Machine Learning on Graphs

Gonzalo Mateos

Dept. of ECE and Goergen Institute for Data Science
University of Rochester
gmateosb@ece.rochester.edu
http://www.ece.rochester.edu/~gmateosb/

Facultad de Ingeniería, UdelaR
Montevideo, Uruguay
February 1, 2021

Roadmap

Introductions, context and motivation

Graph signal processing

Semi-supervised node classification

Network community detection

Link prediction

Who am I, where to find me, lecture times

- Gonzalo Mateos

  Dept. of ECE, University of Rochester

  Email: gmateosb@ece.rochester.edu

- Where? We meet online via Zoom

  Meeting ID: 919 6202 5440, passcode sent via email

- When? Daily from February 1 to 5, 9:00 am to 12:15 pm

- Class website

  https://eva.fing.edu.uy/course/view.php?id=1484

Local instructor team

- We will help you with questions, labs and the project
- Marcelo Fiori
  IMERL, FIng, UdelaR
  Email: mfiori@fing.edu.uy

- Federico La Rocca
  IIE, FIng, UdelaR
  Email: flarroca@fing.edu.uy

- Grateful for the help and for inviting me to teach this course

Tomorrow's guest lecturer

- Fernando Gama
  EECS Dept., UC Berkeley
  Email: fgama@berkeley.edu
- Graph neural networks (GNNs) expert
- Developer of PyTorch library to implement GNNs
  `https://github.com/alelab-upenn/graph-neural-networks`

Prerequisites

(I) Graph theory and statistical inference

- Graphs are mathematical abstractions of networks
- Statistical inference useful to “learn” from network data
- Basic knowledge expected. Asked you to go over review slides

(II) Probability theory and linear algebra

- Random variables, distributions, expectations, Markov processes
- Vector/matrix notation, systems of linear equations, eigenvalues

(III) Programming

- Will use e.g., Python for labs and your project
- You can use the language/network analysis package your prefer
- Several useful resources provided in the class website

Labs and project

(I) Exploratory labs (3 handouts, 10 hours total) worth 20%

- Coding assignments to experiment with data, libraries and methods
- Collaboration accepted, welcomed, and encouraged

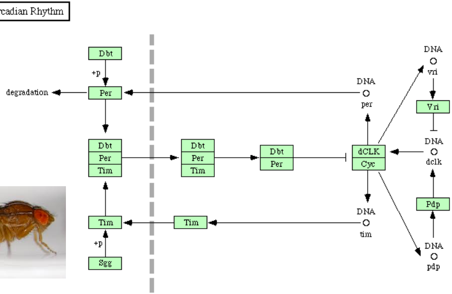
(II) Research project on a topic of your choice, worth 80%

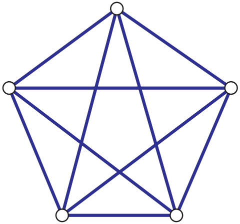
- Important part of this class. Work in pairs. Two deliverables:
  1) Proposal by Monday February 15, worth 15%
  2) Final report by Friday March 26, worth 65%

- This is a special topics, research-oriented graduate level class

  $\Rightarrow$ Focus should be on thinking, reading, asking, implementing

- As per the dictionary: *A collection of inter-connected things*
- Ok. There are multiple things, they are connected. Two extremes



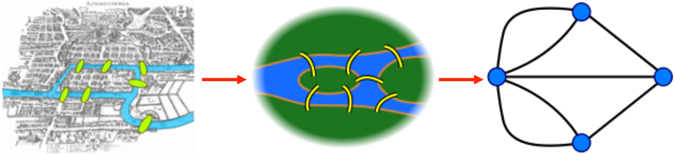


1) A real (complex) system of inter-connected components
2) A graph $G(\mathcal{V}, \mathcal{E})$ representing the system

- Understand complex systems $\Leftrightarrow$ Understand networks behind them

Historical background

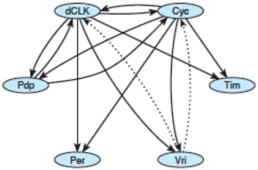
- Network-based analysis in the sciences has a long history
- Mathematical foundations of graph theory (L. Euler, 1735)

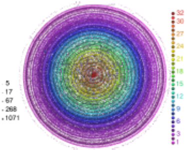


  - The seven bridges of Königsberg
- Laws of electrical circuitry (G. Kirchoff, 1845)
- Molecular structure in chemistry (A. Cayley, 1874)
- Network representation of social interactions (J. Moreno, 1930)
- Power grids (1910), telecommunications and the Internet (1960)
- Google (1997), Facebook (2004), Twitter (2006), . . .

Why networks? Why now?

- Understand complex systems ⇔ Understand networks behind them





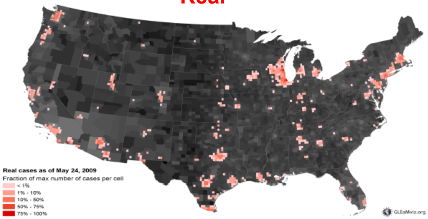
- Relatively small field of study up until $\sim$ the mid-90s
- Epidemic-like explosion of interest recently. A few reasons:
  - Systems-level perspective in science, away from reductionism
  - Ubiquitous high-throughput data collection, computational power
  - Globalization, the Internet, connectedness of modern societies
  - Data complexity: heterogeneity, dependence, dynamism, . . .
- Impact: social networking, drug design, smart infrastructure, . . .

Economic impact

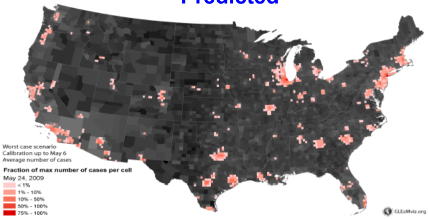
- Google
  Market cap:
  $1.24 trillion
- Facebook
  Market cap:
  $736 billion
- Cisco
  Market cap:
  $188 billion
- Apple
  Market cap:
  $2.22 trillion

- Prediction of epidemics, e.g. the 2009 H1N1 pandemic

Real

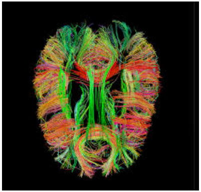


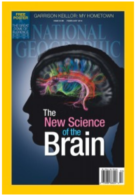
Predicted



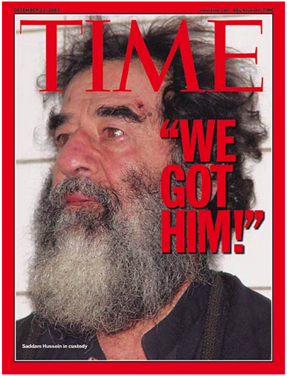
- Human Connectome Project to map-out brain circuitry

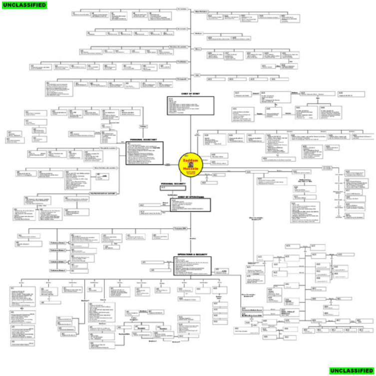
HUMAN Connectome PROJECT





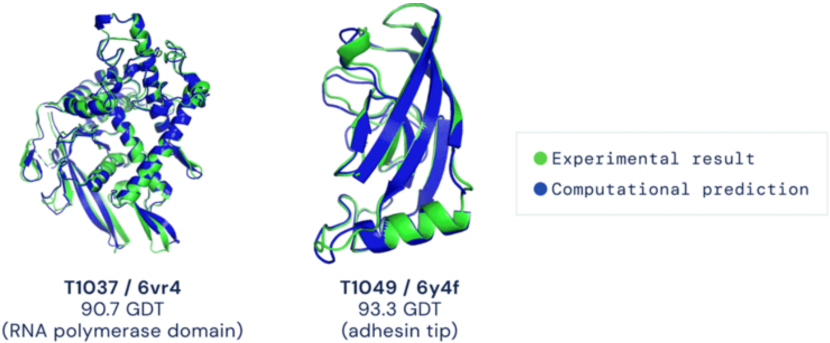
- Social network analysis key to capturing S. Hussein





Scientific discovery impact

- Machine learning on graphs key to solving protein folding



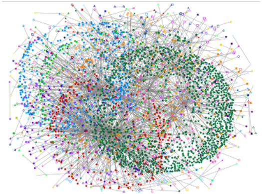
- Predict protein's 3D structure given 1D amino acid sequence
  - $\Rightarrow$ Astronomical ($\approx 10^{300}$) number of possible foldings

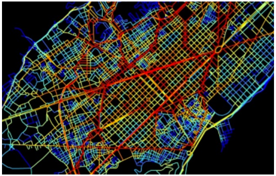
Network data science: goals and characteristics

- Universal language for describing complex systems and data
  - Striking similarities in networks across science, nature, technology
- What are the goals of network data science?
  - Reveal patterns and statistical properties of network data
  - Understand the underpinnings of network behavior and structure
  - Engineer more resource-efficient, robust, socially-intelligent networks
- Characteristics: interdisciplinary, empirical, quantitative, computational
- Empirical study of graph-valued data to find patterns and principles
  - Collection, measurement, summarization, visualization?
- Mathematical models. Graph theory meets statistical inference
  - Understand, predict, discern nominal vs anomalous behavior?
- Algorithms for graph analytics
  - Computational challenges, scalability, tractability vs optimality?

Broad scope and areas of interest

- Network data science key to advance
  - Climate systems
  - Network neuroscience
  - Collaborative intelligence/autonomy
  - Information networks
  - Societies and civilization
  - Urban systems
  - Critical infraestructure





- Broad topics of interest
  - Coupling of natural, technological and social networks
  - Resilience and adaptation: climate change, migration, pandemics, . . .

What is this class about?

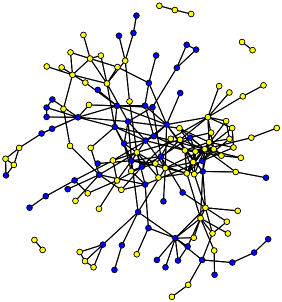
- Our focus: Machine learning for network data
- Measurements of or from a system conceptualized as a network
- Unique challenges
  - Relational aspect of the data
  - Complex statistical dependencies
  - High-dimensional and often massive in quantity
  - Lack of strong structural and geometric priors
- Will examine how these challenges arise in relation to
  - Visualization
  - Summarization and representation learning
  - Sampling and inference
  - Modeling

Machine learning on graphs: tasks

- Graph visualization and pattern discovery
  - Ex: How is the science and technology enterprise developing?
- Graph modeling and generation
  - Ex: Generate new molecules with antibacterial activity?
- Clustering and community detection
  - Ex: Which groups of individuals have similar political beliefs?
- Link prediction
  - Ex: Predict user-item interactions in recommendation systems?
- Node classification and semi-supervised learning
  - Ex: Can we identify protein function from their physical binding?
- Graph classification
  - Ex: Diagnose subjects with cognitive decline from brain connectomes?

Example: Predicting protein function

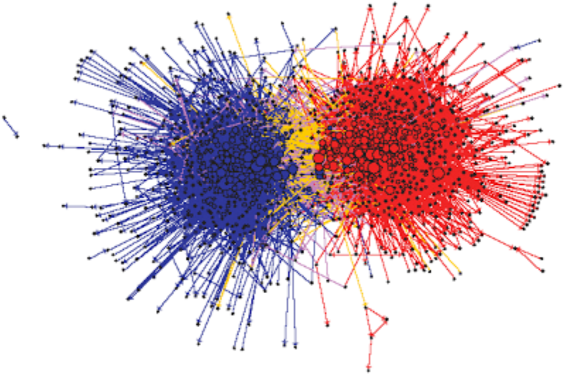
- Baker's yeast data, formally known as *Saccharomyces cerevisiae*
  - Graph: 134 vertices (proteins) and 241 edges (protein interactions)



- Signal: functional annotation intracellular signaling cascade (ICSC)
  - Signal transduction, how cells react to the environment
  - $x_i = 1$ if protein i annotated ICSC (yellow), $x_i = 0$ otherwise (blue)

Example: Unveiling network communities

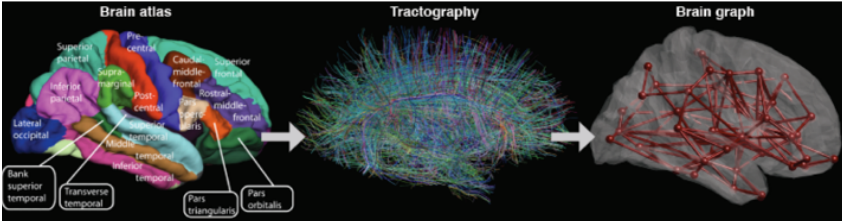
- The political blogosphere for the US 2004 presidential election



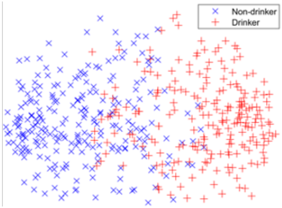
- Community structure of liberal and conservative blogs is apparent

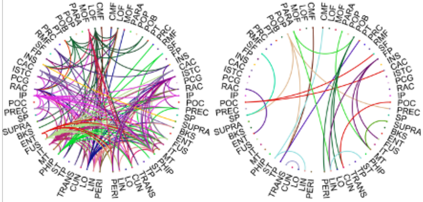
  $\Rightarrow$ People have a stronger tendency to interact with "equals"

- Challenge: understanding human brain function and structure



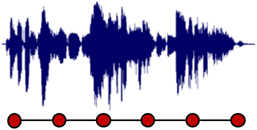
- Does brain connectivity change for heavy drinkers [Li et al'20]?

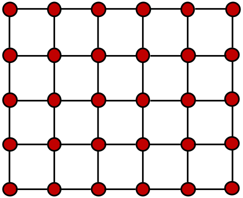




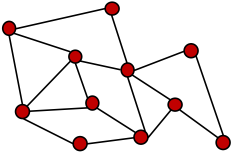
Machine learning on graphs: fundamental challenge

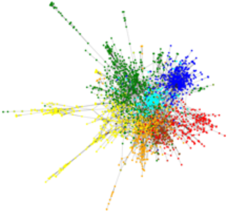
- We've become good at learning from data in Euclidean domains





- But we want to learn from data defined on graphs





$\Rightarrow$ Challenge: no geometry ($\mathcal{V}$ is a set), irregular neighborhoods

$\Rightarrow$ Ordering? Translation? Convolution? Structural priors?

Roadmap

Introductions, context and motivation

Graph signal processing

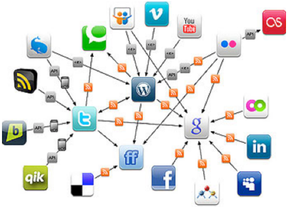
Semi-supervised node classification

Network community detection

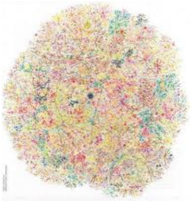
Link prediction

From graphs to graph signals

Online social media



Internet



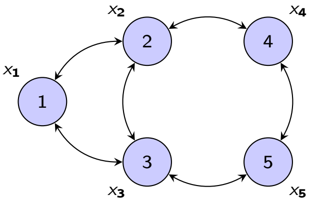
Clean energy and grid analytics

- Network as graph $G = (\mathcal{V}, \mathcal{E})$: encode pairwise relationships
- Desiderata: Process, analyze and learn from network data [Kolaczyk'09]

  $\Rightarrow$ Use G to study graph signals, data associated with nodes in $\mathcal{V}$
- Ex: Opinion profile, buffer congestion levels, neural activity, epidemic

Graph signal processing (GSP)

- Graph G with adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$
  - $\Rightarrow A_{ij}$ = proximity between i and j
- Define a signal $\mathbf{x} \in \mathbb{R}^N$ on top of the graph
  - $\Rightarrow x_i$ = signal value at node i

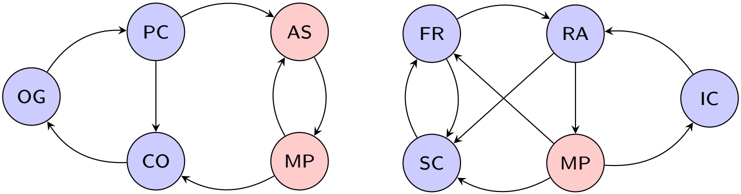


- Graph Signal Processing $\rightarrow$ Exploit structure encoded in $\mathbf{A}$ to process $\mathbf{x}$
- **Q:** Graph signals common and interesting as networks are?
- **Q:** Why do we expect the graph structure to be useful in processing $\mathbf{x}$?

Network of economic sectors of the United States

- Bureau of Economic Analysis of the U.S. Department of Commerce
  - A_{ij} = Output of sector i that becomes input to sector j (62 sectors)

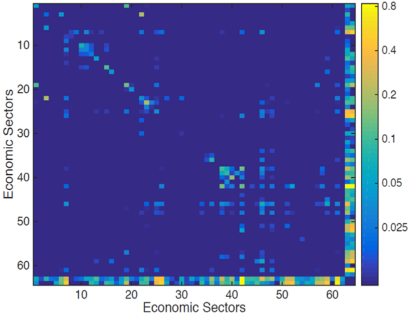
Oil and Gas Services Finance



- Oil extraction (OG), Petroleum and coal products (PC), Construction (CO)
- Administrative services (AS), Professional services (MP)
- Credit intermediation (FR), Securities (SC), Real state (RA), Insurance (IC)
- Only interactions stronger than a threshold are shown

Network of economic sectors of the United States

- Bureau of Economic Analysis of the U.S. Department of Commerce
  - A_{ij} = Output of sector i that becomes input to sector j (62 sectors)



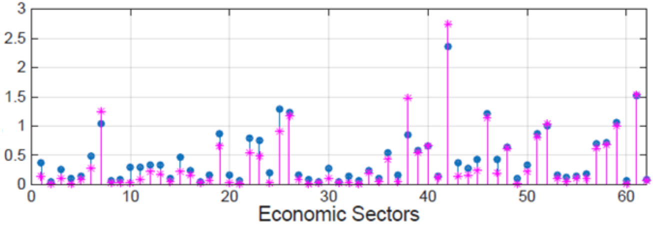
- A few sectors have widespread strong influence (services, finance, energy)
- Some sectors have strong indirect influences (oil)
- The heavy last row is final consumption

- This is an interesting network $\Rightarrow$ Signals on this graph are as well

Disaggregated GDP of the United States

- Signal $\mathbf{x}$ = output per sector = disaggregated GDP

  $\Rightarrow$ Network structure used to, e.g., reduce GDP estimation noise



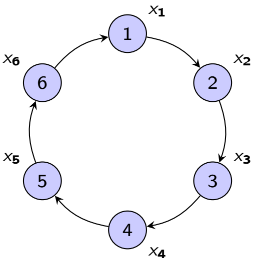
- Signal is as interesting as the network itself. Arguably more
  - Same is true for brain connectivity and fMRI brain signals, ...
  - Gene regulatory networks and gene expression levels, ...
  - Online social networks and information cascades, ...

Importance of signal structure in time

- Signal and Information Processing is about exploiting signal structure

- Discrete time described by cyclic graph
  - $\Rightarrow$ Time n follows time $n-1$
  - $\Rightarrow$ Signal value x_n similar to x_{n-1}

- Formalized with the notion of frequency

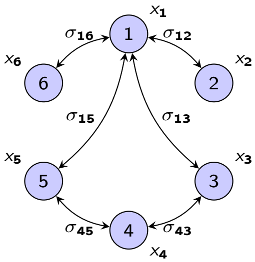


- Cyclic structure $\Rightarrow$ Fourier transform $\Rightarrow \tilde{\mathbf{x}} = \mathbf{F}^H \mathbf{x}$ $\left(F_{kn} = \frac{e^{j2\pi kn/N}}{\sqrt{N}} \right)$

- **Fourier transform** $\Rightarrow$ Projection on eigenvector space of cycle

Covariances and principal components

- Random signal with mean $\mathbb{E}[\mathbf{x}] = 0$ and covariance $\mathbf{C}_x = \mathbb{E}\left[\mathbf{x}\mathbf{x}^H\right]$
  - $\Rightarrow$ Eigenvector decomposition $\mathbf{C}_x = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^H$
- Covariance matrix $\mathbf{A} = \mathbf{C}_x$ is a graph
  - $\Rightarrow$ Not a very good graph, but still
- Precision matrix $\mathbf{C}_x^{-1}$ a common graph too
  - $\Rightarrow$ Conditional dependencies of Gaussian $\mathbf{x}$



- Covariance matrix structure $\Rightarrow$ Principal components (PCA) $\Rightarrow \tilde{\mathbf{x}} = \mathbf{V}^H\mathbf{x}$
- **PCA transform** $\Rightarrow$ Projection on eigenvector space of (inverse) covariance
- **Q:** Can we extend these principles to general graphs and signals?

Graph Fourier Transform

- Adjacency $\mathbf{A}$, Laplacian $\mathbf{L}$, or, generically graph shift $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$
  - $\Rightarrow S_{ij} = 0$ for $i \neq j$ and $(i, j) \notin \mathcal{E}$ (captures local structure in G)
- The Graph Fourier Transform (GFT) of $\mathbf{x}$ is defined as

$$\tilde{\mathbf{x}} = \mathbf{V}^{-1}\mathbf{x}$$

- While the inverse GFT (iGFT) of $\tilde{\mathbf{x}}$ is defined as

$$\mathbf{x} = \mathbf{V}\tilde{\mathbf{x}}$$

  - $\Rightarrow$ Eigenvectors $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_N]$ are the frequency basis (atoms)
- Additional structure
  - $\Rightarrow$ If $\mathbf{S}$ is normal, then $\mathbf{V}^{-1} = \mathbf{V}^H$ and $\tilde{x}_k = \mathbf{v}_k^H \mathbf{x} = < \mathbf{v}_k, \mathbf{x} >$
  - $\Rightarrow$ Parseval holds, $\|\mathbf{x}\|^2 = \|\tilde{\mathbf{x}}\|^2$
- **GFT** $\Rightarrow$ Projection on eigenvector space of graph shift operator $\mathbf{S}$

- Total variation of signal $\mathbf{x}$ with respect to $\mathbf{L}$

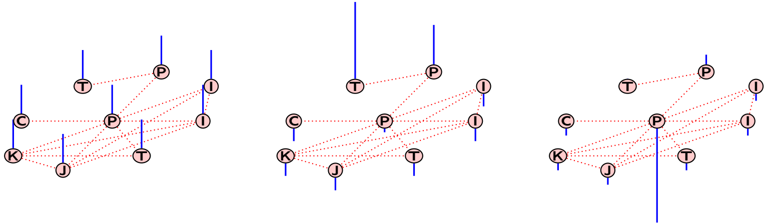
$$\mathrm{TV}(\mathbf{x}) = \mathbf{x}^\top \mathbf{L}\mathbf{x} = \sum_{i,j=1,j>i}^{N} A_{ij}(x_i - x_j)^2$$

$\Rightarrow$ Smoothness measure on the graph G (Dirichlet energy)

- For Laplacian eigenvectors $\mathbf{V} = [\mathbf{v}_1, \cdots, \mathbf{v}_N]$ $\Rightarrow \mathrm{TV}(\mathbf{v}_k) = \lambda_k$

  $\Rightarrow$ Can view $0 = \lambda_1 < \cdots \leq \lambda_N$ as frequencies

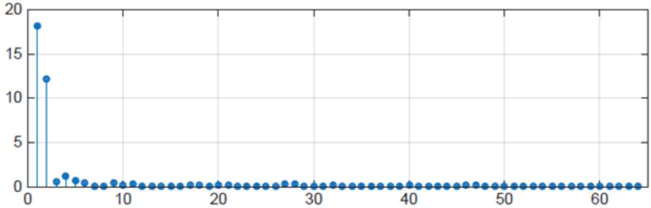
- Ex: gene network, $N = 10$, $k = 1$, $k = 2$, $k = 9$



Is this a reasonable transform?

- Particularized to cyclic graphs $\Rightarrow$ GFT $\equiv$ Fourier transform
- Also for covariance graphs $\Rightarrow$ GFT $\equiv$ PCA transform

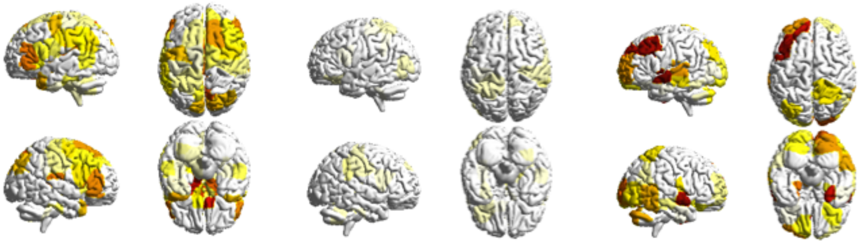
- But really, this is an empirical question. GFT of disaggregated GDP



- Spectral domain representation characterized by a few coefficients
  - $\Rightarrow$ Notion of bandlimitedness: $\mathbf{x} = \sum_{k=1}^{K} \tilde{x}_k \mathbf{v}_k$
  - $\Rightarrow$ Sampling, compression, filtering, pattern recognition

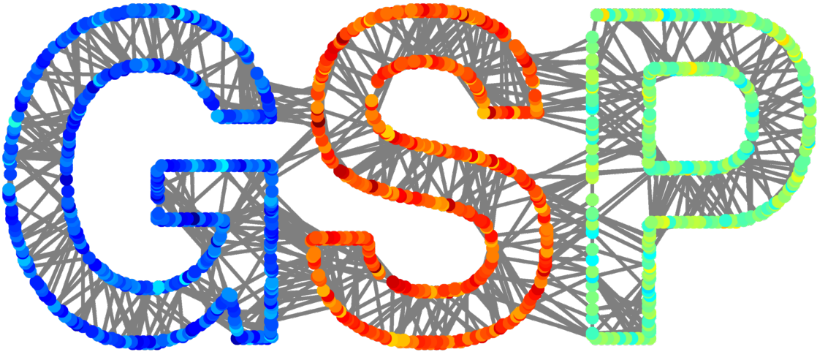
Graph frequency analysis of brain signals

- GFT of brain signals during a visual-motor learning task [Huang et al'16]
  - ⇒ Decomposed into low, medium and high frequency components



- Brain: Complex system where regularity coexists with disorder [Sporns'11]
  - ⇒ Signal energy mostly in the low and high frequencies
  - ⇒ In brain regions akin to the visual and sensorimotor cortices

PyGSP: Graph Signal Processing in Python

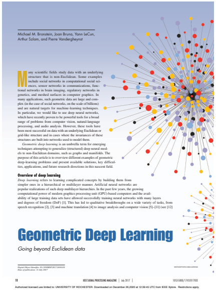


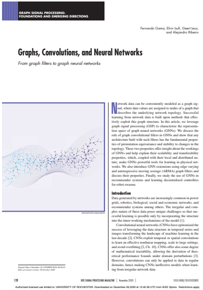
- PyGSP is a Python package to ease SP on graphs. Free software

Available from https://github.com/epfl-lts2/pygsp

- **Goal:** successful learning from network data
  - $\Rightarrow$ Representation methods that effectively exploit graph structure

- From GSP to graph neural networks (GNNs)
  - Linear graph filters and convolutions plus pointwise nonlinearities
  - Permutation equivariance, stability to graph perturbations, transferability
  - Theoretical insights on GNN's strong generalization potential





Roadmap

Introductions, context and motivation

Graph signal processing

Semi-supervised node classification

Network community detection

Link prediction

Nearest-neighbor prediction

- Consider classification of a signal $\mathbf{x} := \{x_i\}_{i \in \mathcal{V}}$ on a graph

Network process prediction

Predict x_i, given observations of the adjacency matrix $\mathbf{A}$ and of all attributes $\mathbf{x}^{(-i)}$ but x_i.

- Semi-supervised learning: only a small fraction of nodes labeled
- Idea: exploit the network graph structure in $\mathbf{A}$ for classification
- For binary $x_i \in \{0, 1\}$, say, simple nearest-neighbor method predicts

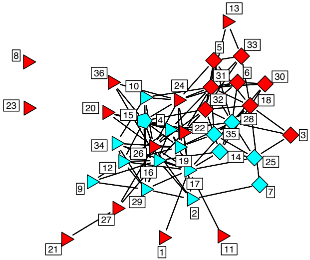
$$\hat{x}_i = \mathbb{I}\left\{\frac{\sum_{j \in \mathcal{N}_i} x_j}{|\mathcal{N}_i|} > \tau\right\}$$

$\Rightarrow$ Average of the observed signal in $\mathcal{N}_i$ (neighborhood of i)

$\Rightarrow$ Called 'guilt-by-association' or graph-smoothing method

Example: predicting law practice

- Network G^{obs} of working relationships among lawyers [Lazega'01]
  - Nodes are $N_v = 36$ partners, edges indicate partners worked together



- Data includes various node-level attributes $\{x_i\}_{i \in \mathcal{V}}$ including

  $\Rightarrow$ Type of practice, i.e., litigation (red) and corporate (cyan)

- Suspect lawyers collaborate more with peers in same legal practice

  $\Rightarrow$ Knowledge of collaboration useful in predicting type of practice

Example: predicting law practice (cont.)

- Q: In predicting practice x_i, how useful is the value of one neighbor?

  ⇒ Breakdown of 115 edges based on practice of incident lawyers

	Litigation	**Corporate**
Litigation	29	43
Corporate	43	43

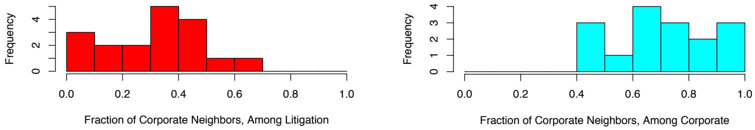
- Looking at the rows in this table
  - Litigation lawyers collaborators are 40% litigation, 60% corporate
  - Collaborations of corporate lawyers are evenly split

  ⇒ Suggests using a single neighbor has little predictive power
- But 60% (29+43=72) of edges join lawyers with common practice

  ⇒ Suggests on aggregate knowledge of collaboration informative

Example: predicting law practice (cont.)

- Incorporate information of all collaborators as in nearest-neighbors
  - Let $x_i = 0$ if lawyer i practices litigation, and $x_i = 1$ for corporate



- Nearest-neighbor prediction rule

$$\hat{x}_i = \mathbb{I}\left\{\frac{\sum_{j\in\mathcal{N}_i} x_j}{|\mathcal{N}_i|} > 0.5\right\}$$

$\Rightarrow$ Infers correctly 13 of the 16 corporate lawyers (i.e., 81%)

$\Rightarrow$ Infers correctly 16 of the 18 litigation lawyers (i.e., 89%)

$\Rightarrow$ Overall error rate is just under 15%

Where do we go from here?

- Nearest-neighbor methods may seem rather informal and simple
  - $\Rightarrow$ But competitive with more formal, model-based approaches
- Model the signal $\mathbf{x} := \{x_i\}_{i \in \mathcal{V}}$ given an observed graph $\mathbf{A}$
  - $\Rightarrow$ Markov random field (MRF) models
  - $\Rightarrow$ Kernel-regression models using graph kernels
- **Key:** implicit is a smoothness assumption of $\mathbf{x}$ w.r.t. G
  - $\Rightarrow$ Usually understood as $\text{TV}(\mathbf{x}) = \mathbf{x}^\top \mathbf{L} \mathbf{x}$ being small
- Will adopt as graph regularization for machine learning tasks

$$\min_{\mathbf{x}} f(\mathbf{x}) + \mathbf{x}^\top \mathbf{L} \mathbf{x}$$

... and in the context of graph learning from data

$$\min_{\mathbf{L}} \mathbf{x}^\top \mathbf{L} \mathbf{x} + g(\mathbf{L})$$

Roadmap

Introductions, context and motivation

Graph signal processing

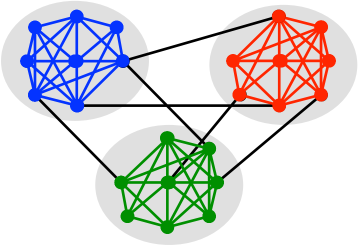
Semi-supervised node classification

Network community detection

Link prediction

Unveiling network communities

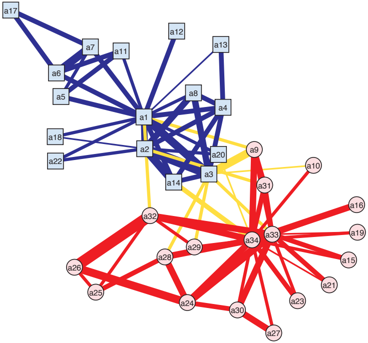
- Nodes in real-world networks organize into communities
  Ex: families, clubs, political organizations, proteins by function, ...



- Community (a.k.a. group, cluster, module) members are:
  - ⇒ Well connected among themselves
  - ⇒ Relatively well separated from the rest
- Exhibit high cohesiveness w.r.t. the underlying relational patterns
- **Q:** How can we automatically identify such cohesive subgroups?

Zachary's karate club

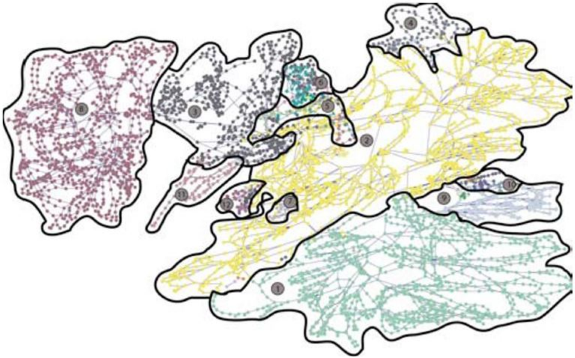
- Social interactions among members of a karate club in the 70s



- Zachary witnessed the club split in two during his study
  - $\Rightarrow$ Toy network, yet canonical for community detection algorithms
  - $\Rightarrow$ Offers "ground truth" community membership (a rare luxury)

Electrical power grid

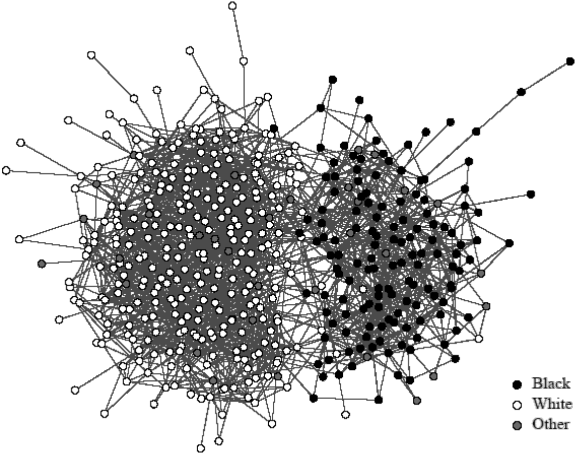
- Split power network into areas with minimum inter-area interactions



- **Applications:**
  - Decide control areas for distributed power system state estimation
  - Parallel computation of power flow
  - Controlled islanding to prevent spreading of blackouts

High-school students

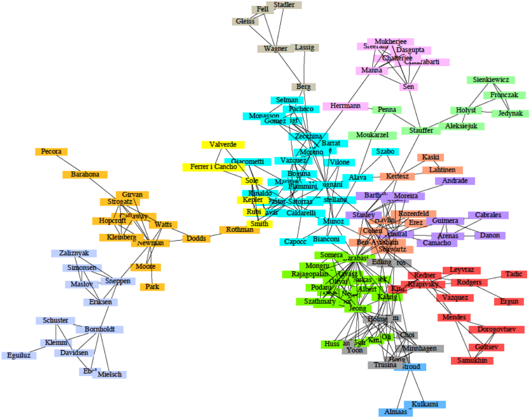
- Network of social interactions among high-school students



- Strong assortative mixing, with race as latent characteristic

Physicists working on Network Science

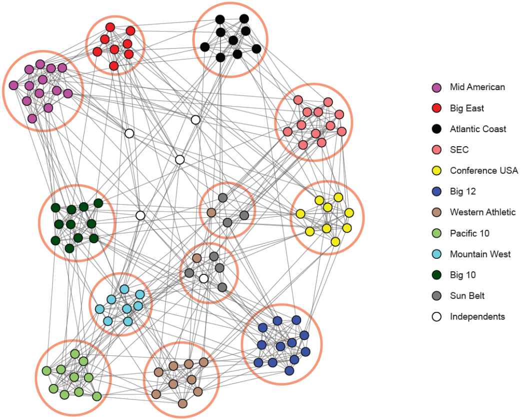
- Coauthorship network of physicists publishing networks' research



- Tightly-knit subgroups are evident from the network structure

College football

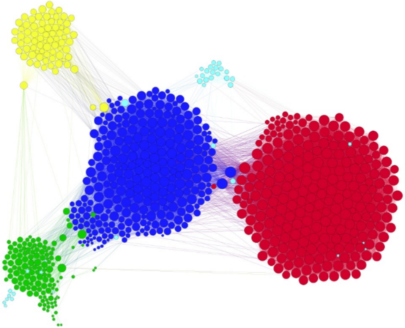
- Vertices are NCAA football teams, edges are games during Fall'00



- Communities are the NCAA conferences and independent teams

Facebook friendships

- Facebook egonet with 744 vertices and 30K edges



- Asked "ego" to identify social circles to which friends belong
  - $\Rightarrow$ Company, high-school, basketball club, squash club, family

Community detection and graph partitioning

- Community detection is a challenging clustering problem
  - C1) No consensus on the structural definition of community
  - C2) Node subset selection often intractable
  - C3) Lack of ground-truth for validation

- Useful for exploratory analysis of network data

  Ex: clues about social interactions, content-related web pages

Graph partitioning

Split $\mathcal{V}$ into given number of non-overlapping groups of given sizes

- **Criterion:** number of edges between groups is minimized (more soon)

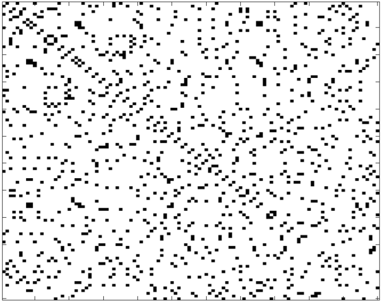
  Ex: task-processor assignment for load balancing

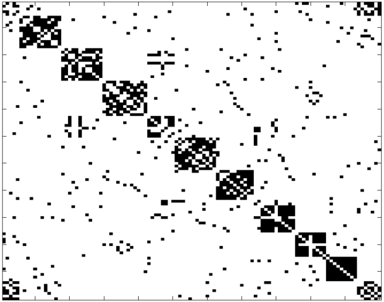
- Number and sizes of groups unspecified in community detection

  $\Rightarrow$ Identify the natural fault lines along which a network separates

Community detection in a nutshell

- Given a graph $G(\mathcal{V}, \mathcal{E})$ with adjacency matrix $\mathbf{A}$ (left)





- Find row/column permutation to reveal block-diagonal structure (right)

  Ex: NCAA college football network we saw earlier [Mateos-Giannakis'12]

Graph partitioning is hard

- Ex: Graph bisection problem, i.e., partition $\mathcal{V}$ into two groups
  - Suppose the groups $\mathcal{V}_1$ and $\mathcal{V}_2$ are non-overlapping
  - Suppose groups have equal size, i.e., $|\mathcal{V}_1| = |\mathcal{V}_2| = N_v/2$
  - Minimize edges running between vertices in different groups
- Simple problem to describe, but hard to solve

$$\text{Number of ways to partition } \mathcal{V}: \quad \binom{N_v}{N_v/2} \approx \frac{2^{N_v}}{\sqrt{N_v}}$$

$\Rightarrow$ Used Stirling's formula $N_v! \approx \sqrt{2\pi N_v}(N_v/e)^{N_v}$

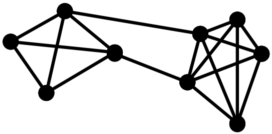
$\Rightarrow$ Exhaustive search intractable beyond toy small-sized networks

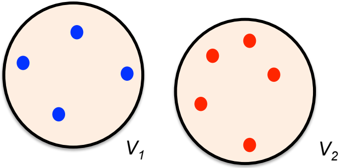
- No smart (i.e., polynomial time) algorithm, NP-hard problem

$\Rightarrow$ Seek good heuristics, e.g., relaxations of natural criteria

Graph bisection

- Undirected graph $G(\mathcal{V}, \mathcal{E})$. Partition $\mathcal{V}$ into two groups
  - Groups $\mathcal{V}_1$ and $\mathcal{V}_2 = \mathcal{V}_1^C$ are non-overlapping
  - Groups have given size, i.e., $|\mathcal{V}_1| = N_1$ and $|\mathcal{V}_2| = N_2$

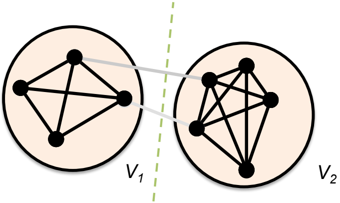




- **Q:** What is a natural criterion to partition the graph?

Graph cut

- Desiderata: Community members should be
  - $\Rightarrow$ Well connected among themselves; and
  - $\Rightarrow$ Relatively well separated from the rest of the nodes



- **Def:** A cut C is the number of edges between groups $\mathcal{V}_1$ and $\mathcal{V} \setminus \mathcal{V}_1$

$$C := \text{cut}(\mathcal{V}_1, \mathcal{V}_2) = \sum_{i \in \mathcal{V}_1, j \in \mathcal{V}_2} A_{ij}$$

- Natural criterion: minimize cut, i.e., edges across groups $\mathcal{V}_1$ and $\mathcal{V}_2$

From graph cuts . . .

- Binary community membership variables per vertex

$$u_i = \begin{cases} +1, & \text{vertex } i \text{ belongs to } \mathcal{V}_1 \\ -1, & \text{vertex } i \text{ belongs to } \mathcal{V}_2 \end{cases}$$

- We can indicate two vertices are in different groups as

$$\mathbb{I}\left\{u_i \neq u_j\right\} = \frac{1}{2}(1 - u_i u_j) = \begin{cases} 1, & i \text{ and } j \text{ in different groups} \\ 0, & i \text{ and } j \text{ in the same group} \end{cases}$$

- Cut expressible in terms of the variables u_i as

$$C = \sum_{i \in \mathcal{V}_1, j \in \mathcal{V}_2} A_{ij} = \frac{1}{2} \sum_{i,j \in \mathcal{V}} A_{ij}(1 - u_i u_j)$$

- First summand in $C = \frac{1}{2}\sum_{i,j} A_{ij}(1 - u_i u_j)$ is

$$\sum_{i,j\in\mathcal{V}} A_{ij} = \sum_{i\in\mathcal{V}} d_i = \sum_{i\in\mathcal{V}} d_i u_i^2 = \sum_{i,j\in\mathcal{V}} d_i u_i u_j \mathbb{I}\{i = j\}$$

- Used $u_i^2 = 1$ since $u_i \in \{\pm 1\}$. The cut becomes

$$C = \frac{1}{2}\sum_{i,j\in\mathcal{V}} (d_i \mathbb{I}\{i = j\} - A_{ij}) u_i u_j = \frac{1}{2}\sum_{i,j\in\mathcal{V}} L_{ij} u_i u_j$$

- Cut in terms of L_{ij}, entries of the graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$, i.e.,

$$C(\mathbf{u}) = \frac{1}{2}\mathbf{u}^\top \mathbf{L}\mathbf{u}, \quad \mathbf{u} := [u_1, \ldots, u_{N_v}]^\top$$

Graph cut minimization

- Since $|\mathcal{V}_1| = N_1$ and $|\mathcal{V}_2| = N_2 = N_v - N_1$, we have the constraint

$$\sum_{i \in \mathcal{V}} u_i = \sum_{i \in \mathcal{V}_1} (+1) + \sum_{i \in \mathcal{V}_2} (-1) = N_1 - N_2 \ \Rightarrow \ \mathbf{1}^\top \mathbf{u} = N_1 - N_2$$

- Minimum-cut criterion for graph bisection yields the formulation

$$\hat{\mathbf{u}} = \arg \min_{\mathbf{u} \in \{\pm 1\}^{N_v}} \mathbf{u}^\top \mathbf{L} \mathbf{u}, \quad \text{s. to } \mathbf{1}^\top \mathbf{u} = N_1 - N_2$$

- Binary constraints $\mathbf{u} \in \{\pm 1\}^{N_v}$ render cut minimization hard

Laplacian matrix properties

- Smoothness: For any vector $\mathbf{x} \in \mathbb{R}^{N_v}$ of "vertex values", one has

$$\mathbf{x}^\top \mathbf{L}\mathbf{x} = \sum_{i,j \in \mathcal{V}} L_{ij} x_i x_j = \sum_{(i,j) \in \mathcal{E}} (x_i - x_j)^2$$

  which can be minimized to enforce smoothness of functions on G

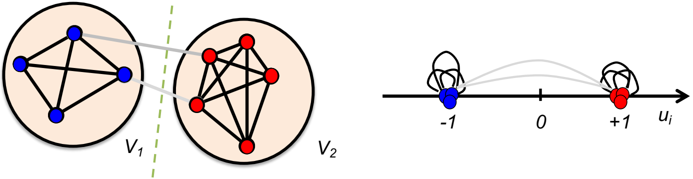
- Positive semi-definiteness: Follows since $\mathbf{x}^\top \mathbf{L}\mathbf{x} \geq 0$ for all $\mathbf{x} \in \mathbb{R}^{N_v}$
- Spectrum: All eigenvalues of $\mathbf{L}$ are real and non-negative
  $\Rightarrow$ Eigenvectors form an orthonormal basis of $\mathbb{R}^{N_v}$
- Rank deficiency: Since $\mathbf{L}\mathbf{1} = \mathbf{0}$, $\mathbf{L}$ is rank deficient
- Spectrum and connectivity: The smallest eigenvalue λ_1 of $\mathbf{L}$ is 0
  - If the second-smallest eigenvalue $\lambda_2 \neq 0$, then G is connected
  - If $\mathbf{L}$ has n zero eigenvalues, G has n connected components

Further intuition

- Since $\mathbf{u}^\top \mathbf{L}\mathbf{u} = \sum_{(i,j)\in\mathcal{E}}(u_i - u_j)^2$, the minimum-cut formulation is

$$\hat{\mathbf{u}} = \arg\min_{\mathbf{u}\in\{\pm 1\}^{N_v}} \sum_{(i,j)\in\mathcal{E}} (u_i - u_j)^2, \quad \text{s. to } \mathbf{1}^\top \mathbf{u} = N_1 - N_2$$

- **Q:** Does this equivalent cost function make sense? **A:** Absolutely!
  - $\Rightarrow$ Edges joining vertices in the same group do not add to the sum
  - $\Rightarrow$ Edges joining vertices in different groups add 4 to the sum



- Minimize cut: assign values u_i to nodes i such that few edges cross 0

Minimum-cut relaxation

- Relax the constraint $\mathbf{u} \in \{\pm 1\}^{N_v}$ to $\mathbf{u} \in \mathbb{R}^{N_v}$, $\|\mathbf{u}\|_2 = 1$

$$\hat{\mathbf{u}} = \arg\min_{\mathbf{u}} \mathbf{u}^\top \mathbf{L} \mathbf{u}, \quad \text{s. to } \mathbf{1}^\top \mathbf{u} = N_1 - N_2 \text{ and } \mathbf{u}^\top \mathbf{u} = 1$$

$\Rightarrow$ Straightforward to solve using Lagrange multipliers

- Characterization of the solution $\hat{\mathbf{u}}$ [Fiedler '73]:

$$\hat{\mathbf{u}} = \mathbf{v}_2 + \frac{N_1 - N_2}{N_v}\mathbf{1}$$

$\Rightarrow$ The 'second-smallest' eigenvector $\mathbf{v}_2$ of $\mathbf{L}$ satisfies $\mathbf{1}^\top \mathbf{v}_2 = 0$

$\Rightarrow$ Minimum cut is $C(\hat{\mathbf{u}}) = \hat{\mathbf{u}}^\top \mathbf{L} \hat{\mathbf{u}} = \mathbf{v}_2^\top \mathbf{L} \mathbf{v}_2 \propto \lambda_2$

- If the graph G is disconnected then we know $\lambda_2 = 0 = C(\hat{\mathbf{u}})$

$\Rightarrow$ If G is amenable to bisection, the cut is small and so is λ_2

Spectral graph bisection

- **Q:** How to obtain the binary cluster labels $\mathbf{u} \in \{\pm 1\}^{N_v}$ from $\hat{\mathbf{u}} \in \mathbb{R}^{N_v}$?
  $\Rightarrow$ Maximize the similarity measure $\mathbf{u}^\top \hat{\mathbf{u}}$

$$u_i = f(\mathbf{v}_2) := \begin{cases} +1, & [\mathbf{v}_2]_i \text{ among the } N_1 \text{ largest entries of } \mathbf{v}_2 \\ -1, & \text{otherwise} \end{cases}$$

- Spectral graph bisection algorithm

  S1: Compute Laplacian matrix $\mathbf{L}$ with entries $L_{ij} = D_{ij} - A_{ij}$

  S2: Find 'second smallest' eigenvector $\mathbf{v}_2$ of $\mathbf{L}$

  S3: Candidate membership of vertex i is $\bar{u}_i = f([\mathbf{v}_2])$ (or $\underline{u}_i = f([-\mathbf{v}_2])$)

  S4: Among $\bar{\mathbf{u}}$ and $\underline{\mathbf{u}}$ pick the one that minimizes $C(\mathbf{u})$

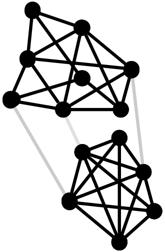
- Nomenclature: $\mathbf{v}_2$ is known as the Fiedler vector
  $\Rightarrow$ Eigenvalue λ_2 is Fiedler value, or algebraic connectivity of G

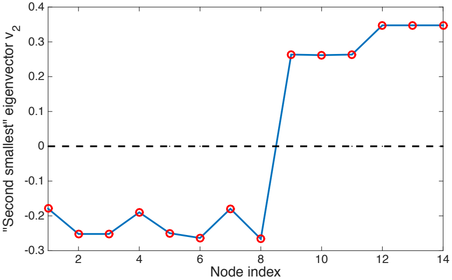
Spectral gap in Fiedler vector entries

- Suppose G is disconnected and has two connected components
  - $\mathbf{L}$ is block diagonal, two smallest eigenvectors indicate groups, i.e.,

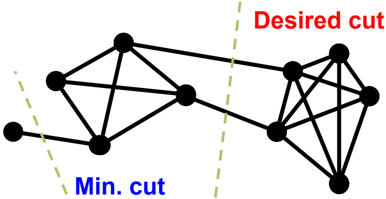
$$\mathbf{v}_1 = [1, 1, \ldots, 1, 0, \ldots, 0]^\top \text{ and } \mathbf{v}_2 = [0, 0, \ldots, 0, 1, \ldots, 1]^\top$$

- If G is connected but amenable to bisection, $\mathbf{v}_1 = \mathbf{1}$ and $\lambda_2 \approx 0$
  - Also, $\mathbf{1}^\top \mathbf{v}_2 = \sum_i [\mathbf{v}_2]_i = 0 \;\Rightarrow$ Positive and negative entries in $\mathbf{v}_2$





- Consider the graph bisection problem with unknown group sizes
  - $\Rightarrow$ Minimizing the graph cut may be no longer meaningful!



  - $\Rightarrow$ Cost $C := \sum_{i \in \mathcal{V}_1, j \in \mathcal{V}_2} A_{ij}$ agnostic to groups' internal structure
- Better criterion is the ratio cut R defined as

$$R := \frac{C}{|\mathcal{V}_1|} + \frac{C}{|\mathcal{V}_2|}$$

  - $\Rightarrow$ Balanced partitions: small community is penalized by the cost

Ratio-cut minimization

- Fix a bisection U of G into groups $\mathcal{V}_1$ and $\mathcal{V}_2$
- Define $\mathbf{f}$: $\mathbf{f}(U) = [f_1, \ldots, f_{N_v}]^\top \in \mathbb{R}^{N_v}$ with entries

$$f_i = \begin{cases} \sqrt{\frac{|\mathcal{V}_2|}{|\mathcal{V}_1|}}, & \text{vertex } i \text{ belongs to } \mathcal{V}_1 \\ -\sqrt{\frac{|\mathcal{V}_1|}{|\mathcal{V}_2|}}, & \text{vertex } i \text{ belongs to } \mathcal{V}_2 \end{cases}$$

- One can establish the following properties:

  P1: $\mathbf{f}^\top \mathbf{L} \mathbf{f} = N_v R(U)$;
  P2: $\sum_i f_i = 0$, i.e., $\mathbf{1}^\top \mathbf{f} = 0$; and
  P3: $\|\mathbf{f}\|^2 = N_v$

- From P1-P3 it follows that ratio-cut minimization is equivalent to

$$\min_{\mathbf{f}} \mathbf{f}^\top \mathbf{L} \mathbf{f}, \quad \text{s. to } \mathbf{1}^\top \mathbf{f} = 0 \text{ and } \mathbf{f}^\top \mathbf{f} = N_v$$

Ratio cut and spectral graph bisection

- Ratio-cut minimization is also NP-hard. Relax to obtain

$$\hat{\mathbf{u}} = \arg\min_{\mathbf{u}\in\mathbb{R}^{N_v}} \mathbf{u}^\top \mathbf{L}\mathbf{u}, \quad \text{s. to } \mathbf{1}^\top \mathbf{u} = 0 \text{ and } \mathbf{u}^\top \mathbf{u} = N_v$$

- Partition $\hat{U}$ also given by the spectral graph bisection algorithm

  S1: Compute Laplacian matrix $\mathbf{L}$ with entries $L_{ij} = D_{ij} - A_{ij}$

  S2: Find 'second smallest' eigenvector $\mathbf{v}_2$ of $\mathbf{L}$

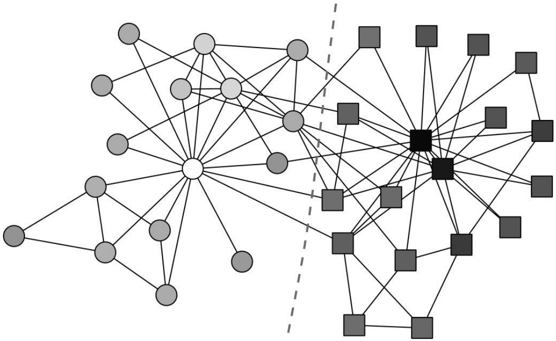
  S3: Cluster membership of vertex i is $u_i = \text{sign}([\mathbf{v}_2]_i)$

- Alternative criterion is the normalized cut NC defined as

$$NC = \frac{C}{vol(\mathcal{V}_1)} + \frac{C}{vol(\mathcal{V}_2)}, \quad vol(\mathcal{V}_i) := \sum_{v\in V_i} d_v, \ i = 1, 2$$

$\Rightarrow$ Corresponds to using the normalized Laplacian $\mathbf{D}^{-1}\mathbf{L}$

Example: Zachary's karate club



- Spectral ratio cut minimization
  - Shapes of vertices indicate community membership
  - Dotted line indicates partition found by the algorithm
  - Vertex colors indicate the strength of their membership

Beyond two communities

- **Q:** What about detecting $K > 2$ communities?
- The ratio cut of a K-way partition U in groups $\{\mathcal{V}_i\}_{i=1}^{K}$ is

$$R(U) := \sum_{i=1}^{K} \frac{C(\mathcal{V}_i, \mathcal{V}_i^c)}{|\mathcal{V}_i|}$$

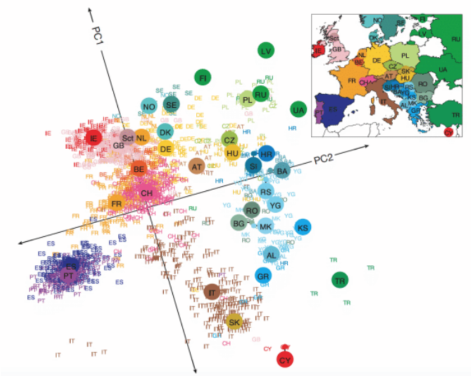
- Relaxed ratio-cut minimization problem formulated as

$$\hat{\mathbf{U}} = \arg \min_{\mathbf{U} \in \mathbb{R}^{N_v \times K}} \text{trace}(\mathbf{U}^\top \mathbf{L} \mathbf{U}), \quad \text{s. to } \mathbf{U}^\top \mathbf{U} = \mathbf{I}$$

- Partition $\hat{U}$ given by the spectral clustering algorithm

  S1: Compute Laplacian matrix $\mathbf{L}$ with entries $L_{ij} = D_{ij} - A_{ij}$
  S2: Find 'K smallest' eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_K$ of $\mathbf{L}$
  S3: Set $\hat{\mathbf{U}} = [\mathbf{v}_1, \ldots, \mathbf{v}_K]$, embedding of node i is row $\hat{\mathbf{u}}_i^\top \in \mathbb{R}^{1 \times K}$
  S4: Assign to clusters via K-means on node embeddings

- Two-dimensional embedding of ‘gene similarity’ matrix
  - ⇒ Consistent with origins of individuals in European map



J. Novembre, “Genes mirror geography within Europe,” *Nature*, 2008

Where do we go from here?

- **Q:** Why does spectral graph partitioning work? **A:** Note that

$$\text{trace}(\hat{\mathbf{U}}^\top \mathbf{L} \hat{\mathbf{U}}) = \sum_{(i,j)\in\mathcal{E}} A_{ij} \|\hat{\mathbf{u}}_i^\top - \hat{\mathbf{u}}_j^\top\|^2$$

  $\Rightarrow$ Embeddings close in $\mathbb{R}^K$ if i, j well connected in G

  $\Rightarrow$ Also known as Laplacian eigenmaps [Belkin-Niyogi'01]

- **Key:** encode graph structure into low-dimensional embeddings


Machine Learning on Graphs:
A Model and Comprehensive Taxonomy

Ines Chami, Sami Abu-El-Haija, Bryan Perozzi, Christopher Ré, and Kevin Murphy

May 11, 2020

Abstract

1 Introduction

arXiv:2005.03675v1 [cs.LG] 7 May 2020


Representation Learning on Graphs: Methods and Applications

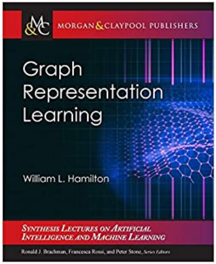
William L. Hamilton, Rex Ying, Jure Leskovec

Department of Computer Science
Stanford University
Stanford, CA, 94305

Abstract

1 Introduction

arXiv:1709.05584v3 [cs.SI] 10 Apr 2018



Roadmap

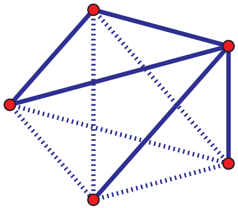
Introductions, context and motivation

Graph signal processing

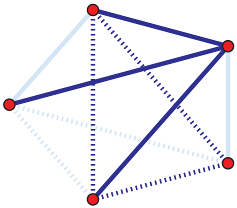
Semi-supervised node classification

Network community detection

Link prediction



Original graph



Link prediction

- Suppose we observe vertex attributes $\mathbf{x} = [x_1, \ldots, x_{N_v}]^\top$; and
- Edge status only observed for subset of pairs $\mathcal{V}^{(2)}_{obs} \subset \mathcal{V}^{(2)} = \mathcal{V} \times \mathcal{V}$
- **Goal:** predict edge status for all other pairs, i.e., $\mathcal{V}^{(2)}_{miss} = \mathcal{V}^{(2)} \setminus \mathcal{V}^{(2)}_{obs}$

Problem statement

- Let $G(\mathcal{V}, \mathcal{E})$ be a random graph, with adjacency matrix $\mathbf{A} \in \{0,1\}^{N_v \times N_v}$

  $\Rightarrow$ $\mathbf{A}^{obs}$ and $\mathbf{A}^{miss}$ denote entries in $\mathcal{V}^{(2)}_{obs}$ and $\mathcal{V}^{(2)}_{miss}$

Link prediction

Predict entries in $\mathbf{A}^{miss}$, given observations $\mathbf{A}^{obs} = \mathbf{a}^{obs}$ and possibly various vertex attributes $\mathbf{X} = \mathbf{x} \in \mathbb{R}^{N_v}$

- Edge status information may be missing due to:

  $\Rightarrow$ Difficulty in observation, issues of sampling

  $\Rightarrow$ Edge is not yet present, wish to predict future status

- Given a model for $\mathbf{X}$ and $(\mathbf{A}^{obs}, \mathbf{A}^{miss})$, jointly predict $\mathbf{A}^{miss}$ based on

$$\mathrm{P}\left[\mathbf{A}^{miss} \,\middle|\, \mathbf{A}^{obs} = \mathbf{a}^{obs}, \mathbf{X} = \mathbf{x}\right]$$

$\Rightarrow$ More manageable to predict the variables A^{miss}_{ij} individually

Informal scoring methods

- Idea: compute score $s(i,j)$ for missing 'potential edges' $\{i,j\} \in \mathcal{V}_{miss}^{(2)}$
  $\Rightarrow$ Predicted edges returned by retaining the top n^* scores
- Scores designed to assess certain local structural properties of G^{obs}
  $\Rightarrow$ Distance-based, inspired by the small-world principle

$$s(i,j) = -\text{dist}_{G^{obs}}(i,j)$$

  $\Rightarrow$ Neighborhood-based, e.g., the number of common neighbors

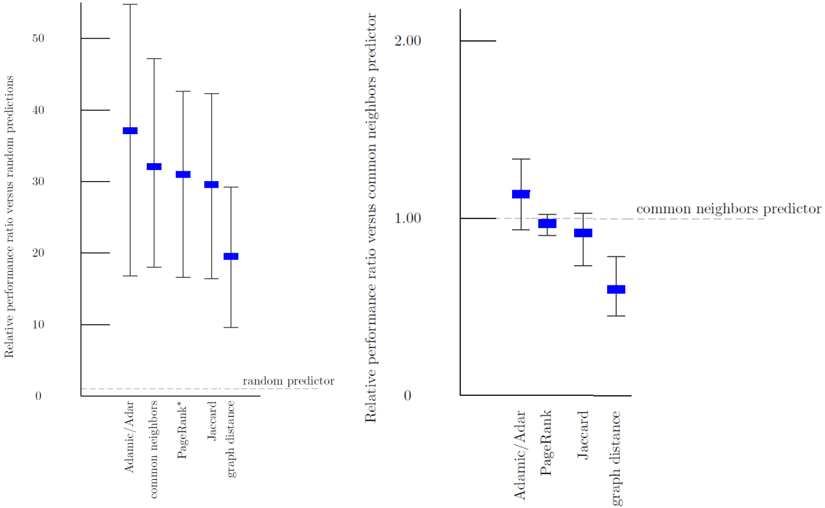
$$s(i,j) = |\mathcal{N}_i^{obs} \cap \mathcal{N}_j^{obs}| \quad \text{or} \quad s(i,j) = \frac{|\mathcal{N}_i^{obs} \cap \mathcal{N}_j^{obs}|}{|\mathcal{N}_i^{obs} \cup \mathcal{N}_j^{obs}|}$$

  $\Rightarrow$ Favor loosely-connected common neighbors [Adamic-Adar'03]

$$s(i,j) = \sum_{k \in \mathcal{N}_i^{obs} \cap \mathcal{N}_j^{obs}} \frac{1}{\log |\mathcal{N}_k^{obs}|}$$

Tests on co-authorship networks

- Results from a link prediction study in [Liben Nowell-Kleinberg'03]



Classification methods

- Idea: use training data $\mathbf{a}^{obs}$ and $\mathbf{x}$ to build a binary classifier
  $\Rightarrow$ Classifier is in turn used to predict the entries in $\mathbf{A}^{miss}$
- Logistic regression classifiers most popular, based on the model

$$\log\left[\frac{\mathrm{P}_\beta(A_{ij}=1 \,|\, \mathbf{Z}_{ij}=\mathbf{z})}{\mathrm{P}_\beta(A_{ij}=0 \,|\, \mathbf{Z}_{ij}=\mathbf{z})}\right]=\boldsymbol{\beta}^\top\mathbf{z}, \quad \text{where}$$

(i) $\boldsymbol{\beta}\in\mathbb{R}^K$ is a vector of regression coefficients; and

(ii) $\mathbf{Z}_{ij}$ is a vector of explanatory variables indexed by $\{i,j\}$

$$\mathbf{Z}_{ij}=[g_1(\mathbf{A}^{obs}_{(-ij)},\mathbf{X}),\ldots,g_K(\mathbf{A}^{obs}_{(-ij)},\mathbf{X})]^\top$$

- Functions $g_k(\cdot)$ encode useful predictive information in $\mathbf{a}^{obs}_{(-ij)}$ and $\mathbf{x}$
  Ex: vertex attributes, score functions, network statistics

Logistic regression classifier

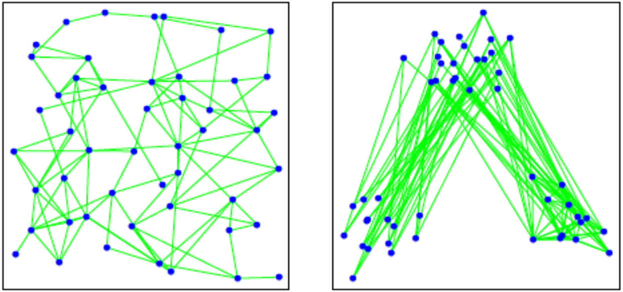
- Train: Obtain MLE $\hat{\boldsymbol{\beta}}$ via iteratively-reweighted LS
- Test: Potential edges (i, j) declared present based on probabilities

$$\mathrm{P}_{\hat{\beta}}(A_{ij} = 1 \,|\, \mathbf{Z}_{ij} = \mathbf{z}) = \frac{\exp\left(\hat{\boldsymbol{\beta}}^\top \mathbf{z}\right)}{1 + \exp\left(\hat{\boldsymbol{\beta}}^\top \mathbf{z}\right)}$$

- Logistic regression assumes $\mathbf{A}_{ij}$ conditionally independent given $\mathbf{z}$
  - $\Rightarrow$ Seldom the case with relational network data
- Underlying mechanism of data missingness is important
  - $\Rightarrow$ Classification for link prediction reminiscent of cross-validation
  - $\Rightarrow$ Assumption that data are missing at random is fundamental

Latent variable models

- In addition to a lineal predictor $\beta^\top \mathbf{z}$, latent models describe A_{ij}
  $\Rightarrow$ As a function of vertex-specific latent variables $\mathbf{u}_i$ and $\mathbf{u}_j$



Homophily — Stochastic equivalence

- Latent models are flexible to capture underlying social mechanisms
  Ex: homophily (transitivity) and stochastic equivalence (groups)

Latent class and distance models

- Latent distance model: node i has unobserved position $\mathbf{U}_i \in \mathbb{R}^d$
  - Positions $\mathbf{U}_i$ in latent space assumed i.i.d. e.g., Gaussian distributed
  - Model cond. probability of edge A_{ij} as function of $\boldsymbol{\beta}^\top \mathbf{z} - \|\mathbf{u}_i - \mathbf{u}_j\|_2$
  - Homophily: Nearby nodes in latent space more likely to link
- Latent class model: node i belongs to unobserved class $U_i \in \{1, \ldots, k\}$
  - Classes U_i assumed i.i.d. e.g., multinomial distributed
  - Model cond. probability of edge A_{ij} as function of $\boldsymbol{\beta}^\top \mathbf{z} - \theta_{u_i, u_j}$
  - Stochastic equivalence: Nodes in same class equally likely to link

P. D. Hoff, "Modeling homophily and stochastic equivalence in symmetric relational data," *NIPS*, 2008

Logistic regression with latent variables

- Let $\mathbf{M} \in \mathbb{R}^{N_v \times N_v}$ be an unknown, random, and symmetric matrix

$$\mathbf{M} = \mathbf{U}^\top \mathbf{\Lambda} \mathbf{U} + \mathbf{E}, \quad \text{where}$$

(i) $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_{N_v}]$ is a random orthonormal matrix of latent variables;
(ii) $\mathbf{\Lambda}$ is a random diagonal matrix; and
(iii) $\mathbf{E}$ is a symmetric matrix of i.i.d. noise entries ϵ_{ij}

- Latent eigenmodel subsumes the class and distance variants [Hoff'08]
  $\Rightarrow$ Notice that $M_{ij} = \mathbf{u}_i^T \mathbf{\Lambda} \mathbf{u}_j + \epsilon_{ij}$

- The logistic regression model with latent variables is

$$\log \left[\frac{\mathrm{P}_\beta(A_{ij} = 1 \,|\, \mathbf{Z}_{ij} = \mathbf{z}, M_{ij} = m)}{\mathrm{P}_\beta(A_{ij} = 0 \,|\, \mathbf{Z}_{ij} = \mathbf{z}, M_{ij} = m)} \right] = \boldsymbol{\beta}^\top \mathbf{z} + m$$

- A_{ij} still assumed conditionally independent given $\mathbf{Z}_{ij}$ and M_{ij}
  $\Rightarrow$ But they are conditionally dependent given only $\mathbf{Z}_{ij}$

Bayesian link prediction

- Specify distributions for $\mathbf{U}, \mathbf{\Lambda}, \mathbf{E}$ to make statistical link predictions
  - Bayesian inference natural $\Rightarrow$ Specify a prior for $\boldsymbol{\beta}$ as well
- To predict those entries in $\mathbf{A}^{miss}$, threshold the posterior mean

$$\mathbb{E}\left[\frac{\exp\left(\boldsymbol{\beta}^\top \mathbf{Z}_{ij} + M_{ij}\right)}{1+\exp\left(\boldsymbol{\beta}^\top \mathbf{Z}_{ij} + M_{ij}\right)} \,\Big|\, \mathbf{A}^{obs} = \mathbf{a}^{obs}, \mathbf{Z}_{ij} = \mathbf{z}\right]$$

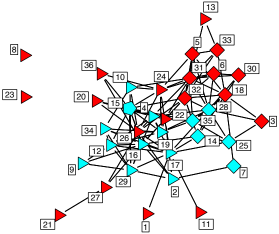
- Use MCMC algorithms to approximate the posterior distribution
  - Gaussian distributions attractive for their conjugacy properties
- Higher complexity than MLE for standard logistic regression

  $\Rightarrow$ Need to generate draws for N_v^2 unobserved variables $\{U_{ij}\}$

  $\Rightarrow$ Major cost reduction with reduced $\text{rank}(\mathbf{U}) = k \ll N_v$ models

Example: predicting lawyer collaborations

- Network G^{obs} of working relationships among lawyers [Lazega'01]
  - Nodes are $N_v = 36$ partners, edges indicate partners worked together



- Data includes various node-level attributes:
  - Seniority (node labels indicate rank ordering)
  - Office location (triangle, square or pentagon)
  - Type of practice, i.e., litigation (red) and corporate (cyan)
  - Gender (three partners are female labeled 27, 29 and 34)
- Goal: predict cooperation among social actors in an organization

Methods tested

- Define the following set of explanatory variables:

$$Z_{ij}^{(1)} = \text{seniority}_i + \text{seniority}_j, \quad Z_{ij}^{(2)} = \text{practice}_i + \text{practice}_j$$

$$Z_{ij}^{(3)} = \mathbb{I}\left\{\text{practice}_i = \text{practice}_j\right\}, \quad Z_{ij}^{(4)} = \mathbb{I}\left\{\text{gender}_i = \text{gender}_j\right\}$$

$$Z_{ij}^{(5)} = \mathbb{I}\left\{\text{office}_i = \text{office}_j\right\}, \quad Z_{ij}^{(6)} = |\mathcal{N}_i^{obs} \cap \mathcal{N}_j^{obs}|$$

Method 1: standard logistic regression with $Z_{ij}^{(1)}, \ldots, Z_{ij}^{(5)}$

Method 2: standard logistic regression with $Z_{ij}^{(1)}, \ldots, Z_{ij}^{(6)}$

Method 3 informal scoring method with $s(i,j) = Z_{ij}^{(6)}$

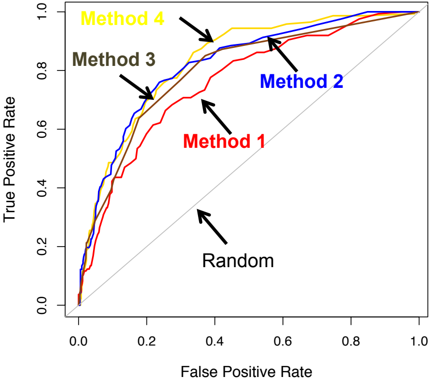
Method 4: logistic regression with $Z_{ij}^{(1)}, \ldots, Z_{ij}^{(5)}$ and latent eigenmodel

- Five-fold cross-validation over the set of $36(36-1)/2 = 630$ vertex pairs

  $\Rightarrow$ For each fold, $630/5 = 126$ pairs in $\mathbf{A}^{miss}$ and the rest in $\mathbf{A}^{obs}$

Receiver operating characteristic

- Receiver operating characteristic curves show predictive performance

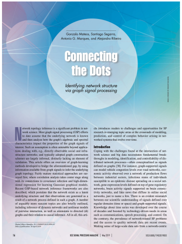


- Method 1 performs worst ⇒ Agnostic to network structure
- Informal Method 3 yields slightly worst performance than 2 and 4

Where do we go from here?

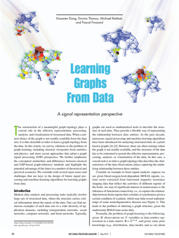
- Got our first glimpse onto statistical models for network data
- Network-based versions of canonical statistical models
  - $\Rightarrow$ Regression models - Exponential random graph models (ERGMs)
  - $\Rightarrow$ Latent variable models - Stochastic block models and graphons
- Link prediction an instance of network topology inference problems

  Q: If G (or a portion thereof) is unobserved, can we infer it from data?


Gonzalo Mateos, Santiago Segarra, Antonio G. Marques, and Alejandro Ribeiro

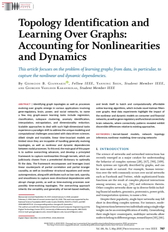
Connecting the Dots

Identifying network structure via graph signal processing


Xiaowen Dong, Dorina Thanou, Michael Rabbat, and Pascal Frossard

Learning Graphs From Data

A signal representation perspective


Topology Identification and Learning Over Graphs: Accounting for Nonlinearities and Dynamics

This article focuses on the problem of learning graphs from data, in particular, to capture the nonlinear and dynamic dependencies.

By Georgios B. Giannakis, Fellow IEEE, Yanning Shen, Student Member IEEE, and Georgios Vasileios Karanikolas, Student Member IEEE

Glossary

- Networks and graphs
- Network data science
- Machine learning on graphs
- Graph signal processing
- Graph Fourier transform
- Laplacian
- Convolution
- Graph neural networks
- Semi-supervised learning
- Nearest-neighbor prediction
- Signal smoothness
- Graph regularization
- Community detection
- Graph cut
- Spectral clustering
- Node embedding
- Graph representation learning
- Link prediction
- Logistic regression
- Latent variable models
- Bayesian inference
- Stochastic block models
- Graphons
- Network topology inference