universität wien

MASTERARBEIT | MASTER'S THESIS

Titel | Title

Information Access Clustering Based on Linear Threshold Model Simulations

verfasst von | submitted by

David Lisica

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien | Vienna, 2026

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 921
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Masterstudium Informatik
Betreut von | Supervisor:	Ass.-Prof. Dott.ssa Dott.ssa.mag. Yllka Velaj PhD

Acknowledgments

As I conclude my formal education with this thesis, I feel a deep sense of gratitude. Many people have offered me their help and support throughout this long journey, though I will not mention them by name; each of them knows who they are. My endless thanks to all of you.

However, I must mention two names: my mother, Anica, and my father, Renato. Their unconditional love and support have been my pillars throughout the years, and without them, this document, like many before it, would not have seen the light of day.

Abstract

Connections and position in social networks are key determinants of one's influence and power, as they constitute core components of social capital, allowing individuals to access resources, information and opportunities that would otherwise remain unavailable.

Building on social capital theory, formal network representations, diffusion models, and prior work on clustering algorithms based on information access, this thesis proposes a novel method for grouping nodes in a network according to their access to information, measured by the similarity of information sources and information diffusion pathways. Previous work on this topic relied exclusively on the Independent Cascade Model (ICM), whereas this thesis adopts the Linear Threshold Model (LTM).

Empirically, the approach is evaluated on three single-layer and two multi-layer datasets, each instantiated with four edge-weight assignment strategies (weighted, random, trivalent, and uniform). The equivalence between the LTM and node reachability via live-edge graphs is explained, and algorithmic performance is compared using the Python NDLib library.

For every combination of parameters, information access clustering under the LTM is compared against the well-known baseline methods. Across different variations of parameters, the proposed method produces clusters comparable to those obtained by well-established community-detection approaches, while offering a transparent and interpretable perspective in terms of access to information.

Kurzfassung

Verbindungen und Positionen in sozialen Netzwerken sind zentrale Determinanten von Einfluss und Macht, da sie wesentliche Bestandteile des sozialen Kapitals darstellen und Individuen den Zugang zu Ressourcen, Informationen und Chancen ermöglichen, die ihnen andernfalls verschlossen blieben.

Aufbauend auf der Theorie des sozialen Kapitals, formalen Netzwerkrepräsentationen, Diffusionsmodellen sowie früheren Arbeiten zu Clustering-Algorithmen auf Basis von Informationszugang schlägt diese Arbeit eine neuartige Methode zur Gruppierung von Knoten in Netzwerken vor, die auf ihrem Zugang zu Informationen basiert. Dieser wird durch die Ähnlichkeit von Informationsquellen sowie von Informationsdiffusionspfaden quantifiziert. Während bisherige Arbeiten in diesem Bereich ausschließlich das Independent Cascade Model (ICM) verwendeten, kommt in dieser Arbeit das Linear Threshold Model (LTM) zum Einsatz.

Die Methode wird empirisch anhand von drei Single-Layer- und zwei Multi-Layer-Datensätzen evaluiert, wobei jeweils vier Strategien zur Kantengewichtung betrachtet werden (gewichtet, zufällig, trivalent und uniform). Die Äquivalenz zwischen dem LTM und der Knotenerreichbarkeit über Live-Edge-Graphen wird hergeleitet, und die algorithmische Performance wird mithilfe der Python-Library NDLib verglichen.

Für jede Parameterkombination wird das Informationszugangs-Clustering unter dem LTM mit etablierten Baseline-Methoden verglichen. Über verschiedene Parametervariationen hinweg liefert der vorgeschlagene Ansatz Cluster, die mit jenen klassischer Community-Detection-Verfahren vergleichbar sind, zugleich jedoch eine transparente und interpretierbare Perspektive im Hinblick auf den Zugang zu Informationen bieten.

Contents

1 Introduction

Information is considered one of the most valuable assets a person can have and can be leveraged for various types of economic and social gain. There are several different ways of information transmission, from old-fashioned word of mouth and mass media to the more recent phenomena such as social media platforms and messaging applications. Whatever the way, it is known that the spread of information is quite a significant process that people want to be a part of, especially for information that is of high importance or interest to them.

The object of the research will be a social network in a broader sense, not just as a platform, but any connected group of people: in both a real and a digital world. For that purpose, a social network will be mathematically represented as a directed weighted graph; nodes are members, and edges are links between them.

This value of possession of information opens the question of how one can access it, from which sources, and in what way. This triangle of questions forms a concept of information access as social capital [10, 16, 51]; a feature that is attributed to a member of a network describing its position and importance in the flow of information. The constant desire of people to receive more information made information access one of the key elements of structural social capital [15], as a consequence of one's position and influence in the network.

Granovetter argued in [24] that weak connections in social networks can play a more significant role in information transfer than strong ones. It is understood that a *weak tie* is some distant connection (a person that one knows, a colleague at work, or an old classmate), while a *strong tie* is a family member, a spouse, or a close friend. The idea behind this theory is that strong ties are there to keep a community together, while weak ones are utilized as bridges between different social circles which often convey a wide range of information.

People are often unaware of the way their social environment shapes their life. Christakis and Fowler in [13] made a huge contribution to understanding how the social network that a person is in defines one in many known and unknown dimensions of life. For example, they explain how suicide epidemics happened in some communities in the USA; how the probability of one becoming obese, starting smoking, or becoming alcoholic can be computed from the network structure; and how more ideologically intense people tend to have more dense and closed communities. Although some ideas they present are transmitted through subconscious channels, those that are conscious and related to information are more interesting for this research. Nevertheless, the underlying principle is the same: a person's opinion and/or behavior is influenced by people not only very close to them but also by people who are a couple of steps distant from them in the network.

The main objective of this thesis is to create a method that will be able to distinguish

members of a network based on their individual access to information, which is the likelihood of receiving information from all other members of that network. Our goal is to attribute an information access representation to each node and cluster them based on those values.

In order to formally represent the flow of information within a network, Kempe [30] introduced two models of diffusion. The Independent Cascade Model (ICM) is a stochastic diffusion model in which each newly activated node has a single independent chance to activate each of its inactive neighbors with a given probability. The Linear Threshold Model (LTM) is a diffusion model in which a node becomes activated when the combined influence of its active neighbors exceeds a predefined threshold.

This research is based on the existing formal representation of information access [4, 28], which uses the Independent Cascade Model (ICM) as the underlying diffusion model. The main contribution of this thesis is the extension of this method to the Linear Threshold Model (LTM), making the analysis of information access possible under a different diffusion logic based on cumulative influence rather than independent activation events.

Unlike traditional community detection approaches that cluster nodes based on structural properties (degree, centrality, edge density, etc.), this work groups nodes according to their access to information. Information access is computed through diffusion processes (LTM), rather than being inferred implicitly from network topology alone. The proposed approach introduces a similarity measure based on shared information sources and diffusion pathways, hence capturing not only whether nodes are connected but also how information reaches them within the network.

Moreover, in addition to using the method only in single-layer networks (graphs with one set of nodes and a single set of edges), this research extends information access clustering to multi-layer networks, defined as graphs with a common set of nodes and multiple edge layers representing different types of connections [7].

Building on the related work, this thesis will try to answer the following research questions:

- Can nodes be clustered according to their information access and can we prove that each of those groups has a certain level of significance using an external measure (node attribute)?
- How is the clustering result affected by changing model parameters?
- Can clustering work on multi-layer networks? How do those results differ from using each layer as an independent single-layer graph?
- What distinguishes information access clustering from other established clustering algorithms?

Thesis structure

Chapter 2 introduces the related work on influence maximization, social capital, and clustering based on information diffusion. Chapter 3 provides a theoretical background on

the topic of this thesis, as well as an overview of the datasets, together with a description of the proposed clustering method. Chapter 4 presents the experimental results followed by their comparison, scalability, and effectiveness analysis. Chapter 5 explores options for improving the algorithm by reducing execution time. Chapter 6 concludes the thesis by summarizing the key findings, limitations of this method, and potential for future research and improvements.

Used tools

We are stating that the author of this thesis utilized Claude and GitHub Copilot for text reformatting and code snippets for supporting code and graphics in Python.

2 Related Work

This chapter presents the relevant existing literature for this research in the areas of diffusion models, influence maximization, social capital, and clustering methods. The two most important papers that deserve to be mentioned before any other are [4] and [28], which serve as research bases.

Beilinson et al. [4] introduced the concept of clustering through information access in networks. The research utilized the concept of structural social capital and the formal definition of the Independent Cascade Model, provided by [30], to introduce a mathematical representation of the access to information as the social capital of each node in the network. Furthermore, the paper provided a novel k-means-based clustering algorithm and showed its performance on real-world datasets. It is important to mention that this research used only uniform edge-weights, i.e. all edges in the graph have the same predefined weight.

The information access clustering method was extended by Halmdienst [28] by introducing several edge-weight assignment strategies (weighted, trivalency, and random). The study concludes that different edge-weighting strategies are more suitable for graphs with different densities. Nevertheless, the results obtained are comparable to those reported in [4], suggesting that the clustering performance is robust to the specific choice of edge-weights under the Independent Cascade Model.

2.1 Influence Maximization

The most important research on influence maximization was conducted by Kempe et al. [30], and focuses on how information spreads in a social network. The network is modeled as a directed graph, where nodes correspond to users and edges to social connections between them. Each connection has a value (edge-weight) that indicates how strongly one user can influence another. The goal is to select an initial set of users who, when initially informed, can cause the largest possible spread of information in the network. This spread depends on a diffusion process that models how users influence each other.

In addition, the mechanisms causing the spread of information in social networks have been studied from several perspectives. Motivated by applications such as influence maximization, viral marketing, and large-scale information spread on online platforms, a significant number of papers focus on maximizing the reach of diffusion processes. Studies such as [3, 11, 36, 49] address how to identify strategies or influential nodes that maximize the spread of information using various diffusion models.

Much of the existing literature focuses on limiting or controlling the spread of information, which is closely related to studies on the suppression of information diffusion. This

is particularly valuable in applications such as network security, epidemic containment, and misinformation mitigation. Studies by [33, 35] investigate structural interventions, mainly edge removal, to suppress diffusion processes in threshold-based models, although these approaches are primarily heuristic and lack formal approximation guaranties. More rigorous formulations are proposed by [32], who study edge addition and deletion with the objective of minimizing information spread and demonstrate that the resulting network modification problem has a supermodular structure, which allows the use of algorithms with formal approximation guaranties, as opposed to previous studies.

This line of work is further extended in [56], which considers both edge and node removal and develops methods that combine theoretical performance computations with strong empirical effectiveness. Related contributions also address diffusion control from an application-oriented perspective, such as reducing the reach of false or harmful information, as done by Song et al. in [50]. Taken together, these studies highlight that, depending on the application context, controlling diffusion through network modification is as central as maximizing diffusion, and that structural properties of the network play a critical role in shaping propagation outcomes.

2.2 Social Capital

The investigation of social capital, before it was introduced in computer science and network research, was started by social studies scientists. Coleman in [16] defines *social capital* as the resources available to individuals and groups through their social networks and relationships. He puts it next to the concept of *human capital*, which refers to the skills, knowledge, and abilities possessed by individuals that can be used for economic gain. His work gives proof that the social position of a person (social capital) plays a significant role in quantitative life success (human capital), with the education of children from families with stronger social networks as an example. Although Coleman's work does not use formal network models, it provides a foundational motivation for this thesis by establishing that access to resources is strongly dependent on social structure. This idea underlies the central assumption of this research, where access to information is explicitly modeled as a consequence of one's position in a network.

Bourdieu [9] defines three types of capital: economic, cultural, and social. He states that social capital is the sum of all real or potential resources that come from strong relationships. His point of view emphasizes that social capital comes from being part of a group that helps and benefits its members. The paper shows how social capital can reproduce social inequalities and how it fits into larger power structures. Because of this, social capital can bring people together or keep them apart, which is why this work is important to understand how social capital is built into social networks and exchanges. In contrast to Bourdieu's qualitative and power-oriented perspective, this thesis translates the abstract notion of social capital into quantitative and measurable differences in information access within a network.

Claridge gave a general description of the types of social capital in [15], where he proposed categorization on two axes. The first axis is a dimension in which social capital

can be structural (configurations and patterns), relational (characteristics and qualities), and cognitive (shared understanding). The second axis explains on which level capital is, such as micro, meso, or macro. This paper had a significant impact on other scholars because it provided a comprehensive identification of different types of social capital. This typology is particularly relevant for this thesis, as it allows the present work to be explicitly situated within the structural dimension of social capital at the micro (individual node with its local neighborhood) and meso (groups of nodes with similar patterns) levels.

He further defined structural social capital as the observable and visible components of social relationships [15], referring mainly to the institutions, networks, and connections that link people or groups. The number and structure of relationships, including who knows whom, how often they interact, and through what channels are all included in this type of social capital. Following this definition, the present thesis operationalizes structural social capital by modeling network topology and quantifying how information propagates through these observable connections using diffusion processes. Although Claridge's contribution is primarily conceptual, this thesis extends his ideas by providing a computational method to measure and compare structural social capital based on the access of each node to information.

This problem was also studied as a more general question: how to distinguish *homophily* from real social influence. Homophily explains the tendency of people to connect with individuals similar to them. Research on this topic [2] was conducted on one messaging platform and provided evidence that the presence of homophily is usually underestimated and plays a more significant role in the network than expected. Although important, this distinction is outside the scope of this research. But generally speaking, information access clustering based on diffusion ideally would account for whether similarities come from genuine information propagation or from pre-existing structural similarity among nodes.

Differences in access to information naturally impose the problem of structural inequality of network members, which is defined as *access gap* by [20]. They are offering an approach to influence maximization that has not been addressed before: "how best to spread information in a social network while minimizing this access gap" [20], by improving network connectivity for underrepresented groups. This method aims to mitigate these gaps and provide more equal access to information. Although Fish et al. focus on intervention strategies to reduce information access gaps, this thesis addresses a complementary problem by measuring and comparing information access itself as an intrinsic property of the network, without assuming a specific intervention or optimization objective. Those questions can be tackled as future improvements or implementations.

Burt explored in [10] how social capital develops from the configuration of social networks, particularly through the presence of *structural holes*, gaps between otherwise unconnected groups. The paper argues that individuals who connect these gaps act as brokers, gaining access to various information and resources that give them a competitive advantage. There are two types of network according to this theory: closed ones encourage cohesion and trust, while open ones promote innovation and higher connectivity through brokerage. One of the key findings of the paper is that people in brokerage positions have

a higher probability of creating novelties and achieving career growth.

Lin [37] suggests a theory that links personal resources to the structures of social networks to explain how people can access and use social capital. He states that social capital is both a resource and a way to get other types of capital to work together. This theory combines the structural (network-based) and functional (use-oriented) parts of social capital. The framework proposed in the paper has been tested in real-life job searches and social mobility and helps to turn vague ideas about social capital into measurable relational constructs. Although Lin connects the network structure to concrete socioeconomic outcomes, this thesis abstracts away from specific outcomes and instead focuses on quantifying access to information itself as a structural feature of the network.

The differences in democratic traditions and the strength of the institutions in different parts of Italy inspired Putnam [43] to investigate their causes. The paper states that the northern Italian regions have significantly higher government effectiveness than the southern regions due to different levels of social capital. Social capital is defined as the norms, networks, and trust that enable participants to act together more effectively. Putnam's work connects sociology and political science by showing how social capital affects things on a large scale. It puts social capital at the center of how well institutional democracy works. Putnam's macro-level analysis motivates the broader importance of social capital, while the present thesis contributes a micro-level, network-based formalization suitable for computational analysis.

The focus of this research is on the structural dimension of social capital, with an emphasis on information as an example. There are two aspects of looking at it, defined by Beilinson et al. in [4]; the first one suggests that information access of every network member is determined by network topology, i.e. one's position and number and type of connections to other members. The more influential people a person knows, the more information one has access to. On the other hand, the second aspect claims that access to information (consequently, amount of information) is a feature that contributes to building the network itself, meaning that more informed people are going to be placed more centrally in the network as opposed to members with lack of that feature that will occupy peripheral network positions. These two aspects lead us to the chicken and an egg problem: is information access a cause or an effect of network topology? Following this idea, [4] left the open question in the conclusion of the paper whether information access is really a feature *external* to the network or not. This thesis builds directly on this open question by explicitly modeling information access through diffusion processes and using it as a basis for comparing nodes and identifying structural similarities in both single-layer and multi-layer networks.

2.3 Clustering Methods

In order to distinguish nodes by how influential they are, we need to give a brief overview of clustering methods in a graph. Our goal is to find out which existing algorithms can be compared to information access clustering, that is developed using the theory provided in the next chapter. In particular, the methods are discussed with respect to whether

they rely purely on static topology, latent representations, or explicit diffusion processes and how these assumptions relate to the notion of information access that underlies the proposed method.

Core-periphery clustering. This method aims to group the nodes in a dense core and a sparse periphery [8, 44]. The ground for this idea is visible in nature and online: there is a small and well-connected minority that is in the core, while the majority is on the periphery sparsely connected while maintaining links to the core. It is a representation of inequality in real-world networks, making this method valuable for comparison with information access clustering. This method utilizes an adjacency matrix and k-core decomposition to identify where there is a dense core in a graph. Although the Core-periphery (CP) method identifies an inequality based on connection density, it does not model how information flows and propagates through the network. However, information access clustering differentiates nodes according to the results of diffusion processes, meaning that nodes outside the core may still have high information access depending on diffusion dynamics.

Role detection. A technique that assigns a role to each node in the graph is called Role2Vec [1]. It performs random walks on the graph trying to capture the structural role for each node based on overall position, not only on neighbors. Using values obtained for each node in random walks, k-means clustering[1] is performed to obtain groups of nodes with similar properties, i.e. similar structural roles in the graph. This approach focuses on structural equivalence rather than functional influence. Unlike information access clustering, Role2Vec does not take into account diffusion outcomes and, therefore, captures similarity in network position rather than similarity in access to information.

Spectral methods. The clustering algorithm proposed in [54] is called Spectral clustering. It is a method based on the eigenvalues and eigenvectors of matrices derived from the graph, for example, adjacency or Laplacian matrix. The algorithm calculates the first k eigenvectors to project the graph's nodes into a lower-dimensional space. After that, the traditional k-means method is used to identify clusters in a matrix built from previously computed eigenvectors. Another valuable spectral method is *SpectralMix* [48]: a technique that takes into consideration categorical node attributes, as well as multiple types of connection in the graph. The proposed information access clustering is similar to spectral methods in that it also utilizes data in the form of a matrix as a starting point for clustering. Although the matrix in spectral methods is derived from static properties, the proposed method argues that information access emerges from dynamic diffusion processes and may not be fully captured by eigenstructure alone.

Community detection. Another method of particular value is Fluid communities [40] which assumes that every node in the network can belong to some community, or many of them, with changeable boundaries. This algorithm is especially suitable for time-evolving graphs and uses random walks to determine the probability that every node v belongs to

[1]Generally speaking, the **k-means algorithm** [5] is partitioning n samples into k groups. First, for each cluster, its centroid (mean) is computed. After that, for each sample point method is finding the nearest centroid using L^2 norm, and assigning the node to the corresponding centroid's cluster.

each of the k communities (k is a predefined parameter). Those probabilities are close to the real-world logic: each person is rarely a member of only one community. Another significant community detection algorithm is the Louvain method [6] which is trying to optimize the modularity of each cluster; maximizing the number of edges within the cluster while minimizing those that connect it to other ones. The result is a structure that shows dense connections within groups and sparse connections between them. Both methods optimize the community structure based on edge density or modularity. Information access clustering differs in that it does not require clusters to be densely connected; instead, nodes are grouped according to similarity in information reachability, which may span across traditional community boundaries.

Node embedding methods. Node2Vec [27] is a popular technique to simulate biased random walks to learn continuous vector representations of nodes in a graph. It models the frequency with which nodes occur together in walk sequences to capture patterns of structural similarity. The main advantage of Node2Vec is that, depending on how the walks are set up, it can encode both structural equivalence (nodes with similar roles) and homophily (nodes connected to similar others). It is better suited for structural clustering than for functional or influence-based analysis because it does not simulate the flow of information or influence through the network. Also worth mentioning is Diff2Vec (D2V) [47], and it learns the node embeddings based on the simulation of the information diffusion process, either Independent Cascade Model or Linear Threshold Model. This algorithm captures node representation that describes its function, not only topological position and proximity to other nodes. Although from a high-level perspective this method is similar to information access clustering, the underlying methodology is different. Diff2Vec clusters nodes by first simulating diffusion from each node to build diffusion graphs, extracting Euler-tour sequences from these subgraphs, feeding those into a skip-gram model to learn low-dimensional embeddings, and finally applying a standard clustering algorithm (e.g., k-means) on those embeddings. In contrast, information access clustering avoids intermediate embedding learning and directly constructs a matrix based on diffusion outcomes, which improves interpretability and explicitly ties the clustering objective to information access representation.

Attributed Network Representation Learning. ANRL [57] learns the node embeddings by jointly modeling graph topology and node attributes in a deep architecture. It preserves the local neighborhood structure while merging attribute information, so that nodes with similar features and connections lie close to each other in the learned representation space. The learned representations are typically clustered with standard algorithms (e.g., k-means). It is well suited for attributed graphs where structural and semantic signals are complementary.

Deep Graph Infomax. DGI [53] is a self-supervised method that maximizes mutual information between local node embeddings and a global summary of the graph. By contrasting real graph signals with corrupted versions, it pushes an informative structure into the embeddings without labels. The resulting representations are strong for downstream clustering. It is lightweight and general, but requires a good corruption strategy and

encoder choice. Information access clustering differs fundamentally by not optimizing representation quality through mutual information but by directly measuring similarity in diffusion behavior, avoiding architectural and hyperparameter dependencies.

Deep Multi-Graph Clustering. DMGC [38] is an unsupervised deep method that groups nodes across multiple related graphs while simultaneously learning which clusters correspond to each other in graphs. It first learns node embeddings for each graph with an autoencoder that preserves both first-order (edge-level) and second-order (neighborhood) proximities. This method outputs graph clusters and cross-graph associations and was shown to outperform strong baselines on several datasets. While DMGC learns latent representations across graphs, the proposed method focuses on diffusion-based similarity in a multi-layer graph, offering a model-driven alternative to autoencoder-based approaches.

Cross-Network Embedding for Multi-Network Alignment. CrossMNA [14] embeds nodes from different but related networks in a shared space while enforcing alignment constraints (e.g., via known anchors or structural correspondences). Although designed for alignment, shared embeddings can be directly clustered to reveal cross-network communities. It is useful when one needs comparable clusters across networks or platforms. In contrast, information access clustering does not require anchor nodes or alignment constraints, as similarity is computed from shared diffusion behavior across layers.

Heterogeneous Attention Network. HAN [55] addresses heterogeneous information networks using hierarchical attention: node-level attention aggregates neighbors, and semantic-level attention weights different semantic paths. This yields meta-path-aware embeddings that capture rich, typed relations. Clustering on HAN embeddings can uncover communities defined by specific semantics, while choosing meta-paths critically affects results. The reliance on predefined meta-paths distinguishes HAN from the proposed approach, which derives similarity directly from diffusion processes without requiring semantic path design.

Multi-relational Network Embeddings with Relational Proximity and Node Attributes. MARINE [19] learns the embeddings in multi-layer graphs by jointly modeling relation-specific proximities and node attributes. The embeddings are then used for standard clustering methods. It is effective when multiple types of relation and attributes both matter.

Deep Multi-Graph Infomax. DMGI [39] extends infomax-style representation learning to multi-layer attributed graphs. It maximizes the agreement between embeddings and global summaries within each view (layer) while enforcing cross-view consensus and leveraging node attributes. This makes the learned space robust to noisy or incomplete layers. Clustering on DMGI embeddings often benefits when layers are complementary but variably reliable. Unlike DMGI, the proposed information access method does not enforce representation consensus but compares diffusion outcomes across layers directly.

Community Detection using Nonnegative Matrix Factorization Methods. [58] introduced a series of methods based on nonnegative matrix factorization (NMF) to jointly leverage graph connectivity and Twitter content features. In addition to the standard k-means

baseline on user-word matrices, they employed the classical NMF, which provides a parts-based representation of content clusters. To better capture local graph structure, they extended this to Graph-regularized NMF (GNMF), which introduces a Laplacian regularizer so that connected nodes are encouraged to have similar embeddings. They further developed Nonnegative Matrix Tri-Factorization (NMTF), which factorizes user-word matrices into three factors to explicitly model user clusters, word groups, and their interactions. To combine multiple views, they proposed TriNMF, which jointly factorizes two matrices such as user-word with user-hashtag or user-domain, thereby exploiting complementary information. MultiNMF generalizes this idea to multiple content matrices simultaneously, sharing user factors across them. Finally, DualNMF couples graph structure and content through co-regularization, aligning the two spaces, and typically yielding the strongest performance between their models. For connectivity-only baselines, [58] used Louvain and CNM, both modularity-maximizing algorithms applied to endorsement-filtered retweet and mention graphs. In contrast to these factorization-based approaches, the matrix constructed in information access clustering represents diffusion reachability rather than content or co-occurrence, giving similarity a direct process-based interpretation.

Multi-View Graph Embedding Using Randomized Shortest Paths. In contrast to matrix factorization, [23] proposed C-RSP, which directly embeds nodes from multiple layers into a unified space. By averaging randomized-shortest-path dissimilarities across layers, C-RSP preserves multi-layer structural patterns while allowing clustering in the embedded space. This approach emphasizes the value of path-based similarity metrics in multi-layer settings. Although C-RSP relies on shortest-path-based dissimilarities, information access clustering uses diffusion-based reachability, capturing influence under stochastic activation rather than optimal paths.

Null Models and Community Detection in Multi-Layer Networks. Paul and Chen in [41] researched clustering from the point of multi-layer modularity optimization under various null models. They compared single-layer baselines with a suite of multi-layer extensions. The MNavrg method applies an independent-degree null model averaged across layers, while SDarvg, SDlocal, and SDratio employ shared-degree assumptions with different normalization strategies to balance contributions of layers. Finally, an Aggregate baseline simply collapses all three layers into a single weighted graph. These approaches extend modularity-based clustering to multi-layer networks, whereas this thesis extends diffusion-based information access.

3 Problem Definition

A social network can be described with a directed weighted graph $G = (V, E, w)$, where V is a set of nodes that represents network members, E is a set of edges that represent connections between nodes, and w is the weight function $w : V \times V \rightarrow [0, 1]$, which assigns weight from the given interval to each edge in the network. Let us denote a weight between the nodes u and v as $w_{u,v}$.

For a long time, graph-like data were represented only using a simple graph structure. This type of network is called a *single-layer* or *monoplex* network and is used to describe a static network with one type of connection between members.

The problem occurs when there is a need to make a graph representation of a network that has multiple types of connection within the same group. For example, in a company everybody is a colleague, but some of them are also friends or even family members. This type of scenario can be described using a single-layer network with different edge weights for each connection type, but this approach would not give a full semantical understanding of the network structure, not to mention limitations regarding assigning edge weights inside the same link group.

In order to capture real-world data in a more precise way, a new network type was introduced called *multi-layer* or *multiplex* network. We will formally define them as they were defined in [17]. There are also more broad and general definitions, such as [7], but for the purpose of this research, we need a simpler network, without inter-layer connections.

Let the multi-layer network $\mathcal{G}$ consist of $m \in \mathbb{N}$ directed weighted graphs $G_1, ..., G_m$. Each graph $G_k = (V, E^k, w^k)$, $k = 1, ..., m$, has the same set of nodes (V), a different set of edges ($E^k \subseteq V \times V$), and a different weight function $w^k : V \times V \rightarrow [0, 1]$. This graph is called a *layer* of a multi-layer network. A pair of nodes $(u, v)^k$ denotes an edge from u to v on a layer k with assigned weight $w^k_{u,v} \in [0, 1]$. The node v has an *in-neighbor* u in the layer k if $(u, v)^k$ exists. We say that for v, the weight of its in-neighbor u in layer k is equal to the weight on the edge connecting them, i.e., $w^k_{u,v}$.

This chapter gives a comprehensive overview the theoretical foundation of the information access clustering; the Linear Threshold Model definitions for both single- and multi-layer networks, algorithm for the information access clustering, description of datasets, and algorithm definitions.

3.1 Information Diffusion

Information diffusion is a process in the network in which some piece of information flows from an initial set of nodes (also known as a seed) to other nodes. That information is anything that can be accepted or adopted by nodes, from actual information about

something, opinion, gossip, strategy, etc. There are two states in which each node can be: if a node is spreading information, it is called *active*, otherwise it is *inactive*. In the majority of models, nodes can only go from inactive state to active, not the other way around. The exception to this rule is epidemic models that model the diffusion of diseases, for which there may be a recovery possibility (such as the SIR model [31]). Two main models explaining this phenomenon are the Independent Cascade Model (ICM) and the Linear Threshold Model (LTM). The focus of this thesis will be on the latter, but both models are mathematically defined and analyzed in [30].

3.1.1 Linear Threshold Model for Single-Layer Networks

The first definition of LTM was given by Granovetter [25] back in 1978 from the perspective of social studies. The underlying idea of this initial model was that each node has its own property called *threshold*, which is a number from the interval [0,1]. This number defines the ratio of neighbors for a certain node that has to be activated in order to activate that node. The threshold can also be interpreted as a minimal percentage of neighbors that have to be active in order to activate the node holding that threshold.

A more formal and detailed definition of LTM was provided in [30], which remains a widely accepted definition for this model. Hence, we are going to formally define LTM here on the basis of the theoretical background of that paper, as well as its shorter version in [17].

Following the network representation from the previous chapter, the social network is a directed weighted graph $G = (V, E, w)$. Furthermore, the weights sum of the incoming edges of a node must not exceed 1, i.e.

$$\sum_{(u,v)\in E} w_{u,v} \leq 1, \forall v \in V.$$

Each node v in the network has an independently assigned threshold $\theta_v \in [0, 1]$, which is a core concept of this model.

The diffusion process unfolds in discrete sequential steps. Let S_t (for $t > 1$) denote a set of active nodes at step t, while the set of seeds is S_0. The process ends at the first step t where $S_t = S_{t+1}$; it is the step in which any further iteration over the graph would result without any new node being activated. The node v is activated in step t if the sum of the weights of its previously activated in-neighbors exceeds θ_v, that is,

$$\sum_{(u,v)\in E \,\wedge\, u\in S_{t-1}} \geq \theta_v.$$

Although the process in these discrete steps is theoretically valuable, it is not computationally efficient. For that reason, [30] proposed another computation method and proved the equivalency between the two of them. The authors of that paper demonstrated that the set of active nodes in the final stage of the process, starting with the seed S_0, is equivalent to the distribution of nodes reachable from the same seed set S_0 in a random graph known as a *live-edge graph*. So, the problem of finding a final set of active nodes S_t

is a *reachability* problem for the live-edge graph, provided both methods have the same seed.

The question that now has to be answered is how to construct such a random graph. Let $G = (V, E, w)$ be the graph for which we are constructing live-edge graph $G_{le} = (V, E_{le})$. For each node $v \in V \setminus S_0$, zero or one of its incoming edges is selected independently with probability equal to its weight, i.e. edge $(u, v) \in E$ is selected and added to E_{le} with probability w_{uv}. Since the sum of the incoming weights of v can be less than 1, there is the possibility of not picking any of the incoming edges. Hence, no edge is selected with probability $1 - \sum_{(u,v)\in E} w_{u,v}$, which would lead to node v in G_{le} having in-degree equal to 0 and being unreachable regardless of S_0.

A live-edge graph G_{le} is undirected, so if there is a path between any $v \in S_0$ and $u \in V$, we say that u is *reachable* in that graph. The path from the seed node to u is called *live-edge path.*

3.1.2 Linear Threshold Model for Multi-Layer Networks

Let us extend the Linear Threshold Model to multi-layer networks, following the definitions and methods given in [17].

In order to have a network compatible with LTM, one more constraint has to be added to the previous definition of a multi-layer network. For every node v in the multi-layer network $\mathcal{G}$ with k layers, the sum of the weight of its in-neighbors in one layer should not exceed 1, i.e.

$$\sum_{(u,v)\in E^k} w^k_{u,v} \leq 1.$$

Furthermore, let us extend the definition of threshold: for every node v, there is an independently assigned threshold on layer k, that is, $\theta^k_v \in [0, 1]$.

Some process steps are the same as for single-layer networks, such as having seed set S_0 in the beginning and running iterations for the t steps until convergence $S_t = S_{t+1}$. However, there is a difference in the definition of the moment when a certain node is activated. When for a node v the sum of the weights of active in-neighbors exceeds its threshold in the layer k, we say that node v received a *positive input* from the layer k, that is,

$$\sum_{(u,v)\in E^k \,\wedge\, u\in S_{t-1}} \geq \theta^k_v.$$

We can analogously define a *negative input* as a signal that indicates that the threshold is not exceeded on the layer k.

From the network structure, we can see that in every step $t > 0$ there will be m conflicting inputs for every node in the network. Hence, we have to introduce a *protocol function* $R : \{0, 1\}^m \to \{0, 1\}$, which takes a vector of length m containing inputs from all layers for one node and returns *activation decision* 0 or 1.

The activation decision is to be made based on the inter-layer threshold parameter $\theta_{inter} \in [0, 1]$, which indicates the minimal ratio of positive inputs that must be sent to

a node $v \in V$ in a step $t-1$ to activate that node in step t. If $\mathbf{x} = (x_1, x_2, ..., x_m)$ is a vector of inputs in step t, then the protocol function is defined as

$$R(\mathbf{x}) = \begin{cases} 1 & \text{if } \frac{1}{m}\sum_{i=1}^{m} x_i > \theta_{inter} \\ 0 & \text{otherwise.} \end{cases}$$

There is a specialization of this approach for two-layer (duplex) networks, provided by [17]. For the two incoming inputs in each node (x_1 and x_2), the activation decision can be $x_1 \wedge x_2$ (protocol *AND*) or $x_1 \vee x_2$ (protocol *OR*). Networks with nodes that are more conservative in making decisions to be active are modeled with the protocol AND, which corresponds to having $\theta_{inter} \in [0.5, 1]$. Those more prone to becoming active would use protocol OR, corresponding to $\theta_{inter} \in [0, 0.5)$.

Similarly to the equivalency of single-layer LTM and live-edge graph, [17] introduced the concept of *live-edge tree* that is equivalent to duplex LTM. To construct a live-edge tree, the same method as before will be used: every node $v \in V \setminus S_0$ will pick at most one incoming edge in each layer with a probability proportional to its weight (w_{uv}^k). The live-edge tree associated with a node v satisfies the following:

> "Every node in the tree corresponds to a node in the duplex network $\mathcal{G}$. The root corresponds to agent v.
> For each parent p in the tree, p's left (respectively, right) child is the node to which p's live edge in layer 1 (respectively, 2) is connected."[17]

For both duplex protocols (OR; AND), there are reachability definitions that are the core of this equivalency.

> "Reachability OR: for a given selection of live edges and a set S_0, a node v is reachable from S_0 by a selection of live edges if the live edge tree associated with v has at least one finite branch, and every finite branch ends with a seed.
>
> Reachability AND: for a given selection of live edges and a set S_0, a node v is reachable from S_0 by a selection of live edges if all branches of the live-edge tree associated with v are finite, and every branch ends with a seed."[17]

3.2 Information Access Clustering

The main objective of this section is to provide a quantitative view of information access for every node in a network. We will present here a formal definition as introduced in [4]. Let us have a directed weighted graph $G = (V, E, w)$ in which the diffusion of information is executed using a model M with a parameter vector α. We will use the abbreviation for the number of nodes $|V|$ as n. For each node $v_j \in V$ we can define an *information access signature* $s_{M(\alpha)}^G : V \to \mathbb{R}^n$ that is,

$$s_{M(\alpha)}^G(v_j) = (p_{1j}, ..., p_{ij}, ..., p_{nj}),$$

where p_{ij} is the probability that the parameter node v_j receives information from the node $v_i \in V$. Less formally, we can say that the information access signature is a property vector of size n associated with every node in the network explaining the probabilities that a node holding that vector receives information from any other node in the network. This vector also represents a point in the n-dimensional vector space: the closer two points in that space are, the more similar the corresponding nodes are regarding information access. The numerical similarity of the two signatures will be the key factor in deciding which nodes have comparable information privileges.

To have a comprehensive measurement for the entire graph G, let us introduce an *information access representation*

$$IA^G_{M(\alpha)} = \{s^G_{M(\alpha)}(v_j) \mid v_j \in V\},$$

where M again denotes the diffusion model with the parameter α. We can see that the information access representation is an $n \times n$ matrix containing all information access vectors from graph G as rows. It is important to note here that this is a matrix in which all diagonal elements are equal to 1.

Computing this matrix is not an easy task since the probabilities depend on the choice of a model and its parameters. The model chosen for information diffusion in this paper is previously mentioned LTM, so we will denote the corresponding information access representation as $IA^G_{LTM(\alpha)}$, or simply $IA(G)$. Beilinson et al. [4] computed IA based on simulations of the Independent Cascade Model; we will follow that approach, apply it to LTM, and extend it to multi-layer networks.

To compute IA, we have to run N rounds of LTM simulations on a graph G; one round means running $|V|$ simulations, each time having a different node $v \in V$ alone as a seed. Thus, we can capture a set of activated nodes S_t from the seed set $S_0 = \{v\}$. Calculating the values in IA consists of simply counting the number of times the nodes activated each other and dividing by a number of simulations.

Running LTM can be achieved in a couple of different ways. Since there is a proof of equivalency between the formal definition of LTM and reachability via live-edge graphs, those two options can be used for simulations. For running using definition, we use the library called NDlib[1], which offers implementation of LTM simulations, but only for unweighted graphs. There is no such implementation for multi-layer graphs. For them, only the live-edge model will be utilized.

Performance may be an issue for larger datasets if simulations are run via definition; in that case, we use parallelization.

Clustering as a final step of this algorithm is to be conducted on the IA. Each row represents a point in the n-dimensional space for which the L^2 norm will be used to find distances in the k-means clustering algorithm.

[1] see `https://ndlib.readthedocs.io/en/latest/reference/models/epidemics/Threshold.html`

3.3 Datasets

This section introduces the datasets on which information access clustering is performed. General statistics is provided in the tables, and detailed explanation about each dataset is given in the following subsections.

Tables 3.1 and 3.2 provide information on datasets for single- and multi-layer networks, respectively. The former table introduces the concept of *external information access measure*, which is "an external node attribute that can be used to quantify information access or social capital within the network" [4]. The purpose of this attribute is to provide insight about node's information access from an external source, meaning that this finding did not come from the network itself. For example, whether a user is a partner or not in the Twitch dataset or an author's h-index in the Google Scholar dataset is not derived from the graph topology, but is immanent to a node regardless of the network.

Dataset	Nodes	Edges	Density	External Inf. Access Measures
Co-sponsorship	438	28,194	0.1473	legislative effectiveness score
Twitch	7,126	35,234	0.0014	partner, views
Google Scholar	14,871	52,768	0.0005	citation count, h-index

Table 3.1: Dataset Information for Single-Layer Networks

Dataset	Layer	Nodes	Edges	Density
Flickr	friendship	10,364	390,938	0.0036
	tag-similarity	10,364	112,966	0.0011
UK Politics	follows	374	25,830	138.1
	mentions	374	14,411	77.1
	retweets	374	7,219	38.6

Table 3.2: Dataset Information for Multi-Layer Networks

3.3.1 Congressional Co-sponsorship

The Co-sponsorship is a dataset created by GovTrack and made available by Fowler, Waugh and Sohn in 2017[2]. It is a network consisting of the members of the 114th sitting of the United States Congress. Each node represents either a sponsor or original co-sponsor of a bill, and there is a directed edge between nodes v and u if node u originally co-sponsored one or more bills that were sponsored by the node v.

From the original dataset, the largest strongly connected component was extracted and used in this research. The same dataset was used in [4]: dataset creation code and input files were reused from the paper repository[3].

[2] see `http://jhfowler.ucsd.edu/cosponsorship.htm`

[3] see `https://github.com/algofairness/info-access-clusters/releases/tag/paper.1.0`

Among node attributes in the original dataset there was no such attribute that could be used as external information access measure, so a new value called *legislative effectiveness score* (*le_ score*) was computed by Volden and Wiseman in 2020[4]. The score is a numerical measurement of how influential a certain congressman is regarding his ability to influence the legislative process by proposing a bill and advancing it into a law through parliamentarian procedure. It is calculated based on 15 weighted indicators [18], with a mean value of 1.0021 and a median 0.6678. The minimal score in the dataset is 0.0027, while the maximum is 7.6571.

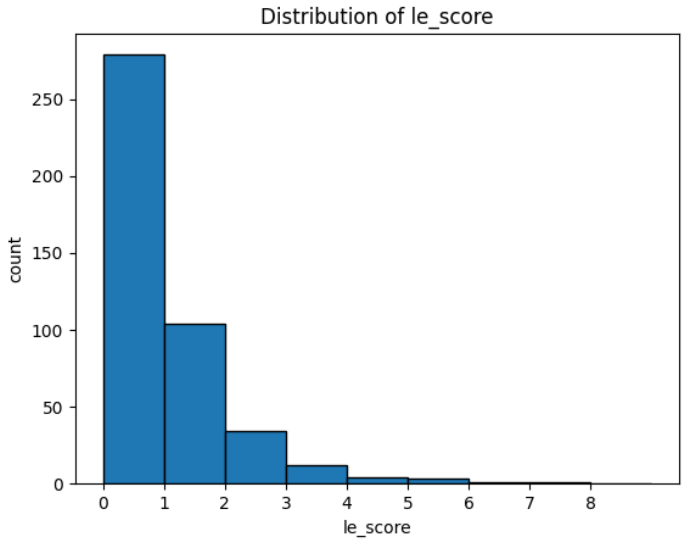


Figure 3.1: Distribution of the *le_ score* attribute in the Co-sponsorship dataset

Another significant attribute for the evaluation of this method is *party affiliation*, which indicates whether a certain congressman is a member of the Democratic Party or not. Although this value cannot be seen as an external information access measure, it can be useful to see if the clustering method recognizes that members of one party are more likely to receive information between each other than from their political opponents, which is an intuitive assumption. This sitting of the U.S. Congress had 44% of democratic members.

3.3.2 Twitch

This dataset is a social network created using the Twitch[5] data, a popular video live-streaming service used mostly by gamers to broadcast themselves playing video games. It was collected in May 2018 by [46], and contains 7,126 nodes representing gamers that stream in the English language. The graph has 35,324 undirected edges that represent

[4]see `https://thelawmakers.org/data-download`
[5]see `https://www.twitch.tv`

mutual friendships between gamers. It is the largest connected component of the original dataset; the pipelines used to create the graph were used again from [4] and repository[3].

In this network, node attributes are also present. For this research, interesting ones are *partner* and *views*. The attribute *partner* is a boolean: one can become a partner if certain criteria are met, e.g., the audience of respectable size on various online platforms, number of hours streaming, number viewers on Twitch, etc. Only 5.39% of users in the dataset are partners. *Views* is a numerical attribute that indicates the total number of views a user has at the time of the dataset creation. This attribute is highly dispersed, with a minimal value of 5, mean 193,470, and maximal 178,500,544. With that in mind, this property is analyzed after applying the log_{10} function.

Since these two attributes are independent of a network topology and give information about how influential a user is, they will be used as external information access measures for this dataset.

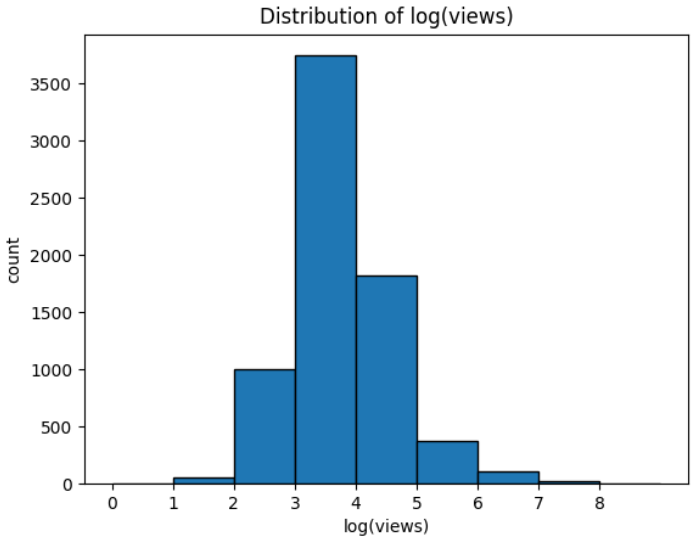


Figure 3.2: Distribution of the *views* attribute in the Twitch dataset, normalized by applying logarithm

3.3.3 Google Scholar Co-authorship

Attribute	Min	Mean	Max
Citation count	0	2,382	181,579
h-index	0	16	126

Table 3.3: Node attributes statistics in the Google Scholar dataset

The original Google Scholar network was created as an output of scientific research [12]

resulting in about 402 thousand nodes and 1.23 million edges; nodes represent authors on Google Scholar, while edges are co-authorship relation for at least one paper. The dataset is fetched from the repository[6], processed and filtered so that the final network used in this research contains 14,871 nodes with 52,768 edges in the largest connected component. The original network was filtered by two criteria: only computer science authors are included, and authors with an unknown academic title were excluded from the sample. Filtering of the data is done purely from the performance perspective; for much larger datasets performance significantly drops, which will be a topic in the scalability and performance analysis in the following chapters.

Significant node attributes are the author's *citation count* and *h-index*, as external measures of importance. The former is a total number of citations received by a given author, while the latter is a measure[7] introduced by Hirsch in 2005.

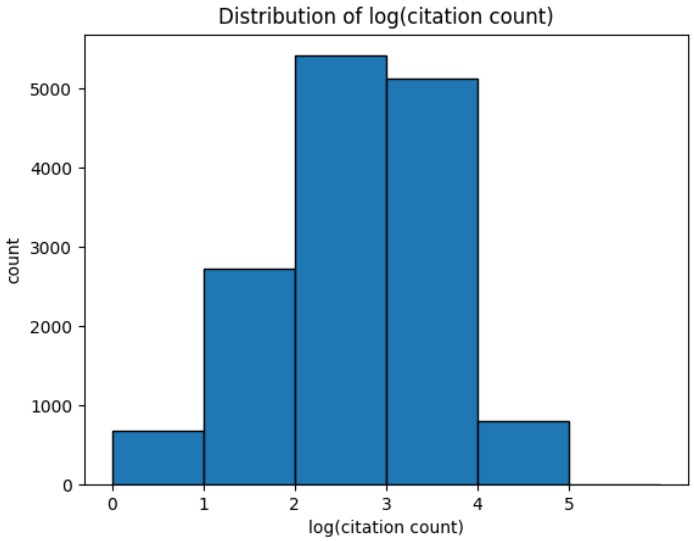


Figure 3.3: Distribution of the *citation count* attribute in the Google Scholar dataset, normalized by applying logarithm

3.3.4 Flickr

Flickr[8] is an online platform for hosting and sharing photos and videos, widely used by photographers, artists, and enthusiasts. Users can upload high-quality images, organize them into albums, and engage with others through comments and groups. The platform also supports tagging and offers access to a large collection of visual content.

[6] see `https://github.com/chenyang03/co-authorship-network`

[7] **h-index** is defined as a maximum value of h such that the given author has published at least h papers that have each been cited at least h times [29].

[8] see `www.flickr.com`

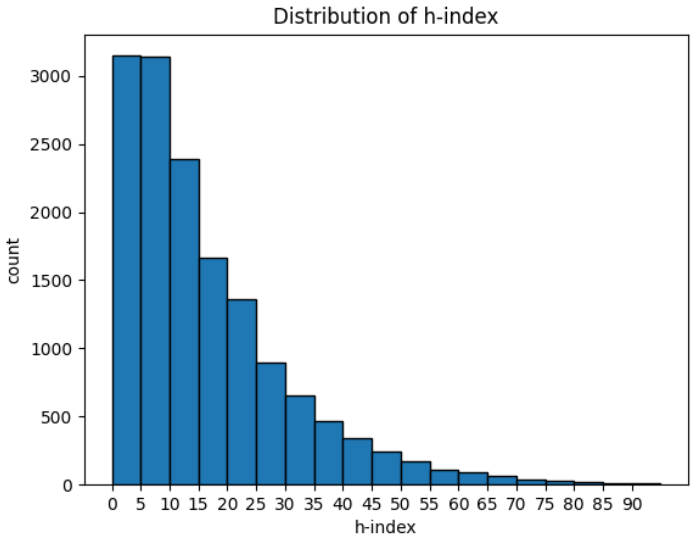


Figure 3.4: Distribution of the *h-index* attribute in the Google Scholar dataset

This dataset[9], created in [52], is a multi-layer graph with two layers; there is the same set of nodes on both layers, while the edge sets are different. The first layer links users based on friendship on this social network, the other one connects them based on their tag-similarity. The nodes in this graph are without attributes, but there is a ground truth: each node belongs to one of the seven social groups (clusters), which are rather similar in size, as indicated in Table 3.4.

Cluster	**0**	**1**	**2**	**3**	**4**	**5**	**6**
Size	1207	1366	1627	1760	1480	1214	1710
Percentage	11.65	13.18	15.70	16.98	14.28	11.71	16.50

Table 3.4: Size of each cluster in the Flickr dataset

3.3.5 UK Politics

The initial UK Politics dataset was introduced in [26] together with a couple of other multi-layer networks suitable for community detection. This dataset captures Twitter interactions between British Members of Parliament (MPs) on three layers: *mentions*, *follows*, and *retweets*. Each node is labeled according to the party affiliation: *Labour*, *Conservative*, *Liberal Democrat*, and *SNP*.

In order to obtain a network that is suitable for information access clustering, we used the version of this dataset from [41], which filters it so that the graph is connected on

[9] see `http://mlg.ucd.ie/networks/politics-uk.html`

each layer. We try to detect the mentioned groups, as shown in Figure 3.5.

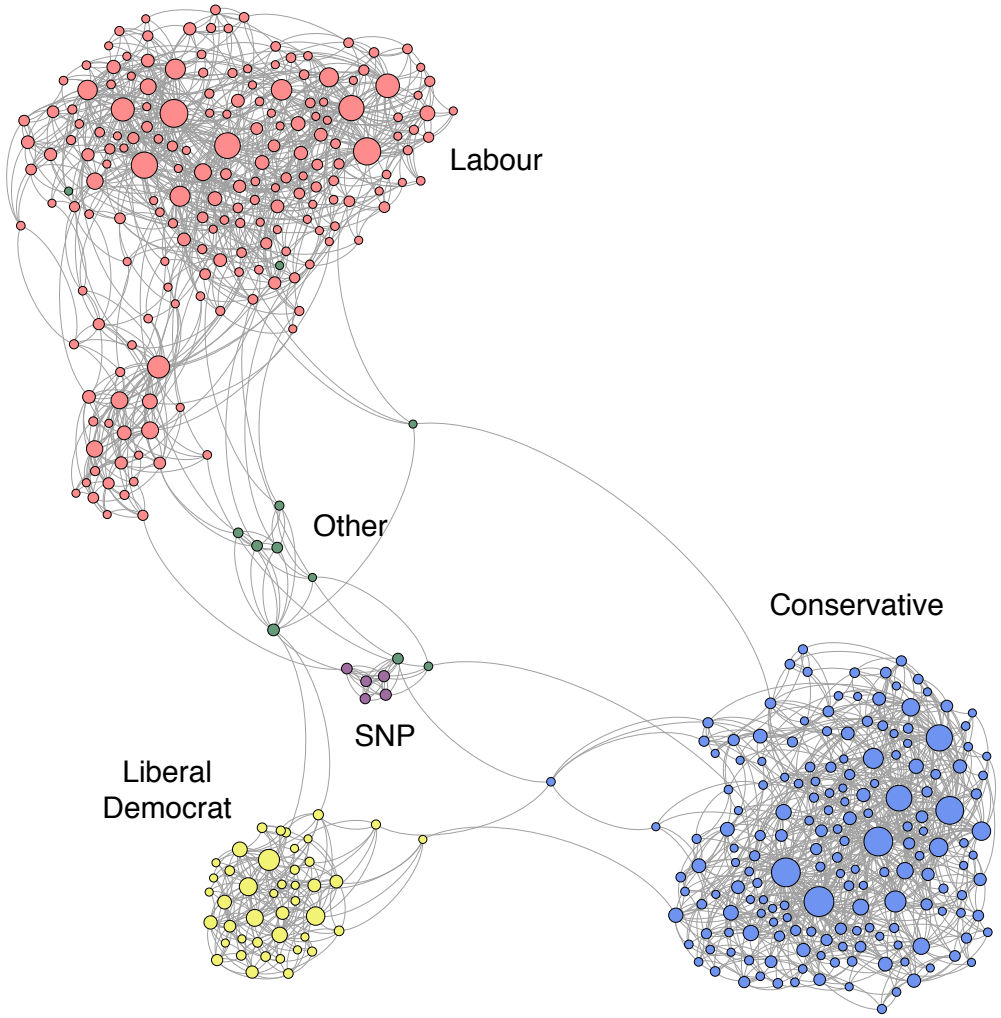


Figure 3.5: UK Politics unified graph [26]

3.4 Assigning Edge Weights

As previously introduced, the Linear Threshold Model takes a weighted directed graph as an input, while all presented graphs are not weighted. Hence, we define the weight function $w_{u,v}$ that assigns weight to each edge in the graph. We will define different sets of edge weights and compare the results obtained using each of them. For that purpose, there are 4 methods (edge-weight assigning strategies) used in this research to convert an unweighted graph to weighted one, and each of them defines mapping function w.

- **Weighted** method is based on a incoming degree of nodes in the network, i.e.

	Weighted	Random	Uniform	Trivalency
Co-sponsorship	0.0155	0.0154	0.0063	0.0145
Twitch	0.1751	0.1363	0.0027	0.0317
Google Scholar	0.2299	0.1698	0.0082	0.0357

Table 3.5: Average edge weights for datasets under each method of weight assignment

$w_{u,v} = \frac{1}{\text{in-degree}(v)}$.

- **Random** method assigns each edge a random value from the interval $[0, 1]$
- In the **uniform** method, all edges in the network receive the same predefined weight α.
- **Trivalency** method selects randomly with equal probability edge-weight of the values $\{0.1, 0.01, 0.001\}$.

All these methods except *uniform* were used in [28] on the Independent Cascade Model, where they showed satisfactory performance.

Since sum of each node's incoming edges must not exceed 1 according to Linear Threshold Model definition, assigning edge-weights has to be done in iterations until validation against this rule passes.

Other important note to mention is the fact that only Co-sponsorship dataset is directed, Twitch and Google Scholar are not. At the very beginning of this pipeline, they are turned into directed by assigning direction to each edge randomly with the equal probabilities.

3.5 Defining the Number of Clusters

Silhouette scores									
Dataset	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Co-sponsorship	0.290	0.196	0.164	0.142	0.036	0.036	0.048	0.024	0.025
Twitch	0.099	0.086	0.080	0.070	0.068	0.071	0.060	0.072	0.081
Google Scholar	0.071	0.071	0.071	0.071	0.071	0.071	0.072	0.072	0.072

Table 3.6: Silhouette scores of datasets with *weighted* edge-weights for k=2 to k=10

The process starts with determining the number of clusters (k), which will be used as a parameter in the clustering algorithm. Since multi-layer networks have ground truth, they have this parameter predefined: $k(flickr) = 7$ and $k(uk\_politics) = 4$.

The number of clusters for single-layer networks is determined using *silhouette values*[10]

[10] **Silhouette values** measure how similar an object is to its cluster compared to other clusters. The values are from the interval $[0, 1]$, with higher values indicating better-defined clusters. These values are then used to determine an optimal number of clusters k [45].

and *elbow method*[11]. To keep the computation simple and clear, the graphs used as input for these methods have been assigned edge weights according to the *weighted* method. This type was chosen because it is a standard method for LTM simulation and also an underlying type in the NDLib algorithm, which makes it easier to compare later.

The silhouette scores on the Table 3.6 and elbow plots Figure 3.6 and Figure 3.7 indicate that for Co-sponsorship and Twitch datasets k=2, which is consistent with the findings in [4]. It also corresponds to the number of values of categorical node attributes described in Table 3.1. For Co-sponsorship, nodes can have *democrat* or *non_democrat* values, while Twitch nodes can be *partner* or *non_partner*. Continuous attributes, such as *le_score* and *views*, will be compared using their distributions in different clusters.

However, the value of k in the Google Scholar dataset is not as evident as in other datasets. The silhouette scores for different values of k are uniform, while Figure 3.8 indicates that the appropriate values for k are 2, 5, and 7. The node attributes in this dataset are continuous, so they do not contain additional information about the best suitable k. Since this dataset is only a subset of the Google Scholar dataset used in [4], where k=2 is found, we will choose this value.

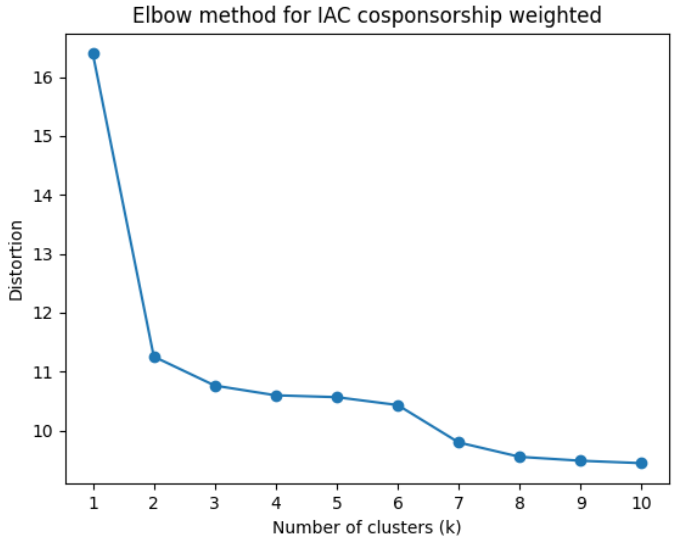


Figure 3.6: Elbow method chart for the Co-sponsorship dataset (weighted)

[11]The **Elbow method** is a technique used to determine an optimal number of clusters by plotting the sum of squared distances from each object to its assigned cluster center; the point where the sum starts to decrease slowly is an optimal value called an *elbow* [5].

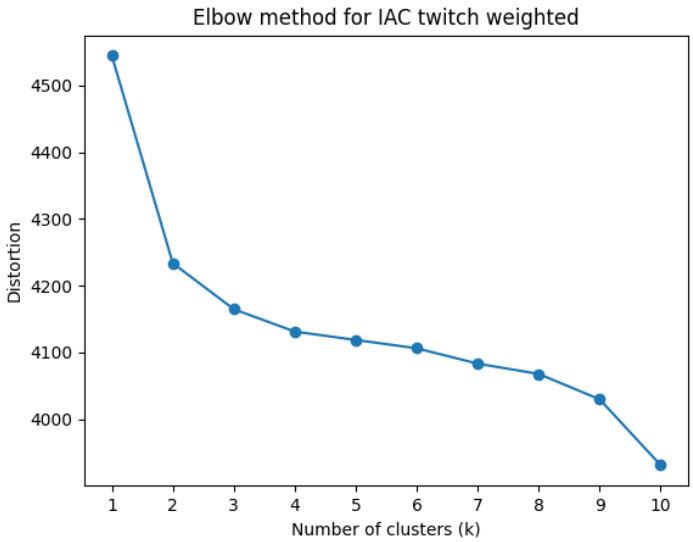


Figure 3.7: Elbow method chart for the Twitch dataset (weighted)

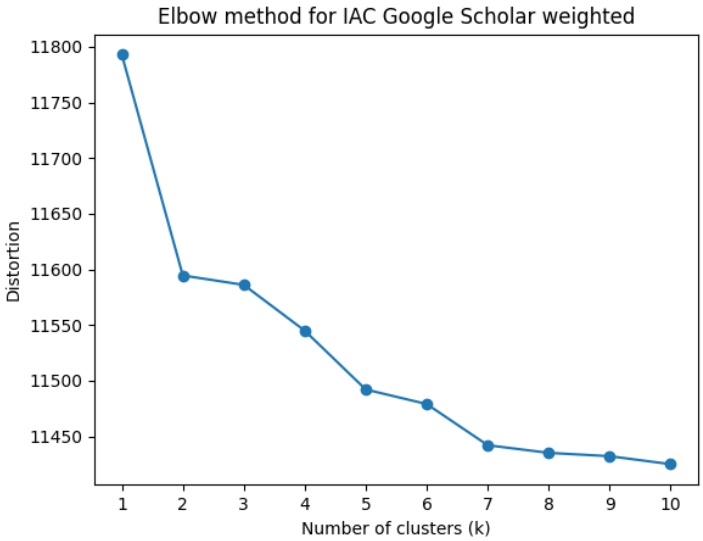


Figure 3.8: Elbow method chart for the Google Scholar dataset (weighted)

3.6 Simulations and Clustering

After we determined the number of clusters (k), the other parameter for information access clustering is still missing: information access representation of the graph G, i.e., matrix $IA(G)$.

There are several ways to obtain this matrix. In the case of a single-layer network, it can be computed either by directly applying the formal definition of the Linear Threshold Model or through LTM simulations using reachability in live-edge graphs. The main steps of those two methods are outlined in Algorithm 1 and Algorithm 2, respectively. The number of simulations is set to 10,000 for every graph, the same as in [4] and [28].

Algorithm 1: Information access computation via LTM definition

Input: Directed unweighted graph $G = (V, E)$, number of simulations S;
Output: Information access representation IA;
Let $n \leftarrow |V|$;
Initialize matrix $IA \leftarrow \mathbf{0}_{n \times n}$;
for $i \leftarrow 1$ **to** S **do**
  Assign each node $v \in V$ a random threshold $\theta_v \sim U(0, 1)$;
  foreach *seed node* $s \in V$ **do**
    Initialize all nodes as inactive;
    Activate seed node s;
    repeat
      Perform one Linear Threshold diffusion step;
    until *no node changes its activation state*;
    Let A be the set of activated nodes after convergence;
    foreach *node* $v \in A$ **do**
      $IA[v, s] \leftarrow IA[v, s] + 1$;
Set $IA \leftarrow \frac{1}{S} \cdot IA$;
return IA;

A similar approach is taken for the multi-layer networks, but using only reachability via live-edge graphs. The key difference in this algorithm is the existence of the inter-layer threshold θ_{inter}, which determines the minimal proportion of the layers from which the node has to receive a positive input in order to be activated. The pseudocode for this method is given in Algorithm 3.

After all simulations are executed, the off-diagonal elements of the matrix were mostly between 0 and 0.1 (diagonal elements are always equal to 1). To avoid potential numerical problems, the matrix is normalized for k-means clustering, so the values are distributed in the interval [0,1]. This normalization performs a row-wise z-score standardization of off-diagonal elements. For each row, the diagonal entry is excluded, and the remaining $n-1$ values are centered by subtracting their mean and scaled by their standard deviation. This yields off-diagonal entries with zero mean and unit variance in each row, while

Algorithm 2: Information access computation via live-edge simulation on a single-layer graph

Input: Directed weighted graph $G = (V, E, w)$, number of simulations S;
Output: Information access representation IA;
Let $n \leftarrow |V|$;
Initialize matrix $IA \leftarrow \mathbf{0}_{n \times n}$;
for $i \leftarrow 1$ **to** S **do**
    Generate a live-edge graph G_{le};
    Initialize seed index $j \leftarrow 0$;
    foreach *seed node* $s \in V$ **do**
        Determine set A of nodes reachable from s in G_{le};
        foreach *node* $v \in A$ **do**
            $IA[v, j] \leftarrow IA[v, j] + 1$;
        $j \leftarrow j + 1$;
Set $IA \leftarrow \frac{1}{S} \cdot IA$;
return IA;

diagonal elements representing self-relations are excluded from the normalization and fixed to one.

Algorithm 3: Information access computation via live-edge simulation on a multi-layer graph

Input: Directed weighted k-layer graph $\mathcal{G} = G_1, ..., G_k$, number of simulations S, inter-layer threshold θ_{inter};
Output: Information access representation IA;
Let $n \leftarrow |V|$;
Initialize matrix $IA \leftarrow \mathbf{0}_{n \times n}$;
for $i \leftarrow 1$ **to** S **do**
  foreach *layer* $\ell = 1, \ldots, k$ **do**
    Generate live-edge graph $G_{le}^{(\ell)}$;
  Initialize seed index $j \leftarrow 0$;
  foreach *seed node* $s \in V$ **do**
    Initialize empty multiset $\{A^{(1)}, \ldots, A^{(k)}\}$;
    foreach *layer* $\ell = 1, \ldots, k$ **do**
      Determine set $A^{(\ell)}$ of nodes reachable from s in $G_{le}^{(\ell)}$;
    Initialize $A \leftarrow \emptyset$;
    foreach *node* $v \in \bigcup_{\ell=1}^{k} A^{(\ell)}$ **do**
      Let $c(v)$ be the number of layers ℓ such that $v \in A^{(\ell)}$;
      if $\frac{c(v)}{k} \geq \theta_{inter}$ **then**
        $A \leftarrow A \cup \{v\}$;
    foreach *node* $v \in A$ **do**
      $IA[v, j] \leftarrow IA[v, j] + 1$;
    $j \leftarrow j + 1$;
Set $IA \leftarrow \frac{1}{S} \cdot IA$;
return IA;

4 Results

This chapter presents the results of the information access clustering (IAC) applied to the previously described datasets. The experiments were conducted either using the Linear Threshold Model (LTM) as implemented in the NDLib library or by leveraging the equivalence between LTM and reachability through live-edge graphs. In the latter approach, one of four edge weight assignment strategies was used: weighted, random, uniform, or trivalency.

4.1 Results for Single-Layer Networks

In this section, we will present clustering results of single-layer networks (Co-sponsorship, Twitch, and Google Scholar) after 10,000 simulations, compare their results using different edge-weight assigning techniques, and contrast them to well-known clustering methods.

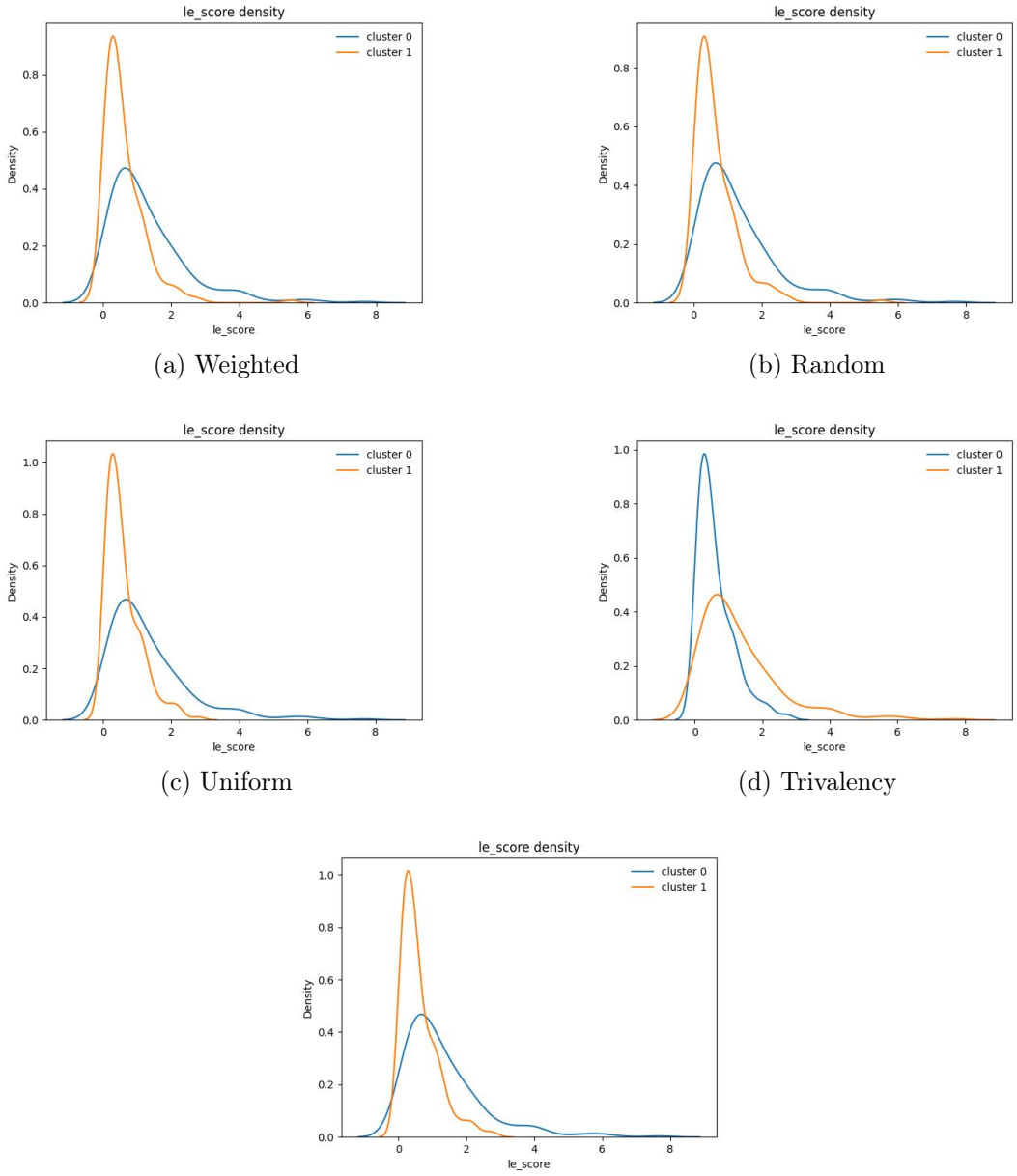
Regarding clustering efficiency, the objective is to investigate whether one cluster has higher information access (influence) than the other, quantified by an external information access measure, which is a specific type of node attribute. In case of the Co-sponsorship dataset, a materialization of social capital that U.S. congressmen and congresswomen in that group are holding is contained in le_score.

Clustering results for the Co-sponsorship dataset are consistently showing that one cluster holds a significantly higher le_score, with $mean(le\_score)$ being approximately 1.3 for the more influential group (irrespective of the edge-weights) and between 0.49 and 0.65 for the less influential one, as presented in Tables 4.1-4.5. For comparison, mean value of le_score throughout the entire dataset is 1. In the tables, democratic nodes are noted as D, and non-democratic as ND. The same notation is applied for partner and non-partner.

Cluster No.	**Nodes**	**#D**	**#ND**	**Mean *le_score***	**Median *le_score***
0	241	2	239	1.31	0.98
1	197	190	7	0.63	0.45

Table 4.1: Clusters for Co-sponsorship *weighted*

Visualization of this claim is on the Figure 4.1, showing that the distribution of more influential cluster is wider and right-skewed, with a longer tail extending toward higher values, while the less influential group has a narrower and more sharply peaked distribution, concentrated more towards 0 than the other one. This statement holds for all experiments, regardless of the edge-weight assignment strategy.



(a) Weighted

(b) Random

(c) Uniform

(d) Trivalency

(e) NDLib

Figure 4.1: Density of *le_score* across clusters in the Co-sponsorship dataset.

Cluster No.	Nodes	#D	#ND	Mean *le_ score*	Median *le_ score*
0	239	3	236	1.29	0.96
1	199	189	10	0.65	0.45

Table 4.2: Clusters for Co-sponsorship *random*

Cluster No.	Nodes	#D	#ND	Mean *le_ score*	Median *le_ score*
0	246	3	243	1.32	0.99
1	192	189	3	0.49	0.44

Table 4.3: Clusters for Co-sponsorship *uniform* with $w_{u,v} = 0.0063$

Regarding categorical values, the results show that a higher le_score is associated with having more non-democratic members of Congress in the cluster. Across all simulation types we can see that the more influential group is created predominantly of non-democratic members of parliament, while the democrats are contained in the other cluster, as visible on Figure 4.2.

In Twitch dataset, the node attributes (external information access measures) against which the evaluation is performed are called *views* (numerical) and *partner* (binary). The statistics of the Twitch *random* experiment described in Table 4.7 shows that the smaller cluster contains 76% of all nodes with attribute *partner*, and has 21x larger mean value of *views* than the other cluster. This findings correspond with the semantics of those two attributes: partners are the most influential users in the network, and grouping them in one cluster together with more influential non-partners, we obtain a group of users that has on average much more *views* than the other one.

While *random* performed outstanding, *uniform* and *trivalency* methods yield much poorer results. In Figure 4.3, we can see that these two methods show very little difference in *views* distribution between two clusters, indicating that one cluster does not have a significantly different level of information access compared to the other. The reason for this rather poor outcome is visible in their statistics; in Table 4.8 and Table 4.9 one cluster is more than 10 times larger than the other in both cases, which is caused by uneven activations through the network because of the respective edge-weights.

On the other hand, graphs for the other methods (*weighted*, *random*, and *NDLib*) show a significant distribution difference between the clusters for *views*, where the most influential group is wider and more shifted to the right.

The frequency of partner status in the resulting groups is shown in Figure 4.4, where it

Cluster No.	Nodes	#D	#ND	Mean *le_ score*	Median *le_ score*
0	201	188	13	0.62	0.45
1	237	4	233	1.33	0.99

Table 4.4: Clusters for Co-sponsorship *trivalency*

Cluster No.	**Nodes**	**#D**	**#ND**	**Mean *le_ score***	**Median *le_ score***
0	243	2	241	1.32	0.99
1	195	190	5	0.6	0.44

Table 4.5: Clusters for Co-sponsorship *NDLib*

Cluster No.	**Nodes**	**#P**	**#NP**	**Mean *views***	**Median *views***
0	2100	41	2059	18,519	2,849
1	5026	343	4683	266,570	6,294

Table 4.6: Clusters for Twitch *weighted*

Cluster No.	**Nodes**	**#P**	**#NP**	**Mean *views***	**Median *views***
0	3079	290	2789	420,864	8,138
1	4047	94	3953	20,467	3,505

Table 4.7: Clusters for Twitch *random*

Cluster No.	**Nodes**	**#P**	**#NP**	**Mean *views***	**Median *views***
0	6937	383	6554	197,385	4,937
1	189	1	188	49,791	1,981

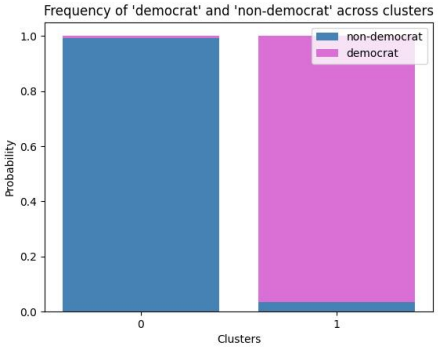
Table 4.8: Clusters for Twitch *trivalency*

Cluster No.	**Nodes**	**#P**	**#NP**	**Mean *views***	**Median *views***
0	455	15	440	31,445	4,794
1	6671	369	6302	204,521	4,845

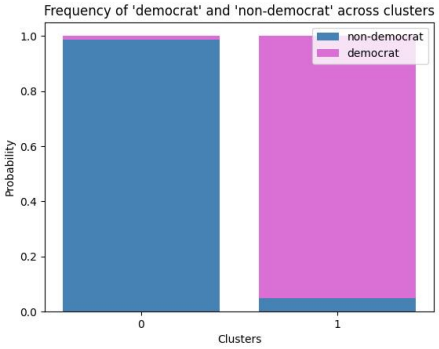
Table 4.9: Clusters for Twitch *uniform* with $w_{u,v} = 0.0027$

Cluster No.	**Nodes**	**#P**	**#NP**	**Mean *views***	**Median *views***
0	4847	338	4509	274,761	6,257
1	2279	46	2233	20,581	3,000

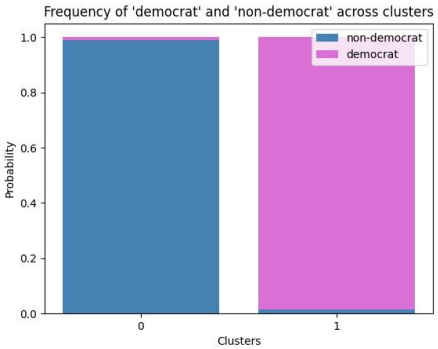
Table 4.10: Clusters for Twitch *NDLib*



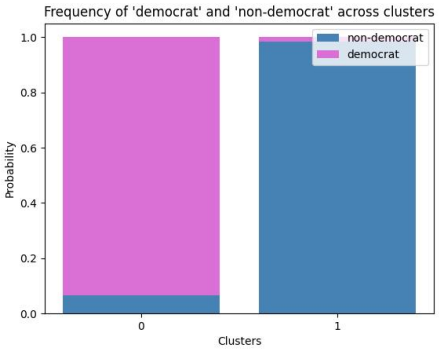
(a) Weighted



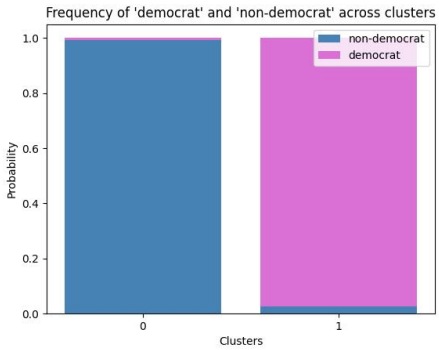
(b) Random



(c) Uniform



(d) Trivalency



(e) NDLib

Figure 4.2: Frequency of *democrat* and *non-democrat* across clusters in the Co-sponsorship dataset.

is visible that the more significant cluster holds more partners.

The Google Scholar dataset has no categorical or binary attributes, but rather two numerical external information access measures: *citation count* and *h-index*. Both of those attributes indicate how successful (consequently influential) a scientist is on Google Scholar. Similarly as above, our goal is to observe whether one cluster contains more influential scientific publishers than the other after conducting information access clustering based on simulations of LTM.

In this dataset all methods except *trivalency* show good performance. As depicted in Tables 4.11-4.14, one cluster has fewer nodes and significantly higher mean values for both external measures. This is an indicator that the clustering algorithm recognized the nodes with the highest social capital in the form of information access (academic impact) and grouped them together.

Cluster No.	**Nodes**	**Mean *citation count***	**Mean *h-index***
0	4,491	4,918	26.3
1	10,380	1,285	11.6

Table 4.11: Clusters for Google Scholar *weighted*

Cluster No.	**Nodes**	**Mean *citation count***	**Mean *h-index***
0	3,358	5,490	28.0
1	11,513	1,476	12.6

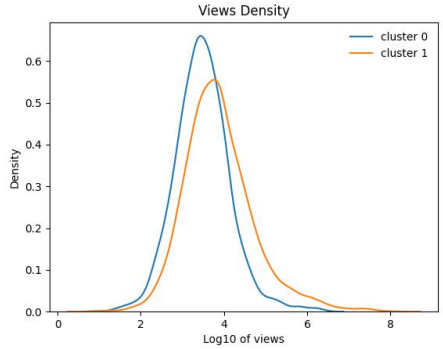
Table 4.12: Clusters for Google Scholar *random*

Cluster No.	**Nodes**	**Mean *citation count***	**Mean *h-index***
0	13,798	2,081	15.0
1	1,073	6,252	29.4

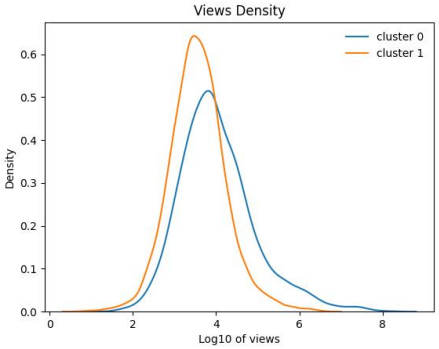
Table 4.13: Clusters for Google Scholar *uniform* with $w_{u,v} = 0.0082$

Visual proofs for these claims are given in Figure 4.5; for a *weighted* method, cluster 0 (blue line) in the KDE plot has a longer tail that extends beyond 100 and a peak around an h-index of 20. On the other hand, cluster 1, which represents a more concentrated group with generally lower h-index values and less variability, is sharply peaked around h=10 and rapidly declines. The results are similar for other methods except *trivalency*, which did not yield meaningful grouping because one cluster has only two nodes.

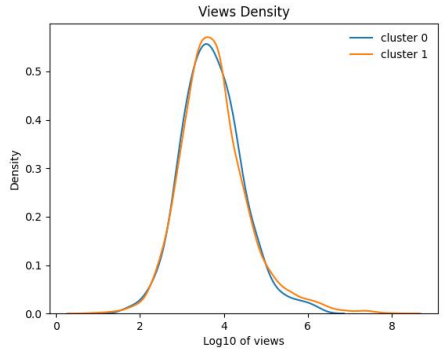
Although the *h-index* would be the best external information access measure in this case, similar results are obtained with *citation count*, as visible on the Figure 4.6.



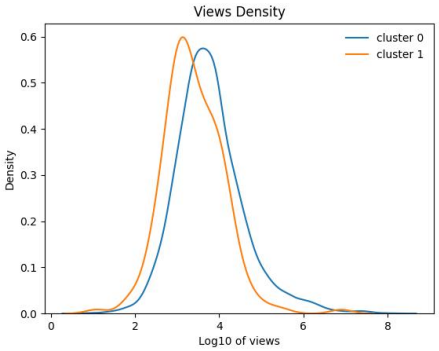
(a) Weighted



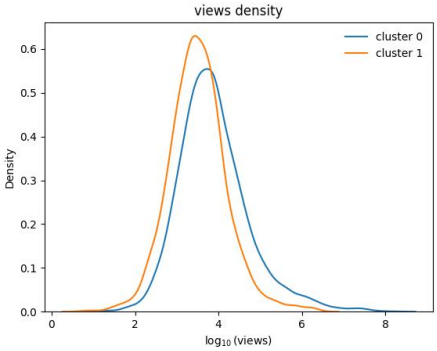
(b) Random



(c) Uniform

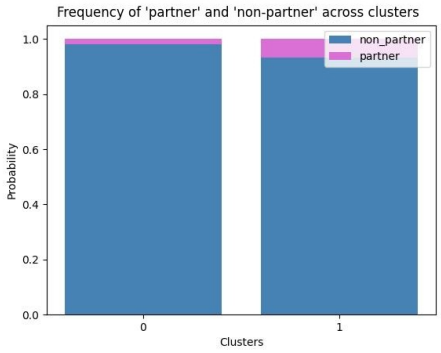


(d) Trivalency

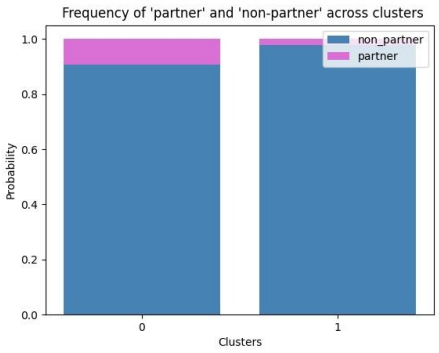


(e) NDLib

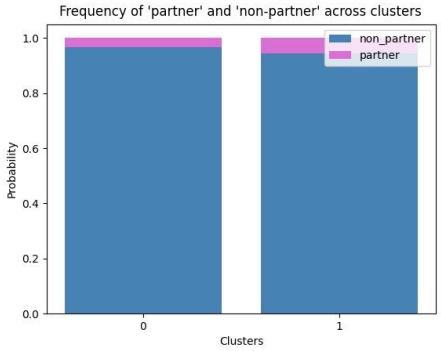
Figure 4.3: Density of *views* across clusters in the Twitch dataset, normalized by applying log_{10}.



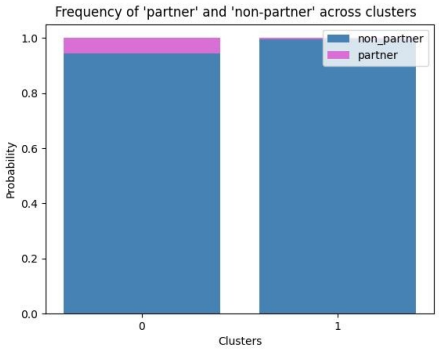
(a) Weighted



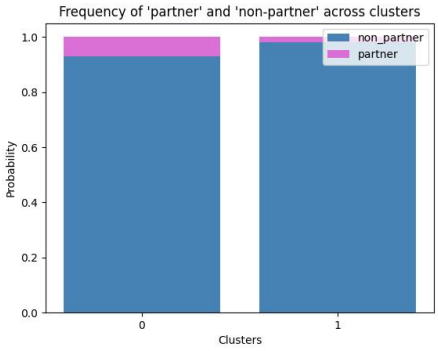
(b) Random



(c) Uniform



(d) Trivalency



(e) NDLib

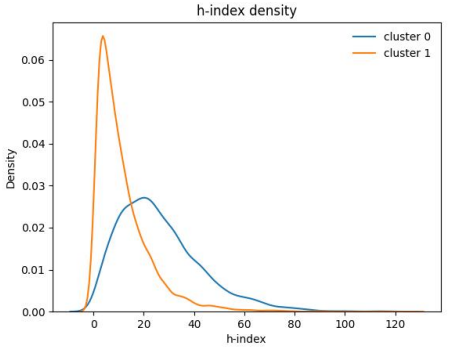
Figure 4.4: Frequency of *partner* and *non-partner* across clusters in the Twitch dataset.

Cluster No.	**Nodes**	**Mean *citation count***	**Mean *h-index***
0	5,006	4,253	23.5
1	9,865	1,433	12.3

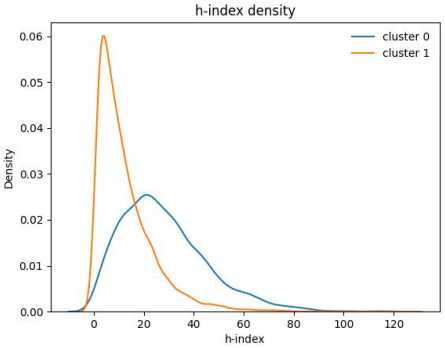
Table 4.14: Clusters for Google Scholar *NDLib*

Cluster No.	Nodes	Mean *citation count*	Mean *h-index*
0	2	784	13.0
1	14,869	2,382	16.1

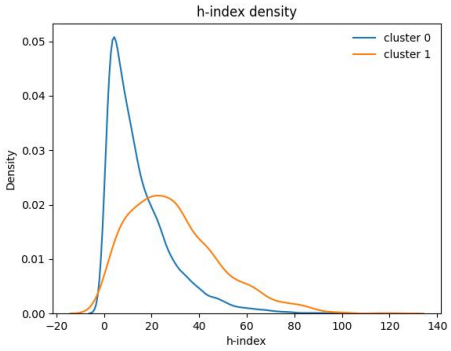
Table 4.15: Clusters for Google Scholar *trivalency*



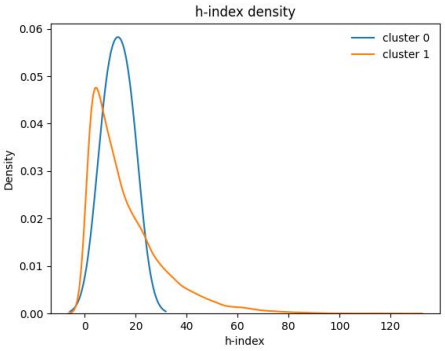
(a) Weighted



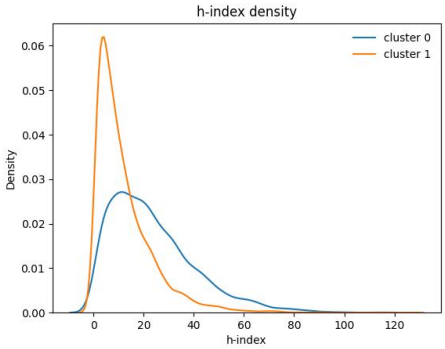
(b) Random



(c) Uniform

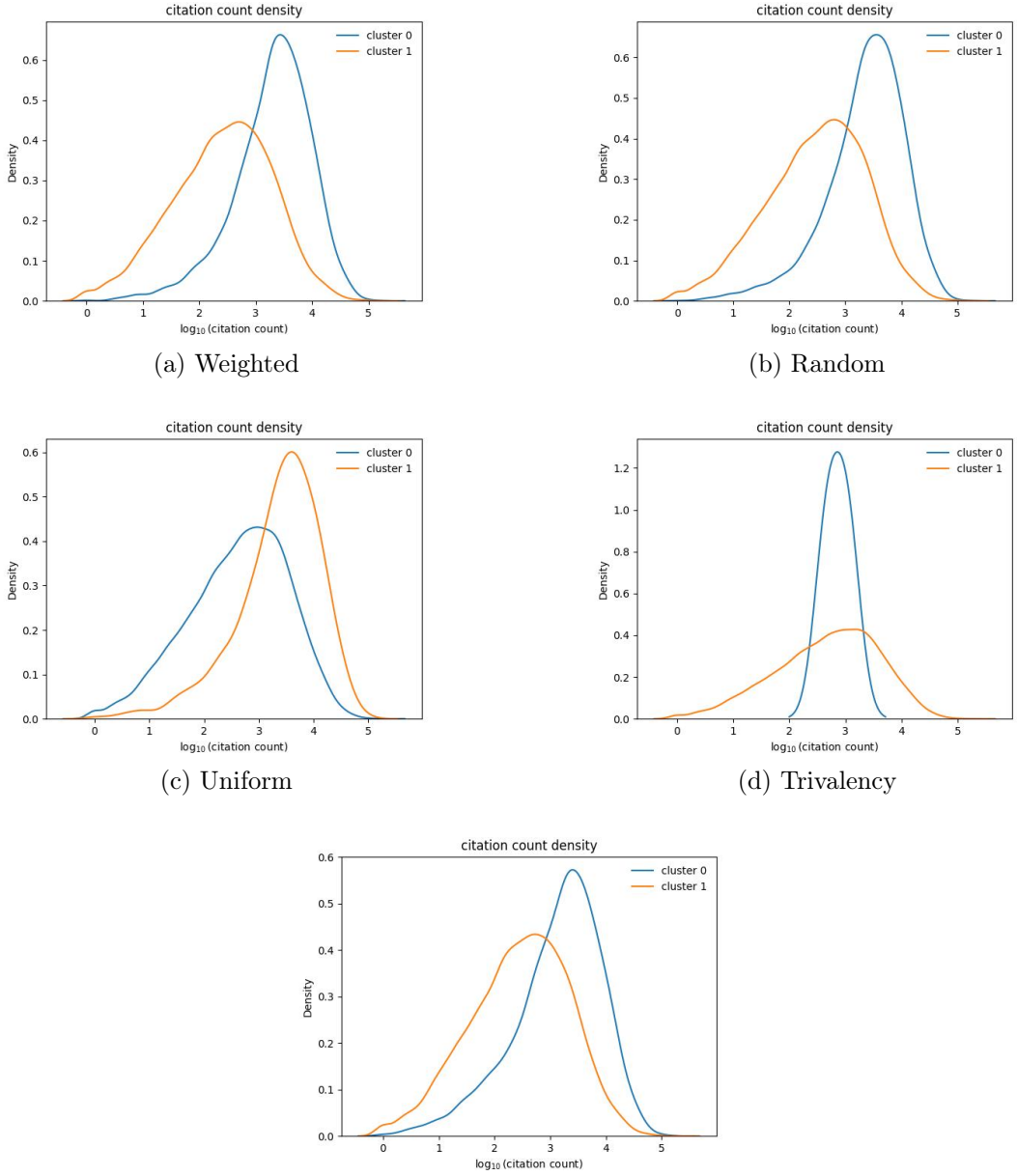


(d) Trivalency



(e) NDLib

Figure 4.5: Density of *h-index* across clusters in the Google Scholar dataset.



(a) Weighted

(b) Random

(c) Uniform

(d) Trivalency

(e) NDLib

Figure 4.6: Density of *citation_count* across clusters in the Google Scholar dataset, normalized by applying log_{10}.

4.1.1 Comparison Between Simulation Types

After analyzing the results as they are, we will compare them for each dataset with respect to different simulation types to observe similarities and differences between them. In order to achieve this, two cluster comparison measures will be utilized: Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI).

NMI[1] score is in range between 0 and 1, and indicates the amount of shared information between true and predicted labels. Score closer to 0 indicates a lower amount, while 1 indicates perfect matching between two sets of labels.

On the other hand, ARI[2] evaluates the similarity between true and predicted labels by counting pairs that are assigned in the same or different clusters compared to true labels. The values go between 1 (perfect clustering), 0 (labels are randomly generated) and -1 (correlation worse than random). While NMI is based on entropy and information theory, ARI is counting pairs and adjusting it for chance grouping.

Different edge-weight assignment strategies are evaluated to assess how sensitive the information access clustering results are to assumptions about edge influence. Simulations with weighted and NDLib graphs present the baseline and the most intuitive approach, and should yield the same results due to the equivalency between them. On the other hand, random, uniform, and trivalency shape synthetic activation patterns that are used to compare performances and analyze which produces the most meaningful partitions.

Among the three single-layer networks, the Co-sponsorship dataset has the highest similarity scores between different simulation types, as depicted in Table 4.16 and Table 4.17 respectively. The biggest difference is for *trivalency* strategy, while *weighted* and *NDLib*, as equivalent methods, have almost the same results (NMI=0.96, ARI=0.98).

	Weighted	**Random**	**Trivalency**	**Uniform**	**NDLib**
Weighted	1.00	0.93	0.77	0.92	0.96
Random		1.00	0.73	0.90	0.90
Trivalency			1.00	0.77	0.78
Uniform				1.00	0.95
NDLib					1.00

Table 4.16: NMI values for the Co-sponsorship dataset

The results of simulations on the Twitch dataset differ much more with respect to the simulation type. As stated in Table 4.18 and Table 4.19, there is a medium level of similarity between *weighted*, *random*, and *NDLib*, while the results of other methods are completely unrelated to the rest.

The Google Scholar dataset interestingly shows a higher similarity between *weighted-random* than *weighted-NDLib*. The reason for this may be the decreasing accuracy of

[1] see `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.normalized_mutual_info_score.html`

[2] see `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.adjusted_rand_score.html`

	Weighted	**Random**	**Trivalency**	**Uniform**	**NDLib**
Weighted	1.00	0.96	0.86	0.95	0.98
Random		1.00	0.83	0.94	0.95
Trivalency			1.00	0.85	0.86
Uniform				1.00	0.97
NDLib					1.00

Table 4.17: ARI values for the Co-sponsorship dataset

	Weighted	**Random**	**Trivalency**	**Uniform**	**NDLib**
Weighted	1.00	0.33	0.03	0.06	0.47
Random		1.00	0.05	0.10	0.30
Trivalency			1.00	0.00	0.02
Uniform				1.00	0.06
NDLib					1.00

Table 4.18: NMI values for the Twitch dataset

	Weighted	**Random**	**Trivalency**	**Uniform**	**NDLib**
Weighted	1.00	0.20	-0.03	-0.06	0.61
Random		1.00	0.02	0.05	0.22
Trivalency			1.00	-0.01	-0.02
Uniform				1.00	-0.05
NDLib					1.00

Table 4.19: ARI values for the Twitch dataset

NDLib simulations as the network grows: the most accurate output NDLib provided on the smallest dataset in this research (Co-sponsorship). It is also important to mention that *trivalency* has scores equal to zero in all cases on this dataset. The reason for that is clustering result of Google Scholar trivalency: 2 nodes are in one cluster, and the rest of nodes is in the other one. This failed grouping did not cause correlation with other methods and had no meaningful output.

	Weighted	**Random**	**Trivalency**	**Uniform**	**NDLib**
Weighted	1.00	0.57	0.00	0.17	0.29
Random		1.00	0.00	0.23	0.26
Trivalency			1.00	0.00	0.00
Uniform				1.00	0.11
NDLib					1.00

Table 4.20: NMI values for the Google Scholar dataset

	Weighted	**Random**	**Trivalency**	**Uniform**	**NDLib**
Weighted	1.00	0.66	0.00	0.18	0.42
Random		1.00	0.00	0.30	0.36
Trivalency			1.00	0.00	0.00
Uniform				1.00	0.12
NDLib					1.00

Table 4.21: ARI values for the Google Scholar dataset

Across all datasets, *weighted* and *NDLib* show the highest mutual agreement, which is expected since both rely on empirical edge information. Nevertheless, the similarity between these two approaches decreases as the network size increases, suggesting reduced performance of NDLib simulations on larger graphs. The *weighted* strategy therefore provides the most consistent results. In contrast, *random*, *uniform*, and especially *trivalency* primarily serve as reference baselines, illustrating how simplified probability assignments affect information access patterns rather than representing realistic diffusion processes.

Finally, as the overall conclusion of this analysis we can state that the *weighted* simulation type is the best option regardless of the network density, cluster division lines, etc.

4.1.2 Comparison Between Different Clustering Methods

In order to compare information access clustering with other methods, it is necessary to perform statistical tests against the significant node attributes for each dataset, as shown in Table 4.22.

Cluster membership is treated as a categorical grouping variable, and node attributes are compared across the resulting clusters; statistical hypothesis testing is then carried out to

	Co-sponsorship		Twitch		Google Scholar	
	le_score	dem.	views	partner	citation	h-index
IAC Weighted	$\mathbf{< 10^{-7}}$	$\mathbf{< 10^{-7}}$	$\mathbf{< 10^{-7}}$	$\mathbf{< 10^{-7}}$	$\mathbf{< 10^{-7}}$	$\mathbf{< 10^{-7}}$
IAC Random	$\mathbf{< 10^{-7}}$	$\mathbf{< 10^{-7}}$	$\mathbf{< 10^{-7}}$	$\mathbf{< 10^{-7}}$	$\mathbf{< 10^{-7}}$	$\mathbf{< 10^{-7}}$
IAC Trivalency	$\mathbf{< 10^{-7}}$	$\mathbf{< 10^{-7}}$	$\mathbf{< 10^{-7}}$	$\mathbf{\sim 10^{-4}}$	0.85	0.9
IAC Uniform	$\mathbf{< 10^{-7}}$	$\mathbf{< 10^{-7}}$	0.66	**0.04**	$\mathbf{< 10^{-7}}$	$\mathbf{< 10^{-7}}$
Spectral	$\mathbf{< 10^{-7}}$	$\mathbf{< 10^{-7}}$	0.22	1.0	$\mathbf{\sim 10^{-5}}$	$\mathbf{\sim 10^{-5}}$
Fluid Comm.	$\mathbf{< 10^{-7}}$	$\mathbf{< 10^{-7}}$	$\mathbf{< 10^{-7}}$	0.67	$\mathbf{< 10^{-7}}$	$\mathbf{< 10^{-7}}$
Louvain	$\mathbf{< 10^{-7}}$	$\mathbf{< 10^{-7}}$	$\mathbf{< 10^{-7}}$	$\mathbf{< 10^{-7}}$	$\mathbf{< 10^{-7}}$	$\mathbf{< 10^{-7}}$
Core-Periphery	$\mathbf{< 10^{-7}}$	$\mathbf{< 10^{-7}}$	$\mathbf{< 10^{-7}}$	$\mathbf{< 10^{-7}}$	$\mathbf{< 10^{-7}}$	$\mathbf{< 10^{-7}}$

Table 4.22: p-values obtained from the Kruskal–Wallis, Fisher's Exact, and Chi-square tests. Values less than or equal to 0.05 are highlighted in bold, while extremely small values are reported as $< 10^{-7}$.

evaluate the quality of the obtained clustering. The tests evaluate whether the distribution of a specific node attribute varies significantly between clusters for each dataset and the clustering method. In all cases, the null hypothesis states that cluster assignments are statistically independent of the given node attribute; if the null hypothesis is rejected, it means that the clustering captures significant structural or semantic information in that attribute.

The choice of statistical test depends on the type of the node attribute. For continuous attributes, such as *le_ score*, *views*, *citation count*, and *h-index*, the Kruskal-Wallis test [34] is used to compare their distributions across multiple clusters. Categorical attributes are tested using the Chi-square test of independence [42] or Fisher's Exact test [21, 22]: the former is applied when sample sizes are sufficiently large, while the latter test is used for binary attributes with small or imbalanced contingency tables. The attribute *partner* in the Twitch datasets is imbalanced (large majority of nodes are not labeled as partner), so Fisher's Exact test is used in that case; on the other hand, *democrat* attribute of the Co-sponsorship dataset was suitable for the use of the Chi-square test.

Non-parametric tests are adopted throughout this analysis because node attributes in social and information networks typically exhibit heavy-tailed distributions, strong skewness, and unequal variances across clusters, as presented in the previous section. These properties violate the assumptions required by parametric alternatives, making rank-based and exact tests more appropriate and robust for the present setting. Moreover, the same tests were used in [4] and [28].

Generally, the tests show that the discovered clusters align strongly with meaningful node attributes across datasets, with the most consistent evidence coming from IAC with *weighted* and *random* edge-weigts, and from Louvain and Core-Periphery (nearly all p-values $< 10^{-7}$). In Co-sponsorship, both le_score and democrat (party affiliation) are highly associated with cluster membership for every method, reflecting the known structured polarization of that network.

Similarly, in the Twitch dataset, IAC with *weighted* and *random* strategies captures

both audience size (views) and partner status extremely well, whereas Spectral and Fluid Communities fail on partner (p=1.0 and p=0.67), and *uniform* performs poorly for views (p=0.66), indicating that discarding edge intensity can remove connection between attributes and communities. In Google Scholar, most methods, including IAC *weighted*/*random* and Spectral, recover clusters aligned with academic impact (citations, h-index), while *trivalency* notably does not (p=0.85 and p=0.90), due to the failed clustering results for this case, as mentioned before.

Overall, the consistently low p-values reported in Table 4.22 provide strong statistical evidence against the null hypothesis of independence between cluster membership and node attributes across datasets. In particular, IAC with *weighted* and *random* edge-weight strategies, as well as Louvain and Core-Periphery methods, demonstrate the most robust and consistent groupings, as indicated by the systematic rejection of independence for nearly all tested attributes. These results support the conclusion that information access clustering produces meaningfully structured partitions that are comparable to well-established baselines, while also highlighting that the statistical strength of the detected associations depends critically on the chosen edge-weight assignment strategy.

In addition, we have to evaluate whether information access clustering leads to new clustering results or yields similar to the well-known methods. The resulting clusters were compared against the same community detection algorithms using Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI). In the case of the Cosponsorship dataset, Table 4.23, the consistently high ARI (0.43-0.92) and NMI (0.40-0.89) values across all IAC edge-weight strategies indicate a strong alignment between IAC and those produced by Spectral, Louvain, and Fluid Community methods. This suggests that for networks with clear and well-organized community structure, IAC tends to recover patterns that are already captured by topology-driven approaches rather than introduce novel partitions. In contrast, the Twitch and Google Scholar datasets (Tables 4.24 and 4.25) have similarity scores close to zero. This result demonstrates that IAC is based on diffusion-related features of networks, resulting in different partitions that are not possible with the topology-oriented clustering methods currently in use.

In summary, these results demonstrate that IAC offers different results than baseline methods for cases where community structure is weak or highly heterogeneous, while giving another interpretation of the results, focused on diffusion and information access as social capital more than on the pure topology.

	Weighted	**Random**	**Trivalency**	**Uniform**
Spectral	0.90 / 0.83	0.92 / 0.86	0.82 / 0.72	0.89 / 0.83
Fluid Comm.	0.76 / 0.69	0.78 / 0.71	0.71 / 0.62	0.75 / 0.71
Louvain	0.84 / 0.77	0.84 / 0.76	0.77 / 0.67	0.86 / 0.80
Core-Periphery	0.45 / 0.41	0.44 / 0.40	0.43 / 0.40	0.47 / 0.42

Table 4.23: Comparison of IAC with baseline clustering methods on the Co-sponsorship dataset (ARI / NMI).

	Weighted	Random	Trivalency	Uniform
Spectral	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
Fluid Comm.	0.00 / 0.00	0.00 / 0.00	0.01 / 0.02	0.00 / 0.00
Louvain	0.01 / 0.01	0.00 / 0.01	0.00 / 0.01	0.00 / 0.01
Core-Periphery	0.04 / 0.04	0.03 / 0.08	0.01 / 0.00	0.00 / 0.00

Table 4.24: Comparison of IAC with baseline clustering methods on the Twitch dataset (ARI / NMI).

	Weighted	Random	Trivalency	Uniform
Spectral	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
Fluid Comm.	0.03 / 0.03	0.03 / 0.03	0.00 / 0.00	0.00 / 0.01
Louvain	0.00 / 0.04	0.00 / 0.04	0.00 / 0.00	0.00 / 0.03
Core-Periphery	0.00 / 0.01	0.01 / 0.01	0.00 / 0.00	0.04 / 0.03

Table 4.25: Comparison of IAC with baseline clustering methods on the Google Scholar dataset (ARI / NMI).

4.1.3 Execution Time Analysis and Scalability

Information access clustering is based on running a large number of simulations with the purpose of computing the average activation probability for every pair of nodes. The significant role in the effectiveness of this algorithm is execution time, i.e. how much does it take on average to run one Linear Threshold Model simulation.

	Weighted	Random	Uniform	Trivalency	NDLib
Co-sponsorship	0.018	0.018	0.012	0.015	1.275
Twitch	0.155	0.102	0.058	0.062	74
Google Scholar	0.496	0.214	0.115	0.120	306

Table 4.26: Average running time per simulation in seconds

Table 4.26 provides average running times for each dataset and each edge-weight assigning strategy in seconds. It is clear that running time is proportional with number of nodes and edges, having lowest numbers at the smallest dataset (Co-sponsorship) and the highest at the biggest dataset being Google Scholar. Regarding edge weight types, they are rather similar, with *weighted* being larger than the rest.

To improve execution time, NDLib simulations were executed in parallel using 8 processes. Reported times for NDLib in Table 4.26 are normalized to a non-parallel execution, ensuring comparability with other methods.

The most important comparison here is the one between running times for definition LTM (NDLib) and live-edge LTM, with the former being the obvious outlier. NDLib simulations are on average 400 times slower than *weighted* simulations, which are their equivalent. These numbers show a huge gap in scalability between those two methods, and

the reason for this is the internal implementation of those two methods. In the simulation run using NDLib, the algorithm is iterating through the graph until convergence (the moment when there are no newly activated nodes) for every seed node. On the other hand, simulation based on live-edge graph is producing one graph of that kind for every simulation, and computing reachable nodes for every seed.

For the reasons mentioned above, NDLib simulations are highly not scalable and will not be used in experiments with multi-layer networks.

4.2 Results for Multi-Layer Networks

This section introduces the results of experiments on multi-layer networks (Flickr and UK Politics). Similarly as before, we will compare outcomes of simulations with different edge-weights, as well as with established clustering methods on multi-layer graphs.

In single-layer networks, information access clustering was utilized to distinguish more influential nodes from less influential ones into different groups, having access to information as a criterion for separation and a type of social capital. Applying algorithm on multi-layer datasets intends to analyze different types of connections among the same set of nodes, utilizing Linear Threshold Model simulations.

Both approaches put IAC in a group of community detection algorithms. The main premise in this approach is that each community has a different level of social capital, which should be distinguishable with many simulations of the spread of information.

Running experiments on multi-layer networks is more complex than on single-layer ones due to an additional argument, the inter-layer threshold $\theta_{inter} \in [0, 1]$. This value indicates a minimal fraction of layers from which a certain node needs to receive activation input to activate itself, which means that for a k-layer graph there are k values of θ_{inter} that are going to produce different results. Hence, for two-layer network Flickr, experiments are run with protocols OR ($\theta_{inter} <= 0.5$) and AND ($\theta_{inter} > 0.5$). For the UK Politics network we ran experiments with 3 threshold values, corresponding to that dataset's number of layers.

In addition to this, experiments will be run on each layer separately, treating it as a single-layer network. This approach will help us determine performance difference between multi-layer graph and its layers as single-layer graphs.

The number of simulations in each experiment is constant with the single-layer experiments (10,000), and they will be executed on graphs with different edge-weight assignment strategies. All datasets have ground truth, so we will evaluate them with respect to those labels, using NMI and ARI as relevant measurements.

The results for the Flickr dataset, described in Table 4.27 and Table 4.28, provide a performance evaluation for a network with different layer semantics, Friendship, and Tag-similarity. Values for Normalized Mutal Information (NMI) vary between 0.00 and 0.45, and for Adjusted Rand Index between 0.00 and 0.30, which are generally low values. This indicates that connectivity in this social network is not that high and that the spread of information computed utilizing simulations is not indicating that community membership is strong enough.

Graph	Weighted	Random	Trivalency	Uniform
Multiplex-OR	0.45	0.38	0.38	0.46
Multiplex-AND	0.36	0.37	0.02	0.00
Friendship	0.31	0.31	0.28	0.42
Tag-similarity	0.33	0.35	0.17	0.24

Table 4.27: NMI values for the Flickr dataset

Graph	Weighted	Random	Trivalency	Uniform
Multiplex-OR	0.30	0.22	0.22	0.22
Multiplex-AND	0.12	0.13	0.00	0.00
Friendship	0.18	0.20	0.08	0.15
Tag-similarity	0.14	0.11	0.01	0.02

Table 4.28: ARI values for the Flickr dataset

The best performance is achieved by far with protocol OR across all edge-weight types, indicating that Friendship and Tag-similarity provide complementary rather than redundant community information. Union of these two layers with OR activation option outperforms each layer separately, showing that each layer holds certain amount of information about communities, but not enough to produce meaningful clustering.

Clustering with protocol AND completely failed on trivalency and uniform edge-weights, performing not much better than random cluster assignment. However, weighted and random strategies yield respectable results, but they are on the same level as the results for each layer separately. This indicates that the set of nodes activated in both Friendship and Tag-similarity graphs is very small and that those two networks do not share a lot of information about communities.

Analysis of individual layers reveals the relative contributions of different types of relationship to the community structure. The Friendship layer consistently outperforms Tag-similarity, showing that communities can be detected better by utilizing real and meaningful connections embedded in the former layer, rather than superficial content-based connections in the latter layer. This semantic difference between layers carries important information when it comes to analyzing the performance of this clustering; layers cannot be seen as equal, but their real-world meaning and method of construction must be taken into account. The reason for the different performance of those two graphs is exactly there.

A similar phenomenon is present in the results for the UK Politics network: semantical difference between layers causes different performance in clustering. The last three rows in Table 4.29 and Table 4.30 clearly show a decrease in the similarity between the single-layer results with the ground truth.

Graph Follows is semantically the closest to the idea of communities (similar to Friendship in the Flickr dataset): politicians that share views or even party affiliation also follow each other on Twitter. Smaller cohesion is in Mentions, because this feature of

Graph		Weighted	Random	Trivalency	Uniform
Entire	$\theta_{inter} \in [0, 0.33]$	0.86	0.89	0.83	0.86
	$\theta_{inter} \in (0.33, 0.67]$	0.84	0.87	0.72	0.58
	$\theta_{inter} \in (0.67, 1]$	0.60	0.49	0.44	0.02
Follows		0.96	0.96	0.83	0.86
Mentions		0.79	0.77	0.43	0.54
Retweets		0.66	0.72	0.55	0.54

Table 4.29: NMI values for the UK Politics dataset.

Graph		Weighted	Random	Trivalency	Uniform
Entire	$\theta_{inter} \in [0, 0.33]$	0.91	0.93	0.80	0.84
	$\theta_{inter} \in (0.33, 0.67]$	0.90	0.92	0.73	0.53
	$\theta_{inter} \in (0.67, 1]$	0.71	0.56	0.36	-0.01
Follows		0.97	0.97	0.80	0.85
Mentions		0.85	0.84	0.53	0.51
Retweets		0.72	0.83	0.55	0.48

Table 4.30: ARI values for the UK Politics dataset.

Twitter can also be used for polemics between people who do not share same political ideology. Finally, Retweets have the lowest rate of semantical significance because users retweet each other's posts whether they agree with them or not, to start a new topic, to prove their point, to answer the original author, etc. This variety of usage of the feature causes it to have a certain correlation with a community structure, but not the highest between these layers.

Interestingly, the best performing graph in this dataset across all simulation types is not any multi-layer, but Follows layer that produced the same results for *weighed* and *random* with scores NMI=0.96 and ARI=0.97. The slightly worse performance is with the same edge weight assigning strategy, but on the multi-layer graph with $\theta_{inter} \in [0, 0.33]$ and $\theta_{inter} \in (0.33, 0.67]$. This results indicate that the other two layers (Mentions and Retweets), although they hold valuable information themselves, are not adding it in case of multi-layer graph: they are adding noise, which causes performance drop. Regarding multi-layer networks with different values of θ_{inter}, the best result was again when the threshold is the smallest possible, and the clustering efficiency declines as we set the activation rule to be more strict.

Analysis of results based on edge-weight type clearly reveals that, regardless of the graph and threshold, weighted and random options perform much better than trivalency and uniform. The most radical situation is for the highest value of θ_{inter} where, similarly to the previous dataset, trivalency and uniform strategies yield clusters only slightly more meaningful than random clustering (NMI is 0.44 and 0.02, ARI is 0.36 and -0.01 respectively).

The IAC algorithm on this dataset achieves much higher scores than in the previous

example, which suggests that political interaction networks exhibit clearer and more well-defined community structures than social media networks, likely reflecting the polarized nature of political discourse where users cluster into distinct ideological groups with clear boundaries.

Since there are no node attributes in the multi-layer datasets that can be interpreted as external information access measures, we construct such a measure for the UK Politics dataset derived from Twitter list membership. Twitter lists are manually organized collections of accounts created by users to group politically relevant actors. Inclusion in such lists reflects externally assigned recognition rather than interaction-based connectivity.

Let $X \in \{0,1\}^{L \times N}$ denote the list-user incidence matrix, where $X_{i,v} = 1$ if user v appears in list i, and 0 otherwise. The *List Influence* of the user v is defined as the total number of public Twitter lists that include the user, i.e.,

$$\text{ListInfluence}(v) = \sum_{i=1}^{L} X_{i,v}.$$

This measure captures externally assigned importance and collective recognition, independently of the follower, mention, or retweet network structure. It is a heuristic for political influence, similar to the citation count or h-index in the Google Scholar dataset.

Furthermore, this measure is computed for all nodes in the UK Politics network, and the values are grouped by the resulting clusters obtained from the three best performing IAC variants: random ($\theta_{inter} \in [0, 0.33]$ and $(0.33, 0.67]$) and weighted ($\theta_{inter} \in [0, 0.33]$), as visible on the Figure 4.7.

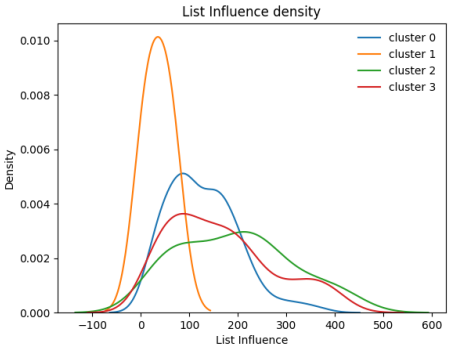
Moreover, some clusters are concentrated at low values (narrow peaks around lower ranges), while others show heavier right tails and higher central mass, indicating more externally recognized politicians. In particular, one cluster repeatedly concentrates highly influential users (broad distributions extending toward large values), whereas another consistently captures low-recognition or peripheral accounts (especially visible on the first two figures).

These results show, similar to the distributions of external information access measures in single-layer networks, that some clusters tend to have significantly higher values than the other ones, showing the difference in network influence. Larger differences are visible on the figures with random strategy, compared to those with weighted.

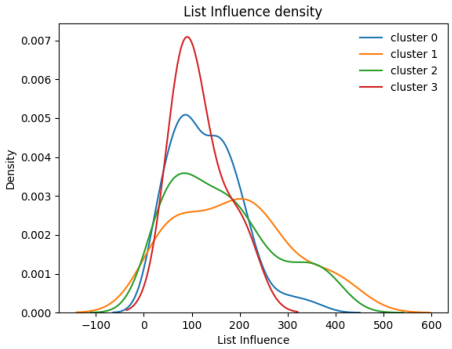
Finally, both high agreement of the IAC results with the ground truth as well as its interpretation through information access and network influence show that this clustering method yields novel insights and valuable results.

4.2.1 Comparison Between Different Clustering Methods

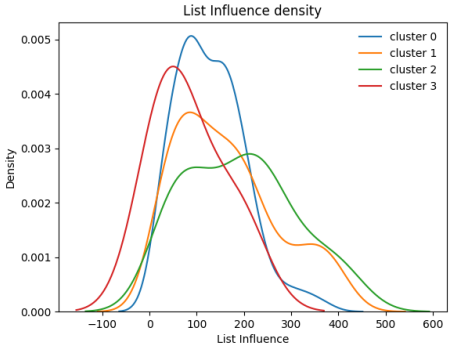
Information access clustering (IAC) of multi-layer networks is an addition to the family of graph clustering algorithms. In order to put it in the context of existing methods, it is important to compare them and evaluate the obtained results. For that purpose, in this subsection the previously presented outputs are contrasted to the results of well-known methods.



(a) Random ($\theta_{inter} \in [0, 0.33]$)



(b) Random ($\theta_{inter} \in (0.33, 0.67]$)



(e) Weighted ($\theta_{inter} \in [0, 0.33]$)

Figure 4.7: Density of *List Influence* across clusters in the three best-performing variants (UK Politics dataset).

Method	**NMI**	**ARI**
SpectralMix	0.64	0.51
IAC weighted (OR)	**0.45**	**0.30**
IAC weighted (AND)	**0.36**	**0.12**
DMGC	0.33	0.29
CrossMNA	0.23	0.19
MARINE	0.22	0.14
HAN	0.10	0.12
DMGI	0.10	0.07
DGI	0.09	0.09
ANRL	0.05	0.05

Table 4.31: NMI and ARI values for the Flickr dataset across different clustering methods. Methods are ordered in decreasing order of NMI.

The comparison between IAC with the weighted strategy (as the best performing one) and the already existing clustering graph methods is presented in Table 4.31 for the Flickr dataset. The results for the baseline methods were reported in [48]. With the value of NMI=0.64 and ARI=0.51, SpectralMix performs the best among the baseline techniques, easily exceeding the others. While techniques like MARINE, HAN, DMGI, DGI, and ANRL typically produce low agreement with ground-truth clustering, with both NMI and ARI scores typically falling below 0.15, DMGC (NMI=0.33, ARI=0.29) and CrossMNA (NMI=0.23, ARI=0.19) are the next best performers.

Compared to these established baselines, the information access clustering achieves competitive results. IAC weighted achieves an NMI of 0.45 and ARI of 0.30 using the protocol OR, making it the second-best method overall, surpassing all other baselines but trailing SpectralMix. The AND protocol outperforms the majority of neural embedding-based techniques, including DMGI, DGI, and ANRL, but produces somewhat worse results (NMI=0.36, ARI=0.12) than the OR protocol. It is worth mentioning that all graphs in this table except DMGC and CrossMNA use node attributes in clustering.

In order to evaluate the effectiveness of the IAC, its results on the UK Politics dataset were compared with those reported in the literature, Table 4.32. This dataset has been extensively used as a benchmark for multi-layer community detection, with ground-truth communities corresponding to the major political parties, as described earlier. Previous work includes a range of approaches: modularity-based algorithms such as Louvain and CNM [58], various non-negative matrix factorization (NMF) models (including GNMF, NMTF, TriNMF, MultiNMF, and DualNMF [58]), embedding-based methods such as C-RSP [23], and multilayer null-model-based modularity methods such as MNavrg, SDarvg, SDlocal, SDratio, and Aggregate [41].

Overall, the literature reports NMI values in the range 0.59–1.00 and ARI values up to 0.76. For example, the best performing variant from [58], DualNMF with endorsement-filtered and weighted retweet+mention connectivity (R+ΔM_w), achieved NMI=0.64 and ARI=0.76. Other NMF-family approaches tend to perform somewhat lower, with NMI typically between 0.40 and 0.55. By contrast, null-model modularity methods [41] produce very high agreement with ground truth, with NMI values often reaching 0.98–1.00, although under more constrained multi-layer formulations. Similarly, C-RSP, which embeds the entire three-layer graph, achieved a more moderate NMI=0.60, closer to the levels reported by the factorization methods.

Compared to these baselines, the IAC framework demonstrates consistently competitive performance (Tables 4.29 and 4.30). On the multi-layer graph, IAC achieves NMI values between 0.83 and 0.89 for low inter-layer threshold ($\theta_{inter} \in [0, 0.33]$), which are directly comparable to the best NMF-based approaches. Even as θ_{inter} increases, IAC maintains a respectable performance (NMI is between 0.72 and 0.87), although at the highest levels of threshold ($\theta_{inter} > 0.67$) the scores decline, indicating reduced stability of cross-layer information access when the threshold is rather strict.

On single layers, IAC also shows competitive performance with respect to other methods. For the Follows layer, IAC yields NMI between 0.96 and 0.97, and ARI=0.97, which match or exceed the best single-layer results, such as NMI=0.96 from [41] for the Follows

Method	Graph	Content	ARI	NMI
Louvain	R+M	-	0.47	0.59
	R+ΔM	-	0.37	0.59
	R+ΔM$_w$	-	0.43	0.59
CNM	R+M	-	0.57	0.53
	R+ΔM	-	0.62	0.65
	R+ΔM$_w$	-	0.62	0.65
k-means	-	user×word	0.24	0.20
NMF	-	user×word	0.15	0.17
GNMF	R+M	user×word	0.50	0.41
	R+ΔM	user×word	0.61	0.49
	R+ΔM$_w$	user×word	0.65	0.55
NMTF	R+M	user×word, tweet×word	0.64	0.26
	R+ΔM	user×word, tweet×word	0.57	0.25
	R+ΔM$_w$	user×word, tweet×word	0.53	0.38
TriNMF	R+M	user×word, user×domain	0.37	0.32
	R+ΔM	user×word, user×domain	0.56	0.46
	R+ΔM$_w$	user×word, user×domain	0.64	0.50
TriNMF	R+M	user×word, user×hashtag	0.52	0.43
	R+ΔM	user×word, user×hashtag	0.46	0.38
	R+ΔM$_w$	user×word, user×hashtag	0.50	0.38
MultiNMF	R+M	user×word	0.40	0.33
	R+ΔM	user×domain	0.57	0.44
	R+ΔM$_w$	user×hashtag	0.61	0.50
DualNMF	R+M	user×word	0.57	0.51
	R+ΔM	user×word	0.73	0.61
	R+ΔM$_w$	user×word	0.76	0.64
C-RSP	Follows + Mentions + Retweets	-	-	0.60
Single (mention)	Mentions only	-	-	0.86
Single (follow)	Follows only	-	-	0.96
Single (retweet)	Retweets only	-	-	0.88
MNavrg	Multi-layer	-	-	1.00
SDarvg	Multi-layer	-	-	0.96
SDlocal	Multi-layer	-	-	1.00
SDratio	Multi-layer	-	-	0.98
Aggregate	Multi-layer	-	-	1.00

Table 4.32: Results comparison between different clustering algorithms on the UK Politics dataset, obtained from [23], [41], and [58].
Graph column indicates the graph or feature set used:
R+M = retweet + mention graph;
R+ΔM = retweet + endorsement-filtered mentions (triadic balance);
R+ΔM$_w$ = weighted version of R+ΔM.
Content column indicates content matrices used in NMF-based methods.

graph. On the Mentions and Retweets layers, IAC still achieves competitive scores (NMI=0.54–0.79 and ARI=0.48–0.85), while the literature values for these layers range between 0.86 and 0.88 in NMI but without corresponding ARI.

To conclude, the experiments demonstrate that the information access clustering is competitive with, and in several cases superior to, current community detection methods. On the Flickr dataset, IAC outperformed all state-of-the-art methods except SpectralMix, especially with the OR protocol. Furthermore, IAC results for the UK Politics dataset report high correlation with ground truth labels, on the similar level as the best performing algorithms on this dataset.

Moreover, a valuable insight from these results is the fact that the Follows layer in the UK Politics dataset performed better than all multi-layer graphs with different inter-layer threshold values. This proves that merging single-layer graphs into a multi-layer graph under a certain edge-weight assignment strategy and θ_{inter} does not necessarily increase performance.

Regarding choice of parameters, edge-weight assigning strategies that yield the best results are weighted and random, while trivalency and uniform underperformed on both datasets. The choice of θ_{inter} is significant for the quality of the results and should be decided on the basis of layer semantics: if layers complement each other with insight, lower threshold values are recommended. In case the layers are similar and there is noise from some of them, higher values would give a better outcome.

These findings confirm that IAC can be useful and provide results comparable to traditional methods based on modularity or matrix factorization.

5 Scalability Limitations

This chapter will address the computational cost of running LTM simulations for this research and analyze the improved version of the information access clustering (IAC) algorithm called Fast-IAC, which is an attempt to create a scalable version of IAC where only a subset of nodes is used as a seed.

The input for the IAC method is a directed weighted graph $G = (V, E, w)$, and for the number of simulations S, the complexity of this algorithm is $O\big(S \cdot n \cdot T_{reach}(G_{le})\big)$, where $n = |V|$ denotes the number of nodes in the network, and $T_{reach}(G_{le})$ the cost of finding the set of active nodes through a reachability search on a live-edge graph (in our case, BFS). It is important to note here that the complexity depends on n because that is a number of seeds from which simulations are executed (each node is the seed in a different iteration). We can also understand n as the number of columns in IA, or the number of elements in the information access signatures $s_{LTM(\alpha)}$.

In order to reduce the complexity and running time of this method, especially for large networks, [4] considered reducing the signature size and observing the performance and running time. Let us define a reduced information access signature

$$s^I(v_j) = (p_{1j}, ..., p_{ij}, ..., p_{bj}),$$

for all $v_i \in I$, where $I \subset V$ and $|I| = b$. Informally, I is a subset of nodes of size b that will be used as seeds for simulations. Furthermore, the information access representation is denoted as follows,

$$IA^I = \{s^I(v_j) \mid v_j \in V\},$$

which is a $n \cdot b$ matrix. This size reduction is beneficial because it will lead to a shorter execution time and lower local memory consumption.

However, the question remains how to determine the set of seeds I. Beilinson et al. [4] opted for the following heuristic metrics, which is used in this chapter with slight modifications:

- out-degree centrality,
- betweenness centrality,
- global PageRank, and
- random sampling.

Experiments using these metrics are run on all three single-layer datasets (Co-sponsorship, Twitch, and Google Scholar) with *weighted* strategy. The results from [4] show that the

minimal value of b at which cluster similarity converges to the clustering where $b = n$ is $\sqrt{n}$. In [28], the Twitch clusters with a reduced number of seeds became closely similar to the standard method at 1% of the nodes, measured by ARI.

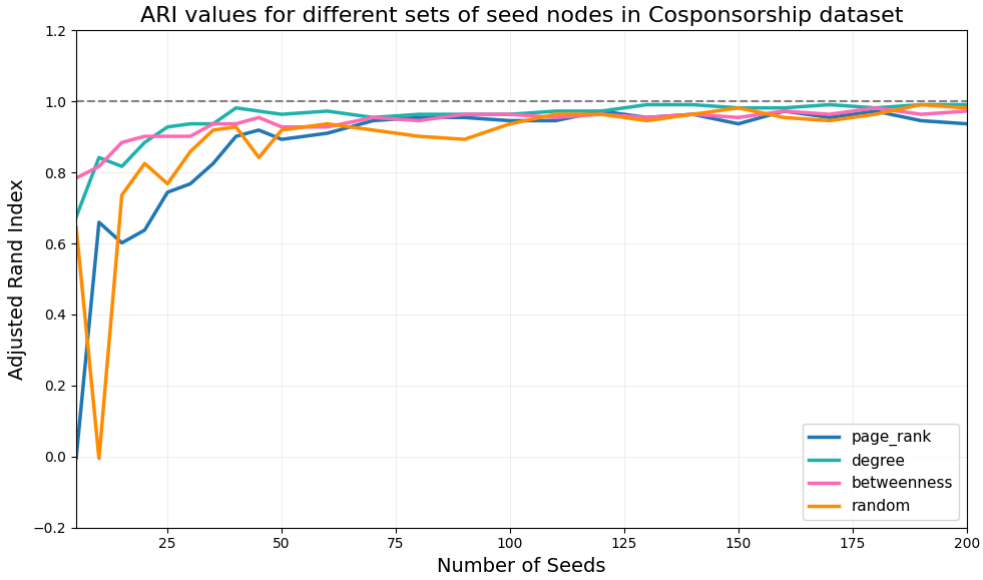


Figure 5.1: ARI values for different sets of seed nodes by four heuristic methods in the Co-sponsorship dataset

However, we obtained substantially different results. For Co-sponsorship, convergence is achieved at about 25% of the nodes used as seeds, regardless of the heuristic method. As visible in Figure 5.1, all methods (including random sampling) yield similar results after obtaining stable ARI values. For the other two datasets performance is even worse; convergence is at all not achieved. Figures 5.2 and 5.3 present an approximately linear growth of ARI values as we increase the number of nodes used as seeds, which means that Fast-IAC failed to be a scalable alternative to the standard IAC method; all nodes are necessary to determine meaningful information access clusters.

Information access clustering based on the LTM showed lack of scalability compared to the one based on ICM, as shown in [4] and [28]. The reason for this could be the different principle of the diffusion method; in ICM, each edge spreads information independently, so results are similar even if we use fewer seed nodes. On the other hand, in the LTM, the nodes are activated only when enough of their neighbors are active, making the process more sensitive to which seeds are chosen. This causes big jumps or drops in how far the information spreads when the seed set changes, which is also visible as outliers in the graphs, e.g. on Figure 5.3 the ARI value is a clear outlier for 7000 seeds.

Execution time analysis for the given networks produces similar results. In the Co-sponsorship dataset, Figure 5.4, at convergence point (25% of nodes is approx. 110) the running time per simulation varies from 0.011 to 0.013 seconds, compared to 0.018

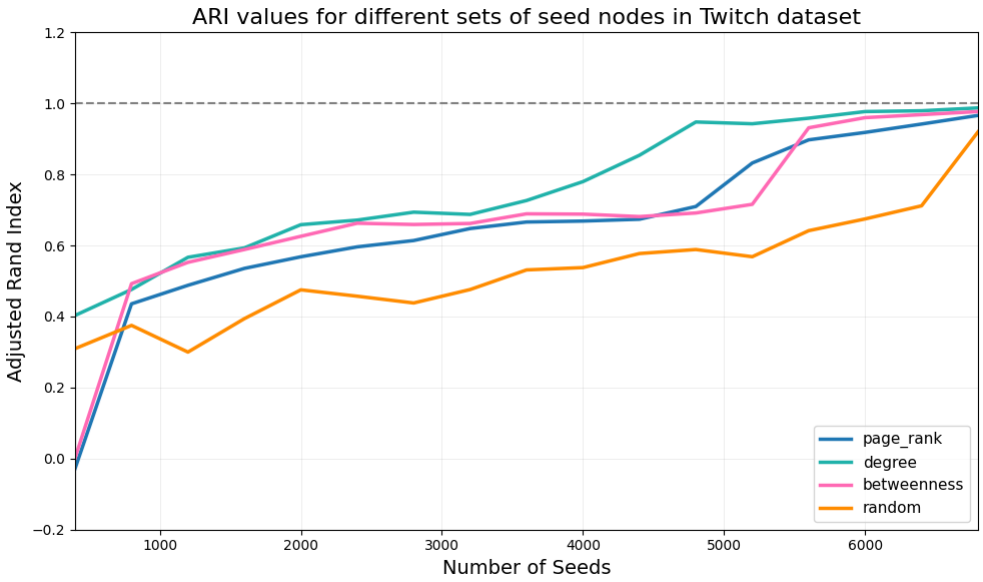


Figure 5.2: ARI values for different sets of seed nodes by four heuristic methods in the Twitch dataset

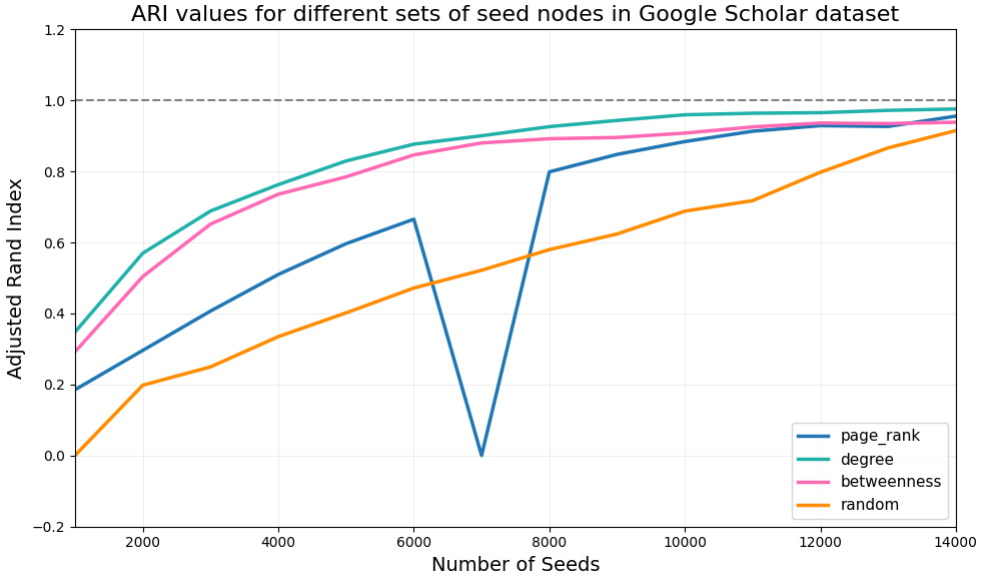


Figure 5.3: ARI values for different sets of seed nodes by four heuristic methods in the Google Scholar dataset

seconds in case of normal IAC. This means that we can use Fast-IAC on this dataset, and achieve almost the same results in a third shorter period of time. Unfortunately, for Twitch and Google Scholar networks, there is no convergence, and the simulation time increases linearly as the number of seeds increases, as visible in Figures 5.5 and 5.6.

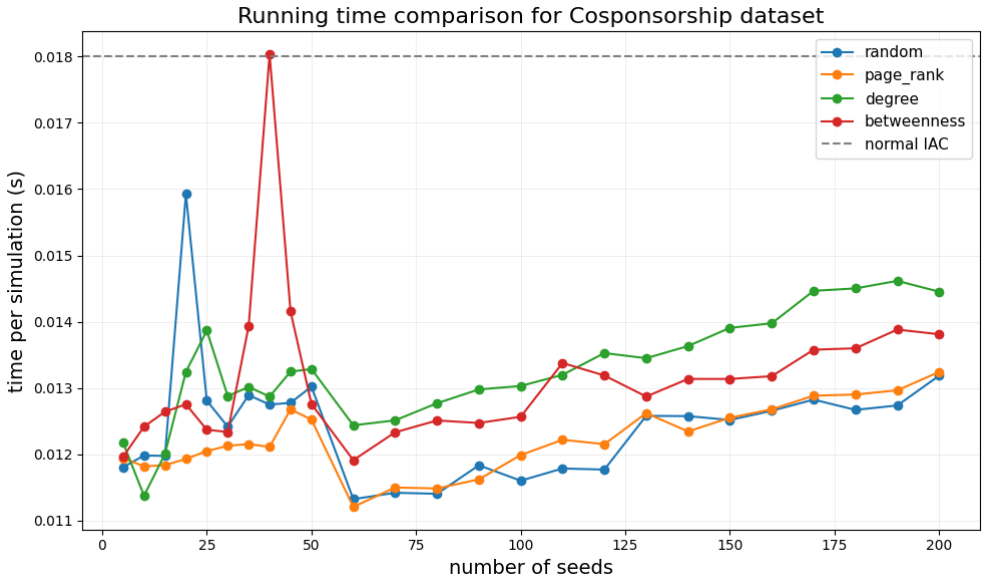


Figure 5.4: Running time comparison for Cosponsorship dataset using different seed sets and heuristic methods

In summary, this chapter demonstrates that reducing the number of seed nodes in information access clustering based on the Linear Threshold Model does not yield results similar to the standard IAC. Although Fast-IAC can offer moderate computational savings on smaller networks such as Co-sponsorship (where convergence is reached at approximately 25% of nodes), the method fails to achieve comparable clustering quality on larger and more complex networks such as Twitch and Google Scholar. The absence of convergence and the near-linear dependence of clustering quality and execution time on the number of seeds indicate that diffusion unfolded as defined in LTM is highly sensitive to seed selection, in contrast to ICM-based approaches reported in previous work.

Finally, for information access clustering based on LTM, the full set of nodes is necessary to obtain stable and meaningful clusters, limiting the practical scalability, and showing that Fast-IAC as an attempt to scale standard IAC did not succeed.

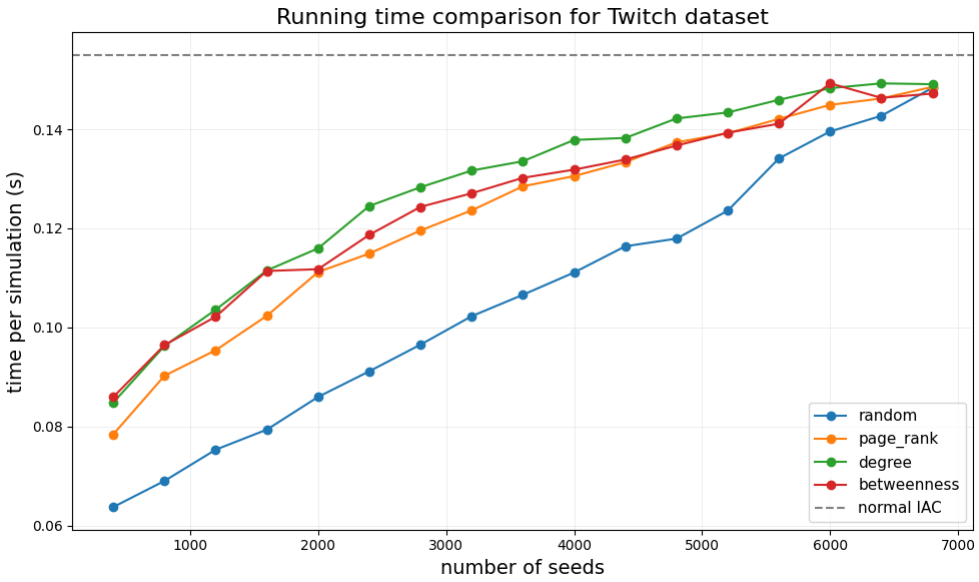


Figure 5.5: Running time comparison for Twitch dataset using different seed sets and heuristic methods

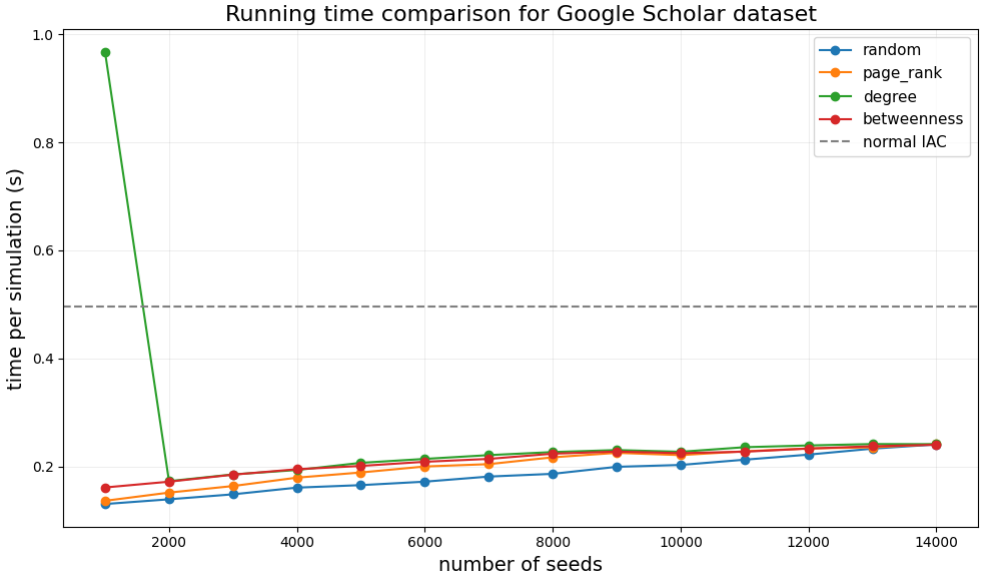


Figure 5.6: Running time comparison for Google Scholar dataset using different seed sets and heuristic methods

6 Conclusion

In the final chapter of this thesis, a summary of the findings is presented, answers to research questions, limitations of this work, and ideas for future improvements.

This research extends the information access clustering presented in [4, 28]; instead of the Independent Cascade Model we used the Linear Threshold Model, four different edge-weight methods were investigated (weighted, random, uniform and trivalency), and the algorithm was extended to cover multi-layer networks as well. In addition to that, we compared the results obtained using definition-based simulations (utilizing NDLib) with the reachability through the live-edge graph, which was proved to be equivalent [30]. Furthermore, in this research, the concept of social capital and information as a type of it was used as a theoretical background to distinguish the most influential members of the network from the less influential.

Let us provide answers to the research questions from the first chapter.

Research Questions Revisited

Can nodes be clustered according to their information access and can the resulting groups be shown to be significant using external node attributes?

Across all three single-layer datasets, the results demonstrate that IAC effectively partitions nodes into two groups showing statistically significant differences in information access, as validated by external information access measures. Specifically, applying simulations to networks of U.S. legislators, Twitch streamers, and computer science scholars produced divisions that correspond to substantial differences in their respective node attributes: legislative effectiveness score (*le_score*), *views* and *partner* status, and *h-index* and *citation count*, respectively.

How is the clustering result affected by changing model parameters?

The clustering results show moderate sensitivity to the change of model parameters, mainly through their impact on the LTM diffusion dynamics and the resulting information access values. Variations in edge-weight assignment strategies (weighted, random, uniform, and trivalency) influence cluster boundaries the most. Our attempt to scale this clustering method by reducing the number of seeds yielded poor results, showing that, generally, all nodes in the network are needed as seeds in order to obtain meaningful clusters.

Nevertheless, across all datasets, two clusters remained stable for weighted, random, and uniform edge-weights. In contrast, the trivalency setting consistently degraded performance by restricting diffusion variability, leading to weaker separation and reduced external validation scores. Overall, while simulation type (edge-weight) choices affect memberships, the clustering behavior of IAC remains robust under most edge-weights.

Can clustering work on multi-layer networks? How do those results differ from using each layer as an independent single-layer graph?

Applying the method to multi-layer networks yielded satisfactory results; consistency with the ground truth is on the same level as when using well-known community detection methods, sometimes even outperforming them. As the best edge-weight assigning methods, we have to emphasize *weighted* and *random*, while the other two performed rather poorly.

In order to achieve the best performance of the multi-layer graphs under IAC, it is important to properly set the inter-layer threshold according to the semantics of the network layers. This is a valuable insight from the research; there is no one-for-all solution, but rather a threshold value should be tailored for each dataset specifically based on how much information layers are sharing between each other. For example, results on the Flickr dataset showed that multi-layer network performs better than each layer separately. On the other hand, it is proven on the UK Politics dataset that clustering of multi-layer network does not necessarily outperform clustering of each of its layers independently, which goes against the implicit hypothesis of this research that multi-layer networks will have superior results because of the compounded effect of multiple sources of information.

What distinguishes information access clustering from other established clustering algorithms?

The core of the IAC method is to recognize information access as a structural social capital that can be used as a measure of distinction between more and less influential nodes in a network. In order to obtain this measure, IAC is running simulations on a network, which makes this method diffusion-driven, as opposed to other methods that are mainly structural and are relying on topology alone. This results in a grouping of nodes based on functional similarity in diffusion behavior rather than solely on connectivity patterns. This makes the method easier to identify nodes that may not be densely connected but have similar roles in the network when it comes to information spread and access, which offers a complementary perspective to the well-known graph clustering algorithms.

Limitations and future work

This research was done within the scope of the master thesis, so it has limitations and constraints. Firstly, it is important to mention that due to computing resources, some original datasets were filtered and reduced in size, as described in the corresponding chapter. Larger datasets would surely yield more relevant results. Furthermore, the Linear Threshold Model imposes a normalization constraint requiring that, for each node, the sum of incoming edge weights does not exceed 1. This modeling constraint made the edge-weight assignment process more complex than originally planned; for some nodes, edge-weights had to be assigned iteratively with reduced values. As a consequence, this affected the stability and performance of the *uniform* and *trivalency* strategies, especially for multi-layer networks.

Future work in this field should tackle different variations of algorithm parameters: clustering with more clusters, change in edge-weight assigning, etc. For multi-layer

clustering, it would be valuable to see how the algorithm works with a dataset with a much larger number of layers (e.g. $k = 10$). Apart from algorithm improvements, there is an entire area of potential research on the obtained results. Applying this method to some domain-specific dataset can lead to getting valuable insights on which potential problems a network has regarding polarization, inequality and division into closed communities. Those issues can be solved by developing computational methods for adding edges to the network to reduce inequality, continuing the work presented in [20].

To conclude, the added value and novelty of this work is in the area of community detection clustering with respect to information access as a distinction measure.

Bibliography

[1] Nesreen K. Ahmed, Ryan Rossi, John Boaz Lee, Theodore L. Willke, Rong Zhou, Xiangnan Kong, and Hoda Eldardiry. Learning role-based graph embeddings, 2018.

[2] Sinan Aral, Lev Muchnik, and Arun Sundararajan. Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks. *Proceedings of the National Academy of Sciences of the United States of America*, 106:21544–9, 12 2009.

[3] Eytan Bakshy, Itamar Rosenn, Cameron Marlow, and Lada Adamic. The role of social networks in information diffusion, 2012.

[4] Hannah C. Beilinson, Nasanbayar Ulzii-Orshikh, Ashkan Bashardoust, Sorelle A. Friedler, Carlos Eduardo Scheidegger, and Suresh Venkatasubramanian. Clustering via information access in a network. *CoRR*, abs/2010.12611, 2020.

[5] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[6] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[7] S. Boccaletti, G. Bianconi, R. Criado, C.I. del Genio, J. Gómez-Gardeñes, M. Romance, I. Sendiña-Nadal, Z. Wang, and M. Zanin. The structure and dynamics of multilayer networks. *Physics Reports*, 544(1):1–122, 2014.

[8] Stephen P Borgatti and Martin G Everett. Models of core/periphery structures. *Social Networks*, 21(4):375–395, 2000.

[9] Pierre Bourdieu. The forms of capital. In John G. Richardson, editor, *Handbook of Theory and Research for the Sociology of Education*, pages 241–258. Greenwood, New York, 1986.

[10] Ronald S. Burt. The network structure of social capital. *Research in Organizational Behavior*, 22:345–423, 2000.

[11] Vineet Chaoji, Sayan Ranu, Rajeev Rastogi, and Rushi Bhatt. Recommendations to boost content spread in social networks. *WWW'12 - Proceedings of the 21st Annual Conference on World Wide Web*, 04 2012.

[12] Yang Chen, Cong Ding, Jiyao Hu, Ruichuan Chen, Pan Hui, and Xiaoming Fu. Building and Analyzing a Global Co-Authorship Network Using Google Scholar Data. In *Proc. of 26th International World Wide Web Conference (WWW 2017) Companion*, 2017.

[13] N.A. Christakis and J.H. Fowler. *Connected: The Surprising Power of Our Social Networks and How They Shape Our Lives*. Little, Brown, 2009.

[14] Xiaokai Chu, Xinxin Fan, Di Yao, Zhihua Zhu, Jianhui Huang, and Jingping Bi. Cross-network embedding for multi-network alignment. In *The World Wide Web Conference (WWW '19)*, pages 273–284, 2019.

[15] Tristan Claridge. Social capital at different levels and dimensions: a typology of social capital, June 2023.

[16] James S. Coleman. Social capital in the creation of human capital. *American Journal of Sociology*, 94(1):S95–S120, 1988.

[17] Yaofeng Desmond Zhong, Vaibhav Srivastava, and Naomi Ehrich Leonard. On the linear threshold model for diffusion of innovations in multiplex social networks. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 2593–2598, 2017.

[18] L.C. Dodd and B.I. Oppenheimer. *Congress Reconsidered, 10th Edition*. SAGE Publications, 2013.

[19] Ming-Han Feng, Chin-Chi Hsu, Cheng-Te Li, Mi-Yen Yeh, and Shou-De Lin. Marine: Multi-relational network embeddings with relational proximity and node attributes. In *Proceedings of the 2019 World Wide Web Conference (WWW '19)*, 2019.

[20] Benjamin Fish, Ashkan Bashardoust, Danah Boyd, Sorelle Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian. Gaps in information access in social networks? In *The World Wide Web Conference*, WWW '19. ACM, May 2019.

[21] Ronald A. Fisher. On the interpretation of χ^2 from contingency tables, and the calculation of P. *Journal of the Royal Statistical Society*, 85(1):87–94, 1922.

[22] Ronald A. Fisher. *Statistical Methods for Research Workers*. Oliver and Boyd, Edinburgh, 5th edition, 1934.

[23] S. Gamage, A. L. de la Peña, and M. Saerens. Multi-view graph embedding using randomized shortest paths. In *Proceedings of the 2018 IEEE International Conference on Big Data (Big Data)*, pages 1508–1517. IEEE, 2018.

[24] Mark S. Granovetter. The strength of weak ties. *American Journal of Sociology*, 78(6):1360–1380, 1973.

[25] Mark S. Granovetter. Threshold models of collective behavior. *American Journal of Sociology*, 83(6):1420–1443, 1978.

[26] Derek Greene and Pádraig Cunningham. Producing a unified graph representation from multiple social network views. *CoRR*, abs/1301.5809, 2013.

[27] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 855–864. ACM, 2016.

[28] Clara Halmdienst. Clustering nodes according to their access to information in networks. Master's thesis, University of Vienna, Vienna, Austria, 2024. Master's thesis; supervisor: Yllka Velaj.

[29] J. E. Hirsch. An index to quantify an individual's scientific research output. *Proceedings of the National Academy of Sciences*, 102(46):16569–16572, November 2005.

[30] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, page 137–146, New York, NY, USA, 2003. Association for Computing Machinery.

[31] William O. Kermack and A. G. McKendrick. A contribution to the mathematical theory of epidemics. *Proceedings of The Royal Society A: Mathematical, Physical and Engineering Sciences*, 115(772):700–721, 1927.

[32] Elias Boutros Khalil, Bistra Dilkina, and Le Song. Scalable diffusion-aware optimization of network topology. In *KDD '14: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, page 1226–1235, New York, NY, USA, 2014. Association for Computing Machinery.

[33] Masahiro Kimura, Kazumi Saito, and Hiroshi Motoda. Solving the contamination minimization problem on networks for the linear threshold model. In *Advances in Knowledge Discovery and Data Mining*, pages 977–984, 12 2008.

[34] William H. Kruskal and W. Allen Wallis. Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, 47(260):583–621, 1952.

[35] Chris Kuhlman, Gaurav Tuli, Samarth Swarup, Madhav Marathe, and S. Ravi. Blocking simple and complex contagion by edge removal. In *Proceedings - IEEE International Conference on Data Mining, ICDM*, 12 2013.

[36] Zhepeng Li, Xiao Fang, Xue Bai, and Olivia Sheng. Utility-based link recommendation for online social networks. *Management Science*, 63, 12 2015.

[37] Nan Lin. *Building a Network Theory of Social Capital: Theory and Research*, pages 3–28. Routledge, 07 2017.

[38] Dongsheng Luo, Jingchao Ni, Suhang Wang, Yuchen Bian, Xiong Yu, and Xiang Zhang. Deep multi-graph clustering via attentive cross-graph association. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, WSDM '20, page 393–401, New York, NY, USA, 2020. Association for Computing Machinery.

[39] Chanyoung Park, Donghyun Kim, Jiawei Han, and Hwanjo Yu. Unsupervised attributed multiplex network embedding (dmgi). In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020. Also available as arXiv:1911.06750.

[40] Ferran Parés, Dario Garcia-Gasulla, Armand Vilalta, Jonatan Moreno, Eduard Ayguadé, Jesús Labarta, Ulises Cortés, and Toyotaro Suzumura. Fluid communities: A competitive, scalable and diverse community detection algorithm, 2017.

[41] Subhadeep Paul and Yuguo Chen. Null models and community detection in multilayer networks, 2020.

[42] Karl Pearson. X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302):157–175, 1900.

[43] Robert D. Putnam, Robert Leonardi, and Raffaella Y. Nonetti. *Making Democracy Work: Civic Traditions in Modern Italy*. Princeton University Press, 1993.

[44] M. Puck Rombach, Mason A. Porter, James H. Fowler, and Peter J. Mucha. Core-periphery structure in networks. *SIAM Journal on Applied Mathematics*, 74(1):167–190, 2014.

[45] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[46] Benedek Rozemberczki, Carl Allen, and Rik Sarkar. Multi-scale attributed node embedding, 2021.

[47] Benedek Rozemberczki and Rik Sarkar. Fast Sequence Based Embedding with Diffusion Graphs. In *International Conference on Complex Networks*, pages 99–107, 2018.

[48] Ylli Sadikaj, Yllka Velaj, Sahar Behzadi, and Claudia Plant. Spectral clustering of attributed multi-relational graphs. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery; Data Mining*, KDD '21, page 1431–1440. ACM, August 2021.

[49] Daniel Sheldon, Bistra Dilkina, Adam N. Elmachtoub, Ryan Finseth, Ashish Sabharwal, Jon Conrad, Carla P. Gomes, David Shmoys, William Allen, Ole Amundsen, and William Vaughan. Maximizing the spread of cascades using network design, 2012.

[50] Chonggang Song, Wynne Hsu, and Mong Lee. Temporal influence blocking: Minimizing the effect of misinformation in social networks. In *ICDE 2017*, pages 847–858, 04 2017.

[51] Christopher Sower. Medical innovation: A diffusion study. *Administrative Science Quarterly*, 12(2):355–361, 1967.

[52] Lei Tang, Xufei Wang, Huan Liu, and Lei Wang. A multi-resolution approach to learning with overlapping communities. In *Proceedings of the First Workshop on Social Media Analytics*, SOMA '10, page 14–22, New York, NY, USA, 2010. Association for Computing Machinery.

[53] Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. Deep graph infomax. In *International Conference on Learning Representations (ICLR)*, 2019.

[54] Ulrike von Luxburg. A tutorial on spectral clustering, 2007.

[55] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Peng Cui, Philip S. Yu, and Yanfang Ye. Heterogeneous graph attention network. In *The World Wide Web Conference (WWW '19)*, 2019. Also available as arXiv:1903.07293.

[56] Yao Zhang, abhijin Adiga, Sudip Saha, Anil Vullikanti, and B Prakash. Near-optimal algorithms for controlling propagation at group scale on networks. *IEEE Transactions on Knowledge and Data Engineering*, PP, 09 2016.

[57] Zhen Zhang, Hongxia Yang, Jiajun Bu, Sheng Zhou, Pinggang Yu, Jianwei Zhang, Martin Ester, and Can Wang. Anrl: Attributed network representation learning via deep neural networks. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3155–3161, 2018.

[58] M. Özer, M. Gök, H. Velibeyoğlu, C. Aykanat, and S. A. Seno. Community detection in political twitter networks using nonnegative matrix factorization methods. *arXiv preprint arXiv:1608.01771*, 2016.