community structures in online social networks. 2011.

[101] Y. Wang and X. Li. Social network analysis of interaction in online learning communities. In *Seventh IEEE International Conference on Advanced Learning Technologies (ICALT 2007)*, pages 699–700, July 2007.

[102] Stanley Wasserman and Katherine Faust. *Social Network Analysis: Methods and Applications*. Structural Analysis in the Social Sciences. Cambridge University Press, 1994.

[103] Duncan J Watts. *Small worlds: the dynamics of networks between order and randomness*, volume 9. Princeton university press, 2004.

[104] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393(6684):440, 1998.

[105] Cong Yu. Beyond simple parallelism: Challenges for scalable complex analysis over social data. In *Proceedings of the NSF Workshop on Social Networks and Mobility in the Cloud*, 2012.

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

[107] Matei Zaharia, Mosharaf Chowdhury, Michael J Franklin, Scott Shenker, and Ion Stoica. Spark: Cluster computing with working sets. *HotCloud*, 10(10-10):95, 2010.