EER→MLN: EER Approach for Modeling, Mapping, and Analyzing Complex Data Using Multilayer Networks (MLNs)

Kanthi Sannappa Komar[1], Abhishek Santra[1], Sanjukta Bhowmick[2], and Sharma Chakravarthy[1(✉)]

[1] Information Technology Laboratory and CSE Department, University of Texas at Arlington, Arlington, TX, USA
sharmac@cse.uta.edu

[2] Department of Computer Science, University of North Texas, Denton, TX, USA

Abstract. Extended Entity Relationship (or EER) modeling is an important step after application requirements for data analysis are gathered, and is critical for translating user requirements to a given executable data model (e.g., relational, or for this paper Multilayer Networks or MLNs.) EER modeling provides a more precise understanding of the application and data requirements and an unambiguous representation from which the data model (on which analysis is performed) can be generated algorithmically. EER has played a central role in the modeling of user-level requirements to relational, object oriented etc. UML, whose roots are in EER modeling, is extensively used in the industry.

Although big data analysis has warranted many new data models, not much attention has been paid to their modeling from requirements. Going straight from application requirements to data model and analysis, especially for complex data sets, is likely to be difficult, error prone, and not extensible to say the least. Hence for data models used in big data analysis, such as Multilayer Networks, there is a need to transform the user/application requirements using a modeling approach such as EER.

In this paper, we start with application requirements of complex data sets including analysis objectives and show how the EER approach can be leveraged for modeling given data to generate MLNs and appropriate analysis expressions on them. This is timely as MLNs are gaining popularity (and also subsume graphs) as a meaningful data representation for big data analysis.

For demonstrating the algorithm and applicability of the proposed approach, we demonstrate our approach on three data sets to generate MLNs, to map analysis requirements into expressions on MLNs. We also demonstrate it for three types of MLNs. The data sets are from DBLP (Database Bibliography-Computer Science Publications), IMDb, a large international movie data set, and US commercial airlines. Our experimental analysis validate modeling and mapping. We do not elaborate on

G. Dobbie et al. (Eds.): ER 2020, LNCS 12400, pp. 555–572, 2020.
https://doi.org/10.1007/978-3-030-62522-1_41

computations as it is a separate topic in itself. The correctness of results are verified using independently available ground truth.

Keywords: Multilayer networks · Network decoupling · EER Modeling

1 Introduction

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

In this paper, our focus is on data sets with diverse types of entities that are defined by multiple features and interact through varied and complex relationships. Although graph modeling is used, the analysis and computations are different from the ones addressed in either RDBMSs or recent NoSQL systems, such as Neo4J. Instead of a database, the data is transformed into MLN data structures using EER modeling and computations are performed on these using packages and libraries that are available. Just to give an idea, an analysis may need community detection, degree-centrality (or hubs) detection, and combine layers using Boolean operator (AND, OR, and NOT) or use weighted bipartite graph matching. We will not go into the details of the operators and computations as this paper is about modeling and mapping of analysis into appropriate computations. However, we present some results to convince the reader that this workflow has been completely defined.

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

1.1 Data Set Descriptions with Analysis Objectives

We have chosen *three data sets for modeling and analysis* from different application domains to illustrate the broader applicability of our proposed framework. While larger data sets can be used, we selected these as reliable ground truth data from orthogonal sources were available. Although we have indicated many

analysis objectives to show the capability of this approach, due to space constraints, we show a subset of them *in the experimental analysis section*. However, all of them have been computed.

1. Internet Movie Database (IMDb): This data set is publicly available and stores information about movies, TV episodes, actor, directors, ratings and genres of the movies, etc. [2]. Here the entities are of different types as they can be actors, directors, movies, etc. The features/relationships can be co-actors, similar-genre-acting, directed-a-movie, same movie ratings etc.

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

(A1) Find co-actor groups that are *most popular* and *most versatile*
(A2) *Cluster* groups of co-actors who have worked in movies with high ratings
(A3) *Predict* new groups of actors who have not worked together before, but benefit from working together in future.

2. Database Bibliography (DBLP): As most researchers are familiar with, the DBLP dataset is publicly available and stores information about computer science publications in various conferences and journals. It captures the author names and institutions, years, conference/journal names and links to the papers [1]. Clearly, there are multiple entities that can be related based of different types of relationships.

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

(A4) Find *strongest* co-author groups who have collaborated on at least 3 papers
(A5) For each conference, find *most popular groups* of co-authors who publish frequently
(A6) For the *most popular* collaborators in each conference, find the 3-year period(s) when they were *most active*
(A7) For each conference that publishes maximum papers in each period, find the *most popular paper review score*.

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

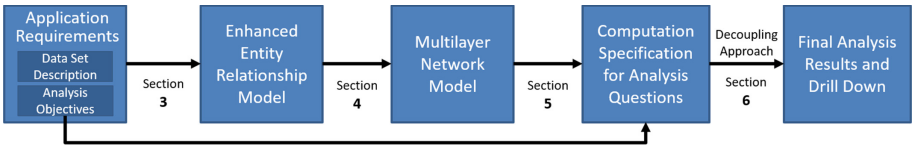

Fig. 1. EER→MLN Flow Chart: Application Requirements to Analysis and Drill Down

Analysis Objectives. For such a diverse data set, the analysis objectives can get complex. For example,

(A8) Find *strong* co-author groups who are also friends on Facebook (if Facebook information is available)
(A9) Find cities where the largest concentrations of authors reside
(A10) What is a good city to hold conferences of authors to maximize attendance?

We have selected the analysis objectives to be varied for the purpose of illustrating the need and effectiveness of the approach being proposed. They range from relatively easy analysis of finding clusters/communities of co-actors to more complicated predictions of potential future teaming of actors and potential city for holding a conference. All objectives have been computed even though we show only a subset in this paper.

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

Figure 1 shows the flow and contributions of the paper. Section 2 has related work. Section 3 shows mapping of user requirements to an EER diagram using the standard notations. In Sect. 4 we discuss the mapping of the EER diagram into homogeneous, heterogeneous, and hybrid MLNs using the proposed algorithm. In Sect. 5.1, we briefly introduce the *decoupling* approach used for big data analysis. In Sect. 5.2 we demonstrate with examples how the *analysis objectives are mapped* to expressions on the generated MLNs. In Sect. 6, we compute the expressions and *validate* our results independent *orthogonal* sources.

2 Related Work

ER and EER models have served as a methodology for database design by representing important semantic information about the real world [6] application. Relational database modeling has clearly benefited from this body of work and has motivated UML for OO design. A good EER diagram based on the user analysis requirements is critical for an error-free relational database schema. Numerous tools have been developed for creating the EER diagram and algorithmically mapping it into relations for different commercial DBMSs.

However, with the emergence of structured data sets with inherent relationships among entities and complex application requirements, such as shortest paths, important neighborhoods, dominant nodes (or groups of nodes), etc., [7,12], the relational data model was not the best choice for modeling as well as analyzing them [5]. This led to the evolution of NoSQL data models including the

graph data model [3]. In many cases, like friendship (Facebook), collaborations (Movies) and follower-followee (Twitter) relationships, relationships needed to be modeled explicitly using graph model. This gave rise to computations over these data models. Recently, there has been some work in the area of graph modeling from EER diagrams, but is limited to simple attributed graphs only [6,17,18]. However, most of these works either do not handle recursive relationships [18], and weak entities [8] or are application-specific [11]. To the best of our knowledge, there has been no systematic approach to modeling MLNs using data sets and requirement objectives.

Multilayer networks were introduced when the data sets necessitated graph data models to capture multiple types of nodes and relationships, features and connections with a need to analyze the effect of different combinations of perspectives [14,23]. There is substantial work on analysing different types of multilayer networks based on meta-paths across graphs [23,25], community detection [13,15,19] and centrality measurements [20]. However, to the best of our knowledge *there is not any work that creates a EER model from the given requirements and provides an algorithm for the generation of a multilayer network given for the analysis of given objectives.*

3 Application Requirements to EER Model

Any analysis objective to be computed from data involving multiple entities, features and complex relationships has been shown to benefit from a multilayer network model [14]. EER diagrams [10] are well-established and have been used to model and design schema for relational databases. An EER diagram is crucial for creating a database that satisfies 2NF. In this section, we illustrate the first stage from Fig. 1 where requirements from three different sets of real-world analysis objectives (Sect. 1.1) are mapped to EER diagrams.

3.1 Internet Movie Database (IMDb) Analysis

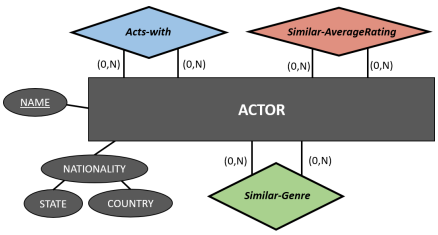

Fig. 2. IMDb EER Diagram

The data set consists of top 500 actors and their co-actors across all movies, giving a total of 9000+ actors. Based on the information in the IMDb data set **and** analysis objectives (**A1-A3**), one can build an EER diagram (shown in Fig. 2) as described below[1]:

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

[1] Note that the relationship details can change based on analysis objectives.

– **Recursive Relationships**:

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

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

3.2 Database Bibliography (DBLP) Analysis

For DBLP, we have considered all publications from VLDB, SIGMOD, ICDM, KDD, DaWaK and DASFAA from the 2001–2018. Based on data set description and analysis objectives (**A4**–**A7**), the EER diagram shown in Fig. 3 has been discussed below

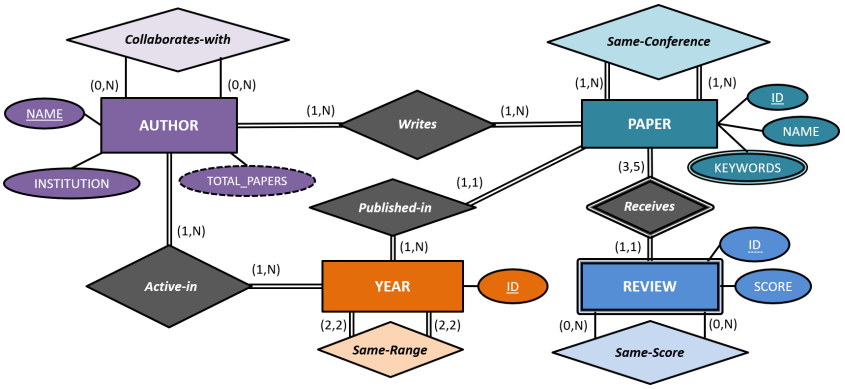

Fig. 3. DBLP EER Diagram

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

– **Entities including Weak:**

- *Author* with attributes - name (key) and institution. *Total_Papers* is a derived attribute that can be calculated using *writes* binary relationship.
- *Paper* with attributes, Paper ID (key), name and keywords (multi-valued)
- *Year* with year ID as the key attribute
- *Review:* Existence of a review is dependent on the existence of a paper, thus it is a *weak entity*. It has ID (partial key) and score as the two attributes.

– **Recursive Relationships:**

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

– **Binary Relationships:**

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

– **(Min, Max) Cardinality Ratios:**

- *Collaborates-with* recursive relationship has cardinality ratio as *(0,N)..(0,N)* as each author can work individually or with any number of authors. *Same-Conference* has cardinality *(1,N)..(1,N)* as many papers are published in the same conference, thus a paper is related to at least one paper. Cardinality of *Same-Range* is *(2,2)..(2,2)* as each year is related to the other 2 years in the 3-year period. *Same-Score* has *(0,N)..(0,N)* cardinality as a review may not be related to any other review.
- Binary relationship *Writes* between author and paper entity has *(1,N)..(1,N)* cardinality as an author can publish one or more papers and also paper can have one or more authors. Similarly, *Active-in* has *(1,N)..(1,N)* cardinality as an author is active in at least one year and in a given year many authors can be active. The *Published-in* relationship has *(1,1)..(1,N)* cardinality as paper is published only in one year but many papers can be published in a year. Finally, for *Receives* the cardinality is *(3,5)..(1,1)* as every paper receives 3 to 5 reviews, however each review is for exactly one paper.

3.3 Author-City Data Set Analysis

For final set of analysis objectives (**A8**–**A10**) based on author-city data set, the EER diagram shown in Fig. 4 has been discussed below

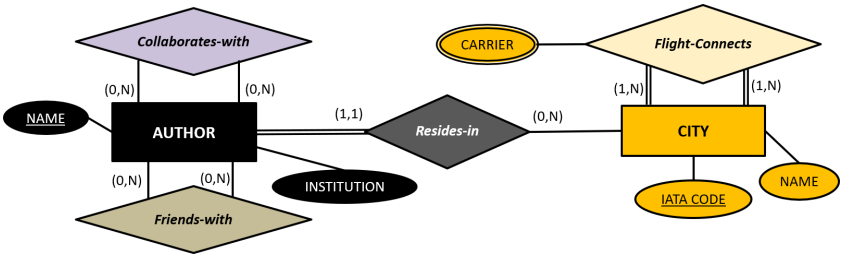

Fig. 4. Author-City EER Diagram

– **Entities:**

- *Author* with attributes - name (key) and institution
- *City* with attributes - IATA/Airport Code (key) and name

– **Recursive Relationships:**

- *Collaborates-with*: Two authors are related if they have worked together on at least 3 published papers.
- *Friends-with*: A relationship to signify if two authors are friends on Facebook.
- Flight-connects: Two cities are related if there is a flight connecting them with a multi-valued attribute to capture the operating *carriers*.

– **Binary Relationships:** A binary relationship, *Resides-in* exists between the author and city entity depicting the residence.

– **(Min, Max) Cardinality Ratios:**

- *Collaborates-with* and *Friends-with* recursive relationships have *(0,N)..(0,N)* cardinality, as an author may work individually and may not be friends with anyone on Facebook, respectively.
- Binary relationship *Resides-in* between author and city entity has *(1,1)..(0,N)* cardinality as an author can reside in only one city. However, a city may not be any author's residence or multiple authors can reside in it.

4 Generating MLNs from an EER Diagram (EER→MLN)

We provide an overview of the MLN models before discussing our algorithm to convert an EER to MLN (Sect. 4.2 and 4.3). We use the algorithm to translate EER diagrams discussed in Sect. 3.1, 3.2 and 3.3.

4.1 Multilayer Networks: An Overview of the Data Model

Multi-feature data comprises of multiple relationships among the same or different types of entities. Relationships among the entities can either be specified by explicit interactions (like flights, co-authors, and friends) or based on a similarity metric depending on the type of the feature like nominal, numeric, time, date, latitude-longitude values, text, audio, video or image. For flexible, loss-less, structure-preserving and efficient analysis of such data sets based on different combinations of features, multilayer networks (or MLNs) have been proposed in the literature to be an ideal choice [14,21].

A multilayer network model is a *network of networks*. In this case, every layer represents a distinct relationship among entities with respect to a single feature. The sets of entities across layers, which may or may not be of the same type, can be related to each other too. Formally, a **multilayer network**, $MLN(G, X)$, is defined by two sets of graphs: i) The set $G = \{G_1, G_2, \ldots, G_N\}$ contains graphs of N individual layers, where $G_i(V_i, E_i)$ is defined by a set of vertices, V_i and a set of edges, E_i. An edge $e(v, u) \in E_i$, connects vertices v and u, where $v, u \in V_i$ and ii) A set $X = \{X_{1,2}, X_{1,3}, \ldots, X_{N-1,N}\}$ consists of bipartite graphs. Each graph $X_{i,j}(V_i, V_j, L_{i,j})$ is defined by two sets of vertices V_i and V_j, and a set of edges (also called links or inter-layer edges) $L_{i,j}$, such that for every link $l(a, b) \in L_{i,j}$, $a \in V_i$ and $b \in V_j$, where V_i (V_j) is the vertex set of graph G_i (G_j.)

Based on the type of relationships and entities, multilayer network are of different types. Layers of a **homogeneous MLN (or HoMLN)** are used to model the diverse relationships that exist among the **same type of entities** like movie actors who are linked based on if they act together or have similar average rating. Thus, $V_1 = V_2 = \ldots = V_n$ and inter-layer edge sets are empty as no relations across layers are necessary. Relationships among **different types of entities** like researchers (connected by co-authorship), research papers (connected if published in same conference) and year (related by pre-defined ranges/eras) are modeled through **heterogeneous MLN (or HeMLN).** The inter-layer edges represent the relationship across layers like writes, published-in and active-in. In addition to being collaborators, researchers may be Facebook friends. Thus, to model multi-feature data that capture **multiple relationships within and across different types of entity sets**, a combination of homogeneous and heterogeneous MLNs is used, called **hybrid MLN (or HyMLN).** Figure 5 shows an example of each generated from the algorithm below.

4.2 Algorithmic Steps for Translating an EER Diagram to MLNs

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

A layer consists of nodes with a node id which is unique and a node label which need not be unique. An edge consists of an edge label which is not unique

and connects two node ids. Typically, node ids are kept unique for the purposes of computation. Below, we assume node ids are generated as part of the translation process. In this paper, we do not show how additional information of nodes and edges that come out of the EER diagram are kept. They cab ne maintained as .csv files (or translated into relations) to be used for drill down analysis of results. EER model also helps in modeling only those attributes of nodes and edges that are relevant to the analysis objectives and drill down.

1. **Each binary relationship** in the EER diagram corresponds to either an individual layer or a bipartite graph (of inter-layer edges) between two layers. Typically, entity id is used as the label of nodes in the layer. Other attributes are not typically stored as part of MLN (to reduce storage), but are stored separately (for example, as a relation or as a .csv file) for drill-down of the results later. The relationship name is used as intra- or inter-edge label and again, other relationship attributes are stored separately for drill down of results. We show some drill down results in Sect. 6.
   For example, the relationship Acts-with in Fig. 2 is translated into a layer Actor with name as node label and acts-with as edge label. In contrast, the relationship writes in Fig. 3 becomes a bipartite graph between the layers Paper and Author.
2. **Each binary recursive relationship** translates to a separate homogeneous layer whose intra-layer connectivity is defined by the relationship.
   For example, the layer Actor(Acts-with) in Fig. 5 (a) is obtained by the binary recursive relationship Acts-with in Fig. 2 on the Actor entity.
3. **Each binary non-recursive relationship** translates to a bipartite graph between the layers corresponding to entities of the relationship.This assumes that the layers have been formed earlier by binary recursive relationships.
   For example, Author-Year inter-layer edges in Fig. 2 (b) are formed by the relationship active-in in Fig. 3 between Author and Year entities.
4. **Translation of the attributes** (of an entity or a relationship) other than the key is done in the same way as we do for a relational model. Atomic, component, and multi-valued attributes are handled in the same manner. Derived attributes are not stored but are computed.
5. Hence, relationships have to be translated **in a specific order**: binary recursive first, followed by binary non-recursive relationships.
6. **Super and Sub entities** can be present in the EER diagram. If an entity type is a **super class**, either a layer can be created for it or layers can be created for each of its sub-class entity types depending on characteristics such as disjoint, overlapping, partial and total. This is quite similar to the translation to the relational model. Relationships present on these entities dictate the translation. Mapping of the relationships will follow the above steps.
   For example, it is possible that the super class may become a separate layer for some analysis objectives and sub classes may become separate layers for other analysis objectives. Different MLNs can be created from the EER diagram to meet the analysis objectives. Person as a super entity may have overlapping

sub entities actors and directors. If there are separate recursive relationships for the Person entity, it will become a separate layer.

7. A **weak entity** and its non-recursive binary relationship is translated as follows. Unlike how it is done for the relational model, a **weak entity** is translated into a separate layer (using a binary recursive relationship on that entity) and the weak relationship is translated into a bipartite graph with edge labels indicating the dependence (combining the primary and the partial key).
   For example: The Review weak entity in Fig. 3 becomes a separate layer in addition to the Layer Paper (Fig. 5 (b)). The intra-layer edges are dictated by the Same-Score recursive relationship. This layer has a bipartite graph with the Paper layer with the inter-layer edge labels corresponding to the Paper ID and Review ID.
8. Currently, **n-ary relationships** that **cannot** be mapped to multiple binary relationships are not supported. If they can be mapped to multiple binary relationships, the above steps handle them. If not, such a relationship involves handling a **hyper-edge** across multiple layers of a MLN which is beyond the scope of this paper.

4.3 Summary of the Algorithm

The above algorithmic steps when applied translates an EER diagram to a MLN(s) along with drill down information in a form that is queryable and searchable. Below we make a few comments on the overall translation of the EER diagram. Note that the **same** EER diagram is used for generating relations or .csv files for drill down thereby enhancing the use of the EER modeling.

- Each entity with **multiple** binary recursive relationships gives rise to a **Homogeneous MLN.**
- **Multiple entites** with *both* binary recursive (one each) and binary non-recursive relationships give rise to a **Heterogeneous MLN**.
- If the EER diagram has both kinds of entities and relationships as indicated above (as in 4) and there is at least one relationship between entities that form the homogeneous and heterogeneous layers, a **Hybrid MLN** is obtained.
- **Strong entities** as well as **weak entities** are translated as described above and become separate layers.
- The **min-max cardinality information** will give an insight into the minimum and maximum associations (or edges) that a node can have. This can help to calculate the *minimum, maximum and average degree* of the corresponding layer or bipartite graph.
- A **partial participation of an entity** translates to a node that is not connected to any other node (i.e., no intra- or inter-edge). *For example, the author can work individually (Partial Collaborates-with relationship).* Whereas a **total participation** implies every node has at least one edge.
- The **direction** of the inter or intra layer edges has to be implied from the semantics of the relationship. This can also be specified as part of the relationship. *For example, co-authorship will be bi-directional, whereas a relationship*

like follows-on-Twitter will be a directional. This is typically specified as part of the application requirement and can be incorporated into the EER model relatively easily as part of the relationship using the (min, max) cardinality information.

4.4 Application of the Above Algorithmmic Steps

For the 3 sets of analysis discussed in Sect. 3, the following MLNs, shown in Fig. 5, are generated by applying the above algorithmic steps. Node and edge labels have not been shown for simplicity.

IMDb Analysis: Based on the EER (Fig. 2), a **Homogeneous MLN** (Fig. 5 (a)) is obtained with 3 layers having every actor element as a separate node with intra-layer edges dictated by *Acts-with*, *Similar-AverageRating* and *Similar-Genre* recursive relationships (Using (**2**)). The node label is the actor name and intra-layer edge labels are the relationship names (Using (**1**)). Relationship semantics do not need a direction, thus edges are undirected.

DBLP Analysis: The EER in Fig. 3 gets translated into a **Heterogeneous MLN** (Fig. 5 (b)) with 4 layers - Author, Paper, Year and Review with intra-layer edges corresponding to *Collaborates-with*, *Same-Conference Same-Range* and *Same-Score* recursive relationships, respectively (Using (**2**), (**7**) for Weak Review Entity). The binary non-recursive relationships - *Writes, Active-in, Published-in, Reviews* generate 4 bipartite graphs between the layer pairs - Author-Paper, Author-Year, Paper-Year and Paper-Review, respectively (Using (**3**)). The node and edge labels are the key attributes and relationship names (Using (**1**), (**7**)). The relationships do not have an explicit requirement for direction, thus every intra/inter layer edge is *undirected.*

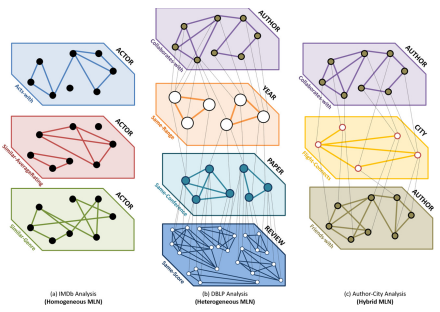

Fig. 5. MLN models for analysis set 1, 2 and 3

Author-City Analysis: The EER model in Fig. 4 leads to the generation of a **Hybrid MLN** (Fig. 5 (c)) with two Author Layers and a City Layer with intra-layer edges based on the *Collaborates-with*, *Friends-with* and *Flight-connects* recursive relationships (Using (**2**)). The binary non-recursive relationship *Resides-in* is used to introduce the inter-layer edges between the City layer and each of the Author layers (Using (**3**)). Node labels are name (Author layers) and IATA code (City layer), while the edge labels are relationship names (Using (**1**)). Collaboration, Residence and Friendship are bi-directional relationships. For the *Flight-connects* relationship it is assumed that if a flight exists from city a to city b, then a reverse flight also exists. Thus, every inter/intra layer edge is undirected in this HyMLN.

5 Analysis Objectives to Computation Specification

For the analysis of MLNs, a number of aggregate features are used for computation of objectives. They are: notions of community, centrality, and substructure. In this paper, we use community and centrality which are briefly summarized below.

Informally, a ***community*** is defined as a connected subgraph whose vertices are more connected to each other than to other vertices in the rest of the network (or layer). This objective is achieved by optimizing network parameters such as modularity or conductance in single layer graphs. Several algorithms are available for community detection of a graph [4]. A community is a weaker group of connected nodes than a clique. Since cliques of size greater than three are hard to find, community as a dense connected graph is used as a substitute.

Centrality Metrics are used for measuring the importance of vertices. They include degree centrality (number of neighbors), closeness centrality (mean distance of the vertex from other vertices), betweenness centrality (fraction of shortest paths passing through the vertex), and eigenvector centrality (the number of important neighbors of the vertex) [16].

It is also interesting to note that both community and centrality detection cannot be expressed in SQL (is an optimization problem).

5.1 Decoupling-Based Approach

Recently, a novel decoupling approach has been proposed for detecting communities and centralities in an efficient manner. This uses the equivalent of "divide and conquer" for MLNs [21,26].

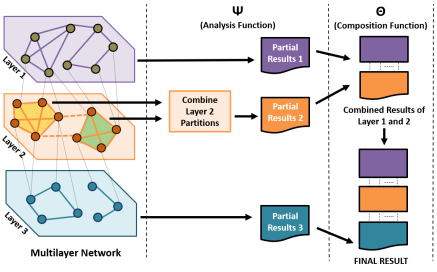

Fig. 6. Decoupling approach

Decoupling requires partitioning (derived from the MLN structure; individual layers as partitions - Fig. 6) and a way to compose partial (or intermediate) analysis results for community/centrality detection of MLNs. Substantial work has been done to identify the composition function (referred to as Θ, see Fig. 6) that is appropriate for efficient community/centrality detection (referred to as Ψ, see Fig. 6) on MLNs.

5.2 Computation Specification Mapping

Once the EER diagram is created based on the application requirements (data set description + analysis objectives) and translated into MLNs, the next step is to map each objective into an expression using Θ and Ψ on the MLNs generated. This step is relatively easier to identify once the operators to apply and the type

of composition to perform is determined, This step is similar to writing SQL queries once the specific database schema is generated and populated.

We show below how aggregate feature computation is specified along with composition to be used. The challenge in successfully applying network decoupling is to match the analysis function, Ψ and the composition function, Θ. Table 1 gives the mapping of each analysis objective **A1** to **A10** to their computation specification (in *left* to *right* order), analysis function (Ψ) and composition function (Θ). For few analysis, the composition is defined, for others we provide a short description of composition process.

Table 1. MLN expression for each analysis objective

Analysis	Mapping		
	Computation specification	Ψ	Θ
IMDb (**HoMLN**): 3 Actor Layers: *Acts-with, Similar-Genre, Similar-AverageRating*			
A1	*Acts-with* Θ *Similar-Genre*	Degree-Centrality	AND[20]
A2	*Acts-with* Θ *Similar-AverageRating*	Community	AND[19]
A3	NOT(*Acts-with*) Θ *Similar-Genre* Θ *Similar-AverageRating*	Community	AND[19]
DBLP (**HeMLN**): Author (Au), Year (Y), Paper (P), Review (R)			
A4	Au	Community	
A5	P Θ Au	Community	MWM[22]
A6	P Θ Au Θ Y	Community	MWM[22]
A7	Y Θ P Θ R	Community	MWM[22]
Set 3: Author-City (**HyMLN**): City (C) and 2 Au Layers - *Collaborates-with*, *Friends-with*			
A8	*Collaborates-with* Θ *Friends-with*	Community	AND[19]
A9	C Θ *Collaborates-with*; C Θ *Friends-with*	Centrality (Degree)	HeMLN-Centrality
A10	*Collaborates-with* Θ *Friends-with* Θ C	Community(Au), Degree-Centrality(C)	MLN-Searching

IMDb Analysis: For **A1** using network decoupling, we first find the *high degree* nodes in *Acts-with* and *Similar-Genre* layers, separately to detect the popular co-actors and versatile actors. Using the AND composition (details in [20]) we find all those *popular co-actors who are also highly versatile*. For **A2**, the AND composition is applied on the communities from the *Acts-with* and *Similar-AverageRating* layers to generate and filter out the groups of co-actors who have high ratings. In **A3** aim is to find actors who have not acted together but act in the same genre and in movies of similar ratings – which increases their possibility

of acting together in future. We apply the NOT operation on the Acts-with layer to find the complement graph of actors who have never acted together. In the first step of network decoupling, we take communities from each of the three layers; the Similar-Genre, Similar-AverageRating and the complement of the Acts-with layer. We then combine the resultant communities using the AND composition function to find *groups of actors who have a high chance of acting together in future.*

DBLP Analysis: For **A4**, the Author layer communities will give the desired result. For **A5**, **A6** and **A7** the communities from Author, Paper, Year and Review layer need to be paired up in the specified order to meet the analysis objectives. In [22], the HeMLN community detection has been proposed where for any two layers a bipartite graph is constructed using their communities. Each community is considered to be a meta-node. Two meta-nodes in two different layers are connected if there is at least one inter-layer edge between them. The weight of these edges (meta-edges) between the meta-nodes is given by the number of inter-layer edges between them. These meta nodes (communities) in the bipartite graph are uniquely paired using the composition function (Θ) Maximal Weighted Matching (MWM) that maximizes the overall meta-edge weight and is based on traditional matching proposed by Jack Edmonds [9]. For **A5**, the Author communities that get *matched* with Paper communities (corresponding to conferences) are the most popular. For **A6**, the matched Author communities from A5 are paired with Year communities to find their most active periods. For **A7**, first Paper communities are matched to Year communities to obtain the highly publishing conferences per period. Then, the matched Paper communities are matched to Review communities, to get the *most popular review score.*

Author-City Analysis: A8 is computed by the AND composition on the communities from two Homogeneous Author layers. For **A9**, the cities having high inter-layer degree with any one of the author layers are the *cities with high author concentrations.* In **A10**, ideally a conference will get more attendance if it is organized in a city that is a) well-connected via flights, b) where large co-author communities reside and c) large sections of those co-author groups are friends in order to maximize the advertisement of the conference. Thus, using the decoupling approach the communities from the two author layers and high degree nodes from the City layer are composed (and filtered) in order to obtain the desired set of *probable venues for a conference.*

6 Experimental Analysis

Using the decoupling approach, we executed the mapped computation specifications (Sect. 5.2) on the generated IMDb HoMLN, DBLP HeMLN and Author-City HyMLN (Fig. 5). Different parts of composition functions (Θ) have been implemented in C++ and Python 3.7.3 and executed on a quad-core 8^{th} generation Intel i7 processor machine with 8 GB RAM.

Due to space constraints, we are presenting few interesting results (and provide validation) from the HoMLN and HeMLN that have been built using real

Table 2. Left: IMDb HoMLN stats. **Right:** DBLP HeMLN stats

IMDb Actor	#Nodes	#Edges
Acts-with	9485	45,581
Similar-Genre	9485	996,527
Similar-AverageRating	9485	13,945,912

DBLP	#Nodes	#Edges
Author	16,918	2,483
Paper	10,326	12,044,080
Year	18	18

world data snapshots (Statistics shown in Table 2). *Also note that the drill down has been performed using the database created from the EER model while translating it into MLNs.*

A3 Analysis: Predicting new collaborations boils down to finding highly-rated actors who have worked in similar genres, but have not acted together. We detected 900 groups of actors with similar genre preferences and average rating *but most of whom have not worked together*. Table 3 shows few recognizable *actors* who have not acted together, obtained after drill-down analysis. Out of these, as per reports in 2017, there had been **talks of casting Johnny Depp and Tom Cruise in pivotal roles in Universal Studios' cinematic universe titled Dark Universe** [24].

Table 3. (A5): Highly rated genre actors who have **not co-acted**.

Actors	Common prominent genres
Dafoe, Crowe	Action, Crime
Swank, Winslet	Drama
Hanks, Witherspoon, Diaz	Comedy, Romance
Depp, Cruise	Adventure, Action
DiCaprio, Gosling	Crime, Romance
Cage, Banderas	Action, Thriller
Grant, Hudson, Stone	Comedy, Romance

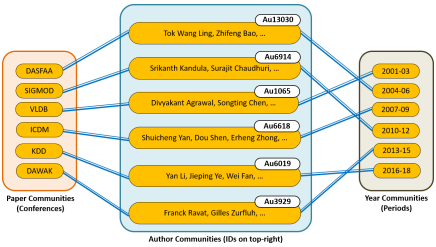

Fig. 7. (A6): Active Periods for Popular Co-authors

A6 Analysis: The **most popular unique co-author groups** for **each conference** are obtained by MWM (first composition). The matched 6 author communities are carried forward to find the *year periods* in which they were **most active** (second composition). Overall, 6 results are obtained (path shown by **bold blue lines** in Fig. 7.) Few prominent names are shown in the Fig. 7 based on citation count (from Google Scholar profiles.) For example, for *SIGMOD, VLDB and ICDM* the most popular researchers include **Srikanth Kandula (15188 citations), Divyakant Agrawal (23727 citations) and Shuicheng Yan (52294 citations)**, respectively who were active in different periods in the past 18 years.

7 Conclusions

In this paper, we have leveraged the EER modeling to generate MLNs and expressions for their analysis. *Ad hoc* big data analysis without a formal approach for generating models from application requirements is difficult and error-prone. In this paper, we have taken the first step towards modeling big data for analysis as well as drill down. We believe that this approach has broader implications.

Acknowledgments. For this work, Dr. Chakravarthy was partly supported by NSF Grant 1955798 and Dr. Bhowmick was partly supported by NSF grant 1916084.

References

1. DBLP dataset. http://dblp.uni-trier.de/xml/
2. The internet movie database. ftp://ftp.fu-berlin.de/pub/misc/movies/database/
3. Angles, R., Gutierrez, C.: Survey of graph database models. ACM Comput. Surv. (CSUR) **40**(1), 1–39 (2008)
4. Blondel, V.D., Guillaume, J., Lambiotte, R., Lefebvre, E.: Fast unfolding of community hierarchies in large networks. CoRR abs/0803.0476 (2008)
5. Chakravarthy, S., Beera, R., Balachandran, R.: DB-subdue: database approach to graph mining. In: Dai, H., Srikant, R., Zhang, C. (eds.) PAKDD 2004. LNCS (LNAI), vol. 3056, pp. 341–350. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24775-3_42
6. Chen, P.P.S.: The entity-relationship model–toward a unified view of data. ACM Trans. Database Syst. (TODS) **1**(1), 9–36 (1976)
7. Das, S., Santra, A., Bodra, J., Chakravarthy, S.: Query processing on large graphs: approaches to scalability and response time trade offs. Data Knowl. Eng. **126**, 101736 (2020)
8. De Virgilio, R., Maccioni, A., Torlone, R.: Model-driven design of graph databases. In: Yu, E., Dobbie, G., Jarke, M., Purao, S. (eds.) ER 2014. LNCS, vol. 8824, pp. 172–185. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-12206-9_14
9. Edmonds, J.: Maximum matching and a polyhedron with 0, 1-vertices. J. Res. Natl. Bureau Stand. B **69**(125–130), 55–56 (1965)
10. Elmasri, R.: Fundamentals of database systems. Pearson Education India (2008)
11. Graves, M., Bergeman, E.R., Lawrence, C.B.: Graph database systems. IEEE Eng. Med. Biol. Mag. **14**(6), 737–745 (1995)
12. Jayaram, N., Khan, A., Li, C., Yan, X., Elmasri, R.: Querying knowledge graphs by example entity tuples. IEEE Trans. Knowl. Data Eng. **27**, 2797–2811 (2015)
13. Kim, J., Lee, J.: Community detection in multi-layer graphs: a survey. SIGMOD Rec. **44**(3), 37–48 (2015)
14. Kivelä, M., Arenas, A., Barthelemy, M., Gleeson, J.P., Moreno, Y., Porter, M.A.: Multilayer networks. CoRR abs/1309.7233 (2013)
15. Melamed, D.: Community structures in bipartite networks: a dual-projection approach. PLoS ONE **9**(5), e97823 (2014)
16. Newman, M.: Networks: An Introduction. Oxford University Press Inc., New York (2010)
17. Pokornỳ, J.: Conceptual and database modelling of graph databases. In: Proceedings of the 20th International Database Engineering & Applications Symposium (2016)

18. Roy-Hubara, N., Rokach, L., Shapira, B., Shoval, P.: Modeling graph database schema. IT Professional **19**(6), 34–43 (2017)
19. Santra, A., Bhowmick, S., Chakravarthy, S.: Efficient community re-creation in multilayer networks using Boolean operations. In: International Conference on Computational Science (2017)
20. Santra, A., Bhowmick, S., Chakravarthy, S.: Hubify: efficient estimation of central entities across multiplex layer compositions. In: IEEE ICDM Workshops (2017)
21. Reddy, P.K., Sureka, A., Chakravarthy, S., Bhalla, S. (eds.): BDA 2017. LNCS, vol. 10721. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-72413-3
22. Santra, A., Komar, K.S., Bhowmick, S., Chakravarthy, S.: A new community definition for multilayer networks and a novel approach for its efficient computation. arXiv preprint arXiv:2004.09625 (2020)
23. Shi, C., Li, Y., Zhang, J., Sun, Y., Philip, S.Y.: A survey of heterogeneous information network analysis. IEEE Trans. Knowl. Data Eng. **29**(1), 17–37 (2017)
24. Stolworthy, J.: Dark universe: Johnny Depp and Javier Bardem join tom cruise in universal's monster movie franchise (2017). https://www.independent.co.uk/us
25. Sun, Y., Han, J.: Mining heterogeneous information networks: a structural analysis approach. ACM SIGKDD Exp. Newslett. **14**(2), 20–28 (2013)
26. Vu, X.S., Santra, A., Chakravarthy, S., Jiang, L.: Generic multilayer network data analysis with the fusion of content and structure. In: CICLing 2019 (2019)