UNIVERSITY OF OXFORD

Centrality measures in multilayer networks

MMath BE Extended Essay

Candidate Number
303269

Hilary 2015

Abstract

Network science is an interdisciplinary academic field that has applications in many disciplines including sociology, biology, economics, operations research, and computer science. ‘Multilayer’ networks allow multiple types of relationships to be represented in modelling. Recently, there have been some efforts to generalise ‘centrality’ measures—indicators that quantify the importance of nodes in a network—in order that they are also applicable in multilayer networks.

In this report, we put forward a proposition that enables existing tools for centrality measures defined by random walks in undirected multilayer networks with edge weights to be directly applicable to undirected multilayer networks with both nonnegative edge weights and nonnegative node weights. We then give detailed derivations and proofs for two approaches to calculating random-walk occupation centrality, PageRank centrality, random-walk betweenness centrality and random-walk closeness centrality in multilayer networks with edge weights. One approach exploits the multilayer structure and the other generalises centrality measures directly from monoplex (i.e. single-layer) networks. We compare the two methods by ranking economies in the international trade system and through general numerical simulations.

Acknowledgements

Firstly, I would like to thank my supervisors for their consistent encouragement and advice from the very beginning of the project, which has strengthened my interest in network science. I am grateful to Dr Mikko Kivelä for creating and developing Multilayer Networks Library for Python, and Dr John D. Hunter and other contributors to the MATPLOTLIB library for Python. I also thank Dr Robert C. Feenstra and other contributors to the documentation, 'World Trade Flows: 1962–2000', as well as the World Bank for publishing data for gross domestic product per capita in year 2000.

Contents

List of Figures

1 Introduction

A *network* is a set of items (called *nodes* or *vertices*) with connections (called *edges* or *links*) between them. Many scientifically interesting systems consist of individual components joined together in some way, and networks are an abstract means of representing patterns of connections between components [28]. One of the most famous and best studied examples of a network is the Internet, a net in which the nodes are computers and the edges are wires between them [12]. Understanding the pattern of connections between components is essential to studying and controlling the behaviour of a system [28]. For example, the structure of the Internet affects how data is transmitted between computers over the Internet and the efficiency of such process [28].

The study of networks uses ideas from *graph theory*, and *network theory* has become an important vehicle for studying the pattern of connections between components in social, biological, physical, information, and engineering sciences [3, 5, 24, 28]. The earliest studies of networks generally simplified systems as *ordinary graphs* in which edges are static and unweighted and a pair of nodes are usually linked by at most one edge [28]. Such simplification may fail to capture some important features of real-world networks and possibily lead to misleading results [4]. The last decade has witnessed an increasing amount of papers studying networks whose structures are irregular, complex and dynamically dependent on time with attention shifting from research on small networks to that on large-scale networks [5]. In particular, *multilayer networks* have recently attracted rapidly growing interest and research because they allow multiple types of interactions between each pair of nodes to be considered [24]. One natural context for the application of multilayer networks is in social sciences. For example, two Oxford students may have several different types of relationships: friendship, being from the same College, studying the same degree, membership of the same sports club, and so on. More examples of using multilayer networks to describe real-life phenomena can be found in the review article [24].

Centrality, a measure of the importance of nodes in a network, comes from the discipline of *social network analysis* and has become a fundamental concept in network science with applications in a range of disciplines [6, 28]. One of the best known examples of centrality is *PageRank* proposed by Page et al. [30], which uses the hyperlink structure of the World Wide Web to rank web pages by their overall relative importance. Centrality has also been applied to understand employment opportunities [17], status in monkey-grooming networks [33, 34], differential growth rates among medieval cities [31], political integration in the context of the diversity of Indian social life [7], and much more. Several methods have been adopted in the development of various centrality measures, including *degree centrality*, *random-walk occupation centrality*, *betweenness*, *closeness*, *eigenvector centrality*, PageRank, *information centrality*, *Katz centrality*, *hyperlink-induced topic search* (*HITS*) *cen-*

trality, and the *rush index*, with some defined according to the topological structure of a network and some using dynamical processes. In particular, random walks have often been a popular vehicle to define centrality measures in *monoplex networks* (i.e. networks that have only one type of edges) because it is a simple dynamical process that abstracts numerous real-world phenomena and can be used to approximate other diffusion processes [11]. We give the definitions of random-walk occupation centrality, PageRank, random-walk betweenness centrality and random-walk closeness centrality in monoplex networks in Chapter 2 and study these centrality measures in weighted undirected multilayer networks in Chapter 3.

There has been considerable effort to generalise familiar tools from monoplex networks to multilayer networks; this includes studying centrality measures in the context of multilayer networks [4, 8, 10, 11, 13, 18, 24, 36]. Multilayer structures can have important effects on centrality measures, and ignoring multilayer structures can lead to different ranking results than what one obtains for multilayer networks [11]. For example, Figure 1.1 illustrates that the *inter-layer* edges of a multilayer network can have significant effect on random walks and thus on centrality measures defined using random-walk processes [10]. Generalising centrality measures from monoplex networks to multilayer networks is not trivial [11]. When ranking nodes in a multilayer network, the key question to be addressed is how one should take into consideration all the different types of edges, not all of which have the same importance [36].

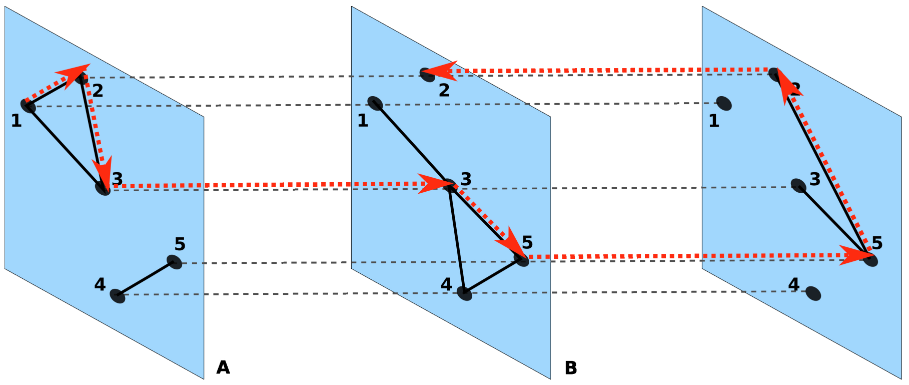


Figure 1.1: Schematic of a random walk (the red dotted path) in a multilayer network. At each step, a random walker can either follow an intra-layer edge (a solid line) or an inter-layer edge (a grey dashed line). The multilayer structure allows a random walker to move between nodes that are adjacent in one layer but not in another. The figure is inspired by [11].

To the best of our knowledge, the vast majority of existing papers that study centrality measures concern only the cases where the diversity of edge properties is described by assigning different *edge weights*, whereas possible heterogeneity of node properties is often neglected. Particularly, in the context of multilayer networks, all of the research so far has been limited to multilayer networks whose nodes are indistinguishable except by the properties of their incident edges.

Real-world networks often possess properties that are natural to be represented through their nodes instead of edges. For example, one straightforward way to rank economies in the world by their economic influence is to view the international trade system as a monoplex network in which economies are represented as nodes and trading values between pairs of economies give the edge weights. However, a country that is important in international trade for one commodity may be insignificant for another. Hence, it is natural to analyse the international trade system as a multilayer network where commodity categories are taken as layers. Another important index for ranking economies is Gross Domestic Product (GDP) per capita, which is natural to be incorporated as a property of the nodes instead of that of the edges. Reference [19] ranked economies by fitting the international trade system to an undirected monoplex network with GDP as *node weights* and acknowledged that adding edge weights would be a much more realistic model. We propose to model the international trade system as an undirected multilayer network as just described and, in addition, incorporate GDP per capita as node weights. We discuss the details and results of this example in Chapter 4.

The remainder of the report is organised as follows. Chapter 2 introduces some important definitions and methods for studying multilayer networks. Chapter 3 first proposes a method that enables existing tools for centrality measures defined by random walks in undirected multilayer networks with edge weights to be directly applicable to undirected multilayer networks with both nonnegative edge weights and nonnegative node weights. We then use two methods [11, 29] to study random-walk occupation centrality, PageRank, random-walk betweenness centrality and random-walk closeness centrality in undirected multilayer networks with edge weights, and give detailed derivations and proofs that are only partially included in the original papers. In Chapter 4, we use the two methods discussed in Chapter 3 to rank economies in the international trade system in 2000, and compare the ranking results with the ones obtained by modelling the trade system as a monoplex network. In Chapter 5, we perform computer simulations for multiplex networks, and compare the ranking results.

2 Preliminary information

2.1 Definitions

We introduce the following concepts, following [24].

Definition 1. A *graph* G is an ordered pair of sets (V, E), where V is a set of *nodes* and $E \subseteq V \times V$ is a set of *edges* that connect pairs of nodes. A *single-layer network* (also called a *monoplex network*) is a graph.

Definition 2. Two nodes are *adjacent* if there exists an edge between them. We say that these two nodes are *neighbours* of each other and the edge is *incident* to each of the nodes. The *degree* of a node is defined as the number of its neighbours.

Definition 3. Let V be a set of nodes as defined in a graph, and let *layer* L be the set of the types of relationships between pairs of nodes. Define $V_M \subseteq V \times L$ as the subset that contains only the node-layer combinations such that a *node-layer tuple* $(v, l) \in V_M$ if and only if v is present in layer l. Let $E_M \subseteq V_M \times V_M$ be the subset of edges between *node-layers*. A *multilayer network* $\boldsymbol{M}$ is a quadruplet $\boldsymbol{M} = (V_M, E_M, V, L)$.

Definition 4. A multilayer network is a *multiplex network* if all layers contain all nodes and all of the inter-layer edges are between nodes and their counterparts in other layers. A multiplex network has *categorical couplings* if each node is adjacent to all of its counterparts in all layers.

Remark. Definitions 3 and 4 are simplified versions that are commonly used in research; more general definitions are given in [24]. We consider undirected multilayer networks in this report.

Figures 2.1, 2.2, and 2.3 give examples of a multilayer network, a multiplex network and a monoplex network respectively. In Figure 2.1, the multilayer network $\boldsymbol{M} = (V_M, E_M, V, L)$ has the set of nodes $V = \{1, 2, 3, 4, 5\}$; three layers (A, B, and C); and the set of node-layer tuples $V_M = \{(1, A), (2, A), (3, A), (4, A), (1, B), (2, B), (3, B), (4, B), (2, C), (3, C), (4, C), (5, C)\} \subseteq V \times L$.

2.1.1 Tensor representations

The main benefits of studying multilayer networks using tensor algebra are two-fold: it gives concise mathematical representation, and it leads to natural generalisations of numerous network diagonostics from monoplex networks to multilayer networks [9].

For a finite n-dimensional space, we give the definition of tensors from [32]:

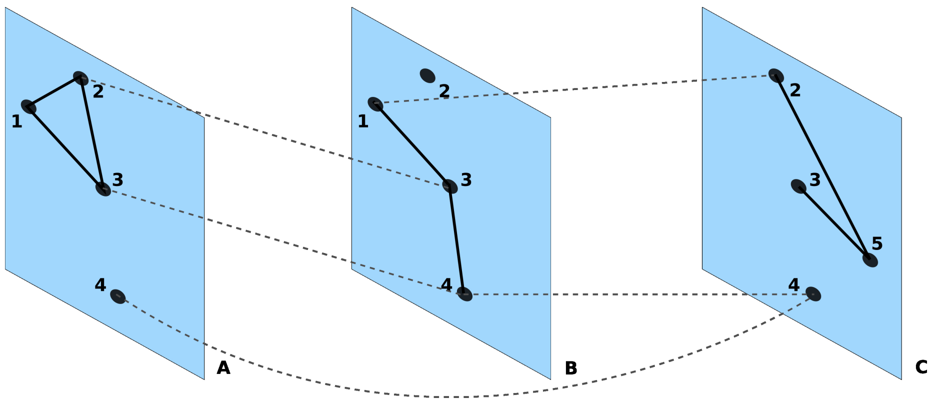


Figure 2.1: An example of the most general type of multilayer network. Intra-layer edges are represented by solid lines, and inter-layer edges by dashed lines. Note that any pair of node-layer tuples can be connected by an intra-layer edge if they exist on the same layer or by an inter-layer edge if they exist on different layers. The figure is inspired by [24].

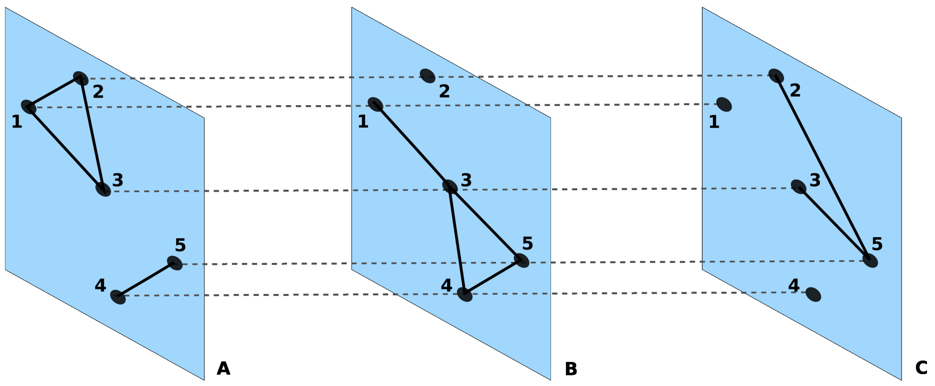


Figure 2.2: An example of a multiplex network. Intra-layer edges are represented by solid lines, and inter-layer edges by dashed lines. Note that inter-layer edges can only possibly exist between a node and its counterpart in a different layer. The figure is inspired by [24].

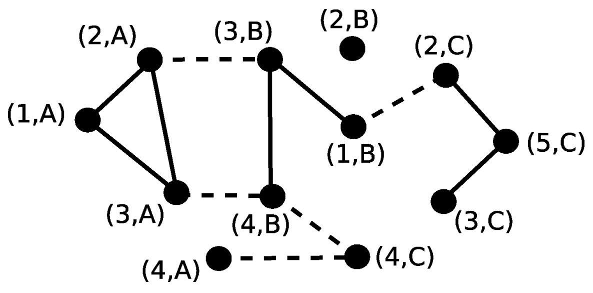


Figure 2.3: An example of a monoplex network, which is the underlying graph of the multilayer network in Figure 2.1. Intra-layer edges are represented by solid lines, and inter-layer edges by dashed lines. The figure is inspired by [24].

Definition 5. Suppose that $\boldsymbol{X}$ is a mathematical or physical entity, which, when it is associated with a basis, can be represented by a set of scalars $X^{i_1\cdots i_r}_{j_1\cdots j_s}$ (called *scalar components*). We call $\boldsymbol{X}$ as a *tensor* of *type* (r, s) if the scalar components of $\boldsymbol{X}$ obey the tensor transformation law:

$$X^{i_1\cdots i_r}_{j_1\cdots j_s} = S^{i_1}_{h_1} \cdots S^{i_r}_{h_r} T^{k_1}_{j_1} \cdots T^{k_s}_{j_s} \tilde{X}^{h_1\cdots h_r}_{k_1\cdots k_s}, \tag{2.1}$$

under a change of basis, where $S^{i_p}_{h_p}$ and $T^{i_q}_{h_q}$ are matrices for $p \in \{1 \dots r\}$ and $q \in \{1 \dots s\}$. The *rank* (or *order*) of $\boldsymbol{X}$ is $r + s$.

Scalars are 0th-order tensors. Vectors are 1st-order tensors. Linear maps are 2nd-order tensors.

Equation 2.1 adopts the *Einstein notation*: when an index is repeated in an expression, it implies summation of that expression over all the values of the index. Thus, Equation 2.1 is equivalent to

$$X^{i_1\cdots i_r}_{j_1\cdots j_s} = \sum_{h_1=1}^{n} \cdots \sum_{h_r=1}^{n} \sum_{k_1=1}^{n} \cdots \sum_{k_s=1}^{n} S^{i_1}_{h_1} \cdots S^{i_r}_{h_r} T^{k_1}_{j_1} \cdots T^{k_s}_{j_s} \tilde{X}^{h_1\cdots h_r}_{k_1\cdots k_s}. \tag{2.2}$$

In the rest of the report, we do not specify bases for tensors because tensors exist independently of any basis [22].

Consider a monoplex network with a set of N nodes $V = \{n_1, \dots, n_N\}$. For $k, l \in \{1, \dots, N\}$, let w_{kl} denote the intensity of the relationship (called the *edge weight*) between node n_k and node n_l and let the set of 2nd-order tensors $E^i_j(k, l)$ represent the canonical basis for the space $\mathbb{R}^{N\times N}$ (i.e. all of the scalar components of $E^i_j(k, l)$ are 0 except for the (k, l) scalar component, which is 1). The adjacency tensor of the monoplex network can then be written as

$$W^i_j = \sum_{k,l=1}^{N} w_{kl} E^i_j(k, l). \tag{2.3}$$

Adopting the convention proposed in [11], we denote nodes with Latin letters and layers with Greek letters for clarity. Let $C^i_j(\gamma, \delta)$ be the inter-layer adjacency tensor for edges between layers γ and δ, and let $w_{kl}(\gamma, \delta)$ be the edge weight between node-layers (k, γ) and (l, δ). We then have

$$C^i_j(\gamma, \delta) = \sum_{k,l=1}^{N} w_{kl}(\gamma, \delta) E^i_j(k, l). \tag{2.4}$$

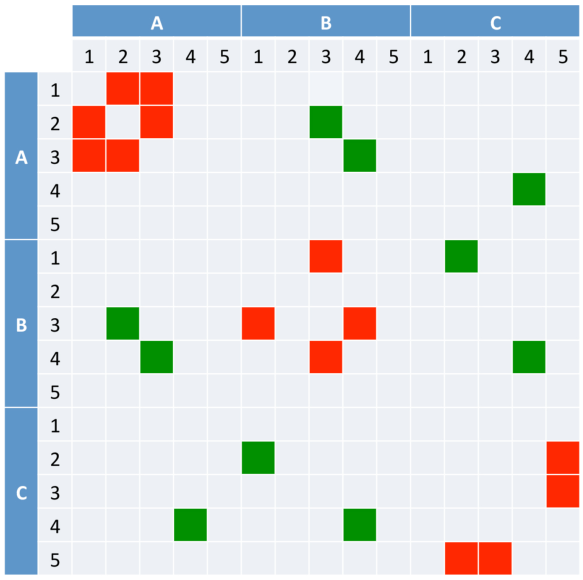
Note that $C^i_j(\gamma, \gamma)$ is the adjacency matrix for the layer γ. Let $E^\alpha_\beta(\gamma, \delta)$ be the canonical basis for the space $\mathbb{R}^{K\times K}$, and let $\varepsilon^{i\alpha}_{j\beta}(k, l, \gamma, \delta) \equiv E^i_j(k, l) E^\alpha_\beta(\gamma, \delta)$. Now we can derive the 4th-order tensor representation of a multilayer network of any type by

$$
\begin{aligned}
M^{i\alpha}_{j\beta} &= \sum_{\gamma,\delta=1}^{K} C^i_j(\gamma,\delta) E^\alpha_\beta(\gamma,\delta) \\
&= \sum_{\gamma,\delta=1}^{K} \left[\sum_{k,l=1}^{N} w_{kl}(\gamma,\delta) E^i_j(k,l) \right] E^\alpha_\beta(\gamma,\delta) \qquad (2.5) \\
&= \sum_{\gamma,\delta=1}^{K} \sum_{k,l=1}^{N} w_{kl}(\gamma,\delta) \varepsilon^{i\alpha}_{j\beta}(k,l,\gamma,\delta),
\end{aligned}
$$

In an undirected multilayer network, we have $w_{kl}(\gamma,\delta) = w_{lk}(\delta,\gamma)$.

2.1.2 Supra-adjacency representations

A *supra-adjacency matrix* for a multilayer network $\boldsymbol{M} = (V_M, E_M, V, L)$ is the adjacency matrix for the underlying graph $G_M = (V_M, E_M)$ of $\boldsymbol{M}$ [24]. Figure 2.4 gives an illustration of the supra-adjacency matrix for the the multilayer network in Figure 2.1.



		A					B					C				
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
A	1		🟥	🟥												
	2	🟥		🟥					🟩							
	3	🟥	🟥							🟩						
	4														🟩	
	5															
B	1								🟥				🟩			
	2															
	3		🟩				🟥			🟥						
	4			🟩					🟥						🟩	
	5															
C	1															
	2						🟩									🟥
	3															🟥
	4				🟩					🟩						
	5												🟥	🟥		

Figure 2.4: An illustration of the supra-adjacency matrix for the multilayer network in Figure 2.1, which is also the adjacency matrix for the underlying graph in Figure 2.3. The three blocks on the main diagonal of the matrix correspond to intra-layer adjacency matrices, and the off-diagonal blocks correspond to inter-layer adjacency matrices. Suppose that the multilayer network in Figure 2.1 is unweighted, then thered and green elements are 1, and the others are 0. The figure is inspired by [24].

Supra-adjacency matrices are better studied than tensors, and they are natural for representing multilayer networks that do not contain all nodes in all layers [24].

On the other side, supra-adjacency matrices can only be obtained by 'flattening' adjacency tensors of multilayer networks, which requires a lot of labelling to record the order of layers if the multilayer network is large and we do not want to lose this information after flattening [24]. We use adjacency tensors to study centrality measures in multilayer networks in Chapter 3, and, we use supra-adjacency matrices for doing computations in Chapters 4 and 5.

2.2 Random walks in multilayer networks

In this report, we consider discrete-time, finite-state random walks in multilayer networks without self-loops (that is, there is no edge that connects a node-layer to itself). Let us first define some notation that we will need later.

Consider a multilayer network $\boldsymbol{M} = (V_M, E_M, V, L)$. Denote node i in layer α by node-layer (i, α) and the edge weight between (i, α) and (j, β) by $w_{ij}(\alpha, \beta)$. Define the *node strength* $s_{i\alpha}$ of (i, α) as the sum of the weights of all edges, including both intra-layer and inter-layer edges, that are incident to (i, α) [9]. Let $n_{i\alpha}$ be the node weight of (i, α), and let $T^{i\alpha}_{j\beta} \in [0, 1]$ be the *transition probability* from (i, α) to (j, β).

2.2.1 Multilayer networks with edge weights

For multilayer networks with edge weights, Reference [11] uses random walks defined by

$$T^{i\alpha}_{j\beta} = \frac{w_{ij}(\alpha, \beta)}{\max(s_{i\alpha}, \epsilon)}, \tag{2.6}$$

where $\epsilon > 0$ is a small constant, for example, $\epsilon \equiv \min\{w_{ij}(\alpha, \beta) : (i, \alpha), (j, \beta) \in V_M\}$. In other words, the transition probability is proportional to the edge weight between the current node and its neighbour. Note that $T^{i\alpha}_{j\beta} = 0$ if (i, α) is isolated (that is, $w_{ij}(\alpha, \beta) = 0$ for all (j, β)). Let $p_{i\alpha}(t) \in [0, 1]$ be the probability of finding a random walker at (i, α) at time t. Conditioning on the position of the random walker at time t, we have

$$p_{j\beta}(t + 1) = \sum_{(i,\alpha)\in V_M} T^{i\alpha}_{j\beta} p_{i\alpha}(t). \tag{2.7}$$

Let $\pi_{i\alpha} \equiv \lim_{t\to\infty} p_{i\alpha}(t)$ be the asymptotic probability of finding a random walker at (i, α) in the limit $t \to \infty$. Note that the random walk that we just defined is a *time-homogeneous Markov chain*. In other words, the transition matrix T is the same after each step, so the k-step transition probability is the kth power T^k of T. Recall the following theorem from Part A Probability (Theorem 6.1 of lecture notes by J. Martin, version of 23 December 2014): an *irreducible* Markov chain has a unique *stationary distribution* if and only if the chain is *positive recurrent*. Thus, $\pi_{i\alpha}$ is well-defined if the random walk is irreducible and possitive recurrent. In this case, we have $\pi_{i\alpha} \propto s_{i\alpha}$ [28].

We expect it to be true that the value of $\pi_{i\alpha}$ tends to increase when (i,α) has more incident edge weights, and/or more neighbours, and/or fewer second neighbours (i.e. neighbours of neighbours).

2.2.2 Multilayer networks with node weights

For random walks in multilayer networks with node weights, we introduce the definition

$$T^{i\alpha}_{j\beta} = \frac{n_{i\alpha} + n_{j\beta}}{\max(N_{i\alpha}, \epsilon)}, \tag{2.8}$$

where $\epsilon > 0$ is a small constant and $N_{i\alpha} = \sum_{((i,\alpha),(k,\gamma)) \in E_M} (n_{i\alpha} + n_{k\gamma})$. We expect it to be true that the value of $\pi_{i\alpha}$ tends to increase when (i,α) has higher node weight, and/or more neighbours, and/or higher adjacent node weights, and/or fewer second neighbours, and/or second neighbours with lower node weights.

2.2.3 Multilayer networks with both edge weights and node weights

Let $\boldsymbol{M}$ be a multilayer network with both nonnegative edge weights and nonnegative node weights. We want to define a random walk in $\boldsymbol{M}$ such that the transition probability is proportional to some function $f_M \colon V_M \times V_M \to \mathbb{R}^{\geq 0}$, where the subscript M of f_M indicates the dependence of the function on the multilayer structure, edge weights, and node weights of $\boldsymbol{M}$. We also want the value of $\pi_{i\alpha}$ to tend to increase when (i,α) has higher node weight, and/or more neighbours, and/or larger incident edge weights, and/or higher adjacent node weights, and/or fewer second neighbours, and/or second neighbours with lower node weights, .

In addition to the above conditions, we also require the function f to satisfy two extra conditions: (i) $\frac{\partial f}{\partial n_{i\alpha}} \geq 0$ and $\frac{\partial f}{\partial w_{ij}(\alpha,\beta)} \geq 0$ for all (i,α),$(j,\beta) \in V_M$; and (ii) $f((i,\alpha),(j,\beta)) = f((j,\beta),(i,\alpha))$.

Among all of the functions that satisfy these conditions, the simplest one is

$$f((i,\alpha),(j,\beta)) = (n_{i\alpha} + n_{j\beta}) w_{ij}(\alpha,\beta). \tag{2.9}$$

We observe two additional benefits of using Equation 2.9 as detailed below.

First, the transition probability from (i,α) to (j,β) is

$$T^{i\alpha}_{j\beta} = \frac{f((i,\alpha),(j,\beta))}{\max(F_{i\alpha}, \epsilon)}, \tag{2.10}$$

where ϵ is a small constant and $F_{i\alpha} = \sum_{(k,\gamma) \in V_M} f((i,\alpha),(k,\gamma))$. Note additionally that $w_{ik}(\alpha,\gamma) = 0$ if (i,α) and (k,γ) are not adjacent to each other. Thus, Equation 2.10 can be expanded to

$$T^{i\alpha}_{j\beta} = \frac{(n_{i\alpha} + n_{j\beta}) w_{ij}(\alpha,\beta)}{\max\left\{\sum_{(k,\gamma) \in V_M} (n_{i\alpha} + n_{k\gamma}) w_{ik}(\alpha,\gamma), \epsilon\right\}}. \tag{2.11}$$

Equation 2.11 reduces to Equation 2.6 in multilayer networks without node weights (i.e. when $n_{i\alpha} = n_{j\beta}$ for all $(i,\alpha),(j,\beta) \in V_M$). Equation 2.11 is also consistent with Equation 2.8 in multilayer networks without edge weights (i.e. when $w_{ij}(\alpha,\beta) = w_{kl}(\gamma,\delta)$ for all $(i,\alpha),(j,\beta),(k,\gamma),(l,\delta) \in V_M$).

Second, node weights and edge weights might differ considerably in magnitude, and Equation 2.9 is the simplest form that allows node weights and edge weights to have equal effect on ranking results.

3 Centralities in weighted multilayer networks

Having defined relevant concepts and random walks, we now propose and prove the following proposition, which is an original contribution of the report that enables tools for centrality measures defined by random walks in undirected multilayer networks with edge weights to be directly applicable to undirected multilayer networks with both nonnegative edge weights and nonnegative node weights.

PROPOSITION 6. *Consider a problem* P*: Let* $\boldsymbol{M} = (V_M, E_M, V, L)$ *be an undirected multilayer network with both nonnegative edge weights and nonnegative node weights; find a centrality measure defined by random walks in* $\boldsymbol{M}$*, in which the transition probability is a function of both the edge weights and the node weights of* $\boldsymbol{M}$*. For every problem* P*, there exists an equivalent problem* P′ *that concerns only undirected multilayer networks with edge weights and no node weights.*

PROOF. Consider an undirected multilayer network $\boldsymbol{M} = (V_M, E_M, V, L)$ with nonnegative edge weights $w_{ij}(\alpha, \beta)$ and nonnegative node weights $n_{i\alpha}$. Let $\hat{\boldsymbol{M}} = (V_M, E_M, V, L)$ be an undirected multilayer network with edge weights defined by

$$\hat{w}_{ij}(\alpha, \beta) = f_M((i, \alpha), (j, \beta)), \tag{3.1}$$

where $f_M \colon V_M \times V_M \to \mathbb{R}^{\geq 0}$ is an appropriate function. By 'appropriate', we mean that function f and the corresponding asymptotic probability $\pi_{i\alpha}$ satisfy all of the conditions discussed in Subsection 2.2.3.

Let $\hat{s}_{i\alpha}$ denote the strength of node-layer (i, α) in the multilayer network $\hat{\boldsymbol{M}}$. The transition probability of a random walker going from (i, α) to (j, β) in $\hat{\boldsymbol{M}}$ is then given by

$$\hat{T}^{i\alpha}_{j\beta} = \frac{\hat{w}_{ij}(\alpha, \beta)}{\max(\hat{s}_{i\alpha}, \epsilon)}. \tag{3.2}$$

We have discussed in Subsection 2.2.3 that Equation 2.9 is consistent with the cases in which multilayer networks have only edge weights or node weights, and that random walks in $\hat{\boldsymbol{M}}$ are equivalent to those in $\boldsymbol{M}$ by the definition of function f. Hence centrality measures defined by random walks in $\hat{\boldsymbol{M}}$ are equivalent to those defined by random walks in $\boldsymbol{M}$. □

This allows us to study centrality measures in undirected multilayer networks that have edge weights but no node weights. We can then use Proposition 6 to study undirected multilayer networks with both nonnegative edge weights and nonnegative node weights.

In the rest of Chapter 3, we consider undirected multilayer networks $\boldsymbol{M} = (V_M, E_M, V, L)$ that have edge weights $w_{ij}(\alpha, \beta)$ and no self-loops. Suppose that the multilayer network $\boldsymbol{M}$ has K layers and that the maximum number of nodes on one layer of $\boldsymbol{M}$ is N. Let $M^{i\alpha}_{j\beta}$ be a scalar component of the adjacency tensor of $\boldsymbol{M}$ (see Equation 2.5). We adopt the Einstein notation in the rest of the chapter. Sections 3.1–3.4 are based on a 2013 paper by De Domenico et al. [11] unless stated otherwise.

3.1 Random-walk occupation centrality

Consider a classical random walker in a multilayer network $\boldsymbol{M}$ such that, at each step, the random walker only can move from one node to one of its neighbours via either an intra-layer edge or an inter-layer edge. Additionally, the transition probability is proportional to the edge weight between the current node and its neighbour.

Let $\boldsymbol{T}$ be the transition tensor with scalar components $T^{i\alpha}_{j\beta} \in [0, 1]$ given in Equation 2.6, and let $p_{i\alpha}(t) \in [0, 1]$ be a scalar component of the time-dependent tensor that gives the probability to find a random walker at (i, α). Define $\Pi_{i\alpha} \equiv \lim_{t\to\infty} p_{i\alpha}(t)$ as the random-walk occupation centrality of (i, α).

Random-walk occupation centrality measures the asymptotic probability of finding a random walker at a particular node-layer as time goes to infinity. The more likely that we find a random walker at a node-layer, the more important the node-layer is. Conditioning on the walker's location at the previous time step t gives the master equation

$$p_{j\beta}(t + 1) = T^{i\alpha}_{j\beta} p_{i\alpha}(t). \tag{3.3}$$

The steady-state solution of Equation (3.3), $\Pi_{i\alpha}$, can be obtained by taking the limit $t \to \infty$ first and then solving for $\Pi_{i\alpha}$ in the equation

$$T^{i\alpha}_{j\beta} \Pi_{i\alpha} = \Pi_{j\beta}. \tag{3.4}$$

For the computations in Chapters 4 and 5, we adopted the tensor-flattening method proposed by [11] to solve Equation 3.4. See Appendix A for details.

In order to write an explicit expression for $T^{i\alpha}_{j\beta}$, we need some more definitions. Let u^{α} and $u_{j\beta}$ be scalar components of the 1-vector and the 1-matrix, respectively. We then have $s_{i\alpha} = M^{j\beta}_{i\alpha} u_{j\beta}$. Define *strength tensor* as the 4th-order tensor with scalar components given by

$$D^{i\alpha}_{k\gamma} = \begin{cases} s_{i\alpha}, & \text{if } i = k \text{ and } \alpha = \gamma, \\ 0, & \text{otherwise.} \end{cases} \tag{3.5}$$

Then the explicit expression for $T^{i\alpha}_{j\beta}$ in tensor notation is

$$T^{i\alpha}_{j\beta} = M^{k\gamma}_{j\beta} \tilde{D}^{i\alpha}_{k\gamma}, \tag{3.6}$$

where

$$\tilde{D}^{i\alpha}_{k\gamma} = \begin{cases} 1/D^{i\alpha}_{k\gamma}, & \text{if } D^{i\alpha}_{k\gamma} \neq 0, \\ 0, & \text{if } D^{i\alpha}_{k\gamma} = 0, \end{cases} \tag{3.7}$$

is a scalar component of the normalising tensor.

We claim that the flattened square rank-2 tensor (see Appendix A) of $\boldsymbol{T}$ is almost a *left-stochastic* matrix and give the proof in Appendix B.

As in Subsection 2.2.1, we have $\Pi_{i\alpha} \propto s_{i\alpha}$, when $\Pi_{i\alpha}$ is well defined [28].

By aggregating the random-walk occupation centralities $\Pi_{i\alpha}$ of node i across all layers α, we get the aggregate random-walk occupation centrality of the node. Reference [11] claims that $\pi_i = \Pi_{i\alpha} u^\alpha$ can be considered as the random-walk occupation centrality for node i in $\boldsymbol{M}$ because each $\Pi_{i\alpha}$ accounts for the whole multilayer structure.

3.2 PageRank centrality

Now consider a random walker that has probability $r \in [0, 1)$ to behave like a classical random walker as defined by Equation 2.6 and probability $1-r$ to 'fly' to any node in the network randomly. The second action is called *teleporting*, and $1-r$ is known as the *teleportation rate*. The teleportation rate often takes the value of 0.15 in research for several reasons: iterative algorithms that approximate PageRank converge quickly at this value; if the teleportation rate is too close to 0, the ranking of the important nodes will be distored; and PageRank is more robust at larger teleportation rate [2]. More empirical results regarding the teleportation rate can be found in [16].

For the purpose of search ranking, the random teleportation process usually has a uniform distribution; in other words, the probability for a random walker to teleport to any node in the network is the same [2]. The corresponding transition tensor has scalar components given by

$$R^{i\alpha}_{j\beta} = \begin{cases} rT^{i\alpha}_{j\beta} + (1-r)/(NK), & \text{if } (i,\alpha) \text{ is not isolated,} \\ 1/(NK), & \text{if } (i,\alpha) \text{ is isolated.} \end{cases} \tag{3.8}$$

Similar to random-walk occupation centrality, PageRank is the steady-state probability of the above random-walk process. Let $\Omega_{i\alpha}$ be the PageRank for (i, α), and it then is a scalar component of the eigentensor of the transition tensor:

$$R^{i\alpha}_{j\beta}\Omega_{i\alpha} = \Omega_{j\beta} \tag{3.9}$$

Let $\boldsymbol{\Omega}$ be the tensor with scalar components $\Omega_{i\alpha}$, and let $\boldsymbol{R}$ be the tensor with scalar components $R^{i\alpha}_{j\beta}$. One way to solve for $\boldsymbol{\Omega}$ is to flatten $\boldsymbol{R}$ to a rank-2 tensor $\boldsymbol{R}_f$ with $NK \times NK$ scalar components and then use the *power iteration* (see Part A Numerical Analysis lecture notes by C. B. Macdonald, version of 9 February 2015) for $\boldsymbol{R}_f$. We discuss here the details that are not included in the lecture notes and [11]. Suppose that a nonnegative matrix $A \in \mathbb{R}^{n\times n}$ is *primitive*; that is, there exists some positive integer k such that the scalar components of A^k are all positive. By the *Perron-Frobenius Theorem* [26], there exists a positive real eigenvalue $\lambda_{\max}$ of A such that all other eigenvalues of A satisfy $|\lambda| \leq \lambda_{\max}$. Moreover, $\lambda_{\max}$ has algebraic (and hence geometric) multiplicity of 1, and has a strictly positive eigenvector. The theorem also ensures that the largest absolute value of the eigenvalues of a stochastic matrix is 1. We observe that $\boldsymbol{R}_f$ is a left-stochastic matrix. Thus, power iteration for $\boldsymbol{R}_f$ guarantees an eigenvector $\boldsymbol{\Omega}_f$ of $\boldsymbol{R}_f$, which corresponds to the eigenvalue 1. Note that $\boldsymbol{\Omega}_f$ is the flattened 1st-order tensor of $\boldsymbol{\Omega}$.

Similar to the random-walk occupation centrality, the PageRank for i is $\omega_i = \Omega_{i\alpha} u^\alpha$.

3.3 Random-walk betweenness centrality

Betweenness centrality measures the extent to which a node lies in paths between other nodes [28]. Recall the classical random walker defined by Equation 2.6. Let $\boldsymbol{T}_{[d]}$ be the transition tensor with absorbing node d in all layers; in other words, a random walker stops as soon as it hits node d in any layer. Tensor $\boldsymbol{T}_{[d]}$ then has scalar components[1]

$$(T_{[d]})^{i\alpha}_{j\beta} = \begin{cases} 0, & \text{if } i = d, \\ T^{i\alpha}_{j\beta}, & \text{if } i \neq d. \end{cases} \tag{3.10}$$

Given M random walks that start at node-layer (o, σ) and stop as soon as they hit node $[d]$, the expected number of times a random walk hits node-layer (j, β) is

$$(\tau_{[d]})^{o\sigma}_{j\beta} = \lim_{M\to\infty} \frac{1}{M} \sum_{m=1}^{M} \sum_{t=0}^{\infty} z^{o\sigma}_{j\beta}(t, m), \tag{3.11}$$

where $z^{o\sigma}_{j\beta}(t, m) = 1$ if walk m is at (j, β) at time t, and $z^{o\sigma}_{j\beta}(t, m) = 0$ otherwise.

The probability of being at (j, β) at time t, given that the walk starts at (o, σ), is

$$p^{o\sigma}_{j\beta}(t) = \lim_{M\to\infty} \frac{1}{M} \sum_{m=1}^{M} z^{o\sigma}_{j\beta}(t, m). \tag{3.12}$$

Let $\boldsymbol{T}_{[d]}$ be the transition tensor with scalar components defined in Equation 3.10, and let $\boldsymbol{\delta}$ be the tensor with scalar components

$$\delta^{o\sigma}_{j\beta} = \begin{cases} 1, & \text{if } j = o = \beta = \sigma, \\ 0, & \text{otherwise.} \end{cases} \tag{3.13}$$

Substituting Equation 3.12 into Equation 3.11 gives the mean number of times the random walker hits (j, β) regardless of the time step:

$$(\tau_{[d]})^{o\sigma}_{j\beta} = \sum_{t=0}^{\infty} p^{o\sigma}_{j\beta}(t) = \sum_{t=0}^{\infty} (\boldsymbol{T}^{t}_{[d]})^{o\sigma}_{j\beta} = \left[(\boldsymbol{\delta} - \boldsymbol{T}_{[d]})^{-1}\right]^{o\sigma}_{j\beta}, \tag{3.14}$$

Averaging over all possible starting layers σ and aggregating the walks that pass through node j in any layer gives

$$(\tau_{[d]})^{o}_{j} = \frac{1}{K} (\tau_{[d]})^{o\sigma}_{j\beta} u^{\beta} u_{\sigma}. \tag{3.15}$$

Finally, we obtain the random-walk betweenness centrality for j by averaging over all possible origin nodes and absorbing nodes:

$$\tau_j = \frac{1}{N(N-1)} \sum_{d=1}^{N} (\tau_{[d]})^{o}_{j} u_o. \tag{3.16}$$

[1]The idea comes from [11], but we corrected a typographical error. Some papers define an absorbing state as a state such that a random walker vanishes as soon as it hits the state. By this definition, the corresponding transition tensor is not stochastic. Reference [11] proposes that $(T_{[d]})^{i\alpha}_{j\beta} = 0$ if $j = d$. However, we prove in Appendix B that $(\boldsymbol{T}_f)^{i}_{j}$, the element in the jth row and ith column of $\boldsymbol{T}_f$, gives the probability for a random walker to jump from state i to state j. This implies that, if d is an absorbing node, then the dth column of $\boldsymbol{T}_f$ should be 0.

3.4 Random-walk closeness centrality

Closeness centrality measures the mean distance from a node to other nodes [28]. Consider an absorbing random walk as defined by Equation 3.10. Let

$$(p_{[d]})^{o\sigma}_{j\beta}(t) = \left(T^t_{[d]}\right)^{o\sigma}_{j\beta} \tag{3.17}$$

denote the probability of visiting node-layer (j,β) after exactly t time steps, given that the walk started at (o,σ) and will stop as soon as it hits node $[d]$. Let h be the first passage time for $[d]$. Then the probability that the walker is absorbed in $[d]$ no later than time t is

$$\left(q_{[d]}\right)^{o\sigma}(h \geq t) = u^{o\sigma} - \left(T^t_{[d]}\right)^{o\sigma}_{j\beta} u^{j\beta}. \tag{3.18}$$

Thus, the probability that the first-passage time for $[d]$ is exactly t is

$$\begin{aligned}\left(q_{[d]}\right)^{o\sigma}(h = t) &= \left(q_{[d]}\right)^{o\sigma}(h \geq t) - \left(q_{[d]}\right)^{o\sigma}(h \geq t-1)\\ &= \left[(T^{t-1}_{[d]}) - (T^t_{[d]})\right]^{o\sigma}_{j\beta} u^{j\beta}.\end{aligned} \tag{3.19}$$

Hence, the *mean first-passage time* for a random walk that originates at (o,σ) and ends at $[d]$ is

$$\left(H_{[d]}\right)^{o\sigma} = \sum_{t=0}^{\infty} t\left(q_{[d]}\right)^{o\sigma}(h = t) = \left[\left(\delta - T_{[d]}\right)^{-1}\right]^{o\sigma}_{j\beta} u^{j\beta}, \tag{3.20}$$

where

$$\delta^{o\sigma}_{j\beta} = \begin{cases} 1, & \text{if } j = o = \beta = \sigma,\\ 0, & \text{otherwise.}\end{cases} \tag{3.21}$$

Averaging over all possible starting node-layers gives the average mean first-passage time

$$h_{[d]} = \frac{1}{NK} u_{o\sigma}\left(H_{[d]}\right)^{o\sigma} + \frac{1}{N}\pi^{-1}_{[d]}, \tag{3.22}$$

where $\pi_{[d]}$ is the occupation probability of node $[d]$. Reference [11] includes the term $\frac{1}{N}\pi^{-1}_{[d]}$ in Equation 3.22 to account for the mean return time that is not considered when using absorbing random walks, and defines $1/h_{[d]}$ as the random-walk closeness centrality of node $[d]$.

3.5 A direct generalisation from monoplex networks

There are several ways to generalise centrality measures from monoplex networks to multilayer networks. One possible way is to calculate a weighted sum of centrality measures in each layer. Such an approach can be naive in comparison to deriving centrality measures directly in a multilayer framework.

For simplicity, we consider *layer-coupled* multiplex networks; that is,

$$w((i,\alpha),(i,\beta)) = w((j,\alpha),(j,\beta)), \tag{3.23}$$

for all nodes i and j in any layers α and β [24].

Suppose that $\boldsymbol{M}$ is a multiplex network with exactly two layers α and β. Let i be a node of $\boldsymbol{M}$ and suppose that node i has centrality measures $C_{i\alpha}$ and $C_{i\beta}$ in layers α and β respectively. We define the aggregate centrality measure of node i in the multiplex network $\boldsymbol{M}$ to be

$$\frac{1}{2}C_{i\alpha} + \frac{1}{2}C_{i\beta}, \tag{3.24}$$

which is simply the mean of the centrality measures of node i in the two layers.

Now consider a multiplex network $\boldsymbol{M}$ with exactly K layers. Let i be a node of $\boldsymbol{M}$ and suppose that node i has a centrality measure of $C_{i\zeta}$ in layer ζ. We define the aggregate centrality measure of node i in the multiplex network $\boldsymbol{M}$ to be

$$\sum_{\zeta=1}^{K} \frac{w_\zeta}{W} C_{i\zeta} = \frac{1}{W} \sum_{\zeta=1}^{K} \sum_{\substack{\eta=1 \\ \eta\neq\zeta}}^{K} w_{\zeta\eta} C_{i\zeta}. \tag{3.25}$$

That is, we weight the centrality measure of node i in layer ζ by w_ζ/W, where $w_{\zeta\eta}$ is the *coupling strength* (i.e. the weight of the inter-layer edges) between layers ζ and η and $W = \sum_{\zeta=1}^{K} w_\zeta$ is a normalising constant.

4 Ranking economies in international trade

In this chapter, we model the international trade system in the year 2000 as an undirected multiplex network with edge weights and node weights, and rank economies using the two approaches that we discussed in Chapter 3. We also needed to conduct data cleaning for the purpose of the calculations. Additionally, for comparison, we model the international trade system as a monoplex network by aggregating all of the layers of the multiplex network. We then compare the ranking results obtained from the three methods with the ranking result according to gross domestic product (GDP) per capita. The Python code for this chapter is our original work except that we used Multilayer Networks Library developed by M. Kivelä [23] to construct the international trade multiplex network.

4.1 Model

We modelled the international trade system in the year 2000 as a multiplex network with 177 nodes, 10 layers, and categorical couplings. Each node corresponds to an economy that participates in international trade, and each layer is a commodity category, which is according to the first digit (that is, the classification of the highest level) of the 4-digit Standard International Trade Classification (SITC4, revision 2) [14]. In each layer, the edge weight between a pair of nodes is the trade value (in thousands of US dollars) between the two nodes for the corresponding commodity category in 2000. The data of international trade values are from [14], and Table 4.1 gives part of the original data, showing the columns that are relevant to our model. We specify the coupling strength of the multiplex network to be 40000, which is a speculative value that is lower than the mean intra-layer edge weight of 157188. Each node has the same node weight across all layers; this weight is the GDP per capita (in current international dollars[1]) based on purchasing power parity (PPP[2]) of the node in 2000. The data of GDP per capita are from [37], and Table 4.2 gives part of the original data, showing the columns that are relevant to our model.

Figure 4.1 is the visualisation of the international trade multiplex network. For the sake of readability, we do not show inter-layer edges in the plot.

I spent two days cleaning the two sets of data. First, the two sets of data adopted different naming conventions for countries, so we changed the names in [37] to the names given in [14] (see Appendix A of [14]). Second, for some areas, the two sets of data grouped regions differently, so we grouped the economies according to the method of [14] (see Appendices A and B of [14]). Third, [37] does not provide GDP

[1]An international dollar has the same purchasing power over GDP as the U.S. dollar has in the United States [37].

[2]PPP GDP is gross domestic product converted to international dollars using purchasing power parity rates [37].

Importer	Exporter	SITC4	Value (thousands of US dollars)
Algeria	South Africa	8745	124
Algeria	Libya	5621	1631
Algeria	Morocco	5417	1955
Algeria	Morocco	5530	1189
Algeria	Morocco	5911	139

Table 4.1: Part of the original data for international trade values [37], with columns that are relevant to the international trade multiplex network.

Country name	GDP per capita, PPP (current international dollars)
Aruba	
Andorra	
Afghanistan	
Angola	2687.82983
Albania	4248.83626

Table 4.2: Part of the original data for GDP per capita based on PPP [14], with columns that are relevant to the international trade multiplex network. The empty cells are missing values.

per capita for some economies that are documented in [14], so we added the missing values using the most reliable sources that we could find online.

We ranked the economies in the international trade network in 2000 according to twelve centrality measures, which are given in Table 4.3. Let us use two examples to explain our nomenclature: simRWOC is the random walk centrality measure calculated using the method in Section 3.5 and aggRWOC is the random walk centrality measure of the aggregate monoplex network that is obtained by adding all the layers of the multiplex network together.

4.2 Ranking results

Data that are in the top of a ranking tend to give more robust results than the lower-ranked ones, so we give the top ten economies according to each centrality measure in Figure 4.2. We observe that RWOC, simRWOC, aggRWOC, PageRank, simPageRank, aggPageRank, simRWBC, and aggRWBC give similar ranking results. Interestingly, two groups of centrality measures give exactly the same results: the first group is RWOC, aggRWOC and aggRWBC, and the second group is simRWOC and simRWBC. In addition, RWCC and its associated measures give similar results. In particular, the ranking results given by RWCC and aggRWCC are very similar. The ranking result given by RWBC is not similar to any of the other results, but is more different from the ones given by RWCC and its related measures. For comparison,

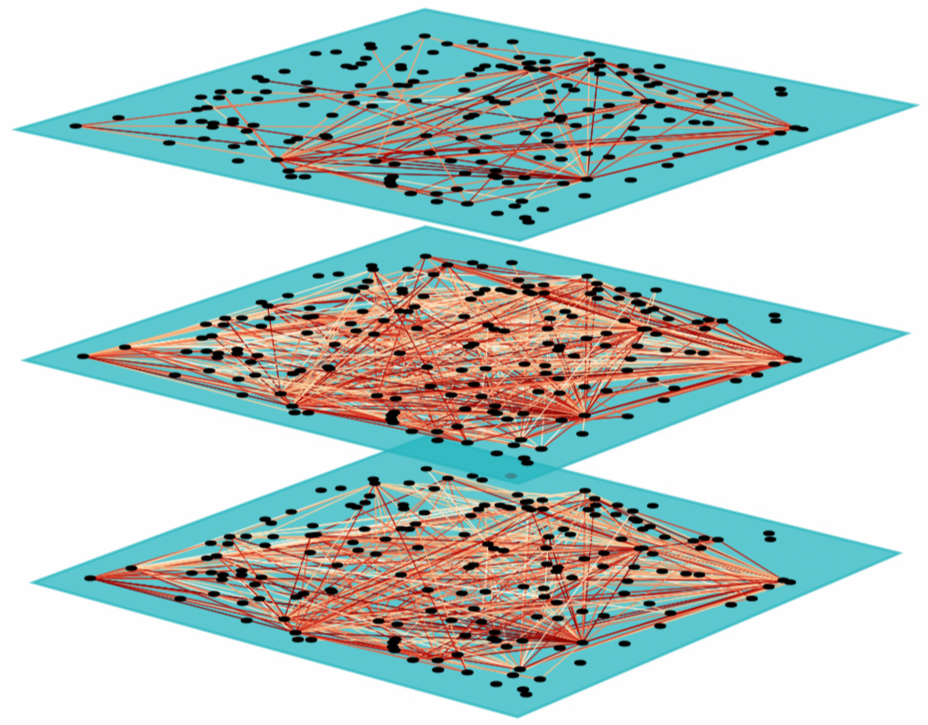


Figure 4.1: Visualisation of the international trade multiplex network (3 layers of the 10 layers) using the Multilayer Networks Library [23]. For the sake of readability, we omit intra-layer edges with weights lower than 100000 as well as inter-layer edges. Each cyan plate is a layer, and black dots represent nodes. Lines represent intra-layer edges, with darker colour indicating larger edge weight. The figure exhibits the heterogeneity of the network in each layer: the top layer has fewer edges with larger weights than the other two layers, and nodes can have very different node strengths in different layers.

Abbreviation	Centrality measure
RWOC	Random-walk occupation centrality
simRWOC	Simple random-walk occupation centrality
aggRWOC	Aggregate random-walk occupation centrality
PageRank	PageRank centrality
simPageRank	Simple PageRank centrality
aggPageRank	Aggregate PageRank centrality
RWBC	Random-walk betweenness centrality
simRWBC	Simple random-walk betweenness centrality
aggRWBC	Aggregate random-walk betweenness centrality
RWCC	Random-walk closeness centrality
simRWCC	Simple random-walk closeness centrality
aggRWCC	Aggregate random-walk closeness centrality

Table 4.3: Centrality measures and their abbreviations.

we also include the top ten economies according to GDP per capita[3] in Figure 4.2. We observe that the ranking result given by GDP per capita are very different from the ones given by our model.

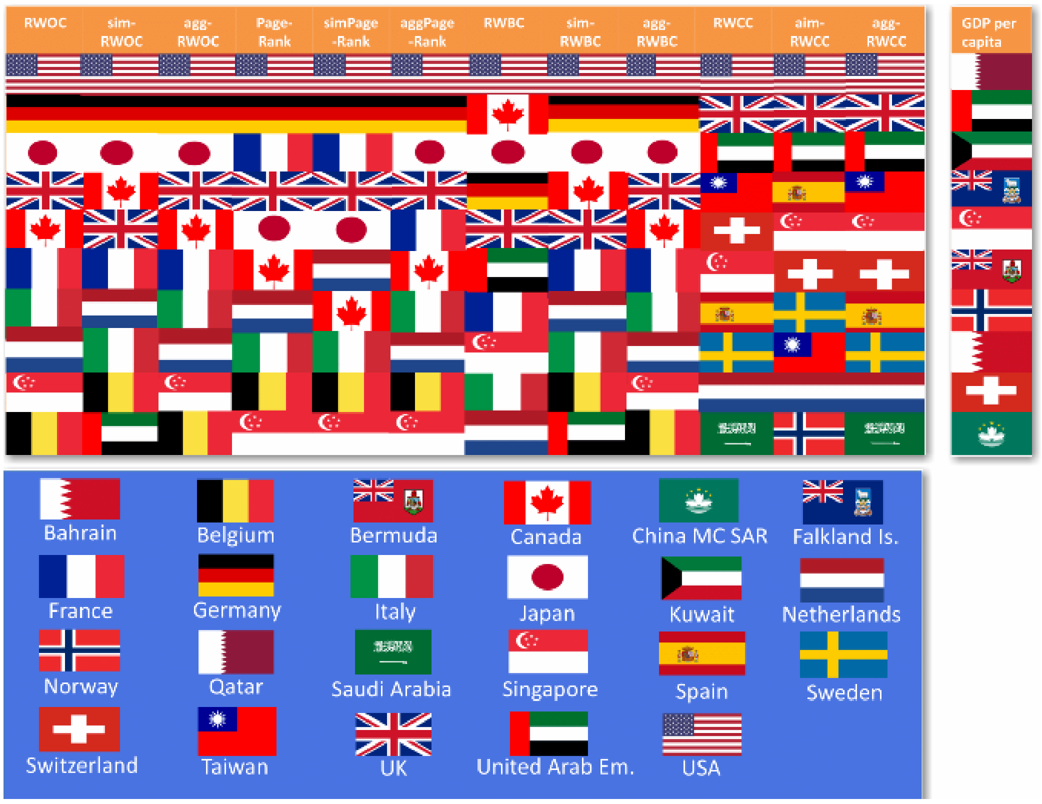

RWOC
sim-RWOC
agg-RWOC
Page-Rank
simPage-Rank
aggPage-Rank
RWBC
sim-RWBC
agg-RWBC
RWCC
aim-RWCC
agg-RWCC
GDP per capita
Bahrain
Belgium
Bermuda
Canada
China MC SAR
Falkland Is.
France
Germany
Italy
Japan
Kuwait
Netherlands
Norway
Qatar
Saudi Arabia
Singapore
Spain
Sweden
Switzerland
Taiwan
UK
United Arab Em.
USA

Figure 4.2: Top ten countries according to the twelve centrality measures and GDP per capita. For each ranking method, the country at the top gives the highest in importance and the rank goes down as we move down the column. Due to limited space, we represent Belgium–Luxembourg Economic Union by Belgium, and Switzerland–Liechtenstein Economic Union by Switzerland. National flags in the legend are arranged in alphabetical order. All the pictures of national flags are from Wikipedia and some are resized slightly.

In order to examine the level of similarity between each pair of centrality measures with respect to all ranking results, we calculated *Kendall's tau-b coefficients* for each pair of the measures (see Figure 4.3).

Kendall's tau coefficient is a measure of ordinal association between two variables, and takes values between -1 and 1 [1]. A coefficient of 1 denotes that the two variables rank data in exactly the same order, and a coefficient of -1 implies that the two variables rank data in exactly the reverse order [1]. The *tau-b* statistic makes adjustments for ties [1].

Figure 4.3 exhibits two blocks of patterns, within which we see more similarity than

[3]The data are originally from [37], but we used the 'cleaned' version as described in Section 4.1, in order that the economies are grouped in the same way as in our model.

compared with centrality measures that are not in the block. The first block is RWOC, simRWOC, aggRWOC, simPageRank, aggPageRank, simRWBC, and aggRWBC. Interestingly, the Kendall tau-b coefficient for RWOC and aggRWOC is 1, which means that the two centrality measures give exactly the same ranking order. This suggests that the two measures are probably identical. There are three pairs of centrality measures that give very similar but not exactly the same ranking orders, which are RWOC–aggRWBC, simRWOC–simRWBC, and aggRWOC–aggRWBC. The second block is RWCC and its related measures, within which RWCC and aggRWCC give very similar but not exactly the same ranking orders.

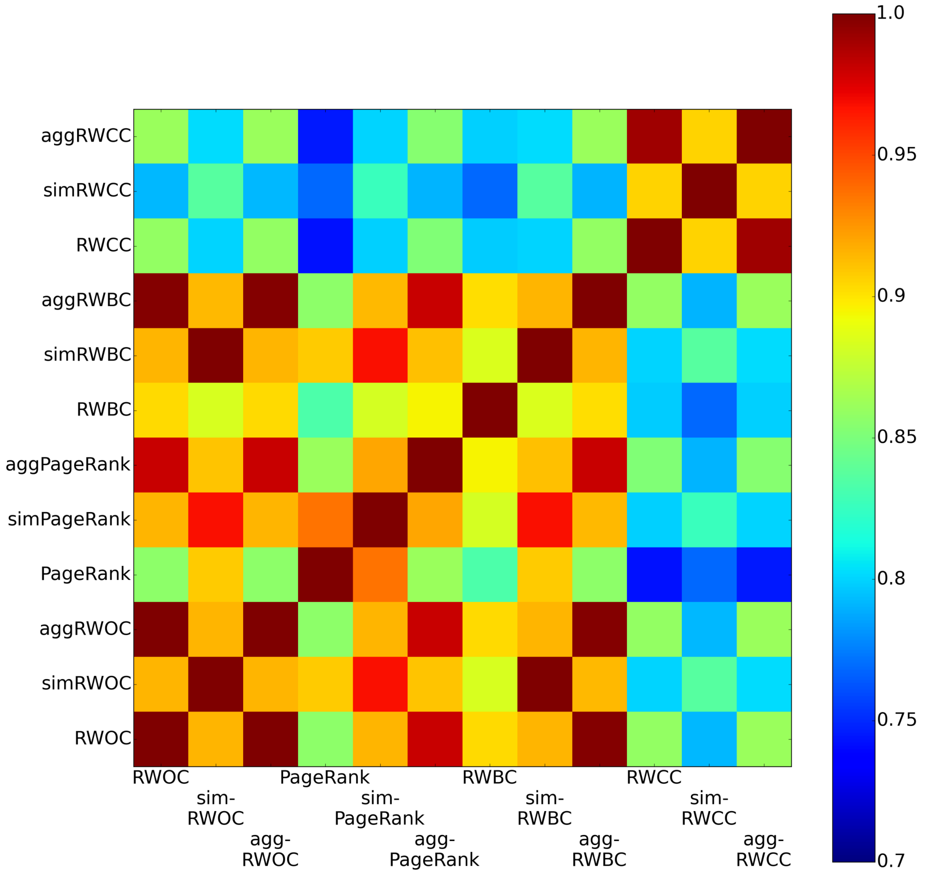


Figure 4.3: Color representation of pairwise Kendall's tau-b coefficients of the twelve centrality measures for the international trade network. The figure is drawn using the MATPLOTLIB library for Python developed by J. D. Hunter [21].

5 Simulations

In this chapter, we perform computer simulations for unweighted and weighted multiplex networks to expand and test the observations in Chapter 4 for the international trade multiplex network. All the Python codes are original except that Subsection 5.2.4 used Multilayer Networks Library [23] to extract the edge weights and the node weights of the international trade network.

We identify several cases where multiple centrality measures are very positively correlated. This observation is important for two main reasons: first, when multiple centrality measures are available to calculate, and give almost the same ranking results, one can calculate the simplest measure for economy of computational cost; second, in this case, if the research data collected has missing values, or some part of the multilayer structure is corrupted, researchers can be more confident to use the aggregated network in calculating centrality measures, as the aggregated network should be closer to the aggregated real-world network and the ranking results should be similar.

5.1 Random-graph ensembles

Random graphs are often used to test results that are obtained from real-world networks. We give the definition of a random graph from [28]:

Definition 7. A *random graph* is a model network in which some specific set of parameters take fixed values, but the network is random in other aspects.

We conduct simulations using Erdős-Rényi $G(N, p)$ random-graph ensembles, which are the most fundamental and widely studied of all random graph models [28]. An Erdős-Rényi $G(N, p)$ random graph is constructed by connecting N nodes randomly, and each of the $\binom{N}{2}$ possible edges is included in the graph with probability p (called *edge probability*) independently of other edges.

Because the international trade multilayer network is a multiplex network with 177 nodes, 10 layers, and categorical couplings of uniform strength, we generate Erdős-Rényi random multiplex networks that have 177 nodes, 10 layers, and categorical couplings of uniform strength for all simulations. The aggregate intra-layer edge probability for the international trade multilayer network is 0.24 accurate to two decimal points, so we let $p = 0.25$ in Subsections 5.2.1–5.2.3. In Subsections 5.2.3 and 5.2.4, we also require that each node have the same node weight across layers, which makes sense in many real-world examples of networks including the international trade system.

For each simulation, we rank the nodes in the randomly-generated multiplex network according to each of the twelve centrality measures.

For PageRank and its associated centrality measures, we use the teleportation rate of 0.15, the one that is often used in the literature [15].

5.2 Analysis of ranking results

We conduct simulations for four different cases: unweighted multiplex networks, multiplex networks with edge weights, multiplex networks with node weights, and multiplex networks with both edge weights and node weights.

5.2.1 Unweighted multiplex networks

First, we conduct 100 simulations using random multiplex networks that have no edge weights or node weights.

Figure 5.1 shows that the distribution of sorted centrality values are very narrow and approximately symmetric for all centrality measures except RWCC and its associated centrality measures. Although none of the plots look exactly the same, the distributions of sorted values for RWOC, PageRank, RWBC, and their associated centrality measures are more similar to each other in shape than those for RWCC and its associated measures. We also observe that each of RWOC and PageRank has more sorted centrality values in the middle than their associated measures. The sorted values of RWCC and its associated centrality measures are much flatter and more spread out than those of other centrality measures. This suggests that, as with monoplex networks [28], closeness centrality does not distinguish node importance as much as the other measures.

For each pair of centrality measures, we plot a scatter plot of the centrality values of one measure against those of the other. The plots segment the centrality measures into three groups, within which we observe more similarity than compared with centrality measures from a different group.

The first group is RWOC, PageRank, and their associated centrality measures. Within the group, we observe almost perfect positive correlation between each pair of centrality measures. The almost-perfect positive correlation between RWOC and PageRank is reasonable, because, if the teleportation rate is small, PageRank can then be understood as RWOC when teleportation is allowed during random walks. Moreover, the almost-perfect positive correlation between RWOC and its associated measures is reasonable because RWOC is equivalent to degree centrality in an unweighted multilayer network (see Section 3.1), and the contribution of inter-layer edges to the degree of each node is the same in a multiplex network with categorical couplings. The scatter plots within this group are very similar and we give an example in Figure 5.2 (Subfigure (a)). In particular, we note that, the centrality values given by RWOC and aggRWOC are not exactly the same, even though they give the same ranking order. This suggest that the two measures are not identical quantities but probably have related formulae.

The second group is simRWBC and aggRWBC, which are very positively correlated with each other. The scatter plot for simRWBC and aggRWBC is similar to

(a) of Figure 5.2.

The third group consists of RWCC and its associated centrality measures. Within the group, simRWCC and aggRWCC are very positively correlated with each other (the plot is similar to (a) of Figure 5.2), and they are also very positively correlated with RWCC except for the largest centrality values (see (c) of Figure 5.2). This observation suggests that the multilayer structure also has a significant effect on RWCC, especially for the most central nodes. Another noteworthy point is that RWCC and its associated centrality measures are not correlated with other centrality measures (see (d) of Figure 5.2 for an example). We calculate Kendall tau-b coefficients for each pair of centrality measures, and the coefficients for RWCC (and its associated measures) with other centrality measures lie between -0.1 and 0.1.

Interestingly, RWBC is not very correlated with any of the other centrality measures (see (b) of Figure 5.2 for an example). This suggests that, for this family of graphs, RWBC depends more on the multilayer structure than RWOC and PageRank.

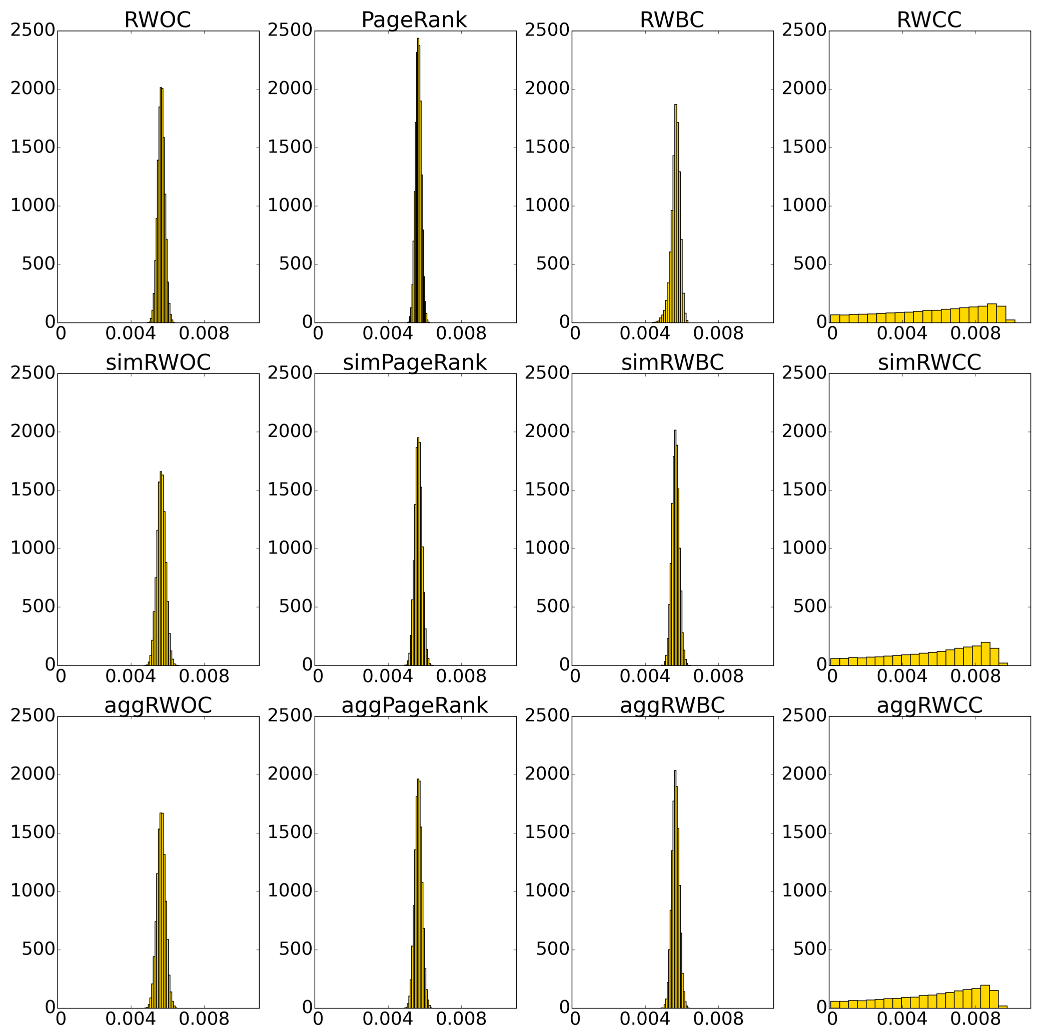


Figure 5.1: Histograms of sorted values for centrality measures based on simulations using random unweighted multiplex networks. The horizontal axes give sorted centrality values and the vertical axes give normalised frequency. The figure is drawn using the MATPLOTLIB library for Python [21].

5.2.2 Multiplex networks with edge weights

Second, we conduct 18 groups of simulations, with 100 simulations for each group, using random multiplex networks that have edge weights. In this subsection, we generate only random multiplex networks with uniform coupling strengths. For each group of simulations, we tried a different inter-layer edge weight: 1, 10, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 2000, 3000, 4000, 5000, 10000, and 19000. We draw real-valued edge weights independently from the uniform distribution on $[1, 38000]$. The plots that correspond to each coupling strength show very similar patterns, so we display the ones that correspond to the coupling strength of 400.

Figure 5.3 is similar to Figure 5.1. However, for each of RWOC, PageRank, RWBC, and RWCC, the distribution of sorted centrality values is the same as that

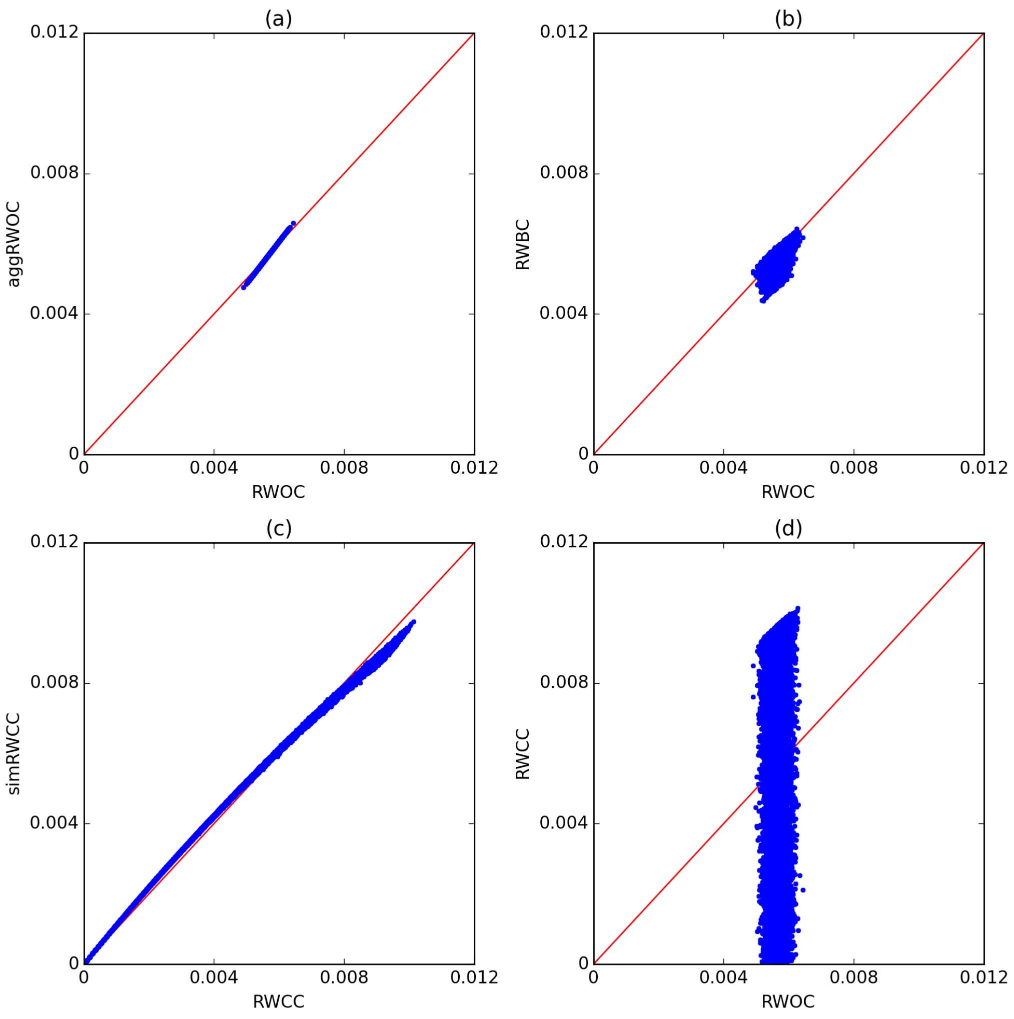


Figure 5.2: Scatter plots of centrality measures based on simulations using random unweighted multiplex networks. We draw red lines through the origin with slope 1 for comparison. The figure is drawn using the MATPLOTLIB library for Python [21].

of their associated measures.

As in Subsection 5.2.1, we plot a scatter plot of the centrality values for each pair of centrality measures. The plots segment the centrality measures into two groups. One consists of RWCC and its associated centrality measures, and the other contains all of the remaining measures. Within each group, the centrality measures are very positively correlated with each other (see (a) and (d) of Figure 5.2 for examples of scatter plots within a group and across groups respectively). In particular, RWCC is very positively correlated with simRWCC and aggRWCC even for large sorted values.

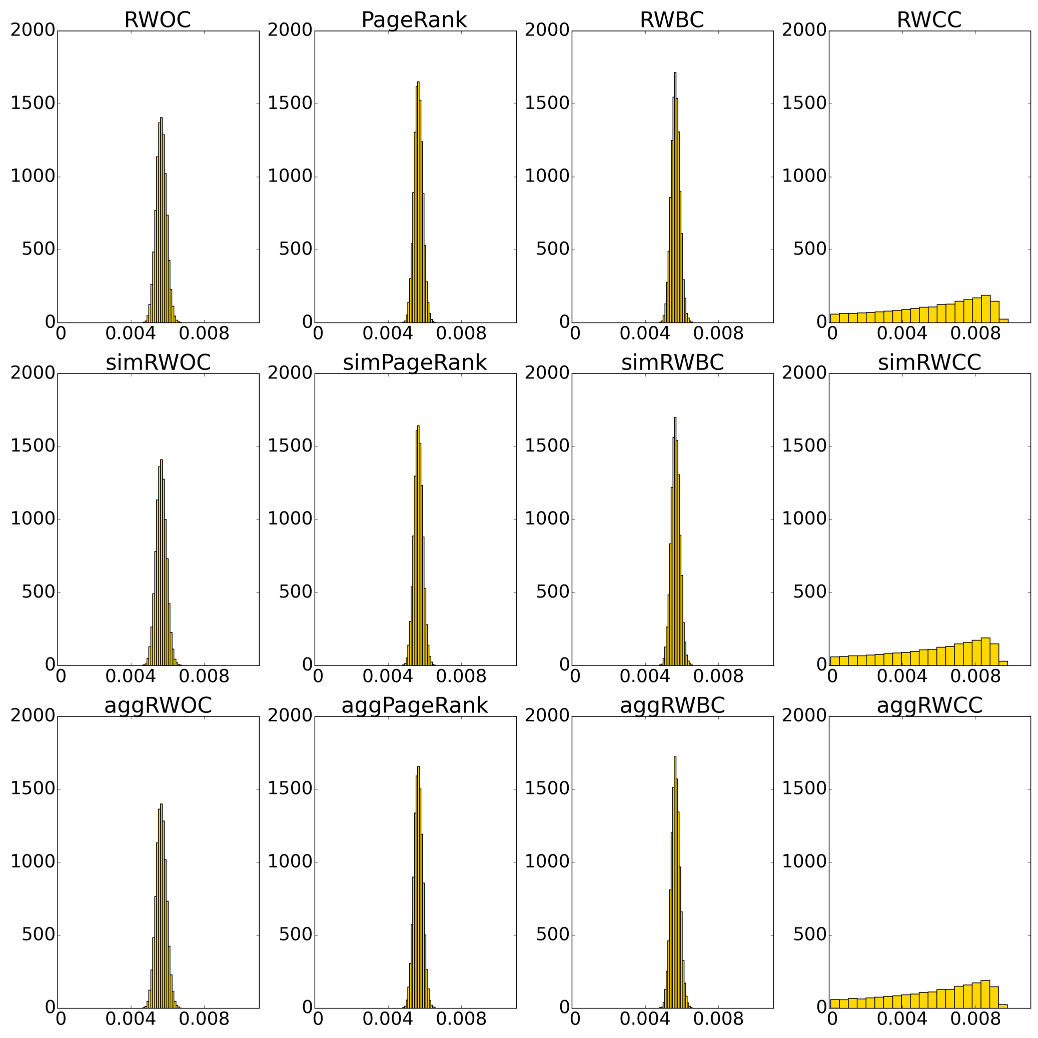


Figure 5.3: Histograms of sorted values for centrality measures based on simulations using random multiplex networks with edge weights. The horizontal axes give sorted centrality values, and the vertical axes give normalised frequency. The figure is drawn using the MATPLOTLIB library for Python [21].

5.2.3 Multiplex networks with node weights

Third, we conduct 100 simulations using random multiplex networks with node weights. We draw real-valued node weights independently from the uniform distribution on $[100, 500]$.

In this case, the scatter plots look very similar to the ones in Subsection 5.2.2. However, compared with the previous two cases in Subsections 5.2.1 and 5.2.2, the distribution of centrality values in this case (see Figure 5.4) is much flatter and wider for RWOC, PageRank, RWBC, and their associated centrality measures, and less uniform for RWCC and its associated measures. This confirms our hypothesis in Chapter 1 that node weights are a nontrivial property for multilayer networks. Additionally, as in Subsection 5.2.2, for each of RWOC, PageRank, RWBC, and RWCC, the distribution of sorted centrality values is the same as that of their associated measures.

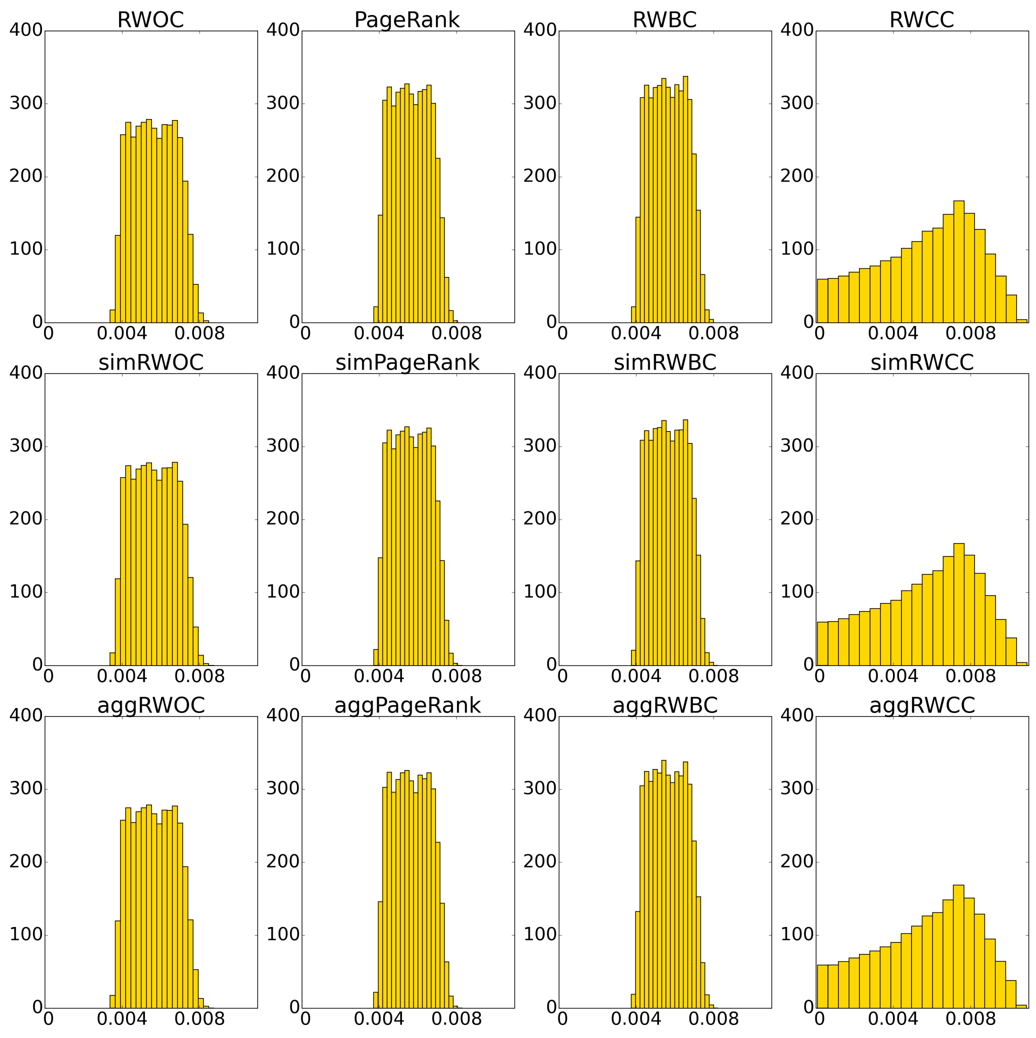


Figure 5.4: Histograms of sorted values for centrality measures for simulations using random multiplex networks with node weights. The horizontal axes give sorted centrality values and the vertical axes give normalised frequency. The figure is drawn using the MATPLOTLIB library for Python [21].

5.2.4 Multiplex networks with both edge weights and node weights

Finally, we conduct 100 simulations using random multiplex networks with both edge weights and node weights. We draw node weights and edge weights from those of the international trade network (see Figure 5.5) without replacement. Same as in Chapter 4, we specify the coupling strength of each random multiplex network to be 40000.

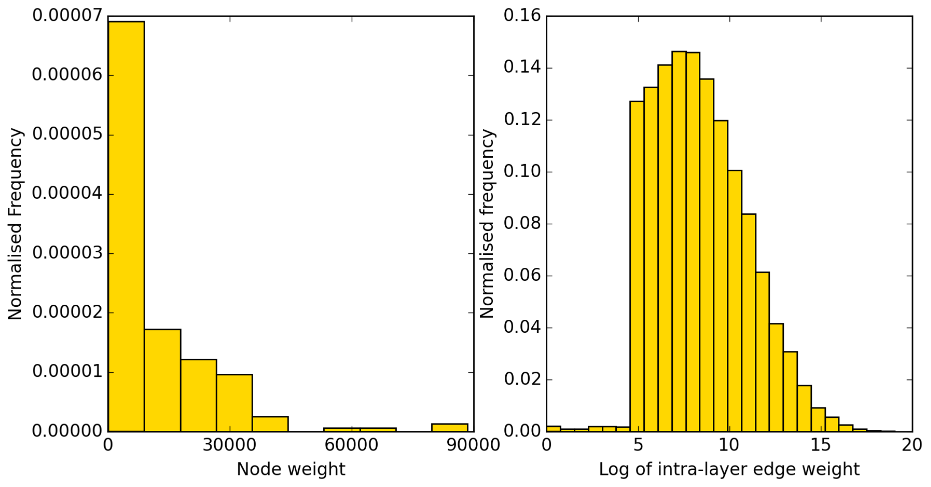


Figure 5.5: Histograms of node weights (left) and log values of intra-layer edge weights (right) of the international trade multiplex network. The figure is drawn using the MATPLOTLIB library for Python [21].

For each centrality measure, the distribution of sorted centrality values is left-skewed (see Figure 5.6), which is probably because of the distributions of the edge weights and the node weights. We observe that two pairs of centrality measures (RWOC–aggRWOC and PageRank–simPageRank) have the same distributions. Compared with Subsections 5.2.1 to 5.2.3, Figure 5.7 shows a less obvious positive correlation for each pair of centrality measures except for two pairs (RWOC–aggRWOC and PageRank–simPageRank), which have perfect positive correlation.

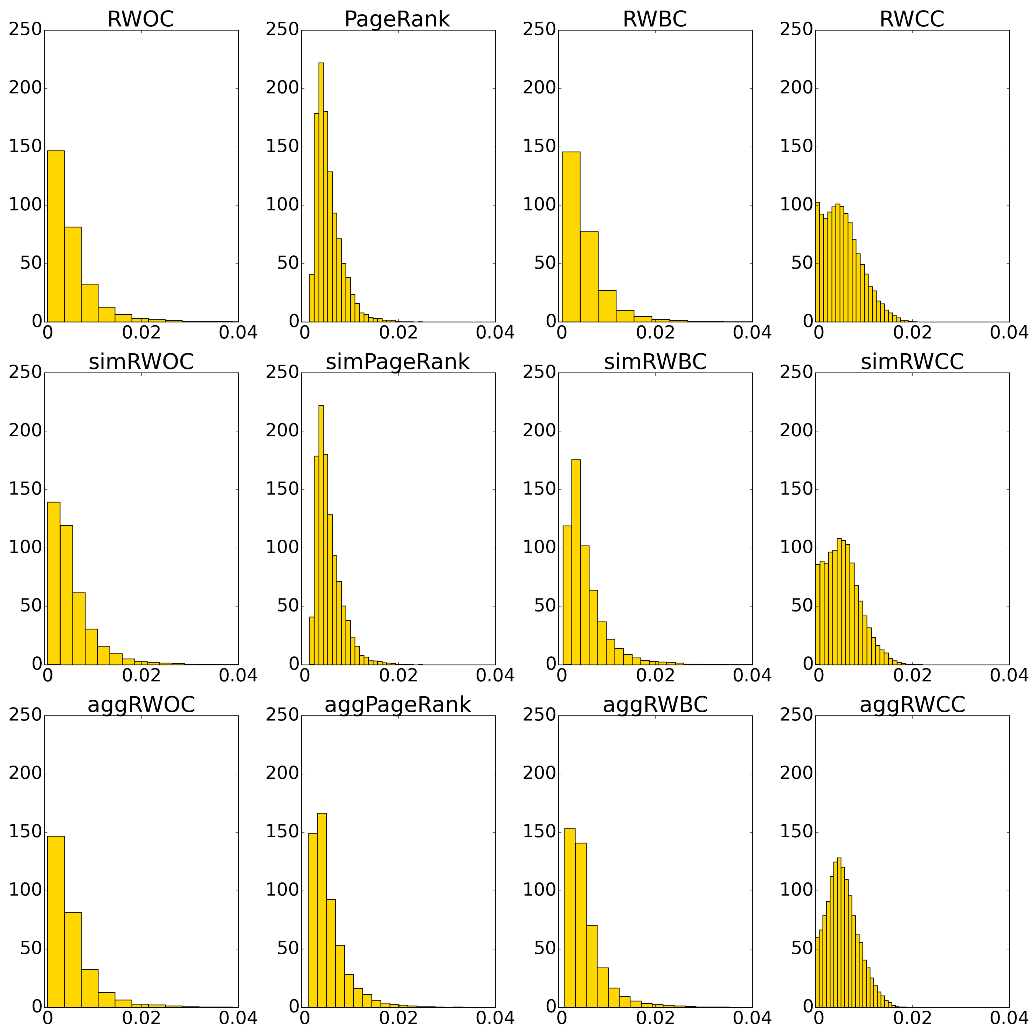


Figure 5.6: Histograms of sorted centrality values for simulations using random multiplex networks with both edge weights and node weights. The horizontal axes give sorted values, and the vertical axes give normalised frequency. The figure is drawn using the MATPLOTLIB library for Python [21].

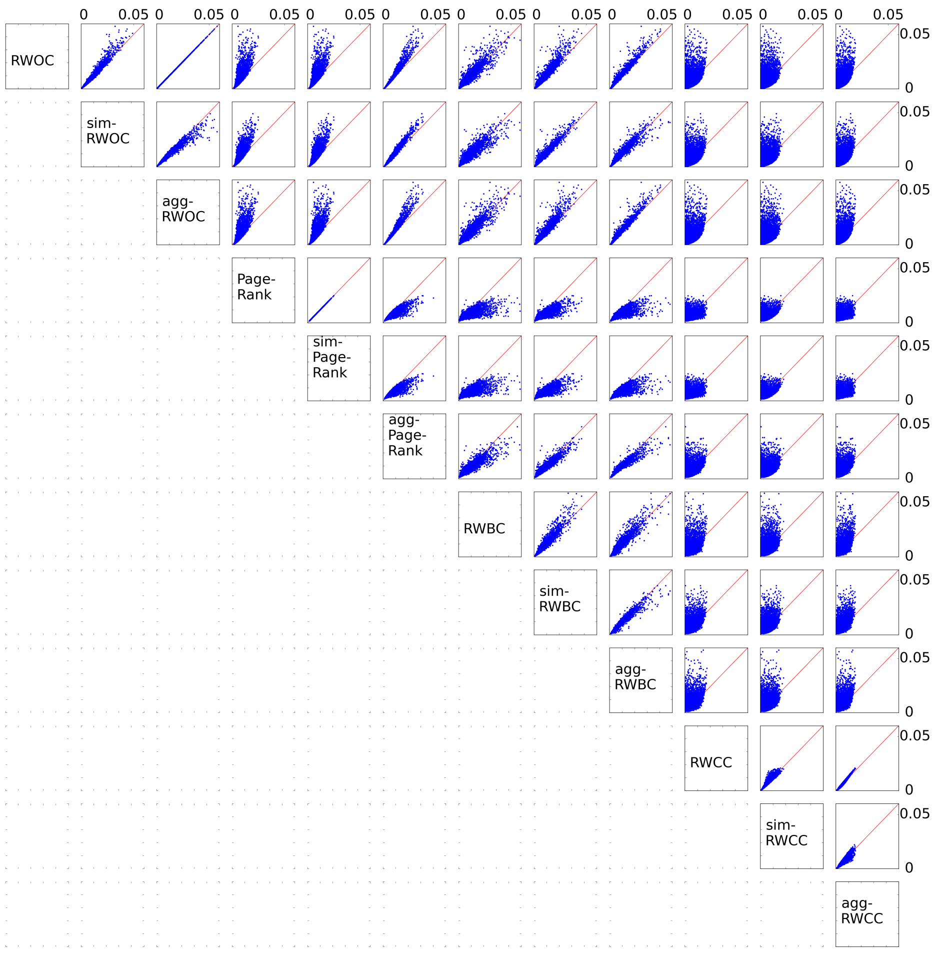


Figure 5.7: Pairwise scatter plots of centrality measures for simulations using random multiplex networks with both edge weights and node weights. The horizontal and vertical axes give sorted centrality values. We draw red lines through the origin with slope 1 for comparison. The figure is drawn using the MATPLOTLIB library for Python [21].

6 Conclusions

In this report, we have proposed a method that allows tools for centrality measures in undirected multilayer networks with edge weights to be directly applicable to undirected multilayer networks with both nonnegative edge weights and nonegative node weights. We also conducted a theoretical and empirical investigation of centrality measures that are based on random walks in multilayer networks and weighted monoplex networks. We studied two related methods to calculate random-walk occupation centrality (RWOC), PageRank centrality (PageRank), random-walk betweenness centrality (RWBC), and random-walk closeness centrality (RWCC) in undirected multilayer networks with edge weights. We ranked economies in the international trade system and conducted simulations with undirected random graphs to compare the results by three ranking methods in four cases: unweighted multiplex networks, multiplex networks with edge weights, multiplex networks with node weights, and multiplex networks with both edge weights and node weights.

For unweighted multiplex networks, simulations segmented the centrality measures into three groups, within which we observed significant positive correlation. The first group is RWOC, PageRank, and their associated centrality measures; the second group is simRWBC and aggRWBC; and the third group is RWCC and its associated centrality measures. Simulations suggested less strong positive correlation between the first two groups and no significant correlation between the third group and the others.

For multiplex networks with edge weights, we conducted 18 groups of simulations, using a different coupling strength ranging from 1 to 19000 for each group. Simulations suggested that coupling strength does not have a significant effect on the distribution of centrality values or the pairwise correlations between the centrality measures. Simulations showed almost perfect positive correlation between RWOC, PageRank, RWBC, and their associated centrality measures this time. Again, RWCC and its associated centrality measures showed no significant correlation with other centrality measures.

Simulations on multiplex networks with node weights exhibited behaviour very different from the previous two cases. In particular, the differences between nodes are more distinguishable by RWOC, PageRank, RWBC, and their associated centrality measures, and the sorted centrality values are less concentrated in the centres of the ditributions.

Simulations on multiplex networks with both edge weights and node weights exhibited much lower correlations between each pair of centrality measures except for two pairs (RWOC–aggRWOC and PageRank–simPageRank). Each of these two pairs shows perfect positive correlation, which was also observed in the previous three cases. However, for simulations with unweighted random graphs, none of the centrality measures have the same distribution of sorted centrality values. This suggests that each pair of RWOC–aggRWOC and PageRank–simPageRank are not identical quantities but probably have related formulae.

Overall, the results of the four cases confirmed our hypothesis that node weights are a property that is not only natural but also nontrivial to add to multilayer networks. Additionally, all simulations suggested that RWCC and its associated centrality measures do not distinguish node importance as much as the other measures; they are not significantly correlated with the other measures either. These observations are also true for monoplex networks [28]. Moreover, RWOC, PageRank, and their associated measures tend to give ranking results that are similar to each other, if not the same, but the correlation is weakened by the presence of both edge weights and node weights except for two pairs (RWOC–aggRWOC and PageRank–simPageRank). This suggests that each pair are probably related analytically. The simulations also suggested that RWBC and RWCC depend more on the multilayer structure than RWOC and PageRank.

There are several possible extensions to the work in this report. First, our simulations strongly suggested that each pair of RWOC–aggRWOC and PageRank–simPageRank is related, and it would be interesting to prove this analytically. Moreover, our simulations used random multiplex networks with 177 nodes and 10 layers to make results comparable to those of the international trade multiplex network. Future work can use more general random graphs such as directed graphs and more general multilayer structures. Additionally, we assumed that node weights and edge weights are independent from each other in the last case of simulations. However, future work can assume that the two variables are correlated at some level, which is reasonable in some real-world models.

Appendix A. Eigenvalue problem with tensors

This appendix is based on [11]. To solve Equation 3.4, we first consider the eigenvalue problem for a rank-2 tensor:

$$W^i_j v_i = \lambda v_j, \tag{A.1}$$

which is equivalent to calculating the eigenvectors of a matrix. We can 'unfold' any tensor to a lower rank tensor, and there are multiple ways to unfold $\boldsymbol{T}$ to lower-rank tensors. For our purpose, it is most convenient to flatten $\boldsymbol{T}$ to a rank-2 tensor $\boldsymbol{T_f}$ with $NK \times NK$ scalar components. Note that $\boldsymbol{T_f}$ is simply a $NK \times NK$ real matrix. Finding the eigentensors of $\boldsymbol{T}$ is equivalent to finding the eigenvectors of $\boldsymbol{T_f}$.

Appendix B. The flattened transition tensor for random-walk occupation centrality

Each scalar component $T^{i\alpha}_{j\beta}$ of $\boldsymbol{T}$ is nonnegative, because both $M^{k\gamma}_{j\beta}$ and $\tilde{D}^{i\alpha}_{k\gamma}$ are nonnegative. Moreover,

$$u_{i\alpha} T^{i\alpha}_{j\beta} = u_{i\alpha} M^{i\alpha}_{k\gamma} \tilde{D}^{k\gamma}_{j\beta} = s_{k\gamma} \tilde{D}^{k\gamma}_{j\beta}. \tag{B.1}$$

By the definition of $\tilde{D}^{k\gamma}_{j\beta}$,

$$s_{k\gamma} \tilde{D}^{k\gamma}_{j\beta} = \begin{cases} 1, & \text{if } (i,\alpha) \text{ is not isolated,} \\ 0, & \text{if } (i,\alpha) \text{ is isolated.} \end{cases} \tag{B.2}$$

After flattening $\boldsymbol{T}$ to $\boldsymbol{T_f}$, each column of $\boldsymbol{T_f}$ sums to 1 if the corresponding node-layer is not isolated, and the columns that correspond to isolated node-layers are 0. Thus, $\boldsymbol{T_f}$ is almost a left-stochastic matrix.

This implies that, $(\boldsymbol{T_f})^i_j$, the element in the jth row and ith column of $\boldsymbol{T_f}$, gives the probability for a random walker to jump from state i to state j.

Appendix C. Python code of simulations

This Python script conducts 100 simulations for the case of Subsection 5.2.2 when the coupling strength is 400. Conducting one such simulation takes a few minutes on a normal computer.

```python
import numpy as np
import csv
from random import *
from numpy import *
from numpy import linalg as la
from itertools import izip

K=177 # number of nodes
L=10 # number of layers
N=K*L # number of (node,layer) pairs

p_values=L*[0.25] # edge probability vector
edgemin=1 # minimum intralayer edge weight
edgemax=38000 # maximum intralayer edge weight
coupling_strength=400 # strength of layer coupling

simulation_number=100
for number in range(simulation_number):
    print "----------------Simulation", number, "begins----------------"
    SAM=np.zeros(shape=(N,N))
    for l in range(L):
        intraAM=np.random.binomial(1,p_values[l],K**2) # Generate edges (repeat a Bernoulli process with success rate p[l] K**2 times)
        intraAM=intraAM.reshape((K,K))
        intraAM[range(K),range(K)]=0 # make diagonal zero (remove self-loops)
        intraAM=np.triu(intraAM) # make sub-diagonal entries all zero
        for i in range(K):
            for j in range(K):
                if intraAM[i,j]==1:
                    intraAM[i,j]=uniform(edgemin,edgemax) ## Add random edge weights to intralayer edges
        for i in range(K):
            for j in range(K-i-1):
                intraAM[i+j+1][i]=intraAM[i][i+j+1] # reflect intraAM in its diagonal
        for i in range(K):
            for j in range(K):
                SAM[i+K*l][j+K*l]=intraAM[i][j] # incorporate intraAM into the corresponding diagonal block of SAM
    SAM=np.asmatrix(SAM)

    ## Add couplings
    for node in range(K):
        for layer1 in range(L):
            for layer2 in range(L):
                if layer2 != layer1:
                    SAM[node+K*layer1,node+K*layer2]=coupling_strength

    ## Transition matrix for the multiplex network
    S=np.sum(SAM,axis=1) # Note that the multiplex network is undirected so the supra-adjacency matrix is symmetric
    d=np.zeros(N)
    for i in range(N):
        d[i]=1/S[i] # Note that S[i] is non-zero because of couplings
    D=np.diag([d[i] for i in range(N)])
    T=np.dot(SAM,D) # Note that T is a left stochastic matrix, i.e. all entries non-negative and each column sums to 1

    ## Get a monoplex network by aggregating the multiplex network across layers
    # i.e. add all the intralayer matrices, which are weighted by layer couplings
    aggAM=np.zeros((K,K))
    for l in range(L):
        for i in range(K):
            for j in range(K):
                aggAM[i,j]=aggAM[i,j]+SAM[i+l*K,j+l*K]

    ## Calculate the transition matrix for the aggregated network
    aggS=np.sum(aggAM,axis=1)
    agg_isolated=set(np.where(aggS==0)[0]) # the index set of isolated nodes in the aggregated network
    aggS[aggS==0]=1 # This is to avoid division by zero later. Columns of aggT that correspond to isolated points are still all zero
    aggd=np.zeros(K)
    for i in range(K):
        aggd[i]=1/aggS[i]
    aggD=np.diag([aggd[i] for i in range(K)])
    aggT=np.dot(aggAM,aggD) # Similar to T, aggT is a left stochastic matrix


    print "A random multiplex network has been generated."



    ## RWOC
    weight=sum(SAM)
    rankRWOC=np.zeros(K)
    for i in range(K):
        k=0
        for l in range(L):
            k=k+sum(SAM[i+l*K,:])
        rankRWOC[i]=k/weight
    print "RWOC done."


    ## Simple RWOC
    simrankRWOC=np.zeros(K)
    for l in range(L):
        intralayer=np.zeros((K,K))
        for i in range(K):
            for j in range(K):
                intralayer[i][j]=SAM[i+l*K,j+l*K] # intralayer is the adjacency matrix for layer l
        weight=sum(intralayer)
        for h in range(K):
            k=sum(intralayer[h,:])
            simrankRWOC[h]=simrankRWOC[h]+k/weight
    simrankRWOC=simrankRWOC/L
    print "simRWOC done."


    ## Aggregate RWOC
    weight=sum(aggAM)
    aggrankRWOC=np.zeros(K)
    for i in range(K):
        aggrankRWOC[i]=sum(aggAM[i,:])/weight
    print "aggRWOC done"


    ## PageRank
    T=np.dot(SAM,D) # Refresh matrix T
    r=0.85 # (1-r) is the teleportation rate
    R=r*T+(1-r)/N*np.ones((N,N))
    RLeigenvector=np.asmatrix(np.ones(N))
    RLeigenvector=RLeigenvector.transpose() # Now RLeigenvector is a N*1 column vector
    while la.norm(RLeigenvector-np.dot(R,RLeigenvector))>10**-8:
        RLeigenvector=np.dot(R,RLeigenvector)/la.norm(np.dot(R,RLeigenvector))
    NRLeigenvector=RLeigenvector/np.sum(RLeigenvector)
    NRLeigenvector=NRLeigenvector.reshape((L,K))
    rankPageRank=np.squeeze(np.asarray(np.sum(NRLeigenvector,axis=0)))
    print "PageRank done."


    ## Simple PageRank
    r=0.85 # (1-r) is the teleportation rate
    simrankPageRank=np.zeros(K)
    for l in range(L):
        intralayer=np.zeros((K,K))
        for i in range(K):
            for j in range(K):
                intralayer[i][j]=SAM[i+l*K,j+l*K] # intralayer is the adjacency matrix for layer l
        S=np.sum(intralayer,axis=1)
        isolated=set(np.where(S==0)[0]) # the index set of isolated nodes in layer l
        S[S==0]=1
        d=np.divide(np.ones(K),S)
        Diag=np.diag([d[k] for k in range(K)]) # the normalising matrix for intralayer matrix
        intramatrix=np.dot(intralayer,Diag)
        R=r*intramatrix+(1-r)/K*np.ones((K,K))
        for k in isolated:
            R[:,k]=np.ones(K)/K
        RLeigenvector=np.asmatrix(np.ones(K)) # Calculate the leading eigenvector of R using power iteration
        RLeigenvector=RLeigenvector.transpose()
        while la.norm(RLeigenvector-np.dot(R,RLeigenvector))>10**-8:
            RLeigenvector=np.dot(R,RLeigenvector)/la.norm(np.dot(R,RLeigenvector))
            NRLeigenvector=RLeigenvector/np.sum(RLeigenvector)
        for h in range(K):
            simrankPageRank[h]=simrankPageRank[h]+NRLeigenvector[h]
    for h in range(K):
        simrankPageRank[h]=simrankPageRank[h]/L
    print "simPageRank done."


    ## Aggregate PageRank
    aggT=np.dot(aggAM,aggD) # Refresh matrix aggT
    # Calculate the transition tensor R
    r=0.85 # (1-r) is the teleportation rate
    R=r*aggT+(1-r)/K*np.ones((K,K))
    for k in agg_isolated:
        R[:,k]=np.ones(K)/K
    RLeigenvector=np.asmatrix(np.ones(K))
    RLeigenvector=RLeigenvector.transpose() # Now RLeigenvector is a K*1 column vector
    while la.norm(RLeigenvector-np.dot(R,RLeigenvector))>10**-8:
        RLeigenvector=np.dot(R,RLeigenvector)/la.norm(np.dot(R,RLeigenvector))
    NRLeigenvector=RLeigenvector/np.sum(RLeigenvector)
    aggrankPageRank=np.squeeze(np.asarray(NRLeigenvector))
    print "aggPageRank done."


    ## RWBC
    T=np.dot(SAM,D) # Refresh matrix T
    RWBCsum=np.zeros(K)
    for i in range(K): # absorbing node
        U=T
        for j in range(L): # absorbing node in all layers
            U[:,i+j*K]=0
        V=la.inv((np.identity(N)-U))
        Vsumcol=np.zeros((N,K))
        for m in range(K):
            for n in range(L):
                Vsumcol[:,m]=Vsumcol[:,m]+np.squeeze(np.asarray(V[:,m+n*K]))
        Vsumrow=np.zeros((K,K))
        for m in range(K):
            for n in range(L):
                Vsumrow[m,:] = Vsumrow[m,:] + Vsumcol[m+n*K,:]
        Vsum=Vsumrow/L
        RWBCsum=RWBCsum+np.sum(Vsum, axis=1)
    RWBC=RWBCsum/(K*(K-1))
    rankRWBC=RWBC/sum(RWBC) # Normalise the rank sizes so that they sum up to 1
    print "RWBC done."


    ## Simple RWBC
    T=np.dot(SAM,D) # Refresh matrix T
    simrankRWBC=np.zeros(K)
    for l in range(L):
        intralayer=np.zeros((K,K))
        for i in range(K):
            for j in range(K):
                intralayer[i][j]=SAM[i+l*K, j+l*K] # intralayer is the adjacency matrix for layer l
        S=np.sum(intralayer, axis=1)
        isolated=set(np.where(S==0)[0]) # the index set of isolated nodes in layer l
        S[S==0]=1
        d=np.divide(np.ones(K),S)
        Diag=np.diag([d[k] for k in range(K)]) # the normalising matrix for intralayer matrix
        intramatrix=np.dot(intralayer, Diag)
        simRWBCsum=np.zeros((K,1))
        for i in range(K): # absorbing node
            U=intramatrix
            U[:,i]=0
            U=np.asmatrix(U)
            V=la.inv((np.identity(K)-U))
            simRWBCsum=simRWBCsum+np.sum(V, axis=1)
        simRWBC=simRWBCsum/(K*(K-1))
        simRWBC=np.squeeze(np.asarray(simRWBC))
        simRWBC=simRWBC/sum(simRWBC) # Normalise the rank sizes so that they sum up to 1
        for h in range(K):
            simrankRWBC[h]=simrankRWBC[h]+simRWBC[h]
    for h in range(K):
        simrankRWBC[h]=simrankRWBC[h]/L
    print "simRWBC done."


    # Aggregate RWBC
    aggT=np.dot(aggAM, aggD) # Refresh matrix aggT
    aggRWBCsum=np.zeros(K)
    for i in range(K): # absorbing node
        U=aggT
        U[:,i]=0
        V=la.inv((np.identity(K)-U))
        aggRWBCsum=aggRWBCsum+np.sum(V, axis=1)
    aggRWBC=aggRWBCsum/(K*(K-1))
    aggrankRWBC=aggRWBC/sum(aggRWBC) # Normalise the rank sizes so that they sum up to 1
    print "aggRWBC done."


    ## RWCC
    T=np.dot(SAM,D) # Refresh matrix T
    s=np.zeros(K)
    for i in range(K):
        for l in range(L):
            s[i]=s[i]+np.sum(SAM[:,i+l*K]) # s[i] is the sum of node strengths across layers
    node_strengths_sum=np.sum(s)
    MFPT=np.zeros(K)
    for i in range(K): # absorbing node
        U=T
        for l in range(L): # absorbing node in all layers
            U[:,i+l*K]=0
        MFPT[i]=np.sum(la.inv(np.identity(N)-U))/N+node_strengths_sum/(s[i]*K) # Note that s[i]>0 because of couplings and MFPT[i]> is well defined
    RWCC=np.divide(np.ones(K),MFPT)
    rankRWCC=RWCC/sum(RWCC) # Normalise the rank sizes so that they sum up to 1
    print "RWCC done."


    ## Simple RWCC
    T=np.dot(SAM,D) # Refresh matrix T
    simrankRWCC=np.zeros(K)
    for l in range(L):
        intralayer=np.zeros((K,K))
        for i in range(K):
            for j in range(K):
                intralayer[i][j]=SAM[i+l*K,j+l*K] # intralayer is the adjacency matrix for layer l
        S=np.sum(intralayer,axis=1)
        isolated=set(np.where(S==0)[0]) # the index set of isolated nodes in layer l
        S[S==0]=1
        d=np.divide(np.ones(K),S)
        Diag=np.diag([d[k] for k in range(K)]) # the normalising matrix for intralayer matrix
        intramatrix=np.dot(intralayer,Diag)
        s=np.sum(intralayer,axis=1) # s[i] is the node strength of node i in layer l (excluding couplings)
        node_strengths_sum=np.sum(s)
        MFPT=np.zeros(K)
        simRWCC=np.zeros(K)
        for i in range(K): # absorbing node
            if s[i] !=0:
                U=intramatrix
                U[:,i]=0
```

```python
                MFPT[i]=np.sum(la.inv(np.identity(K)-U))/K+node_strengths_sum/(s[i]*K) # MFPT[i]>0 is well defined for i such that s[i]>0
                simRWCC[i]=1/MFPT[i] # sim[RWCC]=0 for i such that s[i]=0 (i.e. if i is isolated in layer l)
        simRWCC=simRWCC/sum(simRWCC) # Normalise the rank sizes so that they sum up to 1
        for h in range(K):
            simrankRWCC[h]=simrankRWCC[h]+simRWCC[h]
    for h in range(K):
        simrankRWCC[h]=simrankRWCC[h]/L
    print "simRWCC done."


    # Aggregate RWCC
    aggT=np.dot(aggAM,aggD) # Refresh matrix aggT
    s=np.sum(aggAM,axis=1) # s[i] is the node strength of node i
    node_strengths_sum=np.sum(s)
    MFPT=np.zeros(K)
    aggRWCC=np.zeros(K)
    for i in range(K): # absorbing node
        if s[i] !=0: # if node i is not isolated then we calculate as the following, otherwise aggRWCC[i]=0
            U=aggT
            U[:,i]=0
            MFPT[i]=np.sum(la.inv(np.identity(K)-U))/K+node_strengths_sum/(s[i]*K)
            aggRWCC[i]=1/MFPT[i]
    aggrankRWCC=aggRWCC/sum(aggRWCC) # Normalise the rank sizes so that they sum up to 1
    print "aggRWCC done."


    ## Export rank sizes to one csv file
    if number==0:
        csvfile="Output.csv"
        with open(csvfile,"w") as output:
            writer=csv.writer(output,lineterminator="\n")
            writer.writerows(izip(rankRWOC,simrankRWOC,aggrankRWOC,rankPageRank,simrankPageRank,aggrankPageRank,rankRWBC,simrankRWBC,aggrankRWBC,rankRWCC,simrankRWCC,aggrankRWCC))
        output.close()
    else:
        f=open("Output.csv")
        rank=[row for row in csv.reader(f)]
        f.close()
        for node in range(K):
            rank[node].extend([rankRWOC[node],simrankRWOC[node],aggrankRWOC[node],rankPageRank[node],simrankPageRank[node],aggrankPageRank[node],rankRWBC[node],simrankRWBC[node],aggrankRWBC[node],rankRWCC[node],simrankRWCC[node],aggrankRWCC[node]])
        f=open("Output.csv","w")
        csv.writer(f).writerows(rank)
        f.close()
```

Bibliography

[1] A. Agresti, *Analysis of ordinal categorical data*, 2nd edn (Wiley, Hoboken, 2010).

[2] R. Andersen, F. Chung and K. Lang, 'Local graph partitioning using PageRank vectors', *FOCS 2006*, Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (IEEE, Washington, 2006) 475-486.

[3] A.-L. Barabási, N. Gulbahce and J. Loscalzo, 'Network medicine: A network-based approach to human disease', *Nat. Rev. Genet.* 12 (2011) 56–68.

[4] S. Boccaletti, G. Bianconi, R. Criado, C. I. del Genio, J. Gómez-Gardeñes, M. Romance, I. Sendiña-Nadal, Z. Wang and M. Zanin, 'The structure and dynamics of multilayer networks', *Phys. Reps.* 544 (2014) no. 1, 1–122.

[5] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez and D.-U. Hwang, 'Complex networks: Structure and dynamics', *Phys. Reps.* 424 (2006) no. 4–5, 175–308.

[6] S. P. Borgatti and M. G. Everett, 'A graph-theoretic perspective on centrality', *Soc. Networks* 28 (2006) no. 4, 466–484.

[7] B. S. Cohn and M. Marriott, 'Networks and centres in the integration of Indian civilization', *J. Soc. Res.* I (1958) 1–9.

[8] E. Cozzo, R. A. Baños, S. Meloni and Y. Moreno, 'Contact-based social contagion in multiplex networks', *Phys. Rev. E*, 88 (2013) 050801.

[9] M. De Domenico, A. Sóle-Ribalta, E. Cozzo, M. Kivelä, Y. Moreno, M. A. Porter, S. Gómez and A. Arenas, 'Mathematical formulation of multilayer networks', *Phys. Rev. X* 3 (2013) 041022.

[10] M. De Domenico, A. Sóle-Ribalta, S. Gómez and A. Arenas, 'Navigability of interconnected networks under random failures', *PNAS* 111 (2014) no. 23, 8351–8356.

[11] M. De Domenico, A. Solé-Ribalta, E. Omodei, S. Gómez and A. Arenas, 'Centrality in interconnected multilayer networks', *CoRR* abs/1311.2906 (2013).

[12] S. N. Dorogovtsev and J. F. F. Mendes, *Evolution of networks: From biological nets to the Internet and WWW* (Oxford University Press, Oxford, 2003).

[13] E. Estrada and J. Gómez-Gardeñes, 'Communicability reveals a transition to coordinated behavior in multiplex networks', *Phys. Rev. E*, 89 (2014) 042819.

[14] R. C. Feenstra, R. E. Lipsey, H. Deng, A. C. Ma and H. Mo, 'World Trade Flows: 1962–2000', NBER Working Paper Series, National Bureau of Economic Research, 2005, http://cid.econ.ucdavis.edu/data/undata/undata.html (accessed 17 November 2014).

[15] D. F. Gleich, 'PageRank beyond the web', Preprint, 2014, arXiv:cs.SI:1407.5107.

[16] D. F. Gleich, P. G. Constantine, A. D. Flaxman and A. Gunawardana, 'Tracking the random surfer: Empirically measured teleportation parameters in PageRank', *WWW 2010*, Proceedings of the 19th International Conference on World Wide Web (eds M. Rappa, P. Jones, J. Freire and S. Chakrabarti; ACM, New York, 2010) 381-390.

[17] M. S. Granovetter, *Getting a job: A study in contacts and careers*, 2nd edn (University of Chicago Press, Chicago, 1995).

[18] A. Halu, R. J. Mondragón, P. Panzarasa and G. Bianconi, 'Multiplex PageRank', *PLOS ONE*, 8 (2013) e78293.

[19] J. Heitzig, J. F. Donges, Y. Zou, N. Marwan and J. Kurths, 'Node-weighted measures for complex networks with spatially embedded, sampled, or differently sized nodes', *Eur. Phys. J. B* 85 (2012) 38.

[20] B. D. Hughes, *Random walks and random environments*, vol. I (Oxford University Press, Oxford, 1995).

[21] J. D. Hunter, 'Matplotlib: A 2D graphics environment', *Comput. Sci. Eng.* 9 (2007) no. 3, 90–95, http://matplotlib.org/.

[22] N. Jeevanjee, *An Introduction to tensors and group theory for physicists* (Birkhäuser, Basel, 2011).

[23] M. Kivelä, Multilayer networks library [software —plexmath project webpage], 2013, www.plexmath.eu/?page_id=327.

[24] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno and M. A. Porter, 'Multilayer networks'. *J. Complex Netw.* 2 (2014) no. 3, 203–271.

[25] R. Lambiotte and M. Rosvall, 'Ranking and clustering of nodes in networks with smart teleportation', *Phys. Rev. E* 85 (2012) 056107.

[26] C. D. Meyer, *Matrix analysis and applied linear algebra* (SIAM, Philadelphia, 2000).

[27] G. Menichetti, D. Remondini, P. Panzarasa, R. J. Mondragón and G. Bianconi, 'Weighted multiplex networks', *PLOS ONE* 9 (2014) no. 6, e97857.

[28] M. E. J. Newman, *Networks: An introduction* (Oxford University Press, Oxford, 2010).

[29] J. D. Noh and H. Rieger, 'Random walks on complex networks', *Phys. Rev. Lett.* 92 (2004) no. 11, 118701.

[30] L. Page, S. Brin, R. Motwani and T. Winograd, 'The PageRank citation ranking: Bringing order to the Web', *Technical Report,* Stanford InfoLab (1998).

[31] F. R. Pitts, 'The medieval river trade network of Russia revisited', *Soc. Networks* 1 (1978/79) no. 3, 285–292.

[32] K. F. Riley, M. P. Hobson and S. J. Bence, *Mathematical methods for physics and engineering* (Cambridge University Press, Cambridge, 2006).

[33] D. S. Sade, 'Sociometrics of Macaca mulatta I: Linkages and cliques in grooming matrices', *Folia Primat.* 18 (1972) no. 3, 196–223.

[34] D. S. Sade, 'Sociometrics of Macaca mulatta III: N-path centrality in grooming networks', *Soc. Networks* 11 (1989) no. 3, 273–292.

[35] M. Á. Serrano, M. Boguñá and A. Vespignani, 'Patterns of dominant flows in the world trade web', *J. Econ. Interac. Coord.* 2 (2007) no. 2, 111–124.

[36] L. Solá, M. Romance, R. Criado, J. Flores, A. Garcia del Amo and S. Boccaletti, 'Eigenvector centrality of nodes in multiplex networks', *Chaos* 23 (2013) 033131.

[37] Gross domestic product per capita in year 2000, 'GDP per capita, PPP (current international $)', http://data.worldbank.org/indicator/NY.GDP.PCAP.PP.CD (accessed 17 November 2014).