ANALYSIS OF COMPLEX DATA SETS USING MULTILAYER NETWORKS: A DECOUPLING-BASED FRAMEWORK

by

ABHISHEK SANTRA

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON
August 2020

To Maa and Baba who have made many sacrifices to make me who I am.

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude towards my supervisor, Dr. Sharma Chakravarthy for his constant guidance and support in the course of my doctoral journey. His undying passion and sincerity towards his work, the ability to adapt and being patient during difficult situations and analyzing a problem to the core before solving it are few life skills that will surely act as guiding forces all throughout my life. His philosophy of teaching the *process of learning*, rather than just focusing on solving a particular problem has had a deep impact on my overall outlook. I am highly thankful to him for the way he emotionally motivated and supported me, especially during the uncertain Spring and Summer semesters when the world experienced the pandemic.

Another source of inspiration in my doctoral journey is my co-advisor, Dr. Sanjukta Bhowmick. Through multiple insightful research meetings and presentations, paper and grant proposal writing sessions along with Dr. Sharma, I learnt the importance of critical discussions, the art of problem development, planning, team work and time management. I really admire her intense dedication towards research. I am immensely thankful to her for all the guidance and support. I really admire Dr. Sharma and her for their words of encouragement during the low points of my journey.

I am highly grateful to my esteemed committee members, Dr. Gautam Das, Dr. Ramez Elmasri and Mr. David Levine for their keen interest in my research, for taking time out and accepting to serve in my dissertation committee and for their invaluable feedback, suggestions, guidance and encouragement that have helped me in improving the quality of my work.

I will like to thank my graduate advisors, Dr. Bahram Khalili and Ginger Dickens who made my UTA journey a smooth experience through their immaculate advises. In addition, I would like to acknowledge the assistance I received, from Pamela McBride, Sherri Gotcher as well as the entire staff of the Computer Science and Engineering Department at UTA, during my stay at UTA.

From the initial set to the current one, I will like to extend my heartfelt gratitude towards each and every member of the Information Technology Laboratory (ITLab) for the exciting days and nights of endless whiteboard discussions, words of encouragement and support. My appreciation goes out to all my roommates and friends (especially, ICC Family, PhD Guys) for being with me during my ups and downs and making my six year stay at UTA a memorable one.

I am extremely grateful to all my teachers and mentors from Apeejay School, Hansraj College (University of Delhi) and Faculty of Mathematical Sciences (University of Delhi) for all the human values oriented teachings, support and motivation. I especially thank my Master's thesis advisor, Dr. Vasudha Bhatnagar whose sheer dedication inspired me to pursue research in the area of data mining. I will like to thank all my friends from school, Bachelor's degree (Go.8. Hansraj), Master's degree (Hell Raisers) and DU hostels who through their regular texts and calls have played an important role in my doctoral journey by not letting me feel away from home at any point of time.

Finally, I am extremely grateful to my parents, Dr. Sikha Santra and Dr. Manoranjan Santra for their constant love and belief, emotional support and motivation. A huge vote of thanks to my other family members for their endless love and support. Without their encouragement and endurance, this work would not have been possible.

July 24, 2020

ABSTRACT

ANALYSIS OF COMPLEX DATA SETS USING MULTILAYER NETWORKS: A DECOUPLING-BASED FRAMEWORK

Abhishek Santra, Ph.D.

The University of Texas at Arlington, 2020

Supervising Professor: Sharma Chakravarthy

We are on the cusp of analyzing a variety of data being collected in every walk of life - social, biological, health-care, corporate, climate, to name a few. The data sets are becoming diverse and complex in addition to increased size. Some of the complexity comes from interacting entities that arise in diverse disciplines, such as epidemiology [1], marketing strategy [2], social sciences [3], cybersecurity [4] and drug design [5].

Data sets becoming diverse and complex entails search for appropriate models and concomitant analytical techniques that are also efficient. Our ability to analyze large, complex, and disparate data for a broad set of analysis objectives differentiates big data analytics from mining which is narrow in scope both from data and analysis perspective. For big data analytics, flexibility of analysis (different from scalability) is important. Efficiency is important due to large number of analysis needs.

Elegantly modeling and efficiently analyzing these complex datasets to obtain actionable knowledge presents several challenges. Traditional approaches, such as using single graph (or a single layer network or monoplex) may not be sufficient or appropriate for modeling and computation flexibility. Recently, multilayer networks have been proposed as an alternative for modeling such data elegantly.

In this thesis, we first discuss different types of multilayer networks – homogeneous, heterogeneous and hybrid – from a modeling perspective. The benefits of this modeling, in terms of ease, understanding, and usage, are highlighted. Although big data analysis has warranted many new data models, not much attention has been paid to their modeling from requirements. Going straight from application requirements to data model and analysis, especially for complex data sets, is likely to be difficult, error prone, and not extensible to say the least. Hence for data models used in big data analysis, such as Multilayer Networks, there is a need to algorithmically transform the requirements using a systematic modeling approach, such as EER (Enhanced Entity Relationship). Here, we start with application requirements of complex data sets including analysis objectives and show how the EER approach can be leveraged for modeling given data to generate the MLN model and appropriate analysis expressions on them.

However, this model brings with it a new set of challenges – both algorithmically and efficiency-wise – for its analysis. Since there are not many algorithms available in the literature for the analysis of MLN as a whole, applying currently available techniques to a transformed version of MLN leads to loss of information in terms of structure and semantics. Our proposed approach is to develop an analysis framework without transforming the MLN model so structure and semantics can be easily preserved. The general framework proposed and developed in this thesis is termed network decoupling. This framework is intended to be beneficial to all aggregate computations although this thesis focuses on two of them. The essence of this approach is to analyze each network layer *individually* and then use a *composition function* for aggregating individual layer results. This thesis demonstrates the network decoupling approach and its merits for widely-used graph aggregation analysis, such as community and centrality. For both community and centrality detection of MLN using Boolean operators, efficient composition functions and algorithms have been developed and validated for Homogeneous Multilayer Networks. To demonstrate its effectiveness, this thesis has proposed a new community definition of heterogeneous MLNs using the same framework. This not only uses the decoupling approach based on bipartite graph matching, but also preserves structure and semantics. Structure and semantics preservation for MLNs (both homogeneous and heterogeneous) is crucial for drill down analysis to clearly understand and interpret results. Our definition supports a family of community detection algorithms for heterogeneous MLNs which is very useful for matching analysis objectives. Further, for a broader analysis, we introduce several weight metrics for bringing in individual layer community characteristics on the MLN community. Essentially, this results in an extensible *family of community computations*.

Finally, the framework and the algorithms proposed have been applied to real-world (Internet Movie Database - IMDb, Database Bibliography - DBLP, UK Accidents, US Airlines, Facebook) and synthetic data sets in order to validate the approach, flexibility afforded, accuracy limits, and efficiency aspects. Meticulous drill-down analysis on the final results has been carried out to come up with few surprising analysis results that predicted future potential events that we could verify by independently available ground truth. Based on this work, a dashboard for visualizing MLN analysis is underway.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Table Page

CHAPTER 1

MOTIVATION and PROBLEM STATEMENT

As data sets become more complex with diverse entity, feature, and relationship types, approaches needed for their modeling and analysis also require extensions and/or new alternatives to match the complexity of the data sets. We have already seen a surge in the use of graph-based modeling along with a plethora of relevant computations with the advent of social networks and large graph data sets. Even data sets that may not be inherently graph-based may benefit from the use of graph representation for modeling (from an understanding perspective) and for performing different kinds of analysis that may be difficult or not possible using the traditional Database Management System (or DBMS) or mining approaches. Data analytics requires a suite of diverse techniques to analyze different kinds of data sets and derive meaningful conclusions from them. Holistic/aggregate analysis relates to analyzing a multi-feature data set by including the effect of different combinations of features or perspectives. We have studied a variety of multi-entity, feature and relationship data sets for modeling and flexible aggregate analysis to derive a general modeling and an efficient analysis approach presented in this thesis.

In this thesis, given a set of *analysis objectives* for a data set comprising of *multiple entities, multiple feature and multiple relationships*, we focus on three pivotal questions (corresponding to the stages shown in Figure 1.1.)

- *How to model the data set?* (Stage 1)
- *How to efficiently perform the analysis?* (Stage 2) and,

- *How to drill down and visualize the results to infer actionable knowledge* (Stage 3)?

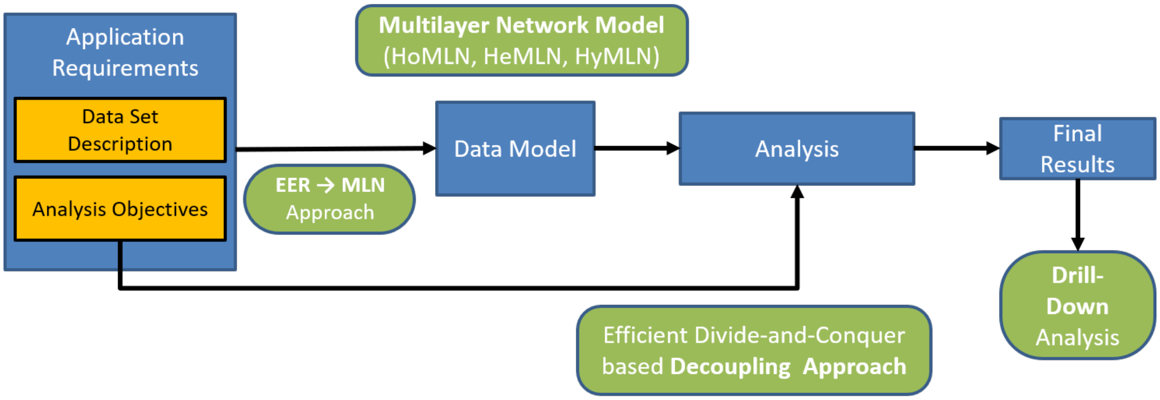


Figure 1.1: Different Stages of Complex Data Set Analysis

1.1 Modeling of Complex Data Sets

Data sets involving interactions/relationships among the entities have traditionally been modeled as simple single graphs or attributed graphs. Recently, multilayer networks has come out to be another viable alternative. In this thesis, we compare all the modeling alternatives with respect to modeling clarity, support for flexible analysis, computational efficiency and ease of drill-down analysis and visualization. Chapter 2 goes into the details of the modeling alternatives and the rationale behind proposing different types multilayer networks has been discussed. Moreover, in order to properly incorporate the application requirements in an error-free and systematic manner, algorithmic steps have been given to convert a Enhanced Entity Relationship (EER) model into the MLN model (5).

1.2 Analysis of Complex Data Sets

Multilayer networks have been used recently to analyze the multi-entity and multi-feature data sets. Most of the prominent analysis techniques like the type-independent and projection approach either collapse the entire MLN into a single graph leading to loss of information in terms of structure and semantics. While other techniques analyse the whole MLN as a single graph and perform random across the layers to obtain the desired analysis results. This technique entails developing new algorithms for different analysis objectives. In this thesis, we propose a novel *divide-and-conquer based network decoupling approach* where each layer is analyzed just once to obtain *partial results* that are combined using a composition function based on the analysis objective, thus allowing support for *flexible efficient analysis of MLNs.* The major challenge in this approach is to develop efficient and accurate composition function. Chapter 4 discusses the decoupling approach in detail focusing on the various requirements, benefits and challenges. In general, the decoupling-based composition algorithms based algorithms have been developed for community detection, centrality measurement and substructure discovery in MLNs, detailed discussion on which in terms of characteristics, efficiency and accuracy have been highlighted in different of the thesis.

1.3 Drill-Down Analysis and Visualization

It becomes imperative in every socio-economic domain to analyze the final analysis results further in order to obtain actionable knowledge to fulfill the business objective. For example, if the analysis results return a group of actors most of whom work who have co-acted in some movie (community), it must be possible to infer from this community result the set of genres that is most prevalent or a highly rated actor pair who have not yet worked together. Support for such actionable knowledge requires the preservation of structure and semantics in the final results in terms of node and edge labels (types) and initial edge connectivity. This is one of the major advantages of the proposed decoupling approach. In the thesis, we illustrate multiple times how decoupling approach has allowed us to drill down and visualize the final communities, hubs and frequent substructures to infer hidden interesting information and predict potential interactions through real-world data sets like IMDb, Facebook, DBLP and Accidents. Information for independent sources have been used to validate the drill down results.

1.4 Contributions of the Thesis and Roadmap

This thesis explores the different stages involved in the analysis of complex data sets starting the application requirements phase to the final analysis results followed by drill-down analysis. The related work from different domains related to this thesis has been discussed in Chapter 3. The contributions of this thesis (along with the roadmap) have been listed below;

1. In Chapter 2 we discuss in detail the alternatives we have for modeling the complex data sets and provide a rationale behind proposing **different types of multilayer networks**.
2. Modeling directly from the application requirements to a data model is likely to be difficult and error-prone. EER has played a central role in the modeling of user-level requirements to relational, object oriented etc. However, currently, there is no modeling approach when it comes to complex and diverse data sets. In Chapter 5, we **propose the algorithmic steps to convert an EER model (built from the application requirements) to the Multilayer Network model**.

3. In Chapter 4, we **propose the generic decoupling approach framework for efficiently analyzing MLNs**. Efficient composition algorithms (Θ) for different analysis functions (Ψ) like community and centrality detection have been proposed as a part of this thesis.
4. In Chapter 6, we have proposed the **Boolean operator based composition functions for the community detection in Homogeneous Multilayer Networks (HoMLN)**. The challenges involved in developing the composition function, cost analysis of proposed algorithms, accuracy inferences based on different graph characteristics and efficiency has been discussed in detail. We have provided case studies in Chapter 7 and 8, where these proposed composition algorithms are used to perform an aggregate analysis of the Facebook and Movie Actor data sets, respectively.
5. **Efficient Boolean composition-based heuristics to detect the degree and closeness centrality based hubs have been proposed** in Chapter 9. Similar to community detection, detailed cost analysis, graph characteristics based accuracy inferences and efficiency discussions have been provided. These heuristics have been used for a case study on the US Commercial Airline data set in chapter 10, where centrality-based analysis has been performed.
6. The notion of community is well defined for single graphs. However, in case of **heterogeneous multilayer networks there is no proper definition of a community**. In chapter 11, we provide a **structure and semantic preserving definition of a HeMLN community** and **propose efficient composition techniques based on weighted maximum matching** to detect them.
7. In every chapter, different types of MLNs have been used from **Facebook, IMDb (Internet Movie Database), UK Accident, US Airline, DBLP**

(Database Bibliography or Computer Publications) to illustrate the proposed modeling and composition algorithms.

8. Extensive experiments have been performed on real-world and synthetic data sets in order to **validate the accuracy and efficiency aspects**.
9. Upon obtaining the final analysis results (community, centrality hubs), we performed **drill-down analysis and visualization in order to find out hidden interesting knowledge and made link-/node-/group-based predictions**, validated from independent sources wherever possible.
10. This thesis has led to various publications:
    - **Published**: [6, 7, 8, 9, 10, 11, 12, 13, 14]
    - **Under Review**: [15, 16]

CHAPTER 2

COMPLEX DATA SET MODELING: USE OF MULTILAYER NETWORKS

In order to discuss the different types application requirements with respect to multi-entity, feature and relationship data set, *three illustrative scenarios* have been discussed in Section 2.1 that have been used to introduce the *chosen* multilayer network data model and its different categories in Section 2.3 due to the limitations of traditional modeling techniques 2.2.

2.1 Categories of Analysis Scenarios

Scenario 1: Consider the problem of modeling and analyzing the **traffic accident** problem (or data set) for a region or a country. A number of features are associated (and collected) with each accident such as location, speed, time of the day, severity of the accident, light, weather, and road conditions. One may want to analyze this data set from multiple angles like

- General accident prone regions?
- Dominant feature associated with most accidents?
- Ordering features based on their effect on the severity of the accident?
- Effect of individual or combination of features on accidents in a region or across all regions?

Scenario 2: Consider another data set where we have information about **scientists** who collaborate with each other, **cities** that have direct flights, and **conferences** that have overlapping research topics. In addition, there is information about who lives in what city and the cities in which annual conferences have been held. Given this data set, it would be useful to understand,

- Where do the most collaborative group of scientists reside?
- Whether a large group of collaborators have attended several conferences?
- Who are the most popular collaborators for different research topics? Can we group cities where work belonging to these research areas is going on?

Scenario 3: If the Facebook friendship information of scientists, carrier information for the flights between cities and attendance information of conference is also available in addition to the data described in Scenario 2, then the extended set of analysis may involve;

- Who are the highly collaborative and socially popular (influential) scientists who have attended conferences with large attendance?
- Which is the best city to hold a workshop on a particular topic to get maximum number of collaborating scientists?
- For the set of cities that are serviced by maximum number of carriers, who are the residing scientists that have maximum friends?

Note that, unlike the problem in Scenario 1, where the analysis referred to the same entity set (accidents), in Scenario 2 different disjoint entity sets are used (scientists, cities, and conferences). In Scenario 3, it is a combination of scenarios 1 and 2, where apart from multiple entities, multiple features or relationships are captured for every entity type (e.g., twitter followers and collaboration for scientists, different flight carriers for cities, overlapping research areas and attendance for conferences). Therefore, there is a need for a data model that can handle these analysis that span multiple entities and their relationships in different ways.

2.2 Traditional Modeling Techniques

Research on graph-based modeling and analysis have been around for a long time. In addition to analysis, search and querying of graphs are also becoming mainstream [17, 18]. As the size of the data sets increases and their characteristics become more complex, efficient and scalable approaches [19] are being developed to cope with the increase in data size and complexity using partitioning and parallel processing techniques. Partitioned approaches with loss-less computation and fast approximate algorithms are being investigated [20].

Single Graph or Simple Graph or Monoplex: Here the data set is represented by a single network or graph. It models entities as nodes and features as edges of the graph. This model gives rise to graphs with single node types (entity types are not distinguished even when multiple node types are modeled using this approach) and single edges between nodes (either for a single feature or for a combination of features). Relationships among the entities can either be specified by explicit interactions (like flights, co-authors and friends) or based on a similarity metric depending on the type of the feature like nominal, numeric, time, date, latitude-longitude values, text, audio, video or image.

Advantages. This approach is the most popular representation as large number of computations exist for simple graphs. There are several algorithms for analyzing simple graphs, such as detecting cliques, communities, centrality metrics, mining subgraphs, motifs etc.

Disadvantages. Single networks are, however, not adequate in representing *multiple features.* Particularly, it is difficult to combine features of different categories (e.g., numerical and categorical), in a meaningful way as one edge. The problem compounds when the entities are also of different types as well. Moreover, when analyzing a subset of entities and/or associated feature types, separate graphs may have to be created for each such combination and analysis.

Attribute or Knowledge Graphs: Here additional features of the data sets can be represented by including node types in terms of labels (even multiple labels) and multiple edges, even self-loops, corresponding to relationships for different features.

Advantages. Attribute graphs have been successfully used in subgraph mining [19, 21, 22], querying [23, 24, 25] and searching [26, 27] over multi-entity types and multi-feature datasets. They capture more semantic information than simple graphs, and can handle both multiple types of features and entities.

Disadvantages. Algorithms for some key analysis functions, such as community and centrality detection are not yet available for general attribute graphs. Hence, these graphs need to be converted to a monoplex for analysis. Although different features can be stored in the graph, for every subset of features, the analysis has to be done separately.

Due to the modeling, analysis and computational inefficiency of traditional approaches, we propose to use multilayer networks (multiple layers of interconnected graphs) as an alternative model.

2.3 Modeling of Complex Data Sets using Multilayer Networks

Multilayer Networks (or MLNs) are *layers of single graphs (or monoplexes)* or *network of networks*[*]. Each layer, typically, captures the semantics of one particular feature. As in a monoplex, the graph vertices represent the entities of the data set and the edges represent similarity between the feature values or the dyadic relationship

[*] The terminology used for variants of multilayer networks varies drastically in the literature and is not even consistent with one another. Please refer to [28] which provides a comprehensive comparison of terminology used in the literature, and their differences clearly.

between the end point vertices. The vertices of two layers can also be connected. To differentiate, we term the edges within a layer as *intra-layer edges* and the edges across the layers as *inter-layer edges*.

Formally, a **multilayer network**, $MLN(G, X)$, is defined by two sets of graphs: i) The set $G = \{G_1, G_2, \ldots, G_N\}$ contains graphs of N individual layers, where $G_i(V_i, E_i)$ is defined by a set of vertices, V_i and a set of edges, E_i. An edge $e(v, u) \in E_i$, connects vertices v and u, where $v, u \in V_i$ and ii) A set $X = \{X_{1,2}, X_{1,3}, \ldots, X_{N-1,N}\}$ consists of bipartite graphs. Each graph $X_{i,j}(V_i, V_j, L_{i,j})$ is defined by two sets of vertices V_i and V_j, and a set of edges (also called links or inter-layer edges) $L_{i,j}$, such that for every link $l(a, b) \in L_{i,j}$, $a \in V_i$ and $b \in V_j$, where V_i (V_j) is the vertex set of graph G_i (G_j.)

Based on the type of relationships and entities, multilayer network are of different types. If each layer of a MLN has the **same set of entities of the same type**, it is termed a Homogeneous MLN (or HoMLN.) Thus, $V_1 = V_2 = \ldots = V_n$. For a HoMLN, intra-layer edges are shown explicitly and inter-layer edges are not shown, as they are implicit. Scenario 1 can be modeled using HoMLN, where the accident instances become nodes in each layer. The similarity between them with respect to the different features like light, weather and road conditions become the basis of the intra-layer edges in their respective layers. Figure 5.4 (a) shows the Accident HoMLN.

When the **set and types of entities are different across layers**, then the MLN is termed as a **heterogeneous multilayer network (HeMLN)**. Scenario 2 can be modeled as a heterogeneous multilayer network, shown in Figure 5.4 (b). Each layer has a different entity type as its nodes (e.g., scientists, cities, and conferences). The graph of a layer is defined with respect to the chosen features and entity types. That is, scientists nodes are connected if they co-authored a paper, conference nodes

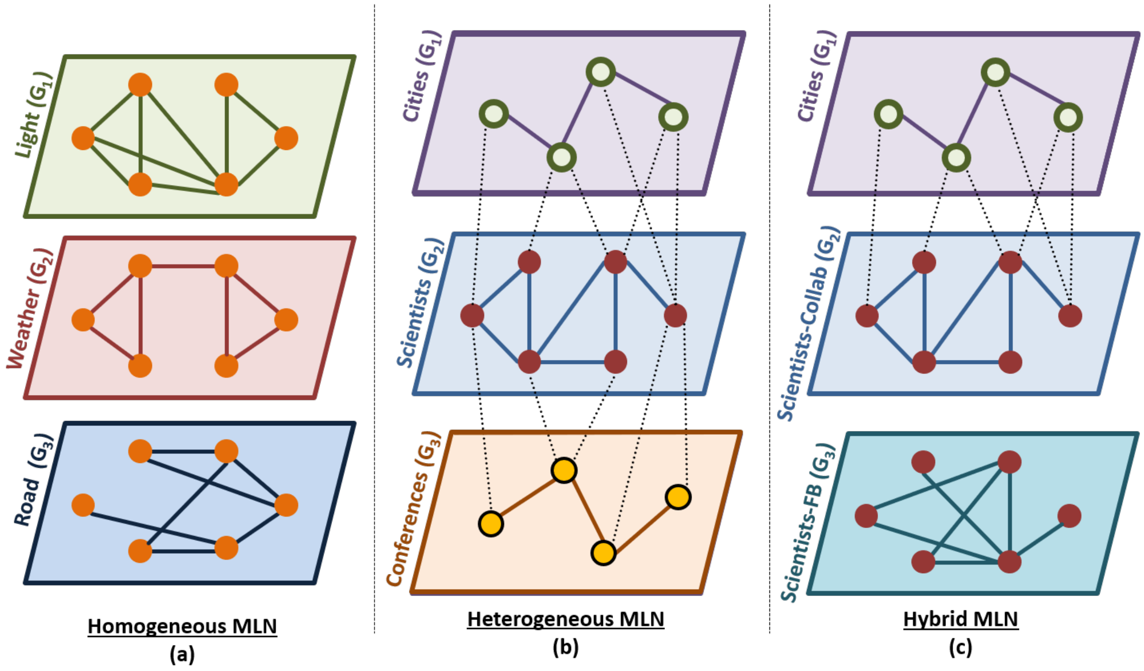


Figure 2.1: Types of Multilayer Networks

are connected if they have similar/overlapping research domains) and city nodes are connected if there is direct flight between them.

In this case of HeMLNs, the inter-layer links are defined explicitly based on feature semantics that corresponds to an edge (e.g., conference-held in, scientist-resides in and attends-conference).

In Scenario 3, the scientists, cities and conferences had multiple features associated with them like in addition to being collaborators, researchers may be Facebook friends. Thus, to model multi-feature data that capture **multiple relationships within and across different types of entity sets**, a combination of homogeneous and heterogeneous MLNs is used, called **hybrid MLN (or HyMLN)**, shown in Figure 5.4 (c).

Advantages. Compared to the other options, multilayer networks are a more natural and elegant choice for modeling data sets with multiple entities, features, and relationships. In MLNs each *chosen* feature (or combination) is modeled in a separate layer and thus this model can support both heterogeneous and homogeneous data sets. MLNs are also better suited from an information representation (i.e., *structure and semantics*) viewpoint and its visualization. Instead of cluttering all the entities and relationships in a single graph (or layer), they are logically separated and hence are easy to understand. The intra- and inter-layer relationships are also separated semantically. Each incremental change to each feature or relationship, as modeled by addition/deletion of vertices and edges can be easily included without extensive re-modeling of the already created MLN. Unlike most currently used approaches there is no need to convert a MLN representation to another one (simple or attributed) for analysis when the decoupling approach, discussed in Section 4.2, is used.

Challenges: Having argued for MLN for modeling, the primary challenge is to preserve the MLN structure during analysis (without collapsing them as is done by current approaches), thereby preserving both structure and semantics. If this can be done, further drill-down of data can be easily accomplished to uncover hidden interesting knowledge or make future link-based/node-based/group-based predictions. Once the structure and semantics are preserved, visualization of the results is straightforward. Another important challenge is to perform analysis efficiently and keep the MLN structure intact. Chapter 4 discusses in detail the proposed *network decoupling approach* that addresses all these challenges.

CHAPTER 3

RELATED WORK

We present the recent work relevant to the areas that we have explored as a part of this thesis - multilayer network analysis, EER modeling and community and centrality detection in MLNs.

Recently, many analytical tasks have used multilayer networks to handle varying interactions among the same or different sets of entities prevalent in complex data sets like co-authorship network in different conferences [29], citation network across different topics [30], interaction network based on calls/bluetooth scans [31] and friendship network across different social media platforms [32]. Multilayer networks have been used in many other diverse applications including improving drug design [5], understanding collaboration patterns [33], comparing the evolution of species [34], finding vulnerabilities in power grids [35], and identifying illegal activities via social interactions [36]. Review of current work on multilayer networks are given in [28, 37, 38]. Some software have been developed by groups at Europe, including Muxviz [39, 40], MAMMULT [41, 42] and Pymnet [43]. However, they focus more on visualization and support only a few analysis functions. It is not easy to add or compare algorithms on these platforms nor can they leverage parallelism as we propose here.

Most of this work focuses on *overall MLN diagnostics* by considering the MLN layers *individually*. However, in order to holistically study entities and relationships of multilayer networks, we also have to study the combinations of different layers in the network. Although, techniques based on information theory have been proposed for multilayer protein-protein interactions [44], this is only for reducing the number of redundant layers through aggregation. We need a principled approach to arbitrarily combine features without having to construct combined layers and analyze them.

ER and EER models have served as a tool for database design by incorporating the important semantic information about the real world [45]. The area of relational database modeling especially benefited from this body of work. A good EER diagram based on the user analysis requirements is critical for an error-free relational database schema. Numerous tools have been developed for creating the EER diagram and algorithmically mapping it into relations for different commercial DBMSs.

However, with the emergence of structured data sets with inherent relationships among entities and complex application requirements, such as shortest paths, important neighborhoods, dominant nodes (or groups of nodes), etc, [23, 46], the relational data model was not the best choice for modeling as well as analyzing them [47]. This led to the evolution of NoSQL data models including the graph data model [48]. In many cases, like friendship (Facebook), collaborations (Movies) and follower-followee (Twitter) relationships, relationships needed to be modeled explicitly using graph data structures. This gave rise to computations over these data models. Recently, there has been some work in the area of graph modeling from EER diagrams, but is limited to simple attributed graphs only [49, 45, 50, 51]. However, most of these works either do not handle recursive relationships ([49]), and weak entities [52] or are application-specific [53]. Moreover, to the best of our knowledge *there is not any work that establishes the set of rules for the generation of a multilayer network given a set of analysis objectives and data set description.*

Community detection on a simple graph involves identifying groups of vertices that are more connected to each other than to other vertices in the network/graph. Most of the work in the literature considers **single networks or sim-**

ple graphs where this objective is translated to optimizing network parameters such as modularity [54] or conductance [55]. As the combinatorial optimization of community detection is NP-complete [56], a large number of competitive approximation algorithms have been developed (see reviews in [57, 58].) Algorithms for community detection have been developed for different types of input graphs including directed [59, 60] edge-weighted [61], and dynamic networks [62, 63]. Recently there have also been algorithms for identifying overlapping communities [64, 65]. However, to the best of our knowledge, there is no community definition and detection that include node and edge labels, node weights as well as graphs with self-loops and multiple edges between nodes. Even the most popular community detection packages such as Infomap [66] or Louvain [67], do not accept non-simple graphs. In contrast, subgraph mining [21, 22, 19], querying [23, 25], and search [26, 27] have used graphs with node and/or edge labels including multiple edges between nodes, cycles, and self-loops.

Recently, **community detection algorithms have been extended to Homogeneous MLNs** (see reviews [38, 68].) Algorithms based on matrix factorization [69], cluster expansion philosophy [70], Bayesian probabilistic models [71], regression [72] and spectral optimization of the modularity function based on the supra-adjacency representation [73] and a significance based score that quantifies the connectivity of an observed vertex-layer set through comparison with a fixed degree random graph model [74] have been developed. However, all these approaches *analyze a MLN either by aggregating all (or a subset of) layers of a HoMLN using Boolean and other operators or by considering the entire MLN as a whole*, leading to loss of information and computational inefficiency.

To the best of our knowledge, there is **no community definition or detection algorithm for Heterogeneous MLNs**. Majority of the work on analyzing

HeMLN (reviewed in [75, 76]) focuses on developing meta-path based techniques for determining the similarity of objects [77], classification of objects [78], predicting the missing links [79], ranking/co-ranking [80] and recommendations [81]. The type-independent [44] and projection-based [82] approaches used for HeMLNs *neither preserve the structure nor the semantics of the community.* The type independent approach collapses all layers into a simple graph keeping *all* nodes and edges (including inter-layer edges) sans their types and labels. The same is true for the projection-based approach as well, that projects the nodes of one layer onto another layer and uses the layer neighbor and inter-layer edges to collapse the two layers into a single graph with a single entity type instead of two. The presence of different sets of entities in each layer and the presence of intra-layer edges makes structure-preserving definition more challenging for HeMLNs and also warrants a novel composition technique. A few existing works have proposed techniques for generating clusters of entities [83], but they have only considered the inter-layer links and not the networks themselves.

Degree centrality [84] and **closeness centrality** [85, 86] have been used in monoplex (single layer network) to detect high centrality nodes. There has been work in determining centrality measures by aggregating all the layers of a multilayer network [87] or performing walks across layers [88]. However, the problem of *inferring the degree centrality or closeness centrality hubs of any arbitrary conjunctively combined network from hubs of individual layers, in a cost-effective manner, has not been addressed earlier.*

CHAPTER 4

NETWORK DECOUPLING-BASED FRAMEWORK FOR MLN ANALYSIS

In this chapter, we propose a novel divide and conquer based network decoupling approach towards MLN analysis due to the undesirable drawbacks of the current analysis approaches.

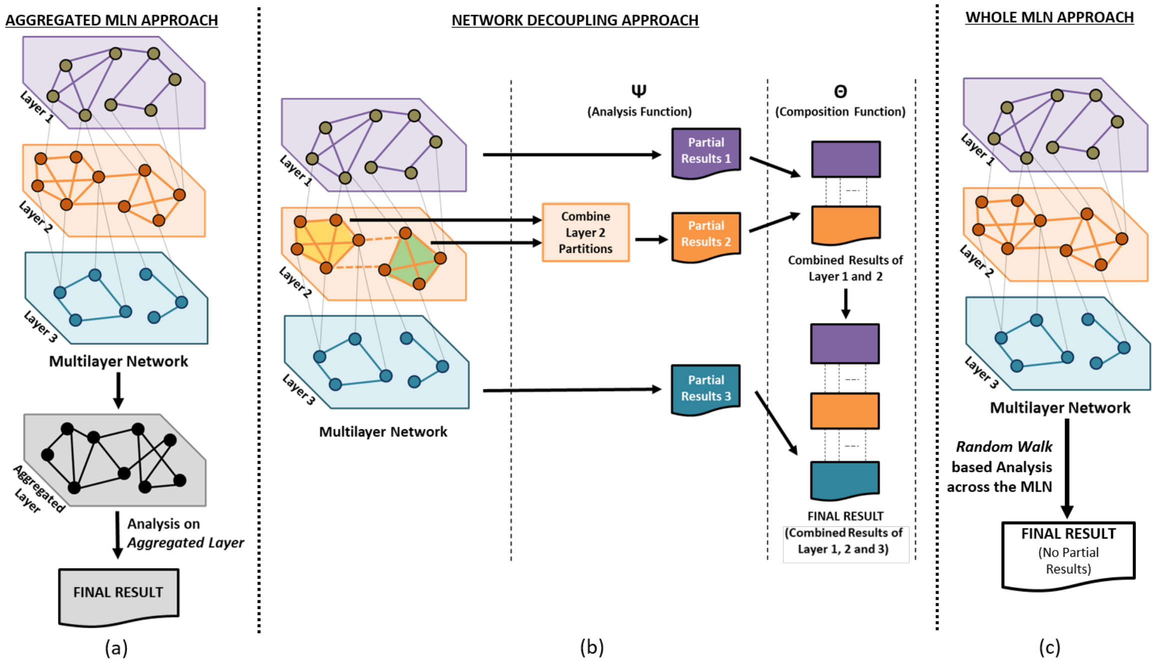


Figure 4.1: Approaches to Analyze MLNs

4.1 Current Approaches to Analyze MLNs

Current Approaches to Analyze MLNs are to map the networks to an equivalent single graph in various ways [37, 28]. However, through this process, many of the information in the multilayer graphs can be lost. There are mainly two approaches for converting a MLN into a single layer network. The first, used for homogenenous MLNs, is to *aggregate the edges of the multilayer network*. Specifically, given two vertices v and u, the edges between them from each layer are aggregated to form a single aggregated edge. This process is repeated for all the vertex pairs. Some typical aggregation functions are Boolean AND (intersection), OR (union) or linear functions when the edges are weighted. An example, from homogeneous MLNs, would be aggregating routes of different airplane carriers [89].

For heterogeneous MLNs, aggregation is performed in many ways. The first is *type independent* [44], that is ignore the varying types of the entities, and thus basically treat it as a homogenenous MLN with a subset of vertices in each layer. The second method is *projection-based* [82]. Here, if two vertices in a layer are connected to a common vertex in another layer, then an edge is inferred between them. Such "projections" of one layer onto another layer produce inferred edges and then these edges are aggregated. An example is connecting drugs that act on common proteins [82].

Another method, used for HeMLNs, is to *transform the multilayer network into an attribute graph*, where the vertices and edges are labeled based on their types. This *MLN-as-a-whole* graph is analyzed to find specified subgraphs, such as patterns of authors, papers and venues [76] or vulnerabilities in infrastructure networks [90].

Issues. Single network approach has the advantage that many analysis algorithms for community and hub detection are available (e.g., Infomap [66], Louvain [67] being prominent ones for community detection). However, the aggregation approaches preserve neither structure nor semantics of MLNs as they aggregate layers. Importantly, aggregation approaches are likely to result in some information loss or distortion of properties [28] or hide the effect of different entity types and/or different intra- or inter-layer relationship combinations [91]. In cases, where the multilayer network is converted to an attribute graph, algorithms for aggregate computations (e.g., community, hub) does not exist. Some approaches use the multilayer network as a whole [92] and use inter-layer edges, but do not preserve the layer semantics completely. An alternative is to separate desired subgraphs and use single network algorithms which defeats the purpose of modeling as attribute graphs and is inefficient.

4.2 Decoupling Approach to Analyse MLNs

We propose **Network decoupling** as a method by which MLNs can be analyzed *without being transformed* to another form. The decoupling approach preserves the structure and semantics of the layers in the result and at the same time can take advantage of the existing algorithms. The *network decoupling* approach is the equivalent of "divide and conquer" for MLNs. This is illustrated in Figure 4.1(b) and is applied as follows, for a given analysis function Ψ and composition function, Θ:

(i) Use the analysis function Ψ to analyze each layer individually like community, centrality metrics, frequent subgraphs, motifs, graph querying etc.

(ii) Second, for any two chosen layers, apply a *composition function* Θ to compose the partial results from each layer to generate intermediate results.

(iii) Finally, apply the composition process until the expression is computed.

This is in contrast to current approaches described earlier. Figure 4.1(a) indicates aggregation-based approaches where structure and semantics are lost. Figure 4.1(c) illustrates MLN approaches where only inter-layer edges are used instead of all edges.

Advantages: The decoupling approach has advantages over the traditional methods.

- By using the aggregation approach, information pertaining to the individual layers is lost and it is difficult to measure their relative importance to the system as a whole. In contrast, network decoupling retains the semantic information of each layer and therefore their individual importance and contribution can be measured.

- Plethora of efficient single graph analysis algorithms exist. They can be leveraged in order to generate the layer-wise results.
- The "divide and conquer" approach also facilitates the mix and match of the features and relationships, thus supporting flexibility in terms of analysis.
- In the aggregation approach, each time a subset of features is selected, the analysis has to be recomputed, even when the subsets might have overlaps. This leads to redundant computations. Using the decoupling approach, redundant analysis are avoided, since each layer, corresponding to a particular feature is analyzed separately, and then combined.
- Another important advantage is that the decoupling approach is amenable optimizations. That is, if the composition function is commutative and associative then a given specification can be optimized (by re-arranging the order of layer compositions) to obtain an alternative more efficient specification producing the same result.
- There are ample parallelization opportunities using this approach right from parallel generation of layer-wise results to parallel combination of independent partial results, which was not possible in with the existing MLN analysis techniques.
- Most importantly, this approach is application independent.

Challenges. The decoupling approach can be applied for both HoMLN and HeMLN. Moreover, the success of this approach is dependent on correctly matching the analysis function, Ψ, with composition function, Θ, that should produce accurate results and be computationally efficient.

In this thesis, we have developed Boolean composition algorithms for community and centrality detection in HoMLNs and weighted maximum matching composition for community detection in HeMLNs. The decoupling approach has also been extended for frequent substructure discovery in HoMLNs [93].

However, before discussing the various types of MLN analysis, it is important to understand that going directly from the application requirements to the MLN model, is bound to be error-prone and difficult. Thus, in the next chapter we first talk about the algorithmic steps that we have proposed to convert an EER (Enhanced Entity Relationship) model to the MLN model.

CHAPTER 5

EER→MLN: EER APPROACH FOR MODELING COMPLEX DATA USING MULTILAYER NETWORKS

Big data analytics is predicated upon our ability to model and analyze disparate, complex data sets. RDBMSs have served well for modeling and analyzing data sets that need to be managed over a long period of time and that are suited for relational representation. Data warehouses and OLAP came about to improve the analysis aspect of RDBMSs using more powerful queries (to provide multi-dimensional analysis) that could not be done earlier. This evolution has continued with NoSQL systems providing alternate data models and analysis for data that were difficult (or inefficient) to model using RDBMSs. We see the applicability of Multilayer Networks (or MLNs), its modeling, and analysis as another important step in the evolution of aggregate analysis of complex data sets.

In this chapter, our focus is on data sets with diverse types of entities that are defined by multiple features and interact through varied and complex relationships. Although graph modeling is used, the analysis and computations are different from the ones addressed in either RDBMSs or recent NoSQL systems, such as Neo4J. Instead of a database, the data is transformed into MLN data structures using EER modeling and computations are performed on these using packages and libraries that are available. Just to give an idea, an analysis may need community detection, degree-centrality (or hubs) detection and combine layers using Boolean operator (AND, OR, and NOT) or use weighted bipartite graph matching, details of which have been discussed in subsequent chapters.

Although EER modeling is widely used for relational and object-oriented data modeling, there is no modeling approach when it comes to complex, diverse data sets. This is likely to create problems for analysis and representation of data sets correctly to match analysis objectives. We will exemplify this with user requirements below.

5.1 Data Set Descriptions and Analysis Objectives

We have chosen *three data sets for analysis* from different application domains to illustrate the general applicability of our proposed framework. While much larger data sets can be used, we selected these because reliable ground truth data from orthogonal sources were available. Although we have indicated many analysis objectives to show the scope of this approach, due to space constraints, we show only a subset of them *in the experimental analysis section.* However, all of them have been computed.

1. Internet Movie Database (IMDb): This data set is publicly available and stores information about movies, TV episodes, actor, directors, ratings and genres of the movies, etc. [94]. Here the entities are of different types as they can be actors, directors, movies, etc. The features/relationships can be co-actors, similar-genre-acting, directed-a-movie, same movie ratings etc.

Analysis Objectives. Analysis requirements on this data sets can be diverse. As sample examples, one may want to analyse *actor-based relationships*:

(A1) Find co-actor groups that are *most popular* and *most versatile*

(A2) *Cluster* groups of co-actors who have worked in movies with high ratings

(A3) *Predict* new groups of actors who have not worked together before, but are likely to work together in future

2. Database Bibliography (DBLP): As most researchers are familiar with, the DBLP dataset is publicly available and stores information about computer science publications in various conferences and journals. It captures the author names and institutions, years, conference/journal names and links to the papers [95]. Clearly, there are multiple entities that can be related based of different types of relationships.

Analysis Objectives. Again, our aim is to be able to perform analysis, such as:

(A4) Find *strongest* co-author groups who have collaborated on at least 3 papers

(A5) For each conference, find *most popular groups* of co-authors who publish frequently

(A6) For the *most popular* collaborators in each conference, find the 3-year period(s) when they were *most active*

(A7) For each conference that publishes maximum papers in each period, find the *most popular paper review score.*

3. Author-City Data Sets: Airline data set contains the flights between different cities. This information can be combined with the author information from the DBLP data set to indicate who lives in which city. It can also be used for actors and directors.

Analysis Objectives. For such a diverse data set, the analysis objectives are also quite involved. For example,

(A8) Find *strong* co-author groups who are also friends (if Facebook information is available)

(A9) Find cities where the largest concentrations of authors reside

(A10) What is a good city to hold conferences of authors to maximize attendance?

We have selected the analysis objectives to be varied for the purposes of illustrating the need and effectiveness of the approach being proposed. They range from relatively easy analysis of finding clusters of co-actors to more complicated predictions of future teaming of actors and potential city for holding a conference.

Problem Statement. *For a given dataset with $\mathcal{F}$ features and $\mathcal{T}$ entity types and a set of analysis objectives ($\mathcal{O}$), develop: (i) an EER diagram for modeling the data set in conjunction with appication requirements, (ii) develop an algorithm to convert* *the EER diagram into the data model (MLNs in this case), (iii) map the analysis objectives ($\mathcal{O}$) into computable expressions on the generated data model, and finally (iv) compute the expressions using available techniques.*

Sec 5.2 shows mapping of user requirements to an EER diagram using the standard notations. In Sec. 5.3 we discuss the mapping of the EER diagram into homogeneous, heterogeneous, and hybrid MLNs along with an algorithm. Finally, in Sec. 5.4.1 we demonstrate with examples how the *analysis objectives are mapped* to expressions on the generated MLNs.

5.2 Application Requirements To EER Model

Any analysis objective to be computed from data involving multiple entities, features and complex relationships has been shown to benefit from a multilayer network model [28]. Enhanced Entity Relation diagrams ([96]) are well-established and have been used to model and design schemas for relational databases. An EER diagram is crucial to creating a good database design. In this section, we illustrate the first stage from figure 1.1 where requirements from three different sets of real-world analysis objectives (Section 5.1) are mapped to EER diagrams.

5.2.1 Internet Movie Database (IMDb) Analysis

The data set consists of top 500 actors and their co-actors across different movies, giving a total of 9000+ actors. Based on the information in the IMDb data set **and** analysis objectives (**A1-A3**), one can build an EER diagram (shown in Figure 5.1) as described below*:

- **Entities**: *Actor* with the key-attribute as name and nationality as a composite attribute comprising of the state and country.

*Note that the relationship details can change based on analysis objectives.

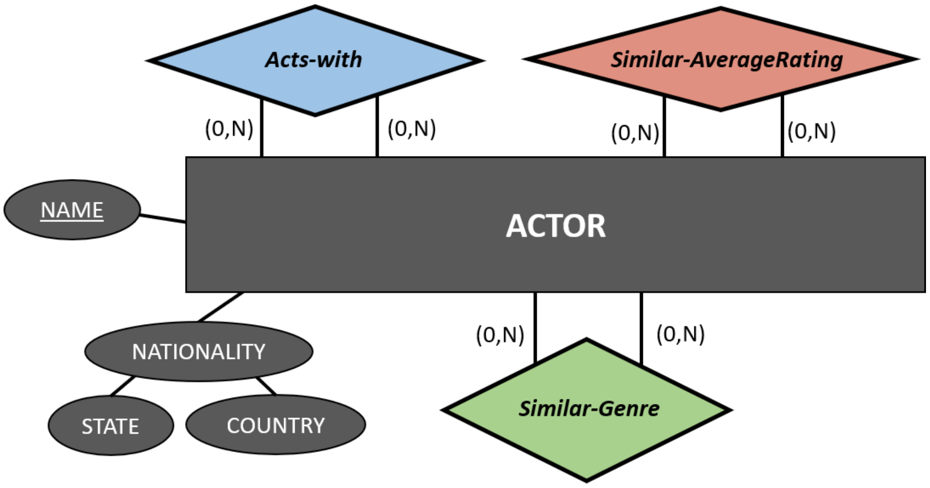


Figure 5.1: IMDb EER Diagram

- **Recursive Relationships**:
  - *Acts-with*: Two actors are related if they have worked in at least one movie
  - *Similar-Genre*: *Genre* is a categorical variable, as it takes fixed, limited number of values, such as "comedy", "action", etc. Also an actor acts in multiple movies of the same genre – i.e., in 3 action movies, 1 comedy movie, etc. For every actor we generate a vector with *number of movies for each genre*. We then compute the Pearsons' Correlation Coefficient (PCC) between the genre vectors for each actor pair. Two actors are related if PCC is at least 0.9[†].
  - *similar-AverageRating*: The movie ratings are given from 0 to 10. Note, however, when we take the average of the ratings, the values become real numbers. To evaluate the similarity we created 10 ranges - [0-1), [1-2), ..., [9-10]. Two actors are related if their average ratings fall in the same range.

[†] Choice of coefficient reflects relationship quality and its value can be based on how actors are weighted against genres. We have chosen 0.9 for relating actors in their top genres.

- **(Min, Max) Cardinality Ratios:** All relationships have *(0,N)..(0,N)* cardinality as an actor can be similar to none or multiple actors.

5.2.2 Database Bibliography (DBLP) Analysis

For DBLP, we have considered all publications from VLDB, SIGMOD, ICDM, KDD, DaWaK and DASFAA from the 2001-2018. Based on data set description and analysis objectives (**A4-A7**), the EER diagram shown in Figure 5.2 has been discussed below

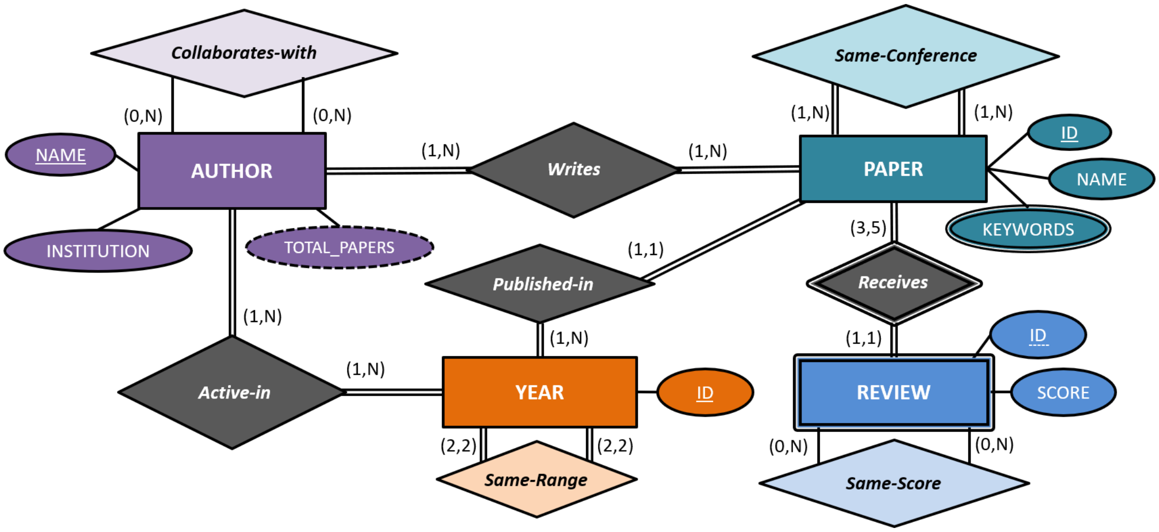


Figure 5.2: DBLP EER Diagram

- **Entities including Weak:**
  - *Author* with attributes - name (key) and institution. *Total_Papers* is a derived attribute that can be calculated using *writes* binary relationship.
  - *Paper* with attributes, Paper ID (key), name and keywords (multi-valued)
  - *Year* with year ID as the key attribute

  - *Review:* Existence of a review is dependent on the existence of a paper, thus it is a *weak entity*. It has ID (partial key) and score as the two attributes.

- **Recursive Relationships:**
  - *Collaborates-with*: Two authors are related if they have worked together on at least 3 published papers
  - *Same-Conference*: Two papers are related if they are published in same conference.
  - *Same-Range*: 3-year periods are required for analysis. Thus, the period from 2001 to 2018 is divided into 6 disjoint 3-year periods, from [2001-2003] to [2016-2018]. Two years are related in they are in the same 3-year period.
  - *Same-Score*: Typically, each review receives an overall score between 1 and 5 that can be rounded off. Thus, two reviews with the same score can be related.

- **Binary Relationships:**
  - *Writes*: A relationship to indicate if an author has written a paper.
  - *Active-in*: A binary relationship is created between author and year entities to denote whether an author was actively publishing in that year.
  - *Published-in*: Similarly relationship between paper and year entities is established to show in which year a paper was published.
  - *Receives*: Every paper published is related to all the reviews that it receives.

- **(Min, Max) Cardinality Ratios:**
  - *Collaborates-with* recursive relationship has cardinality ratio as *(0,N)..(0,N)* as each author can work individually or with any number of authors. *Same-*

Conference has cardinality *(1,N)..(1,N)* as many papers are published in the same conference, thus a paper is related to at least one paper. Cardinality of *Same-Range* is *(2,2)..(2,2)* as each year is related to the other 2 years in the 3-year period. *Same-Score* has *(0,N)..(0,N)* cardinality as a review may not be related to any other review.

- Binary relationship *Writes* between author and paper entity has *(1,N)..(1,N)* cardinality as an author can publish one or more papers and also paper can have one or more authors. Similarly, *Active-in* has *(1,N)..(1,N)* cardinality as an author is active in at least one year and in a given year many authors can be active. The *Published-in* relationship has *(1,1)..(1,N)* cardinality as paper is published only in one year but many papers can be published in a year. Finally, for *Receives* the cardinality is *(3,5)..(1,1)* as every paper receives 3 to 5 reviews, however each review is for exactly one paper.

5.2.3 Author-City Data set Analysis

For final set of analysis objectives (**A8-A10**) based on author-city data set, the EER diagram shown in Figure 5.3 has been discussed below

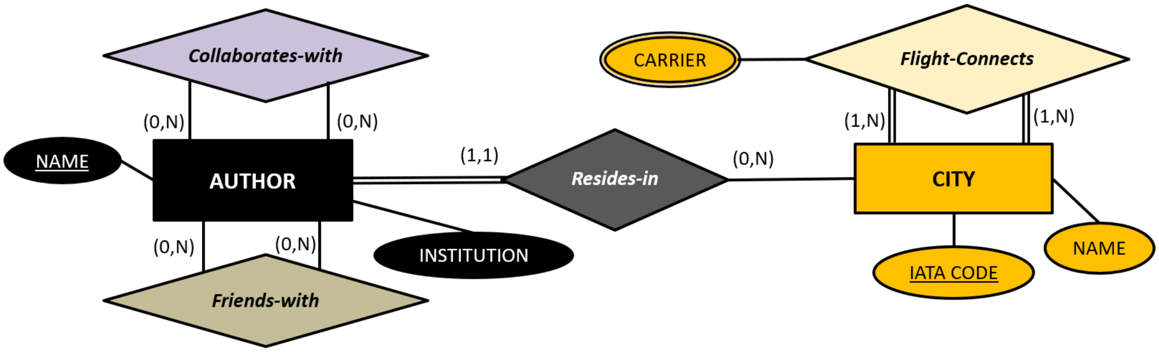


Figure 5.3: Author-City EER Diagram

- **Entities:**
  - *Author* with attributes - name (key) and institution
  - *City* with attributes - IATA/Airport Code (key) and name
- **Recursive Relationships:**
  - *Collaborates-with*: Two authors are related if they have worked together on at least 3 published papers.
  - *Friends-with*: A relationship to signify if two authors are friends on Facebook.
  - Flight-connects: Two cities are related if there is a flight connecting them with a multi-valued attribute to capture the operating *carriers*.
- **Binary Relationships:** A binary relationship, *Resides-in* exists between the author and city entity depicting the residence.
- **(Min, Max) Cardinality Ratios:**
  - *Collaborates-with* and *Friends-with* recursive relationships have *(0,N)..(0,N)* cardinality, as an author may work individually and may not be friends with anyone on Facebook, respectively.
  - Binary relationship *Resides-in* between author and city entity has *(1,1)..(0,N)* cardinality as an author can reside in only one city. However, a city may not be any author's residence or multiple authors can reside in it.

5.3 Generating Multilayer Networks From An EER Diagram

Here we discuss the steps involved in converting an EER model into a Multilayer Network. in Section 5.3.1 and 5.3.2) and address the different EER models discussed in Section 5.2.1, 5.2.2 and 5.2.3.

5.3.1 Algorithmic Steps for Translating An EER Diagram to MLNs

Below, we present our algorithm (8 steps) for generating an MLN (can be homogeneous, heterogeneous, or hybrid) from the EER diagram developed using the application requirements. These steps are somewhat different from the traditional EER diagram translation to a Database model. With each step, we explain the rationale and provide an example from the EER diagrams shown earlier.

A layer consists of nodes with a node id which is unique and a node label which need not be unique. An edge consists of an edge label which is not unique and connects two node ids. Typically, node ids are kept unique for the purposes of computation. Below, we assume node ids are generated as part of the translation process. The additional information of nodes and edges that come out of the EER diagram are maintained as .csv files which are used for drill down analysis of results. EER model also helps in modeling only those attributes of nodes and edges that are relevant to the analysis objectives and drill down.

1. **Each binary relationship** in the EER diagram corresponds to either an individual layer or a bipartite graph (of inter-layer edges) between two layers. Typically, entity id is used as the label of nodes in the layer. Other attributes are not typically stored as part of MLN (to reduce storage), but are stored separately (for example, as a relation or as a .csv file) for drill-down of the results later. The relationship name is used as intra- or inter-edge label and again, other relationship attributes are stored separately for drill down of results. We show some drill down results in Section 12.2.

   For example, the relationship Acts-with in Fig. 5.1 is translated into a layer Actor with name as node label and acts-with as edge label. In contrast, the relationship writes in Fig. 5.2 becomes a bipartite graph between the layers Paper and Author.

2. **Each binary recursive relationship** translates to a separate homogeneous layer whose intra-layer connectivity is defined by the relationship.

   For example, the layer Actor(Acts-with) in Fig. 5.4 (a) is obtained by the binary recursive relationship Acts-with in Fig. 5.1 on the Actor entity.

3. **Each binary non-recursive relationship** translates to a bipartite graph between the layers corresponding to entities of the relationship.This assumes that the layers have been formed earlier by binary recursive relationships.

   For example, Author-Year inter-layer edges in Fig. 5.1 (b) are formed by the relationship active-in in Fig. 5.2 between Author and Year entities.

4. **Translation of the attributes** (of an entity or a relationship) other than the key is done in the same way as we do for a relational model. Atomic, component, and multi-valued attributes are handled in the same manner. Derived attributes are not stored but are computed.

5. Hence, relationships have to be translated **in a specific order**: binary recursive first, followed by binary non-recursive relationships.

6. **Super and Sub entities** can be present in the EER diagram. If an entity type is a **super class**, either a layer can be created for it or layers can be created for each of its sub-class entity types depending on characteristics such as disjoint, overlapping, partial and total. This is quite similar to the translation to the relational model. Relationships present on these entities dictate the translation. Mapping of the relationships will follow the above steps.

   For example, it is possible that the super class may become a separate layer for some analysis objectives and sub classes may become separate layers for other analysis objectives. Different MLNs can be created from the EER diagram to meet the analysis objectives. Person as a super entity may have overlapping sub *entities actors and directors. If there are separate recursive relationships for the Person entity, it will become a separate layer.*

7. A **weak entity** and its non-recursive binary relationship is translated as follows. Unlike how it is done for the relational model, a **weak entity** is translated into a separate layer (using a binary recursive relationship on that entity) and the weak relationship is translated into a bipartite graph with edge labels indicating the dependence (combining the primary and the partial key).

   For example: The Review weak entity in Figure 5.2 becomes a separate layer in addition to the Layer Paper (Figure 5.4 (b)). The intra-layer edges are dictated by the Same-Score recursive relationship. This layer has a bipartite graph with the Paper layer with the inter-layer edge labels corresponding to the Paper ID and Review ID.

8. Currently, **n-ary relationships** that **cannot** be mapped to multiple binary relationships are not supported. If they can be mapped to multiple binary relationships, the above steps handle them. If not, such a relationship involves handling a **hyper-edge** across multiple layers which is beyond the scope of this thesis.

5.3.2 Summary of the Algorithm

The above algorithmic steps when applied translates an EER diagram to a MLN(s) along with drill down information in a form that is queryable and searchable. Below we make a few comments on the overall translation of the EER diagram.

- Each entity with **multiple** binary recursive relationships gives rise to a **Homogeneous MLN.**
- **Multiple entites** with *both* binary recursive (one each) and binary non-recursive relationships give rise to a **Heterogeneous MLN**.

- If the EER diagram has both kinds of entities and relationships as indicated above (as in 5.3) and there is at least one relationship between entities that form the homogeneous and heterogeneous layers, a **Hybrid MLN** is obtained.
- **Strong entities** as well as **weak entities** are translated as described above and become separate layers.
- The **min-max cardinality information** will give an insight into the minimum and maximum associations (or edges) that a node can have. This can help to calculate the *minimum, maximum and average degree* of the corresponding layer or bipartite graph.
- A **partial participation of an entity** translates to a node that is not connected to any other node (i.e., no intra- or inter-edge). *For example, the author can work individually (Partial Collaborates-with relationship).* Whereas a **total participation** implies every node has at least one edge.
- The **direction** of the inter or intra layer edges has to be implied from the semantics of the relationship. This can also be specified as part of the relationship. *For example, co-authorship will be bi-directional, whereas a relationship like follows-on-Twitter will be a directional.* This is typically specified as part of the application requirement and can be incorporated into the EER model relatively easily as part of the relationship using the (min, max) cardinality information.

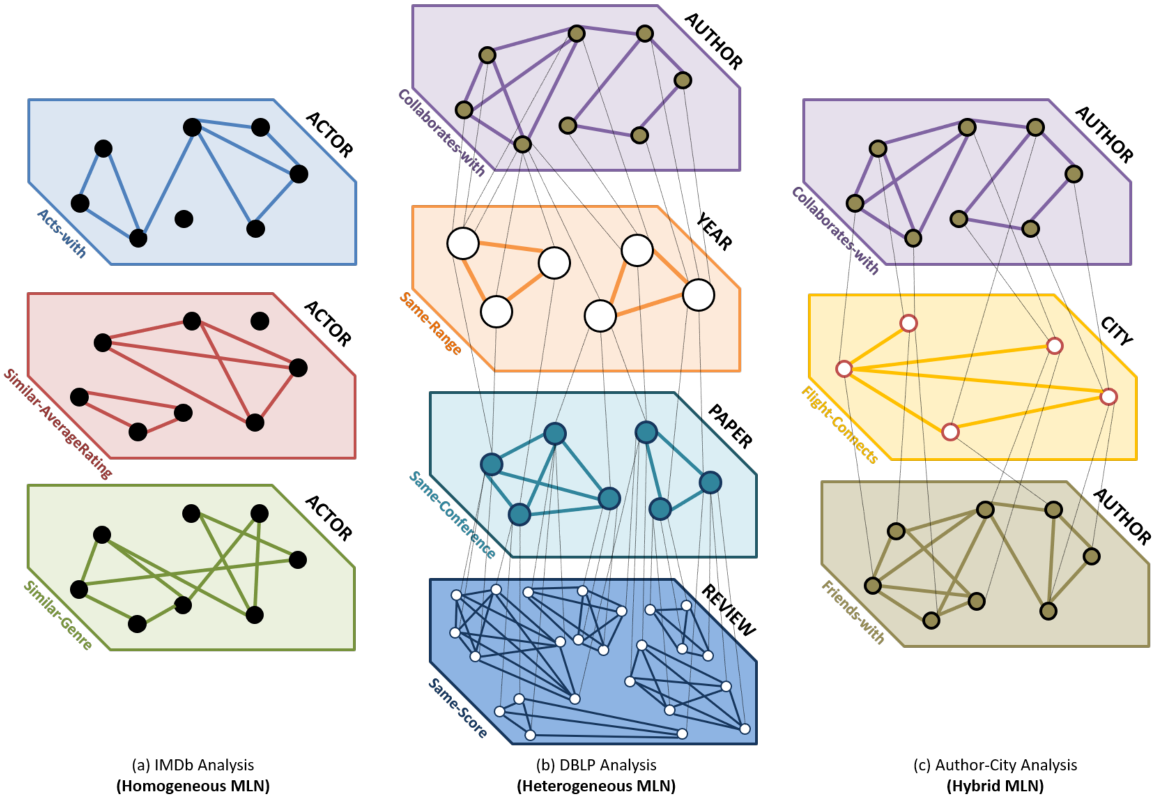
5.3.3 Application of the Above Algorithmmic Steps

For the 3 sets of analysis discussed in Sec. 5.2, the following MLNs, shown in Fig. 5.4, are generated by applying the above algorithmic steps. Node and edge labels have not been shown for simplicity.

IMDb Analysis: Based on the EER (Fig. 5.1), a **Homogeneous MLN** (Fig. 5.4 (a)) is obtained with 3 layers having every actor element as a separate node with intra-layer edges dictated by *Acts-with*, *Similar-AverageRating* and *Similar-Genre* recursive relationships (Using (**2**)). The node label is the actor name and intra-layer edge labels are the relationship names (Using (**1**)). Relationship semantics do not need a direction, thus edges are undirected.

DBLP Analysis: The EER in Figure 5.2 gets translated into a **Heterogeneous MLN** (Figure 5.4 (b)) with 4 layers - Author, Paper, Year and Review with intra-layer edges corresponding to *Collaborates-with*, *Same-Conference Same-Range* and *Same-Score* recursive relationships, respectively (Using (**2**), (**7**) for Weak Review Entity). The binary non-recursive relationships - *Writes, Active-in, Published-in, Reviews* generate 4 bipartite graphs between the layer pairs - Author-Paper, Author-Year, Paper-Year and Paper-Review, respectively (Using (**3**)). The node and edge labels are the key attributes and relationship names (Using (**1**), (**7**)). The relationships do not have an explicit requirement for direction, thus every intra/inter layer edge is *undirected.*

Author-City Analysis: The EER model in figure 5.3 leads to the generation of a **Hybrid MLN** (figure 5.4 (c)) with two Author Layers and a City Layer with intra-layer edges based on the *Collaborates-with*, *Friends-with* and *Flight-connects* recursive relationships (Using (**2**)). The binary non-recursive relationship *Resides-in* is used to introduce the inter-layer edges between the City layer and each of the Author layers (Using (**3**)). Node labels are name (Author layers) and IATA code (City layer), while the edge labels are relationship names (Using (**1**)). Collaboration, Residence and Friendship are bi-directional relationships. For the *Flight-connects* relationship it is assumed that if a flight exists from city a to city b, then a reverse flight also exists. Thus, every inter/intra layer edge is undirected in this HyMLN.



(a) IMDb Analysis
(Homogeneous MLN)

(b) DBLP Analysis
(Heterogeneous MLN)

(c) Author-City Analysis
(Hybrid MLN)

Figure 5.4: MLN Models for Analysis Set 1, 2 and 3

5.4 Analysis Objectives To Computation Specification

For the analysis of MLNs, a number of aggregate features are used for computation of objectives. They are: notions of community, centrality, and substructure.

5.4.1 Computation Specification Mapping

Once the EER diagram is created based on the application requirements (data set description + analysis objectives) and translated into MLNs, the next step is to map each objective into an expression using Θ and Ψ on the MLNs generated. This step is relatively easier to identify once the operators to apply and the type of composition to perform is determined, This step is similar to writing SQL queries once the specific database schema is generated and populated.

We show below how aggregate feature computation is specified along with composition to be used. The challenge in successfully applying network decoupling is to match the analysis function, Ψ and the composition function, Θ. Table 10.1 gives the mapping of each analysis objective **A1** to **A10** to their computation specification (in *left* to *right* order), analysis function (Ψ) and composition function (Θ). We will give a short overview of the composition process for each mapped analysis. The details of various types of composition functions have been discussed in subsequent chapters.

Analysis	**Mapping**		
	Computation Specification	Ψ	Θ
*IMDb (**HoMLN**)* 3 Actor Layers: *Acts-with, Similar-Genre, Similar-AverageRating*			
A1	*Acts-with* Θ *Similar-Genre*	Degree-Centrality	AND[8]
A2	*Acts-with* Θ *Similar-AverageRating*	Community	AND[7]
A3	NOT(*Acts-with*) Θ *Similar-Genre* Θ *Similar-AverageRating*	Community	AND[7]
*DBLP (**HeMLN**)* Author (Au), Year (Y), Paper (P), Review (R)			
A4	Au	Community	
A5	P Θ Au	Community	MWM[12]
A6	P Θ Au Θ Y	Community	MWM[12]
A7	Y Θ P Θ R	Community	MWM[12]
*Author-City (**HyMLN**)* City (C) and 2 Au Layers - *Collaborates-with*, *Friends-with*			
A8	*Collaborates-with* Θ *Friends-with*	Community	AND[7]
A9	C Θ *Collaborates-with*; C Θ *Friends-with*	Centrality (Degree)	HeMLN-Centrality
A10	*Collaborates-with* Θ *Friends-with* Θ C	Community(Au), Degree-Centrality(C)	MLN-Searching

Table 5.1: MLN Expression for Each Analysis Objective

IMDb Analysis: For **A1** using network decoupling, we first find the *high degree* nodes in *Acts-with* and *Similar-Genre* layers, separately to detect the popular co-actors and versatile actors. Using the AND composition we find all those *popular co-actors who are also highly versatile* (Details in Chapter 9.) For **A2**, the AND composition is applied on the communities from the *Acts-with* and *Similar-AverageRating* layers to generate and filter out the groups of co-actors who have high ratings. In **A3** aim is to find actors who have not acted together but act in the same genre and in movies of similar ratings – which increases their possibility of acting together in future. We apply the NOT operation on the Acts-with layer to find the complement graph of actors who have never acted together. In the first step of network decoupling, we take communities from each of the three layers; the Similar-Genre, Similar-AverageRating and the complement of the Acts-with layer. We then combine the resultant communities using the AND composition function to find *groups of actors who have a high chance of acting together in future* (Details in Chapter 6.)

DBLP Analysis: For **A4**, the Author layer communities will give the desired result. For **A5**, **A6** and **A7** the communities from Author, Paper, Year and Review layer need to be paired up in the specified order to meet the analysis objectives. In chapter 11, the HeMLN community detection has been proposed where for any two layers a bipartite graph is constructed using their communities. Each community is considered to be a meta-node. Two meta-nodes in two different layers are connected if there is at least one inter-layer edge between them. The weight of these edges (meta-edges) between the meta-nodes is given by the number of inter-layer edges between them. These meta nodes (communities) in the bipartite graph are uniquely paired using the composition function (Θ) Maximal Weighted Matching (MWM) that maximizes the overall meta-edge weight and is based on traditional matching proposed by Jack Edmonds [97]. For **A5**, the Author communities that get *matched* with Paper communities (corresponding to conferences) are the most popular. For **A6**, the matched Author communities from A5 are paired with Year communities to find their most active periods. For **A7**, first Paper communities are matched to Year communities to obtain the highly publishing conferences per period. Then, the matched Paper communities are matched to Review communities, to get the *most popular review score*.

Author-City Analysis: **A8** is computed by the AND composition on the communities from two Homogeneous Author layers. For **A9**, the cities having high inter-layer degree with any one of the author layers are the *cities with high author concentrations*. In **A10**, ideally a conference will get more attendance if it is organized in a city that is a) well-connected via flights, b) where large co-author communities reside and c) large sections of those co-author groups are friends in order to maximize the advertisement of the conference. Thus, using the decoupling approach the communities from the two author layers and high degree nodes from the City layer are composed (and filtered) in order to obtain the desired set of *probable venues for a conference*. Analysis of HyMLNs have not been handle as a part of this thesis.

5.5 Conclusions

In this chapter, we have addressed the problem of leveraging the power of the EER modeling to generate Multilayer Networks and expressions for their analysis. *Ad hoc* big data analysis without a formal approach to generating models from application requirements is difficult, error-prone, and not amenable to revisions and future extensions. This is a big concern for big data analysis today. The work from this chapter has been accepted in the International Conference on Conceptual Modeling (ER) [14].

In the subsequent chapters, we discuss the proposed techniques to compute the mapped analysis objectives using the decoupling approach.

CHAPTER 6

COMMUNITY DETECTION IN HOMOGENEOUS MLNs

In this chapter, we focus on the class of analysis questions that require one to find out *entity communities with respect to different combinations of features (layers)* based on the homogeneous multilayer networks.

Communities in networks *are groups of tightly connected nodes.* In HoMLNs, the same set of entities are present in each layer. We have assumed each layer to unweighted and undirected. We have proposed the combination of layers (features) using Boolean operations, AND, OR and NOT. For the AND (OR) operation, the combined network will contain an edge, if there exists an edge in *all* (*any one*) of the individual layers. For the NOT operation, the complement of the network will be considered. An *AND-Composition* represents how multiple features together affect an analysis. For example, **in identifying regions that become accident prone due to poor lighting conditions *as well as* bad roads**. An *OR-Composition* represents how any one of the features affects a property. For example, in **finding the group of people who are friends via least one of the social networking platforms among Facebook, LinkedIn and Twitter**. A *NOT operation* represents filtering out some layers, such as **people who are friends in Facebook, but are not connected via LinkedIn**.

The primary challenge is to design appropriate aggregation functions, such that the communities obtained using network decoupling are similar to those obtained by applying community detection on the composed network. [*] Formally, our problem can be stated as follows;

Problem Statement. Given a set of layers $G_1, G_2, \ldots, G_x$, that are combined using a Boolean operation $\bigoplus$ to form the composed network, and a community detection algorithm $COMM$, that is used to find communities, develop an aggregation algorithm Π, such that

$$COMM(\bigoplus_{i=1}^{x}(G_i)) \approx \Pi_{i=1}^{x}(COMM(G_i))$$

.

In other words, we aim to find an aggregation algorithm Θ, such that *the results of finding the communities in the individual layers and then aggregating them via* Π*, should be the same as the communities obtained from the composed network where the layers are combined using the Boolean operator* $\bigoplus$*.* Developing the aggregation algorithm is challenging, since the structure of the composed network can change after the layers are combined, and the aggregation process has to appropriately account for that change when combining the communities.

Our Contribution. Our main contribution, therefore is to *develop correct aggregation functions* that will allow us to apply network decoupling for efficiently finding communities based on different Boolean compositions of networks. The principle is to first analyze the communities in each individual layer and then aggregate the results, using appropriate functions, to obtain the final results on the composed network (see Figure 6.1). Thus, we only need to analyze each network once, and then combine the results as per the aggregation method.

[*] We state that the communities should be similar rather than identical, because community detection is non-deterministic, and even slight changes in the algorithm or order in which the vertices are processed can slightly alter the results.

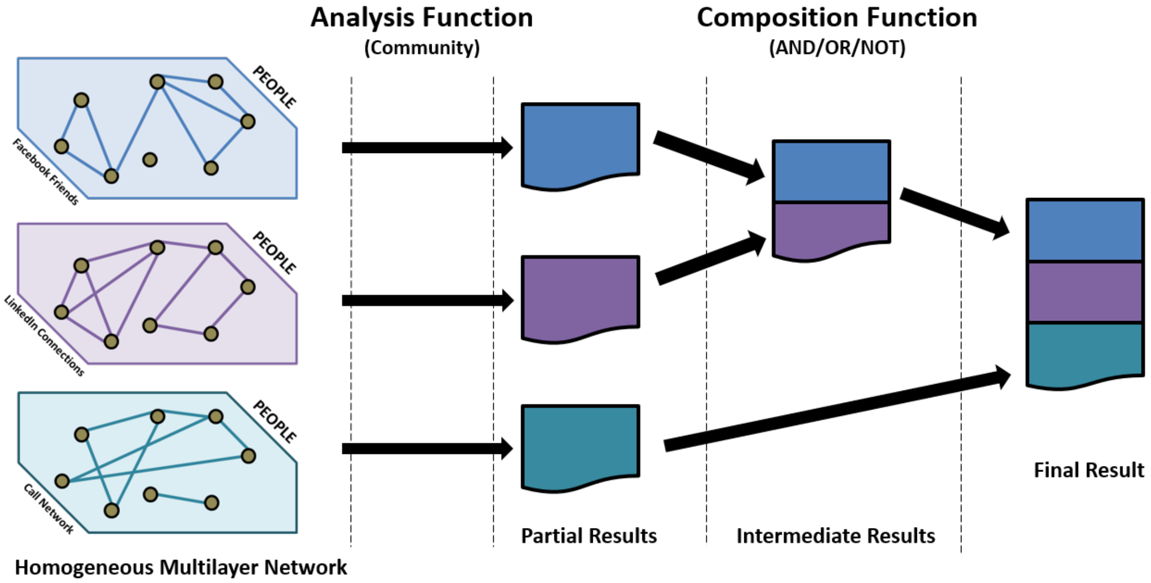


Figure 6.1: Illustration of network decoupling for community detection in Homogeneous Multilayer Networks

The remainder of the chapter is organized as follows. In Section 6.2, we provide a brief description of community detection in MLNs. In Section 6.3, we present our contribution of community detection using network decoupling using AND and OR. In Section 6.7 we present the experimental results related to these operations. In Section 6.9 we show how combination of boolean AND, OR and NOT expressions can be used to answer complex queries.

6.1 Modeling the IMDb Dataset as a HoMLN based on Genres

We use the Internet Movie Database (IMDb) to illustrate how communities are generated for different Boolean combinations of features. The IMDb is an online database that contains information on television programs and movies including actors, directors, genre, and year of release [94].

We create a HoMLN where the entities represent actors and two actors are connected to each other if they have acted in the same movie. Each layer in the HoMLN

Table 6.1: List of notations used for defining the concepts.

N_L	Number of layers
I	Set of entities
f	Set of features/layers
$G(V_k, E_k)$ or G_k	The k^{th} layer
u_k^i	Represntative node for i^{th} entity in the k^{th} layer
V_k	Set of nodes in the k^{th} layer
(u_k^i, u_k^j)	An edge in the k^{th} layer
E_k	Set of edges in the k^{th} layer
$C(V_k^m, E_k^m)$ or C_k^m	The m^{th} community in the k^{th} layer
V_k^m	Set of nodes in C_k^m
E_k^m	Set of edges in C_k^m

represents a movie genre, such as comedy, drama, action, etc. The IMDbActor-Genre HoMLN defined is shown in Figure 6.2.

In Figure 6.2 we have selected two genres, comedy (f^1) and drama (f^2) to form the two layers, G_1 and G_2, respectively. This HoMLN shows the co-actor relationship among 16 actors (denoted by nodes numbered from 1 to 16) with respect to these genres. The same 16 actors are present in both layers. Note that each co-actor network has a distinct structure. By taking the information from the two networks together we can gain interesting insights to the data, as follows.

For example, actors I_3 and I_8 have never worked together in a drama, but have worked together in a comedy. Thus this pair of actors may together be more likely to be considered in a comedy, rather than a drama. Also observe that the actor I_{14} is the actor with most connections in the drama genre, while in the case of comedies, actor

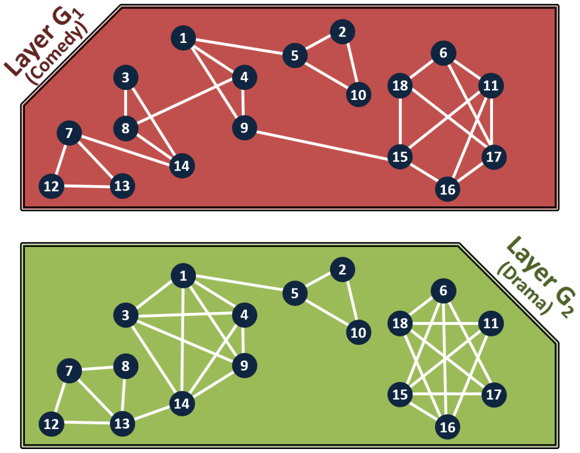


Figure 6.2: Example of the IMDb HoMLN for co-actors with 16 actors and two genres: comedy and drama.

I_{11} is one of the nodes with the most connections, i.e. worked with most number of actors.

6.2 Community Detection in HoMLN

Community detection involves finding collection of items with similar properties by identifying *tightly connected groups of vertices*. Each vertex is mapped to its specific community number. We consider non-overlapping communities, that is, there are no common vertices or edges between two communities. Figure 6.3 shows the communities in the individual layers. For example, Actor I_7, I_{12} and I_{13} for Comedy-based co-actor group. However, with respect to Drama-based movies, these Actor I_8 also joins the group to form tighly connected community.

Bridge Edges. We term the external edges that connect two communities as *bridge edges*. Formally, if there exists an edge, (u_k^i, u_k^j), such that $u_k^i \in C_k^m$ and $u_k^j \in C_k^n$, where $m \neq n$, then this edge is a bridge edge. Bridge edges form links between two distinct communities. For example, Actors I_{13} and I_{14} act as the link between two strongly connected group of Drama-based co-actors.

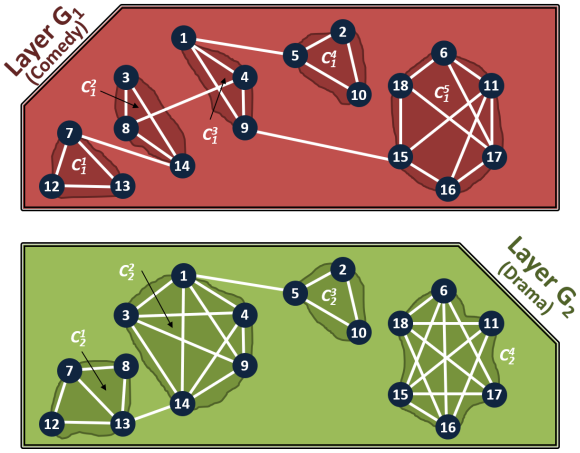


Figure 6.3: Communities in each layer of the IMDbActor-Genre HoMLN

6.2.1 Communities in AND-Composed Layers.

AND composition of layers in a HoMLN allows users to find communities that are *related across multiple features*. Examples of some questions that can be addressed by the AND composition in different domains are;

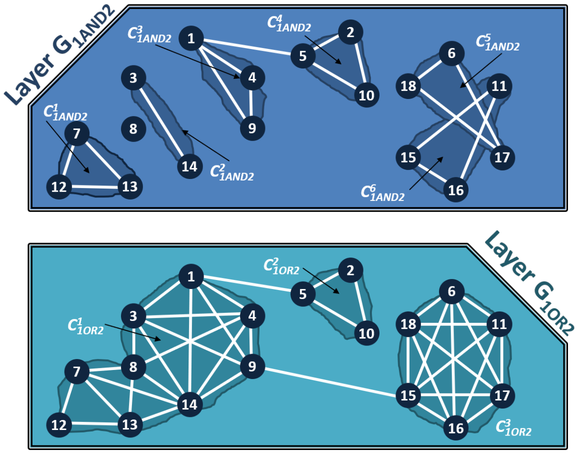


Figure 6.4: Composed Layer Communities of the IMDbActor-Genre HoMLN shown in Figure 6.2

- Groups of actors who have expertise in working together in *both* comedies **and** dramas (IMDbActor-Genre HoMLN).

- Author groups who publish in all of *these* conferences; ICDM, SIGMOD **and** VLDB (DBLP HoMLN).
- Groups of people who are connected to each other through *all these* social networking platforms - Facebook, LinkedIn, WhatsApp, Instagram **and** Twitter (Social Network HoMLN)
- Groups of research papers that cite each other and have the keywords - big data, graph mining, multilayer networks **and** community (ArXiV HoMLN).
- Groups of accidents that have similar conditions for *all these* features; light conditions, weather conditions, road conditions, and speed limit (Accident HoMLN).

The standard practice is to combine the layers using the AND operation, i.e. only edges that occur in all the layers are included. Then a community detection algorithm, such as Infomap, is executed on the combined network. This single graph approach, termed C-SG-AND, is given in Algorithm 1. Figure 6.4 (top) shows the communities for the AND-composed layer, G_{1AND2}, for IMDbActor-Genre HoMLN.

Algorithm 1 Algorithm for C-SG-AND

```
Require: Layers $G_1, G_2, \ldots G_x$
Ensure: return $L^{AND}_{1,2,\ldots,x}$ - a list of communities
    $G_{1AND2 \ldots ANDx} \leftarrow \{G_1$ AND $G_2 \ldots$ AND $G_x\}$
  { $G_{1AND2 \ldots ANDx}$ contains edges that are in all the networks $G_1$, $G_2$, $\ldots$, $G_j$.}
    $L^{AND}_{1,2,\ldots,x} = \text{COMM}(G_{1AND2 \ldots ANDx})$
  {Find communities in $G_{1AND2 \ldots ANDx}$.}
```

6.2.2 Communities in OR-Composed Layers

OR-composition forms a composed network that includes an edge if it appears in any of the layers. Algorithm 2 shows the steps of this single network based community detection using the OR operation, termed as C-SG-OR. Figure 6.4 (bottom) shows the communities for the OR-composed layer, G_{1AND2}, for IMDbActor-Genre HoMLN. Examples of queries that can be addressed by the OR composition are;

- Groups of actors who have acted together in either a comedy **or** drama (IMDbActor-Genre HoMLN).
- Groups of authors who have published in **at least one** of these conferences, ICDM, VLDB, SIGMOD (DBLP HoMLN).
- Groups of accidents that have at **least one** condition in common (Accident HoMLN).

Algorithm 2 Algorithm for C-SG-OR

```
Require: Layers $G_1, G_2, \ldots G_x$
Ensure: return $L^{OR}_{1,2,\ldots,x}$ - a list of communities
    $G_{1OR2\ldots ORx} \leftarrow \{G_1$ OR $G_2 \ldots$OR $G_x\}$
    { $G_{1OR2\ldots ORx}$ contains edges that are in *at least one* of the networks $G_1$, $G_2$, $\ldots$, $G_x$.}
    $L^{OR}_{1,2,\ldots,x} = \text{COMM}(G_{1OR2\ldots ORx})$
    {Find communities in $G_{1OR2\ldots ORx}$.}
```

6.3 Network Decoupling for Community Detection on HoMLNs

The Boolean composition of the layers of a HoMLN provides in-depth analysis of the dataset. However, for any single Boolean operation, say AND, $2^N - 1$ different combinations are possible. Thus the cost of finding the communities on each of them separately is very expensive. Moreover, if the networks do not change, several computations are rendered redundant. For example, consider finding the communities in the composed layer $G_{1AND2AND3}$ and $G_{1AND2AND4}$. In this case, the composed layer related to G_{1AND2} remains unchanged, but has to be recomputed. Moreover, common sub-expressions have to be computed multiple times.

As a solution, we propose network decoupling for efficient community detection on HoMLN networks. In network decoupling, the communities in each layer are identified separately and the results are then aggregated to obtain the results with respect to the composed network. Note that the storage required is only of the order of $O(V * f)$, where V is the number of vertices in each layer and f is the number of features/layers. Figure 6.3 shows the communities in each of the layers of the example IMDB network.

The **challenge** is to develop aggregation algorithms, Θ, that can correctly aggregate the communities from each of the layers to obtain the communities over the composed network. We now present the aggregation methods for AND and OR composition. For ease of understanding we will discuss the algorithms with respect to two layers. Note, however, that any binary operations can be easily extended to multiple layers.

6.4 Vertex based Community Detection of AND Composed Layers (CV-AND)

The first approach, termed CV-AND, for obtaining communities in AND-composed layers was published in [7].

CV-AND (see Algorithm 3) proposed that the communities of the AND-composed graph can be obtained by taking the vertex based intersection of the communities from the individual layers. This method heavily depends on the presence of *self-preserving communities* in the individual layers and its correctness has been discussed in Lemma 6.4.1.

Correctness of CV-AND: We first introduce the concept of *self preserving* communities. A community is self preserving if the vertices in it are so tightly connected such that even if only a subset of connected vertices remain in a community, they

Algorithm 3 Algorithm for CV-AND

```
Require: Communities from layers $G_i$ and $G_j$:
    $\text{COMM}(G_i) = \{C_i^1(V_i^1, E_i^1),\ C_i^2(V_i^2, E_i^2),\ \ldots,\ C_i^x(V_i^x, E_i^x)\}$,
    $\text{COMM}(G_j) = \{C_j^1(V_j^1, E_j^1),\ C_j^2(V_j^2, E_j^2),\ ...,\ C_j^y(V_j^y, E_j^y)\}$
Ensure: return $L_{i,j}^{CV-AND}$ - a list of communities
  $L_{i,j}^{CV-AND} = \Phi$
  {Initialize the set of communities to NULL.}
  for each community pair say, $C_i^p$ and $C_j^q$ do
      $C_{i,j}^{p,q} =(V_i^p \cap V_j^q)$
      {Create new combined community by taking the common vertices of every pair of communities.}
      $L_{i,j}^{CV-AND} = L_{i,j}^{CV-AND} \cup C_{i,j}^{p,q}$
      {Add new community to the set of communities.}
  end for
```

will form a smaller community rather than joining an existing larger community. Formally, consider a network G, that has a community whose vertices are given by the set C_v. Now consider the network induced by a subset of vertices $C_v^S \in C_v$. If the vertices in C_v^S form a community by themselves, for any subset C_V^S of C_v, where $\|C_v^S\| \geq 3$ and the vertices in C_v^S are connected, then community C_v is self preserving.

Lemma 6.4.1. If the communities in networks G_x and G_y are self-preserving, then the AND-combination of the communities produced by each of these networks will be the same as the communities produced by the AND-composed network created from G_x and G_y

Proof. Consider two networks G_x and G_y that have the same set of vertices, but different set of edges. Moreover, both networks have only self-preserving communities.

Now consider the network G_{xANDy}, which is the AND-composition of G_x and G_y. Only edges that are in both G_x and G_y will be in the AND-composed network. Therefore the communities formed in the AND-composed network will be based on a subset of edges from G_x and G_y. Since both G_x and G_y have self preserving communities, therefore the communities formed in G_{xANDy} will be formed subsets of the communities in G_x and G_y. Most importantly, due to the self preserving nature, no new grouping of vertices will be formed in G_{xANDy}. Therefore we can reconstruct the communities in G_{xANDy} by taking the intersection of the communities of G_x and G_y. □

Drawbacks. The main drawback of CV-AND is that for most networks, there is no guarantee that the communities will be self-preserving. If this algorithm is applied without testing for self-preserving communities, the results may not be accurate.

As an example, consider the community C_1^5 in the comedy layer of the network (Figure 6.3). This community is not self preserving, and when combined with community C_2^4 in the drama layer, which has the same vertices, it gives one large community, $\{ I_6, I_{11}, I_{15}, I_{16}, I_{17}, I_{18}\}$. In reality, as seen in Figures 6.4, two separate communities are formed, $\{ I_6, I_{17}, I_{18}\}$ and $\{ I_{11}, I_{15}, I_{16}\}$.

This is a *subtle but important difference* because the community id determines whether two entities are similar. If two disconnected groups of vertices are placed in the same community (as is possible when using CV-AND), then, two dissimilar groups are marked to be similar, which is incorrect.

6.5 Edge-based Community Detection of AND Composed Layers (CE-AND)

We address these limitations by developing a community detection method, CE-AND (see Algorithm 4), that is based on the intersection of *edges* rather than *vertices* as follows.

Algorithm 4 Algorithm for CE-AND

```
Require: Communities from layers $G_i$ and $G_j$:
    $\text{COMM}(G_i) = \{C_i^1(V_i^1, E_i^1),\ C_i^2(V_i^2, E_i^2),\ ...,\ C_i^x(V_i^x, E_i^x)\}$,
    $\text{COMM}(G_j) = \{C_j^1(V_j^1, E_j^1),\ C_j^2(V_j^2, E_j^2),\ ...,\ C_j^y(V_j^y, E_j^y)\}$
Ensure: return $L_{i,j}^{CE-AND}$ - a list of communities
    $L_{i,j}^{CE-AND} = \Phi$
    {Initialize the set of communities to NULL.}
for each community pair say, $C_i^p$ and $C_j^q$ do
    $\{C_{i,j}^{p,q}\} = (E_i^p \cap E_j^q)$
    {Create list of k new communities by taking the common edges of every pair of communities.}
    $L_{i,j}^{CE-AND} = L_{i,j}^{CE-AND} \cup \{C_{i,j}^{p,q}\}$
    {Add new communities to the set of communities.}
end for
```

For every pair of communities, $C_i^m(V_i^m, E_i^m)$ from layer G_i and $C_j^n(V_j^n, E_j^n)$ from layer G_j, the edge-based community intersection, $E_i^m \cap E_j^n$, will produce k disconnected edge-sets, E_{iANDj}^1, E_{iANDj}^2, ..., E_{iANDj}^k. These edge sets will form the AND-composed communities, $C_{iANDj}^1, C_{iANDj}^2, ..., C_{iANDj}^k$.

Figure 6.5 shows how the communities are obtained for the example network using CE-AND. Comparing this result to that in Figure 6.4, we see that most of the communities are obtained with the exception of the singleton node 8. The common bride edge (1, 5) is also missing.

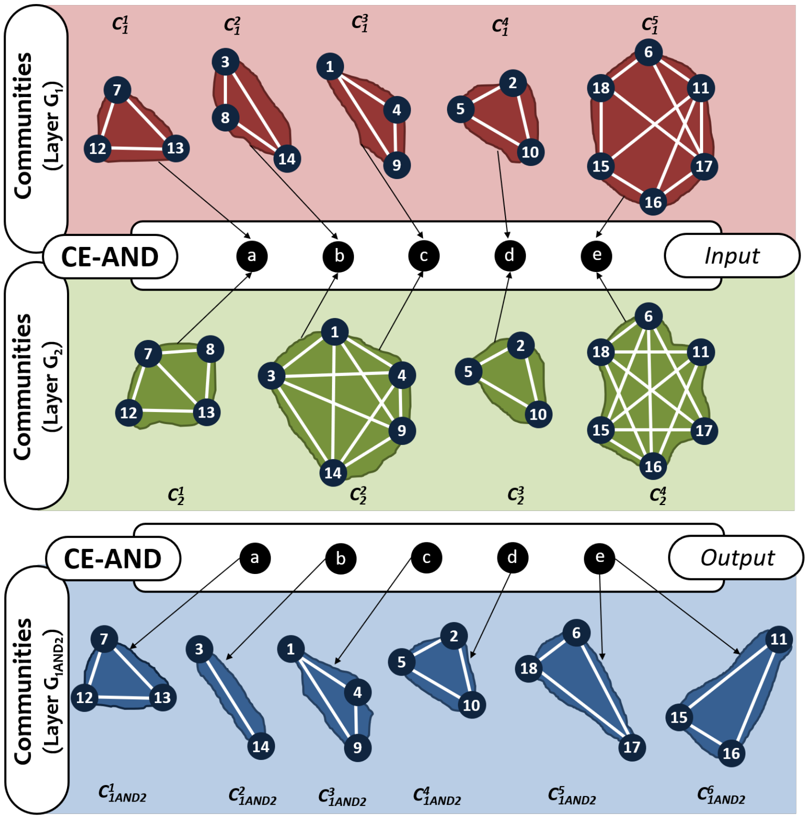


Figure 6.5: AND-Composition Communities of the Multiplex in Figure 6.2, using CE-AND method

Proof of Correctness. Algorithms 3 and 4 produce a set of disjoint clusters. Algorithm 1 produces a set of communities in the AND-composed network. We consider these communities as the ground truth communities. We label each edge as internal (if both end points are in the same community) and external or bridge otherwise.

We assume that the communities in the individual layers and the composed network have high clustering coefficients, i.e. we do not consider accidental communities

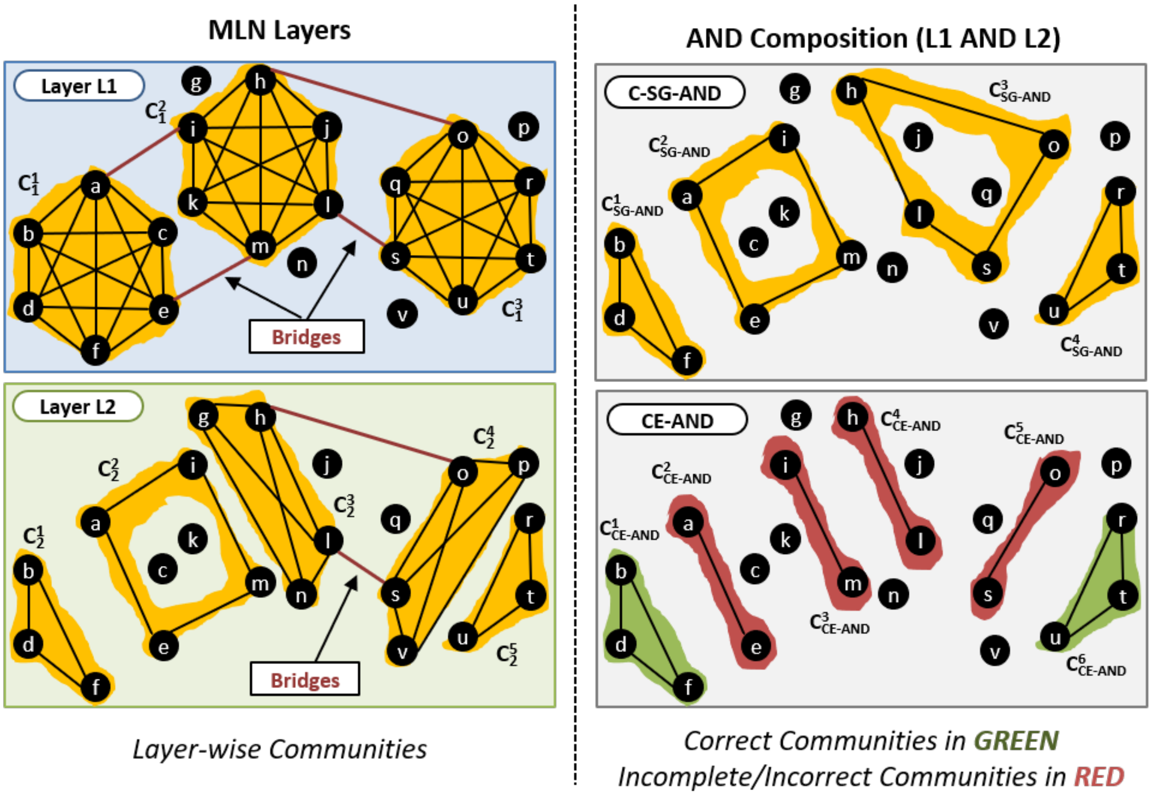


Figure 6.6: Effect of Bridge Edges on AND Composition

such as an edge or a line graph, that are formed due to an artifact of the community detection algorithm rather than the structure of the network. If such trivial communities are formed, we consider each vertex in them as a singleton community. The clusters formed by the intersection algorithms do not have this restriction, since they are not obtained using community detection algorithms.

We now prove the correctness of our proposed algorithms by discussing how well the clusters obtain by our proposed network decoupling method, correspond to the ground truth communities. Let the set of communities obtained from the composed network be Γ. Let the set of clusters obtained using the CE-AND algorithm be Ψ.

Lemma 6.5.1. *For any given cluster $X \in \Psi$, there will exist a set of communities $\{C_1^X, \ldots C_m^X\}$, where $C_i^X \subseteq \Gamma$, $1 \leq i \leq m$, such that the union of the vertices in*

$\{C_1^X, \ldots C_m^X\}$ form a partition of the vertices in X, if and only if, the set of edges common to all layers have the same label in all the layers.

Proof. We first prove the condition that if the common edges have the same label in all the layers, then the set of the union of vertices in $\{C_1^X, \ldots C_m^X\}$ will form a partition of the vertices in $X \in \Psi$.

Let the set of vertices belonging to the cluster X be U_X. Let the set of vertices belonging to community C_i^X be V_i^X, and $\cup_{i=1}^{i=m}(V_i^X) = V_X$, i.e. the union of these vertices in V_X. Since the communities are disjoint to prove that V_X is a partition of U_X, we have to prove that $V_X = U_X$.

It is easy to show that there exists a set of communities such that $U_X \subseteq V_X$. We simply select the communities such that all vertices in U_X are included.

To prove $V_X \subseteq U_X$ by contradiction, let v be a vertex that is in set V_X but not in U_X. Since CE-AND retains all the common internal edges, and v is not in U_X, therefore v will be connected to its neighbors in V_X by one or more external (or bridge) edges. Since we assume that all common edges have the same labels, therefore in none of the layers v is tightly connected to any subset of V_X. Moreover, we assume that the communities in the composed network have high clustering co-efficient (or are singletons). Thus since v is not tightly connected to vertices in V_X it cannot be part of the community. Thus our assumption was wrong, and $V_X \subseteq U_X$.

Taken together, $V_X \subset U_X$ and $U_X \subset V_X$; thus $U_X = V_X$, and V_X is a partition of U_X.

For the only if part we show that if the common edges do not have the same labels in all the layers, then there may not exists set of communities that form a partition of the vertices in a given cluster.

We provide such an example in Figure 6.6. The left-hand panels of Figure 6.6 shows two layers. The top right panel shows the communities obtained by the standard single network approach (C-SG-AND). The bottom right panel shows the communities obtained by CE-AND.

Note that the community C^3_{SG-AND} produced by C-SG-AND contains the edges (h, o) and (l, s) that act as bridges in both Layer L1 and L2. Thus CE-AND is not able to detect this community, and instead produces two communities, C^4_{CE-AND} and C^5_{CE-AND}, which should be merged into one by taking the bridge edges into account.

Also consider the community C^2_{SG-AND} which consists of the edges (a, i) and (e, m) that are bridges in Layer L1, but are part of the community C^2_2 in Layer L2. As only those edges that are within community *in all layers* are considered, CE-AND produces two communities, C^2_{CE-AND} and C^3_{CE-AND}.

□

6.6 Edge based Community Detection of OR Composed Layers

We now consider how to obtain communities in composed networks formed using the OR operation (termed as *OR composed networks*). The number of edges in the OR-composed network is the union of the edges in each layer. For any two layers G_i and G_j, the total number of edges is $|E_i \cup E_j|$.

The computational complexity of community detection algorithms are at least proportional to the size of the graph. The denser the graph, the more time will be required to find the communities. Thus for the OR-composed case, our goal is not only to lower the time by reducing the need to recompute different compositions of layers, but also to reduce the size of the graph to be analyzed.

To obtain communities of OR Composed Layers, we propose the CE-OR algorithm (given in Algorithm 5 and illustrated in Figure 6.7). The CE-OR method reduces the size of the graph to be analyzed by leveraging the fact that the common communities across layers can be processed as a single node. The steps of the CE-OR algorithm are as follows;

Algorithm 5 Algorithm for CE-OR

```
Require: Communities from layers $G_i(V, E_i)$ and $G_j(V, E_j)$:
    COMM($G_i$) = $\{C_i^1(V_i^1, E_i^1)$, $C_i^2(V_i^2, E_i^2)$, ..., $C_i^x(V_i^x, E_i^x)\}$,
    COMM($G_j$) = $\{C_j^1(V_j^1, E_j^1)$, $C_j^2(V_j^2, E_j^2)$, ..., $C_j^y(V_j^y, E_j^y)\}$
Ensure: return $L_{i,j}^{CE-OR}$ - a list of communities
    { Find common communities using CE-AND}
    Apply CE-AND on $COMM(G_i)$ and $COMM(G_j)$ to get $L_{i,j}^{CE-AND}$
    Construct OR-MG($V_{OR-MG}, E_{OR-MG}$)
    {Assign nodes of each common community as a meta node}
    for each community $C_k(U_k, E_k) \in L_{i,j}^{CE-AND}$ do
        $V_{OR-MG} = V_{OR-MG} \cup U_k$
    end for
    { Assign the vertices not in any common community as a meta node}
    for each vertex $u \notin C_k$ ,$\forall C_k \in L_{i,j}^{CE-AND}$ do
        $U_k = \phi$ {Create null set}
        $U_k = U_k \cup u$ {Add $u$ to the set}
        $V_{OR-MG} = V_{OR-MG} \cup U_k$
    end for
    {Add Edges in the metagraph. Two metanodes, $(U, V)$ are connected if there is an intra-community edge from one constituent node of $U$ to a constituent node of $V$ in any one of the layers.}
    for all all metanode pairs $(U, V) \in V_{OR-MG}$ do
        if $\exists\, u, v, r$: $(u, v) \in E_i^r$ or $(u, v) \in E_j^r$, $u \in U$ and $v \in V$ then
            $E_{OR-MG} = E_{OR-MG} \cup (U, V)$
        end if
    end for
    Insert weights on the edges of OR-MG
    L = COMM(OR-MG)
    Expand the community representative nodes in each community from L to get $L_{i,j}^{CE-OR}$
```

Overview of CE-OR. Find the common communities in all the network layers (Line 1) by using CE-AND. Then construct a metagraph (OR-MG), as follows. Each metanode represents *a set* of vertices. Combine each of the vertices in a common community into a metanode (Line 2-4). All vertices that are not assigned into communities are each assigned to a metanode (Line 5-9). Connect two metanodes, U and V via a metaedge, if there exists an internal edge, in *at least one* of the layers between an element (node) of U and an element (node) of V (Line 10-14). Apply appropriate weights to these edges (Line 15). Apply community detection on the metagraph (Line 16). The communities in the OR-composed network are obtained by expanding the metanodes in the communities obtained by the CE-OR algorithm.

Assigning Weights to Metaedges. Note that the metanodes represent vertex sets of varying sizes, and the number of edges between them represent the degree of similarity. Therefore although the original graph is composed of unweighted edges, the edges in the metagraph have to be weighted to quantify the extent of this similarity. A critical component of the CE-OR algorithm is based on correctly assigning these weights. We connect two meta nodes only if at least one pair of vertices from each meta node are connected by an internal edge, *in at least one of the layers.*

For any meta edge (A, B), let V_A and V_B be the set of nodes in the meta communities A and B, respectively. Further, let the set of all edges (internal with respect to at least one layer) between V_A and V_B be $E_{A,B}$. We use the following two strategies to compute the weight of the metaedge;

- *Aggregation:* The weight w_a is the number of edges between the two communities; $w_a(A, B) = |E_{A,B}|$
- *Fractional:* The weight w_f is the fraction of connected nodes between the two communities; $w_f(A, B) = \frac{|E_{A,B}|}{|V_A|*|V_B|}$.

Figure 6.7 illustrates how the CE-OR algorithm is applied to identify communities in the OR-composed layers of the example IMDb graph. First the CE-AND communities obtained in Figure 6.5 and the remaining vertex I_8 are used to form the metanodes (Figure 6.7 (a)). Then these nodes are connected based on the internal edges. These edges are weighted in the metagraph using w_f (Figure 6.7 (b)). A community detection algorithm on the metagraph produces the communities of the OR-composed layers (Figure 6.7 (c)). Comparing with the communities obtained by the C-SG-OR method in Figure 6.4, to those obtained by expanding the communities in the metanodes (Figure 6.7 (d)), we see that all the communities have been obtained. However, the bridge edges between the communities are missing.

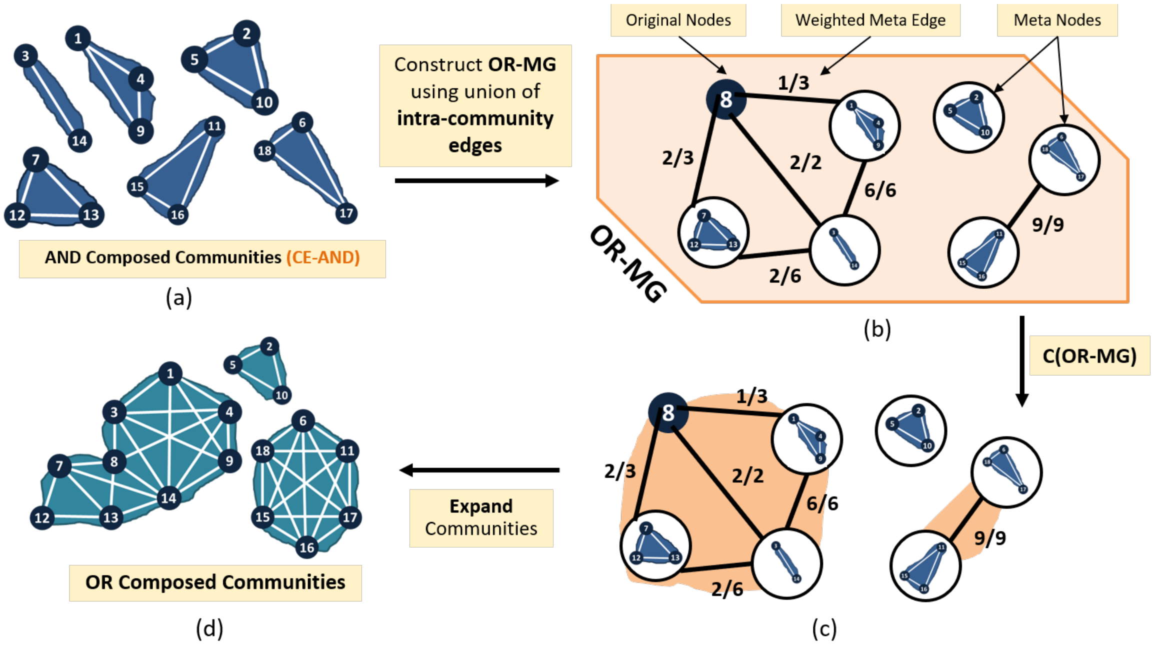


Figure 6.7: Illustration of CE-OR Algorithm on the Example Graph

Proof of Correctness Similar to the CE-AND algorithm, we prove the correctness of our proposed CE-OR algorithm, by comparing the communities obtained by CE-OR to those obtained by executing community detection on the composed network. We define a metanode cluster, Y, as all the metanodes in a connected component of the metagraph. Let the communities obtained through the C-SG-OR algorithm be Λ.

Lemma 6.6.1. *For a given metanode cluster Y, there will exist a set of communities $\{C_1^Y, \ldots C_m^Y\}$, where $C_i^Y \subseteq \Lambda$, $1 \leq i \leq m$, such that $\{C_1^Y, \ldots C_m^Y\}$ forms a partition of the vertices in Y, if and only if, all the internal edges of the communities in Λ were internal edges in at least one of the layers.*

Proof. Let U_K be the set of vertices belonging to the metanode cluster Y, and let V_K be the union of the vertices belonging to the communities $\{C_1^Y, \ldots C_m^Y\}$. Similar to 6.5.1 for the if direction of the statement it is sufficient to prove that $V_K = U_K$, or $U_K \subseteq V_K$ and $V_K \subseteq U_K$.

$U_K \subseteq V_K$, can be easily obtained by selecting the communities to form V_K such that all vertices of U_K are included. To prove $V_K \subseteq U_K$ by contradiction, we assume that there exists a vertex $u \in V_K$, that is not in U_K, i.e. none of the metanodes, that form the metanode cluster Y contains u. As per our construction of the metagraph, this means that u is connected to at least one vertex in V_K by bridge edges (or not connected at all). Thus at least one of the communities has an internal edge that was bridge edge in all the layers. This goes against our criteria that all internal edges for communities in the composed network, should be internal in *at least one* of the layers. Thus our assumption is wrong and $V_K \subseteq U_K$.

Since the communities are disjoint and $U_K = V_K$, thus the statement is proven.

To prove the only if direction, we show that if the communities in the OR-composed layers have internal edges that were bridge edges in all the layers, then there may not exist a set of communities that form a partition for the vertices in a given metanode clusters.

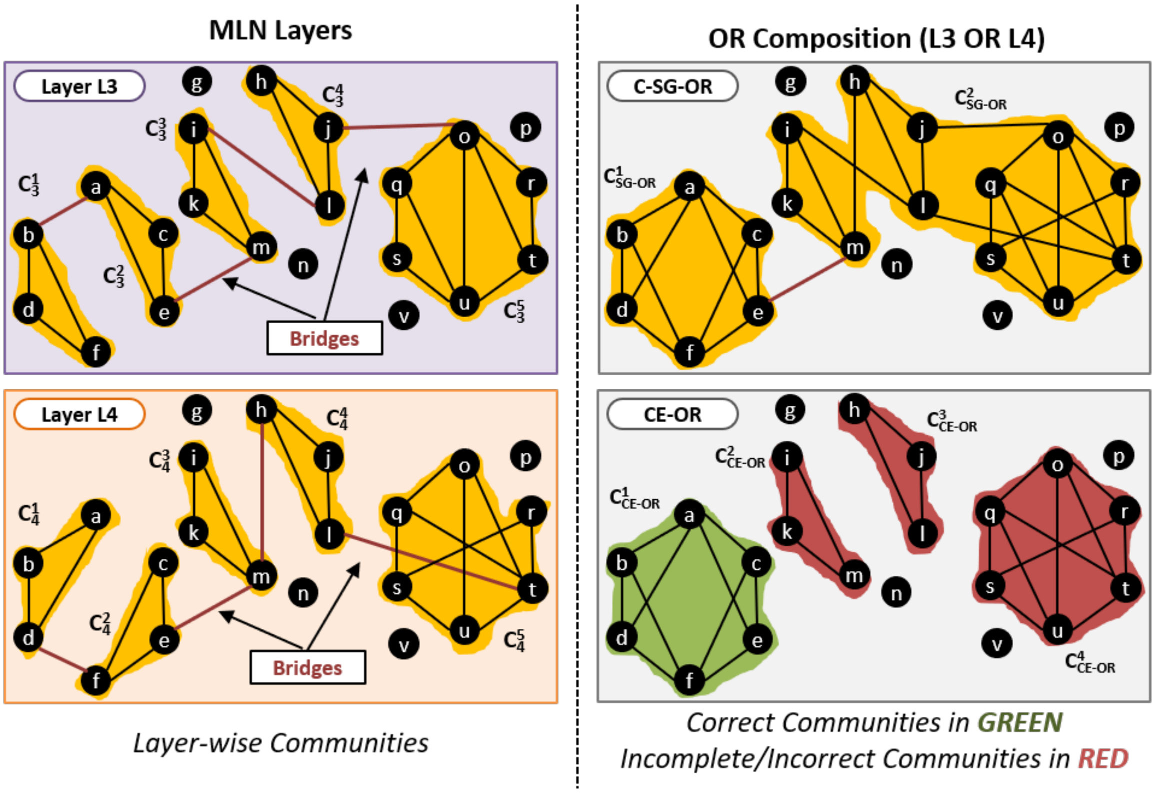


Figure 6.8: Effect of bridge edges on OR Composition

An example of this is given in Figure 6.8. The left-hand of panels show two layers of the network. The top right panel shows the communities obtained by the standard single network approach (C-SG-OR). The bottom right panel shows the communities obtained by our proposed CE-OR method.

Consider the community C^2_{SG-OR} generated by C-SG-OR approach that has edges (i, l), (h, m), (j, o) and (l, t) which are not internal edges in any of the layers, and are present as bridge edges in only one of the layers. These edges will not be part of the metagraph and thus CE-OR does not know that they exist. CE-OR, thus, generates three communities C^2_{CE-OR}, C^3_{CE-OR} and C^4_{CE-OR}, instead of merging them into one community, as per the C-SG-OR method.

However in the community C^1_{SG-OR} generated by C-SG-OR, the edges (a, b) and (d, f) are bridge edges in one layer but are intra-community edges in another layer. Therefore these edges will be part of the metagraph. Thus CE-OR can use these edges and correctly generate the community C^1_{CE-OR}. □

Implications and Limitations The implication of Lemma 6.5.1 and Lemma 6.6.1 is that the CE-AND or CE-OR operations are successful if they create clusters, such that one or more communities in the composed networks, completely cover the cluster. This means that we can divide the communities into groups, such that each group can be mapped to exactly one cluster formed by the CE-AND or CE-OR operation. Going further this means that we can partition the composed network into subgraphs, each subgraph relating to a cluster. Hence CE-AND and CE-OR operations are successful when each layer is formed of several loosely connected subgraphs, and bridges connecting the subgraphs do not change too much across the layers.

The primary limitations of our CE-AND and CE-OR algorithms is due to the non-inclusion of bridge edges. In the AND-composed network, we rationalize this non-inclusion by positing that communities formed solely of bridge edges cannot be dense, and hence are not strong communities. In the OR-composed network, note that we only exclude an edge if it is a bridge edge in *all* the layers. This is an infrequent case where bridge nodes from all layers come together to form communities. Our experiments in Section 12.2, justify this policy of not including bridge edges by demonstrating that the normalized mutual information (NMI) values between the communities returned by CE-AND and C-SG-AND are in general high.

6.7 Empirical Results

In this section we compare the performance and accuracy of our proposed algorithms with the ground truth results obtained by the standard methods, C-SG-AND and C-SG-OR.

6.7.1 Experimental Setup

Since the results of community detection depend heavily on the type of algorithm used [98], to control this parameter in the experiment we use the popular community detection algorithm *Infomap* [66], both to find the communities in the single network approach and the network decoupling approach. Our algorithms were implemented in C++ and were executed on a Linux machine with 8 GB RAM and installed with UBUNTU 16.10.

Datasets Used. We performed our experiments on multiplexes created from three real-world datasets and one synthetic dataset created using the RMAT [99] graph generator. We selected the real-world datasets such that they were sufficiently large and contained communities. To test on larger networks with more vertices, we created the synthetic RMAT dataset. The details of the datasets are as follows (also see Table 6.2);

- ***IMDb:*** From the IMDB dataset [94], we created the following three layers in the multiplex, with the nodes representing the actors. In the first layer, (L1, co-acting) two nodes are connected if they co-acted in at least one movie. In the second layer, (L2, rating) two nodes are connected if the average ratings of the movies where they acted were similar. In the third layer, (L3, genre) two nodes are connected if they acted in movies of similar genres. For every actor a vector was generated with the number of movies for each genre he/she has acted in. Two actors are connected if the Pearson's Coefficient between their corresponding genre vectors is at least 0.9[†].

- ***DBLP:*** From the DBLP dataset of academic publications [95], we selected all papers published from 2000-2018 in top three conferences VLDB (L1), SIGMOD(L2) and ICDM (L3). The nodes were the authors. Two authors in each layer were connected if they had published a paper in the conference corresponding to the layer.
- ***Accident:*** From the dataset of road accidents that occurred in the United Kingdom in 2014 [100], we represented each accident as a node. Two nodes are connected in a layer if they occurred within 10 miles of each other and have similar Light (L1), Weather (L2) or Road Surface Conditions (L3).
- ***RMAT:*** The RMAT generator creates networks based on the Kronecker product of a matrix. We set the number of vertices to 2^{15} and the edges to roughly eight times the number of vertices. We set the probabilities in each quadrant of the matrix as a=0.65, b=c=d=0.15 to create a scale-free graph.

  The first layer (L1) was the graph obtained by the generator. We applied cross perturbation to the other layers. That is we selected two edges (a, b) and (c, d), and replaced them with new edges (a, c) and (b, d). Thus the number of edges remain the same, but the degree distribution and the structure of the graph changes. In layer L2 we applied perturbation to 1% of the edges and in layer L3 to 5% of the edges.

Ground Truth and Accuracy Metrics: Since our goal is to achieve the results obtained by the standard C-SG-AND and C-SG-OR methods, we use the communities obtained from these methods as the ground truth. We disregard communities of just

[†] The choice of the threshold is based on how actors are weighted against the genres. We have To chosen 0.9 for connecting actors in their top genres.

Name	Vertices	Edges in L1	Edges in L2	Edges in L3
IMDB	9,485	45,581	13,945,912	996,527
DBLP	17,204	5,831	17,737	12,986
Accident	5000	193,860	235,175	216,397
RMAT	32,768	230,445	230,445	230,445

Table 6.2: Summary of the sizes of the multiplexes.

one vertex, since these result due to an artifact of the algorithm rather than provide any meaningful analysis. We use two metrics to evaluate the accuracy of the communities - i) Normalized Mutual Information (NMI) that measures the quality with respect to the participating entity nodes only and ii) modified-NMI that also takes into account the topology of the communities. For both metrics higher is better, with maximum value of 1 and minimum of 0 (definitions in [101]).

Each multiplex has 3 layers. Thus, a total of 4 compositions are possible (3 for 2-layers and 1 3-layers). Thus we compare results for 8 (4 combinations X 2 Boolean operations) composed networks.

6.7.2 Accuracy of the Aggregation Algorithms.

For the AND-composed networks we show in Figure 6.9, the average NMI and m-NMI of all the four multiplexes with respect to the ground truth for the CV-AND and CE-AND methods. The results show that the *accuracy obtained with CE-AND is higher than that from CV-AND.*

For the OR-composed networks we show in Figure 6.10, the average NMI and m-NMI of all the four multiplexes with respect to the ground truth for the two weighting metrics; Aggregation (w_a) and Fractional (w_f). The results show that the *accuracy obtained using both the metrics are similar.*

Multiplex	L1, L2		L1, L3		L2, L3		L1, L2, L3	
	NMI	m-NMI	NMI	m-NMI	NMI	m-NMI	NMI	m-NMI
Accuracy Values using CE-AND								
IMDB	.97	.93	.98	.97	.88	.86	.99	.99
DBLP	.92	.84	.99	.96	.98	.96	.98	.95
Accident	.96	.98	.94	.98	.91	.96	.88	.95
RMAT	.92	.82	.90	.79	.90	.78	.90	.77
Accuracy Values using CE-OR using Fractional Weights								
IMDB	<.01	<.01	.97	.99	1	1	1	1
DBLP	.83	.79	.87	.80	.75	.60	.71	.56
Accident	.88	.93	.94	.98	.98	.99	.86	.93
RMAT	.74	.64	.76	.59	.75	.55	.73	.54

Table 6.3: Accuracy Values using CE-AND and CE-OR on the different compositions of the datasets.

In Table 6.3 we provide the accuracy values for all the different layer compositions with respect to CE-AND for the AND composition and CE-OR with Fractional Weights. As can be seen nearly all the values are high, $\geq 70\%$.

Some low values occur for the CE-OR method. An egregious example is IMDb (L1, L2) for which the accuracy results are less than 1%! In this case the metagraph had 193 nodes, and on running the community detection algorithm 56 communities were obtained. However, the ground truth communities obtained by C-SG-OR had only 2 communities. This happened because there existed many bridge edges in the layers that were not included in the metagraph. Moreover, because the communities represented in the metanodes were small in size, the weights were also lower and could not combine the communities.

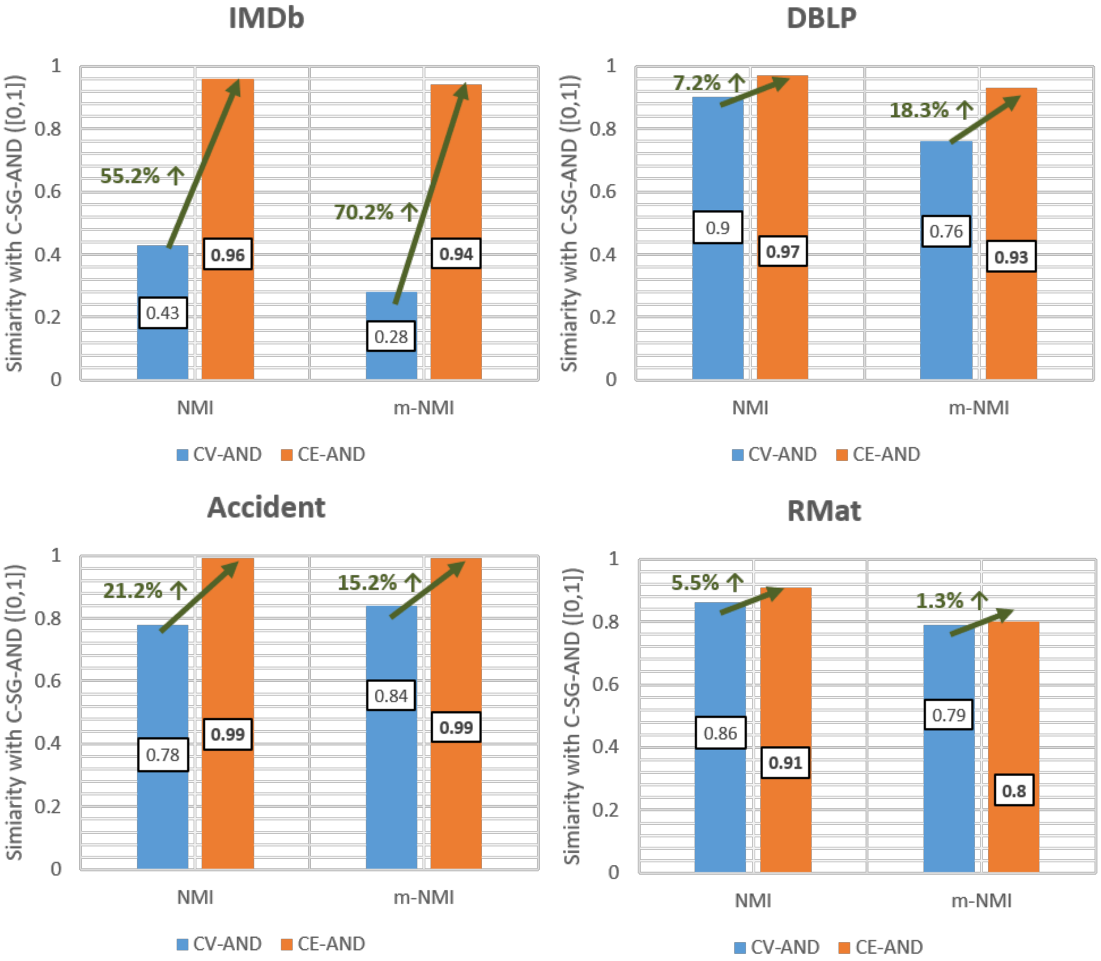


Figure 6.9: Comparison of Accuracy of CE-AND and CV-AND based on NMI and m-NMI.

6.7.3 Performance of the Aggregation Algorithms

6.7.3.1 Efficiency of the Decoupling Approach over the Single Graph Approach

We now compare the time taken to obtain the communities using the aggregation methods (CV-AND, CE-AND and CE-OR) with respect to C-SG-AND and C-SG-OR.

Figure 6.11 shows that the time to compute the communities over all the 4-composed layers is significantly lower for both CV-AND and CE-AND methods than

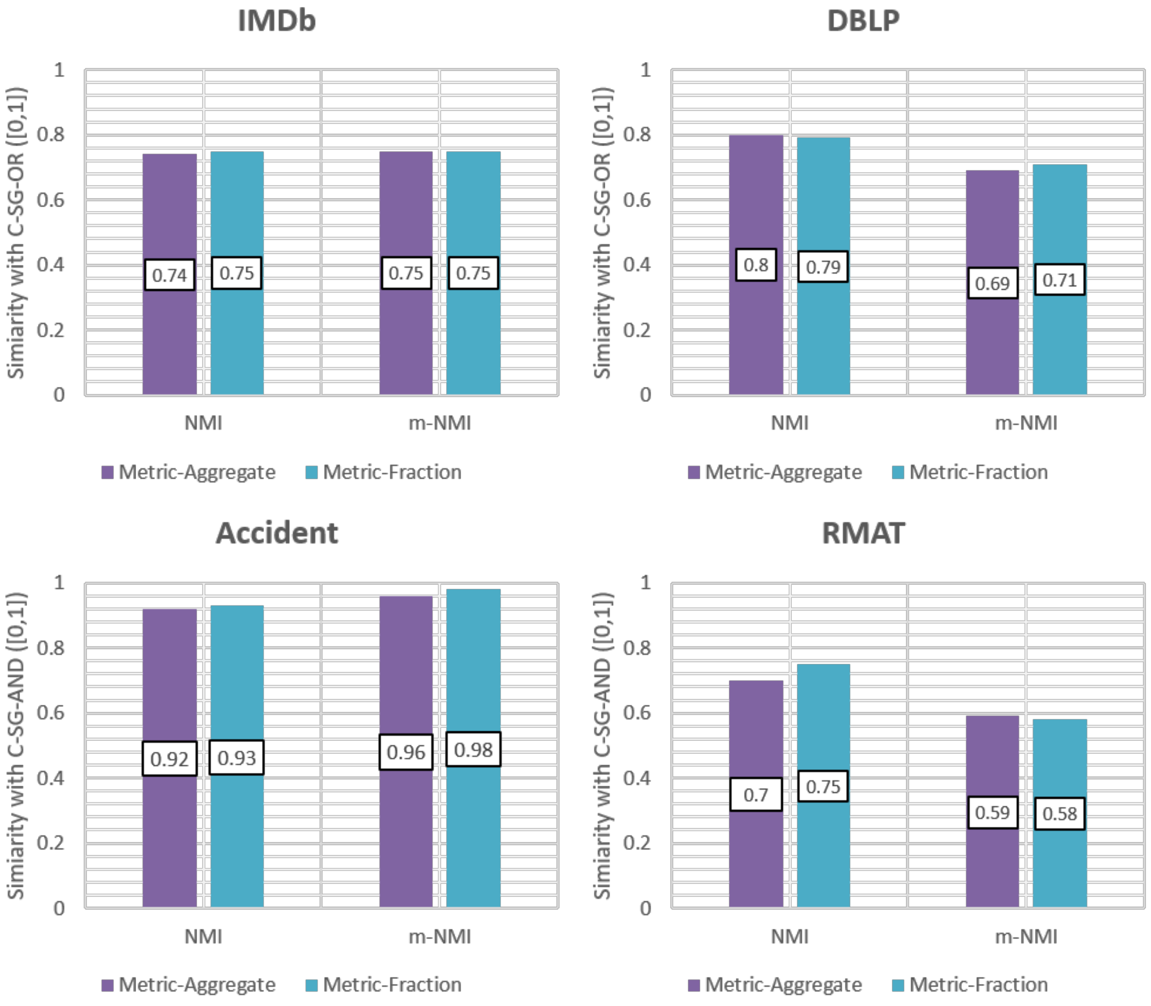


Figure 6.10: Accuracy of CE-OR with Different Weighting schemes based on NMI and m-NMI.

C-SG-AND. When the layers are sparse, CE-AND will be faster than CV-AND, as can be seen for DBLP multiplex. However if the network layers are dense, then the edge-based intersection approach of CE-AND has a higher cost as compared to the CV-AND.

Figure 6.12 gives the time for executing CE-OR. For CE-OR, CE-AND is used as a subroutine. One scan of *community edges* is required to generate the meta graph (OR-MG) on which we apply Infomap. If the layers are sparse and the multiplex contains

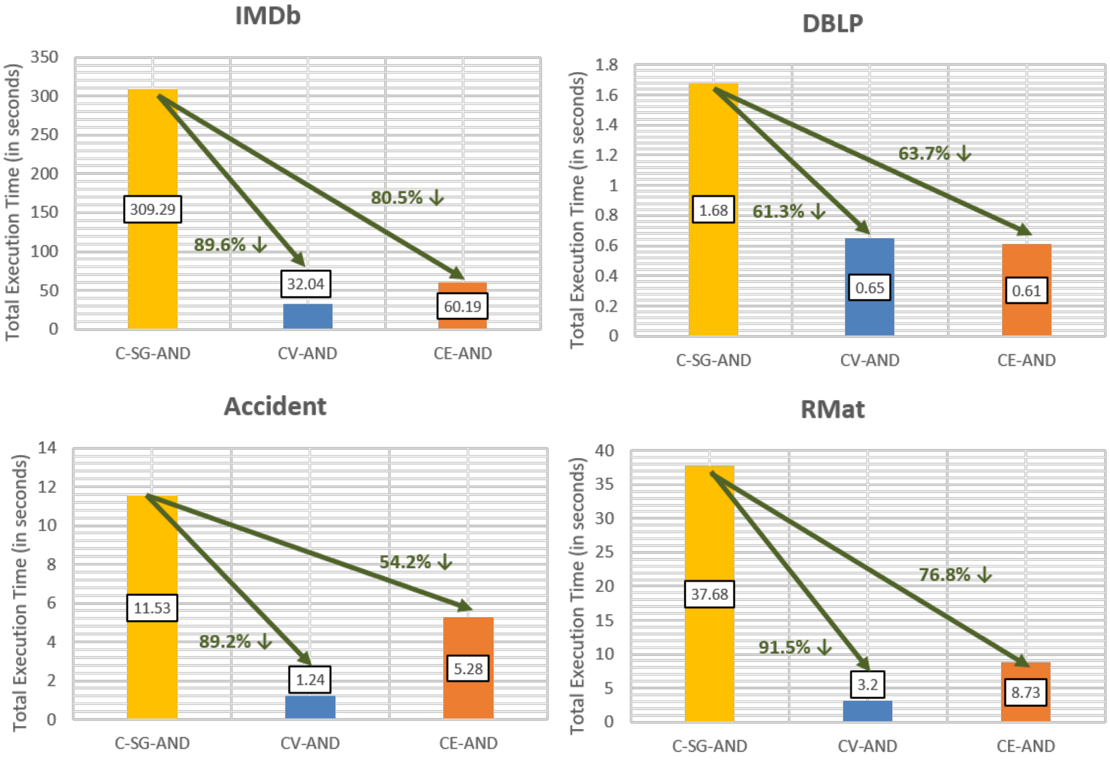


Figure 6.11: Efficiency of CV-AND and CE-AND as compared to C-SG-AND

many bridge nodes, then cost of generating the meta graph and applying Infomap will become an overhead as compared to simply applying Infomap on OR graph (C-SG-OR approach). This can be seen from the DBLP multiplex where sparse layers (density of densest layer (SIGMOD) = 0.0001!) make the CE-OR 67% less efficient as compared to C-SG-OR. However, *for multiplexes with fewer bridge edges (IMDb, Accident), CE-OR is significantly faster.*

6.7.3.2 Component Cost Analysis of the Decoupling Approaches

In general for decoupling based approaches, the cost of the decoupling approach is broken down into a) *one time cost* (that is the time to find layer-wise communities) and b) cost of combining the partial results. On one hand, the **one time cost** involves

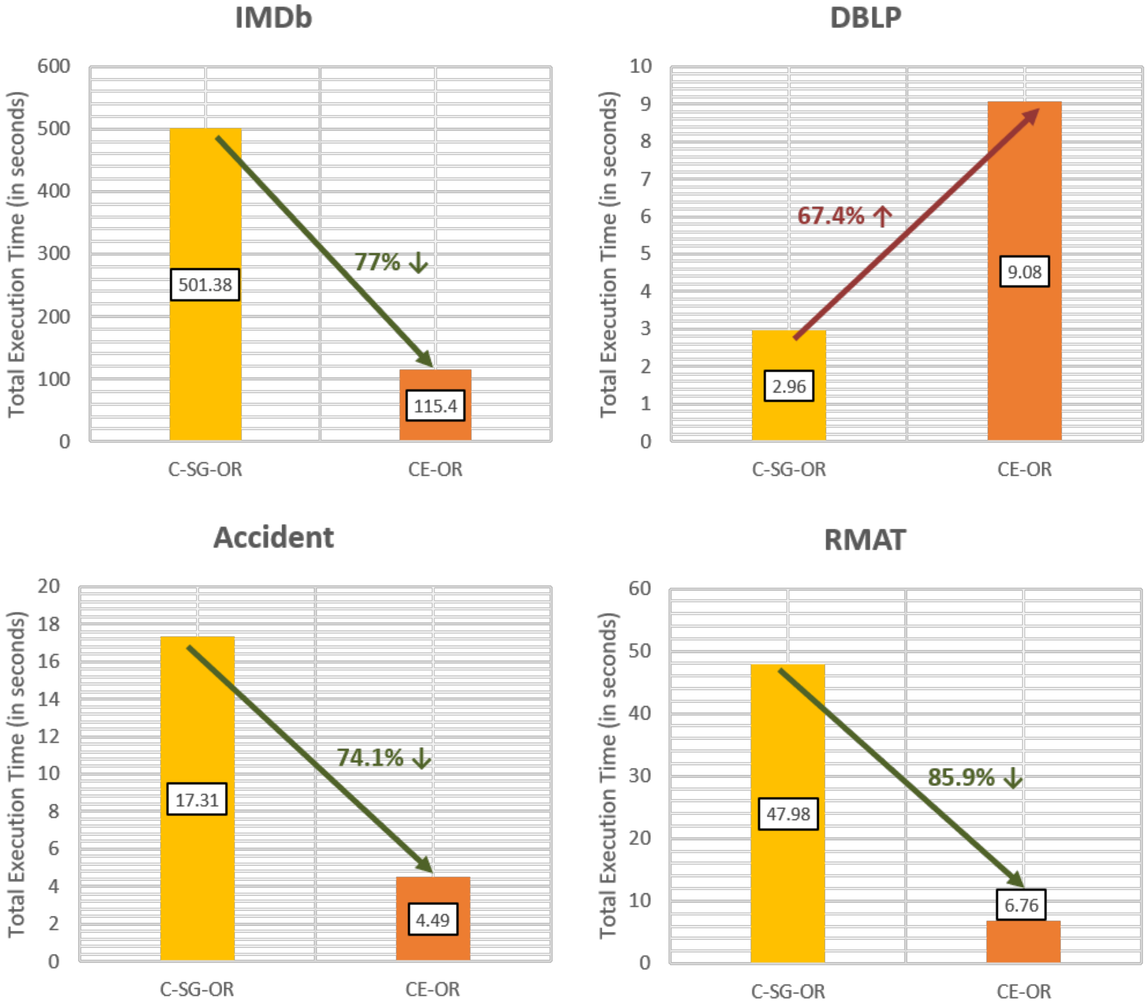


Figure 6.12: Efficiency of CE-OR as compared to C-SG-OR

the application of the existing algorithms on individual layers that detect communities by hierarchically optimizing the tightness metric (map function in the case of Infomap) through random walks across the networks. On the other hand, the **incremental cost of combining the partial** results involves one scan of community nodes/edges (CV-AND/CE-AND) or finding communities in the much smaller (in terms of number of nodes and edges) OR-metagraph (CE-OR).

Figure 6.13 and 6.14 show that for both the AND composition and OR composition approaches the **maximum cost of combining the partial results is significantly**

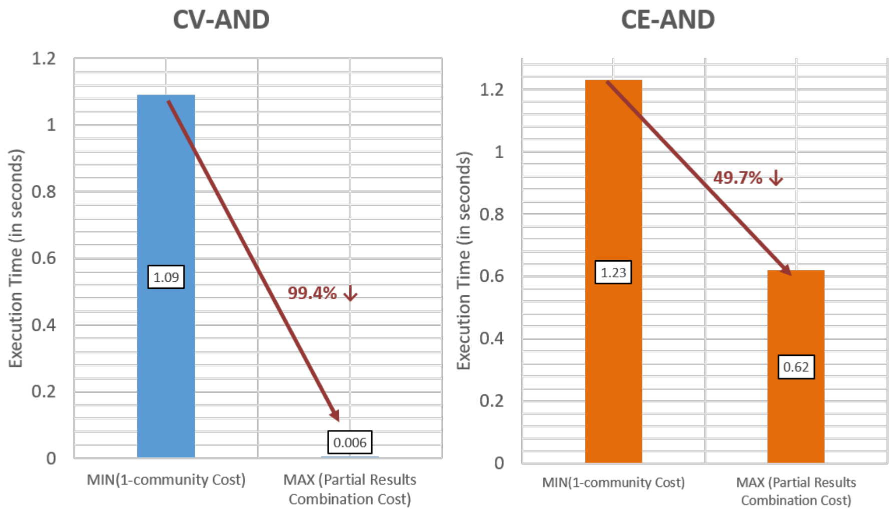


Figure 6.13: Component Cost Analysis of CV-AND and CE-AND (Accident HoMLN)

less than the minimum cost to detect 1 layer communities. The results shown here are for the Accident HoMLN. This empirically validates that the **additional incremental cost to combine the partial results is much smaller than the one time cost**. Thus, justifying the advantages of the decoupling approaches as compared to the single graph approach.

6.8 Effect of Different Parameters on the Composition Function Accuracy

It has been shown that the decoupling approach for AND/OR composition is efficient in general. However, it becomes pivotal to understand the effect of different network characteristics of the MLN layers on the accuracy of the composition algorithms. Depending on the accuracy estimates, one can choose whether to opt for the single graph approach or the decoupling approach. We have considered the modified-Normalized Mutual Information as the accuracy metric between CE-AND and C-SG-AND communities.

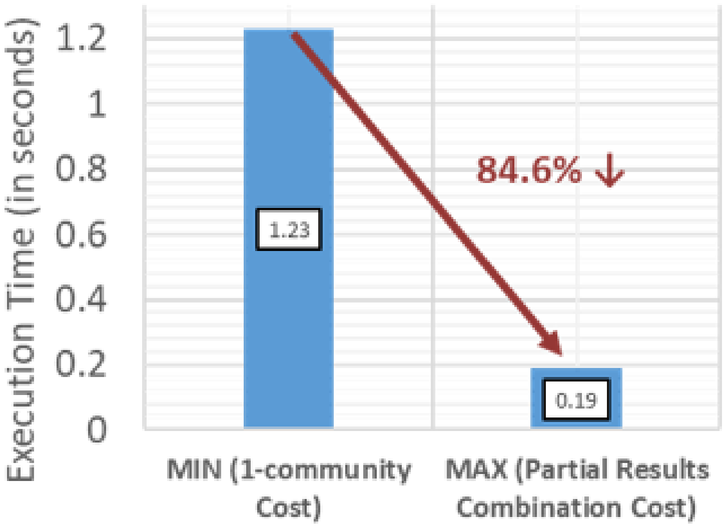


Figure 6.14: Component Cost Analysis of CE-OR (Accident HoMLN)

Synthetic Data Sets (HoMLNs): 2 sets of Synthetic HoMLN layers were generated for this purpose.

- **COMM-MLN-SET1**: 2 initial layers (L1, L2) were generated with 1000 vertices each. In L1, there were **5 cliques of 200 nodes each** and the other one had **10 cliques of 100 nodes each**. Nodes from these cliques were randomly connected by some number of bridge edges. 5% and 10% edges were removed from the two layers in every iteration, in order to generate 60 different pairs of layers.
- **COMM-MLN-SET2**: 2 initial layers (L1, L2) were generated with 1000 vertices each. In L1, there were **10 cliques of 100 nodes each** and the other one had **20 cliques of 50 nodes each**. Nodes from these cliques were randomly connected by some number of bridge edges. 5% and 10% edges were removed from the two layers in every iteration, in order to generate 60 different pairs of layers.

The different network characteristics that have been explored as a part this thesis are as follows:

- First, we aimed to understand the effect of graph characteristics of individual layers on the accuracy. To accomplish this task, for two layers generated in any iteration in the synthetic HoMLN, we analyzed
  - Number of Communities
  - The Average Community Density
  - Number of Bridge Edges
  - Average Clustering Coefficient
- Similarly, to understand the effect of composed layer graph characteristic, we chose to analyze
  - The average clustering coefficient of the AND/OR composed layer
  - The number of bridges in the AND/OR composed layer
- Finally, it was important to understand how two similar layers in terms of detected communities effect the overall accuracy. Few parameters considered were:
  - NMI and m-NMI between the community sets of the two layers
  - Cosine Similarity, Pearson's Correlation, Simple Matching Coefficient, Jaccard Index and Euclidean Distance between the clustering coefficient vectors for vertices from the two layers. The binary clustering coefficient vectors were also considered where any vertex that has a clustering coefficient higher than the average has a value of 1, else it is 0. This way nodes with high and low clustering coefficients are differentiated.

Here we discuss a few conclusive results for the AND and OR composition approaches

6.8.1 Variation in CE-AND Accuracy

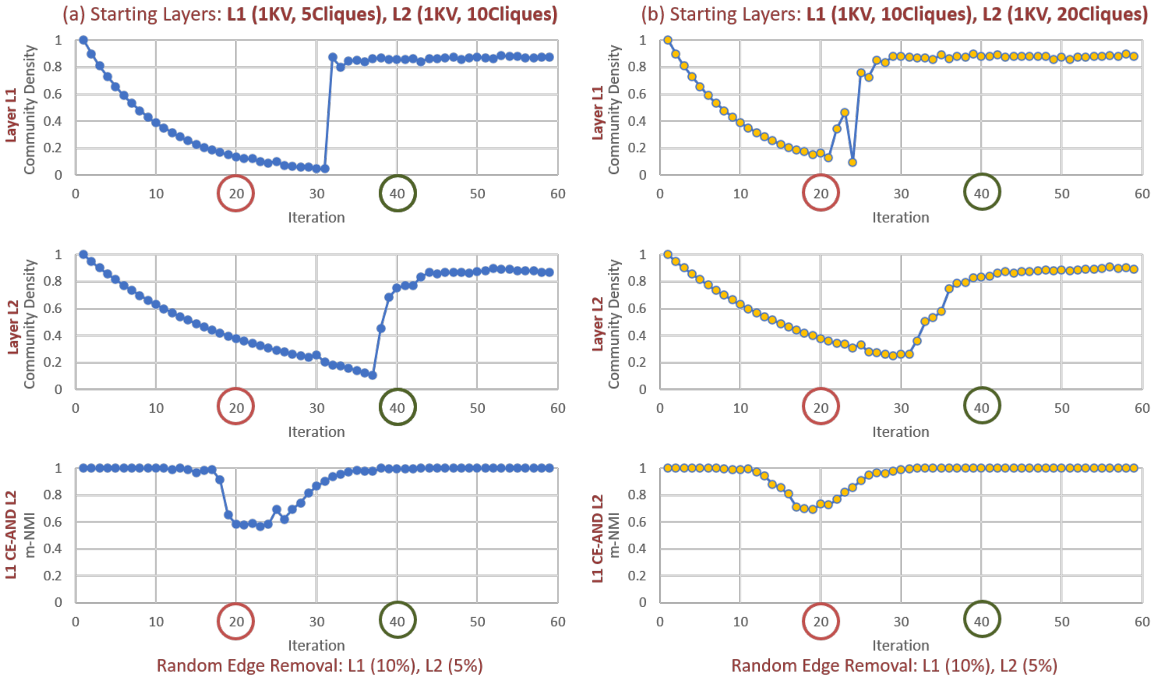


Figure 6.15: Effect of Layer-wise Average Community Density on CE-AND Accuracy

Effect of Layer-wise Average Community Density: 6.15 (a) and (b) show how the average community densities effect the CE-AND accuracy. The experiments validate the fact that **whenever the layers have well-formed / self-preserving / clique communities then the CE-AND accurately detects the communities due to the presence of common dense substructures across layers** (Iteration >= 40: Higher the Density, Higher the m-NMI). Even the contrapositive is true, where low m-NMI values correspond to the iterations where the layers did not have a well defined community structure (Iteration 20 to 30).

Effect of Bridge Edges in Composed layer: Figure 6.16 shows that **if the fraction of edges in the AND-composed network is high, then the accuracy of CE-AND drops**. The AND-composed network has more bridges when the *layer-*

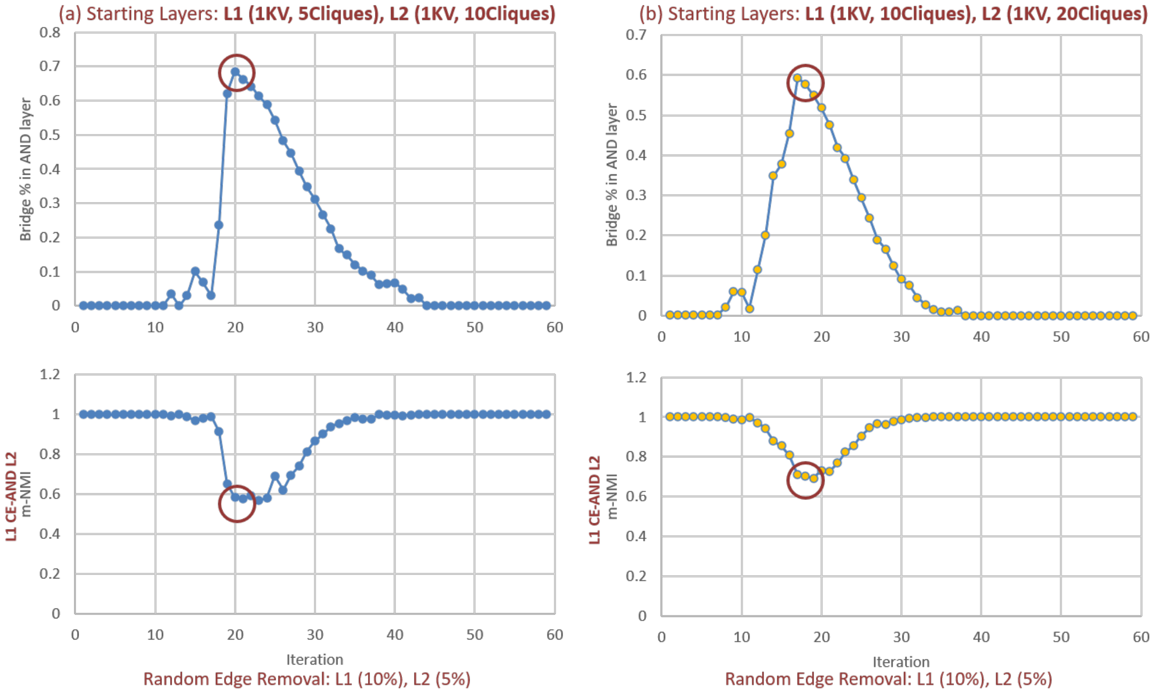


Figure 6.16: Effect of Bridge Edges in Composed Layer on CE-AND Accuracy

wise ill-formed/weak communities tend to split up and form smaller communities. *Due to the absence of dense common substructures across layers, the CE-AND will not work effectively.*

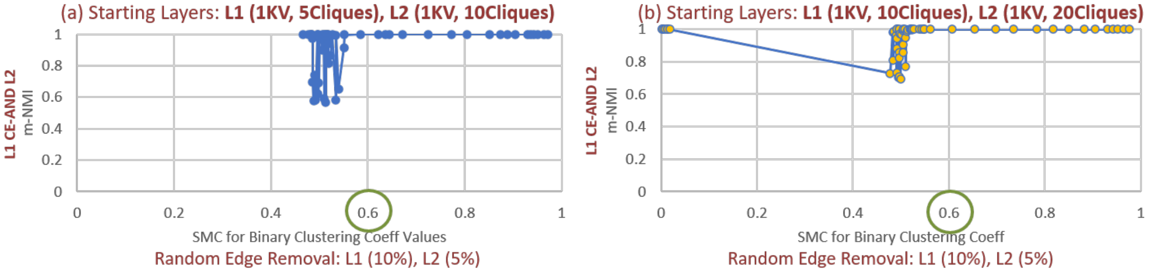


Figure 6.17: Effect of inter-layer Clustering Coefficient Similarity on CE-AND Accuracy

Effect of Clustering Coefficient Similarity: The Simple Matching Coefficient (SMC) between the binary clustering coefficient vectors finds out the fraction of nodes that are consistent with their clustering coefficient values in both layers. In general, if a node has a high clustering coefficient value, then it means that its neighborhood is tightly connected. A high SMC implies that a large fraction of high CC nodes (community core nodes) and low CC nodes (fringe nodes) are *same* across the layers. Thus, **CE-AND will be able to retain the most important nodes in the correct communities, if the SMC between the binary clustering coefficient vectors for vertices from two layers is high**. This fact is validated from Figure 6.17.

6.8.2 Variation in CE-OR Accuracy

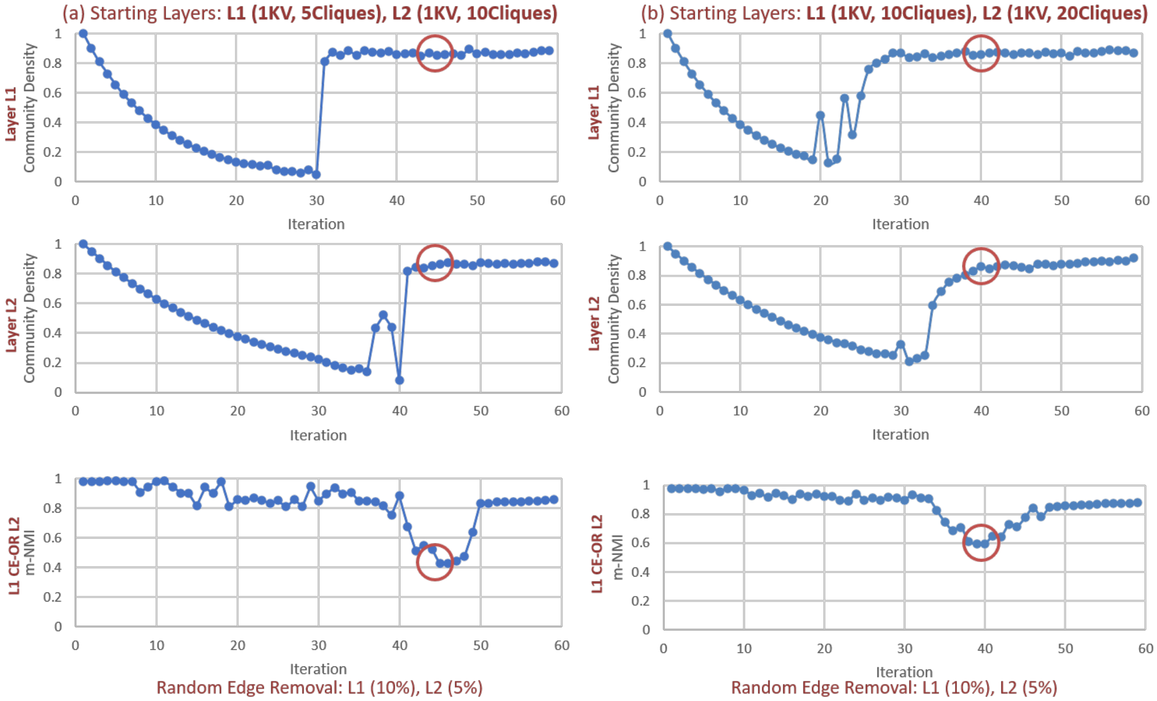


Figure 6.18: Effect of Layer-wise Average Community Density on CE-OR Accuracy

Effect of Layer-wise Average Community Density: It can be observed from Figure 6.18 that even when both the layers have dense communities (self-preserving/cliques), the accuracy of the CE-OR composition can be low (iteration 45 for Figure 6.18 (a) and Iteration 40 for Figure 6.18 (b)). This is because in case of OR-composition, multiple dense communities can get merged into a single large community due to the presence of *substantial number of common bridge edges*. Thus, **community structure alone cannot dictate the accuracy of the CE-OR composition and the role of the neighborhood is elevated**.

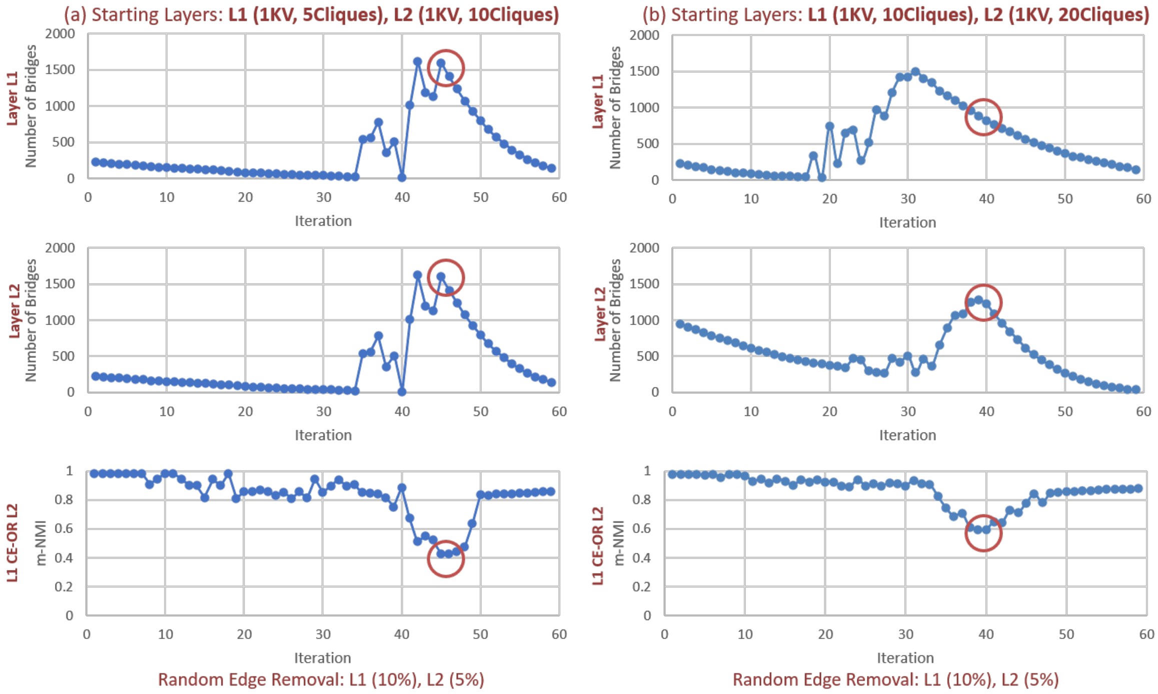


Figure 6.19: Effect of Bridge Edges on CE-OR Accuracy

Effect of Bridge Edges: The role of neighborhood that we discussed above can be validated from Figure 6.19. Here, it can be seen that in the first case, for iteration 45 the layers had dense communities (Figure 6.18 (a)) but the number of bridges were also high in both the layers (Figure 6.19 (a)). Similar scenario is observed for iteration 40 in the second synthetic HoMLN layers. Thus, in the C-SG-OR approach, *due to the presence of substantial number of bridges the dense communities got merged to form larger communities*. However, the CE-OR composition approach splits these kind of larger communities as it does not take into account the bridge edges. Therefore, it can be concluded that **when the layers have well-formed communities and the number of bridge edges is low, the the accuracy of CE-OR is high**.

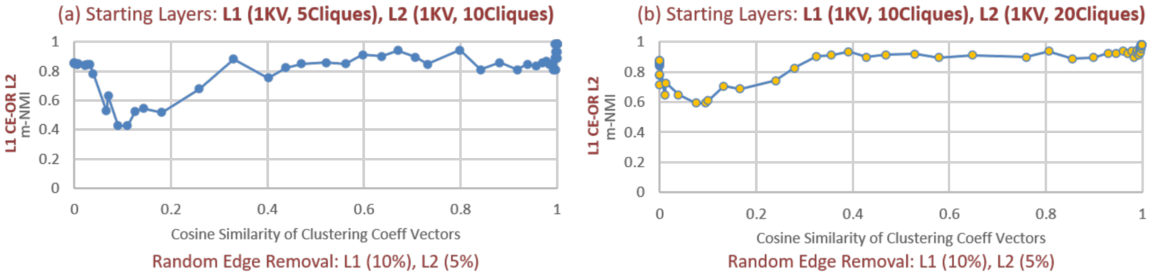


Figure 6.20: Effect of inter-layer Clustering Coefficient Similarity on CE-OR Accuracy

Effect of Clustering Coefficient Similarity: We find out the cosine similarity between the clustering coefficient vectors for vertices from the two layers to study the similarity of community structure between the two layers considered for composition. Higher value of cosine similarity means that the community core (high CC nodes) is not only same in both the layers, but also have similar strength (CC values). Moreover, even the fringe nodes (low CC nodes) are mostly same with same level of weakness (low CC value). Thus, in such a scenario due to the presence of *dense common neighborhood*, the effect of bridges is reduced in the OR composition and thus CE-OR has a high accuracy. Even the experiments validate this as it can be observed from Figure 6.20 that **High Cosine Similarity between the clustering coefficient vectors implies high accuracy for the CE-OR approach**. Even the contrapositive is true, where **low accuracy values (m-NMI) are observed for low similarity**.

6.9 Boolean Expression Evaluation

In this section, we provide the definition of the NOT composition of a multiplex layer in order to express analysis objectives using Boolean operators. Then, we discuss the mapping of real-world analysis to Boolean expressions consisting AND, OR, and NOT. As we have shown composition and efficiency for individual operators above, this will allow us to show empirically the accuracy and efficiency of the decoupled approach for an arbitrary analysis expression.

6.9.1 NOT Composition

NOT of a layer will represent the *complement of the edge set* of that layer, i.e. the new layer will have all those edges that are *not* part of the original layer. Communities in a NOT layer will represent the groups of nodes that are *not strongly connected*. Examples of queries that can be answered using NOT are

- Groups of actors who have *not acted together* in a comedy (IMDb multiplex)
- Groups of authors who have *never co-authored* a paper in VLDB (DBLP multiplex)
- Groups of accidents that *did not have same Light condition* (Accident multiplex)

With respect to the single graph approach, the above types of analysis can be handled by first generating the NOT layer and then applying community detection. Further, AND, OR and NOT together can be applied in different combinations, expanding the spectrum of analyzing a given multiplex.

As a unary operator, NOT gets composed using the previously discussed AND and OR operators. Although it may appear that taking the complement of a layer is expensive and increases the number of edges in that layer, it is important to remember that it depends on the graph density of the layer. Also, rewrites of expressions using the De Morgan's law may be used to obviate this increase in the number of edges where possible.

6.9.2 General Boolean Expression Evaluation: Accuracy, Efficiency and Drill Down Analysis

In this section, we discuss how general Boolean expressions can be computed using the decoupling approach. We use the DBLP multiplex with authors who publish papers in different conferences to address interesting analysis objectives. In order to make the analysis more interesting, we consider all papers that were published from 2003 to 2007 (5 years) in two high ranked conferences (VLDB and SIGMOD) and two medium ranked conferences (DASFAA and DaWaK). Thus, based on whether two authors (nodes) have co-authored a paper in a particular conference, four layers were generated - VLDB (L1), SIGMOD (L2), DASFAA (L3) and DaWaK (L4). Table 6.4 shows the layer-wise statistics.

Layer	**Vertices**	**Edges**	**Communities**
VLDB	5116	3912	327
SIGMOD	5116	3303	254
DASFAA	5116	1519	229
DaWaK	5116	679	154

Table 6.4: DBLP multiplex used for Expression evaluation

Few interesting analysis objectives that can be computed on the DBLP multiplex using Boolean expressions are as follows:

- Which are collaboration groups who have published in *both the highly ranked conferences*, but have *never published in either of the medium ranked conferences*?
- Which co-author groups have only been able to publish in the low to medium rank conferences?
- Which author groups have published only in VLDB?

Based on the requirements of the analysis, it is important to figure out a) the multiplex layers required and b) the order in which the layers have to be composed using AND, OR, NOT. For the first analysis, "**Which are collaboration groups who have published in *both the highly ranked conferences*, but have *never published in either of the medium ranked conferences* ?**", we will compare the evaluation process for the traditional single graph approach and the proposed decoupling approach.

Single Graph Approach (SG): For the SG approach, the Boolean expression will correspond to -

SG: **COMM**((VLDB *AND* SIGMOD) *AND NOT* (DASFAA *OR* DaWaK))

This corresponds to first generating the required composed single graph and then applying the community detection algorithm in order to find the final set of communities. This acts as the ground truth as in the earlier experiments. Here, we have used Louvain [67] to find the communities in order to show the performance of our composition algorithms in presence of different community detection algorithms.

Decoupling Approach: In this case, the expression will correspond to

DEC1: **COMM**(VLDB) *CE-AND* **COMM**(SIGMOD) *CE-AND* **COMM** (*NOT* (DASFAA *OR* DaWaK))

That is, the layer-wise communities are composed in the specified order to obtain the final set of communities.

Alternatively, De Morgan's Laws can be applied to obtain another expression for the decoupling based boolean composition -

DEC2: **COMM**(VLDB) *CE-AND* **COMM**(SIGMOD) *CE-AND* **COMM** (*NOT*(DASFAA)) *CE-AND* **COMM**(*NOT* (DaWaK))

We will compute both DEC1 and DEC2 to compare their efficiency and compare both with the single graph approach. Here, we will evaluate the expression, *NOT* (DASFAA *OR* DaWaK), using the traditional OR of two layers and then take its complement. Whereas DEC2 uses the decoupling approach using operators CE-AND, NOT as discussed.

Note that the layers of DBLP used above are *very sparse*, especially DASFAA and DaWaK. Hence, we can infer that DEC2 will not be as efficient as DEC1 since it has to compute the complement of two layers (resulting in dense graphs) and then apply the decoupling approach. DEC1, on the other hand, has only one complement to compute. Both will compute the same set of results.

Accuracy Results: For accuracy, the NMI and m-NMI values for the communities obtained by DEC1 and DEC2 have been compared against the communities obtained by SG. It can be clearly observed from Figure 6.21 that both the expressions lead to *DEC1 and DEC2 providing more than 95% accuracy*.

Performance Results: Both DEC1 and DEC2 resulted in the *same set of communities*. In DEC1, the number of CE-AND compositions are 2 whereas in DEC2 there are 3. Moreover, as the layers of the DBLP multiplex are sparse, their complement is dense. Thus, in DEC2 the Louvain is applied to two dense NOT layers. Thus, it can be conjectured that the DEC2 will have a higher cost as compared to DEC1. This has been empirically shown in Figure 6.22 where *DEC1 is approx. 2 times faster than DEC2*. Therefore, it is very important to understand when to rewrite the expression (using De Morgans, Distribution, etc.) especially when the NOT operator

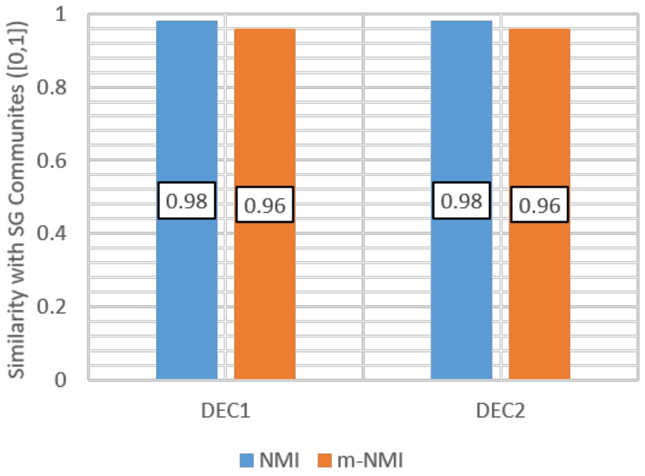


Figure 6.21: Accuracy results for communities obtained by decoupling-based expressions DEC1 and DEC2 as compared to Single Graph approach

is used on a composition of layers. *Finally, it is interesting to note that even with 2 dense graphs, DEC2 comes out better than the single graph approach. This further validates our decoupling approach even in the presence of NOT operator.*

Drill-Down Analysis: 102 communities are obtained from DEC1 and DEC2 that satisfy the requirement. Figure 6.23 shows few well-known groups most of whose members had collaborated on a paper that was published in both VLDB and SIGMOD, but never in DASFAA or DaWaK in the period from 2003 to 2007.

There is a high probability that the work done by these groups is not only of greater quality but also widely accepted. Validity of this claim can be made from the following facts,

- Figure 6.23 (a) community has researchers like **Surajit Chaudhari** who won the **VLDB 10-Year Best Paper Award (2007)** with **Vivek Narasayya** and **VLDB Best Paper Award (2008)** with **Nicolas Bruno**, apart from winning **ACM SIGMOD Contributions Award (2004)**.

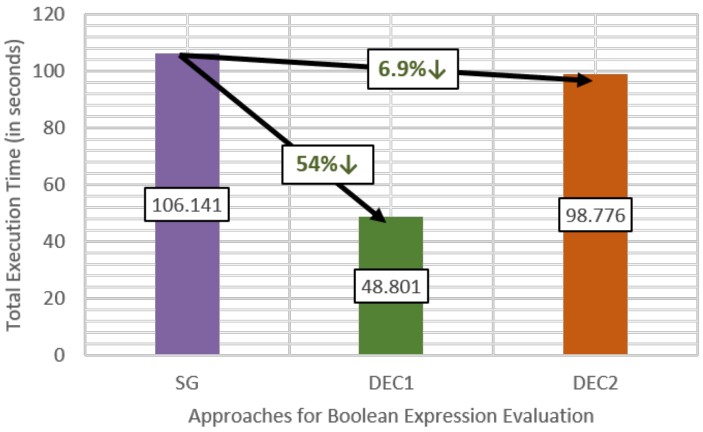


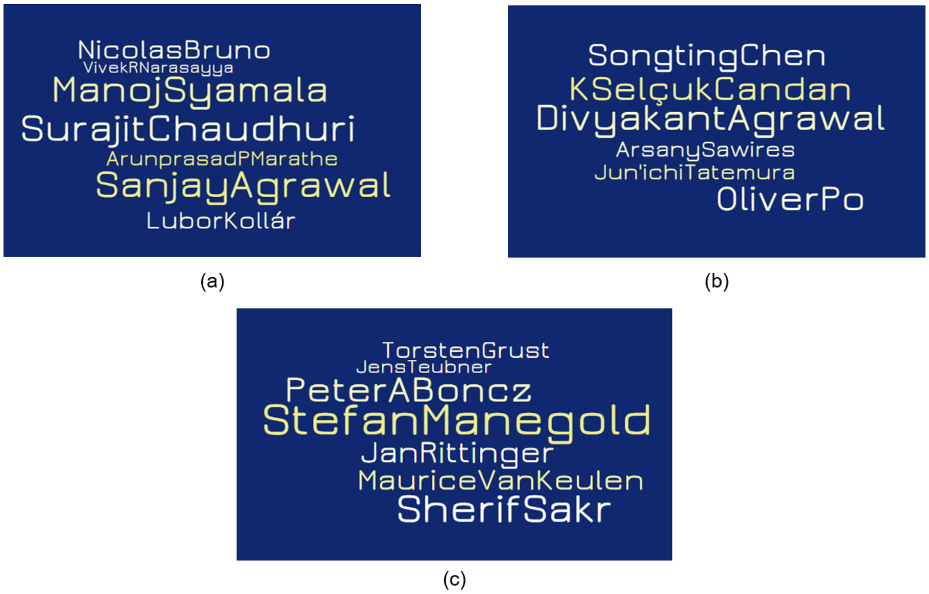
Figure 6.22: Efficiency results for communities obtained by decoupling-based expressions DEC1 and DEC2 as compared to Single Graph approach

- Figure 6.23 (b) has researchers like **Divyakant Agrawal** who has **24000+ citations** (Google scholar).
- **Peter A. Boncz and Stefan Manegold** from Figure 6.23 (c) group not only published a **highly cited paper** (350+ citations for MonetDB/XQuery) in SIGMOD 2006, but also have won the **VLDB 10-year award**.

6.10 Conclusions and Future Work

In this chapter, we presented algorithms for efficiently finding communities in Boolean composed layers of homogeneous multilayer networks. The results show that for most cases our algorithms are significantly faster than the standard methods and produce results of similar quality. The only cases that our algorithm fails is when the layers have significantly more bridge edges.

In future plan, some percentage of bridge edges can be included in the composition process without increasing the computation time. It should be possible to explore adaptive techniques that can select between the network decoupling and standard methods as suitable.



(a)

(b)

(c)

Figure 6.23: Drill-Down Analysis: Prominent Author Groups

The next two chapters, 7 and 8, provide case studies, where the proposed community detection in HoMLNs has been used for the aggregate analysis of the Facebook and Movie Actors data set.

CHAPTER 7

HOLISTIC SOCIAL NETWORK ANALYTICS: LEVERAGING CONTENT ANALYSIS AND MULTILAYER NETWORKS

In this chapter, our goal is to adapt the MLN analysis approach to efficiently and flexibly analyze social network data using explicit (known or given) as well as implicit (derived or extracted) features of the datasets. It is imperative that these datasets be analyzable in a flexible manner as different features have different impacts and importance on the information that can be inferred. For example, for advertising in social networks, influential communities are sought (based on age, gender, friends, interests, political views etc.). For quick propagation of information centrality nodes may be useful.

This work is the first one, to the best of our knowledge, to apply this approach for the analysis of *one of the largest/densest real-world social network data collection*, although it has been used in several experimental studies on smaller/sparser datasets [89, 29].

The contributions of this chapter include: **(1)** using the novel, emerging MLN approach for flexible analysis of a large complex real-world dataset, **(2)** integrating content analysis seamlessly with structural network analysis, and **(3)** extensive analysis and result validation for the social network work datasets.

The remainder of the chapter is organized as follows. Section 11.2 states the general problem and proposed approach. Section 7.2 elaborates modeling and computation aspects of our approach. Section 7.3 details the use of content analysis to integrate into multilayer network approach. Section 7.4 showcases analytical results of queries.

Section 7.5 discusses computational advantage of the adapted approach. Section 12.5 concludes the chapter.

7.1 Dataset Overview, Analytical Queries, and Problem Statement

7.1.1 The Facebook Dataset

The Facebook (FB) data collection fulfill all the characteristics of a complex dataset in terms of a large number of features, content that could be analyzed and the requirement for analysis using combinations of features.

Table 7.1: Statistics of four datasets

Datasets	#users
Demographic Info (D1)	2,676
User's Political Views (D2)	2,695
Personality (D3)	2,485
Facebook Status Updates (D4)	1,645

This data collection from *myPersonality* project [102] is one of the largest and well-known real-world social network research data collection, where the volunteers took real psychometric tests and opted in to share data from their FB profile (period of 2007-2012). The experimental data contains four datasets: Demographic Info (D1), User's Political Views (D2), Personality (D3), and FB Status Updates (D4). We have a total of 260K individuals in the datasets predominantly from USA. Of those, about 2.6K have more common features as compared to others. Hence, as shown in Table 7.1, we have chosen the 2.6K subset for detailed analysis. The dataset includes a number of features, such as age, gender, location, self-assigned political views, relationship status, personality trait, etc. along with access to individual's status updates from which additional features can be extracted using content-based analysis.

We use four features from dataset D1 including age, gender, relationship status, and locale. One self-declared political-view feature from D2. D3 provides five features which are the five personality traits of the Five-Factor Model (*FFM*) [103]. *FFM* is considered the most influential and standard model for personality trait prediction in psychology over the last 50 years. Based on D3 together with D4, one more feature of people's privacy-concern is inferred. For better understanding in later sections, we especially introduce the well-known five personality traits [104], which are defined as openness to experience, conscientiousness, extraversion, agreeableness, and neuroticism.

1. Openness (OPN) to experience: intellectual, insightful vs. shallow
2. Conscientiousness (CON): self-disciplined, organized vs. careless
3. Extraversion (EXT): sociable, playful vs. aloof, shy
4. Agreeableness (AGR): friendly, cooperative vs. antagonistic, faultfinding
5. Neuroticism (NEU): insecure, anxious vs. calm, unemotional

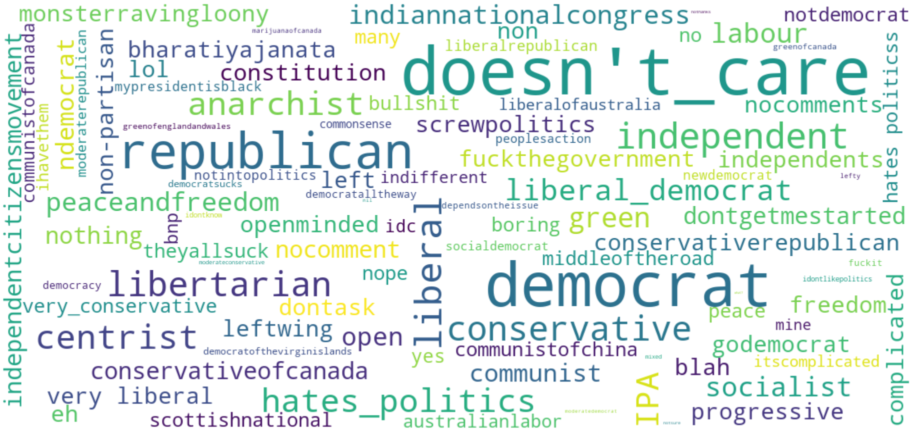


Figure 7.1: Word clouds of political views where bigger cloud representing more dominant political views in the dataset.

7.1.2 Analytical Queries

The focus of this chapter is to demonstrate modeling, flexible and efficient analysis of social network datasets to infer trends and establish correlations, and possible causality. A few analytical queries that are meaningful for this dataset are shown below.

(Q1) Dominant Political Views: How the user-declared political view (e.g., democrat, doesn't care, republican) varies across age groups in the dataset? (There are 100 political views in Facebook dataset used in this chapter - see Figure 7.1).

(Q2) Relationship Status Correlation:

(a) With respect to age groups, how does relationship status (e.g., single, in a relationship, and married) vary?

(b) How do the relationship statuses affect the personality traits of an individual? Does it differ based on gender?

(Q3) Personality Trait Analysis:

(a) How much of the population demonstrate contrasting personality traits (e.g., OPN and NEU)?

(b) How do the personality traits evolve with age? For example, which age group of people deals better with stress?

(Q4) Privacy Concern Correlation:

(a) How does the individuals' age correlate with their comfort level of sharing personal information on social media?

(b) Do personality traits have a bearing on the level of privacy-concern?

7.1.3 Problem Statement

Currently, the above analytical queries are done using a graph-based approach where *a single graph needs to be created for each analysis based on the involved fea-*

tures. Typically, nodes represent entities (i.e., people in our case) and edges represent relationship between nodes based on feature values (e.g., same age group, the same relationship status). These graphs are analyzed using graph metrics such as community, hubs or centrality nodes, and so on. This approach entails the creation of a customized graph for each query using the features involved which can lead to an exponential number of graphs in the worst case. For the above mentioned analytical queries (Q1-Q4), multiple graphs, the number of which depends on the number of features involved, need to be created, stored, and analyzed for each query.

Using the Homogeneous MLN approach (same set of entities have to be analysed), *for a given dataset with M features (whether explicit or derived), determine the layers (nodes and intra-layer edges) based on the analyses requirements of the dataset and use composition for analysis using Boolean or other operations.* Thus, once MLN layers are created as shown, any number of analyses can be performed without generating additional graphs/layers. Necessitated by the queries, this chapter primarily uses *AND* compositions.

7.2 Data Modeling and Layer Composition Using MLN

7.2.1 Data Modeling

It is already explained in Section 11.2 that we have four Facebook datasets (D1-D4) and the corresponding features. *UserID* is common to all datasets to associate with each other as shown in Figure 7.2. *UserID* is also used to make the dataset anonymous to mask the privacy of an individual. Moreover, the social networks analysis demands the analysis of the same set of entities (people) with respect to different features. Thus, we will choose **Homogeneous Multilayer Networks** to model the FB dataset.

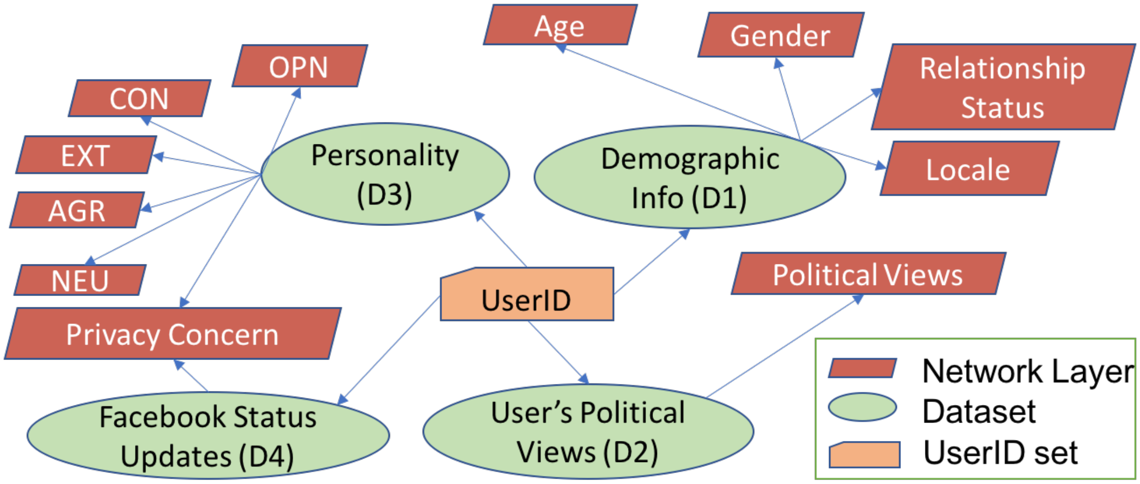


Figure 7.2: Modeling the FB Data Collection for Multilayer Network Analysis

Unlike other features from D1 and D2, the features of personality traits (OPN, CON, EXT, AGR, NEU) and privacy-concern are not explicitly present in the given dataset and will be derived through content analysis. The derived output for each personality trait would be binary label "Yes" or "No". And the derived output for privacy-concern of each FB user in our datasets is "high", "medium", or "low". This clearly indicates the power of the approach in absorbing derived content in the same way as an explicit feature. Different types of content extraction can be supported readily. Our approach to content extraction is presented in Section 7.3.

Table 7.2 shows the statistics regarding the generated multiplex layers along with the number of edges in each layer. The number of nodes in all layers is the same but the number of edges will be different since it depends on the available information for each feature. The 11 layers in Table 7.2 that are generated for the Facebook multilayer network correspond to the features in Figure 7.2. The semantics of the graphs in each layer are described as follows:

(L1) **Age**: Any two users are connected by an edge when they both fall into a same age group, namely [$\leq$ 20], [21-30], [31-40], [41-50], [51-60], and [$\geq$ 61].

Table 7.2: Statistics of 11 MLN layers

Layer	From dataset	# edges
L1: Age	D1	1,228,223
L2: Gender	D1	1,813,638
L3: Relationship Status	D1	1,119,592
L4: Political Views	D2	494,974
L5: Locale	D1	2,799,160
L6: OPN	D3	1,020,306
L7: CON	D3	840,456
L8: EXT	D3	795,691
L9: AGR	D3	718,201
L10: NEU	D3	627,760
L11: Privacy Concern	D3, D4	2,191,659

(L2) **Gender**: Any two users with the same gender are connected.

(L3) **Relationship Status**: Any two users with the same relationship status are connected by an edge.

(L4) **User-Defined Political Views**: Any two users with the same political view are connected by an edge.

(L5) **Locale**: Any two users with the same locale settings (e.g., en_UK, en_US) are connected by an edge.

(L6-L10) ***FFM* (i.e., OPN, CON, EXT, AGR, NEU)**: Each personality trait of the *FFM* forms one network layer. Any two users with the same type of personality trait are connected.

(L11) **Privacy Concern**: Any two users with the same privacy-concern level (i.e., high, medium, or low) are connected.

7.2.2 Metric Computation & Layer Composition

In the multilayer network described above, although each layer has the same nodes, their edge connectivity will vary according to the feature value distribution. For example, groups of people having the same political view may not be present in the same age group, groups of people with the same personality may have different levels of privacy-concern and so on. For detecting the tightly connected groups of people with respect to a particular feature, we compute *communities* in the corresponding layer by applying Infomap[66]. A community in a graph translates to a group of nodes that are more connected to each other than to other nodes/communities in the graph.

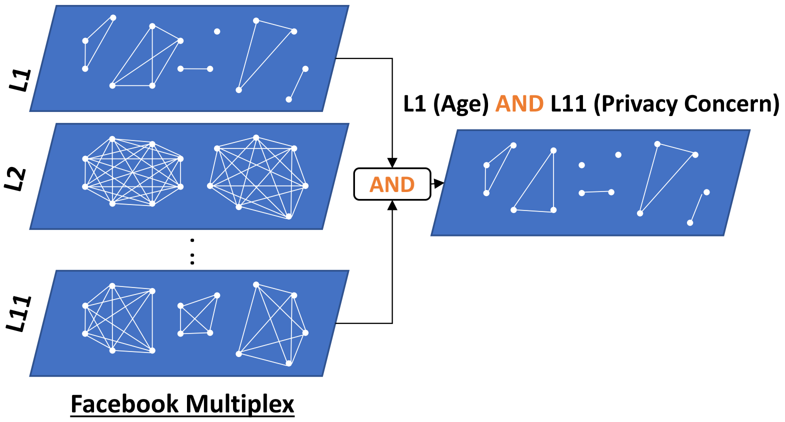


Figure 7.3: An example of AND-Composition

Analytical queries listed in Section 7.1.2, require commonality of information in *at least two multiplex layers*. This corresponds to the Boolean *AND* operation for composing layers. For example, in Q4a (Section 7.1.2), to analyze the effect of age on level of privacy-concern, we need to compute communities where an edge represents people who fall into the same age group **and** have the same privacy-concern level. Figure 7.3 shows a simple example with several small Facebook multiplex layers, and an ***AND*** composition of L1 (Age) and L11 (Privacy Concern) layers. Table 7.3 shows the *AND*-compositions whose *communities* have to be generated in order to perform the flexible analysis of the Facebook multilayer network for our analytical queries listed in Section 7.1.2.

Community Detection for *AND* Compositions: The traditional way to address any of the listed analytical queries is to first generate the required combined graph corresponding to the *AND*-composition and then detect the communities. The

Table 7.3: *AND*-Compositions needed for analytical queries shown in Section 7.1.2

Analysis	**Required *AND*-Compositions of Layers**
Dominant Political Views	
Q1	L1 (Age), L4 (Political View), L5 (Locale)
Relationship Status Correlation	
Q2a	L1 (Age), L3 (Relationship Status), L5 (Locale)
Q2b	Five 3-layer compositions: [L2 (Gender) AND L3 (Relationship Status)] with each of L6 (OPN), L7 (CON), L8 (EXT), L9 (AGR), L10 (NEU)
Personality Traits Analysis	
Q3a	L6 (OPN), L10 (NEU)
Q3b	Five 2-layer compositions: Each of L6 (OPN), L7 (CON), L8 (EXT), L9 (AGR), L10 (NEU) with L1 (Age)
Privacy Concern Correlation	
Q4a	L1 (Age), L11 (Privacy Concern)
Q4b	Five 3-layer compositions: [L2 (Gender) AND L11 (Privacy Concern)] with each of L6 (OPN), L7 (CON), L8 (EXT), L9 (AGR), L10 (NEU)

MLN approach pre-computes the communities of the individual layers. Based on the composition requirements of the analytical query, partial results (i.e., pre-computed communities) are intersected (for *AND*-composition) to generate the combined communities. It can be shown analytically [7] that both computationally and storage-wise, the MLN approach is more efficient. The requirement to apply composition is that the communities of the individual layers must be *self-preserving* in nature.

Checking the Self-Preserving Property: By definition, *"a community is self-preserving if the nodes in it are so tightly connected such that even if only a subset of connected nodes are chosen from that community, they will form a smaller community [7]."* For each of the 11 layers, we computed the internal clustering coefficients of each node, which is the ratio of the number of edges among the neighbors of the node *in the same community* to the total possible edges among those neighbors, and it was *one*. This indicated that the *communities in each layer were self-preserving.* In Section 7.4, we will discuss some of the inferences that have been drawn from this analysis, and in Section 7.5 we highlight the performance analysis between the traditional and MLN approaches.

7.3 Content Analysis on User Generated Content

Content Analysis is a research method for studying documents and communication artifacts, which might be texts of various formats, pictures, audio or video [105]. One of the key advantages of using content analysis to analyze social phenomena is its non-invasive nature, in contrast to simulating social experiences or collecting survey answers. Here, we apply content analysis by text mining techniques to integrate content features into multilayer structural analysis. Motivated by the fact that, we have personality data in the given Facebook data collection and the previously preliminary studies [106, 107] observed a significant correlation between personality and privacy-concern of Facebook users, we improve [107]'s deep neural network model to detect Facebook users' privacy-concern based on their status updates and classify privacy-concern into three different levels (classes): *high, medium*, and *low*. These content-derived classes are treated as content features and modeled as layers to integrate into the multilayer structural analysis.

As briefly introduced in Section 7.2, we apply content analysis to derive the features of five personality traits and privacy-concern to adapt into the proposed multilayer network (MLN) approach. For detecting personality traits, [108] has identified many linguistic features associated with each personality trait in *FFM*. For instance, *Extraversion* (EXT) tends to seek stimulation in the external world, the company of others, and to express positive emotions. *Neuroticism* (NEU) people use more 1st person singular pronouns, more negative emotion words than positive emotion words. As mentioned earlier in Section 7.2.1, five personality trait scores for each user are given in the dataset D3. To form layer L6-L10 using these personality traits in the proposed multilayer network analysis, we followed the strategy from [109] by using five mean-values {3.8, 3.5, 3.6, 3.55, 2.8} of the given personality scores to decide "Yes" or "No" label of {OPN, CON, EXT, AGR, NEU} correspondingly. "Yes" label is assigned if the personality trait score is higher than or equal to the corresponding mean-value. Otherwise, "No" label is assigned. In the following of this section, we will mainly explain how to generate privacy-concern as one layer to adapt the multilayer network.

Previous work [107] has found that using given personality and status updates of users, privacy-concern can be predicted accurately based on UGC data. We extended their model by building a deep neural network model to automatically classify users' privacy-concern to high (HiPC), medium (MePC), and low (LoPC) based on given five personality trait scores and status updates. Motivated from previous studies [108, 107, 110], we extract the following content features from Facebook status updates, which will be used in the following deep neural network model to predict users' privacy-concern:

- **Polarity features:** Since sentiment words embody personality (e.g., *"I do**n't hate** you. Well, you found me ⋯ "*), we use the number of polarity signals appearing in FB status updates as the polarity features. We identify positive and negative words using the sentiment dictionaries provided by Hu and Liu [111]. Additionally, we consider boolean features to check whether or not a negation word is in a FB status (e.g., n't).

- **Syntactic features:** We extract part-of-speech tags (POS tags) for all status updates in the dataset. Afterwards, we use all the POS tags with their corresponding term-frequency and inverse-document-frequency (tf-idf) values as our syntactic features and feature values, respectively.

- **Semantic features:** The major challenges when dealing with user generated data are: (1) the lexicon used in a status update is informal with many out-of-vocabulary words and (2) they are usually short text [112]. The lexical and syntactic features seem not to capture that property very well. To handle this challenge, we apply two approaches to compute vector representations for FB status updates. First, we utilize Latent Dirichlet Allocation (LDA) [113] for discovering the abstract "topics" that occur in all FB status updates. Secondly, we employ 300-dimensional pre-trained embedding models at word level [114] and at character level [115] to compute a representation for a FB status update as the average of the embeddings of words and characters in the status update.

- **Lexical features:** include [*1-5*]-grams in both word and character levels. For each type of n-gram, we only select the top 1,000 n-grams based on *tf-idf*.

Deep Neural Network Model: The recent novel model from [107] was adapted to find users' privacy-concern level based on their social network status updates or personality-trait score. Their neural network model was proposed mainly for privacy-degree prediction, instead of privacy-concern level prediction. To adapt the proposed multilayer network, there is a need of categorized output to create a privacy-concern network layer (L11), where users with the same privacy-concern level is supposed to be connected. Thus, we extended their approach by developing a category-based privacy-concern detection model as shown in Figure 7.4. This extension does not only matter to the output (from discrete to categorization), but the feature representations in hidden layer of deep neural network model are different.

It is a Multilayer Perceptron (MLP) model [116], the architecture of which consists of an input layer, two hidden layers and a softmax output layer. Given all Facebook status updates of a user, the input layer represents the status update by a feature vector which concatenates lexical, syntactic, semantic and polarity feature

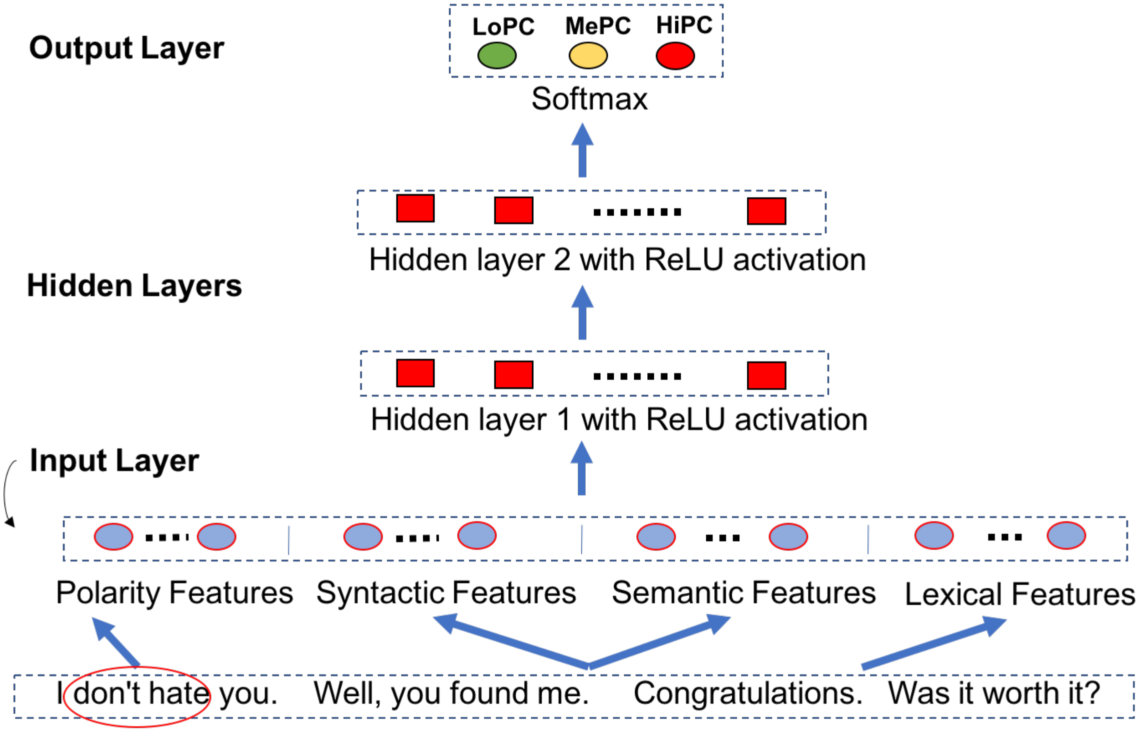


Figure 7.4: Neural network model for privacy-concern detection

representations. The two hidden layers with ReLU activation function [117] take the input feature vector to select the most important features which are then fed into the softmax output layer for privacy-concern level detection and classification. Regarding classification performance evaluation, we split 20% of the data for a blind test. We run 10 fold cross-validation on the rest 80% to train and select the best hyper-parameters. After all, the model achieved an accuracy of 84.44% on the blind test set. Furthermore, we compare the model with other popular models (i.e., support vector machine, random forest) and the recent advanced model (i.e., C-LSTM [118]), and none of them performs as good performance as our model does. This clearly shows the effectiveness of the model to predict privacy-concern levels based on UGC data. As explained above, the predicted user privacy-concern levels are used to create the network layer L11 of the multilayer network.

7.4 Experimental Study and Query Results Analysis

This section shows the results for each analytical queries (Q1-Q4) formulated in Section 7.1.2. Based on the communities obtained for the required *AND*-Compositions listed in Table 7.3, a detailed analysis was performed to draw some insights that are discussed below. To the extent possible, we have related our analysis with various published independent surveys.

Dominant Political Views (Q1): Figure 7.5 shows the distribution of the top three political views over the US population active on Facebook for each age group. Some observations are:

- Among the politically interested and socially active US people across age groups, the majority supported the democrats in the period of 2007-2012. As we know, there was a lot of support for the democratic presidential candidate who got elected in 2008, and we believe this is reflected in the political views of that period. The period of 2007-2009 also indicated the same which makes sense as the campaign was underway in that period. This is confirmed in [119, 120] since Barack Obama, a democrat, who took the US presidency on January 8, 2009 was able to influence people's political leanings through his presidential campaign from February 10, 2007.

- Among the socially active youth ($\leq$ 30 years old), majority of them have the political view of "doesn't care". Although this includes people below the voting age, even the published statistics [121] show that young people are least likely to vote and may not have formed an opinion about their political leanings.

- Among politically interested youth ($\leq$ 30 years old) who are also active on social media, dominant support is for democrats. This may be attributed to a few of president Obama's youth centric movements and the significant use of social media for the first time in a US election and also to his subsequent accomplishments [122].

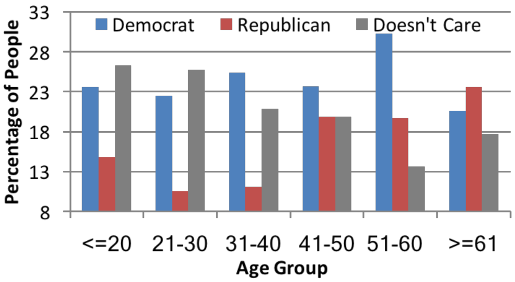


Figure 7.5: Top 3 political views by age group

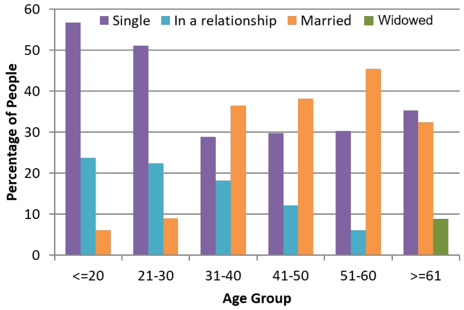


Figure 7.6: Top 3 relationship statuses by age group

- Interestingly, among all age groups, only the ones above 61 years old favored republicans over democrats, which is also reflected in the election reports from 2008 [123].

Relationship Status Correlation (Q2): The preference of a relationship status based on age and the corresponding effect on personalities of different genders were analyzed.

(Q2a) *Variation with Age:* Figure 7.6 shows, for each age group, the distribution of people among the group's top three relationship statuses. A few intuitive inferences that can be drawn are:

- The youth ($\leq$ 30 years old) stay single than be in a relationship or get married, according to the given dataset.
- The percentage of married people steadily increases with age which can be attributed to the popular fact that as people age, they want to be in a longer term commitment (in a relationship or married).
- The transitions from “Single” to “In a relationship” to “Married” are clearly seen with change in age in Figure 7.6 which matches the social trend.
- The third largest fraction of people in age group ($\geqslant$ 61) constitutes those who have lost their spouses (Widowed).

(Q2b) *Effect on Personality and Gender:* Changes in relationship status seem to have effects on the personality. Moreover, this change seems to be correlated with the gender. For the given population, we present the distribution of males and females among the top three relationship statuses - Single (S), In a relationship (R), Married (M), who display different personality traits in Table 7.4. The ones marked in **bold*** and *italics*† represent the category of people with highest and lowest percentages, respectively.

Table 7.4: % of people with different personality traits based on relationship status and gender

Trait	**Males**			**Females**		
	S(%)	**R(%)**	**M(%)**	**S(%)**	**R(%)**	**M(%)**
OPN	**55.7***	54.8	47.3	53.0	55.6	*43.0*†
CON	*44.2*†	50.2	52.7	46.1	51.7	**58.5***
EXT	45.0	**54.8***	*35.5*†	49.1	50.8	44.6
AGR	42.9	50.2	*31.8*†	46.4	45.9	**50.4***
NEU	36.5	*26.6*†	33.6	45.2	**53.8***	48.1

A few observations that can be made from the above analysis are: a) Married females have least openness (OPN) to experience, and highest conscientiousness (CON). b) Married females have the highest agreeableness (AGR), while married males have the least. These observations have been made with respect to the given population and need to be confirmed on a bigger population.

Personality Traits Analysis (Q3): Various analyses based on the five personality traits for each individual are discussed below.

(Q3a) *Contrasting personality traits:* Clearly, if a person feels anxious and does not stay relaxed (NEU) then he/she will try to make his/her life comfortable by indulging in less stressful activities making them be less open to new experiences (OPN). Thus, the fraction of people displaying these contrasting personality traits is supposed to be low. Our analytic results go hand in hand with this intuition as just 23.3% people belonging to this category.

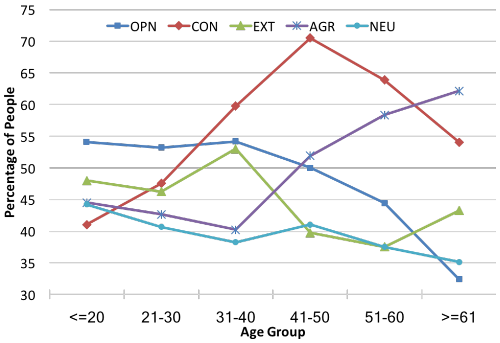


Figure 7.7: Changing personality distribution with age

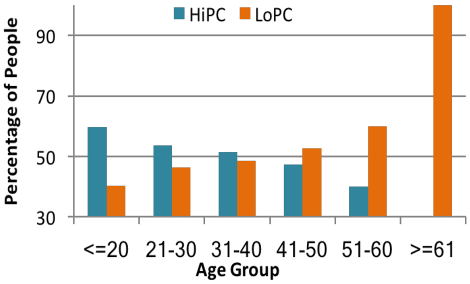


Figure 7.8: Distribution of HiPC and LoPC by age group

(Q3b) *Personality trait evolution with Age:* Figure 7.7 shows how each personality trait varies with age, based on which, few interesting observations can be made as follows.

- Openness (OPN) reflects whether one prefers new experiences and to engage in self-examination. This trait increases with age and peaks around the 30s (54.2% in age group of 31-40). However, older people prefer to go with the tried-and-tested approach (67.6% of the people above 60 years old resist new experiences).

- Conscientiousness (CON) associates with achievement and working systematically, methodically and purposefully. Analysis shows that the age group with conscientiousness the most is 41-50 years old. A recent survey about founders and entrepreneurs indicated that their average age was 45 years old [124].

- Extraversion (EXT) describes one's sociability and enjoy to be the center of attention. This trait seems to peak at two age groups (i.e., [31,40] and [$\geqslant$ 61]) in the dataset.

- Agreeableness (AGR) reflects a tendency to perceive others in a more positive light. Parenthood and grand-parenthood may make the elder generation more empathetic towards others as compared to the younger lot.

- Neuroticism (NEU) reflects one's ability to deal with emotion states, such as stress and anxiety. It can be observed from Figure 7.7 that the younger lot does not deal very well with stress. Even the studies substantiate this finding as around 80-90% adolescent suicides are linked to common psychiatric disorders, such as depression and anxiety [125]. This trait (NEU) seems to be most stable over age compared to other traits. Another age group where neurotic behavior peaks is the middle age group of 41 to 50 years old. This result also corroborates with the psychological surveys that suggest that for most midlife crisis occurs in this age group due to reasons such as age-related health problems, major life changes like death of an elderly parent, menopause in women and financial issues, leading to rise in depression [126, 127]. With age and experience comes maturity and thus, people in older age bracket deal better with stress.

Privacy Concern Correlation (Q4): Three levels of privacy concern (PC) have been considered for this work - HiPC, MePC and LoPC. Here we have taken into account age, gender, and personalities as three parameters for performing analysis to understand the choice of particular level of privacy-concern.

(Q4a) *Variation of privacy-concern across age groups:* The concern level of sharing personal information on the social media varies with age. Irrespective of the age group, the MePC was the most dominant level of privacy. Out of the remaining individuals, Figure 7.8 shows the distribution of the people with extreme levels of privacy - High and Low. Few observations are discussed below.

- People ($\leq$ 40 years old) prefer the higher level of privacy. This can be attributed to the fact that this age group is probably more aware of the cons of sharing sensitive personal information on the web such as identity theft.

- The status updates of people ($\geqslant$ 41 years old) contain more personal information and this trend increases with age. This reflects a lower level of privacy-concern probably due to their unawareness of the potential harm from disseminating personal information on social media.

(Q4b) *Correlation of privacy-concern over personality traits:* Table 7.5 shows the two extreme personality traits and their corresponding privacy-concerns for males and females.

Table 7.5: Dominant personality traits of male and females preferring different levels of privacy-concern

Privacy	**Extraversion(100%)**		**Neuroticism(100%)**	
	Males(%)	**Females(%)**	**Males(%)**	**Females(%)**
HiPC	0	0	09.90	**09.99**
MePC	39.89	**45.36**	30.15	**49.96**
LoPC	07.45	07.30	0	0

- Females have higher privacy-concern than males on both extraversion and neuroticism. This observation kept consistency with the previous study in [128].

- Both males and females on extraversion display low and some medium levels of privacy-concern. This matches the definition of extraversion from social scientists. That is, people who enjoy being the center of attention are likely to share more personal information on the web such as check-ins and day-to-day activity updates.

- Both males and females on neuroticism tend to have predominantly high privacy-concern, without anyone having low level privacy-concern. This matches the social scientists' definition.

Analysis Q4a and Q4b are important as it validates our content extraction approach to derive accurate privacy-concern.

7.5 Efficiency Analysis of The MLN Approach

So far, we have established the modeling and flexibility of analysis of the MLN approach. Below, we will establish its efficiency in general and highlight it with respect to the current dataset analysis.

Processing Layers Instead of a Single (large) Graph (SLG): Separation into layers allows one to process each layer *only once* for *all analysis* and the composition allows us to make use of the pre-computed partial results. Furthermore, in many cases, the size of each layer is likely to be smaller than the size of combinations of layers. On the other hand, a new combined graph needs to be created and processed in the traditional approach for each *unique* analytical query. Storing each layer is more compact and uses less memory than storing all the required layer combinations.

Processing Layers in Parallel: The MLN approach easily lends itself to process all (or subsets of) layers in parallel to further improve efficiency. The total cost is the processing cost of the most complex layer. This can only be done for a set of known analysis queries in the traditional approach in contrast to the MLN approach where it needs to be done only once. In the MLN approach, it is further possible to process each layer in parallel by partitioning and leveraging existing algorithms (again one time). Although this can be done in the traditional approach, it has to be done *after* the combined graph is created which reduces its effect significantly.

Efficiency of Composition: The core of the MLN approach is its ability to compose layers pair-wise to get complete, correct results. Each composition is likely to be on fewer and smaller number of components (from each layer) thereby reducing the resources needed (both storage and processing).

Combinatorial Reduction: As the complexity of dataset increases, it translates to more layers (corresponding to more features). Based on the number of layers, in the MLN approach, there is a significant and non-linear reduction in the processing cost as the number of layers increase as compared to the traditional approach. Assuming M layers, for an exhaustive analysis, they need to be combined in $2^M - 1$ ways, each representing a unique analysis of a combination of features. This translates to creating $2^M - 1$ combined graphs in the traditional approach and processing them individually. In the MLN approach, instead, M layers are processed *once* and $2^M - 1$ combining of partial results from layers are performed. As we show below, these compositions are significantly smaller (by orders of magnitude) computationally as compared to processing of a layer. Essentially, the exponential complexity has been reduced to a linear one one with very little additional processing.

7.5.1 Complexity Analysis

For the complexity analysis, we assume that for a given multilayer network with fixed number of M layers, say $\{G_1, \cdots, G_M\}$, each of the N query analyses requiring K related layers on average, should return a list of communities, L.

- Single (large) Graph (SLG): In this approach, for every analysis it first generates the composed graph, G_{AND}, obtained through (K-1) 2-layer AND-compositions, on average. On this AND-composed layer, we apply the Infomap technique (InfoM) [66] to generate the list of communities, L. Thus, for N analyses the complexity of this approach will be $O(N * (AND_{i=1}^{K} G_i + \text{InfoM}(G_{AND}))$, where $K \leq M$.

- MLN: Its first step is to generate the communities for each of the M layers by applying Infomap. While generating the communities we also obtain the internal clustering coefficient for each node, which are used to determine if the communities are *self-preserving* or not. If the property of *self-preserving* is satisfied, then for each analysis, to generate the list of communities L, for the corresponding AND-composition G_{AND}, communities from K layers are intersected based on nodes. Thus, for N analyses the cost of this approach will be, $O(\sum_{i=1}^{M}(\text{InfoM}(G_i)) + N * \cap_{j=l}^{K-l+1} C_j)$, where $K \leq M$.

In terms of storage space, one needs to store M lists of communities for MLN, whereas in SLG, $N * (AND_{i=1}^{K} G_i)$ graphs need to be stored. Considering both space and time, if the number of analyses, N, and the average number of layers required for each analysis, K, are low then the one-time cost of performing M number of Infomap operations in MLN will dominate and make MLN more expensive as compared to SLG. However, with the increasing values of N and K, the efficiency of MLN over SLG improves significantly, as the cost of producing the number of AND-composed graphs and applying Infomap that traverses through the edges of each of them, begins dominating. In the worst-case for $N = O(2^M)$ and $K = O(M)$, SLG will perform an exponential number of *AND*-Compositions and edge traversal based Infomap operations whereas, MLN will just perform the cost-effective node intersection of layer-wise communities.

7.5.2 Computational Results

Below, we show efficiency results of analysis on the given Facebook datasets.

Experimental Set up: We used a quad-core 8th generation Intel i7 processor Linux machine with 8 GB memory for all of our analysis. Based on Table 7.3, we computed communities for a total of 19 *AND*-compositions to answer the queries Q1 to Q4, each requiring 3 layers on an average.

Figure 7.9 (a) shows the time taken for processing all of 11 layers *with and without parallelism.* As can be seen, in the MLN approach *with parallelism*, it reduces the cost of processing the most complex layer (9.847 seconds for L5: Locale, 2.8 million edges, Density: 0.77 - *most dense*) – a ***reduction of 80.4%*** *approximately.*

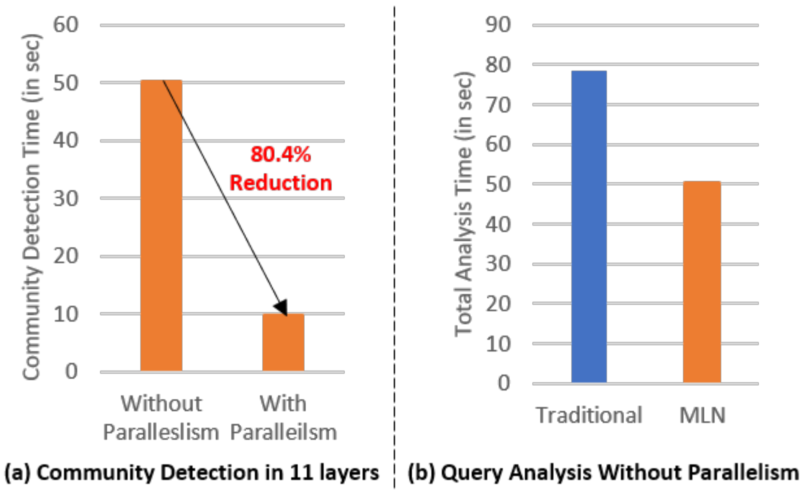


Figure 7.9: Efficiency Comparison of MLN and Traditional Approaches

The incremental computation cost for each query using the MLN approach is extremely small. This can be appreciated from the worst case scenario - comparing minimum layer processing cost with maximum composition cost. The total composition cost to answer the most complex query (Q2b) was **0.039 seconds** and the minimum layer processing cost was **1.61 seconds** (L4: Political View, 494K edges, Density: 0.14 - *least dense*). **The difference is more than two orders of magnitude.**

Figure 7.9 (b) shows the *total time* taken to answer the analysis queries using the traditional and the MLN approach, respectively, *without parallelism*, as 78.520 seconds and 50.354 seconds (for **36% reduction**). Further, if communities for each layer are generated in parallel, total computation time for the MLN approach reduces to 9.987 seconds (for **87% Reduction**). Also, note that the analysis shown in this chapter is **less than 1% of the possible analysis**.

In summary, the experiments on the Facebook Dataset validate the *MLN approach from an efficiency perspective as compared to the traditional approach.*

7.6 Conclusions

In this chapter, we have applied the emerging MLN approach model and analyze a social network data collection in a flexible and efficient way. We have also shown how content analysis can be readily incorporated into the proposed MLN approach. Experimental analysis and evaluation not only demonstrate the flexibility and efficiency of data analysis using the MLN approach but also validate the analysis results. Importantly, the work in this chapter led to a conference publication [10].

For future study, we plan to (1) apply the proposed MLN approach to the bigger full data collection and (2) apply hypothesis testing on two different data distributions (e.g., the current one versus the full data collection) to see the statistical significant degree of our findings.

Acknowledgement

This work is supported by the STINT project funded by The Swedish Foundation for International Cooperation in Research and Higher Education and was done in collaboration with Dr. Lili Jiang and Mr. Xuan-Son Vu from the Umeå University. We also thank the myPersonality project for data contribution.

CHAPTER 8

FLEXIBLE ANALYSIS OF MOVIE ACTOR DATA SET

In this chapter, we use the following **IMDb (Internet Movie Database)** data set to illustrate analysis-driven modeling and computation. The IMDb data set captures movies, TV episodes, actor, directors and other related information, such as rating. This is a large data set consisting of movie and TV episode data from their beginnings. The available ground truth from independent sources has been for validation of the various analysis results obtained in this chapter.

8.1 Analysis-Driven MLN Modeling

To demonstrate the effectiveness of modeling we apply MLNs to answer the following key questions.

(A1) Which is the largest group(s) of *co-actors* that lead to the *most popular movie ratings*?

(A2) Which are the groups of actors who have acted in similar movie genres and are also highly rated?

(A3) Which highly rated actors work in similar genres but have *not co-acted together* in any movie?

The IMDb Actor data set analysis objectives (**(A1)** - **(A3)**) require defining appropriate relationships among *actors* with respect to different features like co-acting, movie genres, and ratings. As the entity set used for analysis is the same, but using different features, these data sets should be **modeled as Homogeneous MLNs**.

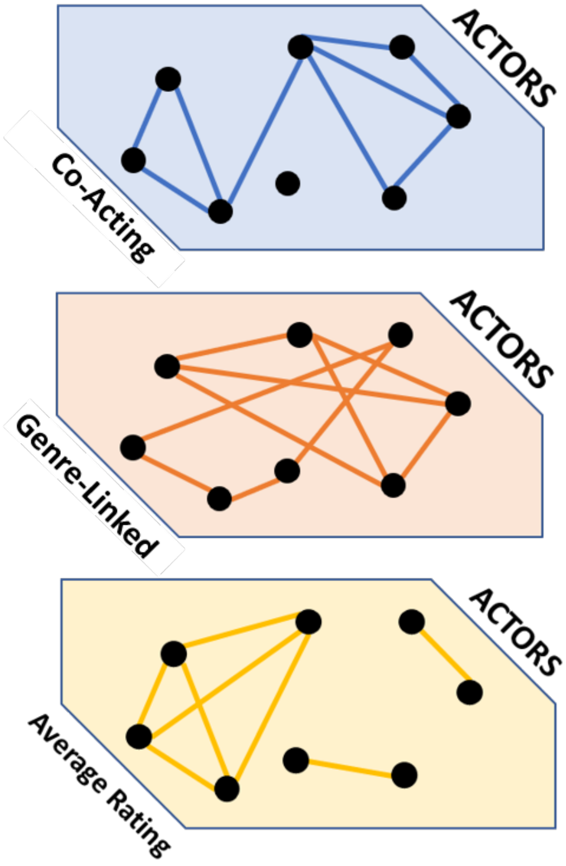


Figure 8.1: IMDb HoMLN for Actor Relationships

Typically, number of layers correspond to the number of relationships that need to be captured (one for each layer). The semantics of the analysis objectives determines the choice of nodes and intra-layer edges. Here, *the nodes in each layer correspond to actors*. Two nodes are connected in a layer, if they have *co-acted in at least one movie* (Layer *Co-Acting*) or belong to the *same average rating range* (Layer *AvgRating*). The average rating of an actor has been calculated by taking into account the IMDb ratings of the movies he/she has acted in. For this, 10 ranges are created - [0-1), [1-2), ..., [9-10]. There are multiple ways of quantifying the similarity of actors based on movie genres they have worked in. For every actor a vector was generated with the number of movies for each genre he/she has acted in. In order to consider the similarity with respect to *frequency of genres*, in layer *Genre*, two actors are connected if the Pearsons' Correlation between their corresponding genre vectors is at least 0.9*. Figure 8.1 shows the proposed HoMLN.

8.2 Mapping Analysis Objectives to Computations

In adherence with the decoupling approach, We need to identify Ψ and Θ for each detailed analysis objective (**(A1)** to **(A3)**) along with their application on layers in a specified order.

We use community as analysis function. For composition of HoMLNs, we use the proposed Boolean AND, OR, and unary NOT. Table 10.1 summarizes the mapping of each detailed analysis **(A1)** to **(A3)** to their actual computation specification (in *left* to *right* order), analysis function (Ψ) and composition function (Θ). This is used for computing the results in the experimental section (Section 12.2.)

For **(A1)**, communities from each layer are composed using the Boolean AND operation. Objective **(A3)** requires all 3 layers, where co-actor groups who have not worked together correspond to finding communities in the NOT (Layer *Co-Acting*). The edges in a NOT layer are complement of the edges in the original layer.

8.3 Experimental Analysis

We compute the results for each detailed objective using the expressions shown in Table 10.1 and compare it, where possible, with independently available ground truth. This helps validate both the modeling and analysis aspects of the approach

*The choice of the coefficient is not arbitrary as it reflects relationship quality. The choice of this value can be based on how actors are weighted against the genres. We have chosen 0.9 for connecting actors in their top genres.

Analysis	**Mapping**		
	Computation Order	Ψ	Θ
*IMDb (**HoMLN**)*			
(A1)	Co-Acting Θ AvgRating	Community (Louvain)	AND
(A2)	AvgRating θ Genre	Community (Louvain)	AND
(A3)	NOT(Co-Acting) Θ Genre Θ AvgRating	Community (Louvain)	AND

Table 8.1: MLN Expression for IMDB-Actors Analysis Objectives

proposed in this chapter. We will also present a few results to highlight the efficiency of the decoupling approach.

A795
A1203
A285
A846
A542
A1478
A2115
A1469
A941
A176
A15
A721
A234
A191
A738
A1043
A62
A517
A90
A1579
A937
A63
A27
A489
A928
A352
A1463
A2208

Figure 8.2: Word Cloud for the Layer Co-Acting

For the top 500 actors, we extracted the movies they have worked in (7500+ movies with 4500+ directors). The actor set was repopulated with the co-actors from these movies, giving a total of 9000+ actors. The HoMLN with 3 layers described in Section 5.1 was built for these *set of actors* (statistics shown in Table 8.2). The Louvain method ([67]) was used to detect the layer-wise communities (partial results.) Figure 8.2 shows the Word Cloud corresponding to the Co-Acting layer, where larger font size depicts the community with more number of actors. The i^{th} community id is denoted by Ai, in the figure.

Around 44% actors (mostly world renowned) had an average rating in the range [6-7) making it the *most popular IMDb rating class*, while only 1.8% actors have the highest average rating in the range [9-10]. On the other hand, the largest co-acting and similar genre groups had 15.6% and 15.3% actors, respectively.

	Co-Acting	**Genre**	**AvgRating**
#Nodes	9485	9485	9485
#Edges	45,581	996,527	13,945,912
#Communities	2246	63	8
Avg. Community Size	4.2	148.5	1185.6

Table 8.2: IMDB HoMLN Statistics

For **(A1)**, 2430 actor groups with similar average ratings were detected in which *most of the actor pairs* have worked with each other. Few observations on the results:

- For the most popular average actor rating, [6-7), the largest co-actor groups were from Hollywood (876 actors), Indian (44 actors), Hong Kong (12 actors) and Spanish (9 actors) movies.
- Among the Hollywood movie based groups, the top group included actors like *Al Pacino, Robert De Niro, Tom Cruise and Will Smith.*
- Famous Bollywood stars like *Amitabh Bachchan*, *Shah Rukh Khan* belonged to largest top rated Indian group.
- *Jackie Chan* was among the prominent actors from the co-actor group from Hong Kong.

In case of **(A2)**, 592 actor groups who opt for *similar movie genres* are detected across *different average rating ranges.* As a part of this analysis, the group corresponding to **Action, Adventure and Sci-Fi** genres and an average rating of [6-7) had actors like **Robert Downey Jr., Mark Ruffalo, Gwyneth Paltrow and Scarlett Johansson**. This is primarily because these actors/actresses have been part of the **Marvel Cinematic Universe movies over the past 10 years**. Similarly, **Sean Connery, Tom Cruise, Robert De Niro and Johnny Depp** have been grouped together in an actor group whose members have primarily worked in Drama and Action related movies and on an average receive [6-7) as their movie ratings.

For **(A3)**, we detected 900 groups of actors with similar genre preferences and average rating *but most of whom have not worked together.* From *highly rated* groups where each actor has acted in *different prominent genres*, Table 8.3 shows a few recognizable *actors* **who have not acted together**. Out of these, interestingly, in 2017, as per reports there had been **talks of casting Johnny Depp and Tom Cruise in pivotal roles in Universal Studios' cinematic universe titled Dark Universe** [129].

Actors/Actresses	**Common Prominent Genres**
Willem Dafoe, Russell Crowe	Action, Crime
Hilary Swank, Kate Winslet	Drama
Tom Hanks, Reese Witherspoon, Cameron Diaz	Comedy, Romance
Johnny Depp, Tom Cruise	Adventure, Action
Leonardo DiCaprio, Ryan Gosling	Crime, Romance
Nicolas Cage, Antonio Banderas	Action, Thriller
Hugh Grant, Kate Hudson, Emma Stone	Comedy, Romance

Table 8.3: **(A3)**: **Never Co-Acted** Highly Rated Genre Actors

8.3.1 Efficiency Analysis of the Decoupling Approach

Experimental Set up: We used a quad-core 8th generation Intel i7 processor Linux machine with 8 GB memory for all of our analysis. The layer-wise results (communities or hubs) are generated *once* and can be done in *parallel*. Thus, this one time cost is bounded by the layer that takes maximum time. Moreover, the cost of composing the partial results using Boolean AND (HoMLN Communities). We compare the total computational cost of the decoupling approach and the traditional single graph approach which includes the *time to generate the combined layer* followed by generating the communities.

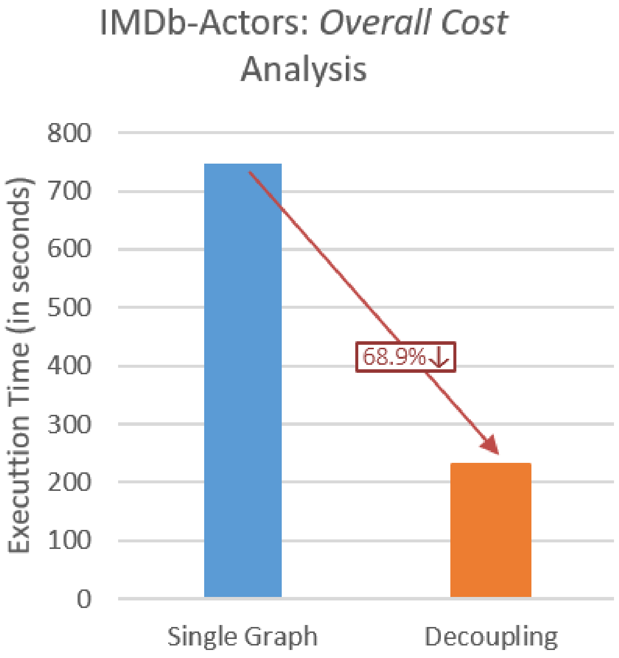


Figure 8.3: Efficiency of Decoupling Approach for IMDb Actors HoMLN analysis

In **(A1)** and **(A2)** 1 Boolean AND composition is required each as per Table 10.1 to generate communities. In case of **(A3)**, 2 Boolean AND compositions are required. Here the one time cost for finding the layer-wise communities is bound by the *AvgRating* layer (densest layer in Table 8.2). Overall, **68.9% reduction in computation time** is observed with the *decoupling approach (231.676 seconds)* as compared to *single graph approach (745.831 seconds)* as shown in Figure 8.3 (a), thus validating the **MLN decoupling approach from an efficiency perspective**.

CHAPTER 9

EFFICIENT ESTIMATION OF CENTRAL ENTITIES FOR HOMOGENEOUS MLNs

In this chapter, we concentrate on finding high degree and closeness centrality vertices, also called hubs, in AND-composed layers of homogeneous multilayer networks using decoupling approach.

Centrality measures are used in networks to find out the *most influential nodes.* Some hub-based insights that can be obtained via computations on MLN layers are - a). High centrality vertices in the accident data set (a pair of traffic accidents may be related if they occurred in the same location, or under the same light condition, weather condition etc.) can help us in *identifying the most dominating traffic accident locations with respect to poor lighting conditions and bad roads* and this information can be used to devise appropriate accident prevention techniques. b). High centrality vertices in the IMDB data set (two actors may be related if they acted in the same genre, such as action, comedy, etc.) indicate the most popular/preferred co-actors across all types of genres to be used by casting companies and production houses. c). High centrality vertices can be used to include the set of people who turn out to be most influential across different communication platforms like Facebook and LinkedIN to be used by advertisement agencies for effective information transfer. In the thesis, we have dealt with the centrality measurement in HoMLN (also, called multiplexes).

However, as discussed earlier, in order to obtain a holistic view of the MLN with n layers, we have to generate, store and analyze a total of $2^n - 1$ networks, leading to extremely expensive operations for multiplexes with large number of layers (for example the network in[29] has 300 layers.)

Problem Formulation and Contributions: Given this challenge of efficiently finding hubs in HoMLN, the main problem we aim to solve is as follows. Given a dataset with multiple entities that are related via a number of distinct features, how can we efficiently find the most influential entities based on any conjunctive (AND) combination of features using the decoupling approach.

To solve this problem, we use multiplexes for representing such multi-feature datasets and *present elegant techniques for estimating the hubs for any conjunctively composed multiplex layer, without actually constructing that composed layer.*

Our main contributions are two-fold. **First** we show that finding high centrality vertices in the AND composed HoMLNs, based on only analyzing the individual layers is a non-trivial problem, and the naive approach of simply taking the intersection of the hubs from each layer does not produce accurate results. **Second**, we present four heuristics (3 for degree centrality and 1 for closeness centrality) to identify hubs in the AND-composed using only the hubs detected in individual layers and their distance-1 neighbors. Our results show that we can identify the vertices with $70 - 80\%$ accuracy while reducing the computation time by at least 30%.

Figure 9.1 shows the MLNs that have been used in this chapter for centrality illustration proposed. Figure 9.1 (a) shows an accident multiplex (or HoMLN) depicting the similarity among 7 accident occurrences based on light (G_{a1}) and weather (G_{a2}) conditions. Similarly, in Figure 9.1 (b), the IMDb multiplex depicts the co-actor relationship among 6 actors based on the movie genres, comedy (G_{m1}) and action (G_{m2}). The notations mentioned in Table 9.1 have been used to formalize the various centrality-related concepts discussed in this chapter.

Table 9.1: List of notations used for defining the concepts.

I	Set of entities
f	Set of features/perspectives
$G(V_k, E_k)/G_k$	The k^{th} layer
u_i^k	Representative node for i^{th} entity in the k^{th} layer
$NBD_k(u_i^k)$	Set of nodes adjacent to the i^{th} node in the k^{th} layer
deg_i^k	Degree of the i^{th} node in the k^{th} layer
$avgDeg^k$	Average degree of the k^{th} layer
clo_i^k	Closeness centrality of the i^{th} node in the k^{th} layer
$avgClo^k$	Average closeness centrality of the k^{th} layer
V_k	Set of nodes in the k^{th} layer
(u_i^k, u_j^k)	An edge in the k^{th} layer
E_k	Set of edges in the k^{th} layer
DH_k	Set of degree centrality based hubs in k^{th} layer
CH_k	Set of closeness centrality based hubs in k^{th} layer

Our proposed methods can be extended to any number of layers. This approach significantly reduces the complexity of analyzing the AND-composed network and also the storage as only n individual layers are constructed and analyzed.

The remainder of this chapter is organized as follows: In Section 9.1, we detect high degree and closeness centrality vertices in each layer. We show how these hub sets vary across different individual and AND-composed layers. In Section 9.2, we present four heuristics to improve the accuracy of computing the degree or closeness centrality based hubs of any conjunctive combination of layers by using the required layer-wise hubs. In Section 9.3, we empirically validate the quality of the hub sets generated by executing our algorithms on two diverse data sets: traffic accidents and IMDb. We use the Jaccard Index to compare the set of hubs obtained through our

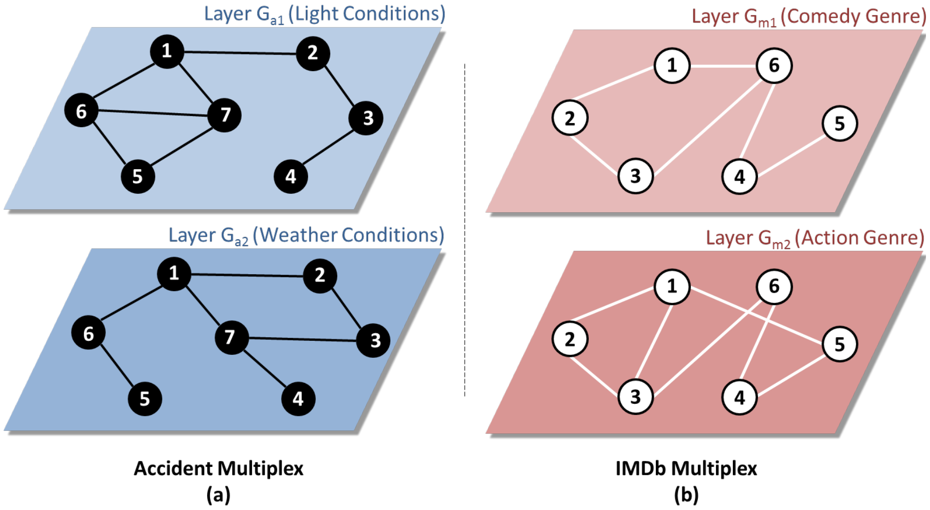

Layer G_{a1} (Light Conditions)
Layer G_{a2} (Weather Conditions)
Layer G_{m1} (Comedy Genre)
Layer G_{m2} (Action Genre)
1
2
3
4
5
6
7
Accident Multiplex
(a)
IMDb Multiplex
(b)

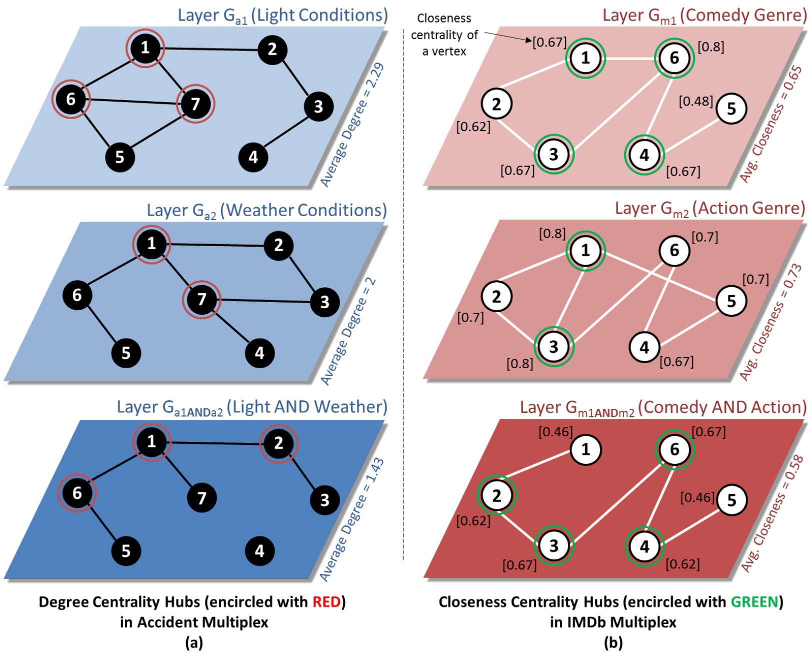
Figure 9.1: Snapshots of accident and IMDb multiplexes/HoMLNs

heuristics with the actual set of hubs. We show that our approach can significantly reduce the computational costs of finding hubs in the composed networks.

9.1 Hubs (High Centrality Vertices) across Multiplex Layers

Entities vary in their influencing capability with respect to the occurrence of events, interaction networks and so on. For example, a particular person might be considered highly influential if he/she is connected to a large majority of people on Facebook. Thus, an advertisement agency will prefer this person in order to enhance their information transfer. However, he/she may not be equally influential on LinkedIN. Thus, in case of multi-featured data, the influencing capability for a particular entity may vary substantially with features. With respect to multiplexes, this translates to generating the hubs across different individual or AND-composed layers.

Degree Centrality (deg_i^k): The number of nodes adjacent to the i^{th} vertex in the k^{th} multiplex layer defines a vertex's layer specific degree. Higher the degree of a node, greater is its influence on the immediate neighborhood. We define high centrality nodes or hubs in the k^{th} layer (or feature) as the ones that have a degree greater than the average degree of the layer, $avgDeg^k$, which is computed by $\frac{2|E_k|}{|V_k|}$. Figure 9.2 (a) encircles the accident nodes in red that have been detected as hubs due to their greater than average degree.


Layer G_{a1} (Light Conditions)
Average Degree = 2.29
Layer G_{a2} (Weather Conditions)
Average Degree = 2
Layer $G_{a1ANDa2}$ (Light AND Weather)
Average Degree = 1.43
Closeness centrality of a vertex
Layer G_{m1} (Comedy Genre)
[0.67]
[0.8]
[0.62]
[0.48]
[0.67]
[0.67]
Avg. Closeness = 0.65
Layer G_{m2} (Action Genre)
[0.8]
[0.7]
[0.7]
[0.7]
[0.8]
[0.67]
Avg. Closeness = 0.73
Layer $G_{m1ANDm2}$ (Comedy AND Action)
[0.46]
[0.67]
[0.46]
[0.62]
[0.67]
[0.62]
Avg. Closeness = 0.58
1
2
3
4
5
6
7

Degree Centrality Hubs (encircled with RED) in Accident Multiplex
(a)

Closeness Centrality Hubs (encircled with GREEN) in IMDb Multiplex
(b)

Figure 9.2: Variation in the Degree and Closeness Centrality based Hubs across Different Individual and Composed Multiplex Layers

Closeness Centrality (clo_i^k): The closeness centrality of a node measures how close are the other nodes in the network from it. Therefore, closeness centrality of the i^{th} vertex in the k^{th} multiplex layer is defined by the average of the summation of reciprocal of shortest paths between the i^{th} node and every other node in the layer. We use the valued closeness centrality variant as proposed in [86, 85] as any multiplex layer need not be comprised of a single connected component. Therefore, $clo_i^k = \frac{1}{|V_k|-1}\sum_{j=1,j\neq i}^{|V_k|}\frac{1}{d(u_i^k,u_j^k)}$, where $d(u_i^k, u_j^k)$ is the shortest path between the i^{th} and the j^{th} vertex in the k^{th} layer. Higher is the closeness centrality of a node, closer it is all other nodes in the layer and greater will be its influence on the network. Similar to degree we define the high centrality nodes or hubs in the k^{th} layer (or feature) as the ones that have their closeness centrality metric value greater than the average closeness centrality of the layer, $avgClo^k$, which is computed by $\frac{\sum_{i=1}^{|V_k|} clo_i^k}{|V_k|}$. Figure 9.2 (b) encircles the actor nodes in green that have been detected as hubs based on closeness centrality.

Characteristics of Hubs in the Composed Layers In Figure 9.2, we show using simple examples that finding hubs of the composed layer from the individual hubs is a non-trivial problem. In some cases, such as for actor 4 (or accident 6) *a vertex may be a hub in the composed layer even if it is not a hub in both the layers.* Further, the actor 1 and accident 7 illustrate that *a node that is a hub in both individual layers may not be a hub in the AND-composed layer.* Moreover, there can be some entities like actor 2 and accident 2 that are *hubs in the AND-composed layer in spite of not being a hub in either of the individual layers.* This is due to the fact that edge connectivity varies across individual and composed layers, thus effecting the values of degree centrality and closeness centrality. Our goal is to develop heuristics that can take into account these connectivity patterns and identify the hubs in the AND-composed layer from the hubs of the individual layers.

9.2 Identifying Hubs in AND-composed Multiplexes

In this section, we introduce four heuristics to identify the degree or closeness centrality hub sets in the AND-composed layer using information about the hubs in the individual layers. Our techniques eliminate the need to generate, store and compute degrees and shortest paths for the AND-composed layers, thus reducing the computational complexity.

For the following discussion, let us assume the two individual layers to be G_x and G_y, with degree centrality based hub sets, DH_x and DH_y, respectively, and closeness centrality based hub sets, CH_x and CH_y, respectively. Further, let us suppose that DH_{xANDy} and CH_{xANDy} are the *actual* degree centrality and closeness centrality based hub sets, respectively, for the AND-composed layer, G_{xANDy}.

9.2.1 Estimating Hubs based on Degree Centrality

As shown in Figure 9.2 (a), a) a node that is not high degree in the individual layers may share enough neighbors across layers to become a hub in the AND-composed layer, whereas b) the node that is a hub across layers may lose its hub property after AND-composition due to the absence of common neighbors. Therefore, the naive way of taking the intersection of layer-wise hubs to find the hubs in the AND-composed layer will generate a large number of false positives and false negatives. Here we propose and discuss three heuristics to estimate degree centrality based hub set of the AND-composed layer.

Heuristic DC1: To reduce the false positives, we estimate the average degree of the AND-composed layer, $avgDeg_{est}^{xANDy}$. Note that the upper bound on the average degree in the AND-composed networks will be the minimum average degree from the individual layers. Therefore, $avgDeg^{xANDy} \leq min(avgDeg^x, avgDeg^x)$. We set the estimated average degree of the AND-composed network to this upper bound: $avgDeg_{est}^{xANDy} = min(avgDeg^x, avgDeg^x)$.

We first obtain the vertices from the intersection of the hubs in the individual layers, i.e. all nodes $u \in DH_x \cap DH_y$. We then check whether these nodes have a common set of one hop neighbors in their individual layers. The larger the set of common neighbors, the greater the degree in the AND-composed network. Formally we only retain the vertex u as a hub if $|NBD_x(u) \cap NBD_y(u)| > avgDeg_{est}^{xANDy}$, where $NBD_x(u)$ and $NBD_y(u)$ denote the sets of one hop neighbors of vertex u in G_x and G_y, respectively.

Algorithm 6 Procedure for Heuristic DC1

```
Require: $DH_x$, $avgDeg^x$, $DH_y$, $avgDeg^y$, $DH'_{xANDy} = \emptyset$
$avgDeg_{est}^{xANDy} = min(avgDeg^x, avgDeg^x)$.
for all $u \in DH_x$ do
    $NBD_x(u) \leftarrow$ one hop neighbors of $u$ in $G_x$
end for
for all $u \in DH_y$ do
    $NBD_y(u) \leftarrow$ one hop neighbors of $u$ in $G_y$
end for
for all $u \in DH_x \cap DH_y$ do
    if $|NBD_x(u) \cap NBD_y(u)| > avgDeg_{est}^{xANDy}$ then
        $DH'_{xANDy} \leftarrow DH'_{xANDy} \cup u$
    end if
end for
```

Heuristic DC2: In the above heuristic, if $avgDeg_{est}^{xANDy}$ is much larger than $avgDeg^{xANDy}$, then a common hub in spite of sharing enough neighbors across the individual layers will not be generated as a hub in the composed layer. A better estimate for the AND-composed layer's average degree is obtained by *maintaining the degree of each vertex in every individual layer.* In the AND-composed layer, the number of neighbors for any vertex will be at most that vertex's least degree among all individual layers. That is, $deg_i^{xANDy} \leq min(deg_i^x, deg_i^y)$. This implies, $avgDeg^{xANDy} \leq \frac{1}{|v_x|} \sum_{i=1}^{V_x} min(deg_i^x, deg_i^y)$. We set the estimated average degree of the AND-composed network to this upper bound, $avgDeg_{est}^{xANDy} = \frac{1}{|v_x|} \sum_{i=1}^{V_x} min(deg_i^x, deg_i^y)$. We execute the steps in heuristic DC1 with this improved estimate. This method provides a better accuracy as compared to DC1, but the computational cost increases.

Heuristic DC3: Heuristics DC1 and DC2 reduce false positives but cannot handle false negatives. Specifically they miss out vertices that are hubs in the AND-composed layer but are not hubs in any of the individual layers. For handling this case, we maintain few low degree nodes from each individual layer that have a degree close to the average degree. That is, *if* $deg_i^x > (1 - \epsilon)avgDeg^x$*, then insert the vertex in* DH_x*, where* $0 \leq \epsilon \leq 1$, and we similarly update DH_y. Therefore, executing heuristic DC2 with these updated layer-wise hub sets, will also generate those common layer-wise non-hubs that share enough neighbors to become hubs in the AND-composed layer. The higher the value of ϵ, more accurate will be the estimated hub set. This increased accuracy comes at a cost of maintaining more overhead information. Thus, from DC2 and DC3 it is evident that there is a trade-off between accuracy and savings in computational costs.

Discussion: If the topology of the individual layers, G_x and G_y is similar, then most of the layer-wise hubs will also be hubs in the AND-composed networks and the naive approach can give a good estimation. Also note that if the average degree estimate for the AND-composed layer is not close enough to the actual average degree then even an ϵ value of 1 may not give 100% accuracy due to the exclusion of common hubs and non-hubs that share more than actual but less than estimated average degree number of neighbors across layers. Therefore, the effectiveness of our heuristics depends on the fraction of AND-composition hubs that are common to the layers, average degree estimate and the value of ϵ.

Algorithm 7 Procedure for Heuristic DC3

```
Require: $DH_x$, $deg_i^x \ \forall \ u_i^x$, $avgDeg^x$, $DH_y$, $deg_i^y \ \forall \ u_i^y$, $avgDeg^x$, $\epsilon$, $DH'_{xANDy} = \emptyset$
for all $u_i^x \in V_x$ do
    if $deg_i^x > (1-\epsilon)avgDeg^x$ then
        $DH_x \leftarrow DH_x \cup u_i^x$
    end if
end for
for all $u_i^y \in V_y$ do
    if $deg_i^y > (1-\epsilon)avgDeg^y$ then
        $DH_y \leftarrow DH_y \cup u_i^y$
    end if
end for
execute Heuristic DC2 with updated $DH_x$ and $DH_y$.
```

9.2.2 Estimating Hubs based on Closeness Centrality

Closeness centrality depends on the shortest paths between any two nodes. As shown in Figure 9.2 (b) that even if a certain node is closest to all the remaining nodes in the individual layers, it may not be a hub in the AND-composed layer due to the absence of common paths between this node and every other node that are short enough. Therefore, the naive way of intersecting the layer-wise closeness centrality based hubs will generate false positives. We propose and analyze a heuristic that maintains minimal neighborhood information to estimate the closeness centrality hubs for the AND-composed layer.

Heuristic CC1: From a high closeness centrality node we can traverse the entire network in minimum number of hops. Therefore, if high degree nodes are close to a node, the chances of this node becoming a high closeness centrality node increase.

Therefore, one way of eliminating the false positives is to check whether the common closeness centrality hubs share high degree neighbors across layers.

Based on this observation, we propose the following heuristic. Initially, for every node, $u \in CH_x$ (or, $u \in CH_y$), we obtain the set of degree based hubs present in its one hop neighborhood, $degNBD_x(u)$ (or $degNBD_y(u)$). We estimate the degree based hub set for AND-composed layer, DH'_{xANDy}, using one of the heuristics discussed above. We then obtain the set of common closeness centrality hubs from CH_x and CH_y. For each of these vertices, we obtain the set of those common degree based hubs in the one hop neighborhood that are also estimated to be hubs in the AND-composed layer. The larger the size of this set, greater are the chances of a node to remain a high closeness centrality node even in the AND-composed layer. Formally, we only retain a vertex u as a closeness centrality based hub if $|degNBD_x(u) \cap degNBD_y(u) \cap DH'_{xANDy}| \geq 1$.

Discussion: If the topology of layer G_x is similar to G_y, then the shortest paths between most of the node pairs will be common. In such a case, the naive approach is capable of generating good hub set estimates of the layer G_{xANDy}. Maintaining information about the alternate paths to every degree based hub beyond 2-3 hops from the closeness centrality hubs and similar path information about some layer-wise non-closeness centrality based hubs will improve the accuracy of the heuristic. However, due to the large overhead costs the computational time will significantly increase.

9.2.3 Estimation of Hubs in k-layer AND Compositions

The input to any of the above heuristics is two hub sets that may either be the actual hub sets of individual layers or the estimated hub sets of AND-composed layers. For any 3 layers, G_x, G_y and G_z, the average degree estimation and neighborhood in-

Algorithm 8 Procedure for Heuristic CC1

```
Require: $CH_x$, $DH_x$, $CH_y$, $DH_y$, $DH'_{xANDy}$, $CH'_{xANDy} = \emptyset$
for all $u \in CH_x$ do
    $degNBD_x(u) = \emptyset$
    for all $v \in NBD_x(u)$ do
        if $v \in DH_x$ then
            $degNBD_x(u) \leftarrow degNBD_x(u) \cup v$
        end if
    end for
end for
for all $u \in CH_y$ do
    $degNBD_y(u) = \emptyset$
    for all $v \in NBD_y(u)$ do
        if $v \in DH_y$ then
            $degNBD_y(u) \leftarrow degNBD_y(u) \cup v$
        end if
    end for
end for
for all $u \in CH_x \cap CH_y$ do
    if $|degNBD_x(u) \cap degNBD_y(u) \cap DH'_{xANDy}| \geq 1$ then
        $CH'_{xANDy} \leftarrow CH'_{xANDy} \cup u$
    end if
end for
```

tersection are both commutative and associative. Therefore, the four proposed heuristics are also commutative ($DH'_{xANDy} = DH'_{yANDx}$, $CH'_{xANDy} = CH'_{yANDx}$) and associative ($DH'_{(xANDy)ANDz} = DH'_{xAND(yANDz)}$, $CH'_{(xANDy)ANDz} = CH'_{xAND(yANDz)}$). Therefore, to estimate the hub sets of a k-layer AND-composed network, any heuristic is applied on the k/2 pairs of hub sets, in parallel and in order, generating k/2 AND-composed hub sets, and so on until the final estimated set of hubs, corresponding to the k-layer AND-composed network, is obtained. Thus, in this way for a multiplex with n layers, the $2^n - n$ AND-composition hub sets can be estimated by only using n layer-wise hub sets and minimal overhead information.

9.3 Experimental Analysis

In this section we present our experimental results on the performance of the four proposed heuristics to estimate the hub sets of the AND-composed multiplex layers with respect to accuracy and computational costs. Specifically, we i) construct multiplexes for datasets from diverse domains, ii) generate the AND-composed layers and the actual sets of high centrality nodes, iii) obtain the estimated hub set based on our heuristics and iv) compute accuracy of the estimated hubs based on the actual hub set.

Experimental Setup and Datasets: Our codes are implemented in C++ and executed on a Linux machine with 4 GB RAM and installed with UBUNTU 13.10.

Our experiments are performed on two different multiplexes built from real-life datasets collected from diverse domains - UK Traffic Accidents [100], Internet Movie Database - IMDb [94]. Detailed structure of these multiplexes is as follows:

Accident Multiplex: We use 1000 random road accidents that occurred in the United Kingdom in the year 2014. This multiplex has 3 basic layers with respect to Light Conditions (Domain = {daylight, darkness: lights lit, darkness: lights unlit, darkness: no lighting, darkness: lighting unknown}), Weather Conditions (Domain = {fine + no high winds, raining + no high winds, snowing + no high winds, fine + high winds, raining + high winds, snowing + high winds, fog or mist, other}) and Road Surface Conditions (Domain = {dry, wet or damp, snow, frost or ice, flood, oil or diesel, mud}). An edge in any layer represents that the corresponding accidents occurred within 10 miles of each other and are similar based on light conditions (layer G_{a1}), weather conditions (layer G_{a2}) or road surface conditions (layer G_{a3}).

IMDb Multiplex: This 3-layer multiplex is built with 5000 random actors. An edge in any basic layer signifies that the corresponding actors have worked together in at least one movie that belongs to the Comedy genre (layer G_{m1}), Action genre (layer G_{m2}) or Drama genre (layer G_{m3}).

Actual Hub Sets in the Individual and AND-composed Layers: Apart from the individual multiplex layers, four AND-composed layers each, for the accident multiplex - $G_{a1ANDa2}$, $G_{a1ANDa3}$, $G_{a2ANDa3}$ and $G_{a1ANDa2ANDa3}$, and IMDb multiplex - $G_{m1ANDm2}$, $G_{m1ANDm3}$, $G_{m2ANDm3}$ and $G_{m1ANDm2ANDm3}$, are generated. Every cell in Table 9.2 lists percentage of hubs followed by the average degree or closeness centrality for the individual and AND-composed multiplex layers. Variation in this information across layers shows that any combination of layers (or features) presents a unique perspective of analyzing the same set of entities.

Comparison Metrics: We compare the similarity of the estimated hub sets with the actual hub sets using the Jaccard Index. For any two sets, X and Y, jaccard index, $J_{X,Y} = \frac{|X \cap Y|}{|X \cup Y|}$. If two sets completely overlap, then jaccard index is 1, denoting highest accuracy of 100%. We compute overall accuracy of a heuristic as the mean of the accuracies obtained by estimating hub sets of every AND-Composed layer.

The computational time to generate the actual hub set for any AND-composition includes the time to generate the AND-composed layer followed by the time it takes

AND-Composed Layer	Accident *(x = a)*		IMDb *(x = m)*	
	$\lvert DH_k \rvert$ avgDeg	$\lvert CH_k \rvert$ avgClo	$\lvert DH_k \rvert$ avgDeg	$\lvert CH_k \rvert$ avgClo
G_{x1}	23.4% 14.92	30.6% 0.0324	34.9% 1.4404	29.4% 0.0181
G_{x2}	20.5% 17.99	36.3% 0.0462	29.4% 0.8564	19% 0.0071
G_{x3}	21.3% 16.44	28.5% 0.0347	47.1% 1.92	39.4% 0.031
$G_{x1ANDx2}$	21% 11.2	28% 0.0251	9.6% 0.1948	9.6% 0.00009
$G_{x1ANDx3}$	20.4% 10.18	25.2% 0.0202	22.7% 0.5176	10.5% 0.0016
$G_{x2ANDx3}$	18.2% 14.35	26.2% 0.0302	11.8% 0.24	9.3% 0.0002
$G_{x1ANDx2ANDx3}$	18.2% 9.28	24.1% 0.0186	1.6% 0.0228	1.6% 0.000005

Table 9.2: Varying Hub Information denoting the Diverse Perspectives obtained through Multiplex Layers

to compute degree based hubs or shortest paths for closeness centrality based hubs. On the other hand, the time to estimate the hub set for the same AND-composed layer includes time it takes to apply the proposed heuristics using the layer-wise hub sets.

The Naive or Single Graph Approach: Table 9.3 shows that the naive approach of intersecting the layer-wise degree or closeness centrality based hub sets will not guarantee a highly accurate estimated hub set for the AND-composed layers, due to the presence of a large number of false positives. Absence of common immediate neighboring nodes and common shortest paths between nodes across the layers may lead to such low accuracies with the naive approach. However, we observed that the Accident multiplex layers have similar topology due to which the naive approach gives relatively better accuracies as most of the layer-wise hubs are also hubs in the composed layers (Table 9.4).

AND-Composed Layers	Degree Centrality	Closeness Centrality
$G_{m1ANDm2}$	59%	**43.3%**
$G_{m1ANDm3}$	67.9%	55.4%
$G_{m2ANDm3}$	54.4%	**48.1%**
$G_{m1ANDm2ANDm3}$	**14.1%**	**13.5%**
Overall	**48.9%**	**40.1%**

Table 9.3: Low Accuracies of the Naive Approach to estimate AND-Composition Hub Sets (IMDb multiplex)

AND-Composed Layers	Degree Centrality	Closeness Centrality
$G_{a1ANDa2}$	84.8%	93%
$G_{a1ANDa3}$	82.6%	82.1%
$G_{a2ANDa3}$	85.4%	93.3%
$G_{a1ANDa2ANDa3}$	79.2%	87.4%
Overall	83%	88.9%

Table 9.4: Similar Topology Across Layers leading to Good Accuracies of the Naive Approach to estimate AND-composition Hub Sets (Accident multiplex)

Estimating Degree Centrality based Hubs: Here we empirically evaluate the performance of the three degree-based hub estimation heuristics.

Performance of Heuristic DC1: In DC1, the average degree estimate for an AND-composed layer is obtained by taking the minimum of the two layer-wise average degrees. This heuristic generates only those common layer-wise hubs that share more than this estimated number of neighbors across layers, thus striking out the possibility of any false positive's presence from the estimated hub sets. Table 9.5 and 9.6 show that the overall accuracy of the estimated hub sets is **79.5%** and **82.8%** for the accident and IMDb multiplexes, respectively. Moreover, there is an overall saving of **70.8%** and **41.9%** in computation time for generating the hub sets of accident and IMDb multiplexes, respectively.

AND-Composed Layer	Accuracy	Hub Set Generation Time (secs)	
		Actual	Estimated by DC1
$G_{a1ANDa2}$	78.6%	0.0523	0.0166
$G_{a1ANDa3}$	77.5%	0.0423	0.0152
$G_{a2ANDa3}$	85.7%	0.0711	0.0152
$G_{a1ANDa2ANDa3}$	76.4%	0.0458	0.0147
Overall	**79.5%**	0.2115	0.0618 **(70.8%↓)**

Table 9.5: Effective Performance of DC1: High Accuracies and Lower Hub Set Generation Times (Accident Multiplex)

AND-Composed Layer	Accuracy	Hub Set Generation Time (secs)	
		Actual	Estimated by DC1
$G_{m1ANDm2}$	88.2%	0.0597	0.0302
$G_{m1ANDm3}$	74.6%	0.0681	0.0483
$G_{m2ANDm3}$	82.4%	0.0634	0.0385
$G_{m1ANDm2ANDm3}$	85.9%	0.0492	0.0226
Overall	**82.8%**	0.2403	0.1396 **(41.9%↓)**

Table 9.6: Effective Performance of DC1: High Accuracies and Lower Hub Set Generation Times (IMDb Multiplex)

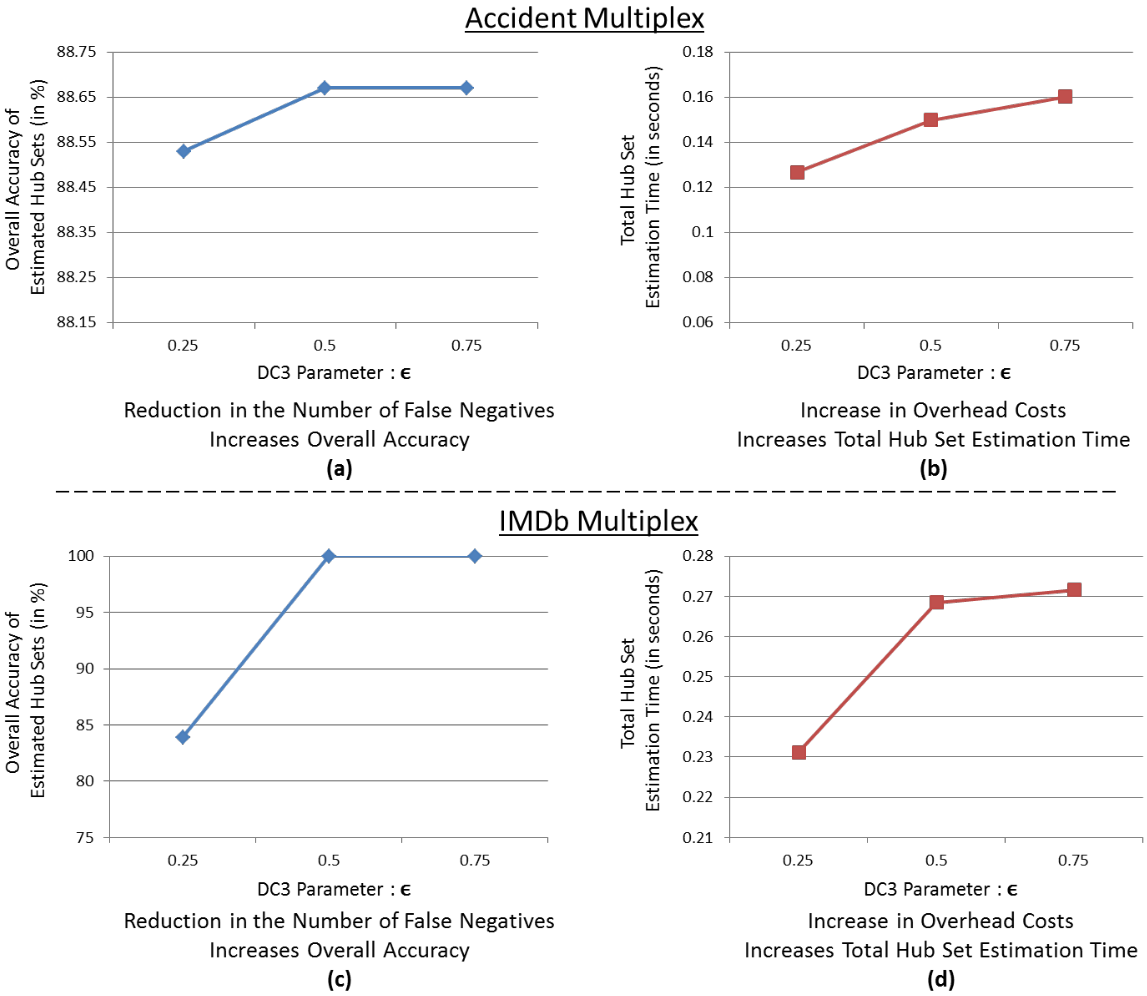
Note that for IMDB the overall accuracy improved from **48.9%** in the naive scheme to **82.8%**. However, the accuracy for the Accident multiplex decreased. This is because the estimated average degree was far larger than the actual average degree of the AND-composed networks. To solve this issue we apply heuristic DC2. An important point to be noted here is that the estimated average degree for Accident multiplex composed layers is not that good due to which few layer-wise common hubs that are also hubs in the composed layers are eliminated leading to a lower overall accuracy (79.5%) as compared to the naive approach (83%).

Performance of Heuristic DC2: Table 9.7 shows that the improved average degree estimate for the AND-composed layers can also improve the accuracy. Using heuristic DC2, increases the overall accuracy from **79.5% to 83.04%** for the Accident Multiplex. Similarly, the accuracy of estimated hub set for IMDb Multiplex increases from from **82.8% to 83.9%**. The proximity of this estimate to the actual average degree allows the generation of some common layer-wise hubs that were excluded by DC1, however the computational costs increase. Therefore, for instance, in case of the Accident multiplex hub set estimation process the overall savings in computational time falls from 70.8% to 58.4%.

AND-Composed Layer	Average Degree		% Change
(Actual Average Degree)	$DC1_{est}$	$DC2_{est}$	in Accuracy
$G_{a1ANDa2}$ (11.2)	14.92	12.988	5.2%↑
$G_{a1ANDa3}$ (10.18)	14.92	12.847	4.4%↑
$G_{a2ANDa3}$ (14.35)	16.44	15.257	1.6%↑
$G_{a1ANDa2ANDa3}$ (9.28)	14.92	12.045	2.7%↑
Overall	–	–	**3.5%↑**

Table 9.7: Improved Accuracies of DC2 over DC1 (Accident Multiplex)

Performance of Heuristic DC3: To consider the case where non-hub layer-wise nodes become hubs in the AND-composed layer, few low degree nodes from each layer are maintained such that their degree is at least $(1 - \epsilon)$ times the individual layer's average degree, where $0 \leq \epsilon \leq 1$. Figure 9.3 (a) and (c) show that by increasing the value of ϵ the overall accuracy increases as the number of false negatives are reduced. However, higher the value of ϵ, more is the number of layer-wise non-hubs carried forward to the estimation process. Therefore, this increased overhead cost increases the time to estimate hub sets (Figure 9.3 (b) and (d)).



Reduction in the Number of False Negatives Increases Overall Accuracy
(a)

Increase in Overhead Costs Increases Total Hub Set Estimation Time
(b)

Reduction in the Number of False Negatives Increases Overall Accuracy
(c)

Increase in Overhead Costs Increases Total Hub Set Estimation Time
(d)

Figure 9.3: Performance of DC3 with respect to the parameter ϵ

Figure 9.3 (c) shows that the average degree estimate for the IMDb multiplex is good enough to give a perfectly accurate estimate for an $\epsilon = 0.5$. However, the average degree estimate becomes a bottleneck in the case of Accident multiplex due to which even with increasing ϵ, the rate of increase in the overall accuracy is low (Figure 9.3 (a)). A better average degree estimate in these cases will prove to be helpful.

The overall accuracy and total hub set estimation times shown in each cell for the three proposed heuristics in the Summary Table 9.8 justify that there is an evident trade-off between accuracy and savings in the computational costs.

		DC3		
DC1 Accuracy Time (secs)	DC2 Accuracy Time (secs)	$\epsilon = 0.25$ Accuracy Time (secs)	$\epsilon = 0.5$ Accuracy Time (secs)	$\epsilon = 0.75$ Accuracy Time (secs)
Accident Multiplex				
79.5% 0.0618	83.04% 0.088	88.5% 0.1268	88.7% 0.1499	88.7% 0.1602
IMDb Multiplex				
82.8% 0.1396	83.9% 0.211	83.9% 0.2312	100% 0.2685	100% 0.2716

Table 9.8: Summarizing the Performances of the Three Degree based Hub Estimation Heuristics

Estimating Closeness Centrality based Hubs using *Heuristic CC1*: In every layer, high degree neighbors for each high closeness centrality node are maintained. The intuition is that if a common high closeness centrality node shares high degree neighbors across layers that are also part of the hub set estimated by heuristic DC2, then its chances of being accessible via less number of hops from every other node in AND-composed layer increase. Table 9.9 and 9.10 show that for both accident and IMDb multiplexes, this heuristic estimates hub sets that have an overall accuracy of **73.8%** and **66.5%**, respectively. Moreover, this process leads to a saving of at least **30%** in computation time.

AND-Composed Layer	Accuracy	Hub Set Generation Time (secs)	
		Actual	Estimated by CC1
$G_{a1ANDa2}$	73.1%	0.3086	0.2028
$G_{a1ANDa3}$	68.9%	0.2834	0.2004
$G_{a2ANDa3}$	78.2%	0.345	0.2017
$G_{a1ANDa2ANDa3}$	75.1%	0.237	0.2051
Overall	**73.8%**	1.174	0.81 **(31%↓)**

Table 9.9: Effective Performance of CC1: High Accuracies and Lower Hub Set Generation Times (Accident Multiplex)

AND-Composed Layer	Accuracy	Hub Set Generation Time (secs)	
		Actual	Estimated by CC1
$G_{m1ANDm2}$	60.4%	2.0534	1.5153
$G_{m1ANDm3}$	71.3%	2.6168	1.5255
$G_{m2ANDm3}$	70.1%	2.0432	1.5159
$G_{m1ANDm2ANDm3}$	64.1%	2.029	1.5071
Overall	**66.5%**	8.7424	6.0637 **(30.64%↓)**

Table 9.10: Effective Performance of CC1: High Accuracies and Lower Hub Set Generation Times (IMDb Multiplex)

The similar topology among the Accident Multiplex layers means that most of the shortest paths among the node pairs across layers are common leading to the naive approach giving a higher accuracy as compared the proposed heuristic that excludes some common layer-wise hubs as it only considers shared one hop high degree neighbors. Even though this heuristic gives good accuracies for the estimated hub sets, but it can be improved by maintaining the path information to high degree nodes beyond 2-3 hops from the high closeness centrality hubs in each layer. However, as stated earlier, maintaining such longer path information will significantly increase the computational costs.

9.4 Effect of Different Parameters on the Composition Function Accuracy

Similar to the Boolean compositions for community detection, it becomes pivotal to understand the effect of different network characteristics of the MLN layers on the accuracy of the centrality composition algorithms. In this thesis, we have explored the accuracy estimates of the degree centrality composition heuristics. Depending on the accuracy estimates, one can choose whether to opt for the single graph approach or the decoupling approach. We have considered Precision as the accuracy metric.

Synthetic Data Sets (HoMLNs): Two sets of Synthetic HoMLN layers were generated for this purpose.

1. **HubMLN-SET1:** 2 initial layers (L1, L2) were generated with 1000 vertices each. In L1, there were 25 degree hubs and L2 had 50 degree hubs. 5% and 10% edges were removed from the two layers in every iteration, in order to generate 60 different pairs of layers.
2. **HubMLN-SET2:** In this case, the two initial layers (L1, L2) with 1000 vertices and 25 degree hubs were identical networks. 5% and 10% edges were removed from L1 and L2 in every iteration, respectively, in order to generate 60 different pairs of layers.

For considering degree hub based similarity between two layers, we considered two metrics here

- **Jaccard Similarity between the Binary Hub Vectors:** For every layer, we constructed a vector where i^{th} index represented whether the i^{th} vertex is a degree hub (1) or not (0). Two layers having a high value for this metric will mean the higher fraction of same nodes are degree hubs in both the layers.
- **Cosine Similarity between the Degree Vectors:** Here for every layer we construct a vector such that the degree information for each vertex in present. Thus, a high value for this metric will mean that not only most of the high

degree and low degree nodes are same, but also most of the nodes have similar number of neighbors, i.e. *most of the nodes have similar influence over the network*.

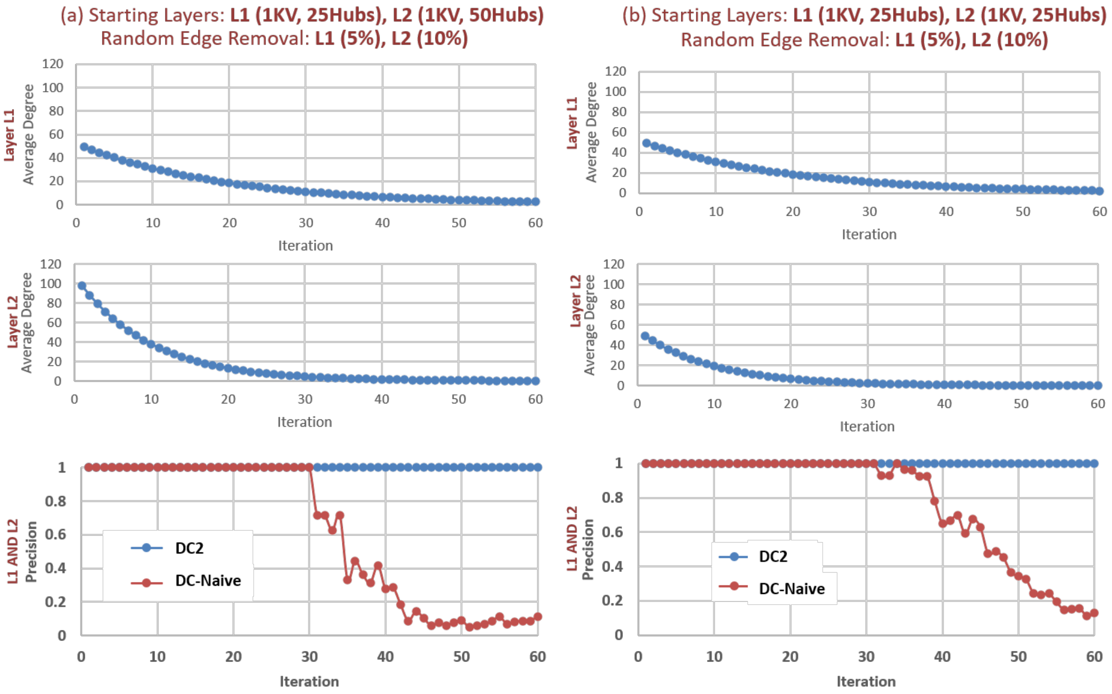


Figure 9.4: Variation in Composition Accuracy based on average layer degree

Figure 9.4 (a) and (b) show that when the average degree of the layers is low, then the accuracy of the NAIVE approach (DC-NAIVE) falls. This is due to the fact that in such a scenario the chances of common nodes having overlapping neighbors across layers decreases. Thus, number of false positives increases. However, as the DC2 is able to eliminate all the false positives, thus **the precision of DC2 is high even when the average degree of the layers is low**.

It can also be seen from Figure 9.5 (a) for HubMLN-Set1 data set, that across all iterations the number of common high degree and low degree is quite low (low Jaccard

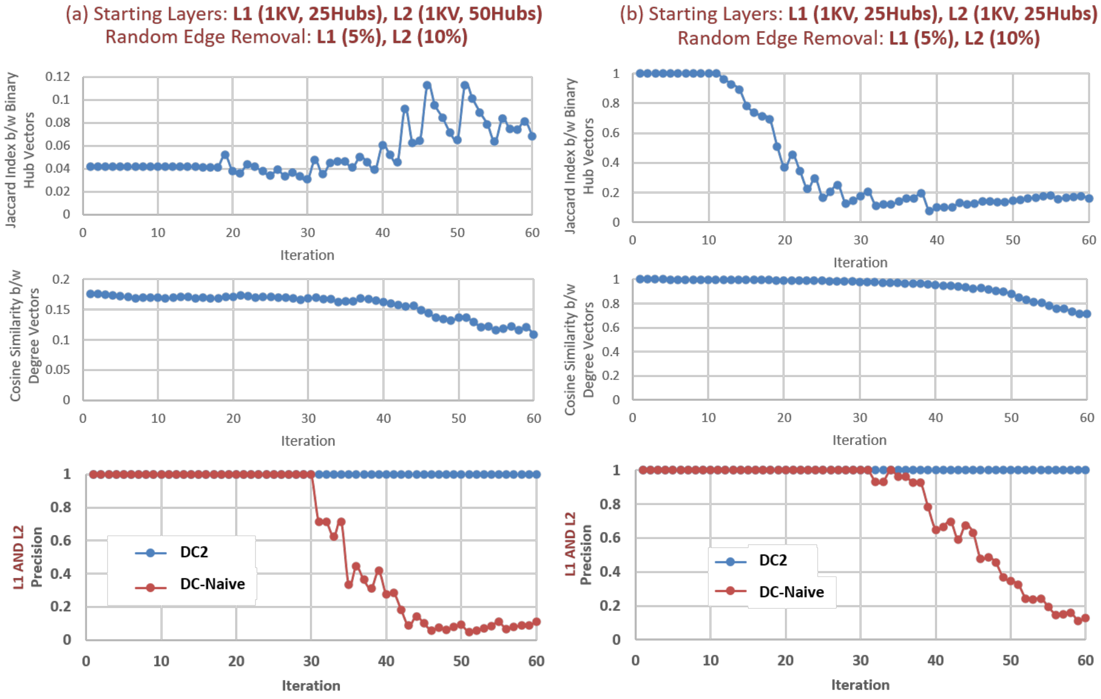


Figure 9.5: Variation in Composition Accuracy based on layer similarity

and Cosine similarity values), i.e. the layers are fairly dissimilar in nature. Even in such a scenario DC2 always gives a high precision. For the NAIVE approach, this precision falls drastically due to the presence of substantial number of false positives. For the HubMLN-Set2 data set shown in Figure 9.5 (b) the dissimilarity among the layers gradually increases upon removal of random edges, however this dissimilarity does not effect the precision results of DC2.

Therefore, in general it can be concluded that as the degree estimate of the DC2 is higher than the actual average degree, thus all false positives will be eliminated and the precision will be high irrespective of network topology of the two layers.

9.5 Conclusion and Future Work

Various heuristics have been presented and validated to efficiently estimate hubs in any conjunctively composed layer of a HoMLN. We have shown that by maintaining minimal neighborhood information along with the layer-wise hubs, it is possible to estimate good quality degree or closeness centrality based hub sets of any AND-composed layer, with an overall accuracy exceeding 80% or 70%, respectively. Moreover, we have shown through various experiments performed on real-life datasets from diverse backgrounds that our proposed heuristics lead to a saving of at least 30% in computation time. Further, such techniques eliminate the need to generate and store any composed layers, thus saving storage space too. This work has been published in 2017 [8].

This hub estimation can be extended to other centrality measures like betweenness and eigenvector and handle weighted and/or directed edges. In addition to conjunction, this composition can be extended to disjunction and negation. Centrality definition and detection in HeMLN is another research area that is currently being explored.

The next chapter discusses how the centrality measures can be used to extensively and efficiently analyse the US Commercial Airlines data set modeled using the homogeneous multilayer network.

CHAPTER 10

CENTRALITY ANALYSIS OF US AIRLINE DATA SET

In this chapter, we use the **US commercial airline data set** to illustrate analysis-driven modeling and computation. It is a data set of six US-based airlines and their flight information among US cities. This information has been collected by us from multiple sources. Although each data set is small, the choice of this data set is for the independent availability of ground truth for validation. Availability of ground truth will allow us to show the efficacy of the proposed approach along with other advantages.

10.1 Analysis-Driven MLN Modeling

To demonstrate the effectiveness of modeling we apply MLNs to answer the following key questions.

(A1) Identify 5 cities for each airline from which there is *best coverage for travel within the US?*

(A2) Can airlines be categorized into *major or minor carriers*?

(A3) Identify preferred cities for an airline to *expand its operations* taking *all* its competitors into consideration?

The analysis objectives ((**A1**) - (**A3**)) requires airline connectivity between *same set of US cities for different airline carriers.* As the entity set used for analysis is the same, but using different features, this data set should be **modeled as HoMLNs**. Typically, number of layers correspond to the number of relationships that need to be captured (one for each layer). The semantics of the analysis objectives determines the

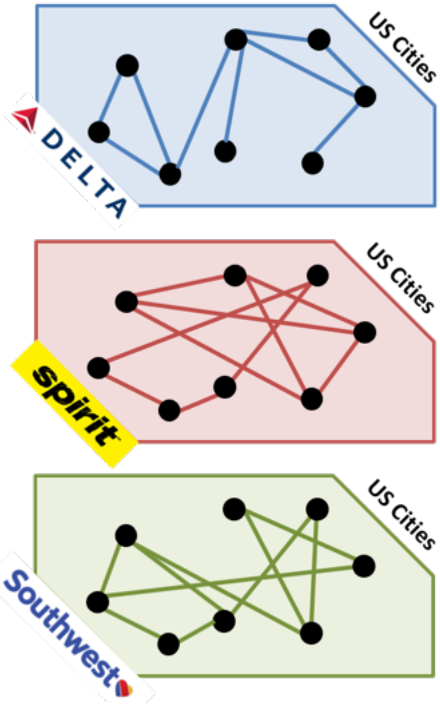


Figure 10.1: US Airline HoMLN

choice of nodes and intra-layer edges. Therefore, each **node in a layer represents a US city**. *Each layer corresponds to a single airline* and connects node pairs (cities) in case of a *direct flight* between them. For this thesis, we have chosen 6 airlines (layers) for analysis - American, Southwest, Spirit, Delta, Allegiant and Frontier. Figure 10.1 shows three layers of the HoMLN.

10.2 Mapping Analysis Objectives to Computations

Decoupling-based approach entails identifying Ψ and Θ for each detailed analysis objective (**(A1)** to **(A3)**) along with their application on layers in a specified order. Table 10.1 summarizes the mapping of each detailed analysis **(A1)** to **(A3)** to their actual computation specification (in *left* to *right* order), analysis function (Ψ) and composition function (Θ). This is used for computing the results in the experimental section (Section 12.2.)

Objective **(A1)** requires closeness centrality (*inverse of mean shortest distance*) for coverage. Categorization ((**A2**)) of the airlines can be done either using degree or closeness centrality. We have chosen degree centrality to reflect connotation of airline hubs for this. In contrast, **(A3)** requires choosing non closeness centrality hubs of the airline considering expansion and AND it with the competitor hubs to eliminate competition. The result will need additional information to rank the resulting cities.

Analysis	**Mapping**		
	Computation Order	Ψ	Θ
*US Airline (**HoMLN**)*			
(A1)	Individual layers	Hub (closeness)	
(A2)	p major airline layers; q minor airline layers	Hub (degree)	AND
(A3)	(Target Θ Competitor) airline layer pair	Hub (closeness)	AND

Table 10.1: MLN Expression for Each US Airline Analysis Objective

10.3 Experiments and Drill-Down Analysis with Validation

We compute the results for each detailed objective using the expressions shown in Table 10.1 and compare it, where possible, with independently available ground truth. This helps validate both the modeling and analysis aspects of the approach proposed in this thesis.

The HoMLN was built for 290 US cities using flights active until February 2018, with number of intra-layer edges being 746 (American), 717 (Southwest), 688 (Delta), 346 (Frontier), 189 (Spirit) and 379 (Allegiant). Closeness centrality handles the coverage aspect of **(A1)**. For each layer (airline), the cities with closeness centrality more than the average (shown in parenthesis) were identified as *closeness* hubs for coverage. 214 American (0.2622), 85 Southwest (0.04995), 213 Delta (0.2552), 77

Frontier (0.0384), 37 Spirit (0.009995) and 113 Allegiant (0.0701) hubs were obtained and *ranked based on closeness centrality value.*

Top 5 hubs (*higher rank, fewer flights required for coverage, more central city*)) were identified for each airline which corresponds to **(A1)** answers. **For all 6 airlines, the ground truth obtained from [130] matched our results.** In Table 10.2 we have listed top 5 hubs for each airline. As a byproduct, it is interesting to see common hubs (highlighted) between airlines which is also verified by the ground truth.

We used Gephi to obtain a visualization (for drill-down), where the node sizes depicted the importance of a node. In figure 10.2, the larger sized nodes depict the cities from where most of the other parts of the US can be covered in lesser number of flights (high closeness centrality) provided American Airlines is the chosen carrier.

American
Dallas
Chicago
Charlotte
Philadelphia
Phoenix

(a)

Southwest
Chicago
Denver
Baltimore
Dallas
`Las Vegas`

(b)

Delta
Atlanta
Minneapolis
Detroit
Salt Lake City
New York

(c)

Frontier
Denver
`Orlando`
Austin
`Las Vegas`
Philadelphia

(d)

Spirit
Fort Lauderdale
`Las Vegas`
`Orlando`
Detroit
Chicago

(e)

Allegiant
Orlando
Tampa
Las Vegas
Phoenix
Fort Myers

(f)

Table 10.2: **(A1)**: Cities With Maximum US Travel Coverage

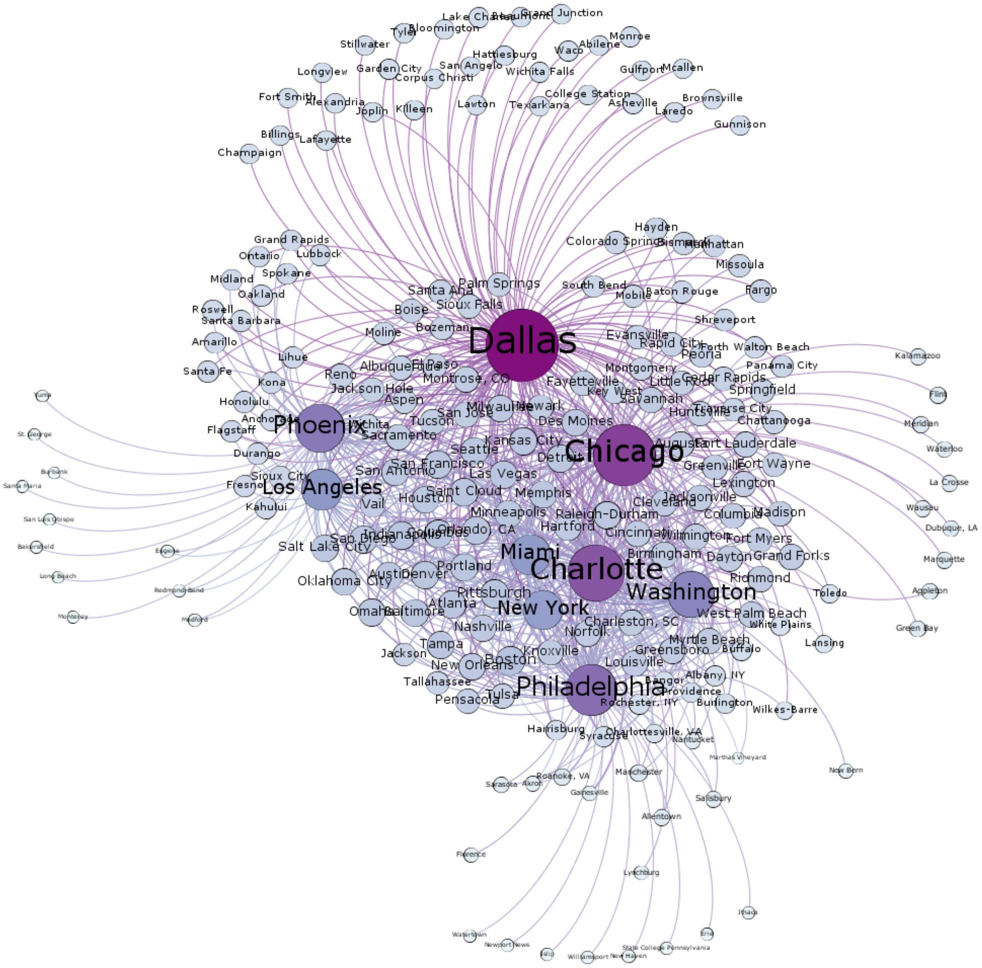


Figure 10.2: American Airlines Graph with respect to US Travel Coverage

For **(A2)**, one of the intuitive ways of categorizing an airline into a *major or minor* airline is to determine the *fraction of US city pairs that it directly connects.* Clearly, more the average degree of a layer, more number of cities the corresponding airline is operating at. Thus, ordering the airlines by the *the average degree of the corresponding MLN layer* categorizes - *American, Southwest and Delta* as **Major Airlines**; *Allegiant, Frontier and Spirit* as **Minor Airlines**. This classification validity can be easily verified using **fleet size, revenue and passengers carried in**

a year from [131] which is borne out by the analysis. There is a clear divide between the two inferred categories of airlines. Additionally, we detected the *common important operating bases* using the *higher than average degree criteria* for both major and minor airlines. We found that, in general most of *the cities* for minor airlines (Tampa, Orlando, Fort Lauderdale, Cleveland, ...) are *smaller cities* in terms of *population and GDP per capita* as compared to the major airline (Dallas, Chicago, Los Angeles, New York, ...), (shown in Table 10.3), showing that these 2 categories of airlines focus on *different types of regions and demographics within the US*.

City	**Population**	**GDP Per Capita**
New York	8.623 million	71,084 USD
Los Angeles	4 million	67,763 USD
Chicago	2.716 million	61,170 USD
Phoenix	1.626 million	44,534 USD
Dallas	1.341 million	64,824 USD
Atlanta	486,290	56,840 USD
Average	***3.13 million***	***61,036 USD***

(a) Major Airline Hubs (*Larger Population, Higher Spending Power*)

City	**Population**	**GDP Per Capita**
Chicago	2.716 million	61,170 USD
Columbus	879,170	63,822 USD
Cleveland	385,525	30,673 USD
Tampa	385,430	41,222 USD
Orlando	280,257	45,807 USD
Fort myers	79,943	32,784 USD
Average	***0.8 million***	***45,913 USD***

(b) Minor Airline Hubs (*Smaller Population, Lower Spending Power*)

Table 10.3: Analysis **(A2)**: Degree Hubs for airline categories

For **(A3)**, we chose Allegiant as the target minor airline which is considering expansion. The remaining airlines are chosen as competitors. Intuitively, among non-hubs, those *cities must be considered for expansion* that a) have fair amount of coverage (high values of closeness centrality) (*to reduce cost of expansion*) and b) do not have large operations (low closeness centrality) for the competitor airlines (*to minimize competition*). Thus, from the high closeness centrality cities of the target airline, we removed the actual hubs first, followed by all those cities that are also high closeness in each of the other competitor airlines. The pruned set was further filtered by using additional external information like population to rank them in order to bring in the aspect of ticket sales likelihood.

Allegiant *Vs. All*
Grand Rapids
Elko
Montrose

Table 10.4: **(A3)**: Expansion Cities

Table 10.4 shows the resulting set of cities where Allegiant Airline can potentially expand its operations. Validation of this process was established using the fact that **Grand Rapids is one of the cities that will be converted to a hub by Allegiant from July 6, 2019** [132].

This shows that *MLN analysis results can be effectively used (augmented with additional information, where needed)* to make real-world business decisions.

10.3.1 Efficiency Analysis of the Decoupling Approach

Experimental Set up: We used a quad-core 8th generation Intel i7 processor Linux machine with 8 GB memory for all of our analysis. The layer-wise results (hubs) are generated *once* and can be done in *parallel.* Thus, this one time cost is bounded by the layer that takes maximum time. Moreover, the cost of composing the partial results using Boolean AND (HoMLN Hubs) is *significantly less.* For HoMLN analysis, we compare the total computational cost of the decoupling approach and the traditional single graph approach which includes the *time to generate the combined layer* followed by generating the degree/closeness hubs.

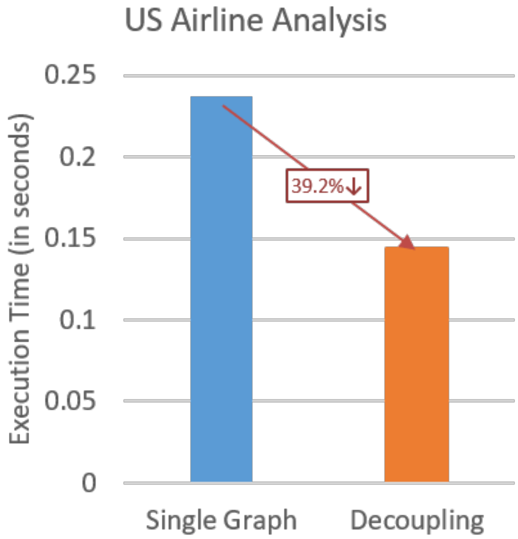


Figure 10.3: Efficiency of Decoupling Approach for US Airline HoMLN Analysis

In total, 7 Boolean AND compositions are required as per Table 10.1 - 2 for **(A2)** and 1 for each of the (Allegiant, Competitor) layer pairs in **(A3)**. The total computation cost with the single graph approach is 0.215 seconds, whereas for the decoupling approach the total time is 0.143 seconds leading to a **reduction of 33.6% in computation time** for just **11% of the total possible analysis** (7 out of 2^6), as shown in Figure 10.3.

In summary, the experiments on the US Airline validate the **MLN decoupling approach from an efficiency perspective**.

CHAPTER 11

A COMMUNITY DEFINITION FOR HETEROGENEOUS MLNs AND A NOVEL APPROACH FOR ITS EFFICIENT COMPUTATION

Heterogeneous MLNs currently lack a community definition and concomitant algorithms. This is where the contribution of this chapter is directed in generalizing the established community definition for HeMLNs along with an efficient computation model using a novel approach. Among the alternatives used in the literature modularity-based community definition (which hierarchically maximizes the concentration of edges within modules (or communities) compared with random distribution of links between all nodes regardless of modules), seems to have consensus for a single graph along with several implementations that are used widely (e.g., Louvain [67]).

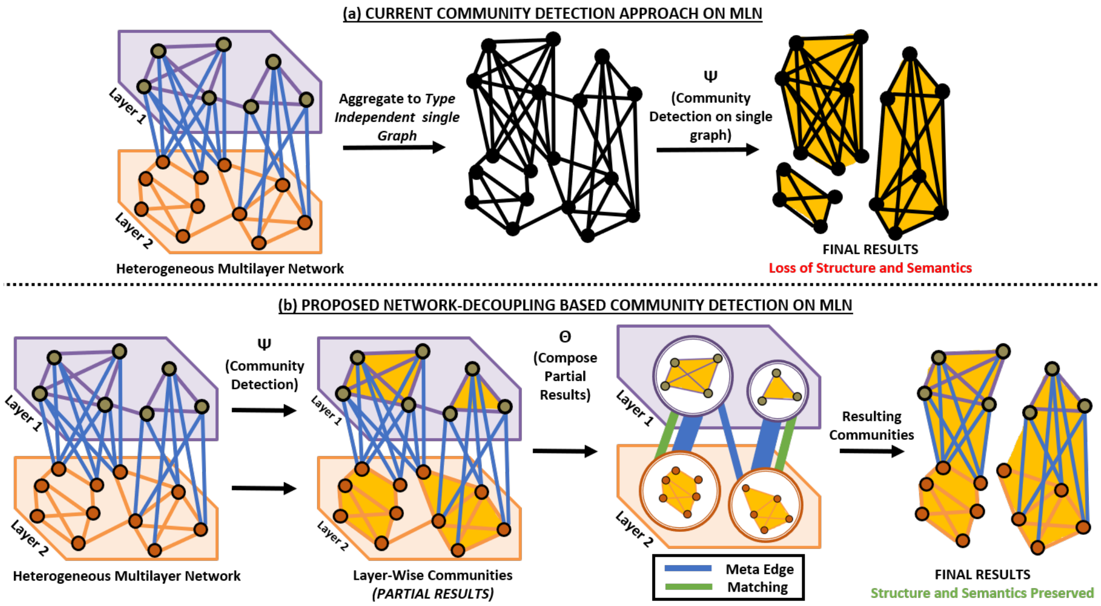

(a) CURRENT COMMUNITY DETECTION APPROACH ON MLN
Layer 1
Layer 2
Heterogeneous Multilayer Network
Aggregate to Type Independent single Graph
Ψ
(Community Detection on single graph)
FINAL RESULTS
Loss of Structure and Semantics
(b) PROPOSED NETWORK-DECOUPLING BASED COMMUNITY DETECTION ON MLN
Layer 1
Layer 2
Heterogeneous Multilayer Network
Ψ
(Community Detection)
Layer-Wise Communities
(PARTIAL RESULTS)
Θ
(Compose Partial Results)
Meta Edge
Matching
Resulting Communities
FINAL RESULTS
Structure and Semantics Preserved

Figure 11.1: Traditional Lossy Approach Vs. Structure and Semantics Preserving Approach

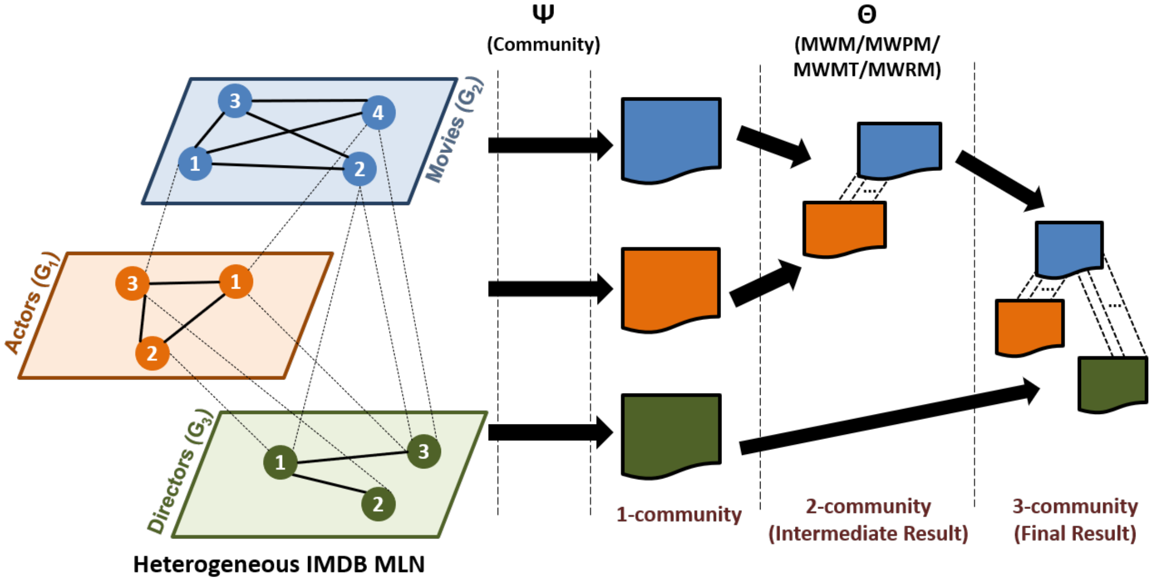


Figure 11.2: Decoupling Approach to Compute HeMLN 3-community Expressed as $((G_2\ \Theta_{2,1}\ G_1)\ \Theta_{2,3}\ G_3);\ \omega_e$*

For a simple graph, the current community definition preserves its structure and semantics in terms of node/edge labels and relationships. Preserving the structure of a community of a MLN (especially HeMLN) entails preserving its multilayer network structure as well as semantics of node/edge types, labels, and importantly inter-layer relationships. In other words, each HeMLN community should be a MLN in its own right. Contributions of this chapter are:

- Definition of *structure and semantics preserving* community for a HeMLN and an approach for its efficient computation (Section 11.3),
- A composition function for formalizing the *decoupling approach* for HeMLN community detection algorithms (Section 11.4),

*Technically, this should be expressed as $((\Psi(G_2)\ \Theta_{2,1}\ \Psi(G_1))\ \Theta_{2,3}\ \Psi(G_3).)$ However, we drop Ψ for simplicity. In fact, Θ with its subscripts is sufficient for our purpose due to pre-defined precedence (left-to-right) of Θ. We retain G for clarity of the expression. ω_e is a weight metric discussed in Section 11.5.

- Two new bipartite pairing algorithms for composing layers which are more appropriate for HeMLN communities (Sections 11.3.2 and 11.4.2.) Also, identification of useful weight metrics and their relevance (Section 11.5),
- Mapping analysis objectives to the proposed community definition (Section 12), and
- Experimental analysis using the IMDb and DBLP data sets to establish the structure and semantics preservation (drill-down capability) of the proposed approach along with efficiency aspects (Section 12.2.)

11.1 Semantics Preservation And Efficiency

Lack of a community definition for a HeMLN has resulted in various *ad hoc* approaches to leverage the single graph community definition and algorithms for its detection. As a consequence, although modeled as MLNs for semantic superiority, they are reduced to a single graph for the purpose of community detection. There have been some attempts to detect (rather extract) communities on the MLN. [92] proposes multilayer extraction for *Homogeneous MLNs*, such as Multilayer social network, transportation network, and collaboration network. They use the notion of vertex neighborhoods with a refinement procedure, to produce a subfamily of high-scoring vertex layer sets. [66] focuses on higher-order network flows *again in Homogeneous MLNs*. All of these approaches do not preserve the structure of the HoMLNs in their results.

Our goal and focus in this chapter has been: i) map analysis objectives to appropriate graph properties and their computation and ii) drill down analysis of the results to understand analysis results (using preserved structure and semantics of computed results. The importance of these are discussed below.

11.1.1 Structure and Semantics Preservation

Current approaches, such as type-independent [44] and projection-based [82, 76], do not accomplish structure and semantics preservation as they aggregate (or collapse) layers into a simple graph in different ways. More importantly, aggregation approaches are likely to result in some information loss [28], distortion of properties [28], or hide the effect of different entity types and/or different intra- or inter-layer relationships as elaborated in [91]. Furthermore, structure and semantics preservation is critical for understanding a HeMLN community and for drill-down analysis of results.

From an analysis perspective, lack of structure and semantics makes the drill down extremely difficult (or even impossible) and hence the understanding and visualization of results. Our computation results clearly show the community structure and how easy it is to drill down to see patterns in terms of original labels and relationships.

Figure 11.1 illustrates the difference between the current approaches and our proposed approach. Fig. 11.1 a) shows type-independent aggregation[†] of two layers into a single graph on which extant community detection is applied. As can be seen, **both structure as well as entity and relationship labels – shown as colored nodes and edges – are lost in the resulting communities.** In contrast, the Fig. 11.1 b) shows the same layers and community detection using the definition and the decoupling approach proposed in this thesis. As there is no aggregation, both structure and semantics are preserved.

11.1.2 Decoupling Approach For HeMLNs

As mentioned earlier, decoupling approach is the equivalent of "divide and conquer" for MLNs. Research on modeling a data set as a MLN *and* computing on the whole MLN has not addressed efficiency issues [92]. Decoupling requires partitioning

[†] Other aggregation approaches have the same problem.

(derived from the MLN structure) and a way to compose partial (or intermediate) results. Here, we identify a composition function (referred to as Θ, see Figure 11.2) that is appropriate for efficient community detection (referred to as Ψ, see Figure 11.2) on MLNs.

Figure 11.2 shows the **proposed decoupling approach**. Three layers and some inter-layer connections are shown. HeMLN community computation is accomplished by combining communities from two layers of a HeMLN using a composition function (Θ) and is extended to k layers by composing the result with additional layers one at a time. Figure 11.2 also shows how a 3 layer HeMLN community is expressed for computation. This approach of partitioning and composing partial results is central to efficiency of computation as elaborated in Section 12.2.

11.2 HeMLN Definitions

A **graph** G is an ordered pair (V, E), where V is a set of vertices and E is a set of edges. An edge (u, v) is a 2-element subset of the set V. In this thesis, we only consider graphs that are undirected.

A **multilayer network**, $MLN(G, X)$, is defined by two sets of graphs: i) The set $G = \{G_1, G_2, \ldots, G_N\}$ contains graphs of N individual layers $L = \{L_1, L_2, \ldots, L_N\}$ as defined above, where $G_i(V_i, E_i)$ is defined by a set of vertices, V_i and a set of edges, E_i. An edge $e(v, u) \in E_i$, connects vertices v and u, where $v, u \in V_i$ and ii) A set $X = \{X_{1,2}, X_{1,3}, \ldots, X_{N-1,N}\}$ consists of bipartite graphs. Each graph $X_{i,j}(V_i, V_j, L_{i,j})$ is defined by two sets of vertices V_i and V_j, and a set of edges (also called links or inter-layer edges) $L_{i,j}$, such that for every link $l(a, b) \in L_{i,j}$, $a \in V_i$ and $b \in V_j$, where V_i (V_j) is the vertex set of graph G_i $(G_j.)$

For a HeMLN, X is explicitly specified. Without loss of generality, we assume unique numbers for nodes across layers and disjoint sets of nodes across layers. We define a *k-community* to be a multilayer community where communities from k distinct connected layers of a HeMLN are combined in a *specified order* as shown in Figure 11.2. Our proposed algorithm using the decoupling approach for finding HeMLN communities can be described as follows with reference to Figure 11.2:

(i) First, use the function Ψ (here community detection) to find communities in each of the layers individually (can also be done in parallel),

(ii) For any two chosen layers, use the partial/intermediate results from these layers and apply the composition function Θ (bipartite graph matching) using the meta edges (whose weight is denoted by ω) between the layers to compute the result. For HeMLN community detection, a bipartite pairing that maximizes modularity (or total weight of the meta edges) is used for Θ.

(iii) The binary composition of step ii) is applied for determining a k-community for a specified order of layers.

Figure 11.2 illustrates the decoupling approach for specifying and computing a HeMLN 3-community from partial results. It illustrates how a set of distinct communities from a layer is used for computing a 2-community (G_2 $\Theta_{2,1}$ G_1) for 2 layers and further a 3-community ((G_2 $\Theta_{2,1}$ G_1) $\Theta_{2,3}$ G_3) for 3 layers using partial results. 1-community is the set of communities generated for a layer L_i using its G_i (simple graph.) We use L_i and G_i interchangeably in the rest of the chapter.

We can now define the problem addressed in this chapter. *For a given HeMLN and a set of analysis objectives, determine the appropriate triad of* Ψ*,* Θ*, and* ω*, and a* k*-community expression for computing* ***each objective****. For this chapter, community is used for* Ψ *and bipartite match algorithms for* Θ *for HeMLN community detection along with defining and identifying* ω*.*

11.3 Community Definition for a HeMLN

The intuition behind a HeMLN community is first explained using an example. The IMDb data set captures movies, TV episodes, actors, directors, and other related information, such as rating, genre, etc. This is a large data set consisting of movie and TV episode data from their beginnings. This data set can be modeled and analyzed in multiple ways as well for different purposes. For the IMDb data set, consider the HeMLN shown in Figure 11.2 that has the following three layers: i) *Actors* layer – connects actors who act in similar genres frequently (intra-layer edges.), ii) *Directors* layer – connects directors who direct similar movie genres frequently, and iii) *Movies* layer – connects movies within the same rating range. The inter-layer edges depict *acts-in-a-movie, directs-a-movie* and *directs-an-actor*.

Consider the analysis objective *"Find dense groups of actors and directors that have high/strong interaction/coupling with each other"* Note that, individually, the actor and director layers can only compute dense groups of actors or directors, who act in or direct similar genre, respectively. The connection (or coupling) between directors and actors only come from inter-layer edges. It is only by identifying the proper meta edges ***and*** preserving the structure of both the communities in actor and director layers as well as the inter-layer edges, can we compute and drill down the answer that indicates the semantics of which actor groups are paired with which director groups. The inter-layer edges preserve the relationships of individual actors and directors as well. Preservation of structure (inter-layer edges) and semantics (labels) is critical for drilling-down to understand the results.

Clearly, multiple strong interactions can exist between groups of actors and directors (in general, among communities from different layers.) A specific co-actor group may be favorites of one or more director groups based on genre or other characteristics, and vice-versa. So, any MLN community definition needs to include these multiple couplings (unlike traditional bipartite matching which identifies only unique pairs) in a way similar to the coupling between nodes in a single layer community definition. In addition, it may also be important, from an analysis perspective, and useful to couple these groups (or communities) using different community characteristics as well. An analysis objective may also want to use or specify different community interactions as *preferences* to meet an analysis objective. As an example, one may be interested in groups (or communities) where the *most important* actors and directors (characterized in terms of their degree) interact rather than the actor community as a whole. Based on this observation, we have proposed two new bipartite matching algorithms in this chapter.

Note that the community definition and detection research in the literature for homogeneous MLNs [6, 7] are not applicable to HeMLNs as each layer has *different sets and types of entities* with *inter-layer edges* between them. It is important that this formulation of communities preserves entity and feature types as compared to other alternatives proposed in the literature.

Hence, the challenge for the definition of a HeMLN community is to not only keep it consistent with the widely-accepted community definition, but also provide alternatives to accommodate broader analysis objectives. This, in conjunction with structure and semantics preservation, will enhance the utility of this modeling as well as analysis efficiency. In the following sections, we provide such a definition, its relationship to modularity for illustration, its efficient computation with algorithms based on the decoupling approach, and importantly demonstrate its usage with respect to diverse analysis objectives later in the chapter.

11.3.1 Formal Definition of a HeMLN Community

A **Community Bipartite Graph** or $\mathbf{CBG}_{i,j}(U_i, U_j, L'_{i,j})$ between graphs (layers) G_i (L_i) and G_j (L_j) is defined as the graph with disjoint and independent nodes U_i (U_j) corresponding to each community from L_i (L_j), respectively, represented as a single meta node and $L'_{i,j}$ being the set of single meta edges between the nodes of U_i and U_j (or bipartite graph edges.) whose weight (ω) corresponds to the number (or strength) of the inter-layer edges between the corresponding nodes. For a layer L_i, a **1-community** is the set of communities identified on the graph corresponding to that layer using any of the community detection algorithms.

For two layers L_i and L_j, and their inter-layer edges $X_{i,j}$, a **HeMLN 2-community** for G_i and G_j is defined as the **community bipartite graph meta node pairs** that maximize total inter-layer edge weights (along with the overall modularity) between the two CBG meta node sets. This pairing (or coupling) can be defined in multiple ways. We start with the coupling being defined as the traditional bipartite matchings that maximizes total edge weight and extend it.

A **HeMLN k-community** is the application of the above binary definition for k layers in a specified order of layers using the previously computed **(k-1)-community**..

11.3.2 Need For Alternative Bipartite Match Algorithms

Traditional bipartite graph matching with edge weights and different size node sets compute pairings (or matchings) that produce maximum weight (termed MWM or Maximum Weight Match [97]) or that produce maximum number of pairings with maximum weight (MWPM or Maximum Weight Perfect Match [133]). A constraint used in all traditional bipartite matches is that the resulting matches/pairs are unique.

However, for a HeMLN community definition that maximizes the coupling between bipartite graphs, the above can be used directly *only if one is interested in unique* *pairings from an analysis perspective.* However, for many analysis using communities, we need pairings (couplings) without the above restriction as one-to-many pairings (from either side) makes sense from an analysis perspective. For the analysis of the IMDb data set mentioned earlier, there is no reason to restrict an actor group to pair with only one director group if another coupling is equally strong or an alternate coupling produces a higher total weight. Hence, we need to: i) relax the unique match restriction to increase total edge weight for the *same number* of pairings and ii) deal with (or include) **ties** of edge weights incident on the same nodes of unique pairings (instead of choosing one randomly!.) These two will maximize $\sum w(e)$ of the CBG across all edges and minimize the number of such pairs. These relaxations make sense semantically as well as a community may have a stronger coupling with multiple communities. These are global maximums as they are derived from the traditional pairings of MWM and MWPM. We believe that MWM (and the variants we are proposing) comes closest to modularity semantics for HeMLN communities by maximizing the inter-layer connectivity for the communities in contrast to inter-layer connectivity of other communities.

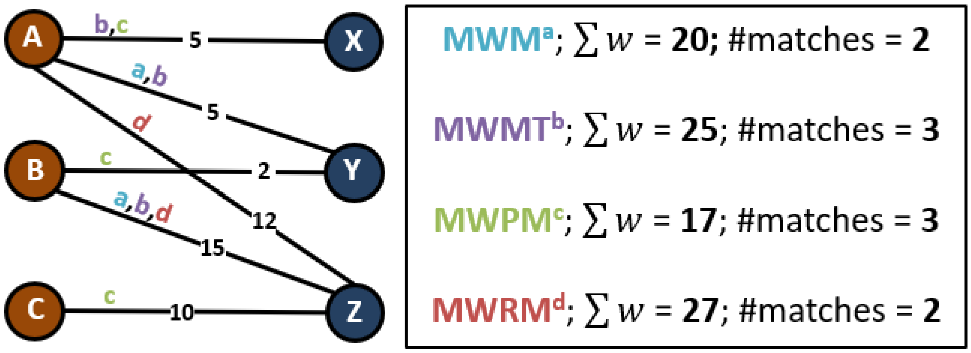

A
B
C
X
Y
Z
b,c
5
a,b
5
d
c
2
a,b,d
12
15
c
10
MWM^a; ∑ w = 20; #matches = 2
MWMT^b; ∑ w = 25; #matches = 3
MWPM^c; ∑ w = 17; #matches = 3
MWRM^d; ∑ w = 27; #matches = 2

Figure 11.3: Illustration of Traditional and Relaxed Pairings on a weighted bipartite graph

Figure 11.3 provides an example of a bipartite graph to illustrate the above discussion. MWM (Maximum Weight Matching); MWMT (MWM with Ties); MWPM (Maximum Weight Perfect Match); MWRM (Maximum Weight with Relaxed Matching). Relaxing the unique pair constraint can increase the maximum weight if alternative pairings exist and they are not unique. In addition, the presence of ties results in additional pairings to maximize modularity under our relaxation. Our matchings are termed MWMT (Maximum Weight Matching with Ties) and MWRM (Maximum Weight with Relaxed Matching by removing the unique pairing constraint). All of the above are commutative and non-associative. Refer to [12, 134] for other possibilities.

Although, in the above definition, we have chosen the weight of the meta edge of the bipartite graph from a modularity perspective, this weight can reflect other participating community characteristics to accommodate a family of HeMLN community definitions as elaborated in Section 11.5.

Most importantly, unlike current alternatives for community of a MLN, there is no need for aggregating or collapsing a MLN into a single graph in our definition and computation, thereby avoiding any kind of information loss. The representation of a HeMLN community preserves the MLN structure along with semantics (node and edge labels, both intra and inter.)

11.3.3 HeMLN k-Community Computation

This section outlines the computation of the HeMLN community definition given above for an arbitrary HeMLN. Although the above definition is commutative, it is not associative. Hence, different HeMLN communities can be obtained depending on the order used for its computation The order of community computation is derived mainly from analysis objectives and is mapped to community composition expressions.

A ***1-community*** for a given layer is computed using any community detection algorithm. The **community bipartite graph** $\boldsymbol{CBG_{i,j}(U_i,\ U_j,\ L'_{i,j})}$ is computed using $X_{i,j}$ once the 1-community for layers L_i and L_j are computed.

A ***2-community***, corresponding to layers L_i and L_j, is computed on the community bipartite graph $CBG_{i,j}(U_i,\ U_j,\ L'_{i,j})$. A 2-community is a set of tuples each with a pair of elements $< c_i^m, c_j^n >$, where $c_i^m \in U_i$ and $c_j^n \in U_j$, that satisfy *one of the weighted bipartite matching algorithms discussed* (composition function Θ defined in Section 11.2) for the bipartite graph of U_i and U_j, **along with** the set of inter-layer edges between them (denoted $x_{i,j}$.) The pairing is done using the specified pairing alternative (one of MWM, MWPM, MWRM, or MWMT) to obtain pairs of communities and their inter-layer links for L_i and L_j. It is possible for several matching algorithms may give the same result.

A ***k-community*** for k layers of a HeMLN is computed by applying the *2-community* computation repeatedly as per the given expression that includes order. As the k-community is defined for a connected set of layers, the number of compositions can be more than k (corresponds to edges.)

We start with the 2-community of the first two layers in the expression – termed a t-community. For each new binary computation step, there are two cases for the 2-community computation under consideration: i) the U_i is from a layer G_i *already in the t-community* and the U_j is from a *new layer* G_j. This bipartite graph match is said to **extend** a t-community (t < k) to a *(t+1)-community*, or ii) **both** U_i (U_j) from layers G_i (G_J) are *already in the t-community.* This bipartite graph match is said to **update** a t-community (t < k).

Both layers are not in the t-community corresponds to the first 2-community computation. That one of them is not in the t-community after the first 2-community

community is not possible, by definition, as the expression is computed from left to right by adding one layer.

For both cases i) and ii) above, two outcomes are possible. Either a meta node from U_i: a) matches one or more meta nodes in U_j resulting in one or many **consistent match**, or b) does not match a meta node in U_j resulting in a **no match**. However, for case ii) above, a third possibility exists which can be characterized as c) matches a node in U_j that is not consistent with a previous match and is termed an **inconsistent match**. Since both communities have already been matched, a previous consistent match exists. If the current match is not the same, then it is an **inconsistent** match. Note that each of the relaxed pairings is a separate HeMLN k-community.

Structure preservation is accomplished by retaining, for each tuple of t-community, either a matching community id (or 0 if no match) and $x_{i,j}$ (or ϕ for empty set) representing inter-layer edges corresponding to the meta edge between the meta nodes (termed **expanded(meta edge)**). The *extend* and *update* carried out for each of the outcomes on the representation is listed in Table 11.1. Note that due to multiple pairing of nodes during any composition, the number of tuples (or t-communities) may increase. Copy & update is used to deal with multiple pairings. In general, each element of a k-community can be total or partial. A **partial k-community element** has ***at least one*** ϕ *or 0* as part of the tuple. Otherwise, it is a **total k-community element**. Any k-community that is **total** reflects a stronger coupling as it includes all inter-layer edges for those communities (as is the case of M-A-D-M in Figure 12.4 in Section 12.2.) A **partial** k-community element, on the other hand, for both acyclic and cyclic cases indicates strong coupling only among *the consistent match layers.*

(G_{left}, G_{right}) **outcome**	**Effect on tuple** $\boldsymbol{t}$
case (i) - one processed and one new layer	
a) consistent match	**Extend (Copy & Extend)** t with paired community id **and** $x_{i,j}$
b) no match	**Extend (Copy & Extend)** t with 0 and ϕ
case (ii) - both are processed layers	
a) consistent match	**Update (Copy & Update)** t **only with** x
b) no match	**Update (Copy & Update)** t **only with** ϕ
c) inconsistent match	**Update (Copy & Update)** t **only with** ϕ

Table 11.1: Cases and outcomes for MWxx (Extend and Update for MWPM/MWM; copy & extend/update or update for MWRM/MWMT) used in Algorithm 10

11.3.4 Characteristics of k-community

The above definition when applied to a specification generates *progressively strong coupling of communities between layers using specified MWxx pairing. Thus, our definition of a k-community is characterized by dense connectivity within the layer (community definition) and semantically strong coupling across layers using one of MWxx.* Hence, we believe, that this definition of k-community matches the original intuition of a community. By refining the pairing used and the edge weight based on participating community characteristics, it supports a family of community definitions that can be customized. Refer [12] for more details.

11.3.4.1 **Space of Analysis Alternatives**

Given a HeMLN with k layers and at least (k-1) inter-layer edges, the number of possible k-community (or analysis space) is quite large. For a HeMLN-graph, the number of potential k-community is a function of the number of unique connected subgraphs of different sizes and the number of possible orderings for each such connected subgraph. With the inclusion of m bipartite pairing choices and n weight metrics (see Section 11.5), it gets even larger. It is important to understand that each subgraph of a given size (equal to the number of edges in the connected subgraph) along with the ordering represents a *different* analysis of the data set and provides a different perspective thereby supporting a large space of analysis alternatives.

The composition function Θ defined above (one of MWM, MWPM, MWRM, MWMT) is commutative and not associative. Hence, for each k-community, the *order in which a k-community* is defined has a bearing on the result (semantics) obtained. In fact, the ordering is important as it differentiates one analysis from the other even for the same set of layers and inter-layer connections as elaborated in Chapter 12.

11.3.4.2 **Importance of Weights**

For traditional weighted bipartite matching, maximum weighted matching (MWM) or maximum weighted perfect matching (MWPM) algorithms (e.g., [97]) are used mainly because each node of a bipartite graph is a simple node. In contrast, each node of our bipartite graph is a meta node and the bipartite edge is also a meta edge. Each meta node, in our case, is a community representing a group of entities with its own characteristics (connectivity, degree, etc.) Each meta edge needs to, at the least, capture the number of edges in that meta edge (i.e., inter-layer edges.) The number of edges between the meta nodes is one of the proposed edge weights (ω_e) which corresponds to the traditional intuition behind a community.

Since edge weights play a significant role in the matching and is also used as a mechanism for determining the strength of the coupling of communities across layers, edge weights are used as a vehicle to include participating community characteristics. In addition to ω_e, it is possible to bring in participating community characteristics to capture additional aspects for coupling. We discuss a number of alternatives for weights (termed weight metrics ω) in Section 11.5, derived from real-world scenarios.

11.3.5 Evaluation of Proposed Community Definition

Ideally one would evaluate a new community definition by comparing the result with existing definitions. Since we do not have a community definition for a HeMLN, the **closest ground truth** is the type-independent aggregation of a heterogeneous multilayer network into a single network (as shown in Figure 11.1 a). Hence, we compare our results with this. Also, modularity is a widely accepted metric to measure the strength of division of a network into communities [135]. So, we use modularity for comparing our HeMLN communities (shown in Figure 11.1 b) with the type-independent communities obtained. We have computed modularity for different weight options as well as different matching algorithms to indicate how coupling strength changes with weights and matching algorithms. Below, we show the pairings for the default ω_e weight metric. For evaluation purpose, we use the HeMLNs, as described in Chapter 12 and whose layer details are shown in Table 12.2 and 12.1 of section 12.2. For IMDb (DBLP), we have used, respectively, the Actor and Director (Author and Paper) layers with their inter-layer edges.

Type-Independent	MWM	MWMT	MWPM	MWRM
0.777	0.643(83)	0.643(220)	**0.698(95)**	0.603(83)

Table 11.2: Modularity (# of Matches) for IMDb with A Θ D; ω_e (with All Communities)

For DBLP, the modularity value for our HeMLN community (Au Θ P; ω_e) obtained with each pairing algorithm is *equal to the modularity value for the type-independent community* (0.69). Good HeMLN communities are also obtained for IMDb (using, A Θ D; ω_e). However, the tuples/matched pairs (shown in parenthesis) vary slightly with the chosen algorithm due to which the structure of the communities change leading to slightly *different* modularity values. MWPM generates the best modularity as compared to the ground truth. It was observed that the Actor and Director communities that were paired by MWPM had dense intra-edge connectivity and many actor-director pairs participated in the interaction, thus resulting in high modularity. However, in case of the type-independent communities the actor and director node types get mixed up and smaller denser communities are produced leading to a higher modularity as compared to the HeMLN community, where the node types are kept separate.

11.4 HeMLN k-Community Algorithm

In this section, we first present a specification of a k-community and elaborate on a structure preserving representation for the result. Then we present a community detection algorithm using any of the bipartite algorithms along with the proposed bipartite matching algorithms.

11.4.1 k-community Representation

Linearization of a HeMLN structure is done using an order of specification which is also used for computation. Although a k-community need to be specified as an expression involving Ψ and Θ, as indicated earlier, we drop Ψ for clarity. For the layers shown in Figure 11.2, an example 3-community specification is $((G_1\ \Theta_{1,2}\ G_2)\ \Theta_{2,3}\ G_3)$. We can drop the parentheses as the precedence of Θ is assumed. However, we need the subscripts for Θ to disambiguate a k-community specification when a composition is done on the layers already used. A 3-community involving a cycle (when an expression corresponds to a HeMLN subgraph with a cycle) can be specified as $G_1\ \Theta_{1,2}\ G_2\ \Theta_{2,3}\ G_3\ \Theta_{3,1}\ G_1$.

A k-community is represented as a set of tuples. Each tuple represents a distinct element of a k-community and includes an ordering of k community ids as items (a path, if you will, connecting community ids from different layers) and at least (k-1) expanded(meta edge) (i.e., $x_{i,j}$) elements. This representation completely preserves the MLN structure along with semantics (labels) to reconstruct a HeMLN for any k-community. It is possible that there are multiple paths originating from the communities in the first layer of the expression due to relaxed pairings. That is, a community in a layer can participate in more than one k-community tuple. All these paths need not remain total as the k-community computations progress, due to no/inconsistent matches. In summary, each k-community is a tuple with 2 distinct components. The first component is a comma-separated sequence of *k community ids (as items)* from a layer. The second component is also a comma-separated sequence of *at least (k-1)* $x_{i,j}$ (with each x having a different pair of subscripts.) Communities for x are uniquely identifiable from the subscripts. It is exactly (k-1) if the k-community is for an acyclic connected graph and more depending upon the number of edges in cyclic subgraph.

11.4.2 MWRM and MWMT Algorthms

Algorithm 9 shows the computation of MWRM pairing. Line 1 gets the edge list from MWM algorithm of [97] and sorts the edges. The while loop starting in line 2 goes through this edge list from lowest weight and replaces it with a higher value edge if it has not been already chosen. This is done for all the edges in the MWM edge list. The additional complexity involves sorting MWM paired edges and only inspecting the number of edges incident on the nodes that are paired by MWM. The algorithm for MWMT is very similar except that it adds (instead of replacing in line 6) edges that are ties. There is no need to sort but only inspect the number of edges incident on the nodes that are paired by MWM. Both of these are substantially less than the number of inter-layer edges. Based on the characteristics of the matching algorithms,

Algorithm 9 MWRM and MWMT Algorithms

```
Require: -
    INPUT: Community bipartite graph (CBG)
    OUTPUT: edge list ($O_{el}$) for MWRM or MWMT
Initialize: $I_{el} \leftarrow$ MWM edges of CBG
    $M_{el} \leftarrow$ Meta edges of the input bipartite graph
    Set $O_{el}$ to $I_{el}$; Sort $I_{el}$ on edge weights
    $I_e \leftarrow$ edge from $I_{el}$ with the lowest weight
while $I_e$ not NULL do
    if MWRM then
        for each $M_e \in M_{el}$ that is incident on $I_e$ do
            if weight($M_e$) > weight($I_e$) and $M_e \notin O_{el}$ then
                replace $I_e$ in $O_{el}$ with $M_e$
            end if
        end for
        $I_e \leftarrow$ next lowest weight edge from $I_{el}$ or NULL
    end if
end while
```

we can assert the following for the total weights, $MWPM <= MWM <= MWRM$ and $MWM <= MWMT$. Moreover, MWM will generate the minimum number of pairs. MWM and MWRM will give same number of pairs.

11.4.3 k-community Detection Algorithm

Algorithm 10 accepts a linearized specification of a k-community expression and computes the result as described earlier. The input is an *ordering of layers*, *composi-*

tion functions indicating the community bipartite graph pairing algorithms to be used and the type of weight to be used. The output is a *set* whose *elements are tuples corresponding to distinct, single HeMLN k-community* as described earlier. The size (i.e., number of tuples) of this set is determined by the pairs obtained during computation. The layers for any 2-community bipartite graph composition are identifiable from the input specification.

Algorithm 10 HeMLN k-community Detection Algorithm

```
Require: -
  INPUT: HeMLN, ($G_{n1}$ $\Theta_{n1,n2}$ $G_{n2}$ ... $\Theta_{ni,nk}$ $G_{nk}$), MWM/MWPM/MWMT/MWMT), a weight metric ($\omega$).
  OUTPUT: Set of distinct HeMLN k-community tuples
Initialize: k=2, $U_i = \phi$, $U_j = \phi$, result$' = \emptyset$
result ← MWxx($G_{n1}$,$G_{n2}$, HeMLN, $\omega$)
left, right ← left and right subscripts of second $\Theta$
while left $\neq$ null && right $\neq$ null do
    $U_i$ ← subset of 1-community($G_{left}, result$)
    $U_j$ ← subset of 1-community($G_{right}, result$)
    MP ← MWxx($U_i$, $U_j$, HeMLN, $\omega$)
    //a set of comm pairs $< c^x_{left}, c^y_{right} >$
    for each tuple $t \in result$ do
        kflag = false
        if both $c^x_{left}$ and $c^y_{right}$ are part of $t$ and $\in$ MP [case ii (already processed layers): consistent match] then
            Update a copy of $t$ with ($x_{left,\ right}$) and append to result$'$
        else if $c^x_{left}$ is part of $t$ and $\in$ MP and $G_{right}$ layer has been processed [case ii (processed layer): no and inconsistent match] then
            Update a copy of $t$ with $\phi$ and append to result$'$
        else if $c^x_{left}$ is part of $t$ and for each $c^x_{left}$ $\in$ $MP$ [case i (new layer): consistent match] then
            copy and Extend $t$ with paired $c^y_{right} \in$ MP and $x_{left,\ right}$ and append to result$'$; kflag = true
        else if $c^x_{left}$ is part of $t$ and $\notin$ MP [case i (new layer): no match] then
            copy and Extend $t$ with 0 (community id) and $\phi$ and append to result$'$; kflag = true
        end if
    end for
    left, right = next left, right subscripts of $\Theta$ or null
    if kflag k = k + 1; result = result$'$; result$' = \emptyset$
end while
```

The bipartite graph for the first 2-community and for each application of Θ is constructed for the participating layers (either one is new or both are from the t-community for some t) and specified MWxx algorithm is applied. The result obtained is used to either extend or update (or copy & extend or update) the tuples of the t-community depending on the algorithm used. All cases are described in Table 11.1.

The algorithm iterates **(lines 2 to 18)** until there are no more compositions to be applied. The number of 2-community computations is equal to the number of Θ in the input (corresponds to the number of inter-layer connections in the expression.) For each layer, we assume that its 1-community has been computed.

Line 5 computes the k^{th} composition. **Lines 6 to 17** apply the results of the specified MWxx algorithm (**line 5**) to generate tuples of the k^{th} composition using the Table 11.1. Care is taken in the composition to make sure either the tuple is updated or extended by keeping a flag and checking it after **line 17**. The order of checking inside the for loop (**lines 6 to 17**) is important to generate the correct k-community tuples.

11.5 Customizing The Bipartite Graph

Algorithm 10 in Section 11.4 uses any of the specified bipartite graph match algorithms with a given weight metric. Without including the characteristics of meta nodes and edges for the match, we cannot argue that the pairing obtained represents analysis based on participating community characteristics. Hence, it is important to identify how qualitative community characteristics can be mapped quantitatively to a weight metric (that is, weight of the meta edge in a community bipartite graph) to influence the bipartite matching. Below, we propose three weight metrics and their intuition. Number of inter-community edges as weight (ω_e) can be used as default.

Number of Inter-Community Edges (ω_e): This metric uses actual number of inter-community edges of participating communities as weight for meta-edges. The intuition behind this metric is *maximum connectivity* (size of the community is to some extent factored into it) without including other community characteristics. This weight connotes *maximum interaction between two communities.* This weight also corresponds to the traditional community definition.

Hub Participation (ω_h) For many analysis, we are interested in knowing whether highly influential nodes within a community also interact across nodes in the other community. This can be translated to the *participation of influential nodes within and across each participating community* for analysis. This is modeled by using the notion of **hub participation** within a community and their interaction across layers. In this chapter, we have used degree centrality for this metric to connote higher influence. Ratio of participating hubs from each community and the edge fraction are multiplied to compute ω_h. Formally,

For every $(u_i^m, u_k^n) \in L'_{i,k}$, where u_i^m and u_k^n are the meta nodes denoting the communities, c_i^m and c_k^n in the community bipartite graph, respectively, the weight,

$$\omega_h(u_i^m,\, u_k^n) = \frac{|H_{i,k}^{m,n}|}{|H_i^m|} * \frac{|x_{i,k}|}{|v_i^{c^m}| * |v_k^{c^n}|} * \frac{|H_{k,i}^{n,m}|}{|H_k^n|},$$

where $x_{i,k} = \bigcup \{(a,b) : a \in v_i^{c^m}, b \in v_k^{c^n}$ *and* $(a,b) \in L_{i,j}\}$*;* H_i^m *and* H_k^n *are set of hubs in* c_i^m *and* c_k^n*, respectively;* $H_{i,k}^{m,n}$ *is the set of hubs from* c_i^m *that are connected to* c_k^n*;* $H_{k,i}^{n,m}$ *is the set of hubs from* c_k^n *that are connected to* c_i^m *.*

Density and Edge Fraction (ω_d) The intuition behind this metric is to bring participating community density which captures internal structure of a community. Clearly, *higher the densities and larger the edge fraction, the stronger is the interaction (or coupling) between two meta nodes (or communities.)* Since each of these three components (each being a fraction) increases the strength of the inter-layer coupling, they are multiplied to generate the weight of the meta edge. The domain of this weight will be $(0, 1]$. The weight computation formula is similar to the previous one and hence not shown.

11.6 k-community Computation Efficiency

1. **Cost of generating 1-community**: For each layer (or a subset of needed layers) this can be done in parallel bounding this **one-time cost** to the largest one.
2. **Cost of computing meta edge weights**: For the proposed analysis metrics, part of them, again, are **one-time costs** and are calculated independently on the 1-community results. Costs for ω_d and ω_h require a single pass of the communities using their node/edge details generated by the community detection algorithm.
3. **The recurring cost** (for each application of Θ): This includes the cost of generating the bipartite graph, computing the weight of each meta edge of the community bipartite graph for a given ω, and the MWxx algorithm cost. Only the edge fraction (or the maximum number of edges) and participating hubs need to be computed during each iteration. The cost of MWxx algorithm is *almost the same as* the cost of computing MWM. The bipartite graph is generated during the computation of weights for the meta edges. Luckily, in our community bipartite graph, the number of meta edges is **an order of magnitude less** than the number of edges between layers. Also, the number of meta nodes is bound by the number of pairings in the previous iteration.

11.7 Conclusions

In this chapter, we have provided a new structure and semantics preserving HeMLN community definition, two new bipartite algorithms (MWRM and MWMT) suited for community detection, and an efficient "decoupling-based" computation framework. We have also demonstrated the ease with which drill-down of the results can be accomplished because of structure and semantics preservation. *Also, with ω and MWxx as customizable parameters, our approach supports a wide range of analysis objectives and is extensible.*

CHAPTER 12

COMMUNITY ANALYSIS OF HETEROGENEOUS DBLP AND IMDb MLNs

In this chapter we demonstrate the application of the proposed structure and semantic-preserving k-community detection in order to fulfill a set of analysis objectives involving interactions among different types of entities. The data sets used for this purpose are from DBLP (Database Bibliography/Computer Science Publications) and IMDb (Internet Movie Database).

12.1 Mapping of Analysis Objectives

DBLP [95] Analysis Objectives

(A1) Conference-wise which are the *most cohesive* group(s) of authors who *publish frequently* (ties included)?

$\mathbf{P}\ \Theta_{P,Au}\ \mathbf{Au};\ \omega_d;\ \Theta = \mathbf{MWMT}$ (2-community)

(A2) For the *most popular unique* collaborators from each conference, which are the *unique most active* 3-year period(s)? $\mathbf{P}\ \Theta_{P,Au}\ \mathbf{Au}\ \Theta_{Au,Y}\ \mathbf{Y};\ \omega_e;\ \Theta = \mathbf{MWM}$

Based on the DBLP analysis requirements, three layers are modeled (See ([134]) for the HeMLN Layer Au connects any two authors (nodes) who have published at least three research papers together. Layer P connects research papers (nodes) that appear in the same conference. Layer Y connects two year nodes if they belong to same pre-defined period. The inter-layer edges depict *wrote-paper ($L_{Au,P}$), active-in-year ($L_{Au,Y}$) and published-in-year ($L_{P,Y}$).* For this analysis, we have chosen all papers that were published from 2001-2018 in top conferences. Six 3-year periods have been chosen: [2001-2003], [2004-2006], ..., [2016-2018].

IMDb [94] Analysis Objectives

(A3) Find the actor and director similar-genre based group pairs such that *overall actor-director collaborations are maximized*? $\mathbf{A}\ \Theta_{A,D}\ \mathbf{D};\ \omega_e;\ \Theta = \mathbf{MWRM}$

(A4) Based on genres, list the *maximum number* of *unique* actor and director groups whose *majority of the most versatile members interact*? $\mathbf{A}\ \Theta_{A,D}\ \mathbf{D};\ \omega_h;\ \Theta = \mathbf{MWPM}$

(A5) For the *most popular unique* actor groups (including ties), from each movie rating class, find the *unique* director groups with *maximum interaction* and who also make movies with similar ratings.

Cyclic: $\mathbf{M}\ \Theta_{M,A}\ \mathbf{A}\ \Theta_{A,D}\ \mathbf{D}\ \Theta_{D,M}\ \mathbf{M};\ \omega_e;\ \Theta = \mathbf{MWMT}$

For the IMDb analysis requirements, three layers for the IMDb data set are formed. Layer A and Layer D connect actors and directors who act-in or direct *similar genres frequently*, respectively. Layer M connects movies within the same rating range. The inter-layer edges depict *acts-in-a-movie ($L_{A,M}$), directs-movie ($L_{D,M}$) and directs-actor ($L_{A,D}$)*. There are multiple ways of quantifying the similarity of actors and directors based on movie genres they have worked in. A vector was generated with the number of movies for each genre he/she has acted-in/directed. In order to consider the similarity with respect to *frequency of genres*, two actors/directors are connected if the Pearsons' Correlation between their corresponding genre vectors is at least 0.9 (Other values can also be used based on similarity strength.)

Choice of weight metric: For the objectives specified in this chapter, *maximum interaction* and *most popular* in (A2), (A3) and (A5), are interpreted as the number of edges between the participating communities. In contrast, interaction with cohesive groups as in (A1), is interpreted to include community density as well. Versatility is mapped to participation of hub nodes in each group as in (A4).

Choice of pairing algorithm: Each pairing algorithm maximizes the overall weight based on a constraint. For (A2), MWM is chosen due to the the unique pairing constraint. For (A1) and (A5), the unique constraint is relaxed to only inlcude ties, thus MWMT is selected. For (A4), the uniqueness criterion is combined with maximizing the number of pairs, thus we chose MWPM. In (A3), the uniqueness restriction is absent making MWRM the choice.

Identifying the k-community: (A1), (A3) and (A4) compute a 2-community. (A2) requires a 3-community (for 3 layers) with an acyclic specification (using only 2 edges). (A5) uses the layer M twice for a 3-community and is also cyclic. Note that the analysis objectives have been chosen carefully to cover the weights and pairing algorithms discussed in the chapter. The limitation on the number of analysis objectives is purely due to space constraints.

12.2 Experimental Set up

The choice of data sets and sizes used in this chapter are primarily for demonstrating the versatility of analysis using the k-community detection and its efficiency as well as drill-down capability based on structure and semantics preservation. We are not trying to demonstrate scalability in this thesis. Also, instead of presenting all communities, we have chosen to show a few important drill-down results to showcase the structure and semantics preservation of our approach.

For DBLP HeMLN, research papers published from 2001-2018 in VLDB, SIGMOD, KDD, ICDM, DaWaK, and DASFAA were chosen. For IMDb HeMLN, we extracted, for the top 500 actors, the movies they have worked in (7500+ movies with 4500+ directors). The actor set was repopulated with the co-actors from these movies, giving a total of 9000+ actors. Widely used Louvain method [67] is used to detect 1-communities. The k-community detection algorithm 10 was implemented in Python version 3.7.3 and was executed on a quad-core 8^{th} generation Intel i7 processor Windows 10 machine with 8 GB RAM.

12.3 Analysis Results, Drill-down and Visualization

Individual Layer Statistics: Table 12.1 shows the layer-wise statistics for IMDb HeMLN. 63 Actor (A) and 61 Director (D) communities based on similar genres are generated. Out of the 10 ranges (communities) in the movie (M) layer, most of the movies were rated in the range [6-7), while least popular rating was [1-2). No movie had a rating in the range [0-1).

IMDb	**Actor**	**Director**	**Movie**
#Nodes	9485	4510	7951
#Edges	996,527	250,845	8,777,618
#Communities (Size >1/all)	63/190	61/190	9/9
Avg. Community Size	148.5	73	883.4

Table 12.1: IMDb HeMLN Statistics

DBLP	**Author**	**Paper**	**Year**
#Nodes	16,918	10,326	18
#Edges	2,483	12,044,080	18
#Communities (Size >1/all)	591/15528	6/6	6/6
Avg. Community Size	3.3	1721	3

Table 12.2: DBLP HeMLN Statistics

Similarly, DBLP HeMLN statistics are shown in Table 12.2. 591 Author (Au) communities are generated based on co-authorship. 6 Paper (P) communities are formed by grouping papers published in same conference. KDD (2942) and DASFAA

(583) have highest and least published papers, respectively. Out of 6 ranges of years (Y) selected, the maximum and minimum papers were published in 2016-2018 (1978) and 2001-2003 (1421), respectively.

Expression	**MWM**	**MWMT**	**MWPM**	**MWRM**
P-Au,ω_d *#Comm::P(6),* *Au(591)*	*total*:6, *partial*:0, $\Sigma\omega$:0.04988	*total*:9, *partial*:0, $\Sigma\omega$:**0.0778**	*total*:6, *partial*:0, $\Sigma\omega$:*0.04988*	*total*:6, *partial*:0, $\Sigma\omega$:0.05003
P-Au-Y,ω_e *#Comm::P(6),* *Au(591), Y(6)*	*total*:6, *partial*:0, Tot.$\Sigma\omega$:548	*total*:6, *partial*:0, Tot.$\Sigma\omega$:548	*total*:6, *partial*:0, Tot.$\Sigma\omega$:*548*	*total*:6, *partial*:3, Tot.$\Sigma\omega$:**986**
A-D,ω_e *#Comm::A(63),* *D(61)*	*total*:5, *partial*:0, $\Sigma\omega$:9902	*total*:69, *partial*:0, $\Sigma\omega$:9970	*total*:57, *partial*:0, $\Sigma\omega$:*5144*	*total*:50, *partial*:0, $\Sigma\omega$:**11640**
A-D,ω_d *#Comm::A(63),* *D(61)*	*total*:53, *partial*:0, $\Sigma\omega$:3.32378	*total*:220, *partial*:0, $\Sigma\omega$:3.36045	*total*:55, *partial*:0, $\Sigma\omega$:*3.3235*	*total*:53, *partial*:0, $\Sigma\omega$:**3.33509**
A-D,ω_h *#Comm::A(63),* *D(61)*	*total*:27, *partial*:0, $\Sigma\omega$:0.62142	*total*:41, *partial*:0, $\Sigma\omega$:0.62263	*total*:29, *partial*:0, $\Sigma\omega$:*0.62024*	*total*:27, *partial*:0, $\Sigma\omega$:0**0.62229**
M-A-D-M,ω_e *#Comm::A(63),* *D(61), M(9)*	*total*:2, *partial*:7, $\Sigma\omega$:6979	*total*:3, *partial*:12, $\Sigma\omega$:6984	*total*:0, *partial*:9, $\Sigma\omega$:*6979*	*total*:2, *partial*:11, $\Sigma\omega$:**11557**

Table 12.3: Effect of Pairing Algorithms on same specification (for Non-Singleton Communities)

Effect of Pairing Algorithm: For an expression with a specified layer order for evaluation and weight metric, the results will vary based on choice of algorithm for pairing communities from the community bipartite graph. This is illustrated in Table 12.3 where we list the number of HeMLN communities (total+partial) and total meta edge weights for the first round of matching ($\sum\omega$) obtained with 4 different pairing algorithms for the same specification. It can be observed that MWM generates the least number of pairs with maximum total weight. On the other hand, when the uniqueness condition is relaxed, the overall sum of weights is improved by MWMT and MWRM.

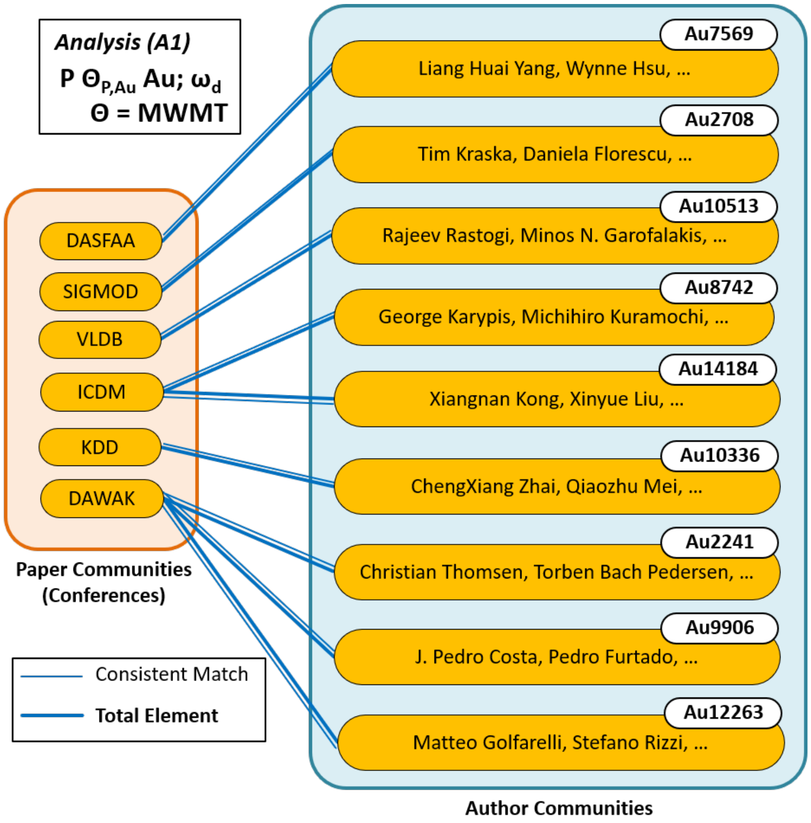

Analysis (A1)
P Θ_{P,Au} Au; ω_d
Θ = MWMT
DASFAA
SIGMOD
VLDB
ICDM
KDD
DAWAK
Paper Communities (Conferences)
Consistent Match
Total Element
Au7569
Liang Huai Yang, Wynne Hsu, …
Au2708
Tim Kraska, Daniela Florescu, …
Au10513
Rajeev Rastogi, Minos N. Garofalakis, …
Au8742
George Karypis, Michihiro Kuramochi, …
Au14184
Xiangnan Kong, Xinyue Liu, …
Au10336
ChengXiang Zhai, Qiaozhu Mei, …
Au2241
Christian Thomsen, Torben Bach Pedersen, …
Au9906
J. Pedro Costa, Pedro Furtado, …
Au12263
Matteo Golfarelli, Stefano Rizzi, …
Author Communities

Figure 12.1: (A1) Result: **9 Total Elements**[*]

(A1) Analysis: On applying MWMT on the CBG created with all Paper and Author communities, we obtained 9 total elements that correspond to the *most cohesive co-authors who also publish frequently in each conference* (shown in Figure 12.1 with list of few prominent authors.) ICDM and DaWaK have **multiple author communities** that are **equally important**. Prominent researchers like Tim

[*] Louvain numbers all communities from 1 and we only consider communities having *at least two members* for this chapter. The numbering used in the chapter have layer name followed by the Louvain-generated community ID (e.g. A91, Au8742).

Kraska and Daniela Florescu; Rajeev Rastogi and Minos N. Garofalakis, and; George Karypis and Michihiro Kuramochi are members of the frequently publishing co-author group (in the last 18 years) for SIGMOD, VLDB and ICDM, respectively. Quality of these obtained frequently publishing collaborators can be validated from the facts that a) *Tim Kraska* has been a recipient of **Best of SIGMOD Award (2008, 2016)**, b) *Rajeev Rastogi*'s published papers in VLDB (in past 18 years) have received over 900 citations c) *George Karypis* has been a recipient of **IEEE ICDM 10-Year Highest-Impact Paper Award (2010) and IEEE ICDM Research Contributions Award (2017)** [†].

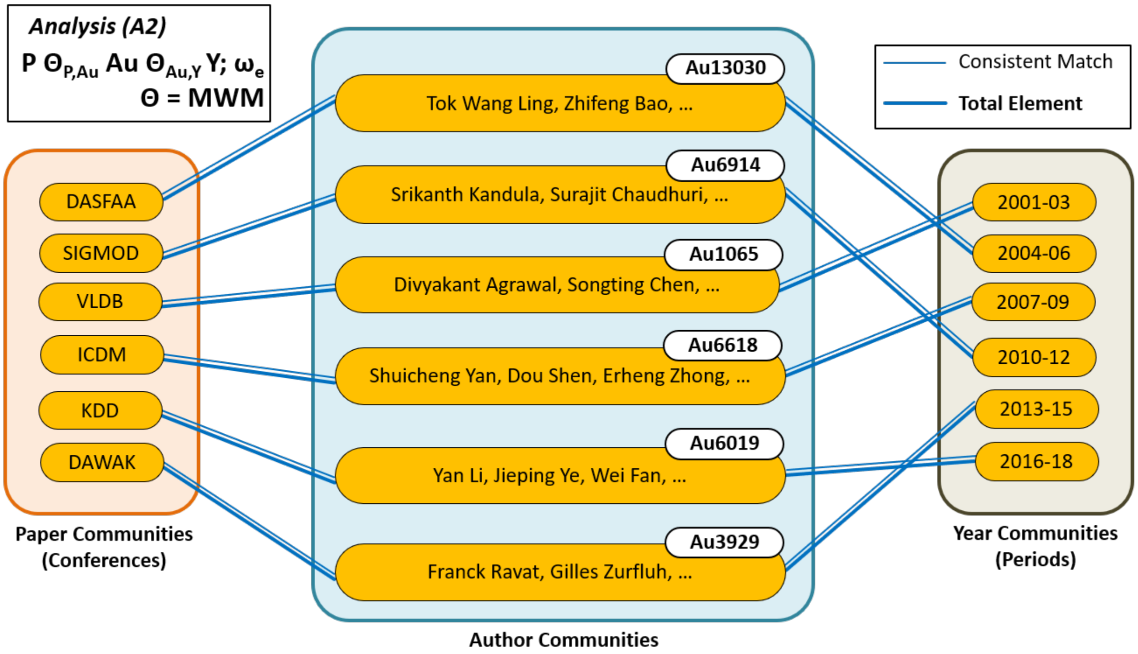

Analysis (A2)
P $\Theta_{P,Au}$ Au $\Theta_{Au,Y}$ Y; ω_e
Θ = MWM
Consistent Match
Total Element
DASFAA
SIGMOD
VLDB
ICDM
KDD
DAWAK
Paper Communities (Conferences)
Au13030
Tok Wang Ling, Zhifeng Bao, ...
Au6914
Srikanth Kandula, Surajit Chaudhuri, ...
Au1065
Divyakant Agrawal, Songting Chen, ...
Au6618
Shuicheng Yan, Dou Shen, Erheng Zhong, ...
Au6019
Yan Li, Jieping Ye, Wei Fan, ...
Au3929
Franck Ravat, Gilles Zurfluh, ...
Author Communities
2001-03
2004-06
2007-09
2010-12
2013-15
2016-18
Year Communities (Periods)

Figure 12.2: (A2) Result: **6 Total Elements**

[†] Intra-layer edge weights are not considered in this analysis. Hence, for an author (e.g., Jiawei Han) who has authored large *number of papers*, his co-authors are distributed among different co-author communities due to lack of weight information and hence does not come out.

(A2) Analysis: For the required *acyclic 3-community* results, the *most popular unique author groups* for *each conference* are obtained by MWM (first composition). The matched 6 author communities are carried forward to find the *disjoint year periods* in which they were *most active* (second composition). 6 total elements are obtained (path shown by **bold blue lines** in Figure 12.2.) Few prominent names have been shown in the Figure 12.2 based on citation count (from Google Scholar profiles.) For example, for *SIGMOD, VLDB and ICDM* the most popular researchers include **Srikanth Kandula (15188 citations), Divyakant Agrawal (23727 citations) and Shuicheng Yan (52294 citations)**, respectively who were active in different periods in the past 18 years.

(A3) Results: 83 A-D (Actor-Director) similar genre-based *overlapping* community pairs were obtained by MWRM, that maximised the *overall number of actor-director interactions*. Due to the absence of uniqueness criterion, some actor communities were paired with multiple director communities, and vice-versa.

(A4) Results: MWPM *maximizes the number of unique* A-D (Actor-Director) similar genre-based community pairs (29), where *majority of most versatile members interact*. Intuitively, a group of directors that prominently makes movies in some genre (say, Drama, Comedy, Romance, ...) must pair up with the group of actors who primarily act in similar kind of movies. This can be validated from the few sample similar genre-based pairings shown in Figure 12.3 (drill down) , such as a) **Comedy** - Directors like **Bobby Farrelly, Todd Phillips, John Landis** etc. (from D35) pair up with actors like **Jim Carrey, Zach Galifianakis and Eddie Murphy** (from A1), b) **Action/Drama** - Directors like **Clint Eastwood, Ridley Scott and Steven Spielberg** (from D102) pair up with Actors like **Brad Pitt, Tom Cruise and Will Smith** (from A144) and c) **Romance** - Directors **Woody Allen, Tim**

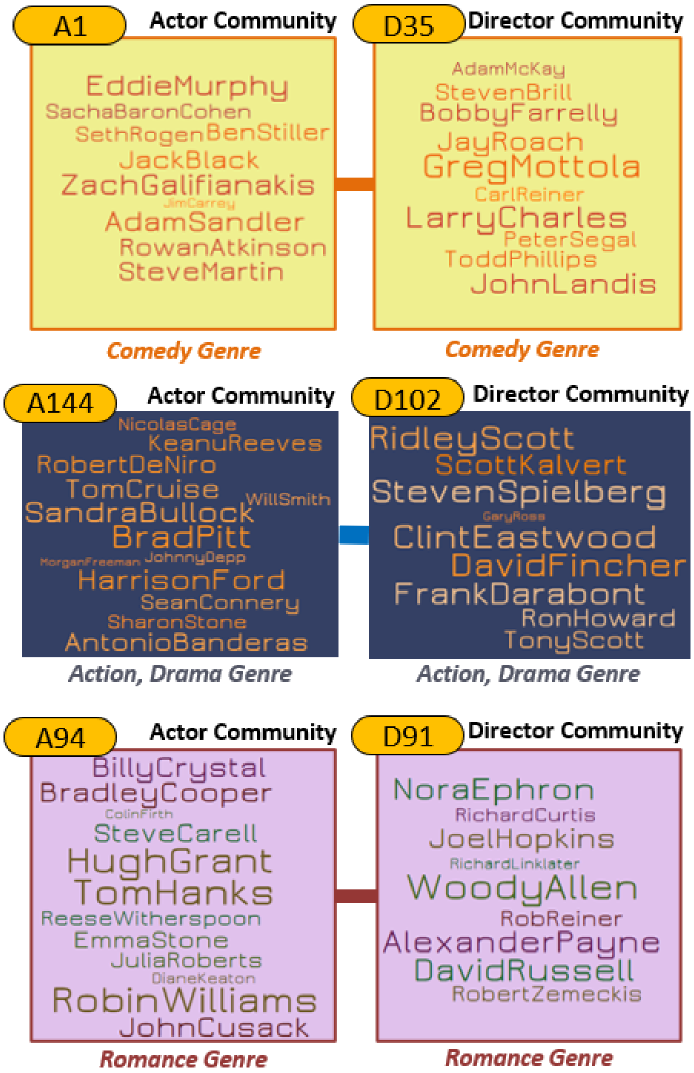
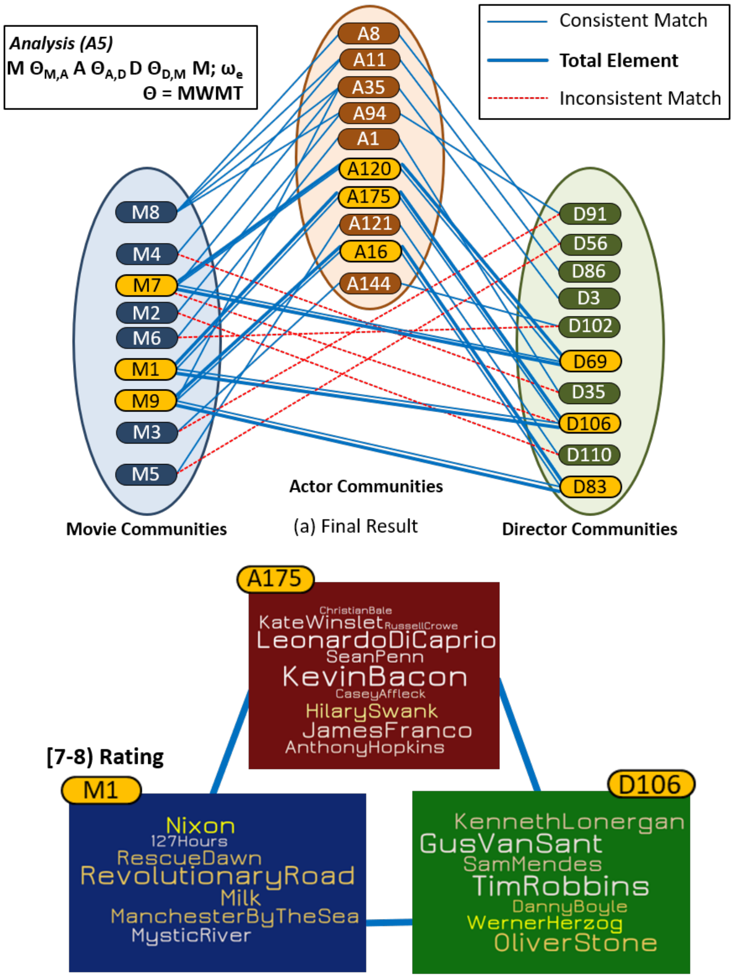


Figure 12.3: Sample (A4) Result for *Comedy, Action/Drama, Romance* Genres

Burton etc. (from D91) pair up with the actors like **Diane Keaton, Emma Stone and Hugh Grant** (from A94).

(A5) Results: Here, the *most popular unique actor groups for each movie rating class are further coupled with directors.* These *unique director groups are coupled again with movies to check whether the director groups also have similar ratings.* In every round for every pairing the ties are also included (MWMT). Results of each successive pairing (there are 3) are shown in Figure 12.4 (a) using the same color notation. Coupling of movie and actor communities (first composition) results in



(a) Final Result

(b) Sample *Movies, Actors and Directors* from a Total Element

Figure 12.4: (A5) Result: **3 Total, 12 Partial Elements**

14 consistent matches. In the second composition with the director layer, using all director communities and the matched 10 actor communities, we got 10 consistent matches. The final composition to complete the cycle uses 10 director communities and 9 movie communities as left and right sets of community bipartite graph, respectively.

Only 3 consistent matches are obtained to generate the 3 total elements for the cyclic 3-community (bold blue triangle.) The total element M1-A175-D106-M1 (sample members shown in Figure 12.4 (b)) *groups together popular highly* *rated* (**Average Rating of [7-8)**) *Drama* genre-based actors like **Leonardo DiCaprio, Sean Penn, Kate Winslet, Hilary Swank, Kevin Bacon, Anthony Hopkins, Russell Crowe, Christian Bale, James Franco and Casey Affleck** (from A175) with *popular drama directors* like **Danny Boyle, Sam Mendes, Werner Herzog, Gus Van Sant, Tim Robbins, Oliver Stone, Kenneth Lonergan**. This actor-director group is involved in few of the iconic award wining masterpieces like **Revolutionary Road, 127 Hours, Rescue Dawn, Milk, Mystic River, Nixon and Manchester By The Sea**.

Most importantly, this genre-based group is also able to flesh out *potential actor-actor or actor-director collaborations.* For instance, on drill-down it is observed Leonardo DiCaprio has not only worked with Sam Mendes (inter-layer edge present) but is also similar to Hilary Swank in terms of the type of movies worked in (intra-layer edge present). However, DiCaprio and Swank have never worked together. Thus, one potential collaboration that has expertise in similar genres and has highly rated members can be **DiCaprio-Swank-Mendes**. On similar grounds, another potential collaboration is **Bacon-Hopkins-Stone**, who have not worked together yet.

It is interesting to see 6 inconsistent matches (red broken lines) between the communities which clearly indicate that all couplings are not satisfied by these pairs. This results in 12 partial elements which represent the similar genre-based actor and director groups but with *different most popular movie rating classes.*

The inconsistent matches also highlight the importance of mapping an analysis objective to a k-community specification for computation. If a different order had been chosen (viz. director and actor layer as the base case), the result could have included the inconsistent matches.

12.4 Efficiency Results of Proposed Approach

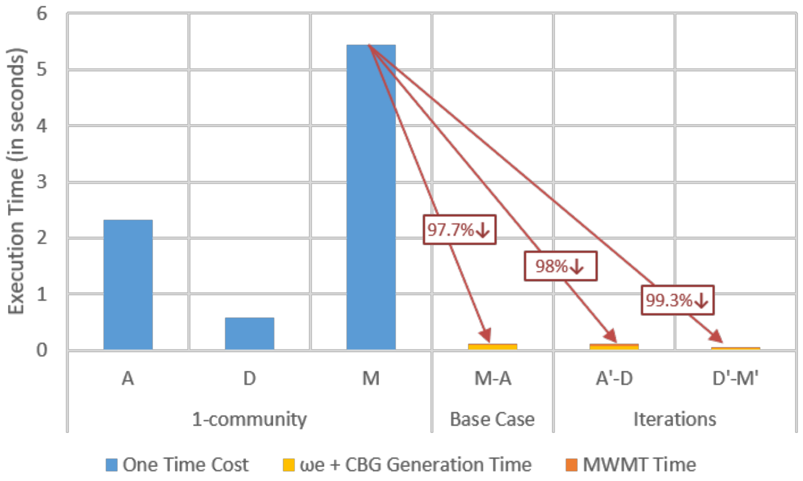


Figure 12.5: Performance Results for a 3-community using (A5)

The goal of the decoupling approach was to preserve the structure as well as improve the efficiency of k-community detection using the divide and conquer approach. We illustrate that with the largest k-community we have computed which uses 3 compositions. Figure 12.5 shows the execution time for the one-time and iterative costs discussed earlier for (A5). The difference in one-time 1-community cost for the 3 layers follow their density shown in Table 12.1. We can also see how the iterative cost is insignificant as compared to the one time cost (by an order of magnitude.) Cost of each iteration includes creating the bipartite graph, computing ω_e for meta edges, and MWxx (in this case MWMT) cost. **The cost of all iterations together (0.27 sec) is more than *an order of magnitude less than the largest one-time cost* (5.43 sec for Movie layer.)** We have used this case as this subsumes all other cases. The **additional incremental cost for computing a k-community is extremely small validating the efficiency of decoupled approach**.

12.5 Conclusions

In this chapter, we used the proposed k-community detection approach for demonstrating its analysis versatility over the IMDb and DBLP data sets.

CHAPTER 13

CONCLUSION

In this thesis, we discussed the rationale behind choosing multilayer network model for modeling complex data sets into its Homogeneous, Heterogeneous or Hybrids alternatives. Moreover, we provide the algorithmic steps to convert an EER model to MLN model to make the effective incorporation of the analysis requirements error-free.

A novel network decoupling based framework has been proposed for efficiently analyzing MLNs. As a part of this framework, we have proposed Boolean composition based Community and centrality detection in Homogeneous MLNs. Extensive experiments have been performed on synthetically generated HoMLN layers in order to infer the effect on graph characteristics on the accuracy of composition algorithms. In case of HeMLNs, we have provided a structure-and-semantic preserving definition, which was lacking till date. Moreover, we have used the decoupling approach in order to propose a family of algorithms for its computation using the concept of weighted maximum bipartite graph pairings.

Elaborate experiments have been performed on synthetically generated data sets (using R-Mat) and real world data sets like Facebook, UK Accidents, IMDb, DBLP and US Airlines for showcasing the modeling clarity, analysis flexibility and computational efficiency provided by MLNs. Drill-down analysis and visualizations have been done on the final results for inferring interesting hidden knowledge, which have been verified with independently available sources.

Overall, we have successfully proposed a framework for effective and efficient analysis of complex multi-entity, feature and relationship data sets.

In the future, the decoupling-based framework can be extended to include composition algorithms for graph querying, detecting centrality for HeMLNs and so on. Currently, only unweighted and undirected intra- and inter-layer edges have been considered. Thus, the composition algorithms can be extended to include other types of edges as well. Apart from this, parallelization techniques can be applied to MLN analysis.

REFERENCES

[1] L. Danon, A. P. Ford, T. House, C. P. Jewell, M. J. Keeling, G. O. Roberts, J. V. Ross, and M. C. Vernon, "Networks and the epidemiology of infectious disease," *Interdisciplinary perspectives on infectious diseases*, vol. 2011, 2011.

[2] P. R. Berthon, L. F. Pitt, K. Plangger, and D. Shapiro, "Marketing meets web 2.0, social media, and creative consumers: Implications for international marketing strategy," *Business horizons*, vol. 55, no. 3, pp. 261–271, 2012.

[3] S. P. Borgatti, A. Mehra, D. J. Brass, and G. Labianca, "Network analysis in the social sciences," *science*, vol. 323, no. 5916, pp. 892–895, 2009.

[4] N. Adams, N. Heard, N. Adams, and N. Heard, *Data Analysis for Network Cyber-Security.* River Edge, NJ, USA: World Scientific Publishing Co., Inc., 2014.

[5] A. L. Hopkins, "Network pharmacology: the next paradigm in drug discovery," *Nature chemical biology*, vol. 4, no. 11, p. 682, 2008.

[6] A. Santra and S. Bhowmick, "Holistic analysis of multi-source, multi-feature data: Modeling and computation challenges," in *Big Data Analytics - Fifth International Conference, BDA 2017*, 2017.

[7] A. Santra, S. Bhowmick, and S. Chakravarthy, "Efficient community re-creation in multilayer networks using boolean operations," in *International Conference on Computational Science, ICCS 2017, 12-14 June 2017, Zurich, Switzerland*, 2017, pp. 58–67. [Online]. Available: https://doi.org/10.1016/j.procs.2017.05.246

[8] ——, "Hubify: Efficient estimation of central entities across multiplex layer compositions," in *2017 IEEE International Conference on Data Mining Workshops, ICDM Workshops 2017, New Orleans, LA, USA, November 18-21, 2017*, 2017, pp. 142–149. [Online]. Available: https://doi.org/10.1109/ICDMW.2017.24

[9] S. Chakravarthy, A. Santra, and K. S. Komar, "Humble data management to big data analytics/science: A retrospective stroll," in *Big Data Analytics - 6th International Conference, BDA 2018, Warangal, India, December 18-21, 2018, Proceedings*, 2018, pp. 33–54. [Online]. Available: https://doi.org/10.1007/978-3-030-04780-1_3

[10] X.-S. Vu, A. Santra, S. Chakravarthy, and L. Jiang, "Generic multilayer network data analysis with the fusion of content and structure," in *Computational Linguistics and Intelligent Text Processing - 20th International Conference, CICLing 2019, La Rochelle, France, April 7-13, 2019*, 2019.

[11] S. Chakravarthy, A. Santra, and K. S. Komar, "Why multilayer networks instead of simple graphs? modeling effectiveness and analysis flexibility and efficiency!" in *Big Data Analytics - 7th International Conference, BDA 2019*, 2019, pp. 227–244. [Online]. Available: https://doi.org/10.1007/978-3-030-37188-3_14

[12] A. Santra, K. S. Komar, S. Bhowmick, and S. Chakravarthy, "A new community definition for multilayer networks and a novel approach for its efficient computation," *arXiv preprint arXiv:2004.09625*, 2020.

[13] A. Santra, K. Komar, S. Bhowmick, and S. Chakravarthy, "Making a case for mlns for data-driven analysis: Modeling, efficiency, and versatility," *CoRR*, vol. abs/1909.09908, 2019. [Online]. Available: http://arxiv.org/abs/1909.09908

[14] K. Komar, A. Santra, S. Bhowmick, and S. Chakravarthy, “Eer→mln: Eer approach for modeling, mapping, and analyzing complex data using multilayer networks (mlns),” in *Conceptual Modeling - 39th International Conference, ER 2020, Vienna, Austria, November 3-6, 2020, Proceedings*, accepted.

[15] A. Santra, K. Komar, S. Bhowmick, and S. Chakravarthy, “A community definition for multilayer networks and a novel approach for its efficient computation,” *IEEE Trans. Knowl. Data Eng.*, submitted.

[16] A. Santra, S. Bhowmick, and S. Chakravarthy, “Efficient community detection in boolean composed multilayer networks,” *ACM Trans. Knowl. Discov. Data*, submitted.

[17] Y. Mass and Y. Sagiv, “Knowledge management for keyword search over data graphs,” in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China*, 2014, pp. 2051–2053. [Online]. Available: http://doi.acm.org/10.1145/2661829.2661846

[18] S. Bu, X. Hong, Z. Peng, and Q. Li, “Integrating meta-path selection with user-preference for top-k relevant search in heterogeneous information networks,” in *Computer Supported Cooperative Work in Design (CSCWD), Proceedings of the 2014 IEEE 18th International Conference on.* IEEE, 2014, pp. 301–306.

[19] S. Das and S. Chakravarthy, “Duplicate reduction in graph mining: Approaches, analysis, and evaluation,” vol. 30, no. 8, 2018, pp. 1454–1466.

[20] M. Rahman and M. Al Hasan, “Approximate triangle counting algorithms on multi-cores,” in *2013 IEEE International Conference on Big Data.* IEEE, 2013, pp. 127–133.

[21] M. Kuramochi and G. Karypis, “Finding frequent patterns in a large sparse graph*,” *Data Min. Knowl. Discov.*, vol. 11, no. 3, pp. 243–271, 2005.

[22] L. B. Holder, D. J. Cook, and S. Djoko, “Substucture Discovery in the SUBDUE System,” in *Knowledge Discovery and Data Mining*, 1994, pp. 169–180.

[23] N. Jayaram, A. Khan, C. Li, X. Yan, and R. Elmasri, “Querying knowledge graphs by example entity tuples,” in *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 10, 2015, pp. 2797–2811.

[24] S. Das, A. Santra, J. Bodra, and S. Chakravarthy, “Query processing on large graphs: Approaches to scalability and response time trade offs,” *Data & Knowledge Engineering*, p. 101736, 2019.

[25] S. Das, A. Goyal, and S. Chakravarthy, “Plan before you execute: A cost-based query optimizer for attributed graph databases,” in *DaWaK 2016, Porto, Portugal, September 6-8, 2016*, 2016, pp. 314–328.

[26] Y. Hao, H. Cao, Y. Qi, C. Hu, S. Brahma, and J. Han, “Efficient keyword search on graphs using mapreduce,” in *2015 IEEE International Conference on Big Data, Big Data 2015, Santa Clara, CA, USA, October 29 - November 1, 2015*, 2015, pp. 2871–2873.

[27] D. Shasha, J. T.-L. Wang, and R. Giugno, “Algorithmics and applications of tree and graph searching,” in *PODS*, 2002, pp. 39–52.

[28] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter, “Multilayer networks,” *CoRR*, vol. abs/1309.7233, 2013.

[29] B. Boden, S. Günnemann, H. Hoffmann, and T. Seidl, “Mining coherent subgraphs in multi-layer graphs with edge labels,” in *Proc. of the 18th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (SIGKDD 2012)*, 2012, pp. 1258–1266.

[30] M. K.-P. Ng, X. Li, and Y. Ye, “Multirank: co-ranking for objects and relations in multi-relational data,” in *Proceedings of the 17th ACM SIGKDD*

international conference on Knowledge discovery and data mining, 2011, pp. 1217–1225.

[31] X. Dong, P. Frossard, P. Vandergheynst, and N. Nefedov, “Clustering with multi-layer graphs: A spectral perspective,” *CoRR*, vol. abs/1106.2233, 2011. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1106.html#abs-1106-2233

[32] M. Magnani and L. Rossi, “Formation of multiple networks,” in *International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction.* Springer, 2013, pp. 257–264.

[33] M. De Domenico, A. Lancichinetti, A. Arenas, and M. Rosvall, “Identifying modular flows on multilayer networks reveals highly overlapping organization in interconnected systems,” *Physical Review X*, vol. 5, no. 1, p. 011027, 2015.

[34] Z. Wang, L. Wang, A. Szolnoki, and M. Perc, “Evolutionary games on multilayer networks: a colloquium,” *The European physical journal B*, vol. 88, no. 5, p. 124, 2015.

[35] G. A. Pagani and M. Aiello, “The power grid as a complex network: a survey,” *Physica A: Statistical Mechanics and its Applications*, vol. 392, no. 11, pp. 2688–2700, 2013.

[36] A. Calvó-Armengol and Y. Zenou, “Social networks and crime decisions: The role of social structure in facilitating delinquent behavior,” *International Economic Review*, vol. 45, no. 3, pp. 939–958, 2004.

[37] S. Boccaletti, G. Bianconi, R. Criado, C. del Genio, J. Gómez-Gardeñes, M. Romance, I. Sendiña-Nadal, Z. Wang, and M. Zanin, “The structure and dynamics of multilayer networks,” *Physics Reports*, 2014.

[38] J. Kim and J. Lee, “Community detection in multi-layer graphs: A survey,” *SIGMOD Record*, vol. 44, no. 3, pp. 37–48, 2015.

[39] M. De Domenico, M. A. Porter, and A. Arenas, “Muxviz: a tool for multilayer analysis and visualization of networks,” *Journal of Complex Networks*, p. cnu038, 2014.

[40] “Muxviz: Framework for the multilayer analysis and visualization of networks,” http://muxviz.net/.

[41] F. Battiston, V. Nicosia, and V. Latora, “Structural measures for multiplex networks,” *Physical Review E*, vol. 89, no. 3, p. 032804, 2014.

[42] “Mammult: Collection of programs (c and python) for the analysis and modeling of multilayer networks,” http://complex.ffn.ub.es/~lasagne/news_full.php?q=MAMMULT.

[43] M. Kivelä, “Pymnet: Free library for analysing multilayer networks,” http://people.maths.ox.ac.uk/kivela/mln_library/.

[44] M. D. Domenico, V. Nicosia, A. Arenas, and V. Latora, “Layer aggregation and reducibility of multilayer interconnected networks,” *CoRR*, vol. abs/1405.0425, 2014.

[45] P. P.-S. Chen, “The entity-relationship model—toward a unified view of data,” *ACM transactions on database systems (TODS)*, vol. 1, no. 1, pp. 9–36, 1976.

[46] S. Das, A. Santra, J. Bodra, and S. Chakravarthy, “Query processing on large graphs: Approaches to scalability and response time trade offs,” *Data Knowl. Eng.*, vol. 126, p. 101736, 2020. [Online]. Available: https://doi.org/10.1016/j.datak.2019.101736

[47] S. Chakravarthy, R. Beera, and R. Balachandran, “DB-Subdue: Database Approach to Graph Mining,” in *PAKDD*, 2004, pp. 341–350.

[48] R. Angles and C. Gutierrez, “Survey of graph database models,” *ACM Computing Surveys (CSUR)*, vol. 40, no. 1, pp. 1–39, 2008.

[49] N. Roy-Hubara, L. Rokach, B. Shapira, and P. Shoval, "Modeling graph database schema," *IT Professional*, vol. 19, no. 6, pp. 34–43, 2017.

[50] J. Pokornỳ, "Conceptual and database modelling of graph databases," in *Proceedings of the 20th International Database Engineering & Applications Symposium*, 2016.

[51] R. Angles, "The property graph database model." in *AMW*, 2018.

[52] R. De Virgilio, A. Maccioni, and R. Torlone, "Model-driven design of graph databases," in *International Conference on Conceptual Modeling.* Springer, 2014, pp. 172–185.

[53] M. Graves, E. R. Bergeman, and C. B. Lawrence, "Graph database systems," *IEEE Engineering in Medicine and Biology Magazine*, vol. 14, no. 6, pp. 737–745, 1995.

[54] A. Clauset, M. E. J. Newman, , and C. Moore, "Finding community structure in very large networks," *Physical Review E*, pp. 1– 6, 2004. [Online]. Available: www.ece.unm.edu/ifis/papers/community-moore.pdf

[55] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, "Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters," 2008.

[56] U. Brandes, M. Gaertler, and D. Wagner, "Experiments on graph clustering algorithms," in *In 11th Europ. Symp. Algorithms.* Springer-Verlag, 2003, pp. 568–579.

[57] J. Xie, S. Kelley, and B. K. Szymanski, "Overlapping community detection in networks: The state-of-the-art and comparative study," *ACM Comput. Surv.*, vol. 45, no. 4, pp. 43:1–43:35, Aug. 2013. [Online]. Available: http://doi.acm.org/10.1145/2501654.2501657

[58] S. Fortunato and A. Lancichinetti, “Community detection algorithms: A comparative analysis: Invited presentation, extended abstract,” in *Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools*, ser. VALUETOOLS ’09. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009, pp. 27:1–27:2. [Online]. Available: http://dl.acm.org/citation.cfm?id=1698822.1698858

[59] E. A. Leicht and M. E. J. Newman, “Community structure in directed networks,” *Phys. Rev. Lett.*, vol. 100, p. 118703, Mar 2008. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevLett.100.118703

[60] T. Yang, Y. Chi, S. Zhu, Y. Gong, and R. Jin, “Directed network community detection: A popularity and productivity link model.”

[61] J. W. Berry, B. Hendrickson, R. A. LaViolette, and C. A. Phillips, “Tolerating the community detection resolution limit with edge weighting,” *Phys. Rev. E*, vol. 83, p. 056119, May 2011. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevE.83.056119

[62] D. S. Bassett, M. A. Porter, N. F. Wymbs, S. T. Grafton, J. M. Carlson, and P. J. Mucha, “Robust detection of dynamic community structure in networks,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 23, no. 1, pp. –, 2013. [Online]. Available: http://scitation.aip.org/content/aip/journal/chaos/23/1/10.1063/1.4790830

[63] S. Bansal, S. Bhowmick, and P. Paymal, “Fast community detection for dynamic complex networks,” in *Complex Networks*, ser. Communications in Computer and Information Science, L. da F. Costa, A. Evsukoff, G. Mangioni, and R. Menezes, Eds. Springer Berlin Heidelberg, 2011, vol. 116, pp. 196–207. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-25501-4_20

[64] J. Yang and J. Leskovec, "Overlapping community detection at scale: A nonnegative matrix factorization approach," in *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, ser. WSDM '13. New York, NY, USA: ACM, 2013, pp. 587–596. [Online]. Available: http://doi.acm.org/10.1145/2433396.2433471

[65] T. Chakraborty, S. Kumar, N. Ganguly, A. Mukherjee, and S. Bhowmick, "GenPerm: An Unified Method For Finding Overlapping and Non-overlapping Communities." 2015, accepted to TKDE.

[66] L. Bohlin, D. Edler, A. Lancichinei, and M. Rosvall, "Community detection and visualization of networks with the map equation framework," 2014.

[67] V. D. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of community hierarchies in large networks," *CoRR*, vol. abs/0803.0476, 2008.

[68] S. Fortunato and C. Castellano, "Community structure in graphs," in *Ency. of Complexity and Systems Science*, 2009.

[69] X. Dong, P. Frossard, P. Vandergheynst, and N. Nefedov, "Clustering with multi-layer graphs: A spectral perspective," *IEEE Transactions on Signal Processing*, vol. 60, no. 11, pp. 5820–5831, 2012.

[70] H. Li, Z. Nie, W.-C. Lee, L. Giles, and J.-R. Wen, "Scalable community discovery on textual data with relations," in *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, 2008, pp. 1203–1212.

[71] Z. Xu, Y. Ke, Y. Wang, H. Cheng, and J. Cheng, "A model-based approach to attributed graph clustering," in *Proceedings of the 2012 ACM SIGMOD international conference on management of data*. ACM, 2012, pp. 505–516.

[72] D. Cai, Z. Shao, X. He, X. Yan, and J. Han, "Mining hidden community in heterogeneous social networks," in *Proceedings of the 3rd international workshop on Link discovery*. ACM, 2005, pp. 58–65.

[73] H. Zhang, C.-D. Wang, J.-H. Lai, and S. Y. Philip, "Modularity in complex multilayer networks with multiple aspects: a static perspective," in *Applied Informatics*, vol. 4, no. 1. Springer Berlin Heidelberg, 2017, p. 7.

[74] J. D. Wilson, J. Palowitch, S. Bhamidi, and A. B. Nobel, "Community extraction in multilayer networks with heterogeneous community structure," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 5458–5506, 2017.

[75] C. Shi, Y. Li, J. Zhang, Y. Sun, and S. Y. Philip, "A survey of heterogeneous information network analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, 2017.

[76] Y. Sun and J. Han, "Mining heterogeneous information networks: a structural analysis approach," *ACM SIGKDD Explorations Newsletter*, vol. 14, no. 2, pp. 20–28, 2013.

[77] C. Wang, Y. Sun, Y. Song, J. Han, Y. Song, L. Wang, and M. Zhang, "Relsim: relation similarity search in schema-rich heterogeneous information networks," in *Proceedings of the 2016 SIAM International Conference on Data Mining.* SIAM, 2016, pp. 621–629.

[78] C. Wang, Y. Song, H. Li, M. Zhang, and J. Han, "Text classification with heterogeneous information network kernels." in *AAAI*, 2016, pp. 2130–2136.

[79] J. Zhang, P. S. Yu, and Y. Lv, "Organizational chart inference," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM, 2015, pp. 1435–1444.

[80] C. Shi, Y. Li, S. Y. Philip, and B. Wu, "Constrained-meta-path-based ranking in heterogeneous information network," *Knowledge and Information Systems*, vol. 49, no. 2, pp. 719–747, 2016.

[81] C. Shi, Z. Zhang, P. Luo, P. S. Yu, Y. Yue, and B. Wu, "Semantic path based personalized recommendation on weighted heterogeneous information

networks," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management.* ACM, 2015, pp. 453–462.

[82] B. A.J., M. M.P., C. A., and A. F., "A multilayer network approach for guiding drug repositioning in neglected diseases." *PLoS Negl Trop Dis*, vol. 20, no. 1.

[83] D. Melamed, "Community structures in bipartite networks: A dual-projection approach," *PloS one*, vol. 9, no. 5, p. e97823, 2014.

[84] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social networks*, vol. 1, no. 3, pp. 215–239, 1978.

[85] Y. Rochat, "Closeness centrality extended to unconnected graphs: The harmonic centrality index," in *ASNA*, no. EPFL-CONF-200525, 2009.

[86] A. Dekker, "Conceptual distance in social network analysis," *Journal of Social Structure (JOSS)*, vol. 6, 2005.

[87] M. De Domenico, A. Solé-Ribalta, E. Omodei, S. Gómez, and A. Arenas, "Centrality in interconnected multilayer networks," *arXiv preprint arXiv:1311.2906*, 2013.

[88] A. Solé-Ribalta, M. De Domenico, S. Gómez, and A. Arenas, "Centrality rankings in multiplex networks," in *Proceedings of the 2014 ACM conference on Web science.* ACM, 2014, pp. 149–155.

[89] A. Cardillo, J. Gómez-Gardenes, M. Zanin, M. Romance, D. Papo, F. Del Pozo, and S. Boccaletti, "Emergence of network features from multiplexity," *Scientific reports*, vol. 3, 2013.

[90] J. Banerjee, C. Zhou, A. Das, and A. Sen, "On robustness in multilayer interdependent networks," in *Critical Information Infrastructures Security*, E. Rome, M. Theocharidou, and S. Wolthusen, Eds. Cham: Springer International Publishing, 2016, pp. 247–250.

[91] M. De Domenico, A. Solé-Ribalta, S. Gómez, and A. Arenas, “Navigability of interconnected networks under random failures,” *Proceedings of the National Academy of Sciences*, 2014.

[92] J. D. Wilson, J. Palowitch, S. Bhamidi, and A. B. Nobel, “Community extraction in multilayer networks with heterogeneous community structure,” *J. Mach. Learn. Res.*, vol. 18, 2017.

[93] A. Rai, “Mln-subdue: Decoupling approach-based substructure discovery in multilayer networks (mlns),” Master’s thesis, Univ. of Texas at Arlington, May 2020.

[94] OpenSourceData, “The internet movie database,” 2018. [Online]. Available: ftp://ftp.fu-berlin.de/pub/misc/movies/database/

[95] ——, “Dblp dataset,” 2018. [Online]. Available: http://dblp.uni-trier.de/xml/

[96] R. Elmasri, *Fundamentals of database systems.* Pearson Education India, 2008.

[97] J. Edmonds, “Maximum matching and a polyhedron with 0, 1-vertices,” *Journal of research of the National Bureau of Standards B*, vol. 69, no. 125-130, pp. 55–56, 1965.

[98] B. Abrahao, S. Soundarajan, J. Hopcroft, and R. Kleinberg, “On the separability of structural classes of communities,” in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’12. New York, NY, USA: ACM, 2012, pp. 624–632. [Online]. Available: http://doi.acm.org/10.1145/2339530.2339631

[99] D. Chakrabarti, Y. Zhan, and C. Faloutsos, “R-MAT: A recursive model for graph mining,” in *In SDM*, 2004.

[100] “Road safety - accidents 2014,” 2014. [Online]. Available: https://data.gov.uk/dataset/road-accidents-safety-data/resource/1ae84544-6b06-425d-ad62-c85716a80022

[101] V. Labatut, “Generalized measures for the evaluation of community detection methods,” *CoRR*, vol. abs/1303.5441, 2013.

[102] M. Kosinski, S. Matz, S. Gosling, V. Popov, and D. Stillwell, “Facebook as a social science research tool,” *American Psychologist*, 2015.

[103] F. Mairesse, M. A. Walker, M. R. Mehl, and R. K. Moore, “Using linguistic cues for the automatic recognition of personality in conversation and text,” *J. Artif. Int. Res.*, pp. 457–500, 2007.

[104] J. Costa, P. T. and R. R. McCrae, *The Revised NEO Personality Inventory (NEO-PI-R)*, 2008, pp. 179–198.

[105] A. H. Pashakhanlou, “Fully integrated content analysis in international relation,” *International Relations*, vol. 31, no. 4, pp. 447–465, 2017.

[106] C. Sumner, A. Byers, and M. Shearing, “Determining personality traits and privacy concerns from facebook activity,” *Black Hat Briefings*, pp. 197–221, 2011.

[107] X.-S. Vu and L. Jiang, “Self-adaptive privacy concern detection for user-generated content,” in *Proceedings of the 19th International Conference on Computational Linguistics and Intelligent Text Processing*, 2018.

[108] J. W. Pennebaker and L. A. King, “Linguistic styles: Language use as an individual difference.” *Journal of Personality and Social Psychology*, pp. 1296–1312, 1999.

[109] F. Celli, F. Pianesi, D. S. Stillwell, and Kosinski, “Workshop on computational personality recognition (shared task),” *The Seventh International AAAI Conference on Weblogs and Social Media*, 2013.

[110] T. Vu, D. Q. Nguyen, X.-S. Vu, D. Q. Nguyen, and M. Trenell, “Nihrio at semeval-2018 task 3: A simple and accurate neuralnetwork model for irony

detection in twitter," in *Proceedings of the 12nd International Workshop on Semantic Evaluation (SemEval-2018)*, 2018, pp. 525–530.

[111] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004, pp. 168–177.

[112] X.-S. Vu, L. Flekova, L. Jiang, and I. Gurevych, "Lexical-semantic resources: yet powerful resources for automatic personality classification," in *Proceedings of the 9th Global WordNet Conference*, 2018, pp. 173–182.

[113] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[114] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013.

[115] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, "Character-aware neural language models," ser. AAAI'16, 2016, pp. 2741–2749.

[116] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, 1989.

[117] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," ser. ICML'10, 2010, pp. 807–814.

[118] C. Zhou, C. Sun, Z. Liu, and F. C. M. Lau, "A C-LSTM neural network for text classification," *CoRR*, 2015.

[119] Wikipedia contributors, "United states presidential election, 2008 — Wikipedia, the free encyclopedia," 2018. [Online]. Available: https://en.wikipedia.org/w/index.php?title=United_States_presidential_election,_2008&oldid=850473763

[120] ——, "Barack obama presidential campaign, 2008 — Wikipedia, the free encyclopedia," 2018. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Barack_Obama_presidential_campaign,_2008&oldid=838416057

[121] Jordan Fabian, "Poll: Young people (still) least likely to vote," 2013. [Online]. Available: https://splinternews.com/poll-young-people-still-least-likely-to-vote-1793839910

[122] Kimberly Amadeo, "What has obama done? 13 major accomplishments," 2018. [Online]. Available: https://www.thebalance.com/what-has-obama-done-11-major-accomplishments-3306158

[123] Roper Center for Public Opinion Research, Cornell University, "How groups voted in 2008," 2008. [Online]. Available: https://ropercenter.cornell.edu/polls/us-elections/how-groups-voted/how-groups-voted-2008/

[124] P. Azoulay, B. Jones, J. D. Kim, and J. Miranda, "Age and high-growth entrepreneurship," National Bureau of Economic Research, Tech. Rep., 2018.

[125] S. J. Cash and J. A. Bridge, "Epidemiology of youth suicide and suicidal behavior," *Current opinion in pediatrics*, vol. 21, no. 5, p. 613, 2009.

[126] D. G. Blanchflower and A. J. Oswald, "Is well-being u-shaped over the life cycle?" *Social science & medicine*, vol. 66, no. 8, pp. 1733–1749, 2008.

[127] Kathleen Doheny, "Midlife crisis: Transition or depression?" 2009. [Online]. Available: https://www.webmd.com/depression/features/midlife-crisis-opportunity#1

[128] M. Rowan and J. Dehlinger, "Observed gender differences in privacy concerns and behaviors of mobile device end users," *Procedia Computer Science*, vol. 37, pp. 340–347, 2014.

[129] J. Stolworthy, "Dark universe: Johnny depp and javier bardem join tom cruise in universal's monster movie franchise," *https://www.independent.co.uk/us*, 2017.

[130] A. Diaz, "Major airline hubs," *https://www.travelmiles101.com/list-of-major-airline-hubs/*, 2017.

[131] Forbes, “The world’s largest public companies,” 2018.

[132] J. Dawes, “Grand rapids is ‘sweet spot’ for airline base,” *Grand Rapids Business Journal (GRBJ)*, 2019.

[133] H. N. Gabow, “The weighted matching approach to maximum cardinality matching,” *Fundamenta Informaticae*, vol. 154, no. 1-4, pp. 109–130, 2017.

[134] K. Komar, “Data-driven modeling of heterogeneous multilayer networks and their community-based analysis using bipartite graphs,” Master’s thesis, Univ. of Texas at Arlington, Aug. 2019. [Online]. Available: http://itlab.uta.edu/students/alumni/MS/Kanthi_Sannappa_Komar/KanthiK_MS2019.pdf

[135] M. E. Newman, “Modularity and community structure in networks,” *Proceedings of the national academy of sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.

BIOGRAPHICAL STATEMENT

Abhishek Santra received his Bachelor of Science (Honours) degree in Computer Science from Hansraj College, University of Delhi, India in June 2011. Following this, he completed his Master's in Computer Science from University of Delhi in June 2013. He worked as an independent researcher at the South Asian University, New Delhi, India till June 2014. He began his doctoral research at the University of Texas at Arlington in August 2014 under the supervision of Dr. Sharma Chakravarthy. He has served as a Graduate Teaching Assistant in the Department of Computer Science and Engineering at The University of Texas at Arlington from 2014 till 2020. He is the recipient of the GTA and the STEM fellowship from 2014 to 2020. In his last semester, he was awarded the Dissertation Fellowship. In 2018, he was part of a collaborative international research project between Umeå University (Sweden) and University of Texas at Arlington (USA) funded by the Swedish Foundation for International Cooperation in Research and Higher Education (STINT), which even required him to undertake a research travel to Sweden. During his Ph. D., he has been a co-author of 10+ conference and journal papers, that have garnered 40+ citations. He has served as an external reviewer for DAWAK, DEXA and PAKDD and also served in the program committee of the International Conference on Big Data Analytics and Knowledge Discovery (DaWaK) conference.