Why Multilayer Networks Instead Of Simple Graphs? Modeling Effectiveness And Analysis Flexibility & Efficiency!

Sharma Chakravarthy, Abhishek Santra, and Kanthi Sannappa Komar

Information Technology Laboratory (IT Lab)
Computer Science and Engineering Department
University of Texas at Arlington
Arlington, Texas 76019, USA
Contact author: sharmac@cse.uta.edu

Abstract. We are on the cusp of analyzing a variety of data being collected in every walk of life - social, biological, health-care, corporate, climate, to name a few. We are also in search for models and analytical techniques that can accommodate more complex and increasingly large size data (scalability). Our ability to analyze large complex, disparate data for a broad set of analysis objectives differentiates big data analytics from mining which is narrow in scope. Hence, flexibility of analysis (different from scalability) is important. Concomitantly, efficiency is important due to large number of analysis needs. Our ultimate goal is to go from vertical analysis of data individually (corresponding to one of the 4 V's) to holistically (also termed fusion-based) analyze that that corresponds to all or a subset of V's!

In order to accomplish the above, we are always in search for more effective models to represent data and different analysis techniques that support flexibility of analysis, efficiency, and scalability. We want to use techniques that have worked well – whether it is for modeling, efficiency or scalability. We also want to extend these techniques and/or develop new and improved ones to accommodate more complex, diverse, and larger size data.

The goal of this paper is to provide the reader an understanding of data analysis approaches using graphs. Our thesis is that there are several ways in which a graph representation can be used – both for modeling and analysis. We will take the reader through the evolution of graph usage and relevance leading to the current state of the use of multilayer Networks (MLNs) or multiplexes for modeling and analysis. Graphs are not new, but how they are used for big data analytics is going through a transformation which is important to understand. The hope is that the reader understands the path that has led us to this juncture and how graph usage is extended!

Keywords: Graph-based modeling and analysis; Multilayer Networks; Decoupling approach; Efficiency and scalability.

1 Introduction

This is a paper on big data analytics and science that is intended to capture the progress of graph mining (one form of analysis) to a broader analysis of complex data sets using graphs. We start with traditional graph mining and trace the path towards multilayer networks for effective modeling and efficient & flexible analysis.

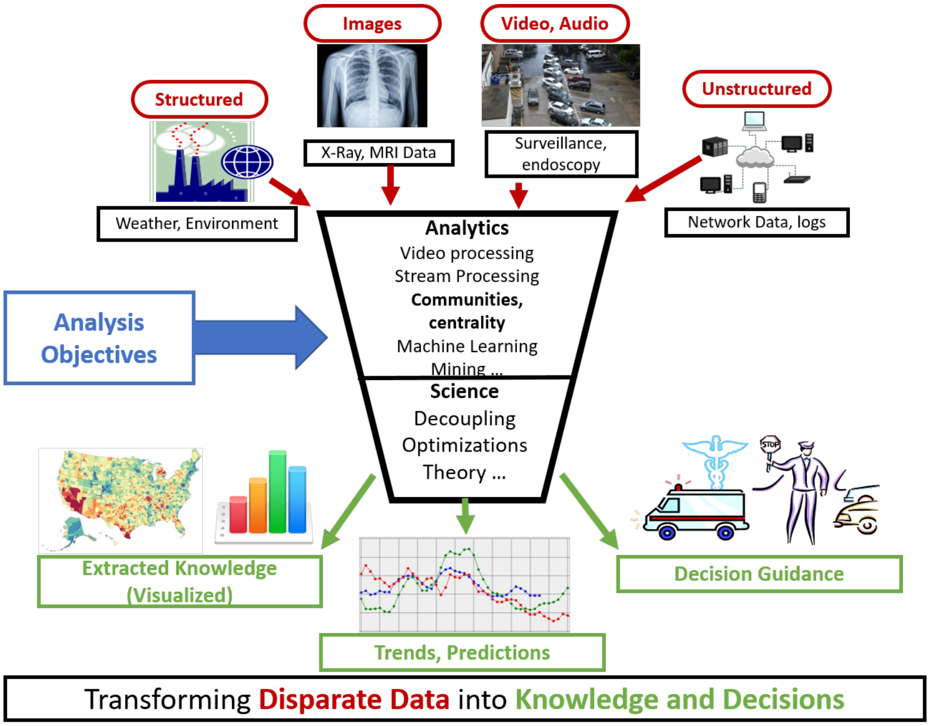

Fig. 1. High level Architecture of Big Data Analytics/Science

Data mining is the process of automatically discovering useful information from large data repositories. Data collected and validated (termed labeled) has been used for generating models (Decision trees, SVM, termed supervised approach) that can be used for predicting outcomes for new data. Data has also has been processed in a number of ways to glean patterns without using labeled data (e.g., clustering, association rules, subgraph mining, termed unsupervised). Data mining is different from querying or generating different types of reports from managed data sets using a DBMS or a data warehouse. Big data analytics use or incorporate appropriate mining and other techniques for a broader holistic analysis. Starting with text mining, a number of traditional mining techniques have been developed (decision trees, clustering, neural nets, etc.) Both supervised and unsupervised approaches have shown to be useful for different applications. Graph mining although developed in the 80's have become popular recently due to their usage in modeling and processing internet and social network. More recently, association rule mining (and its flavors) became possible with the advances in data representation, availability of large real-world data, large and cheap storage availability, and relevant technical advances. Data Mining, as we have it today, became even more important from a business perspective (similar to Data warehouses, but with different requirements) when we progressed in our ability (storage, networking, processing, and algorithms) to handle vast amounts of *real business data* (in contrast to samples of representative data) for identifying non-intuitive nuggets with certain confidence for driving business goals.

In our view, the fundamental difference between mining and big data analytics is the scope and diversity of data. The holistic aspect of analysis and the breadth (or diversity) of data along with their characteristics are the challenges that need to be taken into account. Here, the goal is more ambitious than traditional mining in that this analytics is likely to need multiple approaches working in concert. Hence, not only the need for large number of formalism, techniques, and algorithms, but also a mechanism to combine or compose them in novel ways based on user-defined or user-specified analysis objectives (see Figure 1). The long-term goal is even more ambitious in terms of requirements for holistically analyzing (and developing formalism for) disparate data that corresponds to 4V's (Volume, Velocity, Variety, and Veracity) or even 5V's (plus Value). This is the challenge currently faced by the community addressing big data analysis/science. The basic premise here is that dealing with each of the V's individually or in small combinations (as has been done up to this point to a large extent) is not sufficient, but need to include *all or combinations* of them as warranted for a holistic analysis leading to inferring better and concise **(actionable) knowledge** for decision making. Towards this end, we will present *our* previous and ongoing contributions towards big data analysis.

Figure 1 shows, at a high level, the problem of big data analysis and science. As shown in the figure, the ultimate goal is to synthesize meaningful and beneficial actionable knowledge with good confidence that can be used for decision making (what humans call wisdom which is culled from data and events based on a combination of nature and nurture (together as experience) as biologists put it) by using all available relevant disparate data coming from a variety of sources. The inverted triangle shows the reduction of large raw data into small nuggets of knowledge. Figure also shows some of the technologies that are available today (used for analytics), and a partial list of underpinnings (i.e., science) using which we develop to support these technologies. The way we see big data analysis is that instead of addressing each V (or small combinations of V), a holistic approach is the desired goal driven by analysis objectives/expectations. However, the problem is still the same as that of culling, filtering, aggregating, and inferring nuggets of (actionable) knowledge that can be used for decision making including real-time decision making. Most of the current approaches have addressed a small subset of this problem.

Although not explicitly shown in the figure, techniques for the visualization of data as well as the derived knowledge is quite important. Pictorial representation and multi-dimensional subject-oriented analysis of the results are also very useful for understanding the results of analysis.

Personalized health care is a good example of how it is critical to avail and process all types of data related to a single person (or even a community) over a period of time (lab results, X-rays, EKG, endoscopy video, MRI, etc.) to make a meaningful decision for that individual (or community) rather than using average cases which is how it is done today. Similar applications include climate change studies, monitoring earthquakes, pollution, and others.

The remainder of the paper is organized as follow. We briefly indicate our contributions to data mining over the years in Section 2. Then we focus on graph-based modeling and analysis that includes our current work in Section 3. Finally, section 4 has conclusions.

2 Data Mining or Knowledge Discovery in Databases

Data mining aims at discovering important and previously unknown patterns from the data sets. Although not explicitly termed data mining and might not have used real-world business data, the concept of understanding data, in ways that are different from querying and analysis that was available through RDBMSs and data warehouses, pre-dates them. Classification, clustering, prediction, deviation analysis, and neural networks were used by many businesses for selective marketing, credit card transaction approval, and mortgage and other types of lending. Supervised and unsupervised approaches were developed and multi-fold cross validation was widely used for establishing the accuracy of models. A number of algorithms were developed, some couched in expert systems used by businesses. Due to the limitations of storage and processing, the sizes of the data sets used were small and often statistically representative samples were used for processing (instead of all available data) and results extrapolated (or generalized) for larger data sets.

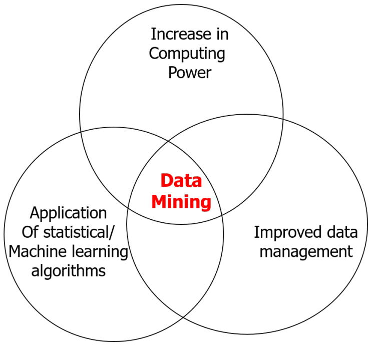
Increase in Computing Power
Data Mining
Application Of statistical/ Machine learning algorithms
Improved data management

Fig. 2. Convergence of Technologies That Facilitated Data Mining

The rapid improvement in the size of the storage devices along with with the associated drop in the cost in the 1990s, and increase in the computing power as well as the wide use of statistical approaches for processing data gave rise to the field of data mining as we know it today (see Figure 2. Suddenly, it became feasible for organizations to store unprecedented amounts of organizational data and process it. These organizations, though having a gold mine of data, were not able to fully capitalize on its value mainly because the algorithms and approaches had to be scaled to very large data sizes. Typically, the data captures the business trends over a period of time and hence using real-world business data (rather than samples) became the goal. However, the nuggets of useful knowledge hidden were not so easy to discern. To compete effectively, decision makers felt they needed to identify and utilize "nuggets of knowledge" buried in the collected data and take advantage of the high return opportunities in a timely manner. Association rule mining is a good example of this trend.

While these developments were ongoing, although graph theory has been around for a very long time, graphs for data representation were not that popular. Graph theory belonged more to the realm of mathematicians rather than its application in computer science. A few researchers were using graphs for representing data in specific domains and were trying to identify patterns in graphs around the same time businesses were trying to do the same using their transactional data. Although association rule mining took off and became widely popular with many commercial implementations, graph usage and mining caught the attention of businesses much later. One of the early work on graph mining was Subdue [28] which developed main memory algorithms for identifying substructures in graphs (or forests) that were "interesting" based on some metric. They used a information theoretic metric termed minimum description length (or MDL) for this purpose. The data sets were drawn from chemical representations, CAD circuits, etc.

Data mining became a hot research area with the advent of association rule mining and graph mining. Association rule mining [6] started with market basket analysis for identifying items bought together with given support and confidence from actual point-of-sales data. These data sets were huge (for example, Walmart's point-of-sales data around that time was estimated to be around 1Gb per day) and multiple years of data could not be held in main memory for analysis. For association rule mining of this data, novel data structures (e.g., hash tree) and approaches to reduce the number of passes on the data (to minimize I/O's) were developed. Teradata developed a parallel processing system to facilitate analysis of very large amounts of transactional data. As the search space was prohibitively large, *a priori* and other properties were identified to reduce the number of item sets carried over from iteration to the next one. Partitioning approaches were developed to reduce the number of passes on data (stored in disks) and the response time. A large amount of work followed resulting in a number of mining systems marketed by almost every major vendor.

Along the same lines, the expressiveness of graphs became important with the advent of Internet and social networks. Graphs both for modeling and analysis (e.g., mining, PageRank, etc.) and their importance for identifying important and useful graph patterns became apparent. Frequent subgraphs, identification and counting of triangles and other substructures in very large graphs became important in addition to interesting substructures. Today, there is renewed interest in using graphs for modeling and analysis of complex data as discussed in Section 3.2.

2.1 IT Lab Contributions

Information Technology Laboratory (or IT Lab) at UT Arlington or UTA (also in its previous incarnation at the University of Florida or UFL, Gainesville) has been engaged in managing, processing, and analyzing large amounts of data using diverse techniques, such as semantic and multiple query optimization [13, 23–26, 60], real-time transaction scheduling [42–44], incorporating active capability [1, 20, 27, 38, 39, 45, 59] into DBMSs, etc. An active object-oriented DBMS termed **Sentinel** was developed at UFL incorporating **Snoop** as the event specification languages. This was later integrated with the stream processing system **MavStream** to provide both event and stream processing capabilities in a seamless manner in a single system termed, MavEStream. Currently it is being extended [12] to support continuous queries on videos by extending it to support object comparison, spatial, and temporal aggregate computations MavEStream.

Specifically, from the view point of mining, we have contributed to both association rule mining [36, 52–57, 74–76] and substructure discovery [7, 16, 22]. We have also developed a number of techniques for aggregating streaming, real-time relational data termed stream data processing/analysis [2–5, 14, 15, 17–19, 21, 29, 37, 40, 46–50, 68, 69, 71, 73]. A prototype data stream management system (DSMS) **MavEStream** was implemented at UTA and the work was consolidated into an authored book.

For association rules, we have extensively evaluated performance of database algorithms on different RDBMSs and compared them. This helped us to identify some of the quirks in the optimization of SQL queries by different vendors and understand the difficulties of optimizing queries with 10 to 20 joins, a large number of them being self joins. RDBMSs query optimization were not designed with those number and types of joins in mind.

On the graph mining side, we have tried to scale the main memory substructure mining algorithm of Subdue using a number of alternative approaches. The first one [8, 16, 58] mapped graphs into relations and the substructure discovery algorithm to SQL in order to leverage the built-in capabilities of a DBMS (buffer manager, query optimizer) instead of re-inventing them for mining. This allowed us to scale the size of the graphs to millions of nodes and edges. This approach has certain limitations due to large number of joins as well DBMS's inability to order a relation using columns. Recently, we have been able to successfully scale this algorithm even further to arbitrary sizes using the map/reduce paradigm.

We have achieved data scalability using divide and conquer with and without using map/reduce. We have developed generic Map/ Reduce based algorithms for horizontal scalability of substructure discovery that can work with any partitioning strategy. The basic components of graph mining - subgraph expansion, duplicate removal and counting of isomorphic substructures were incorporated into the algorithms for the Map/Reduce paradigm by carefully orchestrating new representations. Vertical scalability was achieved by showing that these algorithms produce the same results (loss-less property) irrespective of the number of partitions. Experiments validated the advantage of using Map/Reduce based substructure discovery to scale to arbitrarily large graphs [30–32]. In an effort to analyze the partitioning strategies and associated algorithms from a performance standpoint, we have done a component cost analysis of substructure discovery in a distributed framework. The cost analysis identified places for improvements in using the range-based partitioning strategy over its counterpart. Theoretical justification along with experimental evaluation of the improvements were verified by varying a number of user parameters. The cost analysis also pointed out the portability of our algorithms to a different paradigm such as Spark to reap similar benefits.

Our approach to query processing on graphs developed a cost-based plan generator by defining and using a catalog that is relevant to graphs [33, 41]. To process a plan on large graphs, partitioning of graphs was used for processing the plan on each partition separately and combining the results in a loss-less manner. Several heuristics were developed for optimizing the number of partitions loaded for this purpose [10, 11].

3 Graph-Based Modeling and Analysis

To achieve the goals of big data analytics/science as explained in Section 1, a number of perspectives on how to analyze a data set as well as a number of approaches and their combinations need to be taken into consideration. Research is ongoing by a large number of scientists with a broad brush covering data sets from a variety of domains. In this section, we present some of the approaches that we are working on to address the big data analysis problem in a small way. This tutorial mainly focuses on this. Although they do not right now solve the big data analysis problem completely as is posited in Figure 1, they address components whose solutions are likely to contribute to the overall solution.

3.1 Modeling Data Using Single Graphs

The basic idea of graphs were first introduced in the 18th century by Swiss mathematician Leonhard Euler. Interestingly, it seems to have been applied for a real-world problem (famous Konigsberg bridge problem) – to formulate and solve it. We have come a long way from there as a topic of mathematics and recently with the explosion of social networks and internet as a computer science topic. Graph theory is ultimately the study of entities as nodes (or vertices) and relationships as edges. Weights and labels can be associated with nodes, and can include directions as well for edges. Other properties such as cycles, spanning trees can be associated with graphs. This rich graph representation can be used to abstract a large number of real-world problems from city layouts to printed circuits to social and biological networks. Studying graphs through a framework provides answers to logistics, networking, optimization, matching, and operational problems.

For the purpose of modeling and analyzing complex data we have in mind, let us consider the following data sets and associated analysis objectives.

US-based Airlines: This is a data set of six US-based airlines and their flight information among US cities. This information has been collected by us from multiple sources. The number of US cities is the same for all airlines. But their connectivity (direct flights between cities) varies.

Potential Analysis Questions: Airlines are always looking for expansion into new/existing markets (cities) to increase business. Using the above publicly available data, an interesting analysis objective would be: **Can we identify and rank cities for potential expansion as a hub for each airline taking all competitors into consideration?**

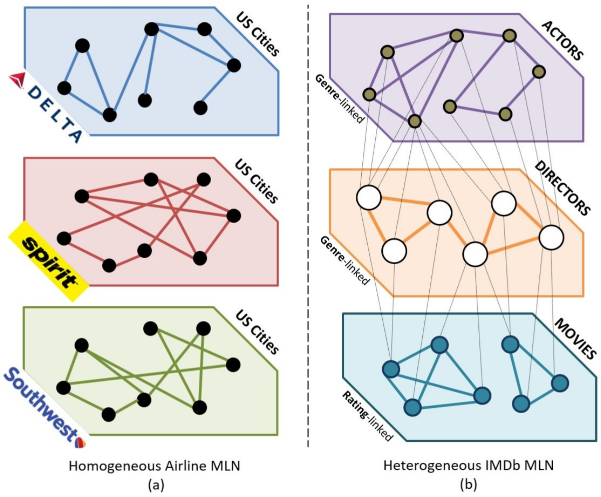

Fig. 3. MLN Modeling Alternatives

If we were to represent each airline and its direct flight connectivity, we can either do it as 3 separate simple graphs as shown in Figure 3(a) or as one graph in which we show them combined into a single graph as shown in Figure 4(a). It is easy to see that there are multiple edges in the combined graph and colors correspond to edge labels. If we do not keep relationships in each layer separate, there is information loss. We will not be able to know how many airlines have direct flights between the cities. If we keep them separate, there are no algorithms to compute communities (or hubs) on such a graph (termed an attribute graph)! This data set has the same entities in all layers and different relationships and is termed **Homogeneous Multilayer Network or HoMLN**.

However, if we represent this data set as 3 layers as shown in Figure 3(a), the challenge is to compute network properties such as communities and hubs. Currently, it is done in a way where there is loss of information due to combining multiple edges into a single edge using Boolean operations. Or it is done on the entire MLN resulting in in-efficient algorithms. Our decoupling-based multilayer network (MLN) analysis provides a solution for this.

IMDb data set: The IMDb data set captures movies, TV episodes, actor, directors and other related information, such as rating. This is a large data set consisting of movie and TV episode data from their beginnings.

Potential Analysis Question: For a large data set such as above, is it possible to **infer, through data-driven analysis, well-known actors who have worked in several common genres, *but have never worked together!*** This would be of interest to directors and producers who are scouting for bankable and novel casting for their next project!

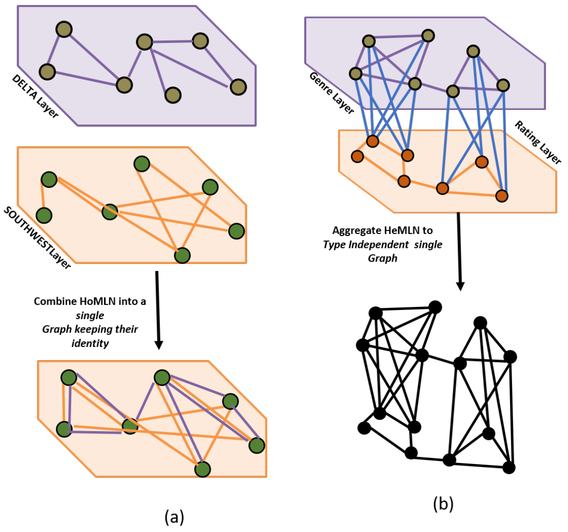
DELTA Layer
SOUTHWESTLayer
Combine HoMLN into a single Graph keeping their identity
(a)
Genre Layer
Rating Layer
Aggregate HeMLN to Type Independent single Graph
(b)

Fig. 4. Reducing MLN to Single Graphs

This data set is more complicated in that there are different entity types – actors, directors, and movies. Each entity is connected among themselves – actors acting in a specific genre, directors directing a specific genre, and movies whose ratings are in a range. In addition, there are also links (or inter-layer edges) between the actors and director entity nodes indicating which director has directed which actor at least once. Figure 3(b) shows each entity type separately and also the connections between different entity types as inter-layer edges. Note that HoMLNs have a single entity type and their connections across layers are implicit and hence not shown in Figure 3(a). Figure 4(b) shows what happens when the HeMLN in Figure 3(b) is converted into a single graph shown in Figure 4(b) using type-independence approach. That is, all the type information (essentially labels of both nodes and edges) is lost. Although network properties can be computed on the resulting graph shown in Figure 4(b), both structure and semantics of the original representation is lost.

On the other hand, if we keep edge and node labels in the aggregated graph, we will have an attribute graph that has node and edge labels. There are no algorithms for computing networking properties that we are dealing with such as community and centrality on such a graph. However, attributed graphs are widely used for substructure discovery, querying etc. An attributed graph is shown in Figure 5 (source) for a better understanding.

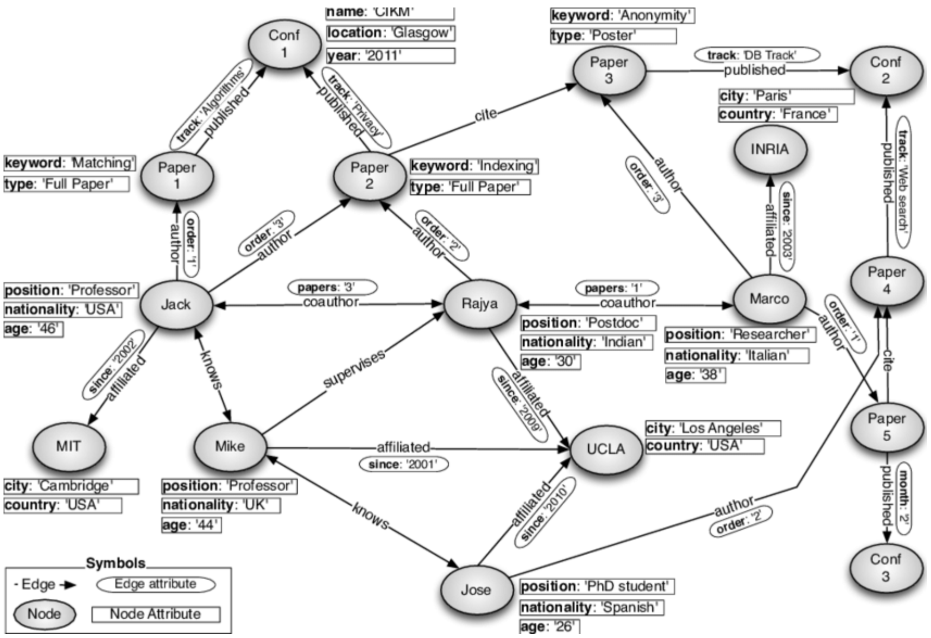

Fig. 5. Example of an Attributed Graph

This representation with multiple entity types is termed **Heterogeneous multilayer network or HeMLN**. Again the issue is which representation is more effective from a modeling perspective and which one from a computation perspective.

The above are just a couple of examples to indicate the analysis complexity and potential of the approach proposed in this paper. Data sets being analyzed may also contain features that are derived from contents (e.g., posts in Facebook) in addition to explicit ones. Such derived information can also be incorporated into this approach as shown in [77].

3.2 Modeling and Analyzing Complex Data Using MLNs

In the previous section we illustrated the difficulties of modeling complex data sets as a single graph without loss of information. In addition, computation of network properties, such as community and centrality (e.g., hub), which are available for single graphs, are not available for MLNs or multiplexes. Hence, if one wants to use multiplexes for modeling, it is imperative that we develop approaches and algorithms for computing network properties and efficiently. Network Decoupling approach is one such techniques which we will discuss in detail in this tutorial with its application to real-world data sets.

Before we go into modeling using MLNs (HoMLN or HeMLN) and computations on that, let us clearly understand the alternatives. Figure 7 shows three alternatives ways of modeling and computing network properties (currently, community and centrality). Figure 7(a) shows a HeMLN aggregated into a single graph removing node/edge identities resulting in losing entity and/or relationship identities. Computing network properties on this single graph is not preferred due to loss of information. On the other end of the spectrum, Figure 7(c) shows the same HeMLN layers and computing the result using the entire MLN as a whole. Although this has been proposed in the literature (e.g., Infomap recently), this is likely to be computationally expensive as the number of layers and data sizes become large. The advantage of modeling is lost to some extent due to computational complexity.

Figure 7(b) on the other hand proposes an approach developed by us (termed networking decoupling) where network property for each layer is computed independently (possibly in parallel) and compose them using a binary operator Θ as shown. We have shown this to be effective, can be done for Boolean operations for HoMLNs and for HeMLNs as well without aggregating and losing type information. Furthermore, we have shown it be more efficient than the approaches shown in Figures 6(a) or (c). Furthermore, the advantages of modeling is retained as well.

More clearly, current approaches, such as using the MLN as a whole [78], type-independent [35], and projection-based [9, 72], do not accomplish this as they aggregate (or collapse) layers into a simple graph in different ways. More importantly, aggregation approaches are likely to result in some information loss [51], distortion of properties [51], or hide the effect of different entity types and/or different intra- or inter-layer relationships as elaborated in [34]. Structure-preservation is critical for understanding a HeMLN community and for drill-down analysis of communities.

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

Figures 6 a) and b) illustrate the difference between the current approaches and our proposed approach. Figure 6 a) shows type-independent aggregation[1] of two layers into a single graph on which extant community detection is applied. As can be seen, **both structure as well as entity and relationship labels – shown as colored nodes and edges – are lost in the resulting communities.** In contrast, the Figure 6 b) shows the same layers and community detection using the proposed definition and the decoupling approach. As there is no aggregation, both structure and semantics are preserved.

[1] Other aggregation approaches have the same problem.

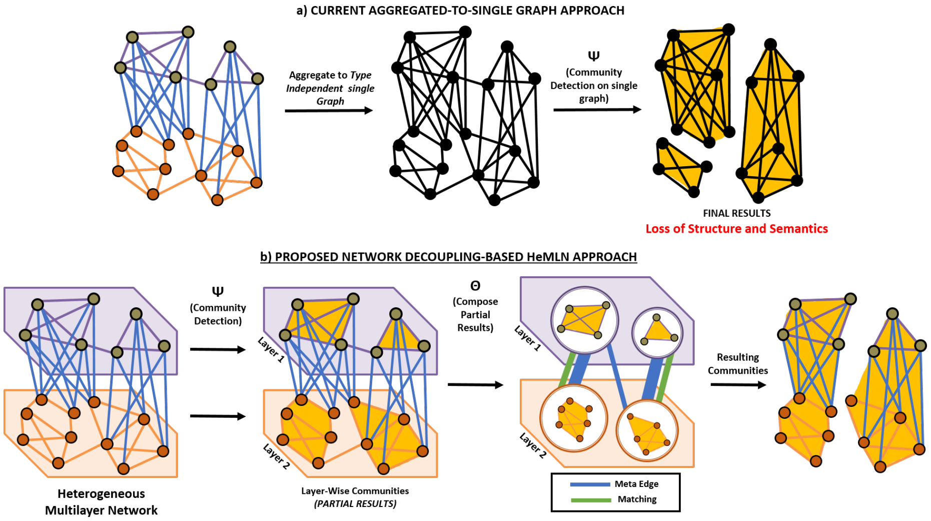
a) CURRENT AGGREGATED-TO-SINGLE GRAPH APPROACH
Aggregate to *Type Independent single Graph*
Ψ
(Community Detection on single graph)
FINAL RESULTS
Loss of Structure and Semantics
b) PROPOSED NETWORK DECOUPLING-BASED HeMLN APPROACH
Ψ
(Community Detection)
Layer 1
Layer 2
Θ
(Compose Partial Results)
Layer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

Fig. 6. Lossy Traditional Approach Vs. Structure- and Semantics-Preserving Decoupling Approach

3.3 Our Contributions Towards MLN modeling and Analysis

Instead of modeling complex data as a monoplex, we have proposed multiplexes [51, 63], In a multiplex, instead of creating a single graph with colored nodes and/or edges, a number of graphs are created, each representing one aspect or feature or perspective. For example, Figure 3 shows both a homogeneous and a heterogeneous multiplex. In each layer, actors are nodes and one genre is used to create edges. We believe this model is better than a monoplex as the graph in each layer is smaller, will not have multiple edges, and easier to understand semantically.

Efficient Analysis Of Multiplexes: Our ongoing research towards big data analytics includes modeling and efficient analysis of multiplexes. Applying community and hub detection algorithms for a multiplex-as-a-whole is being explored by the research community [70, 79]. This, in our view, is not the best way as the complexity of community and hub detection increases and the decomposition of the problem into its components (layers) is not leveraged.

For analyzing multiplexes in a holistic and flexible way, we propose **decoupling** as the approach of choice. The basic idea of decoupling is to analyze **individual layers** using extant algorithms and *compose* the results of individual layers to obtain results for any combinations of layers in a loss-less manner ([61, 62, 64–67]).

Figure 7(b) shows our decoupled approach to HeMLN community detection. The steps of our decoupling approach is given below:

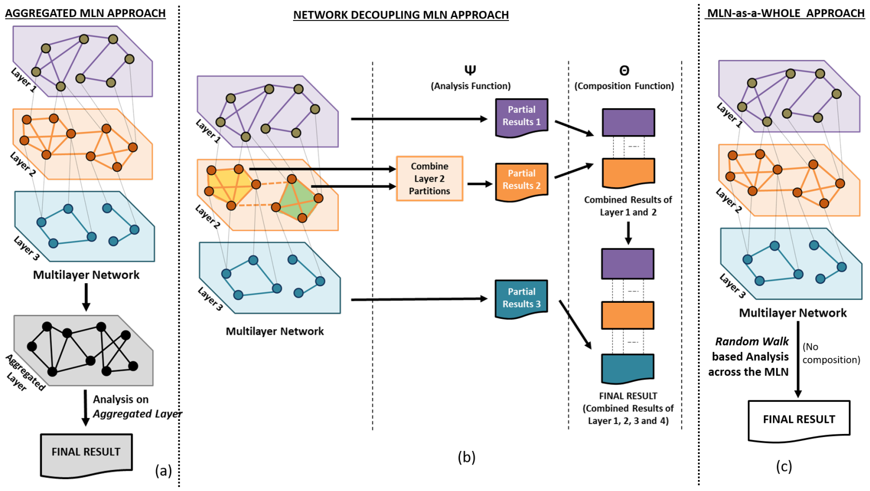

Fig. 7. Lossy Traditional Approach Vs. Structure- and Semantics-Preserving Decoupling Approach

(i) First use the function Ψ (e.g., community detection, but any network property in general) to find the property in each of the layers individually (can also be done in parallel),

(ii) for any two chosen layers, compose the partial results from each layer using the identified composition function Θ. For example, we have proposed vertex- and edge-based composition for Boolean composition of Homogeneous MLNs and a bipartite-graph and maximal matching for Hetereogeneous MLNs.

(iii) repeat this process for computing the network property for given number of layers in the order specified.

In this tutorial, we will discuss the MLN approach to modeling and analysis and contrast it with traditional approaches for efficiency and scalability. We will discuss how the objectives proposed in Section 1 can be efficiently computed using the proposed approach. We will also co consider other analysis objectives on different data sets.

4 Conclusions

In this paper, we have demonstrated why a multilayer network is better both from a modeling as well as analysis perspective. Multilayer networks provide an effective way of compartmentalizing the information in the data set using analysis objectives. Further, when appropriate composition mechanisms are available or developed (e.g., decoupling approach demonstrated here), computational efficiency as well as analysis flexibility can be accomplished. We have also been able to fold content extraction [77] into this framework in our analysis of Facebook data. We are currently working on approaches to extend this decoupling technique for a variety of analysis, in addition to community and centrality (such as substructure, motifs, querying), with the goal of broader analysis capability.

Acknowledgment

We would like to thank Dr. Sanjukta Bhowmick on her collaboration with us on the multilayer network analysis.

References

1. Adaikkalavan, R., Chakravarthy, S.: Event Specification and Processing for Advanced Applications: Generalization and Formalization. In: DEXA. pp. 369–379 (2007)
2. Adaikkalavan, R., Chakravarthy, S.: Events must be complete in event processing! In: Proceedings, Annual ACM SIG Symposium On Applied Computing. pp. 1038–1039 (2008)
3. Aery, M., Chakravarthy, S.: eMailSift: Mining-Based Approaches to Email Classification. In: SIGIR. pp. 580–581 (2004)
4. Aery, M., Chakravarthy, S.: eMailSift: Email Classification Based on Structure and Content. In: ICDM. pp. 18–25 (2005)
5. Aery, M., Chakravarthy, S.: InfoSift: Adapting Graph Mining Techniques for Text Classification. In: FLAIRS Conference. pp. 277–282 (2005)
6. Agrawal, R., Imielinski, T., Swami, A.: Database mining: A performance perspective. IEEE Transactions on Knowledge and Data Engineering 5(6), 914–925 (December 1993)
7. Balachandran, R., Padmanabhan, S., Chakravarthy, S.: Enhanced DB-Subdue: Supporting Subtle Aspects of Graph Mining Using a Relational Approach. In: PAKDD. pp. 673–678 (2006)
8. Balachandran, R., Padmanabhan, S., Chakravarthy, S.: Enhanced DB-Subdue: Supporting Subtle Aspects of Graph Mining Using a Relational Approach. In: PAKDD. pp. 673–678 (2006)
9. Berenstein, A., Magarinos, M.P., Chernomoretz, A., Aguero, F.: A multilayer network approach for guiding drug repositioning in neglected diseases. PLOS (2016)
10. Bodra, J.: Processing Queries Over Partitioned Graph Databases: An Approach and it's Evaluation. Master's thesis, The University of Texas at Arlington (May 2016), http://itlab.uta.edu/students/alumni/MS/Jay_D_Bodra/JBod_MS2016.pdf
11. Bodra, J., Das, S., Santra, A., Chakravarthy, S.: Query processing on large graphs: Scalability through partitioning. In: Big Data Analytics and Knowledge Discovery - 20th International Conference, DaWaK 2018, Regensburg, Germany, September 3-6, 2018, Proceedings. pp. 271–288 (2018), https://doi.org/10.1007/978-3-319-98539-8_21
12. Chakravarthy, S., Aved, A., Shirvani, S., Annappa, M., Blasch, E.: Adapting stream processing framework for video analysis. Procedia Computer Science 51, 2648–2657 (2015)
13. Chakravarthy, S.: Divide and Conquer: A Basis for Augmenting a Conventional Query Optimizer with Multiple Query Proceesing Capabilities. In: ICDE. pp. 482–490 (1991)

14. Chakravarthy, S., Adaikkalavan, R.: Ubiquitous Nature of Event-Driven Approaches: A Retrospective View. In: Chandy, M., Etzion, O., von Ammon, R. (eds.) Event Processing. No. 07191 in Dagstuhl Seminar Proceedings, Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, Dagstuhl, Germany (2007), http://drops.dagstuhl.de/opus/volltexte/2007/1150
15. Chakravarthy, S., Adaikkalavan, R.: Event and Streams: Harnessing and Unleashing Their Synergy. In: International Conference on Distributed Event-based Systems. pp. 1–12 (July 2008)
16. Chakravarthy, S., Beera, R., Balachandran, R.: DB-Subdue: Database Approach to Graph Mining. In: PAKDD. pp. 341–350 (2004)
17. Chakravarthy, S., Beera, R., Balachandran, R.: DB-Subdue: Database Approach to Graph Mining. In: PAKDD. pp. 341–350 (2004)
18. Chakravarthy, S., Jiang, Q.: Stream Data Management: A Quality of Service Perspective. Springer (April 2009)
19. Chakravarthy, S., Pajjuri, V.: Scheduling Strategies and Their Evaluation in a Data Stream Management System. In: BNCOD. pp. 220–231 (2006)
20. Chakravarthy, S., Sanka, A., Jacob, J., Pandrangi, N.: A learning-based approach for fetching pages in WebVigiL. In: Proceedings, Annual ACM SIG Symposium On Applied Computing. pp. 1725–1731 (2004)
21. Chakravarthy, S., Venkatachalam, A., Telang, A.: A graph-based approach for multi-folder email classification. In: ICDM. pp. 78–87 (2010)
22. Chakravarthy, S., Zhang, H.: Visualization of Association Rules Over Relational DBMSs. In: Proceedings, Annual ACM SIG Symposium On Applied Computing. pp. 922–926 (2003)
23. Chakravarthy, U.S., Grant, J., Minker, J.: Foundations of semantic query optimization for deductive databases. In: Foundations of Deductive Databases and Logic Programming., pp. 243–273. Morgan Kaufmann (1988)
24. Chakravarthy, U.S., Grant, J., Minker, J.: Logic-Based Approach to Semantic Query Optimization. ACM Trans. Database Syst. 15(2), 162–207 (1990)
25. Chakravarthy, U.S., Minker, J.: Multiple Query Processing in Deductive Databases using Query Graphs. In: VLDB. pp. 384–391 (1986)
26. Chakravarthy, U.S., Minker, J., Grant, J.: Semantic Query Optimization: Additional Constraints and Control Strategies. In: Expert Database Conf. pp. 345–379 (1986)
27. Chamakura, S., Sachde, A., Chakravarthy, S., Arora, A.: WEBVIGIL: Monitoring Multiple Web Pages and Presentation of XML Pages. In: ICDE Workshops. p. 1276 (2006)
28. Cook, D.J., Holder, L.B.: Substructure discovery using minimum description length and background knowledge. Journal of Artificial Intelligence Research 1, 231–255 (1994)
29. Cuzzocrea, A., Chakravarthy, S.: Event-based compression and mining of data streams. In: Proceedings of the 12th international conference on Knowledge-Based Intelligent Information and Engineering Systems, Part II. pp. 670–681. KES '08, Springer-Verlag, Berlin, Heidelberg (2008), http://dx.doi.org/10.1007/978-3-540-85565-1_83
30. Das, S.: Divide and Conquer Approach to Scalable Substructure Discovery: Partitioning Schemes, Algorithms, Optimization and Performance Analysis Using Map/Reduce Paradigm. Ph.D. thesis, The University of Texas at Arlington (May 2017), http://itlab.uta.edu/students/alumni/PhD/Soumyava_Das/SDas_PhD2017.pdf

31. Das, S., Chakravarthy, S.: Partition and conquer: Map/reduce way of substructure discovery. In: International Conference on Big Data Analytics and Knowledge Discovery. pp. 365–378. Springer (2015)
32. Das, S., Chakravarthy, S.: Duplicate reduction in graph mining: Approaches, analysis, and evaluation. IEEE Trans. Knowl. Data Eng. 30(8), 1454–1466 (2018), https://doi.org/10.1109/TKDE.2018.2795003
33. Das, S., Goyal, A., Chakravarthy, S.: Plan before you execute: A cost-based query optimizer for attributed graph databases. In: DaWaK 2016, Porto, Portugal, September 6-8, 2016. pp. 314–328 (2016)
34. De Domenico, M., Solé-Ribalta, A., Gómez, S., Arenas, A.: Navigability of interconnected networks under random failures. Proceedings of the National Academy of Sciences (2014), https://www.pnas.org/content/early/2014/05/21/1318469111
35. Domenico, M.D., Nicosia, V., Arenas, A., Latora, V.: Layer aggregation and reducibility of multilayer interconnected networks. CoRR abs/1405.0425 (2014), http://arxiv.org/abs/1405.0425
36. Dudgikar, M., Chakravarthy, S., Liuzzi, R.A., Wong, L.: A Layered Optimizer for Mining Association Rules Over Relational Database Management Systems. In: IKE. pp. 422–430 (2003)
37. Elkhalifa, L., Adaikkalavan, R., Chakravarthy, S.: InfoFilter: a system for expressive pattern specification and detection over text streams. In: Proceedings, Annual ACM SIG Symposium On Applied Computing. pp. 1084–1088 (2005)
38. Eppili, A., Jacob, J., Sachde, A., Chakravarthy, S.: Expressive Profile Specification and Its Semantics for a Web Monitoring System. In: ER. pp. 420–433 (2004)
39. Garg, V., Adaikkalavan, R., Chakravarthy, S.: Extensions to Stream Processing Architecture for Supporting Event Processing. In: DEXA. pp. 945–955 (2006)
40. Gilani, A., Sonune, S., Kendai, B., Chakravarthy, S.: The Anatomy of a Stream Processing System. In: BNCOD. pp. 232–239 (2006)
41. Goyal, A.: QP-SUBDUE: Processing Queries Over Graph Databases. Master's thesis, The University of Texas at Arlington (December 2015), http://itlab.uta.edu/students/alumni/MS/Ankur_Goyal/AGoy_MS2015.pdf
42. Hong, D., Chakravarthy, S., Johnson, T.: Locking Based Concurrency Control for Integrated Real-Time Database Systems. In: Proceedings, International Workshop on Real-Time Databases (RTDB). pp. 138–143 (1996)
43. Hong, D., Johnson, T., Chakravarthy, S.: Real-Time Transaction Scheduling: A Cost Conscious Approach. In: Proceedings, International Conference on Management of Data (SIGMOD). pp. 197–206 (1993)
44. Hong, D.K., Kim, M.J., Chakravarthy, S.: Incorporating load factor into the scheduling of soft real-time transactions for main memory databases. In: Proceedings, IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA). pp. 60–66 (1996)
45. Jacob, J., Sachde, A., Chakravarthy, S.: CX-DIFF: A Change Detection Algorithm for XML Content and Change Presentation Issues for WebVigiL. In: ER (Workshops). pp. 273–284 (2003)
46. Jiang, Q., Adaikkalavan, R., Chakravarthy, S.: MavEStream: Synergistic Integration of Stream and Event Processing. In: International Conference on Digital Communications. pp. 29–29 (2007)
47. Jiang, Q., Chakravarthy, S.: Scheduling Strategies for Processing Continuous Queries over Streams. In: Proceedings of the Annual British National Conference on Databases. pp. 16–30 (2004)
48. Jiang, Q., Chakravarthy, S.: Anatomy of a Data Stream Management System. In: ADBIS Research Communications. vol. 215, pp. 654–655 (2006)

49. Kendai, B., Chakravarthy, S.: Load Shedding in MavStream: Analysis, Implementation, and Evaluation. In: Proceedings, International British National Conference on Databases (BNCOD). pp. 100–112 (2008)
50. Kendai, B., Chakravarthy, S.: Runtime optimization of continuous queries. In: COMAD. pp. 104–115 (2008)
51. Kivelä, M., Arenas, A., Barthelemy, M., Gleeson, J.P., Moreno, Y., Porter, M.A.: Multilayer networks. CoRR abs/1309.7233 (2013), http://arxiv.org/abs/1309.7233
52. Kona, H., Chakravarthy, S.: An SQL-based Approach to Incremental Association Rule Mining. in special issue of the Foundations of Computing and Decision Sciences Journal (2006)
53. Kona, H., Chakravarthy, S.: Partitioned Approach to Association Rule Mining over Multiple Databases. In: DaWaK. pp. 320–330 (2004)
54. Mishra, P., Chakravarthy, S.: Performance Evaluation and Analysis of K-Way Join Variants for Association Rule Mining. In: BNCOD. pp. 95–114 (2003)
55. Mishra, P., Chakravarthy, S.: Performance evaluation of sql-or variants for association rule mining. In: DaWaK. pp. 288–298 (2003)
56. Mishra, P., Chakravarthy, S.: Performance evaluation of sql-or variants for association rule mining. In: DaWaK. pp. 288–298 (2003)
57. Mishra, P.: Performance Evaluation and Analysis of SQL-based Approaches for Association Rule Mining. Master's thesis, The University of Texas at Arlington (December 2002)
58. Padmanabhan, S.: HDB-Subdue: A Relational Database Approach to Graph Mining and Hierarchical Reduction. Master's thesis, The University of Texas at Arlington (December 2005)
59. Pandrangi, N., Jacob, J., Sanka, A., Chakravarthy, S.: WebVigiL: User Profile-Based Change Detection for HTML/XML Documents. In: BNCOD. pp. 38–57 (2003)
60. Rosenthal, A., Chakravarthy, U.S.: Anatomy of a Mudular Multiple Query Optimizer. In: VLDB. pp. 230–239 (1988)
61. Santra, A., Bhowmick, S., Chakravarthy, S.: Efficient community re-creation in multilayer networks using boolean operations. In: Int. Conf. on Computational Science, Zurich, Switzerland. pp. 58–67 (2017), https://doi.org/10.1016/j.procs.2017.05.246
62. Santra, A., Bhowmick, S., Chakravarthy, S.: Hubify: Efficient estimation of central entities across multiplex layer compositions. In: IEEE International Conference on Data Mining Workshops (2017)
63. Santra, A., Bhowmick, S.: Holistic analysis of multi-source, multi-feature data: Modeling and computation challenges. In: Big Data Analytics - Fifth International Conference, BDA 2017 (2017)
64. Santra, A., Bhowmick, S., Chakravarthy, S.: Efficient community detection in boolean composed multiplex networks. University of Texas at Arlington June (2019), http://itlab.uta.edu/research/current/Multi%20Source%20Data%20Analysis/ArXiv2019-HoMLN-Final.pdf
65. Santra, A., Komar, K.S., Bhowmick, S., Chakravarthy, S.: An efficient framework for computing structure and semantics-preserving community in a heterogeneous multilayer network. University of Texas at Arlington June (2019), http://itlab.uta.edu/research/current/Multi%20Source%20Data%20Analysis/ArXiv2019-HeMLN-Final.pdf

66. Santra, A., Komar, K.S., Bhowmick, S., Chakravarthy, S.: Making a case for mlns for data-driven analysis: Modeling, efficiency, and versatility. University of Texas at Arlington August (2019), http://itlab.uta.edu/research/current/Multi%20Source%20Data%20Analysis/BigData_2019-final.pdf
67. Santra, A., Komar, K.S., Bhowmick, S., Chakravarthy, S.: Structure-preserving community in a multilayer network: Definition, detection, and analysis. arXiv preprint arXiv:1903.02641 (2019)
68. Savla, S., Chakravarthy, S.: A Single Pass Algorithm for Detecting Significant intervals in Time-Series Data. In: ADMKD. pp. 49–60 (2006)
69. Savla, S., Chakravarthy, S.: An Efficient Single Pass Approach to Frequent Episode Discovery in Sequence Data. In: International Conference on Intelligent Environments (IE08) (2008)
70. Solé-Ribalta, A., De Domenico, M., Gómez, S., Arenas, A.: Centrality rankings in multiplex networks. In: Proceedings of the 2014 ACM conference on Web science. pp. 149–155. ACM (2014)
71. Srinivasan, A., Bhatia, D., Chakravarthy, S.: Discovery of interesting episodes in sequence data. In: Proceedings, Annual ACM SIG Symposium On Applied Computing. pp. 598–602 (2006)
72. Sun, Y., Han, J.: Mining heterogeneous information networks: a structural analysis approach. ACM SIGKDD Explorations Newsletter 14(2), 20–28 (2013)
73. Telang, A., Mishra, R., Chakravarthy, S.: Ranking Issues for Information Integration. In: ICDE Workshops. pp. 257–260 (2007)
74. Thomas, S.: Architectures and Optimizations for Integrating Data Mining Algorithms with Database Systems. Ph.D. thesis, The University of Florida at Gainesville (December 1998)
75. Thomas, S., Chakravarthy, S.: Performance Evaluation and Optimization of Join Queries for Association Rule Mining. In: DaWaK. pp. 241–250 (1999)
76. Thomas, S., Chakravarthy, S.: Incremental Mining of Constrained Associations. In: HiPC. pp. 547–558 (2000)
77. Vu, X.S., Santra, A., Chakravarthy, S., Jiang, L.: Generic multilayer network data analysis with the fusion of content and structure. In: CICLing 2019, La Rochelle, France (2019)
78. Wilson, J.D., Palowitch, J., Bhamidi, S., Nobel, A.B.: Community extraction in multilayer networks with heterogeneous community structure. J. Mach. Learn. Res. 18(1), 5458–5506 (2017), http://dl.acm.org/citation.cfm?id=3122009.3208030
79. Zhang, H., Wang, C.D., Lai, J.H., Philip, S.Y.: Modularity in complex multilayer networks with multiple aspects: a static perspective. In: Applied Informatics. vol. 4, p. 7. Springer Berlin Heidelberg (2017)