Machine Intelligence on Graphs

Danilo Mandic

Imperial College London, UK

Ackn: Ljubisa Stankovic, Milos Dakovic, Bruno Scalso Dees, Shengxi Li, Yao Lei Xu

December 16, 2021

Lecture Outline

1. Introduction and relevance
2. Problem statement: An ollustrative example
3. Building a graph from signal measurements
4. System on a graph and graph filtering
5. Graph clustering
6. Dimensionality reduction through graphs
7. Graphs and tensors
8. Graphs for finance
9. Multi-graph tensor network (MGTN)
10. Graph cuts for portfolio managament
11. Graphs for public transportation planning and analysis

Graph Data Analytics at Imperial College

Comm.3 Principles of communication networks

DR A.G. CONSTANTINIDES

10 lectures in the autumn term.

Basic ideas on connectivity constraints, limited and unlimited reachability of switching centres. The location of switching centres for optimum network covering.

Traffic flow and generalized network flows. The maximum flow minimum cut theorem.

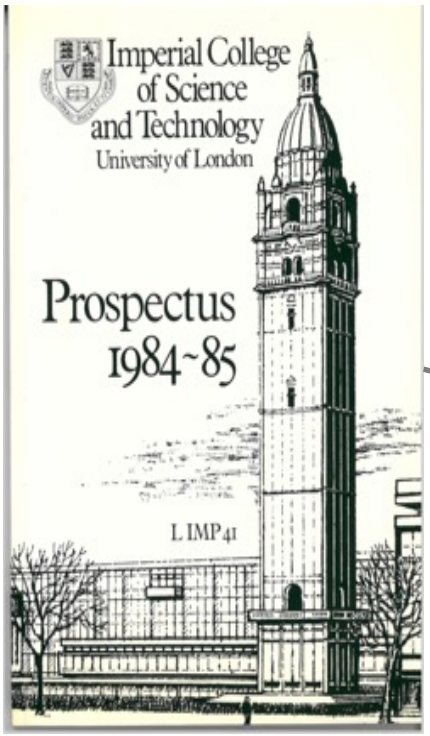
Communication networks with limited link capacities. The routing problem. Spanning tree and shortest spanning tree. Routing subject to congestion.

- In the early days, the main goal was to optimize the graphs themselves, rather than signals on graphs! Fast forward 40 years, we are in a position to optimize both the graphs and the graph signals!

Technical courses

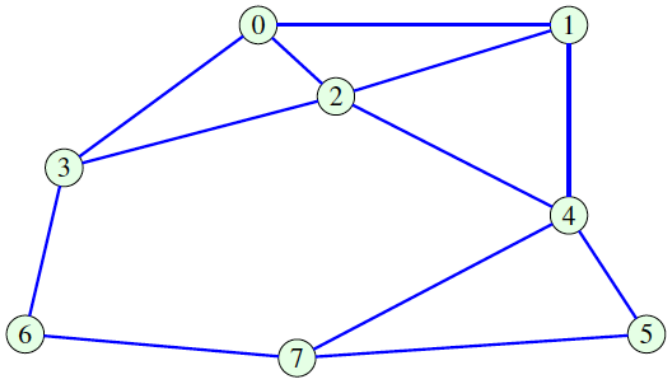
Introduction to power systems
Electrical drive systems
Advanced power systems
Electrical machines II
Statistics
Graph theory

- Graphs were taught within Communication Networks and in Business School.

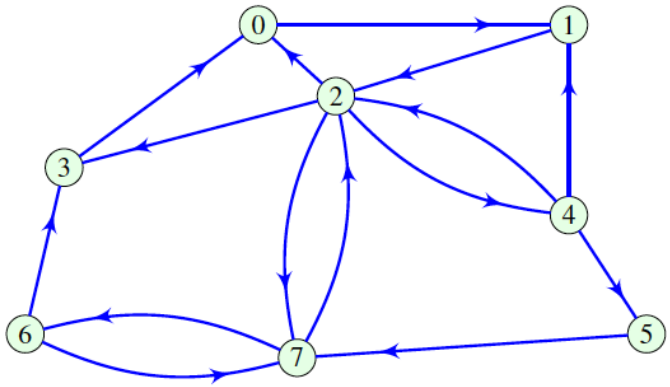


- Beginnings of Graph Theory at EEE Department, Imperial College London.

Graph basics



(a)



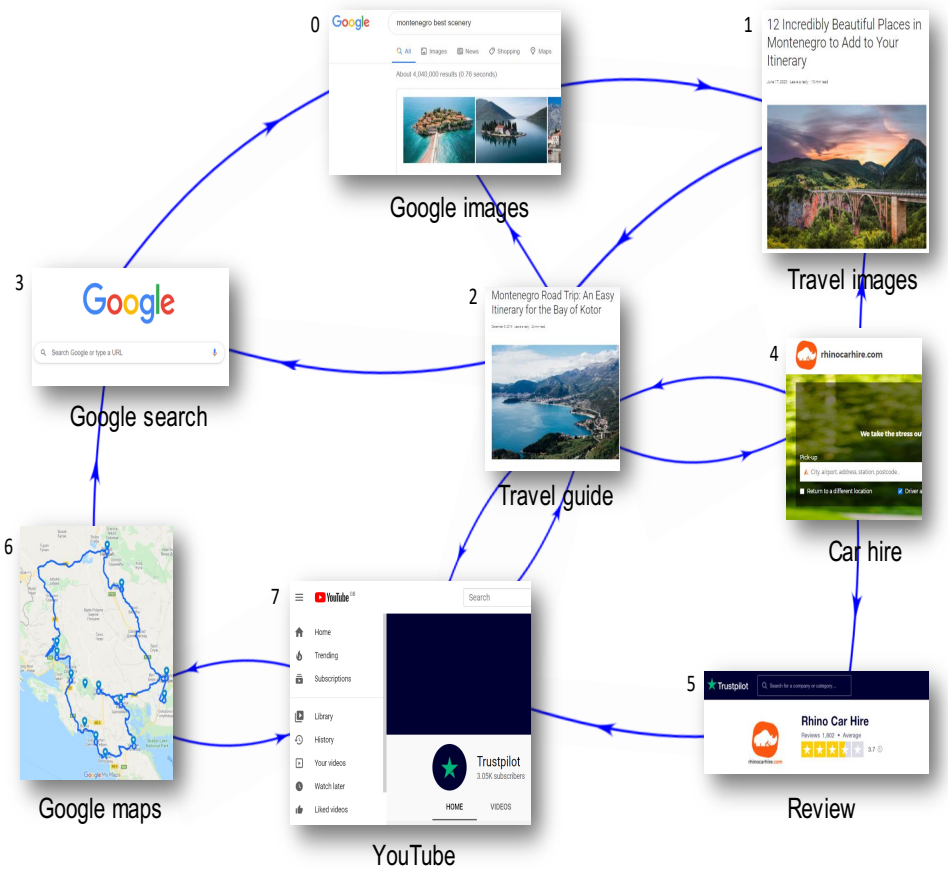
(b)

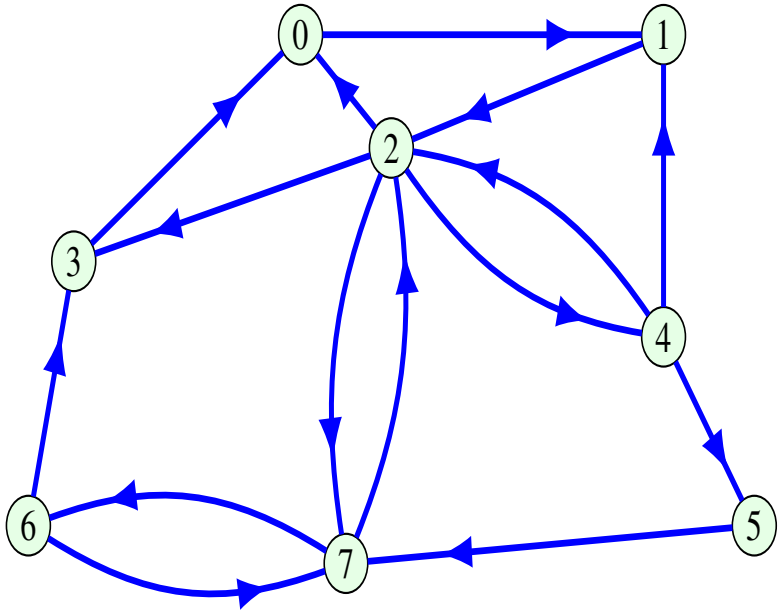
Figure 1: Typical graph structures. (a) Undirected graph and (b) Directed graph.

$$\mathbf{A}_{\text{un}} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array}$$

$$\mathbf{A}_{\text{dir}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Modern Applications: Graphs for recommender systems





The adjacency matrix from the previous slide describes our search for a holiday destination!

Graph operators

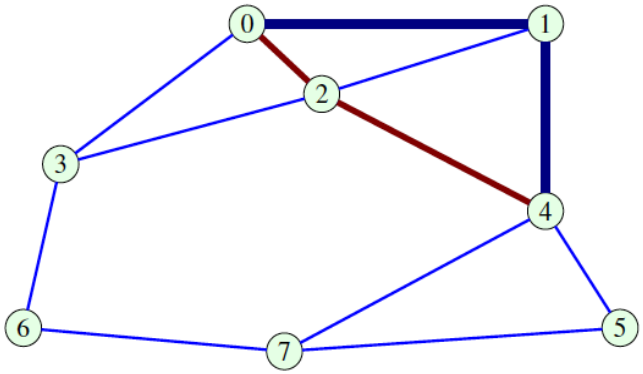


Figure 4: Walks of length $K = 2$ from vertex 1 to vertex 5.

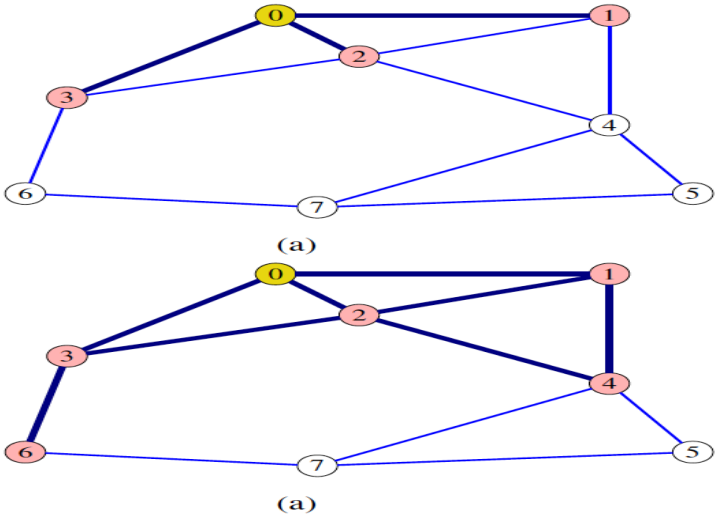


Figure 5: The K-neighborhoods of vertex 0 for the graph from Fig. 4, where: (a) $K = 1$ and (b) $K = 2$. The neighboring vertices are shaded.

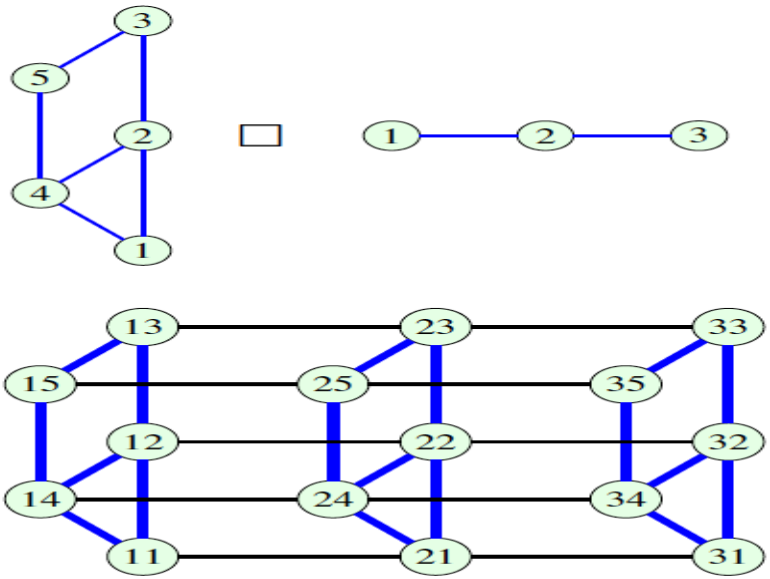


Figure 8: Cartesian product of two graphs

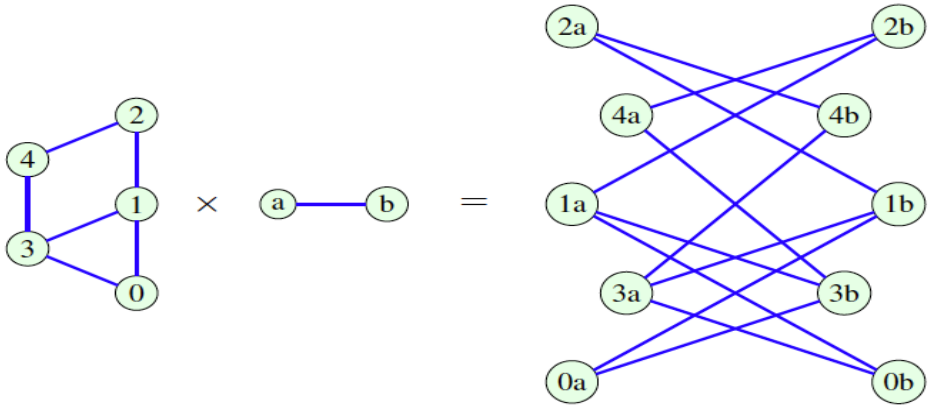
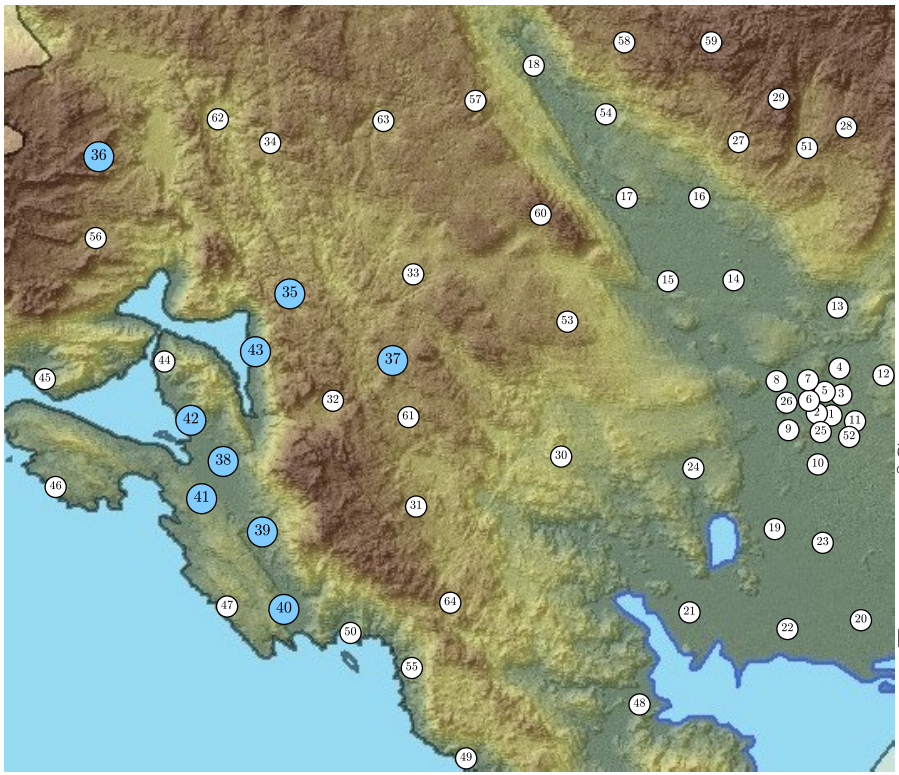
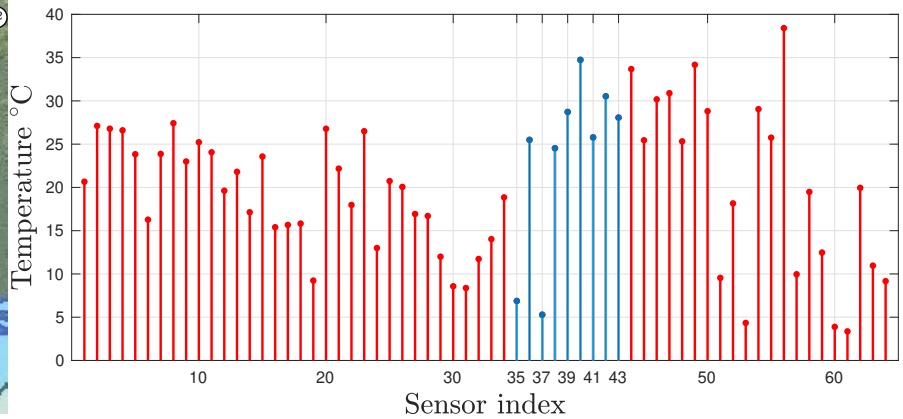


Figure 7: Kronecker (tensor) product of two graphs.

An Illustrative Example

Consider a multi-sensor setup for measuring a temperature field in a known geographical region. The temperature sensing locations are chosen according to the significance of a particular geographic area to local users, with $N = 64$ sensing points in total.





An Illustrative Example

- Classical Data Analytics requires an arrangement of the quintessentially spatial temperature samples into a linear structure
- “Lexicographic" ordering is not amenable to exploiting the spatial information related to the actual sensor arrangement, dictated by the terrain.
- This exemplifies that even a most routine temperature measurement setup requires a more complex estimation structure than the simple linear one corresponding to the classical signal processing framework
- To introduce a “situation-aware" noise reduction scheme for the temperature field, we proceed to explore a graph-theoretic framework to this problem, starting from a local signal average operator. An effective estimation strategy should include domain knowledge.

Graph Topology (Edges and Weights) I

There are three possible classes of problems which dictate the definition of graph edges:

- **Geometry of the vertex positions:** The distances between vertex positions play a crucial role in establishing relations between the sensed data. In many physical processes, the presence of edges and their associated connecting weights is defined based on the vertex distances. An exponential function of the Euclidean distance between vertices, r_{mn}, may be used, where for a given distance threshold, τ,

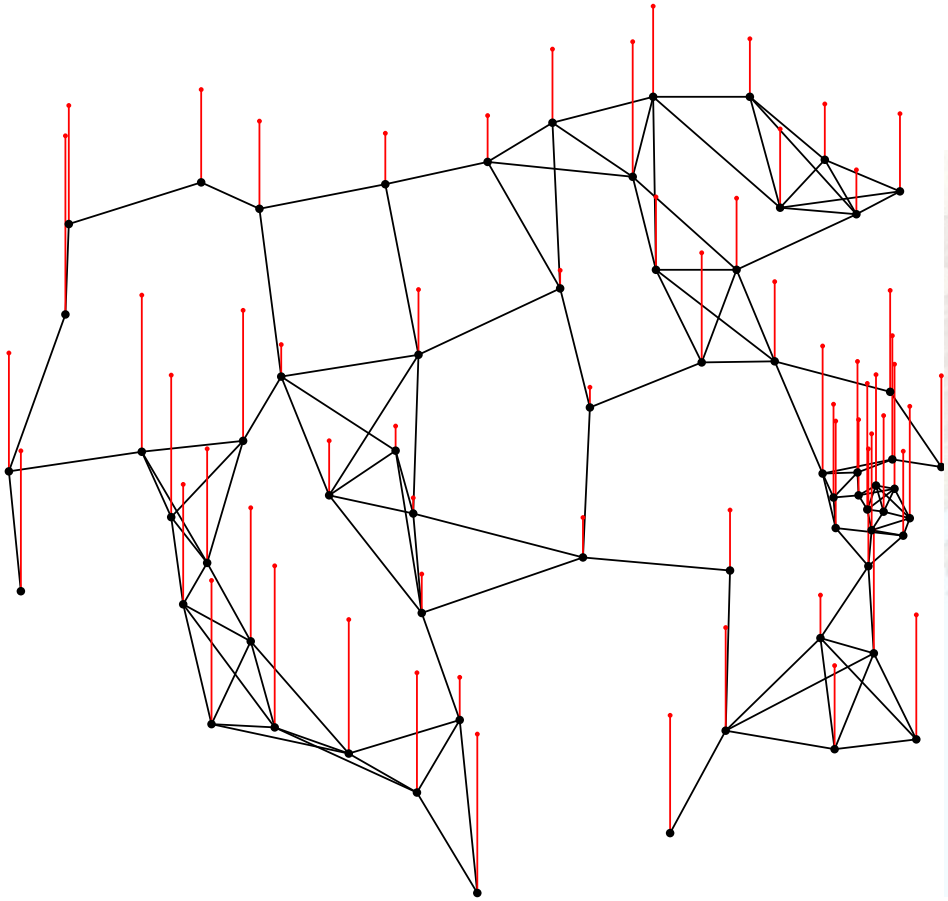
$$W_{mn} = e^{-r_{mn}^2/\alpha} \text{ or } W_{mn} = e^{-r_{mn}/\alpha}$$

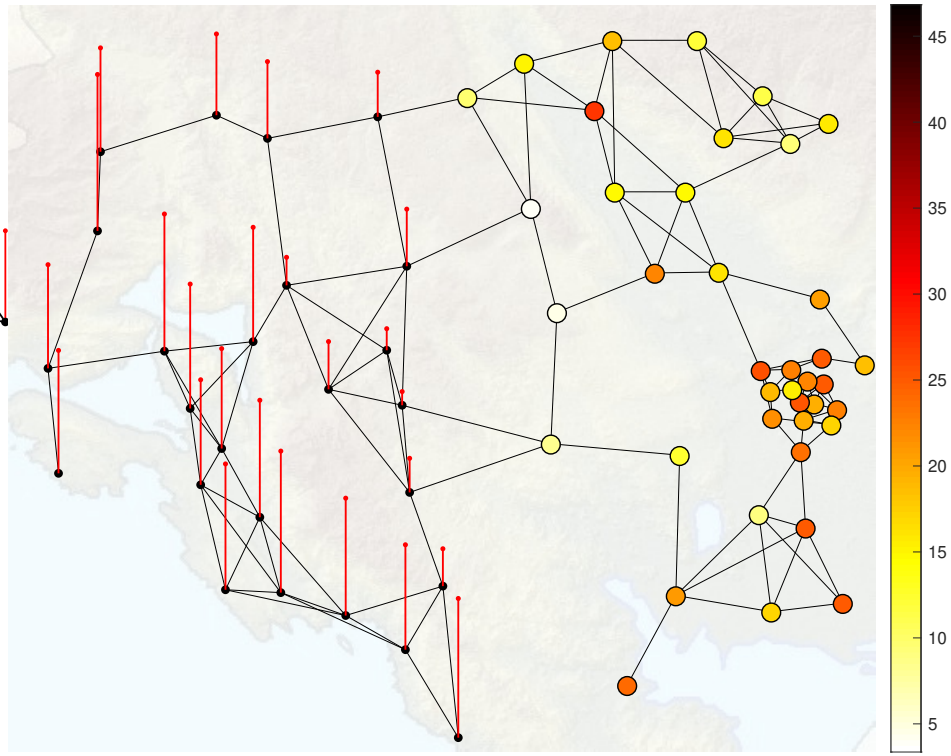
if $r_{mn} < \tau$ and $W_{mn} = 0$ for $r_{mn} \geq \tau$. This form has been used in the graph in Fig. 2, whereby the altitude difference, h_{mn}, was accounted for as $W_{mn} = e^{-r_{mn}/\alpha} e^{-h_{mn}/\beta}$.

Graph Topology (Edges and Weights) II

- **Physically well defined relations among the sensing positions:** Examples include electric circuits, linear heat transfer systems, spring-mass systems, and various forms of networks like social, computer or power networks. In these cases, the edge weights are defined as a part of problem definition.
- **Data similarity dictates the underlying graph topology:** This scenario is the most common in image and biomedical signal processing. Various approaches can be used to define data similarity, including the correlation matrix between the signals at various sensors or the corresponding inverse covariance (precision) matrix. Learning a graph (its edges) based on the set of the available data is an interesting and currently extensively studied research area.

Graph Signal





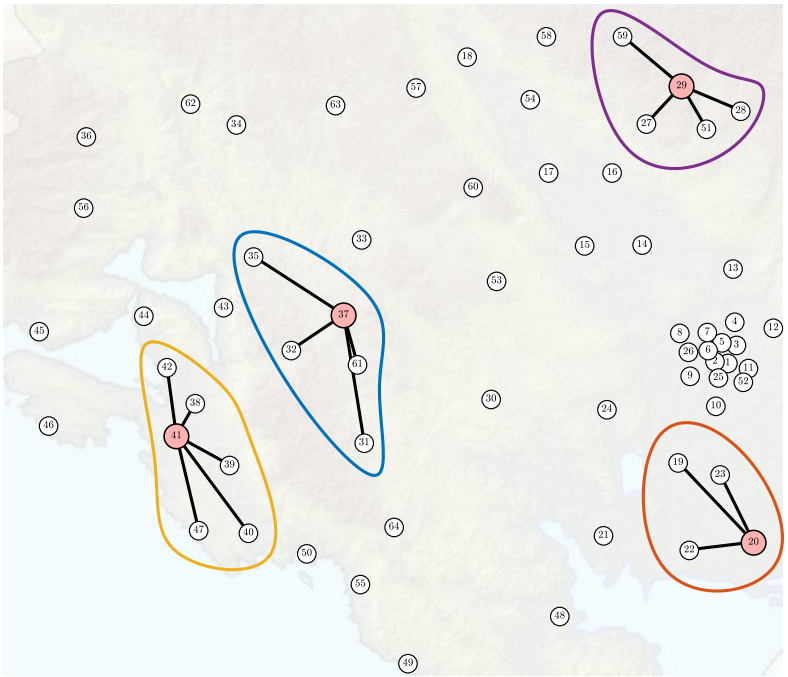
From a multi-sensor measurement to a graph signal

Local Signal Average

- For example, for the sensing points $n = 20$ and $n = 37$, the "domain knowledge aware" local estimation takes the form

$$y(20) = x(20) + x(19) + x(22) + x(23) \tag{1}$$

$$y(37) = x(37) + x(32) + x(33) + x(35) + x(61). \tag{2}$$



Local Signal Average

- In classical Signal Processing this can be achieved through a moving average operator, through averaging across the neighboring neighboring nodes, in the linear graph.
- Since the sensor network measures a set of related temperatures from irregularly spaced sensors, an effective estimation strategy should include domain knowledge.
- For example, for the sensing points $n = 20$ and $n = 37$, the “domain knowledge aware" local estimation takes the form

$$y(20) = x(20) + x(19) + x(22) + x(23) \tag{3}$$

$$y(37) = x(37) + x(32) + x(33) + x(35) + x(61). \tag{4}$$

- The full set of relations among the sensing points can be arranged into the matrix form

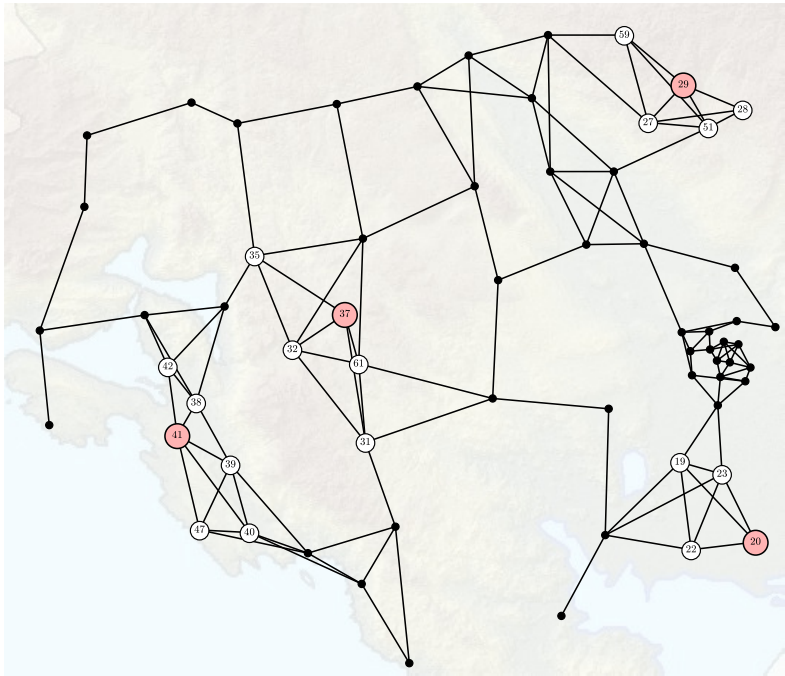
$$\mathbf{y} = \mathbf{x} + \mathbf{A}\mathbf{x}, \tag{5}$$

Adjacency Matrix

- The full set of relations among the sensing points can be arranged into the matrix form

$$\mathbf{y} = \mathbf{x} + \mathbf{A}\mathbf{x}, \tag{6}$$

- The matrix $\mathbf{A}$ is the **connectivity or adjacency matrix** of a graph. It indicates the neighboring sensing locations for each n. The elements of $\mathbf{A}$ are either 1 (vertices are related) or 0 (not related).



Weighted Graph

- To emphasize our trust in a particular sensor (i.e., to model sensor relevance), a weighting scheme may be imposed on the edges (connectivity) between the sensing points,

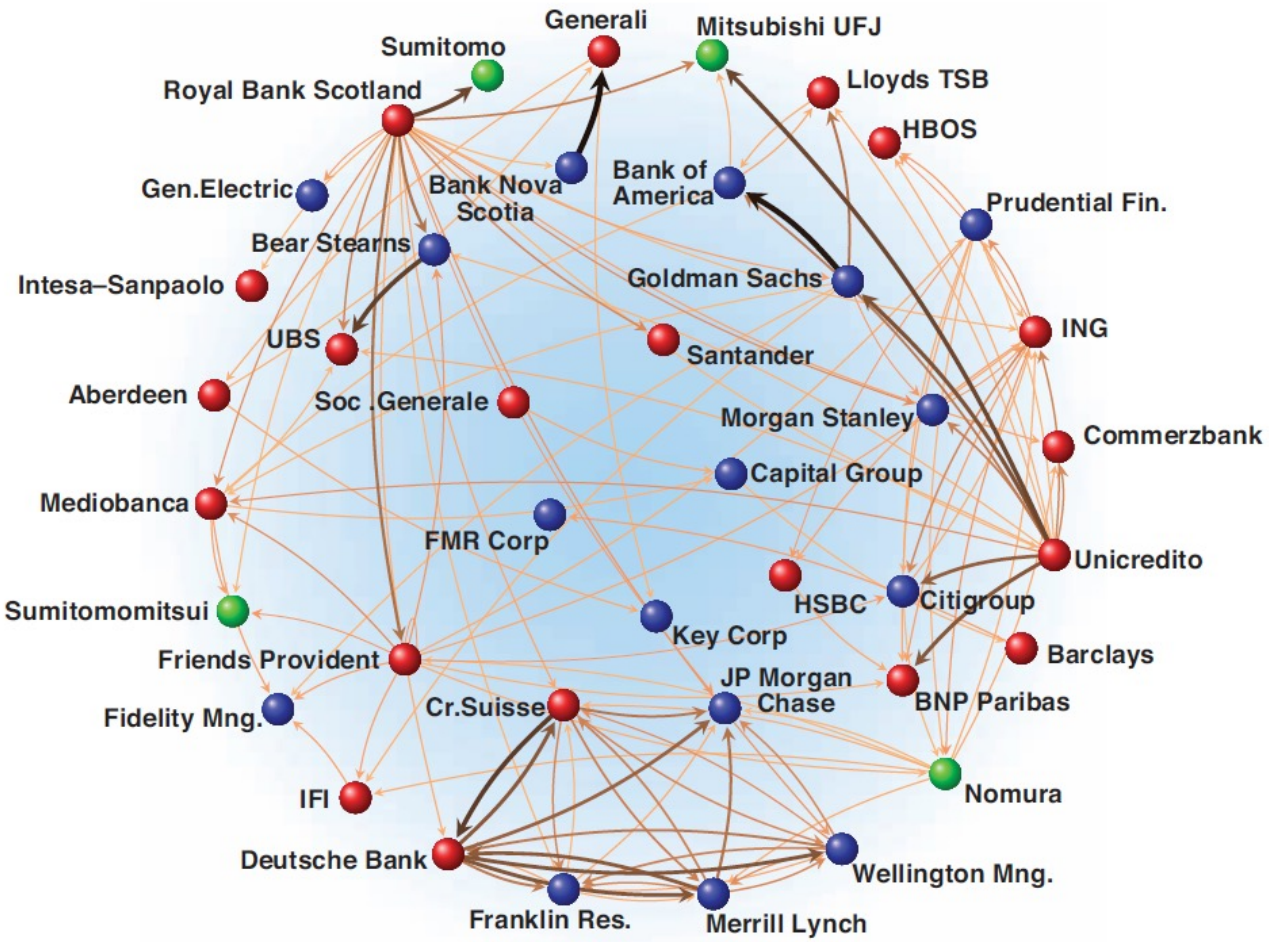
$$y(n) = x(n) + \sum_{m \neq n} W_{nm} x(m). \tag{7}$$

- We have now arrived at a weighted graph, whereby each edge has an associated weight, W_{nm},
- A matrix form of a weighted cumulative graph signal

$$\mathbf{y} = \mathbf{x} + \mathbf{W}\mathbf{x}. \tag{8}$$

- The weighting coefficients within the estimate for each $y(n)$ should sum up to unity.

International financial network: A weighted graph



International financial networks

- The graph represents relations (edges) among major financial institutions (nodes) across the world
- The network shows a high connectivity among the financial institutions that have mutual share-holdings and closed loops involving several nodes
- This indicates that the financial sector is strongly interdependent, which may affect market competition and systemic risk and make the network vulnerable to instability

European Blue: North American Green: Asian

Image from Schweizer *et al.*, Science, vol. 325, pp. 422-425, 2009

Degree Matrix and Laplacian

- A normalized form of (8)

$$\mathbf{y} = \frac{1}{2}(\mathbf{x} + \mathbf{D}^{-1}\mathbf{W}\mathbf{x}), \tag{9}$$

- The diagonal normalization matrix, $\mathbf{D}$, is called **the degree matrix**, are $D_{nn} = \sum_m W_{nm}$.
- An important operator for graph signal processing is the **graph Laplacian**, $\mathbf{L}$, which is defined as

$$\mathbf{L} = \mathbf{D} - \mathbf{W}$$

is a combination of the degree matrix and weighting matrix.

An illustrative example

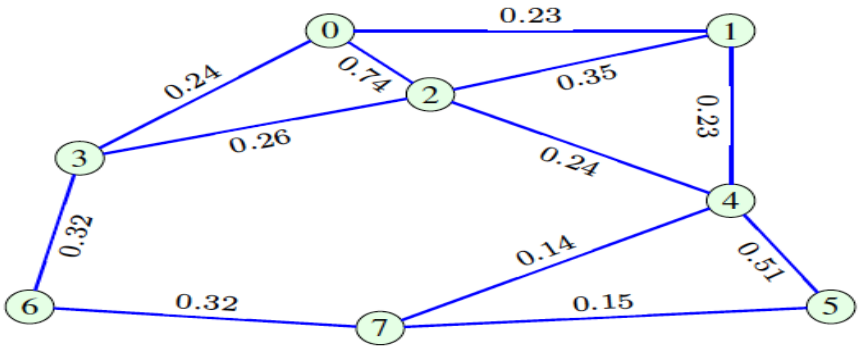


Figure 2: An example of a weighted graph.

Definition: A degree matrix, $\mathbf{D}$, for an undirected graph is a diagonal matrix with the diagonal elements, D_{mm}, equal to the sum of the weights of all edges connected to the vertex m, that is

$$D_{mm} = \sum_n W_{mn}.$$

For an unweighted and undirected graph, the value of the element D_{mm} is equal to the number of edges connected to the m-th vertex.

For the undirected weighted graph from Fig. 2, the degree matrix is given by

$$\mathbf{D} = \begin{bmatrix} 1.21 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.81 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.59 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.82 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.12 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.66 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.64 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.61 \end{bmatrix}. \quad (5)$$

$$\mathbf{W} = \begin{bmatrix} 0 & 0.23 & 0.74 & 0.24 & 0 & 0 & 0 & 0 \\ 0.23 & 0 & 0.35 & 0 & 0.23 & 0 & 0 & 0 \\ 0.74 & 0.35 & 0 & 0.26 & 0.24 & 0 & 0 & 0 \\ 0.24 & 0 & 0.26 & 0 & 0 & 0 & 0.32 & 0 \\ 0 & 0.23 & 0.24 & 0 & 0 & 0.51 & 0 & 0.14 \\ 0 & 0 & 0 & 0 & 0.51 & 0 & 0 & 0.15 \\ 0 & 0 & 0 & 0.32 & 0 & 0 & 0 & 0.32 \\ 0 & 0 & 0 & 0 & 0.14 & 0.15 & 0.32 & 0 \end{bmatrix}.$$

Definition: The Laplacian matrix is defined as

$$\mathbf{L} = \mathbf{D} - \mathbf{W}.$$

For an undirected graph the Laplacian matrix is symmetric $\mathbf{L} = \mathbf{L}^T$, while e.g. the Laplacian for the weighted graph from Fig. 2 is

$$\mathbf{L} = \begin{bmatrix} 1.21 & -0.23 & -0.74 & -0.24 & 0 & 0 & 0 & 0 \\ -0.23 & 0.81 & -0.35 & 0 & -0.23 & 0 & 0 & 0 \\ -0.74 & -0.35 & 1.59 & -0.26 & -0.24 & 0 & 0 & 0 \\ -0.24 & 0 & -0.26 & 0.82 & 0 & 0 & -0.32 & 0 \\ 0 & -0.23 & -0.24 & 0 & 1.12 & -0.51 & 0 & -0.14 \\ 0 & 0 & 0 & 0 & -0.51 & 0.66 & 0 & -0.15 \\ 0 & 0 & 0 & -0.32 & 0 & 0 & 0.64 & -0.32 \\ 0 & 0 & 0 & 0 & -0.14 & -0.15 & -0.32 & 0.61 \end{bmatrix} \quad (6)$$

For many reasons, it is often advantageous to deal with the normalized Laplacian, defined as

$$\mathbf{L}_N = \mathbf{D}^{-1/2}(\mathbf{D} - \mathbf{W})\mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}.$$

System on a Graph I

- A system on a graph – graph shifted input

$$\mathbf{y} = h_0\mathbf{W}^0\mathbf{x} + h_1\mathbf{W}^1\mathbf{x} + \cdots + h_{M-1}\mathbf{W}^{M-1}\mathbf{x} = \sum_{m=0}^{M-1} h_m\mathbf{W}^m\mathbf{x}, \quad (10)$$

  where, $\mathbf{W}^0 = \mathbf{I}$, while $h_0, h_1, \ldots, h_{M-1}$ are system coefficients.

- The corresponding classic system is a standard FIR filter,

$$y(n) = h_0x(n) + h_1x(n-1) + \cdots + h_{M-1}x(n-M+1). \quad (11)$$

- A system defined using the Laplacian

$$\mathbf{y} = \mathbf{L}^0\mathbf{x} + h_1\mathbf{L}^1\mathbf{x} + \cdots + h_{M-1}\mathbf{L}^{M-1}\mathbf{x} \quad (12)$$

  gives an unbiased estimate of a constant, i.e. if $\mathbf{x} = \mathbf{c}$ then $\mathbf{y} = \mathbf{c}$.

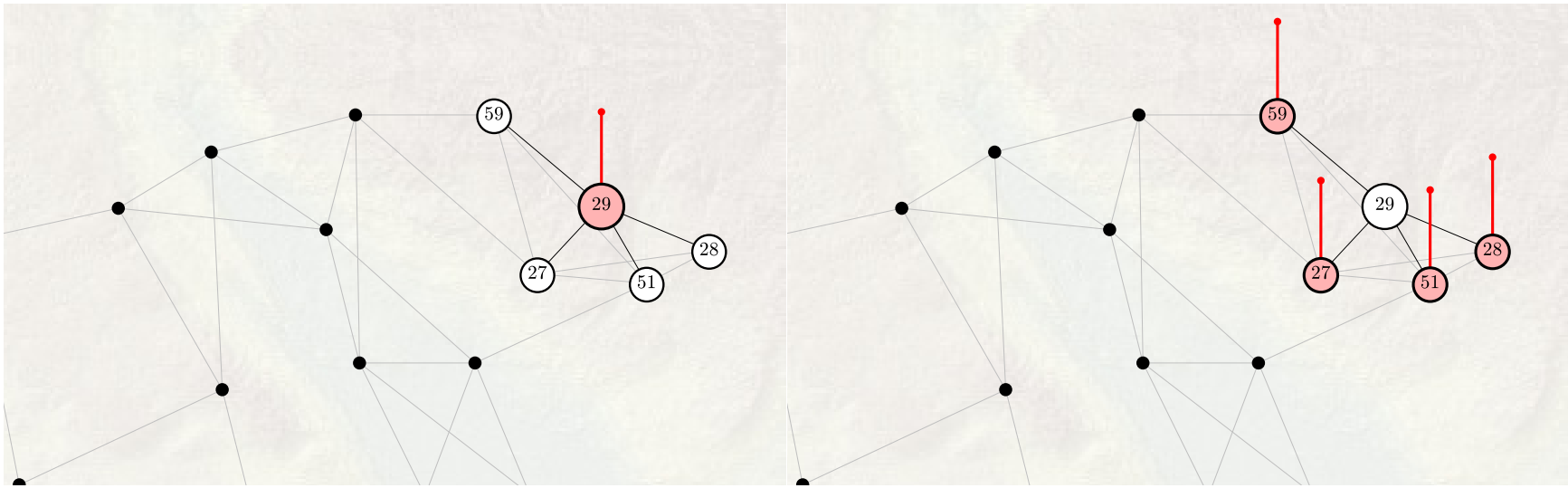
- A simple first order system based on the graph Laplacian can be written as

$$\mathbf{y} = \mathbf{x} + h_1\mathbf{L}\mathbf{x} \quad (13)$$

The Problem with the Graph Shift Operator

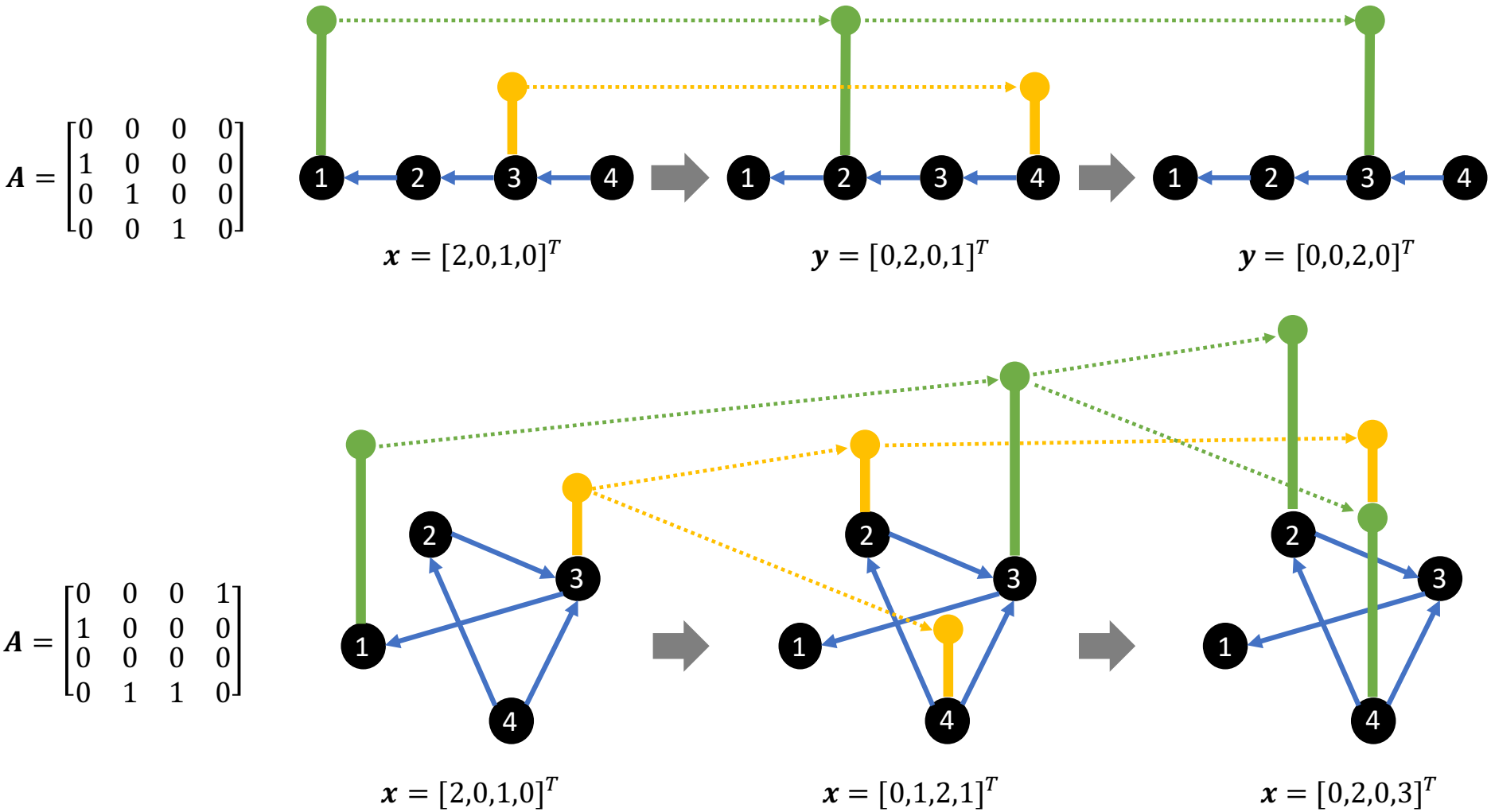
- **The signal shift** on a graph can be viewed as the movement of a signal sample from the considered vertex along all edges connected to this vertex. The signal shift operator can then be compactly defined using the graph adjacency matrix as

$$\mathbf{x}_{shifted} = \mathbf{A}\mathbf{x}.$$



The energy of the shifted signal is not the same as the energy of the original signal (graph shift is not isometric).

Standard Shift vs Graph Shift Operator



A solution: A Class of Doubly Stochastic Shift Operators for Random Graph Signals and their Boundedness, B. Scalzo, D. P. Mandic, et al. arXiv:1908.01596

Vertex Domain Filtering

Physically, the minimum of $\mathbf{x}\mathbf{L}\mathbf{x}^T$ implies the smoothest possible signal and to arrive at this solution we may employ steepest descent.

- The signal value at an iteration p is adjusted in the opposite direction of the gradient, $\partial E_x/\partial \mathbf{x}^T = 2\mathbf{L}\mathbf{x}$
- This yields the iterative procedure

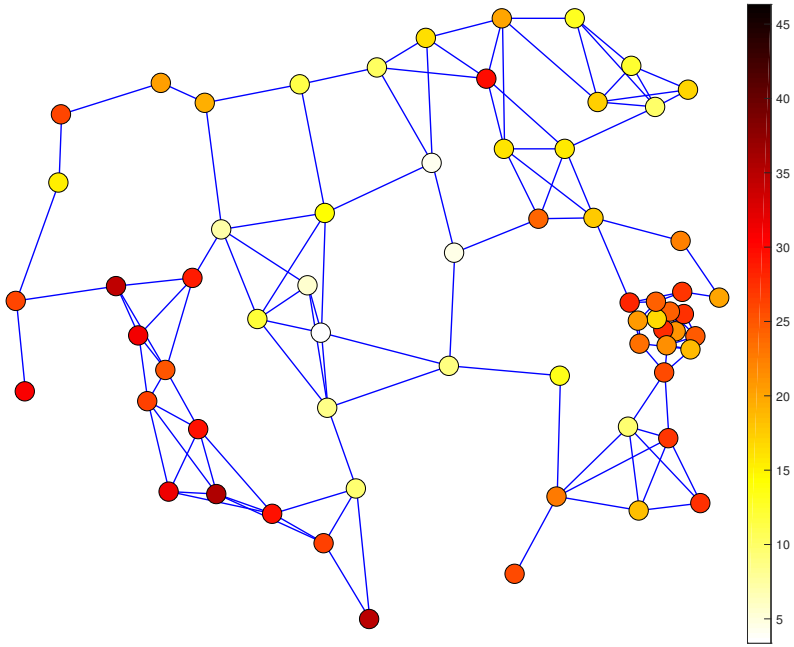
$$\mathbf{x}_{p+1} = \mathbf{x}_p - \alpha \mathbf{L}\mathbf{x}_p = (\mathbf{I} - \alpha \mathbf{L})\mathbf{x}_p .$$

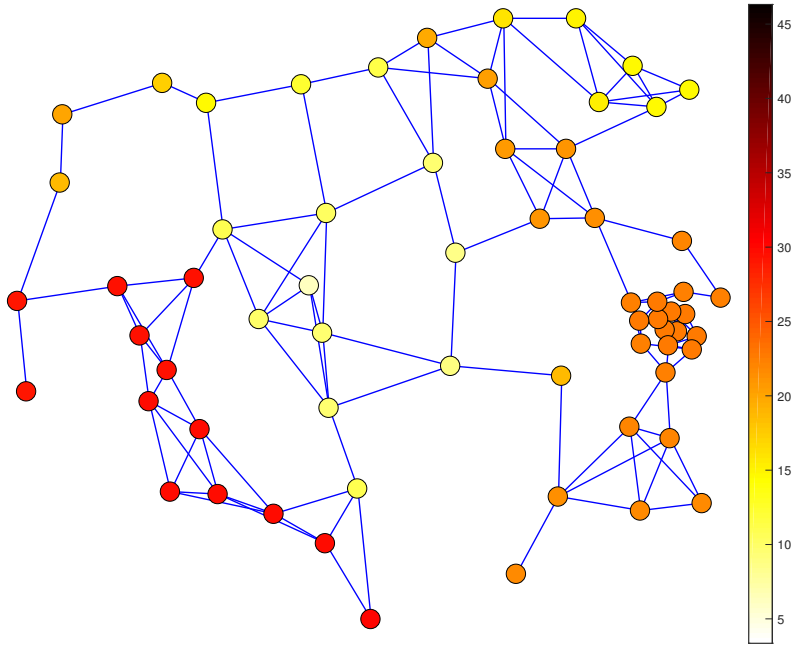
- The signal $\mathbf{x}_{p+1}$ is as an output of the first order system.
- The minimum of the quadratic form $\mathbf{x}\mathbf{L}\mathbf{x}^T$ corresponds to a constant signal. To avoid obtaining only constant steady state, the above iteration process can be used in alternation with

$$\mathbf{x}_{p+2} = (\mathbf{I} + \beta \mathbf{L})\mathbf{x}_{p+1}$$

- This two-step iterative processes is known as **Taubin's $\alpha - \beta$ algorithm.**

Vertex Domain Filtering Results

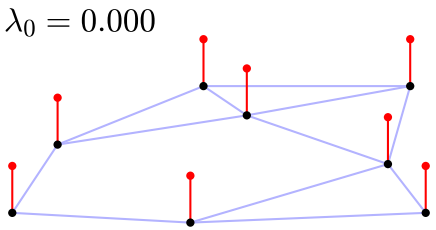


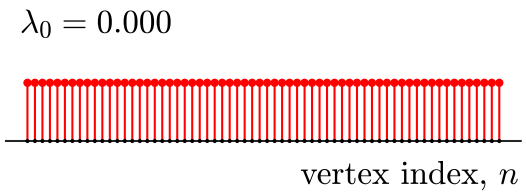


For appropriate values of α and β, this system can give a good and very simple approximation of a graph low-pass filter.

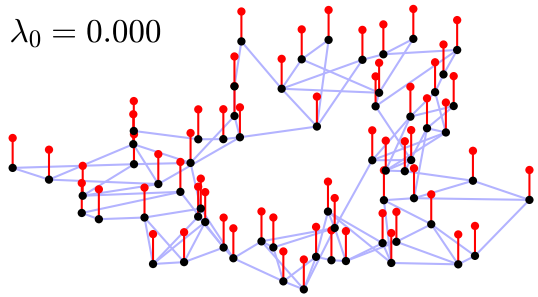
The original noisy signal was filtered using Taubin's algorithm, with $\alpha = 0.2$ and $\beta = 0.1$. After 50 iterations, the signal-to-noise ratio improved from the original $SNR_0 = 14.2$ dB to 26.8 dB.

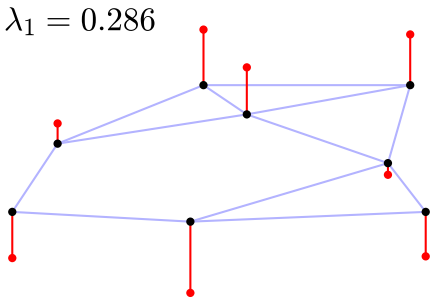
Eigen-properties of Graph Laplacian: An Analogy with Fourier Transform

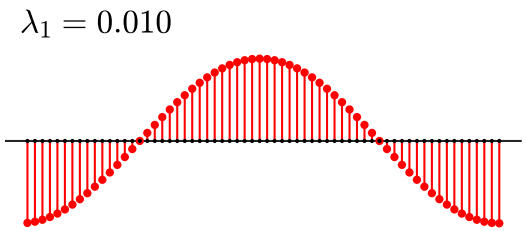




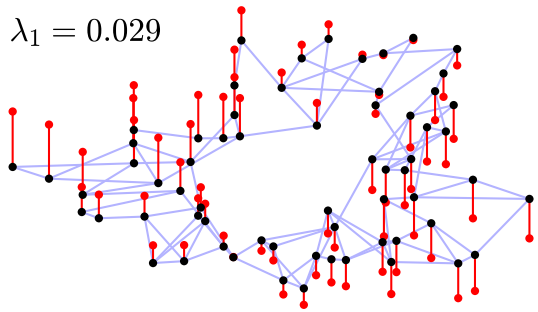
a) constant eigenvector, $u_0(n)$

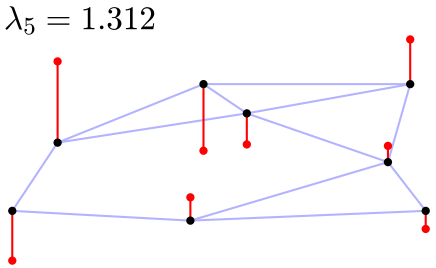


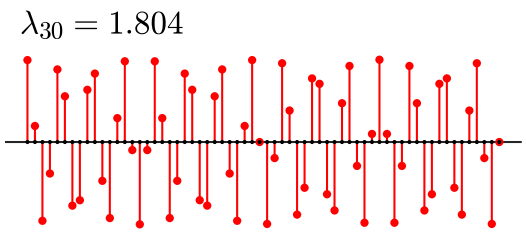




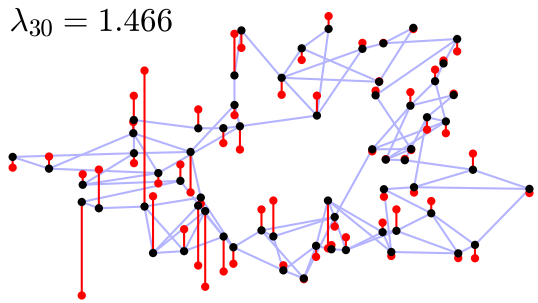
b) slow-vaying eigenvector, $u_1(n)$







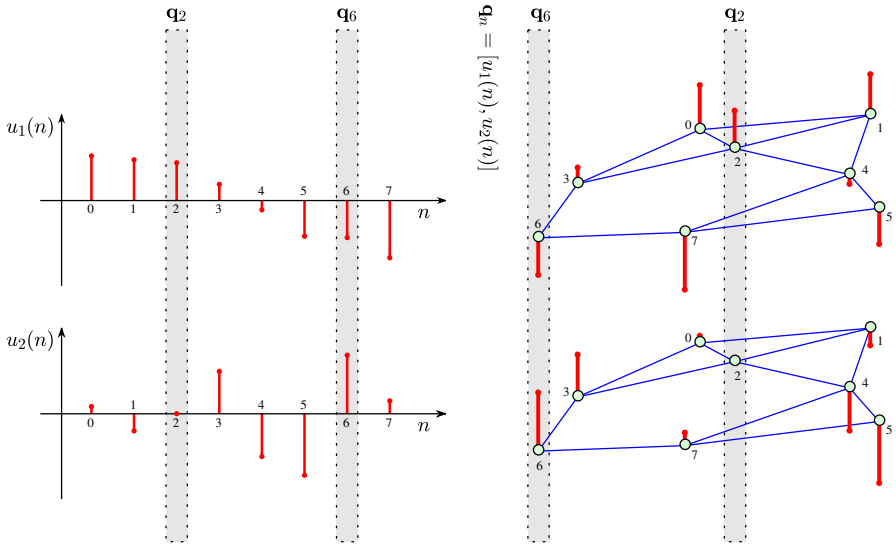
c) fast-varying eigenvector, $u_k(n)$



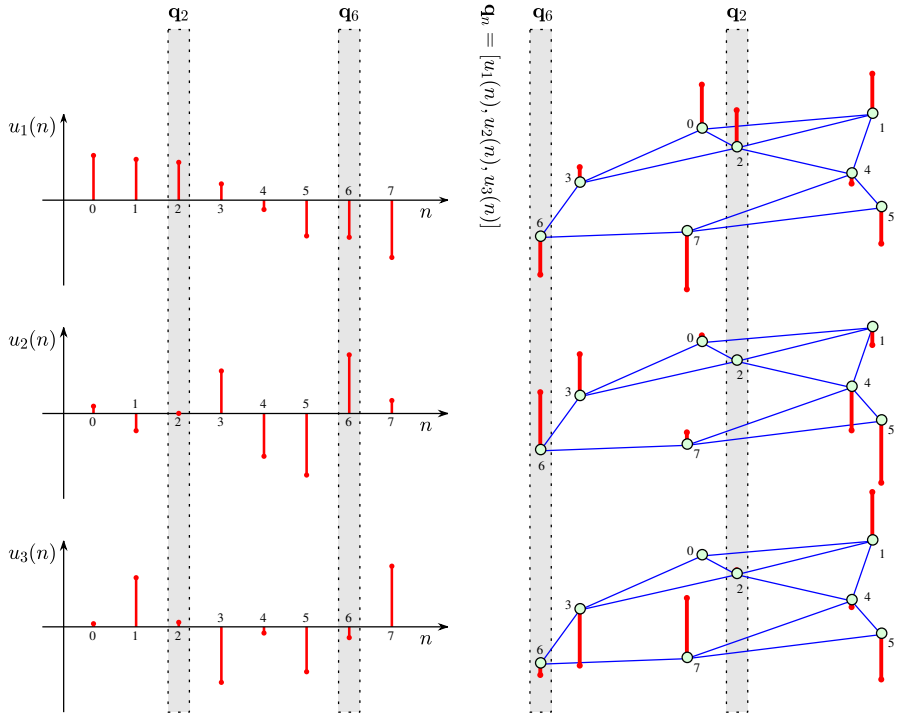
The

Let us remind ourselves of spectral vectors

With spectral vectors we can choose the number of features to represent data and perform dimensionality reduction



Two- dimensional spectral vectors:
$\boldsymbol{q_2} = [u_1(2), u_2(2)]$ and $\boldsymbol{q_6} = [u_1(6), u_2(6)]$



Three-dimensional spectral vectors:
$\boldsymbol{q_2} = [u_1(2), u_2(2), u_3(2)]$ and $\boldsymbol{q_6} = [u_1(6), u_2(6), u_3(6)]$

Dimensionality reduction through spectral vectors is particularly useful in large-dimensional graphs which exhibit lower-dimensional clusters, as illustrated in the next example

Vertex Clustering Based on the Eigenvectors

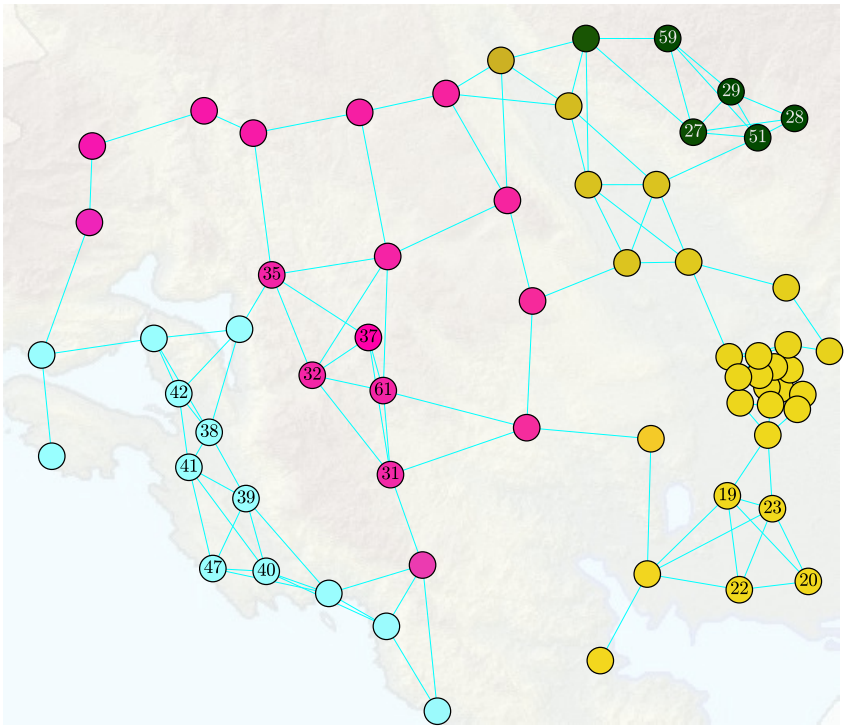
- The Laplacian quadratic form of an eigenvector (its smoothness index) is equal to the corresponding eigenvalue,

$$\mathbf{u}_k^T(\mathbf{L}\mathbf{u}_k) = \mathbf{u}_k^T(\lambda_k \mathbf{u}_k) = \lambda_k .$$

- The eigenvector corresponding to $\lambda_1 = 0$ is a constant (maximally smooth for any vertex ordering).
- Spectral similarity of vertices is defined using eigenvectors, if the eigenvector elements $u_k(n)$, $k = 1, 2, \ldots, P$ are assigned to the vertex n. If $\mathbf{u}_1$ is omitted, then a $(P-1)$-dimensional spectral vector becomes $\mathbf{q}_n = [u_2(n), \ldots, u_P(n)]^T$.
- The spectral similarity between vertices n and m is defined as the two-norm $\|\mathbf{q}_n - \mathbf{q}_m\|_2$.

Vertex clustering: The temperature graph

- Keep the original vertex positions and color them according to the spectral vectors $\mathbf{q}_n$.
- Coloring is performed using the eigenvector elements $u_2(n)$, $u_3(n)$, and $u_4(n)$ as color coordinates for the vertex n.



- Graph segmentation, by grouping vertices with similar colors.
- The graph segmentation is a signal-independent operation. It roughly indicates the data connectivity between sensor data values on this graph, and suggests that the data processing will predominantly be localized within these regions.

Image clustering: Very straightforward via graphs

Fig.1. Original image

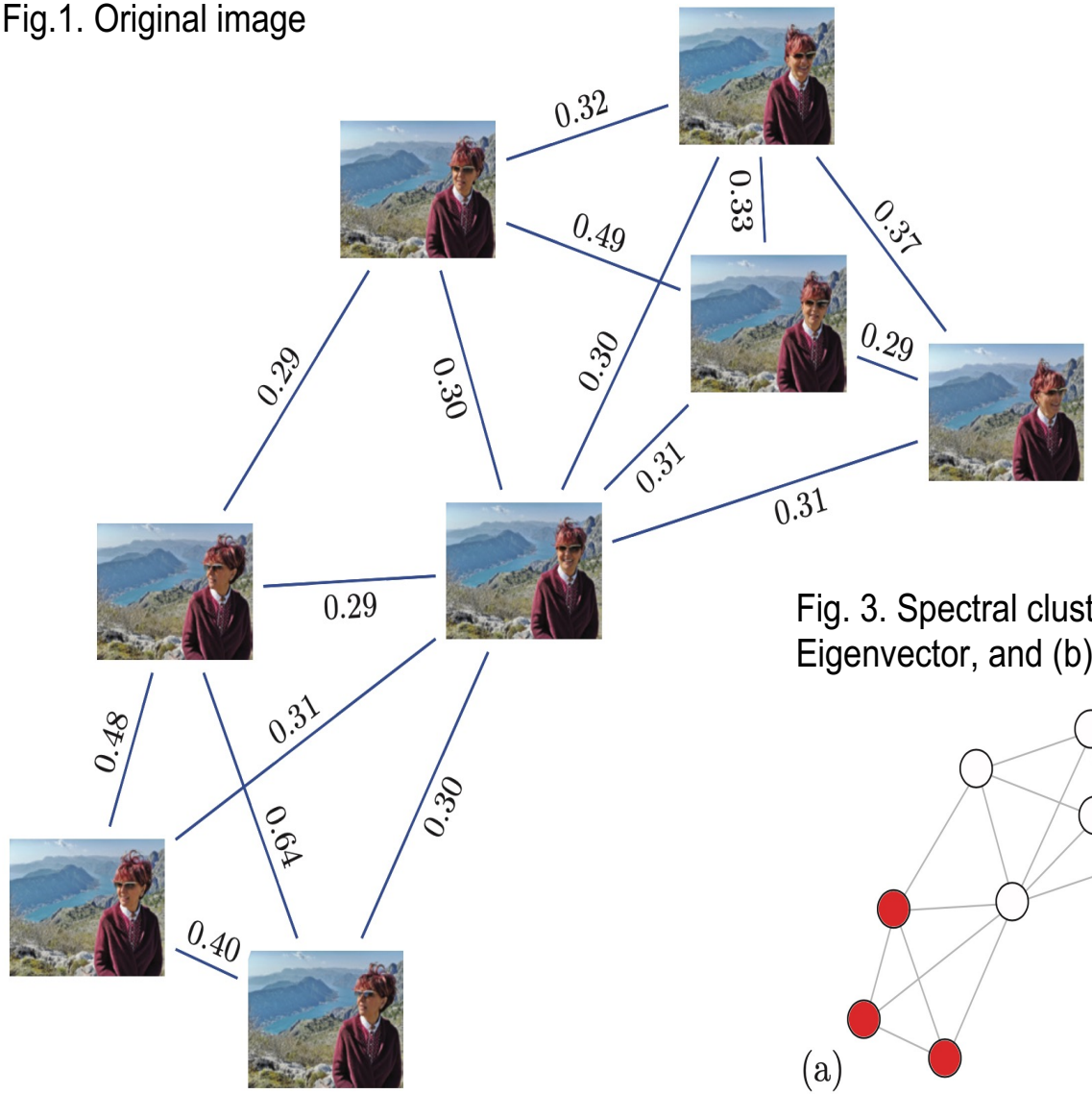


Fig. 2. Graph based on structural similarity

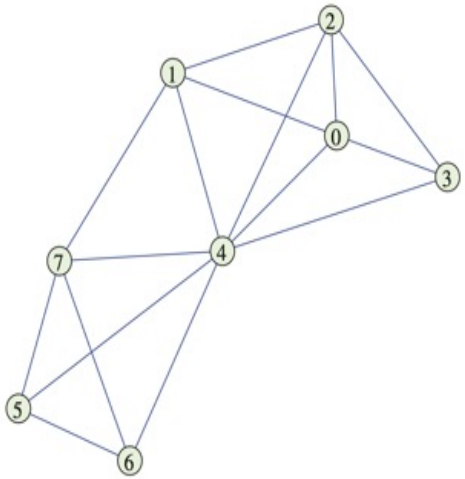


Fig. 3. Spectral clustering based on (a) the Fiedler (smoothest) Eigenvector, and (b) the two smoothest Eigenvectors.

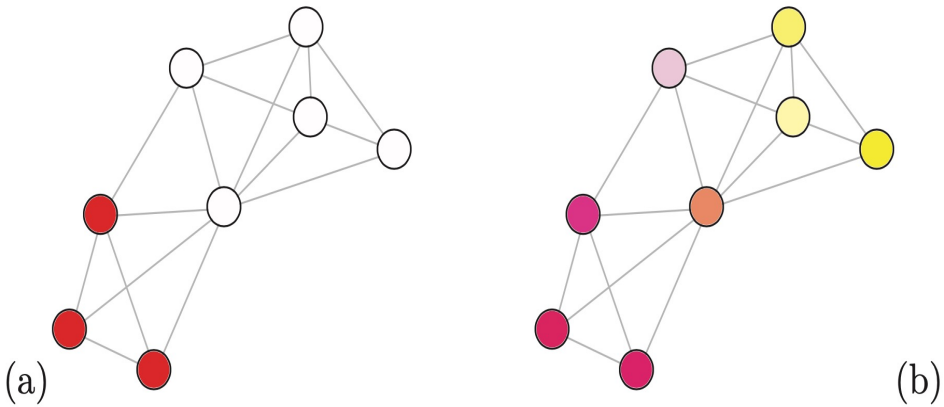
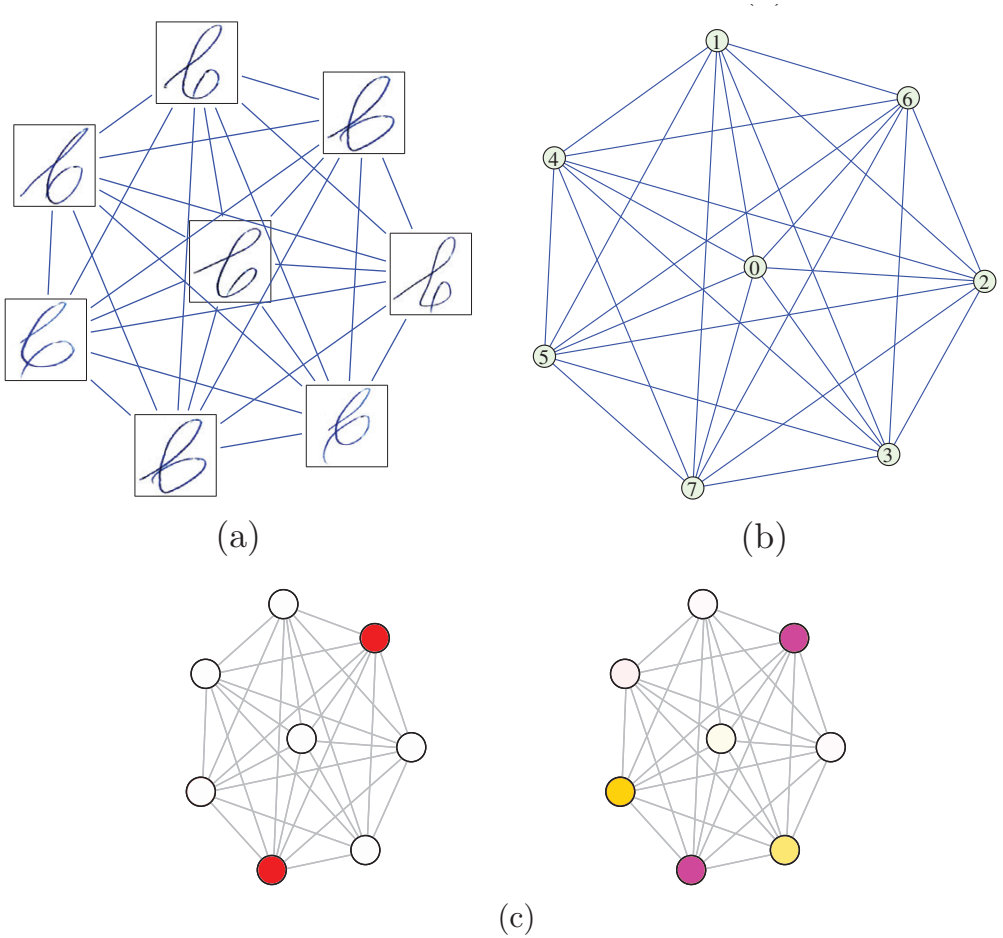
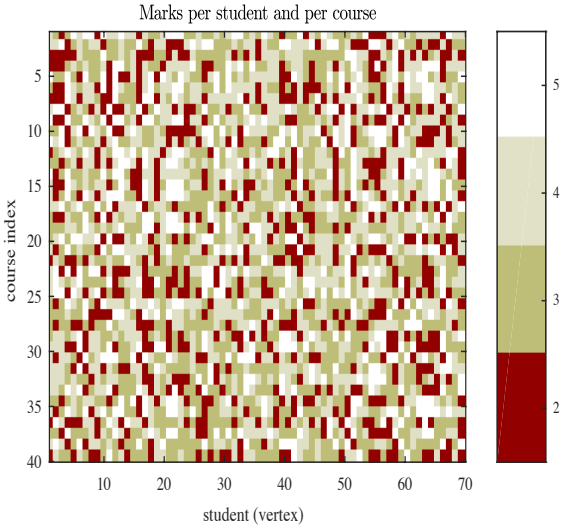


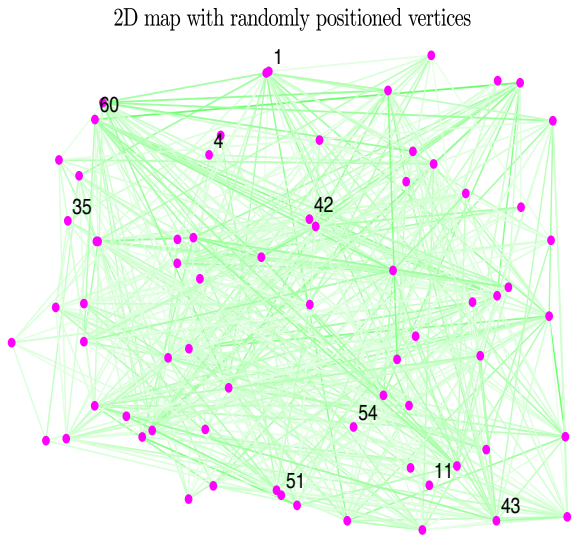
Image Classification: Optical Character Recognition

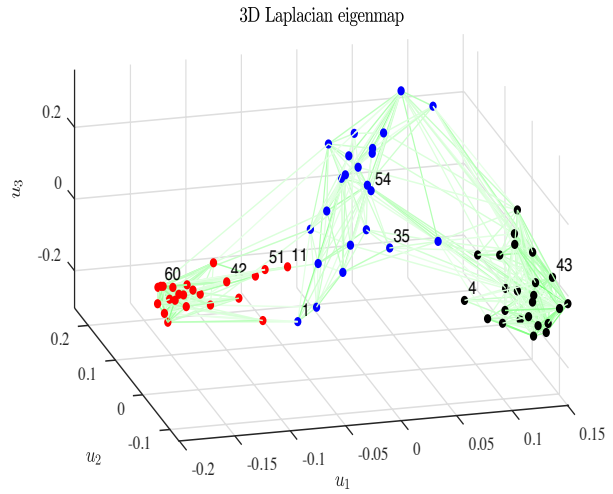


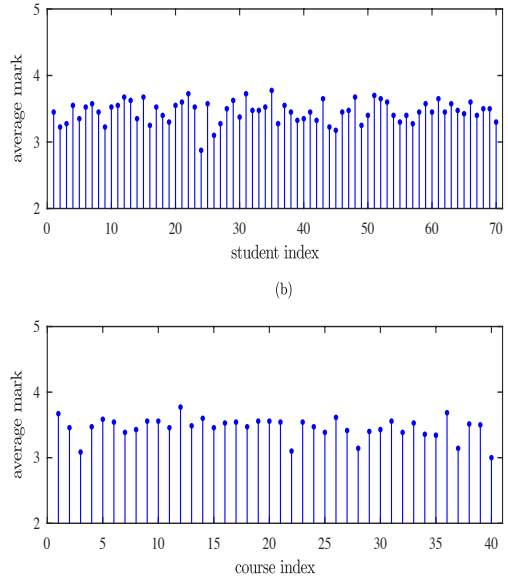
Graph representation of a set of hand-written images of the letter “b”. The images serve as vertices, while the weight matrix for the edges is defined through the structural similarity index metric (SSIM) between the images. The vertices are colored in (c) using first the smoothest (Fiedler) eigenvector, **u**1 (left), and then the two smoothest eigenvectors, **u**1 and **u**2, of the generalized eigenvectors of the Laplacian (right).

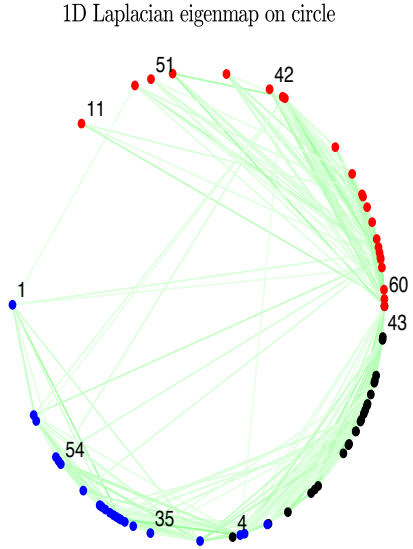
Graphs for Dimensionality Reduction

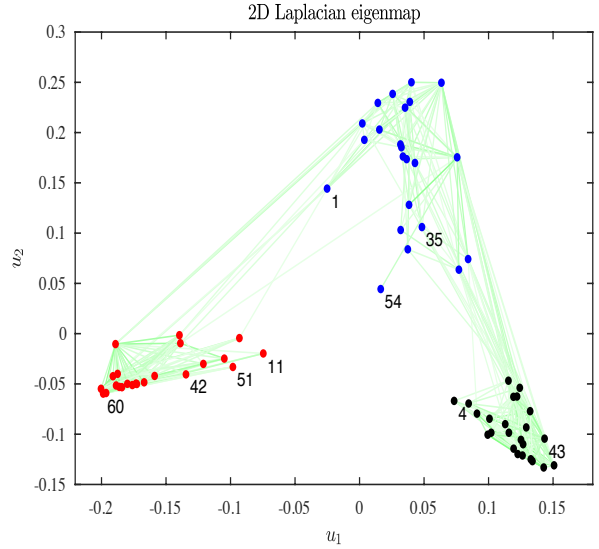












Graph Topology Learning from Data

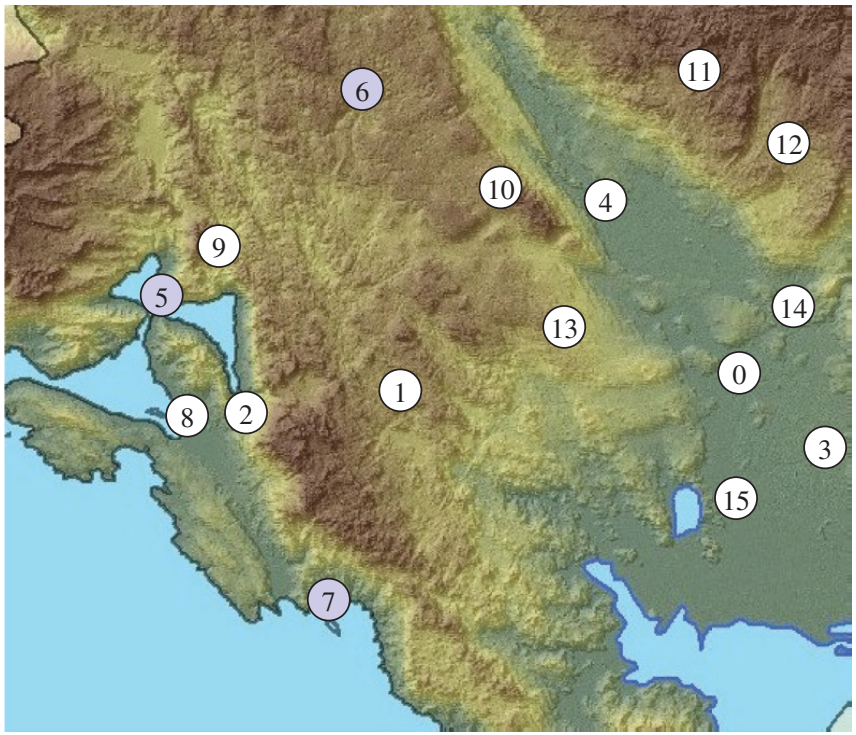


Fig. 1. Sensing locations in a geographic region near the Adriatic sea.

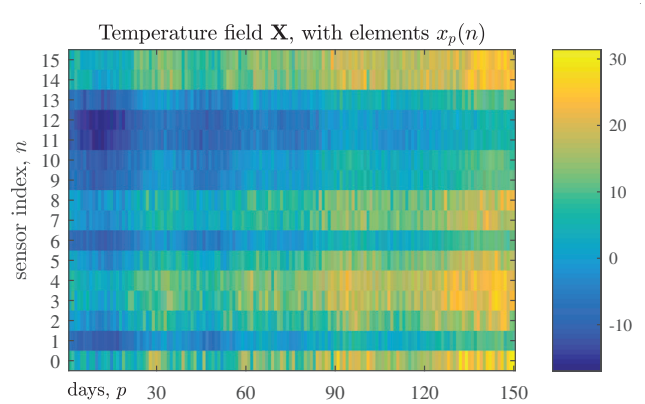


Fig. 2. Temperatures measured at N = 16 sensing locations over P = 150 days

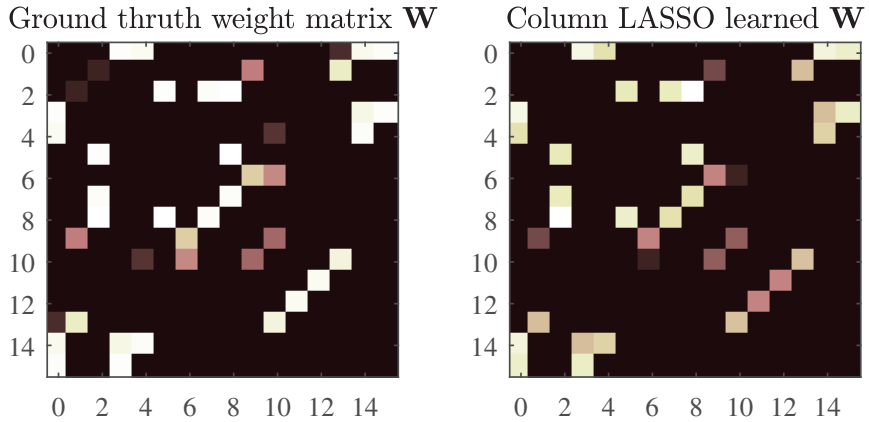
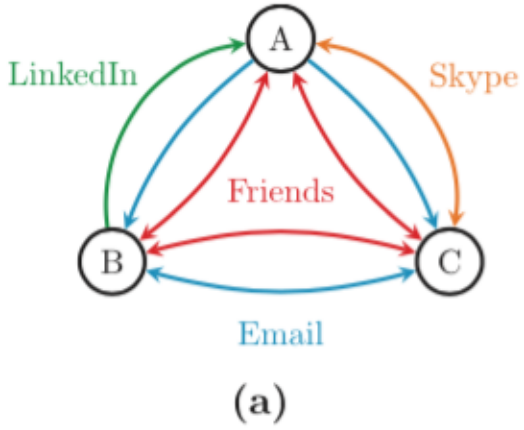
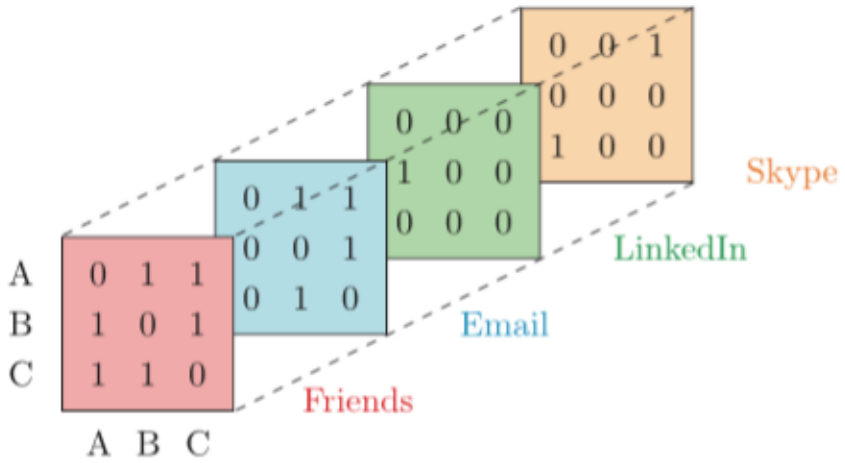


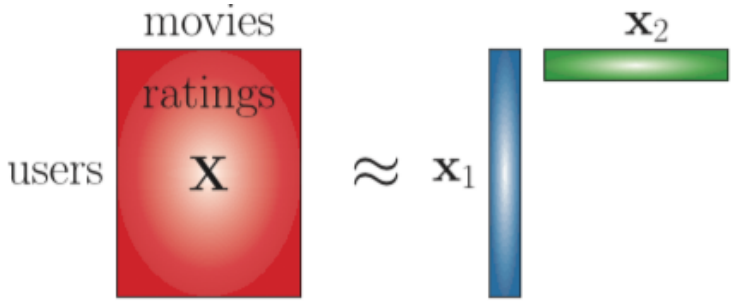
Fig. 3. Ground truth weight matrix, **W**, obtained through geographic properties of the sensing locations (left), and the learned weight matrix, **W**, estimated using the LASSO approach from data.

Some data that are naturally both tensor and graph



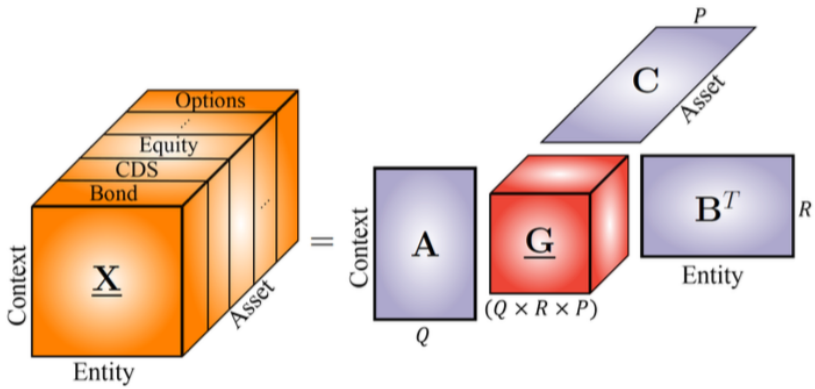
(a)

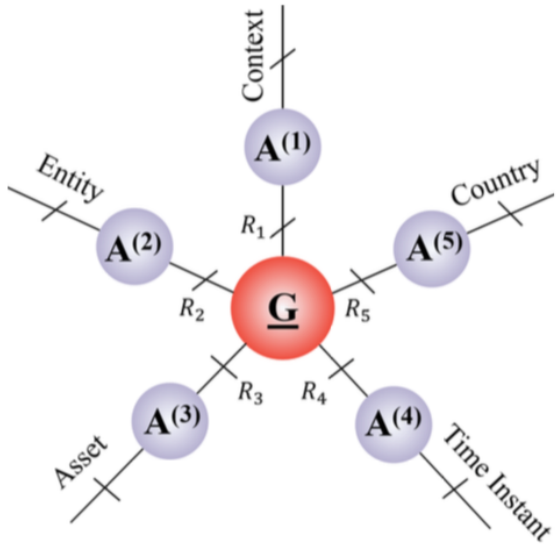




Also, a social network can be seen as a tensor, or even Netflix movie

Some tensor data: Financial system



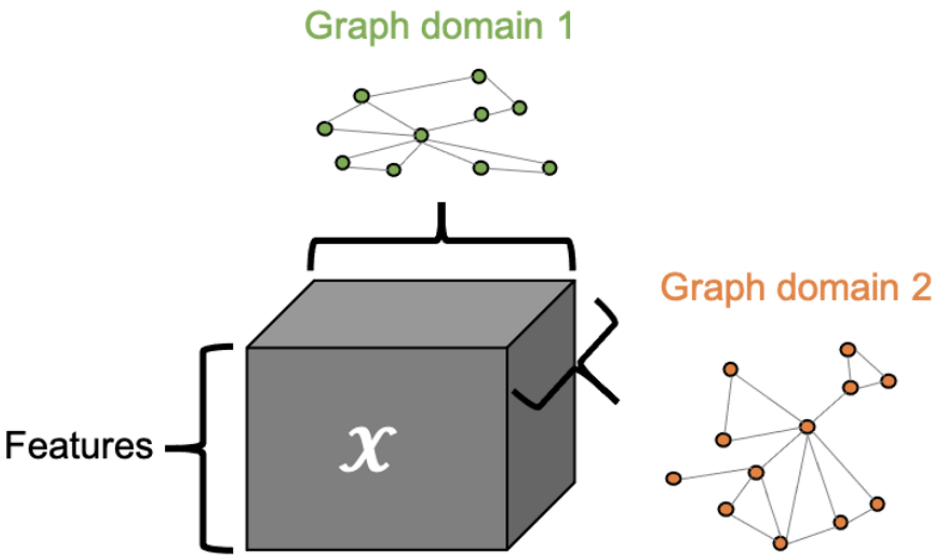


Financial system represented as a tensor of

$context \times asset \times entity \times country \times \cdots$

Each physical mode of this tensor also has a graph structure, which can be a graph on a regular or irregular domain

Graphs and tensors in finance

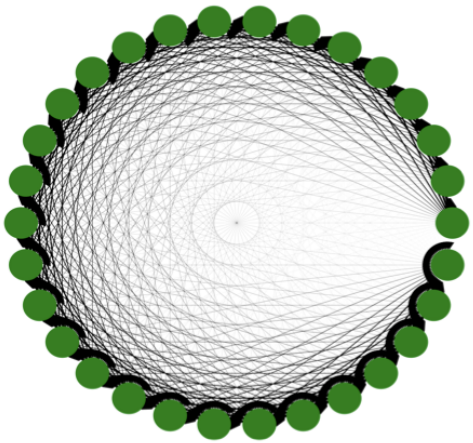


We can establish a graph structure for e.g. portfolios, currency exchange, ...

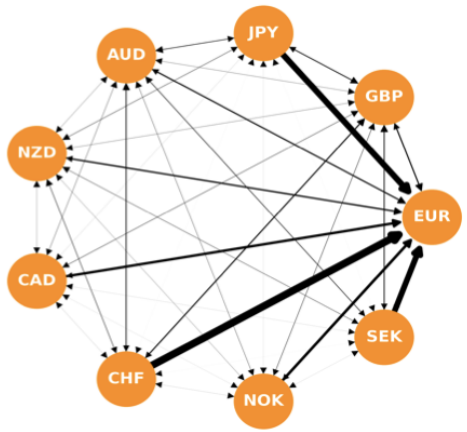
Left: A time dimension graph

Right: A graph representation of foreign exchange, where the edges are derived from the carry factor (related to interest rate in a particular country)

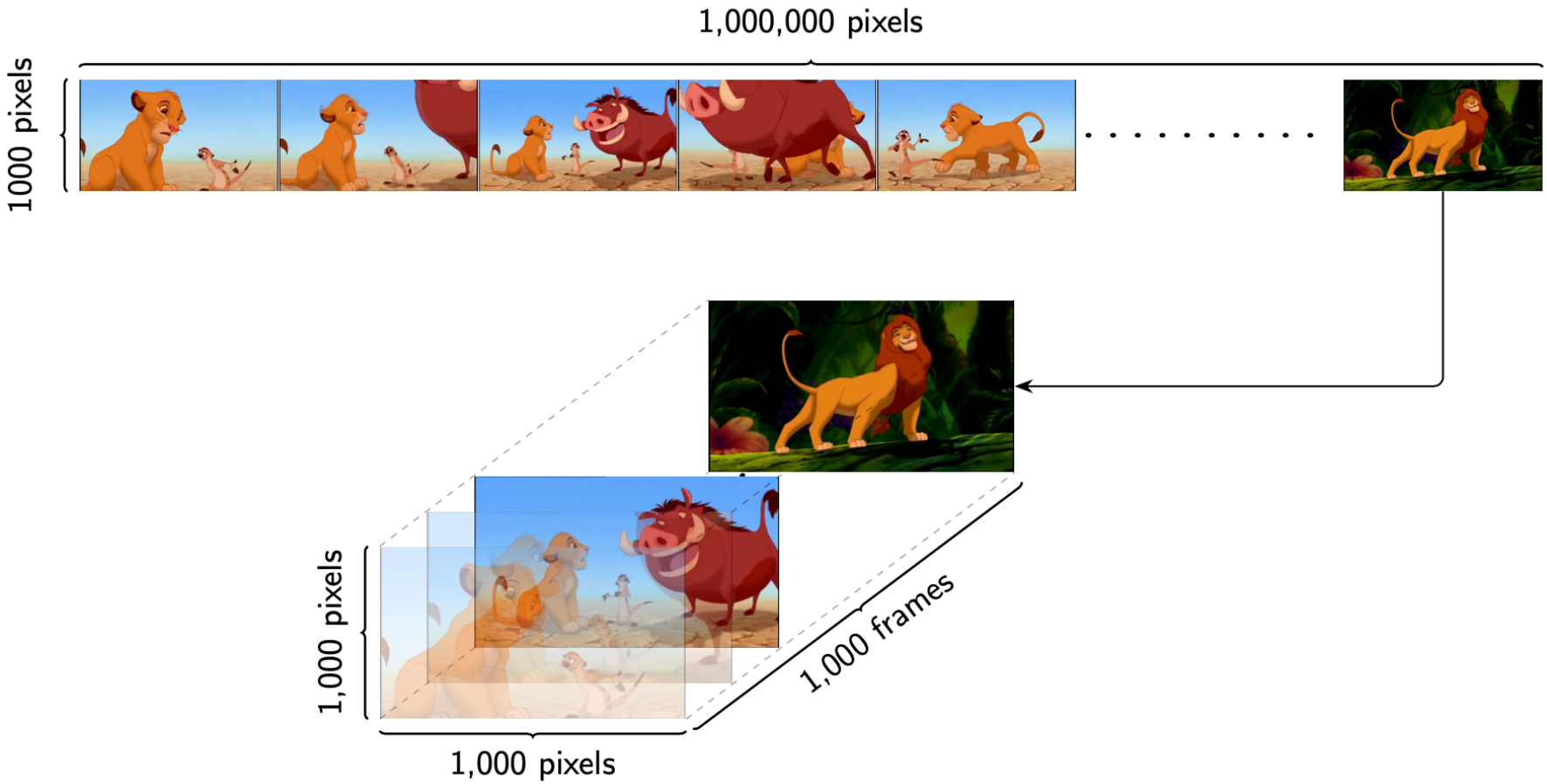
Directed time graph



Carry graph

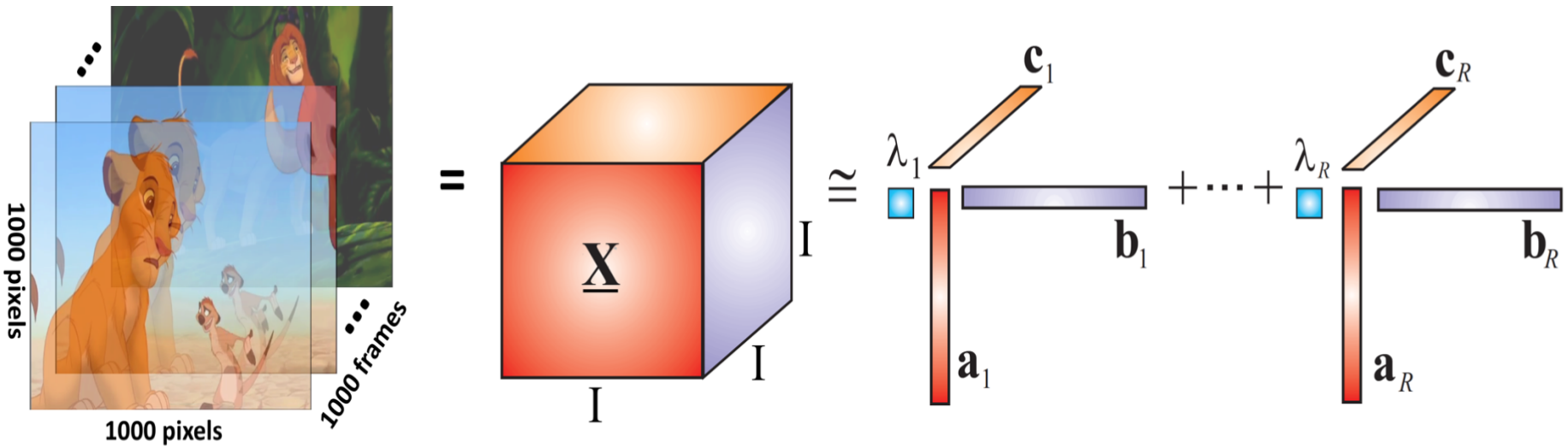


Why tensors?



- A simple re-arrangement into a cube transforms the $1{,}000 \times 1{,}000{,}000$ matrix of frames into a 3-way tensor of size $1{,}000 \times 1{,}000 \times 1{,}000$

Canonical Polyadic Decomposition (CPD)



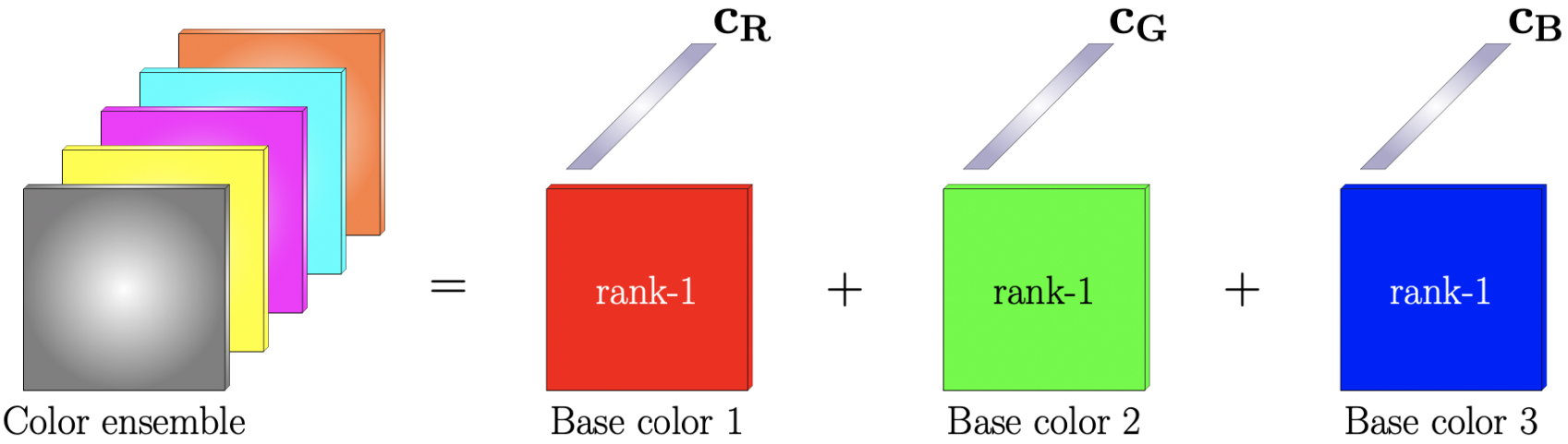
After tensorizing the video clip, tensor order $N = 3$, the dimension in every mode $I = 1000$, and the tensor rank is R. Typically $R \lll I$.

with $\texttt{length}(\mathbf{a}_i)$=1000, $\texttt{length}(\mathbf{b}_i)$=1000, $\texttt{length}(\mathbf{c}_i)$=1000, $\quad i = 1, 2, \ldots, R$

- **Raw data format** $\looparrowright I^N = 1000 \times 1000 \times 1000 = 10^9$ pixels $= 1$ Giga-pixel
- **In the CPD format**, this becomes $N \times I \times R = 3 \times 1000 \times 10 = 30,000$ pixels (for R=10), that is, compression of almost 5 orders of magnitude
- In scientific computing, if we sample a cube at $I = 10,000$ points, then $I^N = 10^{12}$ raw samples become $N \times I \times R = 3 \times 10^5$ samples in CPD

  For N=4, I=10^4, R=10, the $I^N = 10^{16}$ raw samples $\rightsquigarrow 4 \times 10^5$ samples in CPD

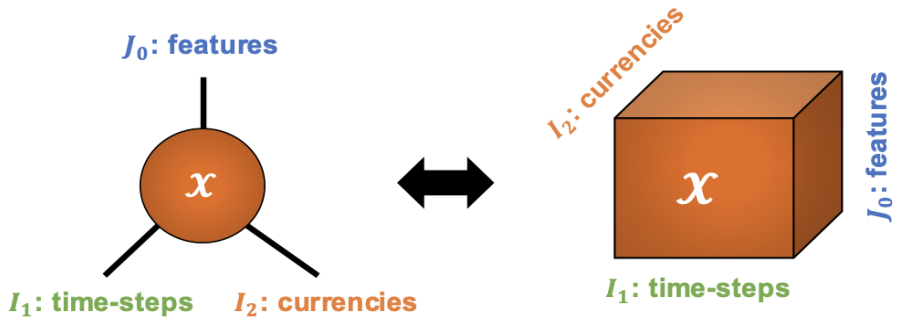
Tensor Rank: An intuitive example

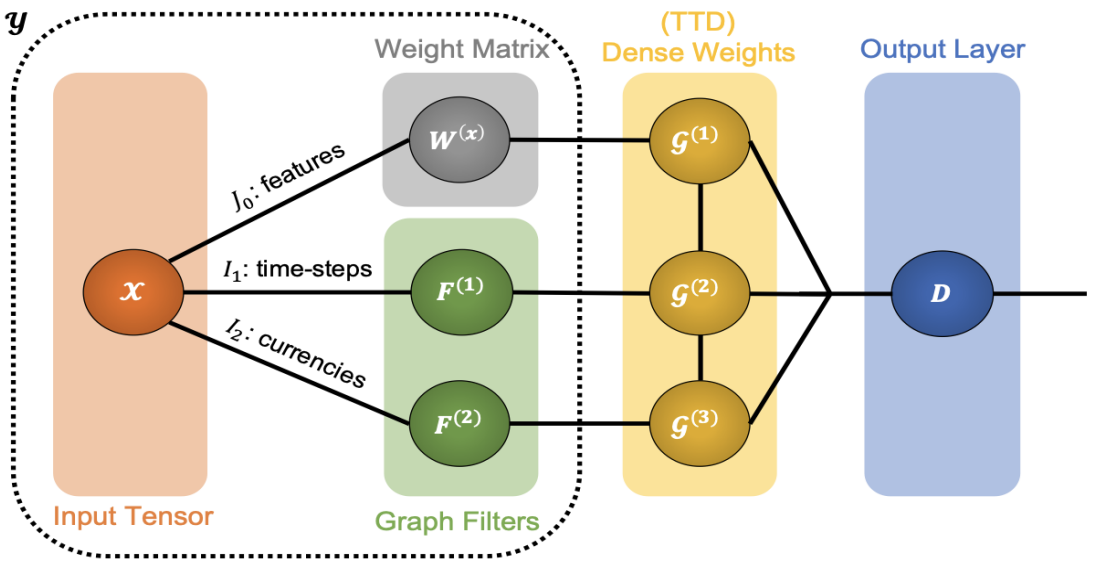


- All colors are just combination of three base colors: red, green and blue $\leadsto$ rank $= 3$
- Vectors $\mathbf{c}_R, \mathbf{c}_G, \mathbf{c}_B$ represent intensity, i.e. each value characterises how much of the base color there is in the corresponding slice

$$\mathbf{c}_R = \begin{bmatrix} 128 \\ 256 \\ 256 \\ 0 \\ 256 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} \quad \mathbf{c}_G = \begin{bmatrix} 128 \\ 256 \\ 0 \\ 256 \\ 128 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 1 \\ 0 \\ 1 \\ 0.5 \end{bmatrix} \quad \mathbf{c}_B = \begin{bmatrix} 128 \\ 0 \\ 256 \\ 256 \\ 32 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0 \\ 1 \\ 1 \\ 0.125 \end{bmatrix}$$

Putting everting together: From a Forex tensor to a Forex graph tensor network



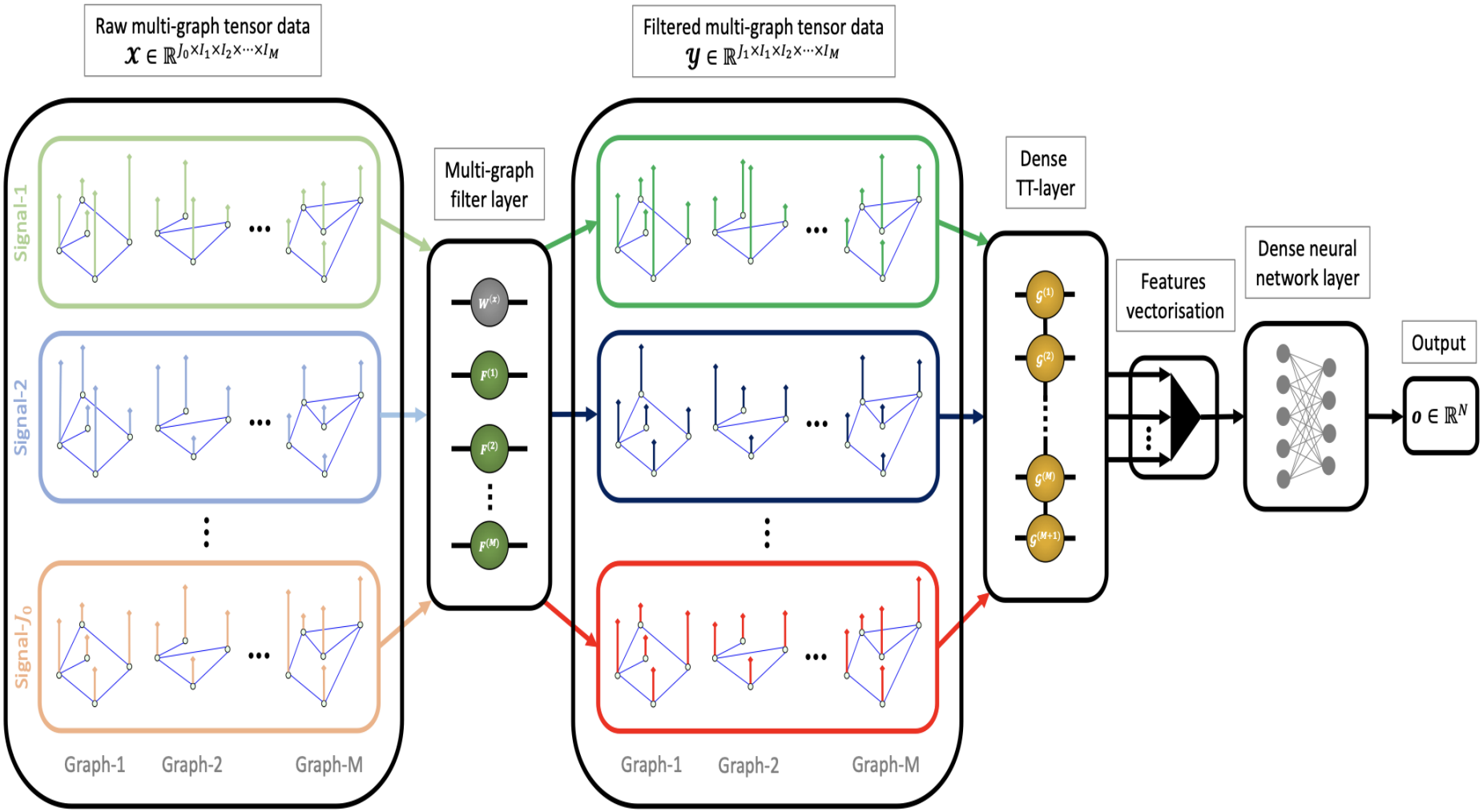


A Multi-Graph Tensor Network

Features:
- Low price
- High Price
- Log return
- Peak to peak return
- Trading volume
- Start price
- End price, ...

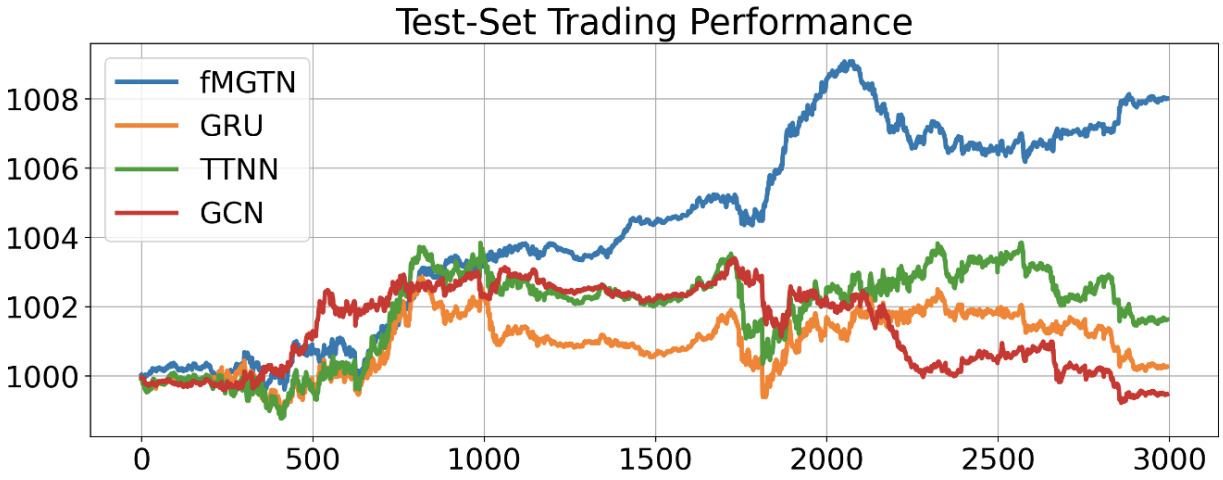
Features do not admit graph representation and are instead combined using standard (NN) weight matrix

The Multi-Graph Tensor Network (MGTN) concept



- Raw tensor data may have multiple graph repres. The graph filtering operation exhibits locality maintains keeps structure
- The feature mode (in grey) does not undergo graph filtering but standard (perhaps NN) filtering through weight matrix **W**
- The filtered graph data are processed first through a NN which is shown in a TT format, then through standard NN

The MGTN performance in ForEx, over 9 currencies, 4 features, and 30 time steps



Out-of-sample trading performance, averaged over 5 European currencies.

Vertical axis: Investment growth of an Initial portfolio value of $1000.

MGTN required much lower parameter complexity, using:

- 10% of trainable parameters compared with GCN
- 20 % of trainable parameters compared with GRU

Agent	TR (%)	SR	MD (%)	HR(%)	NP
fMGTN	**0.8018**	**0.0445**	**0.2893**	**52.8056**	**531**
GRU	0.0260	0.0012	0.3477	50.4008	3107
TTNN	0.1628	0.0064	0.3493	50.6346	451
GCN	-0.0538	-0.0032	0.4180	50.2338	5891

Performance comparison for the considered agents for the task of algorithmic trading of currencies. The evaluation metrics are: Total Return (TR), Sharpe ratio (SR), Max Drawdown (MD), Hit Rate (HR), and Number of Parameters (NP).

Portfolio optimisation through graph cuts

The optimal portfolio holdings then become

$$\mathbf{w} = \frac{\mathbf{\Sigma}^{-1}\mathbf{1}}{\mathbf{1}^T\mathbf{\Sigma}^{-1}\mathbf{1}} \tag{4}$$

Instability issues remain prominent, as the matrix inversion of $\mathbf{\Sigma}$ required in (4) may lead to significant errors for ill-conditioned (singular) matrices.

Problem: The more collinear portfolio assets the more unstable the above solution

Solution: The obvious need for greater diversification, e.g. through graph cuts

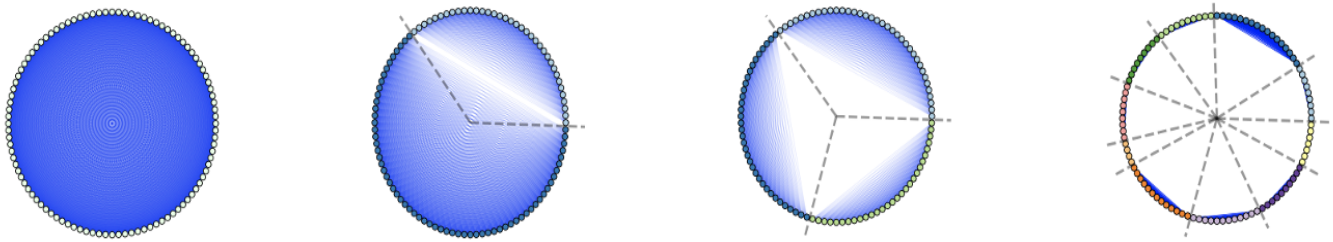
A universe of N assets can be represented as a set of vertices on a *market graph*, whereby the edge weight, W_{mn}, between vertices m and n is defined as the absolute correlation coefficient, $|\rho_{mn}|$, of their respective returns of assets m and n, that is

$$W_{mn} = \frac{|\sigma_{mn}|}{\sqrt{\sigma_{mm}\sigma_{nn}}} = |\rho_{mn}| \tag{8}$$

where $\sigma_{mn} = \mathrm{cov}\{r_m(t), r_n(t)\}$ is the covariance of returns between the assets m and n.

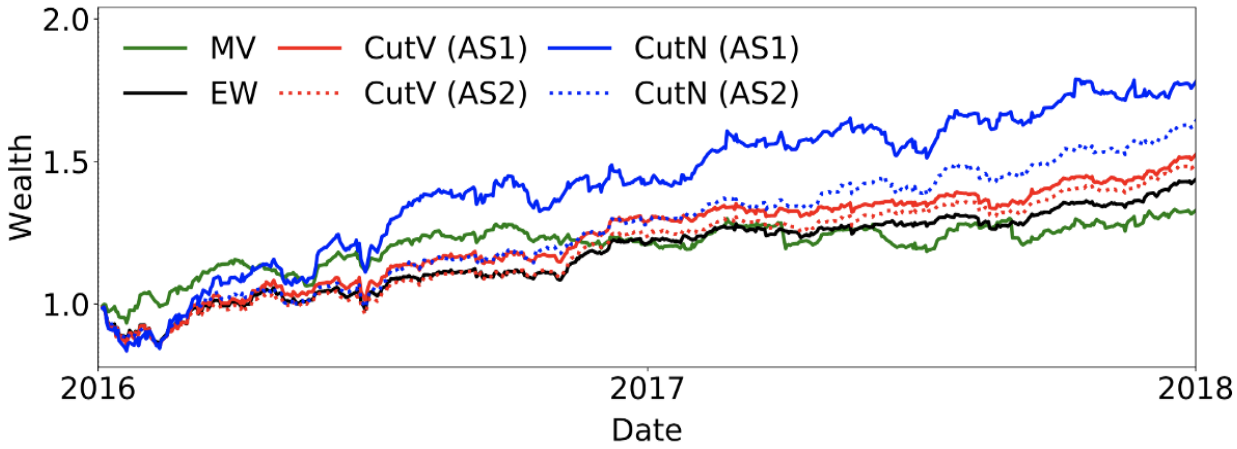
In this way, we have $W_{mn} = 0$ if the assets m and n are statistically independent (not connected), and $W_{mn} > 0$ if they are statistically dependent (connected on a graph).

Portfolio optimisation through graph cuts



(a) (b) (c) (d)

Figure: Visualisation of the 100-vertex market graph connectivity and its partitions into disjoint sub-graphs (separated by dashed grey lines). The edges (blue lines) were calculated based on the correlation between assets. (a) Fully connected market graph with 5050 edges. (b) Partitioned graph after $K = 1$ portfolio cuts (CutV), with 2746 edges. (c) Partitioned graph after $K = 2$ portfolio cuts (CutV), with 1731 edges. (d) Partitioned graph after $K = 10$ portfolio cuts (CutV), with 575 edges. Notice that the number of edges required to model the market graph is significantly reduced with each subsequent portfolio cut, since $\sum_{i=1}^{K+1} \frac{1}{2}(N_i^2 + N_i) < \frac{1}{2}(N^2 + N)$, $\forall K > 0$.



(a) Evolution of wealth for both the traditional (EW and MV) and graph-theoretic asset allocation strategies, based on ($K = 10$) portfolio cuts.

AS1: A kind of ratio cut (the subgraphs simultaneously as large as possible)

$$CutN(\mathcal{V}_1, \mathcal{V}_2) = \frac{\mathbf{x}^T \mathbf{L} \mathbf{x}}{\mathbf{x}^T \mathbf{x}}$$

AS2: A kind of volume normalized minimum cut (similar total degrees of nodes in subgraphs)

$$CutV(\mathcal{V}_1, \mathcal{V}_2) = \frac{\mathbf{x}^T \mathbf{L} \mathbf{x}}{\mathbf{x}^T \mathbf{D} \mathbf{x}}$$

Graphs for Transportation Networks

- The rapid development of world's economies has been followed by an increasing proportion of population moving to cities
- World's urban traffic congestion has become an overwhelming issue
- Underground traffic networks frequently experience signal failures and train derailments
- The economic costs of such transport delays to central London business are estimated to be £1.2 billion per year
- **Appropriate and physically meaningful tools to understand, quantify, and plan for the resilience of transportation networks to disruptions are therefore a pre-requisite for the planning and daily running of public transport**

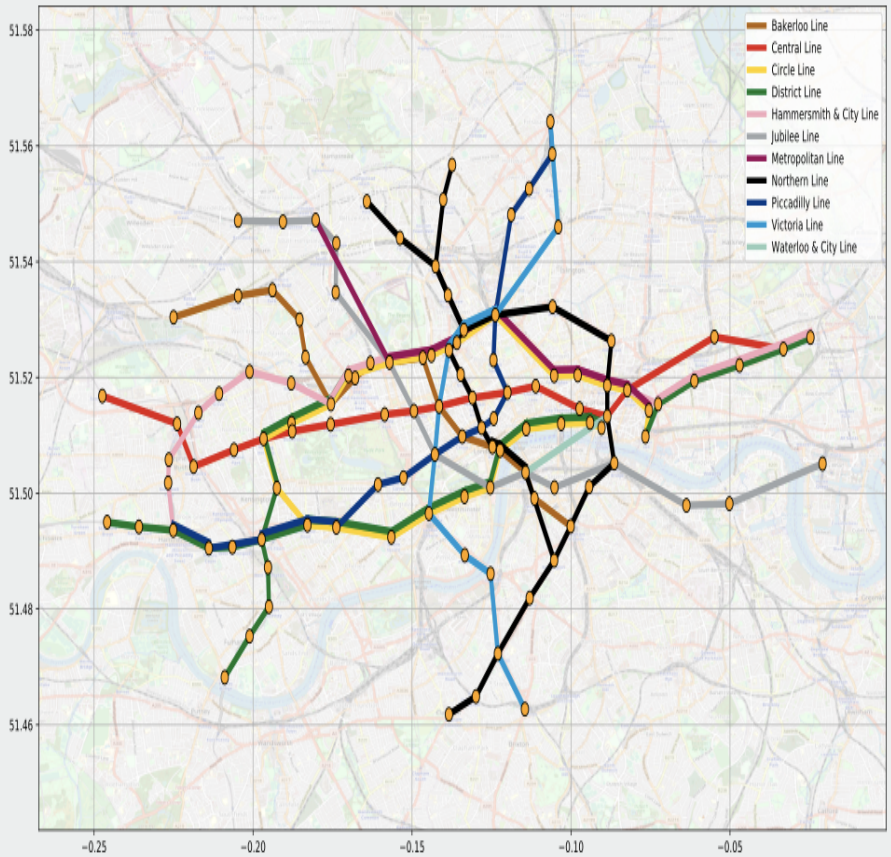


Fig. 1. *Graph representation of the London underground network in Zones 1 and 2. The circles denote the vertices (stations) and the lines between the circles designate the underground lines.*

Passenger flow as a diffusion process

- We can model data on graph via Fick's law of diffusion [3]:
  - $q = -k\nabla\phi$
- Where:
  - q: flux (amount per unit area)
  - k: coefficient of diffusivity
  - ϕ: concentration
- Given:
  - Diagonal degree matrix: $\boldsymbol{D} \in \mathbb{R}^{N\times N}$, where $D_{ii} = \sum_j A_{ij}$ and 0 otherwise
  - Graph Laplacian matrix: $\boldsymbol{L} \in \mathbb{R}^{N\times N}$, defined as $\mathbf{L} = (\boldsymbol{D} - \boldsymbol{A})$
- Diffusion on graph can be modelled as:

$$\mathbf{q} = -\mathrm{k}\mathbf{L}\boldsymbol{\phi}$$

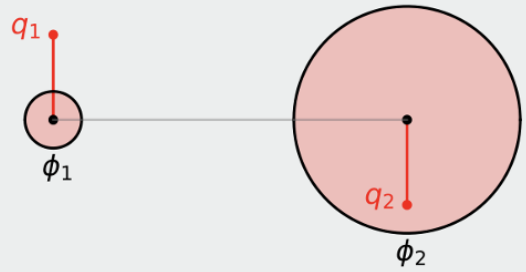


Fig. 5. *A graph representation of the London underground network through Fick's law of diffusion. Consider a simplified path graph with two stations surrounded by the respective populations, ϕ_1 and ϕ_2 (proportional to the circle area), which exhibit the corresponding net fluxes, q_1 and q_2. Stations surrounded by large population (residential areas) exhibit a net in-flow of passengers, while stations surrounded by low population (business districts) experience net out-flow of passengers. The overall net flow (in-flow and out-flow) of passengers across the entire network sums up to zero.*

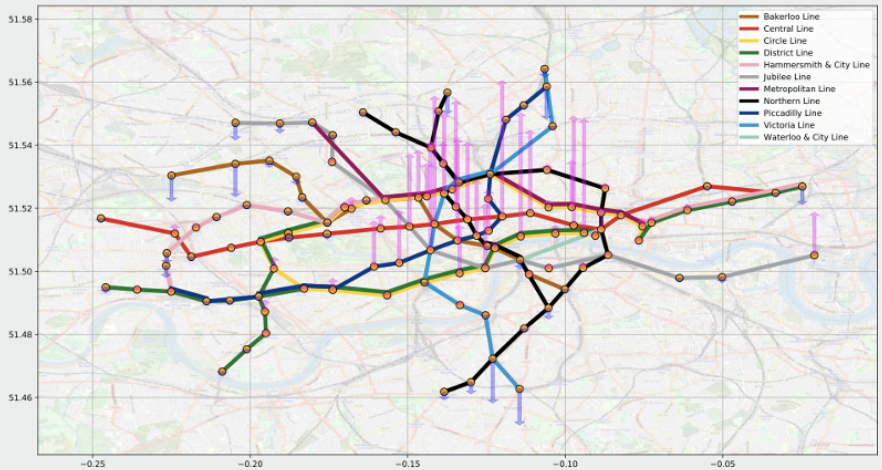


Fig. 6. *Net passenger out-flow during the morning rush hour. The magenta bars denote a net out-flow of passengers while the blue bars designate a net in-flow of passengers. Stations located within business districts exhibit the greatest net out-flow of passengers, while stations located in residential areas, toward the outskirts, exhibit the largest net in-flow of passengers.*

Vulnerability of Underground Stations

Betweenness centrality [1] metric as a measure of the vulnerability

It measures the extent to which a given vertex lies in between pairs or groups of other vertices of the graph:

$$B_n = \sum_{k,m \in \mathcal{V}} \frac{\sigma(k,m|n)}{\sigma(k,m)}$$

- $\sigma(k,m)$ denotes the number of shortest paths between k and m
- $\sigma(k,m|n)$ the number of those paths passing through a vertex n

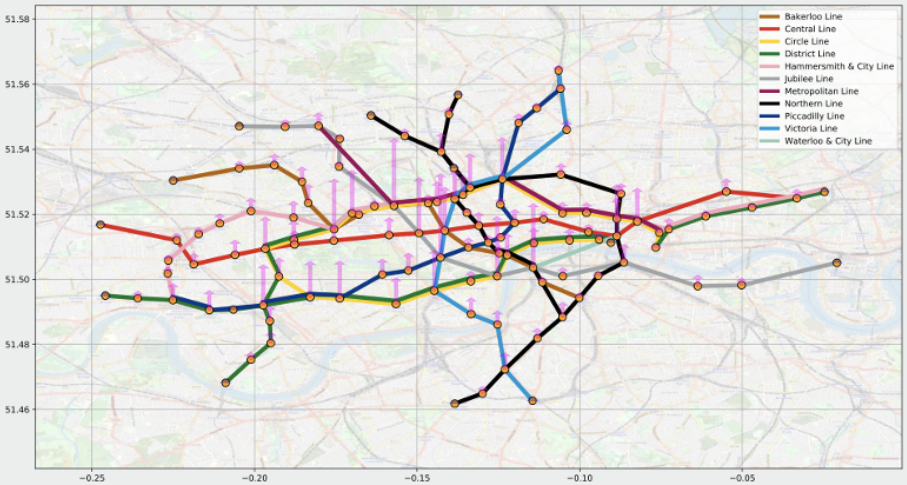


Fig. 2. *Betweenness centrality, designated by magenta-coloured bars, of the London underground network in Zones 1 and 2.*

- We can invert Fick's law of diffusion, $\mathbf{q} = -\mathrm{k}\mathbf{L}\boldsymbol{\phi}$, to estimate the population surrounding various stations as:

$$\widehat{\boldsymbol{\phi}} = -\frac{1}{k}\boldsymbol{L}^{+}\mathbf{q}$$

- Where:
  - $\boldsymbol{L}^{+}$ denotes the pseudo-inverse of the Laplacian matrix

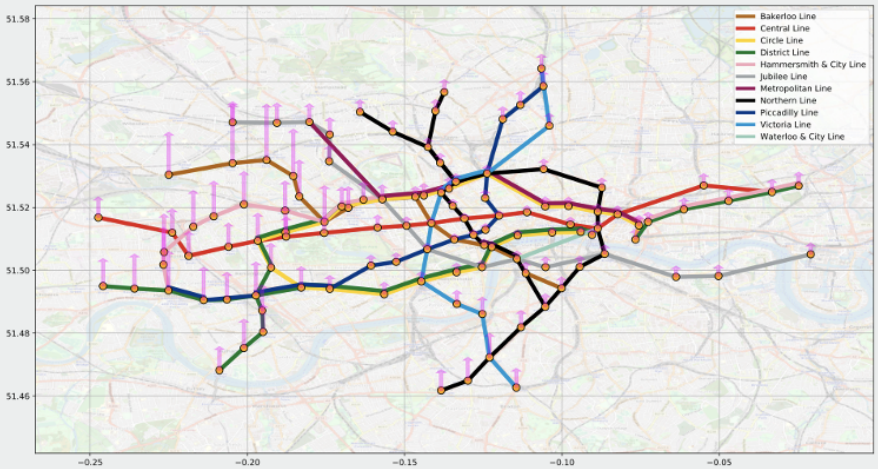


Fig. 7. *Population density implied by our diffusion graph model, obtained from the net passenger out-flow during the morning rush hour within Zones 1 and 2. As expected, business districts exhibit the lowest population density, while residential areas (Zone 2) exhibit the highest commuter population density*

Introducing new underground lines

- A graph is said to be k-edge connected if it cannot be regrouped into distinct sub-graphs unless k or more edges are removed
- The transport network forms a 1-edge connected graph, meaning that removal of 1 edge is enough to disconnect some stations from the network
- Robust network by k-*edge augmentation [2]:*
  - Determine the minimum set of additional edges, $\mathcal{A}$, such that $\mathcal{A} \cap \mathcal{E} = \emptyset$ and the resulting graph $\mathcal{G}_k = \{\mathcal{V}, \mathcal{E} \cup \mathcal{A}\}$ remains k-edge connected

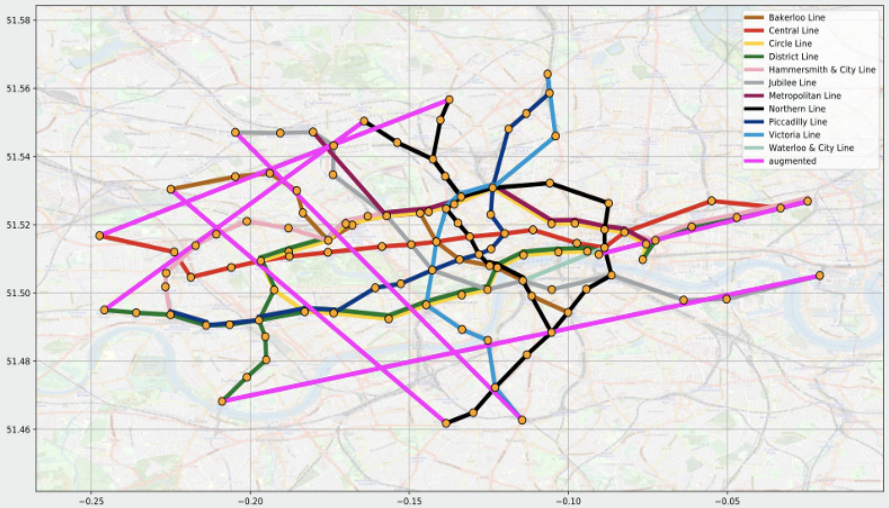


Fig. 3. *Graph of the London underground network in Zones 1 and 2, after performing a naive k-edge augmentation, for k=2.*

- Previous solution is optimal in terms of the number of new connections required, $|\mathcal{A}|$
- However, it is unrealistic to build such direct long-range connections
- Assuming that the cost of building a new connection is proportional to the geographic distance between two stations, $d(v_1, v_2)$, we can constrain the search space of the k-edge augmentation problem with a threshold, α :

$$d(v_1, v_2) < \alpha \text{ for all } (v_1, v_2) \in \mathcal{A}$$

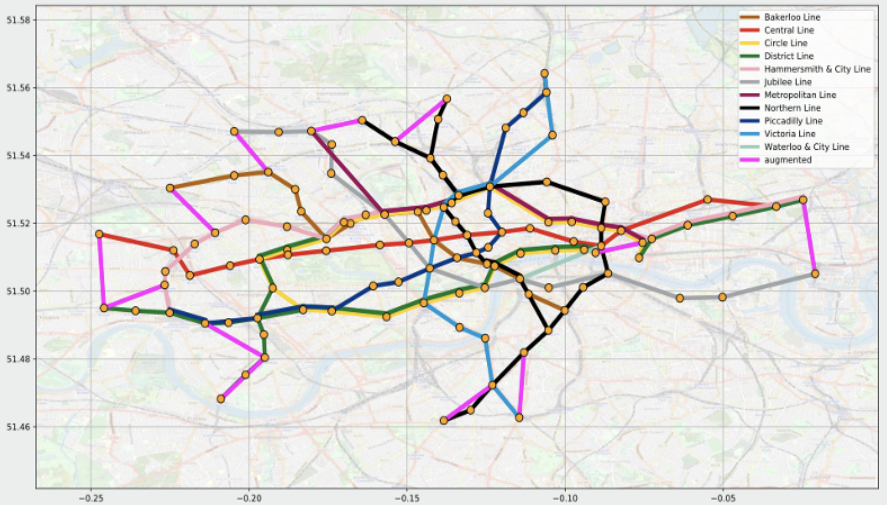


Fig. 4. *Graph of the London underground network in Zones 1 and 2 after performing geographically constrained k-edge augmentation, for k=2.*

Graphs for Transportation Networks

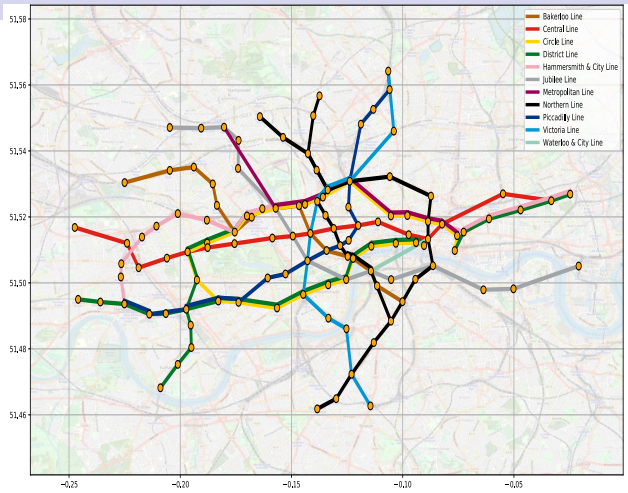


Fig. 1. Graph representation of the London underground network in Zones 1 and 2. The circles denote the vertices (stations) and the lines between the circles designate the underground lines.

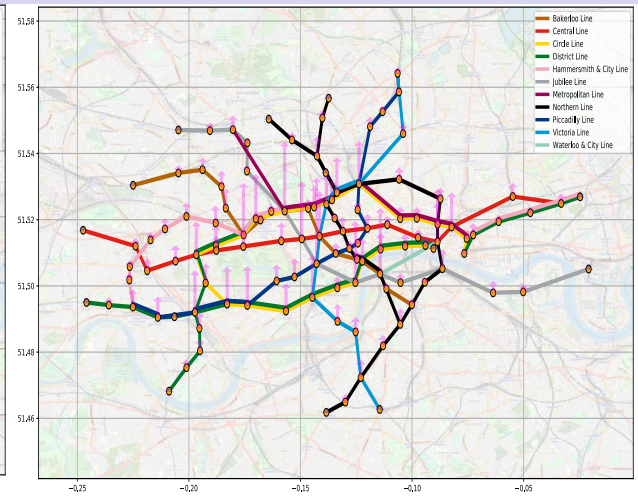


Fig. 2. Betweenness centrality, designated by magenta-coloured bars, of the London underground network in Zones 1 and 2.

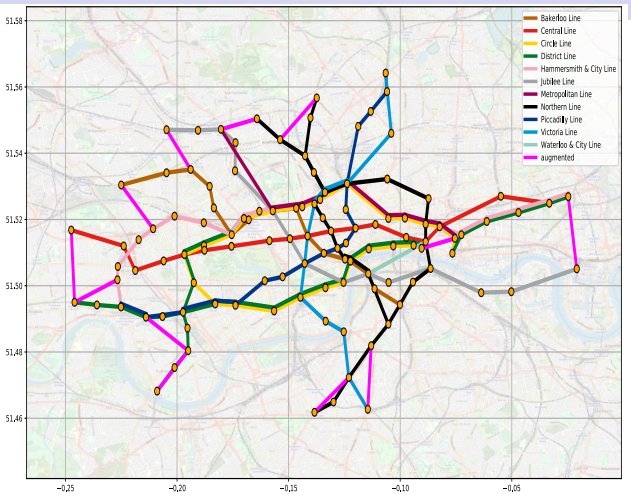


Fig. 3. Graph of the London underground network in Zones 1 and 2 after performing geographically constrained k-edge augmentation, for k=2.

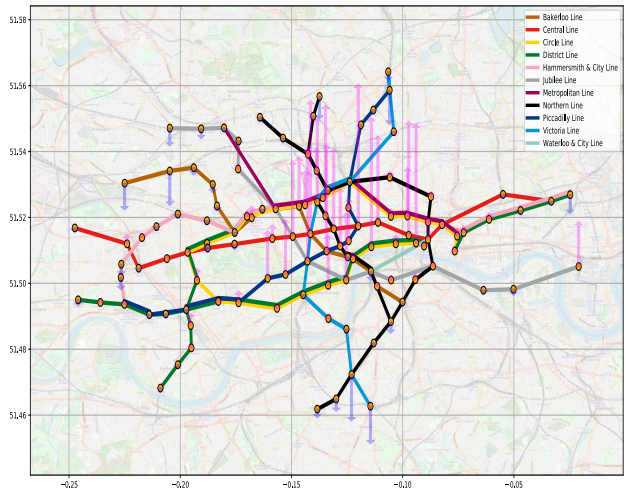


Fig. 4. Net passenger out-flow during the morning rush hour. The magenta bars denote a net out-flow of passengers while the blue bars designate a net in-flow of passengers. Stations located within business districts exhibit the greatest net out-flow of passengers, while stations located in residential areas, toward the outskirts, exhibit the largest net in-flow of passengers.

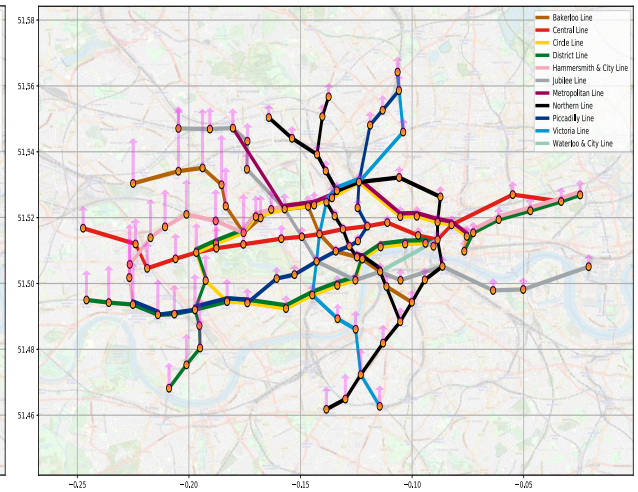


Fig. 5. Population density implied by our diffusion graph model, obtained from the net passenger out-flow during the morning rush hour within Zones 1 and 2. As expected, business districts exhibit the lowest population density, while residential areas (Zone 2) exhibit the highest commuter population density

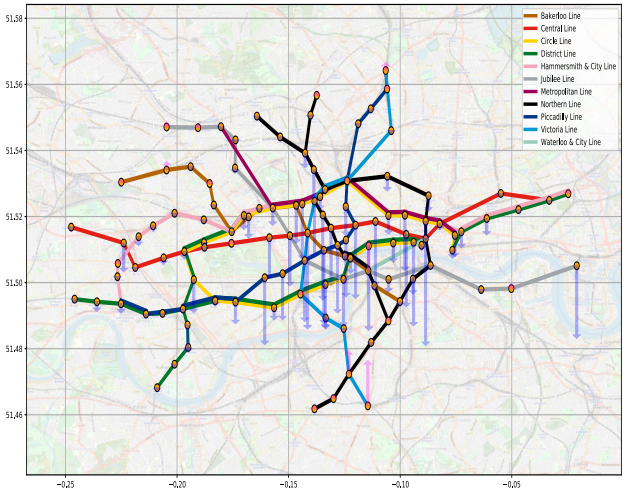
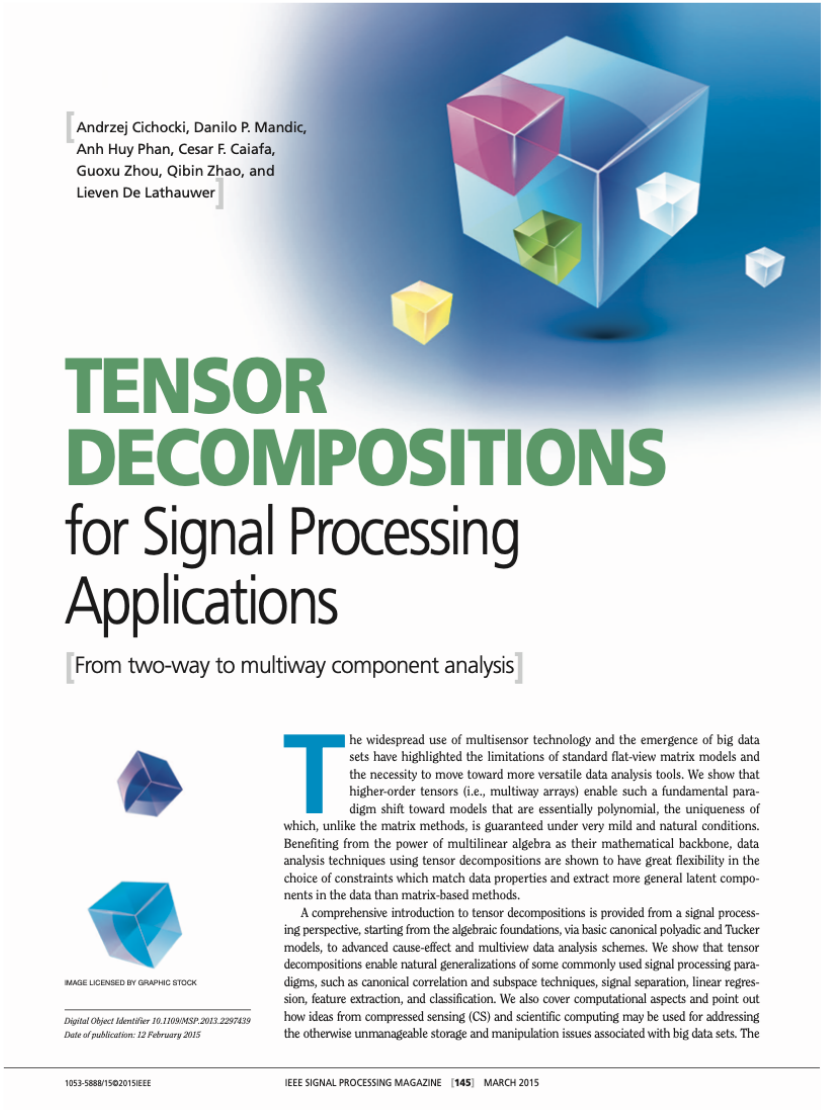


Fig. 6. The evening net passenger-flow, estimated from the morning rush hour data, based on the hypergraph neural network model. The power of the proposed hypergraph model is reflected in the reverse net flow from the morning rush hours, as people move from central business areas back to residential zones.

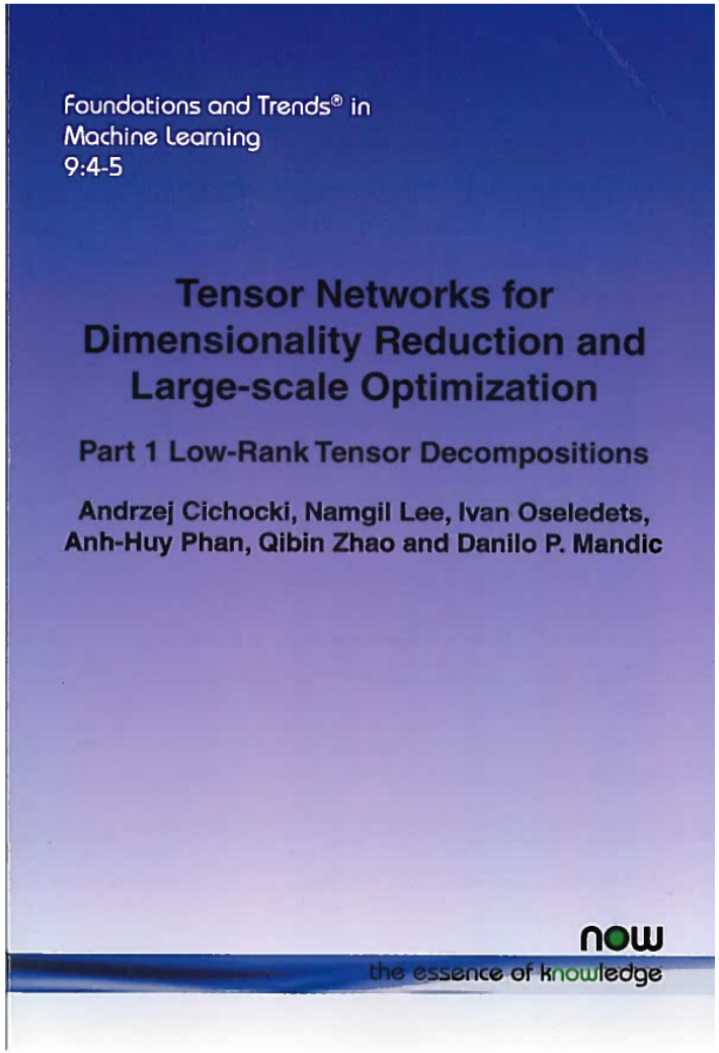
Our recent work on tensors

A. Cichocki, D. Mandic, *et al.*



IEEE SPM, March 2015

A. Cichocki, D. Mandic, *et al.*

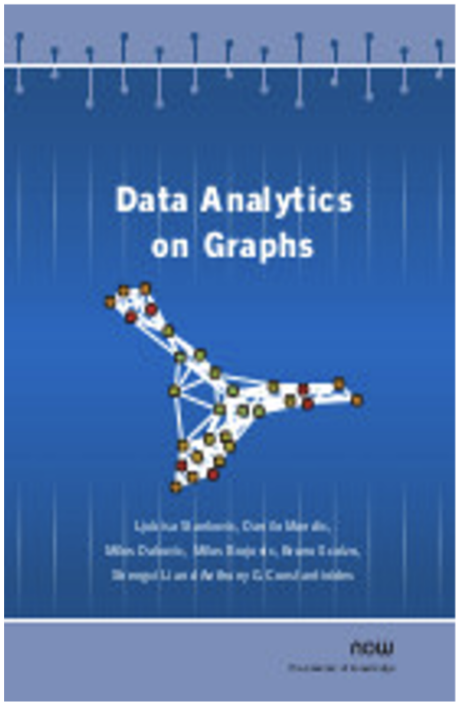


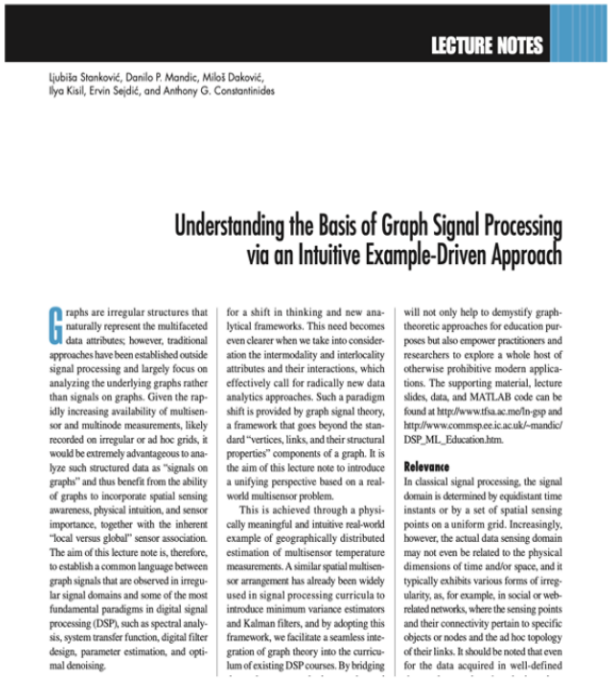
Foundations and Trends in Machine Learning, Parts 1 & 2

Our recent work on graphs

L. Stankovic, D. P. Mandic, *et al.*

A series of thee articles on **Data Analytics on Graphs** in three consecutive issues of **Foundations and Trends in Machine Learning**, 2020.
Part 1: **Graphs and Spectra on Graphs**, FnTML, vol. 13, no. 1, pp. 1-157
Part II: **Signals on Graphs**, vol. 13, no. 2-3, pp. 158-331
Part III: **Machine Learning on Graphs, from Graph Topology to Applications**, pp. 332-530




LECTURE NOTES

Ljubiša Stanković, Danilo P. Mandic, Miloš Daković, Ilya Kisil, Ervin Sejdić, and Anthony G. Constantinides

Understanding the Basis of Graph Signal Processing via an Intuitive Example-Driven Approach

Graphs are irregular structures that naturally represent the multifaceted data attributes; however, traditional approaches have been established outside signal processing and largely focus on analyzing the underlying graphs rather than signals on graphs. Given the rapidly increasing availability of multisensor and multinode measurements, likely recorded on irregular or ad hoc grids, it would be extremely advantageous to analyze such structured data as "signals on graphs" and thus benefit from the ability of graphs to incorporate spatial sensing awareness, physical intuition, and sensor importance, together with the inherent "local versus global" sensor association. The aim of this lecture note is, therefore, to establish a common language between graph signals that are observed in irregular signal domains and some of the most fundamental paradigms in digital signal processing (DSP), such as spectral analysis, system transfer function, digital filter design, parameter estimation, and optimal denoising.

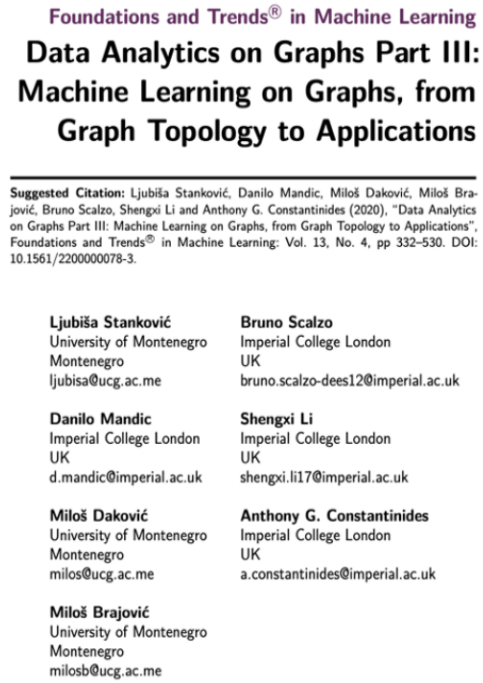
for a shift in thinking and new analytical frameworks. This need becomes even clearer when we take into consideration the intermodality and interlocality attributes and their interactions, which effectively call for radically new data analytics approaches. Such a paradigm shift is provided by graph signal theory, a framework that goes beyond the standard "vertices, links, and their structural properties" components of a graph. It is the aim of this lecture note to introduce a unifying perspective based on a real-world multisensor problem.

This is achieved through a physically meaningful and intuitive real-world example of geographically distributed estimation of multisensor temperature measurements. A similar spatial multisensor arrangement has already been widely used in signal processing curricula to introduce minimum variance estimators and Kalman filters, and by adopting this framework, we facilitate a seamless integration of graph theory into the curriculum of existing DSP courses. By bridging

will not only help to demystify graph-theoretic approaches for education purposes but also empower practitioners and researchers to explore a whole host of otherwise prohibitive modern applications. The supporting material, lecture slides, data, and MATLAB code can be found at http://www.tfsa.ac.me/In-gsp and http://www.commsp.ee.ic.ac.uk/~mandic/DSP_ML_Education.htm.

Relevance

In classical signal processing, the signal domain is determined by equidistant time instants or by a set of spatial sensing points on a uniform grid. Increasingly, however, the actual data sensing domain may not even be related to the physical dimensions of time and/or space, and it typically exhibits various forms of irregularity, as, for example, in social or web-related networks, where the sensing points and their connectivity pertain to specific objects or nodes and the ad hoc topology of their links. It should be noted that even for the data acquired in well-defined


Foundations and Trends® in Machine Learning

Data Analytics on Graphs Part III: Machine Learning on Graphs, from Graph Topology to Applications

Suggested Citation: Ljubiša Stanković, Danilo Mandic, Miloš Daković, Miloš Brajović, Bruno Scalzo, Shengxi Li and Anthony G. Constantinides (2020), "Data Analytics on Graphs Part III: Machine Learning on Graphs, from Graph Topology to Applications", Foundations and Trends® in Machine Learning: Vol. 13, No. 4, pp 332–530. DOI: 10.1561/2200000078-3.

Ljubiša Stanković
University of Montenegro
Montenegro
ljubisa@ucg.ac.me

Bruno Scalzo
Imperial College London
UK
bruno.scalzo-dees12@imperial.ac.uk

Danilo Mandic
Imperial College London
UK
d.mandic@imperial.ac.uk

Shengxi Li
Imperial College London
UK
shengxi.li17@imperial.ac.uk

Miloš Daković
University of Montenegro
Montenegro
milos@ucg.ac.me

Anthony G. Constantinides
Imperial College London
UK
a.constantinides@imperial.ac.uk

Miloš Brajović
University of Montenegro
Montenegro
milosb@ucg.ac.me