Dynamically Motivated Models for Multiplex Networks[1]

Daryl DeFord

Dartmouth College
Department of Mathematics

Santa Fe institute
Inference on Networks:
Algorithms, Phase Transitions, New Models, and New Data
December 14–18, 2015

[1] Joint work with Scott Pauls

Abstract

In this talk I will present a dynamically motivated model for a class of multiplex networks that provides natural extensions for many of the standard network tools to the multiplex setting, including centralities, diffusion, and clustering. I will also present some spectral results related to the Laplacian formulation of this model for diffusion and clustering applications on multiplex networks.

Dartmouth

Outline

1. Introduction
   - Abstract
   - Outline
2. Motivation
   - Setup
   - World Trade Web
   - Social Networks
   - Goals
3. Methodology
   - Projections
   - Simplifications
4. Applications
   - Adjacency
   - Diffusion
   - Random Walks
5. Acknowledgements

Dartmouth

Motivation

- In the terminology introduced in[2] these are the diagonal, node–aligned multilayer networks.

[2] Kïvela et al.: *Multilayer Networks*, Journal of Complex Networks, July 2014.

Motivation

- In the terminology introduced in[2] these are the diagonal, node–aligned multilayer networks.
- Disaggregated Data
  - A single set of objects of interest
  - Many different types of relations or connections
  - Intra–object interactions that are distinct from the inter–layer dynamics

[2] Kïvela et al.: *Multilayer Networks*, Journal of Complex Networks, July 2014.

Motivation

- In the terminology introduced in[2] these are the diagonal, node–aligned multilayer networks.
- Disaggregated Data
  - A single set of objects of interest
  - Many different types of relations or connections
  - Intra–object interactions that are distinct from the inter–layer dynamics
- Examples:
  - World Trade Web
  - Social Networks
  - Neural Networks
  - Many others ...

[2] Kïvela et al.: *Multilayer Networks*, Journal of Complex Networks, July 2014.

WTW Setup

- Nodes → Countries
- Edges → Trade Volume
- Disaggregation: Commodity Type

WTW Setup

- Nodes → Countries
- Edges → Trade Volume
- Disaggregation: Commodity Type

Layer	Description	Volume	% Total	Transitivity
0	Food and live animals	291554437	5.1	.82
1	Beverages and tobacco	48046852	0.9	.67
2	Crude materials	188946835	3.3	.79
3	Mineral fuels	565811660	10.0	.62
4	Animal and vegetable oils	14578671	0.3	.64
5	Chemicals	535703156	9.5	.83
6	Manufactured Goods	790582194	13.9	.87
7	Machinery	2387828874	42.1	.85
8	Miscellaneous manufacturing	736642890	13.0	.83
9	Other commodities	107685024	1.9	.56
All	Aggregate Trade	5667380593	100	.93

Table : Commodity information for the 2000 WTW

WTW Aggregate Figure

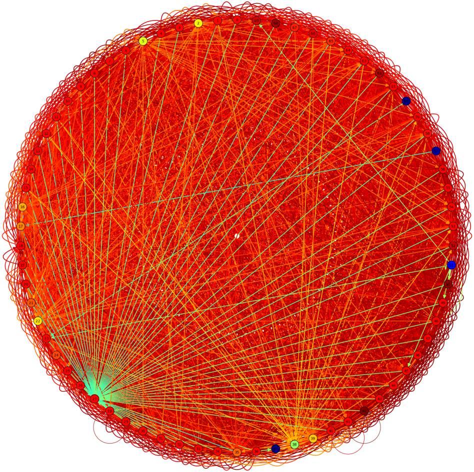

Figure : Aggregate 2000 World Trade Web[3]

[3] Feenstra et al.: *World Trade Flows: 1962–2000*, Working Paper 11040, NBER.

WTW Dynamics

- Edge weights reflect the volume of trade flow

[4] Foti et al.: *Stability of the World Trade Web Over Time: An Extinction Analysis*, Journal of Economic Dynamics and Control, September 2013.

WTW Dynamics

- Edge weights reflect the volume of trade flow
- Stability analysis[4] can reveal sensitivity of the global network to various perturbations. This approach can be refined by considering:

[4] Foti et al.: *Stability of the World Trade Web Over Time: An Extinction Analysis*, Journal of Economic Dynamics and Control, September 2013.

WTW Dynamics

- Edge weights reflect the volume of trade flow
- Stability analysis[4] can reveal sensitivity of the global network to various perturbations. This approach can be refined by considering:
    - Exchanges between the various industries within each country

[4] Foti et al.: *Stability of the World Trade Web Over Time: An Extinction Analysis*, Journal of Economic Dynamics and Control, September 2013.

WTW Dynamics

- Edge weights reflect the volume of trade flow
- Stability analysis[4] can reveal sensitivity of the global network to various perturbations. This approach can be refined by considering:
  - Exchanges between the various industries within each country
  - Net trade surpluses and deficits

[4] Foti et al.: *Stability of the World Trade Web Over Time: An Extinction Analysis*, Journal of Economic Dynamics and Control, September 2013.

WTW Dynamics

- Edge weights reflect the volume of trade flow
- Stability analysis[4] can reveal sensitivity of the global network to various perturbations. This approach can be refined by considering:
  - Exchanges between the various industries within each country
  - Net trade surpluses and deficits
- A model that allows us to distinguish intra–country dynamics from international dynamics permits a more nuanced view of the data and hence a more complete analysis.

[4] FOTI ET AL.: *Stability of the World Trade Web Over Time: An Extinction Analysis*, Journal of Economic Dynamics and Control, September 2013.

Social Networks

- Survey Data
  - Heterogeneous layers

[5] Banerjee et al.: *The Diffusion of Microfinance*, Science, (2013).

Social Networks

- Survey Data
  - Heterogeneous layers
  - Hierarchical layers

[5] Banerjee et al.: *The Diffusion of Microfinance*, Science, (2013).

Social Networks

- Survey Data
  - Heterogeneous layers
  - Hierarchical layers
  - (un–)Directed layers

[5] BANERJEE ET AL.: *The Diffusion of Microfinance*, Science, (2013).

Social Networks

- Survey Data
  - Heterogeneous layers
  - Hierarchical layers
  - (un–)Directed layers
- Information dynamics[5]

[5] Banerjee et al.: *The Diffusion of Microfinance*, Science, (2013).

Social Networks

- Survey Data
  - Heterogeneous layers
  - Hierarchical layers
  - (un–)Directed layers
- Information dynamics[5]
  - Diffusive
  - Transactional

[5] Banerjee et al.: *The Diffusion of Microfinance*, Science, (2013).

Social Layers

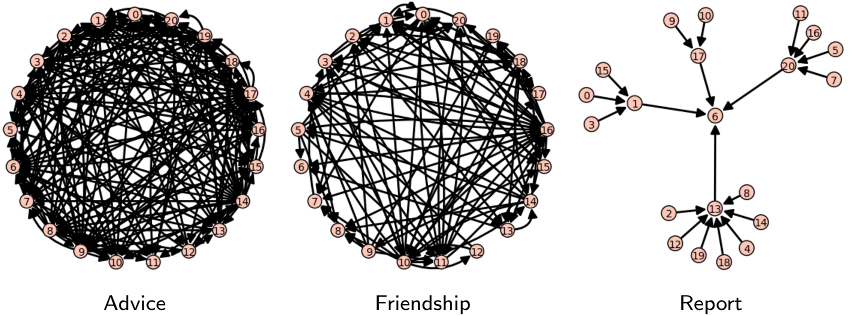

Figure : Krackhardt Hi-Tech Manager Relationships[6]

[6] KRACKHARDT: *Cognitive Social Structures*, Social Networks (1987).

Goals

We want a model that

- preserves the dynamics captured by the individual layers (data)

Goals

We want a model that

- preserves the dynamics captured by the individual layers (data)
- permits control over the mixing at the nodes themselves

Goals

We want a model that

- preserves the dynamics captured by the individual layers (data)
- permits control over the mixing at the nodes themselves
- allows for the generalization of standard network metrics and processes

Goals

We want a model that

- preserves the dynamics captured by the individual layers (data)
- permits control over the mixing at the nodes themselves
- allows for the generalization of standard network metrics and processes

- Refined Aggregation

Goals

We want a model that

- preserves the dynamics captured by the individual layers (data)
- permits control over the mixing at the nodes themselves
- allows for the generalization of standard network metrics and processes

- Refined Aggregation
- Layer effects pass through copies. Copies don't interact directly.

Notation

- n nodes
- k layers
- v a $nk \times 1$ vector of "quantities"
- v_j^i the quantity at node j on layer i
- i and ℓ layer indices
- j node index

General Approach

Two–step iterative process

- Initialize the $nk \times 1$ vector of "quantities" v

General Approach

Two–step iterative process

- Initialize the $nk \times 1$ vector of “quantities” v
- Step 1: Inter–layer dynamics: given by the layer relationships

General Approach

Two–step iterative process

- Initialize the $nk \times 1$ vector of "quantities" v
- Step 1: Inter–layer dynamics: given by the layer relationships
  - Linear Case: Given a collection of dynamic operators $\{D_i\}$, one for each layer, form $D = \mathrm{diag}(D_1, D_2, \ldots, D_n)$.

General Approach

Two–step iterative process

- Initialize the $nk \times 1$ vector of "quantities" v
- Step 1: Inter–layer dynamics: given by the layer relationships
    - Linear Case: Given a collection of dynamic operators $\{D_i\}$, one for each layer, form $D = \mathrm{diag}(D_1, D_2, \ldots, D_n)$.
- Step 2: Intra–node mixing:

General Approach

Two–step iterative process

- Initialize the $nk \times 1$ vector of "quantities" v
- Step 1: Inter–layer dynamics: given by the layer relationships
  - Linear Case: Given a collection of dynamic operators $\{D_i\}$, one for each layer, form $D = \mathrm{diag}(D_1, D_2, \ldots, D_n)$.
- Step 2: Intra–node mixing:
  - Mix the effects of the D_i as a scaled, convex combination of the resulting values at each copy of each node.

General Approach

Two–step iterative process

- Initialize the $nk \times 1$ vector of "quantities" v
- Step 1: Inter–layer dynamics: given by the layer relationships
  - Linear Case: Given a collection of dynamic operators $\{D_i\}$, one for each layer, form $D = \text{diag}(D_1, D_2, \ldots, D_n)$.
- Step 2: Intra–node mixing:
  - Mix the effects of the D_i as a scaled, convex combination of the resulting values at each copy of each node.

$$(v')^i_j = \alpha^i_j \sum_{\ell=1}^{k} c_j^{i,\ell} (Dv)^\ell_j$$

General Approach

Two–step iterative process

- Initialize the $nk \times 1$ vector of "quantities" v
- Step 1: Inter–layer dynamics: given by the layer relationships
    - Linear Case: Given a collection of dynamic operators $\{D_i\}$, one for each layer, form $D = \operatorname{diag}(D_1, D_2, \ldots, D_n)$.
- Step 2: Intra–node mixing:
    - Mix the effects of the D_i as a scaled, convex combination of the resulting values at each copy of each node.

$$(v')^i_j = \alpha^i_j \sum_{\ell=1}^{k} c^{i,\ell}_j (Dv)^\ell_j$$

- Return to Step 1

Step 2: Scaled Projections

- Orthogonal projections onto the "node subspaces"
- Gather and redistribute
- $c_j^{i,\ell}$ – "pass–through proportion"
- α_j^i – scaling coefficient

Matrix Representations

Step 2 can be expressed as a single block matrix acting on v, with $C^{i,\ell} = \mathrm{diag}(\alpha_1^\ell c_1^{i,\ell}, \alpha_2^\ell c_2^{i,\ell}, \ldots, \alpha_n^\ell c_n^{i,\ell})$:

$$M = \begin{bmatrix} C^{1,1} & C^{1,2} & \cdots & C^{1,k} \\ C^{2,1} & C^{2,2} & \cdots & C^{2,k} \\ \vdots & \ddots & \ddots & \vdots \\ C^{k,1} & C^{k,2} & \cdots & C^{k,k} \end{bmatrix},$$

The final multiplex dynamic operator is a product of the layer dynamics matrix D and the redistribution matrix M

$$\mathfrak{D} = MD = \begin{bmatrix} C^{1,1}D_1 & C^{1,2}D_2 & \cdots & C^{1,k}D_k \\ C^{2,1}D_1 & C^{2,2}D_2 & \cdots & C^{2,k}D_k \\ \vdots & \ddots & \ddots & \vdots \\ C^{k,1}D_1 & C^{k,2}D_2 & \cdots & C^{k,k}D_k \end{bmatrix}.$$

Unified Node

The **unified node model** assumes that each node has a distinct set of weighted preferences between its copies.

Unified Node

The **unified node model** assumes that each node has a distinct set of weighted preferences between its copies.

As an example, in the WTW network each country can evaluate each commodity's importance as the total volume of trade at that country in the respective layer. Then the total trade flow at each country can be redistributed proportionally to these weighted degrees.

Hierarchical Layers

A further natural simplification occurs if we assume that the global network has an ordering of layers, so that the effect of layer i on layer ℓ is fixed across all nodes. In this **hierarchical layer model** the blocks $C^{i,\ell}$ are just scalar multiples of I_n.

In the absence of application specific choices of $C^{i,\ell}$ the layer densities provide a natural hierarchy either by taking $C^{i,\ell}$ to be the density itself or the ratio of the density of layer i to the density of layer j.

When the layer dynamics are the adjacency matrices, this simplification is the asymmetric influence matrix introduced in[7].

[7] Solá et al. Eigenvector centrality of nodes in multiplex networks. Chaos (2013).

Equidistribution

The simplest version of this operator, the **equidistribution model**, sets $c_j^{i,\ell} = \frac{1}{k}$ for all $1 \leq j \leq n$ and $1 \leq i, \ell \leq k$. At every step, this operator simply averages the quantities at each node copy. This is a natural simplification for applications where the flow is equally likely to move between layers or represents the probabilities of a binary process.

Centrality Score Comparison

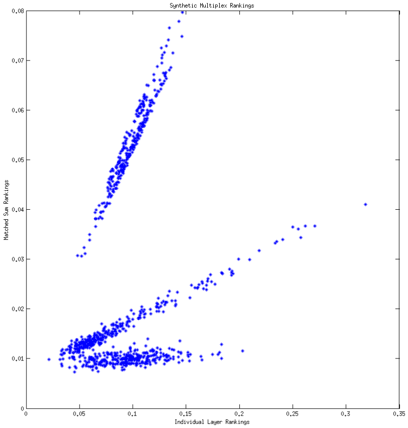

Figure : Monoplex Comparison

Centrality Score Comparison

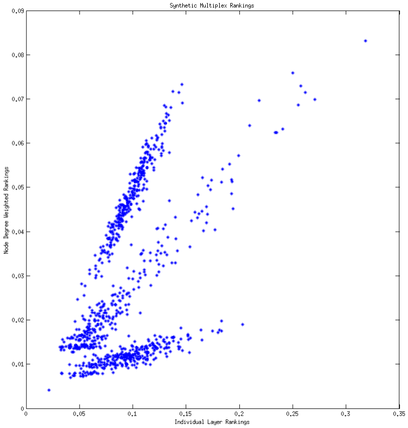

Figure : Unified Node Comparison

Centrality Score Comparison

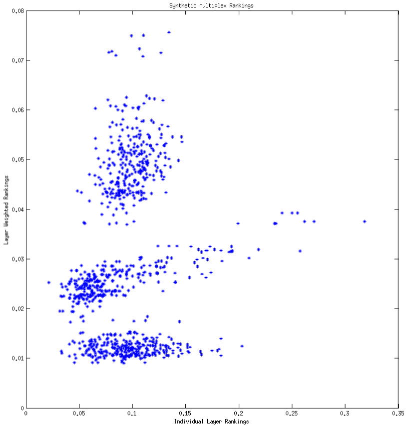

Figure : Hierarchical Layer Comparison

Dartmouth

Centrality Score Comparison

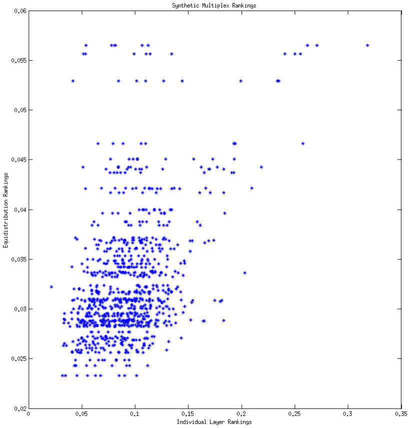

Figure : Equidistribution Comparison

Multiplex Diffusion

To extend the standard interpretation of the Laplacian operator to the multiplex setting, we allow the $c_j^{i,\ell}$ to represent the proportion of the effect on layer ℓ that passes to the jth node on layer i:

$$\frac{dv_j^i}{dt} = K \sum_{\ell=1}^{k} c_j^{i,\ell} \sum_{n_j^\ell \sim n_m^\ell} (v_j^\ell - v_m^\ell).$$

Here K is the diffusion constant and the $c_j^{i,\ell}$ represent the proportion of the effect on layer ℓ that passes through to n_j^i. Linear algebraically, this is:

$$\frac{dv_j^i}{dt} = K \left[\sum_{\ell=1}^{k} c_j^{i,\ell} L^\ell v^\ell \right]_j \tag{1}$$

Eigenvalue Bounds

The theory of Hermitian matrices, and in particular the Weyl bounds, allow us to bound the eigenvalues of this diffusion operator using the hierarchical layer or equidistribution models:

- Fiedler Value: $\max_i(\lambda_f^i) \leq k\lambda_f \leq \lambda_f^m + \sum_{j\neq\ell}\lambda_1^j$,
- Leading Value: $\max_i(\lambda_1^i) \leq k\lambda_1 \leq \sum_i \lambda_1^i$,

These bounds are special cases of the following more general but less computationally feasible bounds:

$$\max_i(\lambda_{n-\ell}^i) \leq k\lambda_{n-\ell} \leq \min_{J\vdash n+k-(\ell+1)}\left(\min_{\sigma\in S_n}\left(\sum_{i=1}^{k}\lambda_{j_i}^{\sigma(i)}\right)\right),$$

Preserved Properties

If we take the original layer dynamics to be the corresponding random walk matrices then many of the matrix properties are preserved in the equidistribution model:

- Stochastic
- Irreducible
- Primitive

[8] De Domenico et al.: *Navigability of interconnected networks under random failures*, PNAS (2014).

[9] Trpevski et al: *Discrete–time distributed consensus on multiplex networks.*, New Journal of Physics (2014).

Preserved Properties

If we take the original layer dynamics to be the corresponding random walk matrices then many of the matrix properties are preserved in the equidistribution model:

- Stochastic
- Irreducible
- Primitive

This version of the multiplex random walk operator is equivalent to versions used in[8] and[9] to model transportation networks and information diffusion respectively.

[8] De Domenico et al.: *Navigability of interconnected networks under random failures*, PNAS (2014).

[9] Trpevski et al: *Discrete–time distributed consensus on multiplex networks.*, New Journal of Physics (2014).

Preserved Properties

If we take the original layer dynamics to be the corresponding random walk matrices then many of the matrix properties are preserved in the equidistribution model:

- Stochastic
- Irreducible
- Primitive

This version of the multiplex random walk operator is equivalent to versions used in[8] and[9] to model transportation networks and information diffusion respectively.

Additionally, we may project the random walk to the original node space to derive a $n \times n$ transition matrix. This is an example of the refined aggregation aspect of our model.

[8] De Domenico et al.: *Navigability of interconnected networks under random failures*, PNAS (2014).

[9] Trpevski et al: *Discrete–time distributed consensus on multiplex networks.*, New Journal of Physics (2014).

WTW Applications

- Commute Time Clustering
  - Distance proxy for gravity model of trade

WTW Applications

- Commute Time Clustering
  - Distance proxy for gravity model of trade

- Random Walk Betweenness Centrality
  - Aggregate: US/Canada
  - Individual Layers: Sources and sinks
  - Multiplex: Good measure of global flow

Conclusions

- Our operator represents a dynamically motivated approach to better understanding the properties of multiplex networks
- Control of the intra–node mixing allows us to examine a continuum of results for each data set that reveals different aspects of the underlying data.
- This approach generalizes several standard methodologies from both monoplex and mutliplex perspectives.

That's all...

Thank You!

Small Example

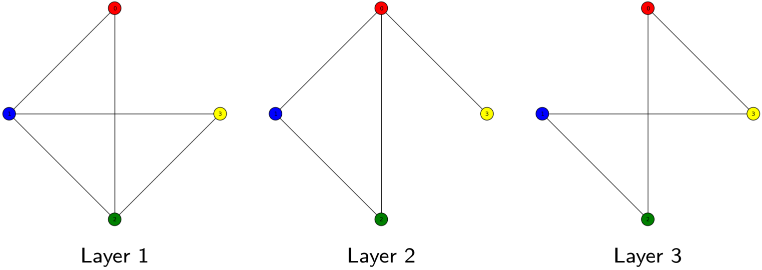

Figure : A toy multiplex model

Eigenvector Centrality

Node	Level 1	Level 2	Level 3	$\hat{D}$
1	.5883	.5	.7071	.6438
2	.3922	.5	.4714	.4416
3	.3922	.5	.4714	.4190
4	.5883	.5	.2357	.4636

Table : Eigenvector centrality scores for the toy multiplex network

In[10] the authors use diffusion centrality as a proxy for their communication centrality. This approach also translates directly to this multiplex operator.

[10] Banerjee et al.: *The Diffusion of Microfinance*, Science (2013)