Centrality-Preserving Exact Reductions of Multi-Layer Networks

Tatjana Petrov(✉) and Stefano Tognazzi(✉)

University of Konstanz, Konstanz, Germany
{tatjana.petrov,stefano.tognazzi}@uni-konstanz.de

Abstract. Multi-Layer Networks (MLN) generalise the traditional, single layered networks, by allowing to simultaneously express multiple aspects of relationships in collective systems, while keeping the description intuitive and compact. As such, they are increasingly gaining popularity for modelling Collective Adaptive Systems (CAS), e.g. engineered cyber-physical systems or animal collectives. One of the most important notions in network analysis are centrality measures, which inform us about the relative importance of nodes. Computing centrality measures is often challenging for large and dense single-layer networks. This challenge is even more prominent in the multi-layer setup, and thus motivates the design of efficient, centrality-preserving MLN reduction techniques. Network centrality does not naturally translate to its multi-layer counterpart, since the interpretation of the relative importance of nodes and layers may differ across application domains. In this paper, we take a notion of eigenvector-based centrality for a special type of MLNs (multiplex MLNs), with undirected, weighted edges, which was recently proposed in the literature. Then, we define and implement a framework for exact reductions for this class of MLNs and accompanying eigenvector centrality. Our method is inspired by the existing bisimulation-based exact model reductions for single-layered networks: the idea behind the reduction is to identify and aggregate nodes (resp. layers) with the same centrality score. We do so via efficient, static, syntactic transformations. We empirically demonstrate the speed up in the computation over a range of real-world MLNs from different domains including biology and social science.

Keywords: Multi-Layer Networks · Centrality measures · Model reduction · Efficient algorithms

1 Introduction

Traditional network analysis has facilitated key developments in research on Collective Adaptive Systems (CAS). CAS are a focus of important research efforts of today, such as ensuring the safety of cyber-physical systems, planning for smart cities, or understanding animal collective behaviour. These systems consist of a large number of entities which continuously interact with each other and the

T. Margaria and B. Steffen (Eds.): ISoLA 2020, LNCS 12477, pp. 397–415, 2020.
https://doi.org/10.1007/978-3-030-61470-6_24

environment, they self-organise and often give rise to a system-level dynamics, *emergent behaviours*, which can not be seen by studying individuals in isolation. Network representation of a collective system is intuitive, and it allows to reason over the different aspects of the modelled system, e.g. information flows, or its evolution over time. Network analysis often centers around classification of network components – nodes, edges etc. – wrt. different importance notions. Importance is defined through a *centrality* measure, and different algorithms for computing such measures have been proposed over time. A centrality measure is a real-valued function which associates nodes to their importance and, therefore, allows to rank them accordingly. Historically, the Bonacich index [5,6] (most often referred to as *eigenvector centrality*) and other extensions inspired by the Bonacich index such as Katz centrality [31] and PageRank [42] played a prominent role in network analysis. Other measures of centrality are based accordingly on different factors such as shortest paths [28], diffusion capability [1] and nodes with high contagion potential [14]. Although each of these notions measure different features of the nodes, they share common mathematical traits [4].

However, the traditional, single-layered networks allow to capture only one type of interaction among nodes. In many real-world scenarios, relations among individuals have multiple facets: in social networks, the same individuals may communicate via multiple communication platforms (i.e., they can use different online social networks to spread and gather information [54]). During epidemics, individuals interact both in the physical world, in which they spread the infection, and in a virtual communication network, where awareness about the disease is spread [26]. Moreover, animals belonging to the same collective (herd, fish school, etc.) can relate to each other differently through different activities such as grooming, social aggregation, foraging, as shown for baboons [2,23], dolphins [25] and birds [21].

Any finite, discrete number of different communication aspects among a set of agents, can be formally captured by adding typed edges or edge colours to the network description. Enriching the network formalism with multiple views/layers results in a multi-layer network (MLN) [17]. MLNs offer a novel way to model interactions among the components of a system as connected layers of different types of interactions. General MLNs allow for stacking up a collection of graphs over possibly different node-sets, through arbitrary coupling relationships between pairs of layers. In this work, we focus on a class of MLNs called *multiplex networks.* A multiplex is a collection of graphs over the same set of nodes but different edge sets, each of which is modelling a different type of interaction. Single-layer Networks are conveniently represented as matrices and many tools from matrix analysis have proven to be useful in identifying important network components. Along these lines, multiplex MLNs can be represented using tensors.

Carrying over the theory from network analysis to MLNs is desirable but non-trivial: most of the notions and concepts that are fundamental for single-layer network centrality do not naturally translate to its multi-layer counterpart, since the interpretation of the relative importance of nodes and layers may differ across application domains. For instance, in an effort to extend the Bonacich index to MLNs, several eigenvector-based centrality measures have been defined for multiplexes in the last few years [3,18,19,46]. In this work, we focus on the extension presented in [50] which is based on eigenvector centrality for undirected and (potentially) weighted multiplex MLNs. Among the large variety of methodologies for single-layer network analysis [29] such as clustering [7], blockmodeling [8] and role-equivalent partitioning [35,53], we here aim for exact, centrality-preserving network reduction. In general, model reduction techniques aim to provide a smaller or simpler dynamical model from the original one. Reductions are *exact*, when they guarantee an exact, provable relationship between their respective solutions, without error (otherwise, the reductions are *approximate*, when the error is either guaranteed or estimated). Exact, centrality-preserving network reduction was proposed in the context of single-layer networks [48]. This method is based on efficient model reduction framework for more general dynamical systems [27,49]; The core of these frameworks is based on an efficient partition-refinement procedure of Paige and Tarjan [43]. More specifically, some model reduction techniques based on lumping states have shown to preserve centrality properties of single-layer networks (such as Eigenvector centrality, Katz centrality and PageRank centrality) while, at the same time, relating to a variety of notions from different fields: exact role assignment [53], equitable partitions [37,38], lumpability [22,24] and bisimulation [44,51].

In this work, we define and implement a framework for exact model reduction of multiplex MLNs, by lumping states and layers. Reduction is designed so to preserve the eigenvector centrality for multiplex MLNs, defined in [50] (i.e., two nodes equivalent in the ODEs enjoy the same eigenvector centrality). While our proposed framework directly extends the concept used in [48] for single-layer networks, the major technical challenge arising in the multi-layer setup is that the iterative scheme for computing eigenvector centrality for MLNs contains non-linear terms. In addition, two real-valued exponents, introduced to guarantee convergence, require additional care when lifting from the reduced solution to the original one. The relevance of our framework is demonstrated by benchmarking over a number of real-world multiplex MLNs.

Paper outline. Section 2 reviews the background notions, while Sect. 3 introduces the proposed model reduction framework. Section 4 features an experimental evaluation on real-world multiplex MLNs. Section 5 concludes the paper.

2 Background

In this section we provide an overview of the notions that will be used throughout the paper: single- and multi-layer networks (MLNs), eigenvector centrality measure for MLNs, IDOL programs for specifying dynamical systems and model reduction techniques based on Backward Differential Equivalence.

Notation. Throughout this work, when clear from context, we will use x_i both to denote the i-th element of vector $\boldsymbol{x}$ or the value of the map $x(i)$ (following Definition 3). For a partition $\mathcal{H}$ over a variable set $V_p \subseteq \{\mathtt{x}_1, \mathtt{x}_2, \ldots\}$, induced by an equivalence relation $\sim_{\mathcal{H}} \subseteq V_p \times V_p$, we will denote elements of a partition class $H \in \mathcal{H}$ by $\mathtt{x}_{H,1}, \mathtt{x}_{H,2}, \ldots, \mathtt{x}_{H,|H|}$. We denote by $\|\cdot\|_1$ the 1-norm. We will denote with $V_N = \{1, \ldots, N\}$, $V_L = \{1, \ldots, L\}$ the set of nodes and layers, respectively. Vectors will be assumed to be written in column notation.

2.1 Networks and Multiplex Multi-Layer Networks

Definition 1. A (weighted, directed) *graph* is a pair $G = (V_N, E)$, where V_N is a set of $N \geq 1$ nodes and $E : V_N \times V_N \to \mathbb{R}_{\geq 0}$ is an edge-weighting function, such that $E(i,j) = 0$ reflects that there is no edge in the graphical representation of the network. In matrix notation, a graph is given by a non-negative *adjacency matrix* $\boldsymbol{A} = (A_{ij}) \in \mathbb{R}_{\geq 0}^{N \times N}$. Graph G is *undirected*, if the matrix $\boldsymbol{A}$ is symmetric.

In this paper, we will work with a generalisation of networks called *multiplex networks* or *edge-colored-graphs*, which are useful for simultaneously representing different kinds of relationships over the same set of nodes. This paper will focus on weighted, undirected multiplex networks.

Definition 2. A *multiplex network* with L layers is an ordered collection of L graphs over the same set of nodes:

$$\mathcal{G} = \{G^{(l)} = (V_N, E^{(l)})\}_{l \in V_L},$$

where $E^{(l)} : V_N \times V_N \to \mathbb{R}_{\geq 0}$ are the edge weights at layer $l \in V_L$. For every layer l, we denote the non-negative adjacency matrix of the graph $G^{(l)}$ by $\mathbf{A}^{(l)} = (A_{ij}^{(l)}) \in \mathbb{R}_{\geq 0}^{N \times N}$. Then, the multiplex network can be represented by a 3^{rd}-order *adjacency tensor*:

$$\boldsymbol{\mathcal{A}} = (\mathcal{A}_{ijl}) \in \mathbb{R}_{\geq 0}{}^{N \times N \times L}, \text{ such that } \mathcal{A}_{ijl} := A_{ij}^{(l)} = E^{(l)}(i,j),$$

that is, $\mathcal{A}_{ijl}$ is the weight of the edge from node i to node j in layer l.

Example 1. The adjacency tensor for the multiplex depicted in Fig. 1 left is given by layers $\boldsymbol{\mathcal{A}}^{(1)} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$ and $\boldsymbol{\mathcal{A}}^{(2)} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$.

Remark 1. While in this work we will focus on multiplex networks, they are a special case of a more general notion of interconnected *multilayer networks* (MLNs), where layers can have different node sets, and, moreover, they can be coupled across layers in arbitrary ways. For example, modelling public transport by different means (e.g. bus, train or metro) requires such a model.

2.2 Centrality Measures

Given an undirected graph $G = (V_N, E)$ and its adjacency matrix $\boldsymbol{A} \in \mathbb{R}_{\geq 0}^{N \times N}$, we first recall the definition of eigenvector centrality for single-layer networks [39].

Definition 3. *Eigenvector centrality* $x : V_N \to \mathbb{R}_{\geq 0}$ maps each node to the weighted sum of eigenvector centralities of all nodes directly reachable from it: for $i \in V_N$, $x(i) = \frac{1}{\lambda} \sum_{j \in V_n} A_{ij} x(j)$, where $\frac{1}{\lambda}$ is some positive constant. In vector notation, the eigenvector centrality vector $\boldsymbol{x} \in \mathbb{R}_{\geq 0}^{V_N}$ is such that $\boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{x}$, that is, $\boldsymbol{x}$ is the right eigenvector wrt. the adjacency matrix $\boldsymbol{A}$.

For a given graph with adjacency matrix $\boldsymbol{A}$, eigenvector centrality may not be well-defined, that is, there may exist no unique non-negative right eigenvector (up to linear scaling). By the famous Perron-Frobenius result, whenever the largest real eigenvalue of $\boldsymbol{A}$ is unique, eigenvector centrality is guaranteed to be well-defined, and it is the respective eigenvector, with all non-negative entries. When eigenvector centrality is well-defined, it can be efficiently computed with the power iteration scheme. We restate this well-known result, for the sake of transparent analogy with the case of MLN's, which we introduce next.

Theorem 1 ([39])**.** If there exists a unique, non-negative eigenvector centrality on $\boldsymbol{A}$, denoted by $\boldsymbol{x}^*$, and such that $\|\boldsymbol{x}^*\|_1 = 1$, it can be computed as a limit of the power iteration sequence $\boldsymbol{x}^{(k)} = \frac{\boldsymbol{A}\boldsymbol{x}^{(k-1)}}{\|\boldsymbol{A}\boldsymbol{x}^{(k-1)}\|_1}$ for $k \geq 0$ and initially $\boldsymbol{x}^{(0)} = \mathbb{1}^N$.

In this paper, we will use one possible extension of eigenvector centrality for multiplex MLNs, proposed in [50]. The authors propose a 2-map, $\boldsymbol{f}$*-eigenvector centrality*, in which the first component of the map represents the centrality associated to the *nodes*, while the second component is centrality associated to the *layers*.

Definition 4 ([50])**.** Let $\boldsymbol{\mathcal{A}} \in \mathbb{R}_{\geq 0}^{N \times N \times L}$ be the adjacency tensor of an MLN with weighted, undirected layers, and let $\alpha, \beta > 0$ be such that $\frac{2}{\beta} < (\alpha - 1)$. Then, define $\boldsymbol{f} = (f_1, f_2) : \mathbb{R}_{\geq 0}^{N} \times \mathbb{R}_{\geq 0}^{L} \to \mathbb{R}_{\geq 0}^{N} \times \mathbb{R}_{\geq 0}^{L}$ as follows:

$$f_1(\boldsymbol{x}, \boldsymbol{t})_i = \left(\sum_{j=1}^{N} \sum_{l=1}^{L} A_{ijl} x_j t_l \right)^{\frac{1}{\alpha}} \text{ for } i \in V_N, \; f_2(\boldsymbol{x}, \boldsymbol{t})_l = \left(\sum_{i=1}^{N} \sum_{j=1}^{N} A_{ijl} x_i x_j \right)^{\frac{1}{\beta}} \text{ for } l \in V_L.$$

In words, the centrality x_i of node i is a sum of the centralities of each of its neighbouring nodes, weighted by the product of the edge-weight and the centrality of the layer at which that connection lies. At the same time, the centrality of a layer t_l is a sum of the centrality of all edges at that layer, where an importance of an edge is, in addition to its own weight, weighted by the centrality of the two nodes which constitute it. The parameters α and β are introduced in order to guarantee convergence and respectively well-definedness in case of undirected MLNs. Further discussion is beyond the scope of this manuscript and we refer the interested reader to [50].

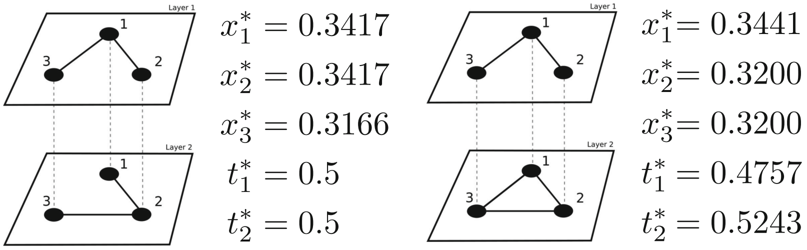

Fig. 1. An example with two MLNs and their respective f-eigenvector centralities.

Similarly as in the case of single-layer networks, a power iteration scheme for computing $\boldsymbol{f}$-centrality is desired. Throughout the rest of the work, we will use a normalised version of $\boldsymbol{f}$-mapping, denoted by $\boldsymbol{g}$:

$$\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{t}) = \left(\frac{\boldsymbol{f}_1(\boldsymbol{x}, \boldsymbol{t})}{\|\boldsymbol{f}_1(\boldsymbol{x}, \boldsymbol{t})\|_1}, \frac{\boldsymbol{f}_2(\boldsymbol{x}, \boldsymbol{t})}{\|\boldsymbol{f}_2(\boldsymbol{x}, \boldsymbol{t})\|_1} \right)$$

We now restate a result from [50], that, for a given MLN with undirected layers, $\boldsymbol{f}$-centrality is well-defined and it can be computed as a limit of a power iterative sequence.

Theorem 2 ([50]). There exists a unique, non-negative fixed point of the mapping $\boldsymbol{g}$. Moreover, this fixed point, denoted by $(\boldsymbol{x}^*, \boldsymbol{t}^*) \in \mathbb{R}^N_{\geq 0} \times \mathbb{R}^L_{\geq 0}$, is a limit of the following iterative scheme[1]:

$$(\boldsymbol{x}^{(k)}, \boldsymbol{t}^{(k)}) = \boldsymbol{g}(\boldsymbol{x}^{(k-1)}, \boldsymbol{t}^{(k-1)}) \text{ for } k \geq 1 \text{ and initially } (\boldsymbol{x}^{(0)}, \boldsymbol{t}^{(0)}) = (\mathbb{1}^N, \mathbb{1}^L) \tag{1}$$

Notice that, from the definition of $\boldsymbol{g}$, independently of $k \geq 0$, it holds that $\left\|\boldsymbol{x}^{(k)}\right\|_1 = \left\|\boldsymbol{t}^{(k)}\right\|_1 = 1$, including the limit value $(\boldsymbol{x}^*, \boldsymbol{t}^*)$.

Example 2. Consider the MLN depicted in Fig. 1 (left). The iterative scheme to compute the $\boldsymbol{f}$-eigenvector centrality (Definition 4) is the following:

$$\begin{aligned}
x_1^{(k+1)} &= (1x_2^{(k)}t_1^{(k)} + 1x_3^{(k)}t_1^{(k)} + 1x_2^{(k)}t_2^{(k)})^{\frac{1}{\alpha}} / ||\boldsymbol{f}_1(\boldsymbol{x}, \boldsymbol{t})||_1 \\
x_2^{(k+1)} &= (1x_1^{(k)}t_1^{(k)} + 1x_1^{(k)}t_2^{(k)} + 1x_3^{(k)}t_2^{(k)})^{\frac{1}{\alpha}} / ||\boldsymbol{f}_1(\boldsymbol{x}, \boldsymbol{t})||_1 \\
x_3^{(k+1)} &= (1x_1^{(k)}t_1^{(k)} + 1x_2^{(k)}t_2^{(k)})^{\frac{1}{\alpha}} / ||\boldsymbol{f}_1(\boldsymbol{x}, \boldsymbol{t})||_1 \\
t_1^{(k+1)} &= (2x_1^{(k)}x_2^{(k)} + 2x_1^{(k)}x_3^{(k)})^{\frac{1}{\beta}} / ||\boldsymbol{f}_2(\boldsymbol{x}, \boldsymbol{t})||_1 \\
t_2^{(k+1)} &= (2x_1^{(k)}x_2^{(k)} + 2x_2^{(k)}x_3^{(k)})^{\frac{1}{\beta}} / ||\boldsymbol{f}_2(\boldsymbol{x}, \boldsymbol{t})||_1
\end{aligned}$$

[1] We refer the interested reader to the original reference, for a discussion on the error and rate of convergence.

Example 3. In Fig. 1 we show two different MLNs and their respective $\boldsymbol{f}$-eigenvector centralities. Adding an edge at Layer 2 changes both the node centrality and the layer centrality scores. More specifically, Node 1 and 3 gain importance while Node 2 loses importance. Moreover, if in the left example the two layers had equivalent centralities, in the right one, Layer 2 becomes more important because it contains more connections between high-centrality-nodes. This shows that when we choose $\boldsymbol{f}$-eigenvector centrality as the measure of choice, the role played by the nodes and layers is intertwined and therefore the two aspects of the $\boldsymbol{f}$ mapping can not be computed independently.

2.3 Intermediate Drift Oriented Language (IDOL)

The Intermediate Drift Oriented Language (IDOL) is a language for describing non-linear, first-order, autonomous and explicit finite systems of coupled ordinary differential equations (ODEs). We here report the fragment of the syntax and semantics of IDOL which is useful for presenting this work.

Syntax. An IDOL program p over a set of variables V_p is written in the following syntax:

$$p \ ::= \varepsilon \mid \mathtt{x}_i' = \eta, p$$

$$\eta \ ::= n \mid \mathtt{x}_i \mid \eta + \eta \mid \eta \cdot \eta$$

where $\mathtt{x}_i \in V_p$, $n \in \mathbb{Z}$ and ε is used to define the end of the program.

Semantics. We will consider conventional ODE semantics for a given IDOL program p, as the solution of the system of ODE's that it represents. The solution map $[\![\cdot]\!] : \mathbb{R}_{\geq 0}^{|V_p|} \to (V_p \to ([0,T) \to \mathbb{R}_{\geq 0}))$ will (deterministically) map each initial condition and a variable to a trace from the time domain with horizon $T \in \mathbb{R}_{\geq 0}$ to a value. For simplicity, we will denote the solution for variable $\mathtt{x}_i$ by $[\![\mathtt{x}_i]\!]_{\boldsymbol{x}_0}$, and we omit the dependency on initial condition $\boldsymbol{x}_0 \in \mathbb{R}_{\geq 0}^{|V_p|}$ when clear from context.

2.4 Backward Differential Equivalence

Backward differential equivalence (BDE) is a model reduction technique for dynamical systems written in IDOL [9,12]. BDE groups IDOL variables which are *exact fluid lumpable* - they have the same ODE semantics whenever they are given the same initial assignment. Finding the (largest) BDE amounts to finding the coarsest partition over the variable set, which ensures that the semantic criterion is met. This criterion allows to construct a smaller IDOL program, using only one representative variable from each partition class. The reduction algorithms proposed in [9,12] are only syntactically manipulating the IDOL program, and they are of polynomial complexity in the number of variables of the program. We propose in this paper to use BDE reductions, to reduce the computation of $\boldsymbol{f}$-centrality measure for MLNs.

Definition 5. We call $\boldsymbol{x} \in \mathbb{R}_{\geq 0}^{|V_p|}$ *constant on* $\mathcal{H}$ if for all $H \in \mathcal{H}$ and all $x_i, x_j \in H$, it holds that $x_i = x_j$.

Definition 6. Let p be an IDOL program and $\mathcal{H}$ a partition over the variable set V_p. Then, the IDOL program p is *exact fluid lumpable wrt. partition* $\mathcal{H}$, if $[\![\boldsymbol{x}]\!]_{\boldsymbol{x}_0}(t)$ is constant on $\mathcal{H}$ for all $t \geq 0$, whenever $\boldsymbol{x}_0$ is constant on $\mathcal{H}$. Then, we will call $\mathcal{H}$ a *BDE partition* of V_p.

Following [9], the coarsest BDE partition can be computed in polynomial time complexity, for any IDOL program which corresponds to a set of chemical reactions with mass-action kinetics.

We now state the result which shows how to use a BDE to construct a reduced IDOL program, operating over only the representative variables (BDE quotient).

Theorem 3 ([10])**.** Let p be an IDOL program and $\mathcal{H}$ a BDE partition over its variable set V_p, and $T > 0$ a time horizon. The *backward reduced program of* p *with respect to* $\mathcal{H}$, denoted by $\tilde{p}_{\mathcal{H}}$, is defined over a set of variables $V_{p_{\mathcal{H}}} = \{\tilde{\mathtt{x}}_1, \ldots, \tilde{\mathtt{x}}_{|\mathcal{H}|}\}$, and with the following update functions:

$$\tilde{\mathtt{x}}'_H = \eta_{H,1}[\mathtt{x}_{\bar{H},1}/\tilde{\mathtt{x}}_{\bar{H}}, \ldots, \mathtt{x}_{\bar{H},|\bar{H}|}/\tilde{\mathtt{x}}_{\bar{H}} : \bar{H} \in \mathcal{H}], \text{ for } H \in \mathcal{H},$$

where, $\mathtt{x}_{\bar{H},i}/\tilde{\mathtt{x}}_{\bar{H}}$ denotes the action of renaming variable $\mathtt{x}_{\bar{H},i}$ by $\tilde{\mathtt{x}}_{\bar{H}}$.

Originally designed for reducing ODEs, BDE techniques have also been applied for reducing single-layer networks, continuous-time Markov chains (CTMCs) and differential algebraic equations. In particular, in [48], a property-preserving exact model reduction algorithm for networks is shown. The given network is first transformed into an IDOL program, and then a BDE reduction ensuring exact fluid lumpability is applied. We restate a Theorem showing that BDE reduction also preserves the measure of eigenvector centrality.

Theorem 4 ([48])**.** Given a graph $G = (V_N, E)$ with adjacency matrix $\boldsymbol{A}$, let p_G be the IDOL program over the set of variables V_N:

$$\mathtt{x}'_i = \sum_{1 \leq j \leq n} A_{ij} \cdot \mathtt{x}_j \ , \quad \text{for all } \ i \in V_N.$$

Let x_i^* denote the eigenvector centrality of node i. Then, $\mathcal{H}$ is a BDE of p_G if and only if, for all $H \in \mathcal{H}$ and for all $x_i, x_j \in H$, it holds that $x_i^* = x_j^*$.

In words, the transformation from network to IDOL program is such that the equation for the derivative of variable $\mathtt{x}_i$ is the weighted sum of its direct (outgoing) edges[2]. So, the key idea in the transformation from the network to an IDOL program is that the equations in the IDOL program exactly match the iterative scheme for computing the centrality measure of interest.

Notice that the obtained IDOL program contains only linear transformations over its variables. We next propose an analogue of Theorem 4 for multiplex MLNs. The translation to an IDOL program will encode the iterative scheme for computing $\boldsymbol{f}$-centrality, which involves non-linear terms (of second order).

[2] In case of symmetric graphs, ingoing and outgoing edges will be indistinguishable and overall neighbours are accounted for.

3 Centrality-Preserving MLN Reduction

Given a multi-layered network, its $\boldsymbol{f}$-centrality can be computed with the iterative scheme $\boldsymbol{g}$ presented in Theorem 2 (Eq. 1) . Our aim is to bypass the direct computation and instead compute the $\boldsymbol{f}$-centrality indirectly, by first detecting sets of variables which evolve equivalently throughout the iterations, and then proposing a reduced iterative scheme $\tilde{\boldsymbol{g}}$, where only one representative variable is kept for each set of equivalent ones.

To do so, we first introduce an assignment of an IDOL program to a given MLN. Then, given an IDOL program, we compute its BDE-equivalent quotient, as described in Sect. 2. BDE equivalence guarantees that the original and smaller IDOL programs have the same differential, continuous-step semantics. On the other hand, the iterative scheme $\boldsymbol{g}$ defines a discrete-step semantics over the variables of the MLN. Our proposal is to compute the centrality measure of the original MLN using the smaller IDOL program. To do so, we need to show that the reduced iterative scheme $\tilde{\boldsymbol{g}}$ over the BDE-quotient of the original IDOL program, preserves the solutions of the iterative scheme $\boldsymbol{g}$. A diagram of the workflow of the proposed framework is presented in Fig. 2.

The following theorem shows which are the quantities that we should account for when we search for equivalences among the centrality scores.

Theorem 5. For $i \in V_N$, define the quantity of interest

$$\bar{x}_i^{(k)} := \sum_{a=1}^{N} \sum_{l=1}^{L} A_{ial} x_a^{(k)} t_l^{(k)},$$

$$\begin{tikzcd}
\mathcal{G} \arrow[r, thick] & \boldsymbol{g} \arrow[d, thick] \arrow[dr] \arrow[rr, dashed, thick, bend left] & & \\
& CRN\ (\mathcal{S}, \mathcal{R}) \arrow[r, leftrightarrow, thick] \arrow[d, leftrightarrow] & \text{IDOL}\ p \arrow[r] \arrow[d, thick] & (\boldsymbol{x}^*, \boldsymbol{t}^*) \\
& CRN\ (\tilde{\mathcal{S}}, \tilde{\mathcal{R}}) \arrow[r, leftrightarrow] \arrow[d] & \text{IDOL}\ \tilde{p}_{\mathcal{H}} \arrow[dl, thick] & (\tilde{\boldsymbol{x}}^*, \tilde{\boldsymbol{t}}^*) \arrow[u, thick] \\
& \tilde{\boldsymbol{g}} \arrow[urr, thick, bend right] & &
\end{tikzcd}$$

Fig. 2. An illustration of the proposed methodology. The arrows are used for illustrative purpose and they are not to be formally interpreted. For a given MLN $\mathcal{G}$, its $\boldsymbol{f}$-centrality vector $(\boldsymbol{x}^*, \boldsymbol{t}^*)$ can be computed directly through the iterative map $\boldsymbol{g}$ (dotted line). Alternatively, as depicted with thick full arrows, the equations in $\boldsymbol{g}$ can first be translated into an IDOL program p with variables, and its BDE quotient program $p_{\mathcal{H}}$ is used to define a reduced iterative scheme $\tilde{\boldsymbol{g}}$ over a reduced set of variables, the solution of which, $(\tilde{\boldsymbol{x}}^*, \tilde{\boldsymbol{t}}^*)$, allows to exactly reconstruct the $\boldsymbol{f}$-centrality of the original MLN.

which is the right hand side of the mapping $\boldsymbol{f}$ from Definition 4, without the exponential operator α. Then, for all pairs of nodes $i, j \in V_N$, it holds that

$$\text{if } \bar{x}_i^{(k)} = \bar{x}_j^{(k)} \text{ then } x_i^{(k+1)} = x_j^{(k+1)}, \text{ for all } k \geq 0.$$

The key idea is to go from the definition of the multilayer network eigenvector centrality obtained with the iterative scheme (1) to an IDOL program p such that there is a correspondence between the node and the layer eigenvector centrality and the variables of the IDOL program.

Definition 7. (IDOL translation) Let $\mathcal{G}$ be a multiplex network and let $\boldsymbol{\mathcal{A}}_{ijl}$ be the 3-rd order adjacency tensor of the multiplex $\mathcal{G}$. We define an IDOL program p, with $V_p = V_N \cup V_L$, as follows:

$$\mathtt{x}_i' = \sum_{j=1}^{N} \sum_{l=1}^{L} A_{ijl} \mathtt{x}_j \mathtt{t}_l \qquad \mathtt{t}_l' = \sum_{i=1}^{N} \sum_{j=1}^{N} A_{ijl} \mathtt{x}_i \mathtt{x}_j$$

for all $i \in V_N$ and for all $l \in V_L$. With $\boldsymbol{x}_0 = \mathbb{1}^N$ and $\boldsymbol{t}_0 = \mathbb{1}^L$.

We now want to identify which nodes in the MLN have identical $\boldsymbol{f}$-eigenvector centrality. This holds if they follow equivalent equations in the iterative scheme used for computing them. Similarly to the result presented in Theorem 4 which shows a similar translation for single layer networks, the iterative scheme equations used to compute the $\boldsymbol{f}$-eigenvector centralities on MLNs can be translated to an IDOL program. The major technical difference is that the MLN translation contains non-linear terms, and the exponents α and β. Once we have obtained the corresponding IDOL program we can apply the general technique for computing the equivalences among its variables.

The next Theorem shows how to write an IDOL program, such that if two variables have the same semantics in the dynamical system of the IDOL program, then, the respective nodes in the iterative scheme of a given MLN have identical centrality scores over all the steps of the computation, provided the equivalence over initial conditions.

Theorem 6. Let $\mathcal{G}$ be a multiplex network and let $\boldsymbol{\mathcal{A}}_{ijl}$ be the 3-rd order adjacency tensor of the multiplex network $\mathcal{G}$. Let $\boldsymbol{f}$ be the mapping as defined in Definition 4, and let $\boldsymbol{g}$ its normalized version. Let $(\boldsymbol{x}^*, \boldsymbol{t}^*)$ be the unique solution (the centrality scores). Given any initial conditions $(\boldsymbol{x}^{(0)}, \boldsymbol{t}^{(0)}) \in \mathbb{R}^N_{\geq 0} \times \mathbb{R}^L_{\geq 0}$ and $(\boldsymbol{x}^{(k+1)}, \boldsymbol{t}^{(k+1)}) = \boldsymbol{g}(\boldsymbol{x}^{(k)}, \boldsymbol{t}^{(k)})$, the following holds:

$$\lim_{k \to \infty} (\boldsymbol{x}^{(k)}, \boldsymbol{t}^{(k)}) = (\boldsymbol{x}^*, \boldsymbol{t}^*)$$

Then, in the IDOL program p obtained via Definition 7, for some $i, j \in \{1, \ldots, N\}$ and $l, q \in \{1, \ldots, L\}$, the following holds:

- If $\forall t \in [0, T) \;.\; [\![\mathtt{x}_i]\!](t) = [\![\mathtt{x}_j]\!](t)$ in the IDOL program p, then $\forall k \in \mathbb{N} \;.\; x_i^{(k)} = x_j^{(k)}$
- If $\forall t \in [0, T) \;.\; [\![\mathtt{t}_l]\!](t) = [\![\mathtt{t}_q]\!](t)$ in the IDOL program p, then $\forall k \in \mathbb{N} \;.\; t_l^{(k)} = t_q^{(k)}$

From Theorem 6 we now know that we can build a non-linear IDOL program p such that if two variables are equal in the IDOL program then, the corresponding nodes (or layer) centrality are equal. Now, we can use the established results on the IDOL program and calculate the BDE partition $\boldsymbol{\mathcal{H}}$ on the IDOL program p generated with Theorem 6. With the next Theorem we show that, because of the relationship established by Theorem 6 and the established results on the notion of BDE, we can carry over the results that we obtain on the IDOL program to the procedure to calculate the multilayer node (or layer) centrality.

Up to this point, starting from a multiplex graph $\mathcal{G}$ we provided a procedure to translate it into an IDOL program and we provided a technique to calculate a BDE partition $\boldsymbol{\mathcal{H}}$ on the IDOL program. Now, we introduce the following lemma and definition to formally translate partition $\boldsymbol{\mathcal{H}}$, which is defined over the IDOL program's variables, to its counterpart $\boldsymbol{\mathcal{H}}^*$ defined over the nodes and the layers of the multiplex graph $\mathcal{G}$.

Lemma 1. Let $\mathcal{G}$ be a multiplex network and let p be the IDOL program defined in Theorem 6 and let $\boldsymbol{\mathcal{H}} = (\mathcal{H}_x, \mathcal{H}_t)$ be a BDE partition over the set of variables such that there is no overlap between the nodes and the layers, i.e. $\mathcal{H}_x$ is a partition over the node variables $\{\mathtt{x}_1, \ldots, \mathtt{x}_N\}$ and $\mathcal{H}_t$ is a partition over the layer variables $\{\mathtt{t}_1, \ldots, \mathtt{t}_L\}$. Then, for all initial conditions the following holds:

- $\forall t \in [0, T)$, $\forall \mathtt{x}_i, \mathtt{x}_j \in H_x$, $\forall H_x \in \mathcal{H}_x$. $[\![\mathtt{x}_i]\!](t) = [\![\mathtt{x}_j]\!](t) \implies \forall k \in \mathbb{N}$. $x_i^{(k)} = x_j^{(k)}$
- $\forall t \in [0, T)$, $\forall \mathtt{t}_l, \mathtt{t}_q \in H_t$, $\forall H_t \in \mathcal{H}_t$. $[\![\mathtt{t}_l]\!](t) = [\![\mathtt{t}_q]\!](t) \implies \forall k \in \mathbb{N}$. $t_l^{(k)} = t_q^{(k)}$

Moreover, let $\mathcal{G}$ be the corresponding multiplex graph and we define $\boldsymbol{\mathcal{H}}^* = (\mathcal{H}_x^*, \mathcal{H}_t^*)$ as the corresponding partition over the node and layer variables $\{x_1, \ldots, x_N, t_1, \ldots, t_L\}$ of $\mathcal{G}$. We define $\boldsymbol{\mathcal{H}}^*$ as follows:

$$\forall i, j \in \{1, \ldots, N\}, H_{a,x} \in \mathcal{H}_x \;.\; \mathtt{x}_i, \mathtt{x}_j \in H_{a,x} \implies H^*_{a,x} \in \mathcal{H}^* \;.\; x_i, x_j \in H^*_{a,x}$$
$$\forall i, j \in \{1, \ldots, L\}, H_{a,t} \in \mathcal{H}_t \;.\; \mathtt{t}_i, \mathtt{t}_j \in H_{a,t} \implies H^*_{a,t} \in \mathcal{H}^* \;.\; t_i, t_j \in H^*_{a,t}$$

Example 4. If we go back to the running example presented in the left of Fig. 1 and we apply Theorem 6 we obtain the following IDOL program p:

$$\begin{aligned} \mathtt{x}_1' &= 1\mathtt{x}_2\mathtt{t}_1 + 1\mathtt{x}_3\mathtt{t}_1 + 1\mathtt{x}_2\mathtt{t}_2 & \mathtt{t}_1' &= 2\mathtt{x}_1\mathtt{x}_2 + 2\mathtt{x}_1\mathtt{x}_3 \\ \mathtt{x}_2' &= 1\mathtt{x}_1\mathtt{t}_1 + 1\mathtt{x}_1\mathtt{t}_2 + 1\mathtt{x}_3\mathtt{t}_2 & \mathtt{t}_2' &= 2\mathtt{x}_1\mathtt{x}_2 + 2\mathtt{x}_2\mathtt{x}_3 \\ \mathtt{x}_3' &= 1\mathtt{x}_1\mathtt{t}_1 + 1\mathtt{x}_2\mathtt{t}_2 \end{aligned}$$

We consider the following partition $\mathcal{H} = \{\{\mathtt{x}_1, \mathtt{x}_2\}, \{\mathtt{x}_3\}, \{\mathtt{t}_1\}, \{\mathtt{t}_2\}\}$, which is a BDE of p and we shall use $\tilde{\mathtt{x}}_1$ as the representative of block $\{\mathtt{x}_1, \mathtt{x}_2\}$, $\tilde{\mathtt{x}}_2$ as the representative of block $\{\mathtt{x}_3\}$ and $\mathtt{r}_1, \mathtt{r}_2$ representatives of the blocks $\{\mathtt{t}_1\}, \{\mathtt{t}_2\}$, respectively. The IDOL quotient of p given $\mathcal{H}$ is the following:

$$\begin{aligned} \mathtt{y}_1' &= 1\mathtt{y}_1\mathtt{r}_1 + 1\mathtt{y}_2\mathtt{r}_1 + 1\mathtt{y}_1\mathtt{r}_2 & \mathtt{r}_1' &= 2\mathtt{y}_1\mathtt{y}_1 + 2\mathtt{y}_1\mathtt{y}_2 \\ \mathtt{y}_2' &= 1\mathtt{y}_1\mathtt{r}_1 + 1\mathtt{y}_1\mathtt{r}_2 & \mathtt{r}_2' &= 2\mathtt{y}_1\mathtt{y}_1 + 2\mathtt{y}_1\mathtt{y}_2 \end{aligned}$$

Now that we established the relationship between the partitions we proceed to define the proper reduced system to calculate the multiplex node and layer centrality as follows.

Definition 8. Let $(\boldsymbol{x}^{(k)}, \boldsymbol{t}^{(k)}) = \boldsymbol{g}(\boldsymbol{x}^{(k-1)}, \boldsymbol{t}^{(k-1)})$ be the iterative scheme and let $\boldsymbol{\mathcal{H}} = (\mathcal{H}_x, \mathcal{H}_t)$ be the BDE partition on the IDOL program p obtained using Theorem 6 and Lemma 1. Let $\boldsymbol{\mathcal{H}}^* = (\mathcal{H}_x^*, \mathcal{H}_t^*)$ be the corresponding partition on the iterative scheme as defined in Lemma 1. We define $(\boldsymbol{y}^{(k)}, \boldsymbol{r}^{(k)}) = \tilde{\boldsymbol{g}}(\boldsymbol{y}^{(k-1)}, \boldsymbol{r}^{(k-1)})$ as the *Reduced iterative scheme* with respect to $\boldsymbol{\mathcal{H}}^*$:

$$\tilde{\boldsymbol{f}}_1 = \boldsymbol{f}_1[x_{H_x,1}/y_{H_x}, \dots, x_{H_x,|H_x|}/y_{H_x}, t_{H_t,1}/r_{H_t}, \dots, t_{H_t,|H_t|}/r_{H_t} : H_x \in \mathcal{H}_x \ , \ H_t \in \mathcal{H}_t]$$
$$\tilde{\boldsymbol{f}}_2 = \boldsymbol{f}_2[x_{H_x,1}/y_{H_x}, \dots, x_{H_x,|H_x|}/y_{H_x}, t_{H_t,1}/r_{H_t}, \dots, t_{H_t,|H_t|}/r_{H_t} : H_x \in \mathcal{H}_x \ , \ H_t \in \mathcal{H}_t]$$

Next, we define $\bar{y}_i^{(k)}$, similarly as we previously defined $\bar{x}_i^{(k)}$, and the reduced computation to retrieve the values of $\boldsymbol{x}^{(k)}$:

$$\bar{y}_i^{(k)} = \sum_{j=1}^{N}\sum_{l=1}^{L} A_{ijl} y_{H_{x,j}}^{(k-1)} r_{H_{t,l}}^{(k-1)}, \qquad \boldsymbol{x}^{(k)} = \frac{\bar{\boldsymbol{x}}^{(k)}}{\left\|\bar{\boldsymbol{x}}^{(k)}\right\|_1} = \frac{\bar{\boldsymbol{y}}^{(k)}}{\sum_{j=1}^{m} |H_{x,j}| \bar{y}_j^{(k)}} = \boldsymbol{y}^{(k)}.$$

where, $H_{x,j} = i$ if $x_j \in H_{x,i}$ and $H_{t,q} = l$ if $t_q \in H_{t,l}$. We can now focus on the second component of the mapping and we define $\bar{r}_l^{(k)}$ and its reduced computation:

$$\bar{r}_l^{(k)} = \sum_{i=1}^{N}\sum_{j=1}^{N} A_{ijl} y_{H_{x,i}}^{(k-1)} y_{H_{x,j}}^{(k-1)}, \qquad \boldsymbol{t}^{(k)} = \frac{\bar{\boldsymbol{t}}^{(k)}}{\left\|\bar{\boldsymbol{t}}^{(k)}\right\|_1} = \frac{\bar{\boldsymbol{r}}^{(k)}}{\sum_{j=1}^{Q} |H_{t,j}| \bar{r}_j^{(k)}} = \boldsymbol{r}^{(k)}.$$

where, $H_{x,j} = i$ if $x_j \in H_{x,i}$.

Example 5. If we consider the running example presented in the left of Fig. 1, we know that the partition $\mathcal{H} = \{\{\mathtt{x}_1, \mathtt{x}_2\}, \{\mathtt{x}_3\}, \{\mathtt{t}_1\}, \{\mathtt{t}_2\}\}$ is a BDE of the IDOL program p and we obtained the following reduced IDOL program:

$$\mathtt{y}_1' = 1\mathtt{y}_1\mathtt{r}_1 + 1\mathtt{y}_2\mathtt{r}_1 + 1\mathtt{y}_1\mathtt{r}_2 \qquad \mathtt{r}_1' = 2\mathtt{y}_1\mathtt{y}_1 + 2\mathtt{y}_1\mathtt{y}_2$$
$$\mathtt{y}_2' = 1\mathtt{y}_1\mathtt{r}_1 + 1\mathtt{y}_1\mathtt{r}_2 \qquad \mathtt{r}_2' = 2\mathtt{y}_1\mathtt{y}_1 + 2\mathtt{y}_1\mathtt{y}_2$$

In order to compute the original $\boldsymbol{f}$-eigenvector centrality values we set up the following iterative scheme:

$$\bar{y}_1^{(k)} = 1y_1^{(k-1)}r_1^{(k-1)} + 1y_2^{(k-1)}r_1^{(k-1)} + 1y_1^{(k-1)}r_2^{(k-1)} \qquad \bar{r}_1^{(k)} = 2y_1^{(k-1)}y_1^{(k-1)} + 2y_1^{(k-1)}y_2^{(k-1)}$$
$$\bar{y}_2^{(k)} = 1y_1^{(k-1)}r_1^{(k-1)} + 1y_1^{(k-1)}r_2^{(k-1)} \qquad \bar{r}_2^{(k)} = 2y_1^{(k-1)}y_1^{(k-1)} + 2y_1^{(k-1)}y_2^{(k-1)}$$

4 Experimental Results

In this section we present the results of our experimental evaluation on some real world case studies. We measure the performance of our approach in terms of model reduction ratio and we measure the speed up in the computation of the desired centrality measures.

Implementation and Environment. The tools used for the experiments are MATLAB and ERODE [11], a state-of-the-art tool for model reduction for systems of ODEs and Chemical Reaction Networks. The input is the list of edges $E^{(l)}$ for all $l \in \{1, ..., L\}$ representing a multiplex network $\mathcal{G} = \{G^{(l)} = (V_N, E^{(l)})\}_{l \in V_L}$. ERODE accepts the input as a file that encodes an ODE system or a Chemical Reaction Network (CRN). Due to a bottleneck in the processing of files in the ODE format we had to input the files in the CRN format. A MATLAB script translates the list of edges in the CRN. ERODE then proceeds with the model reduction and provides the reduced CRN as its output. The centrality scores are computed with a MATLAB script and another MATLAB script is used to convert the reduced CRN into the reduced model and used to calculate the centrality score on the reduced model. All the experiments have been conducted on a MacBook Pro with a 2.6 GHz Intel Core i7 with 16 GB of RAM.

The Instances. In order to provide some real-world case studies we ran our proposed reduction technique on multiplex MLNs retrieved from the CoMuNe Lab repository (https://comunelab.fbk.eu). The results for both undirected and directed instances are presented in Table 1. We first present the *undirected graphs* instances. These instances are undirected in the repository.

- *Padgett-Florentine-Families* (1): this multiplex describes the relationships between Florentine families in the Renaissance, the two layers represent marriage alliances and business relationships, respectively [41].
- *CS-Aarhus* (2): this multiplex social network consists of five kinds of relationships between the employees of the Computer Science department at Aarhus university. The layers represent the following relationships: Facebook, Leisure, Work, Co-Authorship and Lunch [36].
- *London-Transport* (3): the nodes in this multiplex represent the train stations in London and edges encode existing routes between stations. The layers represent the Underground, Overground and DLR stations, respectively [16].
- *EUAirTrainsportation* (4): the multilayer network is composed by thirty-seven different layers each one corresponding to a different airline operating in Europe [13].
- *PierreAuger* (5): this instance represents the different working tasks carried out over a two years span within the Pierre Auger Collaboration between the CoMuNe Lab and the Pierre Auger observatory. Each layer represents 16 different topics based on the keywords and the content of each submission [20].
- *arxiv-netscience* (6): this multiplex consists of layers corresponding to different arXiv categories. Nodes represent authors and edges represent the weighted co-authorship relationship [20].

Due to the fact that many of the undirected instances are small we do not obtain sensible reductions nor speed up in the computation. Despite this, we can observe a meaningful reduction for the largest of the undirected instances, namely the *arxiv-netscience* instance. We now present the instances that in the repository are *directed*. It is worth noting that, because of the fact that the centrality measure we considered throughout this paper is defined for the undirected case only, we modified these instances to make them undirected in order to prove the effectiveness of our proposed methodology. Another reason to do so is the fact that there is a small number of undirected instances. Moreover, the undirected instances present a limited variety of nodes, edges and layer sizes.

- *Krackhardt-High-Tech* (7): this multiplex social network describes the relationships between managers of an high-tech company. The layers represent advice, friendship and “reports to” relationships, respectively [33].
- *Vickers-Chan-7thGraders* (8): this data was collected from 29 seventh grade students in a school in Victoria, Australia. Students were asked to nominate their classmates on a number of relations including the following three (layers): Who do you get on with in the class? Who are your best friends in the class? Who would you prefer to work with? [52].
- *Kapferer-Tailor-Shop* (9): this instance represents the interactions in a tailor shop in Zambia over a period of ten months. The layers represent two different types of interactions, recorded at two different times. The relationships captured by this multiplex are *instrumental* (work-related) and *sociational* (friendship, socio-emotional) interactions [30].
- *Lazega-Law-Firm* (10): this multiplex social network consists of three kinds of relationships between partners and associates of a corporate law partnership. The layers represent co-work, friendship and advice relationships, respectively [34,45].
- *Genetic interaction instances* (11-28): we consider a variety of genetic interactions networks that are present in the CoMuNe Lab repository [15]. In turn, these instances were taken from the Biological General Repository for Interaction Datasets (BioGRID) and represent different types of genetic interactions for organisms [47]. More specifically, according to the nomenclature used in the repository we present experimental results on the following instances: HepatitusC (11), DanioRerio (12), HumanHerpes4 (13), CElegans Connectome (15), Bos (16), Candida (17), Xenopus (18), HumanHIV1 (19), Plasmodium (20), Rattus (21), CElegans (22), Sacchpomb (23), Sacchere (24), Arabidopsis (25), Mus (26), Drosophila (27), Homo (28).
- *CKM-Physicians-Innovation* (14): this multiplex describes how new drugs adoption spreads in a community of physicians.
- *Fao-Trade* (29): this multiplex describes different types of trade relationships among countries, it was originally obtained from the Food and Agriculture Organization of the United Nations. The worldwide food import/export network is an economic network in which layers represent products, nodes are countries and edges at each layer represent import/export relationships of a specific food product among countries. It is worth pointing out that, due to the nature of this instance, it has the peculiarity that there are more layers than nodes [15].
- *MoscowAthletics2013* (30): this multiplex represents the different types of social relationships among Twitter users during the 2013 World Championships in Athletics. The three layers correspond to retweets, mentions and replies over the time frame of the event. These are the relationships that will be also used for the following Twitter instances [40].

- *NYClimateMarch2014* (31): this instance describes the Twitter interactions among the users during the People's Climate March in 2014 [40].
- *MLKing2013* (32): this instance describes the Twitter interactions among the users during the 50^{th} anniversary of Martin Luther King's speech in 2013 [40].
- *Cannes2013* (33): this instance describes the Twitter interactions among the users during the Cannes Film Festival in 2013 [40].

As expected, similarly to the undirected instances, the small instances do not provide much insight on the effectiveness of the methodology but, as the size of the instance increases we can see significant reductions and speed ups. Notably, when tackling the big instances the proposed methodology yields reductions that reduce the size of the network to half of its original size and, in the case of the largest instance we can obtain a reduction that at least provides some information about which nodes have the same centrality. Such information could not be retrieved by calculating the centrality directly on the original multiplex because of the computational cost.

Table 1. Experimental results. Columns show the ID of the instance, the number of nodes (N), the number of layers (L), the number of nodes in the reduced model (rN), the number of layers in the reduced model (rL), the time spent to compute the centrality measure using the original model (Cen), the time spent to do the BDE reduction (BDE) and the time spent to compute the centrality measure using the reduced model (rCen).

Undirected instances								Directed instances							
ID	N	L	rN	rL	Cen(s)	BDE(s)	rCen(s)	ID	N	L	rN	rL	Cen(s)	BDE(s)	rCen(s)
(1)	16	2	16	2	0.26	0.00	-	(18)	461	5	276	5	0.18	0.01	0.08
(2)	61	5	61	5	0.07	0.01	-	(19)	1005	5	137	5	0.51	0.02	0.13
(3)	368	3	366	3	0.14	0.01	0.15	(20)	1203	3	994	3	1.04	0.04	0.94
(4)	450	37	374	37	0.67	0.03	0.55	(21)	2640	6	1264	6	3.50	0.06	1.19
(5)	514	16	351	16	1.28	0.18	0.52	(22)	3879	6	2372	6	7.90	0.10	4.16
(6)	14488	13	8008	13	192.53	0.66	83.71	(23)	4092	7	3613	7	53.71	0.75	59.47
Directed Instances															
ID	N	L	rN	rL	Cen(s)	BDE(s)	rCen(s)								
(7)	21	3	21	3	0.05	0.00	-	(24)	6570	7	6087	7	494.21	10.14	428.68
(8)	29	3	29	3	0.06	0.00	-	(25)	6980	7	4527	7	32.74	0.11	17.21
(9)	39	4	39	4	0.07	0.01	-	(26)	7747	7	4887	7	35.99	0.30	19.58
(10)	71	3	71	3	0.14	0.02	-	(27)	8215	7	7397	7	76.42	0.45	72.34
(11)	105	3	11	3	0.06	0.00	0.03	(28)	18222	7	13978	7	747.52	1.97	562.42
(12)	155	5	90	5	0.08	0.00	0.04	(29)	214	364	214	364	49.17	0.50	–
(13)	216	4	46	4	0.09	0.00	0.03	(30)	88804	3	37773	3	3359.26	6.68	1316.53
(14)	246	3	242	3	0.21	0.01	0.18	(31)	102439	3	48018	3	6188.51	5.36	3296.76
(15)	279	3	279	3	0.70	0.05	-	(32)	327707	3	63136	3	21289.98	2.51	1806.65
(16)	325	4	162	4	0.12	0.01	0.09	(33)	438537	3	180443	3	>10 h	12.05	>3 h
(17)	367	7	62	7	0.13	0.00	0.09								

5 Conclusions and Future Work

In this paper we have related an extension of eigenvector centrality on undirected and (possibly) weighted multiplex MLNs to BDE, an exact model reduction technique for dynamical systems. We have shown that we can use a BDE-inspired technique to introduce a framework that allows to reduce MLNs while preserving the $\boldsymbol{f}$-eigenvector centrality measure. The relevance of the result was demonstrated by efficiently computing reduction of real-world MLNs and by showing a speed up in the computation of such measure of interest. Throughout this work we considered exact reductions although it is worth noting that one of the possible future directions is to consider approximate reductions which are already prominent in the study of clustering in networks [7] and approximate lumping in agent-based models [32]. Future work will focus on the extension of these results to multiplex MLNs that feature directed layers. Other directions will include extending the framework to other centrality measures and other families of MLNs. Thanks to the theory established in this paper, we can naturally approach the study of approximate versions of this reduction technique because it is known that exact reductions might not yield significant reductions in very asymmetric real-world case studies. Moreover, this framework is a very versatile cornerstone work that, with few appropriate changes, can be easily modified to deal with other types of notions such as extensions of role equivalence on MLNs.

Acknowledgements. The authors' research is supported by the Ministry of Science, Research and the Arts of the state of Baden-Württemberg, and the DFG Centre of Excellence 2117 'Centre for the Advanced Study of Collective Behaviour' (ID: 422037984). The authors would like to thank Ulrik Brandes and Giacomo Rapisardi for the inspiring discussions on the topic, Andrea Vandin for the support and the insights on the use of the tool ERODE and the anonymous reviewers for their suggestions and comments.

References

1. Banerjee, A., Chandrasekhar, A., Duflo, E., Jackson, M.: The diffusion of microfinance. Science **341**, 1236498 (2013)
2. Barrett, L., Henzi, P., Lusseau, D.: Taking sociality seriously: the structure of multi-dimensional social networks as a source of information for individuals. Phil. Trans. Roy. Soc. Lond. Ser. B Biol. Sci. **367**, 2108–2118 (2012)
3. Battiston, F., Nicosia, V., Latora, V.: Structural measures for multiplex networks. Phys. Rev. E Stat. Nonlinear Soft Matter Phys. **89**, 032804 (2014)
4. Bloch, F., Jackson, M., Tebaldi, P.: Centrality measures in networks. SSRN Electron. J. (2016)
5. Bonacich, P.: Factoring and weighting approaches to status scores and clique identification. J. Math. Sociol. **2**(1), 113–120 (1972)
6. Bonacich, P.: Power and centrality: a family of measures. Am. J. Sociol. **92**(5), 1170–1182 (1987)
7. Brandes, U., et al.: On modularity clustering. IEEE Trans. Knowl. Data Eng. **20**, 172–188 (2008)

8. Brandes, U., Lerner, J.: Structural similarity: spectral methods for relaxed block-modeling. J. Classif. **27**(3), 279–306 (2010)
9. Cardelli, L., Tribastone, M., Tschaikowski, M., Vandin, A.: Efficient syntax-driven lumping of differential equations. In: Chechik, M., Raskin, J.-F. (eds.) TACAS 2016. LNCS, vol. 9636, pp. 93–111. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49674-9_6
10. Cardelli, L., Tribastone, M.,Tschaikowski, M., Vandin, A.: Symbolic computation of differential equivalences. In: 43st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL) (2016)
11. Cardelli, L., Tribastone, M., Tschaikowski, M., Vandin, A.: ERODE: a tool for the evaluation and reduction of ordinary differential equations. In: Legay, A., Margaria, T. (eds.) TACAS 2017. LNCS, vol. 10206, pp. 310–328. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-662-54580-5_19
12. Cardelli, L., Tribastone, M., Tschaikowski, M., Vandin, A.: Maximal aggregation of polynomial dynamical systems. PNAS **114**(38), 10029–10034 (2017)
13. Cardillo, A., et al.: Emergence of network features from multiplexity. Sci. Rep. **3**, 1344 (2013)
14. Christakis, N., Fowler, J.: Social network sensors for early detection of contagious outbreaks. PloS One **5**, e12948 (2010)
15. De Domenico, M., Nicosia, V., Arenas, A., Latora, V.: Structural reducibility of multilayer networks. Nat. Commun. **6**, 6864 (2015)
16. De Domenico, M., Solé-Ribalta, A., Gómez, S., Arenas, A.: Navigability of interconnected networks under random failures. Proc. Natl. Acad. Sci. **111**, 8351–8356 (2014)
17. De Domenico, M., et al.: Mathematical formulation of multi-layer networks. Phys. Rev. X **3**, 07 (2013)
18. De Domenico, M., Solé-Ribalta, A., Omodei, E., Gomez, S., Arenas, A.: Centrality in interconnected multilayer networks. Phys. D Nonlinear Phenom. **323**, 11 (2013)
19. De Domenico, M., Solé-Ribalta, A., Omodei, E., Gomez, S., Arenas, A.: Ranking in interconnected multilayer networks reveals versatile nodes. Nat. Commun. **6**, 6868 (2015)
20. De Domenico, M., Lancichinetti, A., Arenas, A., Rosvall, M.: Identifying modular flows on multilayer networks reveals highly overlapping organization in social systems. ArXiv, abs/1408.2925 (2015)
21. Farine, D., Aplin, L., Sheldon, B., Hoppitt, W.: Interspecific social networks promote information transmission in wild songbirds. Proc. Biol. Sci. Roy. Soc. **282**, 20142804 (2015)
22. Feret, J., Henzinger, T., Koeppl, H., Petrov, T.: Lumpability abstractions of rule-based systems. Theor. Comput. Sci. **431**, 137–164 (2012)
23. Franz, M., Altmann, J., Alberts, S.: Knockouts of high-ranking males have limited impact on baboon social networks. Curr. Zool. **61**, 107–113 (2015)
24. Ganguly, A., Petrov, T., Koeppl, H.: Markov chain aggregation and its applications to combinatorial reaction networks. J. Math. Biol. **69**(3), 767–797 (2014)
25. Gazda, S., Iyer, S., Killingback, T., Connor, R., Brault, S.: The importance of delineating networks by activity type in bottlenose dolphins (Tursiops truncatus) in cedar key, Florida. Roy. Soc. Open Sci. **2**, 140263 (2015)
26. Granell, C., Gomez, S., Arenas, A.: Dynamical interplay between awareness and epidemic spreading in multiplex networks. Phys. Rev. Lett. **111**, 128701 (2013)
27. Iacobelli, G., Tribastone, M., Vandin, A.: Differential bisimulation for a markovian process algebra. In: MFCS, pp. 293–306 (2015)

28. Jackson, M.: Social and Economic Networks (2008)
29. Johnson, J., Borgatti, S., Everett, M.: Analyzing Social Networks (2013)
30. Kapferer, B.: Strategy and transaction in an African factory: African workers and Indian management in a Zambian town. Manchester Univ. Press **43**(4), 362–363 (1972)
31. Katz, L.: A new status index derived from sociometric analysis. Psychometrika **18**(1), 39–43 (1953)
32. KhudaBukhsh, W.R., Auddy, A., Disser, Y., Koeppl, H.: Approximate lumpability for markovian agent-based models using local symmetries. J. Appl. Probab. **56**, 04 (2018)
33. Krackhardt, D.: Cognitive social structures. Social Netw. **9**(2), 109–134 (1987)
34. Lazega, E.: The Collegial Phenomenon: The Social Mechanisms of Cooperation among Peers in a Corporate Law Partnership. Oxford University Press, Oxford (2001)
35. Lerner, J.: Role assignments. In: Network Analysis: Methodological Foundations [Outcome of a Dagstuhl Seminar, 13–16 April 2004], pp. 216–252 (2004)
36. Magnani, M., Micenková, B., Rossi, L.: Combinatorial analysis of multiple networks. CoRR, abs/1303.4986 (2013)
37. McKay, B.: Practical graph isomorphism. Congressus Numerantium **30**, 45–87 (1981)
38. McKay, B., Piperno, A.: Practical graph isomorphism, II. J. Symb. Comput. **6009**, 94–112 (2013)
39. Newman, M.: Networks: An Introduction. Oxford University Press Inc., New York (2010)
40. Omodei, E., De Domenico, M., Arenas, A.: Characterizing interactions in online social networks during exceptional events. Front. Phys. **3**, 06 (2015)
41. Padgett, J.F., Ansell, C.K.: Robust action and the rise of the medici, 1400–1434. Am. J. Sociol. **98**(6), 1259–1319 (1993)
42. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: bringing order to the web. In: WWW 1999 (1999)
43. Robert Paige and Robert Endre Tarjan: Three partition refinement algorithms. SIAM J. Comput. **16**(6), 973–989 (1987)
44. Pappas, G.J.: Bisimilar linear systems. Automatica **39**(12), 2035–2047 (2003)
45. Snijders, T.A.B., Pattison, P.E., Robins, G.L., Handcock, M.S.: New specifications for exponential random graph models. Sociol. Methodol. **36**(1), 99–153 (2006)
46. Conde, L.S., Romance, M., Herrero, R., Flores, J., del Amo, A.G., Boccaletti, S.: Eigenvector centrality of nodes in multiplex networks. Chaos (Woodbury, N.Y.) **23**, 033131 (2013)
47. Stark, C., Breitkreutz, B.-J., Reguly, T., Boucher, L., Breitkreutz, A., Tyers, M.: Biogrid: a general repository for interaction datasets. Nucleic Acids Res. **34**, D535–D539 (2006)
48. Tognazzi, S., Tribastone, M., Tschaikowski, M., Vandin, A.: Differential equivalence yields network centrality. In: Margaria, T., Steffen, B. (eds.) ISoLA 2018, Part III. LNCS, vol. 11246, pp. 186–201. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03424-5_13
49. Tschaikowski, M., Tribastone, M.: Exact fluid lumpability for Markovian process algebra. In: CONCUR, pp. 380–394 (2012)
50. Tudisco, F., Arrigo, F., Gautier, A.: Node and layer eigenvector centralities for multiplex networks. SIAM J. Appl. Math. **78**(2), 853–876 (2018)
51. van der Schaft, A.J.: Equivalence of dynamical systems by bisimulation. IEEE Trans. Autom. Control **49**, 2160–2172 (2004)

52. Vickers, M., Chan, M.: Representing classroom social structure. Victoria Institute of Secondary Education, Melbourne (1981)
53. Stanley, W., Katherine, F.: Social Network Analysis: Methods and Applications, vol. 8. Cambridge University Press, Cambridge (1994)
54. Wei, X., Valler, N.B., Prakash, A., Neamtiu, I., Faloutsos, M., Faloutsos, C.: Competing memes propagation on networks: a case study of composite networks. ACM SIGCOMM Comput. Commun. Rev. **42**, 5–11 (2012)