Linear Algebra in Data Exploration

Daniel L Boley
University of Minnesota

- Graph properties via linear algebra.
- Analysis of graph-organized data via methods based on linear algebra
- Explore together fast algorithms.

Outline

- Organization and Administrative
- Linear Algebra Basics
  - Linear Equations, matrix inversion
  - Least Squares Approximation
  - Symmetric Eigenvalue Problem
  - Singular Value Decomposition (SVD)
  - Application of SVD
  - Non-symmetric matrix eigenvalue: Perron Frobenius
- Graph Analysis
  - Clustering and Graph Partitioning
  - Principal Direction Divisive Partitioning
  - Spectral Partitioning
  - Importance Measures
  - Network Link Analysis: Pagerank, HITS

Outline

- Organization and Administrative
- Linear Algebra Basics
  - Linear Equations, matrix inversion
  - Least Squares Approximation
  - Symmetric Eigenvalue Problem
  - Singular Value Decomposition (SVD)
  - Application of SVD
  - Non-symmetric matrix eigenvalue: Perron Frobenius
- Graph Analysis
  - Clustering and Graph Partitioning
  - Principal Direction Divisive Partitioning
  - Spectral Partitioning
  - Importance Measures
  - Network Link Analysis: Pagerank, HITS

Linear Algebra in Data Exploration

- Many large unstructured data sets are represented as tables and tables of numbers or connections.
  - Text documents (news, laws, WWW documents).
  - Gene expression profiles
  - Attributes for individual people, transactions, locations, ecosystems, ...

  } tabular

  - Gene-gene or protein-protein interaction networks
  - WWW connectivity graph
  - Computer inter-connect in Internet
  - People-people affinities in Social Media

  } graph

- Many example datasets can easily have up to $O(10^{9+})$ data points.
- Many datasets have much noise or many attributes.
- Many example datasets are sampled, subject to sampling bias.

Tools to Explore

- Dimensionality Reduction, Clustering.
  - Represent each data sample with a reduced set of attribute values
  - Minimize loss of information
  - Implicit assumption: data is subject to some level of noise.
- Graph Properties
  - partitioning
  - identify important nodes or links
  - aggregrate properties
  - model propagation of influence/information
- Graph Tools (to explore)
  - fast computation of matrix problems to obtain graph properties
  - graph neural networks – generalize convolutional NNs
  - random sampling: sketching
  - model propagation of influence/information

Class Info

Pointers

- Instructor: Daniel Boley, office 6-209, e-mail boley@cs.umn.edu.
- Class meets in AmundH 116 on MW 4-5:15pm.
- Web page: `http://www-users.cselabs.umn.edu/classes/Fall-2022/csci8363/`

Organization

- 1-3 introductory lecture reviewing linear algebra and some optimization.
- 21-23 lectures: Student presentation of a scheduled paper:

Web Site

- Semester Plan
  - the schedule of papers to be presented.
  - alternative papers (you can use one of these instead)
  - most papers are research papers. Long papers need a "highlight" talk.
- Canvas
  - All items should be submitted through Canvas.
  - Each week: short synopsis, including what you consider the take-away message.
  - Each week: feedback to the speaker on their presentation.
  - Early October: submit your proposed term project idea.

Class Plan

- 11-12 weeks of lectures: A student presents a paper:
  - Primary papers are listed on the course plan.
  - First part of course will be on methods for graph analysis based on linear algebra
  - Second part of course will be on methods for graph-orgnaized data (such as graph deep learning and sampling)
  - You can choose a paper from the list, or a paper on the same topic found on your own with instructor approval.
  - Everyone listening submits a short synopsis of what they got out of the lecture (the main take-away message), and separately feedback to be passed back to each speaker.
  - All submissions are submitted through Canvas.

NEED VOLUNTEERS FOR NEXT WEEK!

Term Projects

- Each student carries out an independent project.
  - Small groups are OK with prior approval. The work scope must be commensurate with the size of the group. Each member of the group will be expected to present their piece of the work.
- You choose your project.
  - A project can involve implementation of an algorithm proposed in a paper on a new big data set. You can report whether or not the results are as advertised.
  - A literature review must include some implementation results: either one of the methods in the literature review on new data or a new variant of one of the methods on their data.
  - Some sample sources of data are posted, though this list may be out of date.
- Steps
  - Submit a short 1-page project proposal after about 1 month.
  - Give a 10-minute presentation of your project during the last 6 lectures (3 weeks) of the semester (part after Thanksgiving).
  - Submit a report of about 10 pages at the end of the semester.

Outline

- Organization and Administrative
- Linear Algebra Basics
  - Linear Equations, matrix inversion
  - Least Squares Approximation
  - Symmetric Eigenvalue Problem
  - Singular Value Decomposition (SVD)
  - Application of SVD
  - Non-symmetric matrix eigenvalue: Perron Frobenius
- Graph Analysis
  - Clustering and Graph Partitioning
  - Principal Direction Divisive Partitioning
  - Spectral Partitioning
  - Importance Measures
  - Network Link Analysis: Pagerank, HITS

Linear Algebra Basics

- Systems of linear equations $\Longleftrightarrow$ matrix inversion
- Solve $A\mathbf{x} = \mathbf{b}$ for $\mathbf{x}$, or compute product $A^{-1}b$.
  - Basic computational tool in optimization, graph analysis, other matrix problems below
  - Solved via direct methods (Gaussian Elim) $O(n^3) time$.
  - Large sparse cases solved via iterative methods (just multiply by A)
- Basis to find hitting times in random walks.

Least Squares

- Minimize $\|A\mathbf{x} - \mathbf{b}\|_2$ over $\mathbf{x}$.
- at heart of many fitting problems
- low rank approximation: find low rank B minimizing $\|A - B\|$.

Eigenvalue problems

- For symmetric A: give canonical directions, e.g., covariance matrix for a Gaussian
- Can characterize eigenvalues by variational principle
- Plays big role in graph partitioning,
- Non-symmetric eigenvalues related to influence monitoring or node importance measures (e.g., pagerank).

Some Details

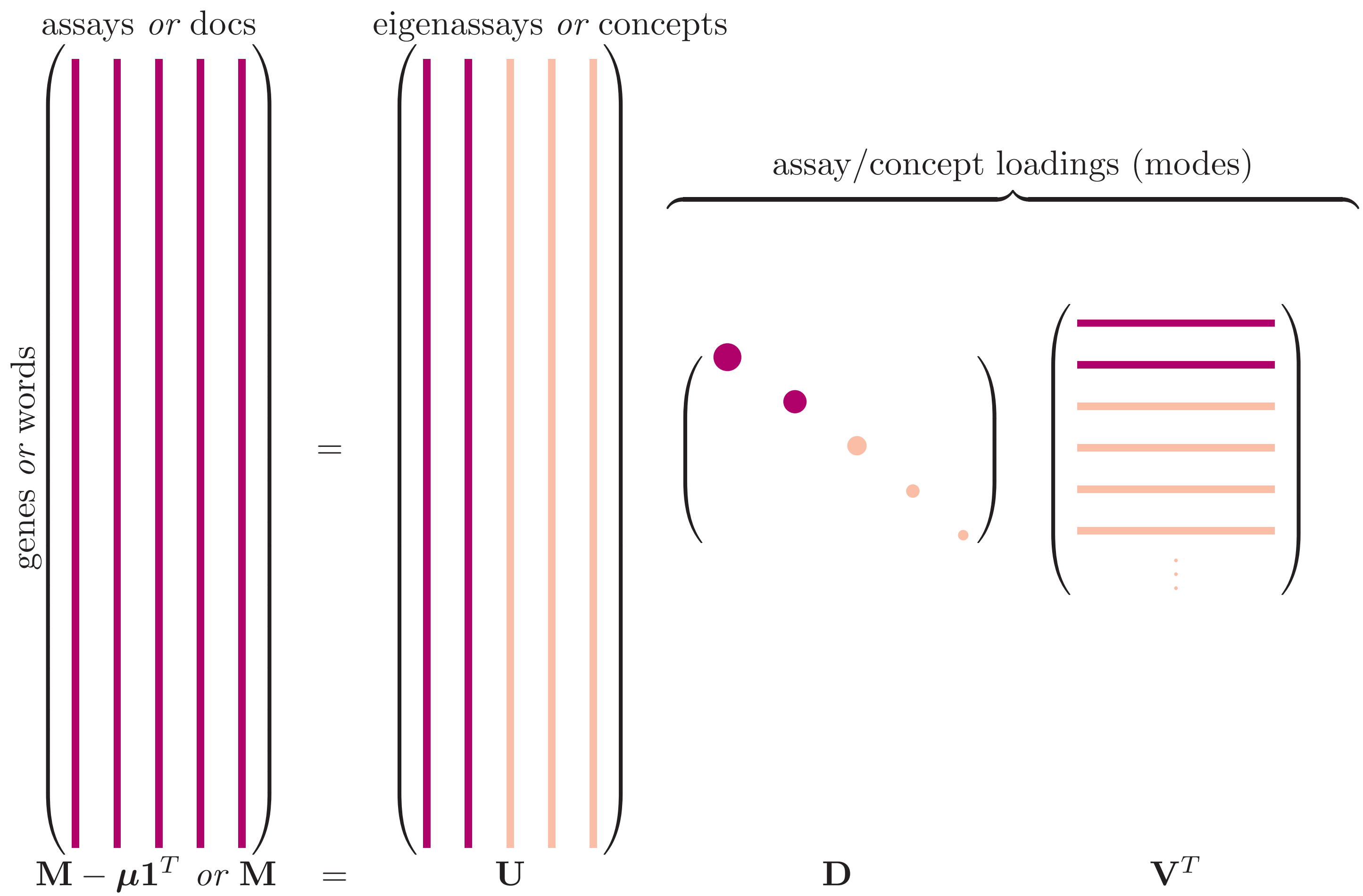
Eigenvalues of Symmetric Matrices

- A^* denotes conjugate transpose of A (plain transpose if real)
- Assume $A = A^*$ (Hermitian), or $A = A^T$ (real symmetric).
- Suppose $A\mathbf{v}_k = \lambda_k \mathbf{v}_k$ with $\|\mathbf{v}_k\|_2 = 1$, for $k = 1, 2, \cdots$.
- Then $\phi(\mathbf{X}) = \mathbf{v}^* A \mathbf{v}$ must be real ($\phi(\mathbf{v}) = \overline{\phi(\mathbf{v})}$).
- $\phi(\mathbf{v}_k) = \lambda_k \mathbf{v}^*\mathbf{v} = \lambda_k^* \mathbf{v}^*\mathbf{v}$ so λ_k must be real
- $\mathbf{v}_j^* A \mathbf{v}_k = \lambda_j \mathbf{v}_j^* \mathbf{v}_k = \lambda_k \mathbf{v}_j^* \mathbf{v}_k$, so when $\lambda_j \neq \lambda_k$, $\mathbf{v}_j^* \mathbf{v}_k$ must be zero.
- Hence we have spectral decomp $A = V \Lambda V^*$ with $V^* V = I$, Λ diagonal.

Variational Principles (Symmetric matrices)

- Assume eigenvalues are ordered: $\lambda_1 \leq \cdots \leq \lambda_n$.
- $\lambda_1 = \min_{\mathbf{x}} \frac{\mathbf{x}^* A \mathbf{x}}{\mathbf{x}^* \mathbf{x}}$.
- $\lambda_2 = \min_{\mathbf{x} \perp \mathbf{x}_1} \frac{\mathbf{x}^* A \mathbf{x}}{\mathbf{x}^* \mathbf{x}}$.
- $\lambda_3 = \min_{\mathbf{x} \perp \mathbf{x}_1, \mathbf{x}_2} \frac{\mathbf{x}^* A \mathbf{x}}{\mathbf{x}^* \mathbf{x}}$.
- $\cdots$
- $\lambda_2 = \max_{\mathcal{S}_{n-1}} \min_{\mathbf{x} \in \mathcal{S}_{n-1}} \frac{\mathbf{x}^* A \mathbf{x}}{\mathbf{x}^* \mathbf{x}}$.

Singular Value Decomposition – SVD



Singular Value Decomposition – SVD

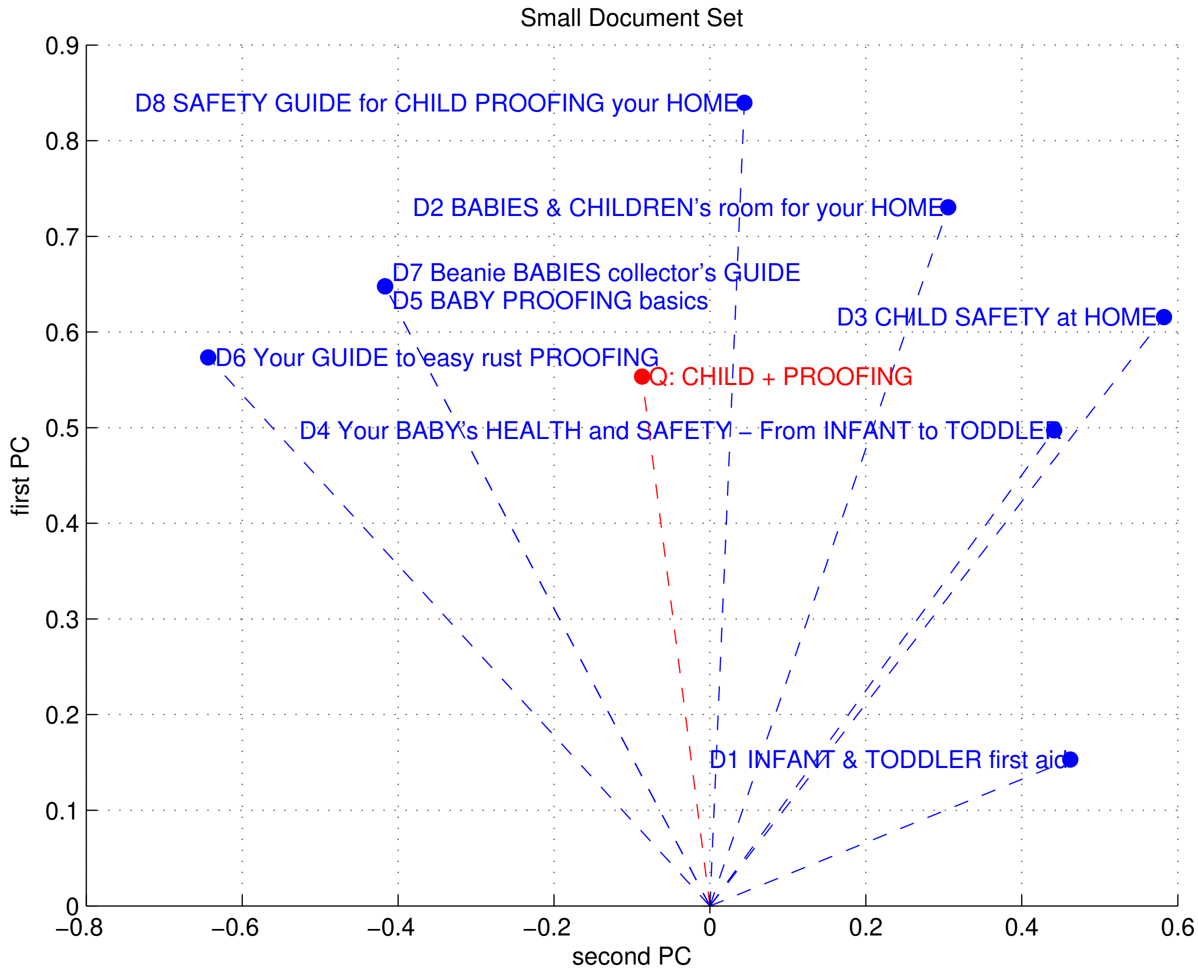
- Eliminate Noise
- Reduce Dimensionality
- Expose Major Components
- Suppose samples are columns of $m \times n$ matrix $\mathbf{M}$.
- Try to find k pseudo-data columns such that all samples can be represented by linear combinations of those k pseudo-data columns.
- Primary criterion: minimize the 2-norm of the discrepancy between the original data and what you can represent using k pseudo-data columns.
- Answer: Singular Value Decomposition of $\mathbf{M}$.
  If centered, get Principal Components of $\mathbf{M}$ (PCA).
- Sometimes, for statistical reasons, want to remove uniform signal:
  - $\mathbf{M} \leftarrow \mathbf{M} - \boldsymbol{\mu}\mathbf{1}^T$,
    where $\boldsymbol{\mu} = \mathbf{M} \cdot \mathbf{1}$.
  - Then $\mathbf{M}^T\mathbf{M}$ is the Sample Covariance Matrix.
  - Even without centering, $\mathbf{M}^T\mathbf{M}$ is a "Gram" matrix.

Text Documents – Data Representation

- Each document represented by n-vector $\mathbf{d}$ of word counts, scaled to unit length.
- Vectors assembled into Term Frequency Matrix $\mathbf{M} = (\, \mathbf{d}_1 \quad \cdots \quad \mathbf{d}_m \,)$.

	D1 INFANT & TODDLER first aid	D2 BABIES & CHILDREN's room for your HOME	D3 CHILD SAFETY at HOME	D4 Your BABY's HEALTH and SAFETY - From INFANT to TODDLER	D5 BABY PROOFING basics	D6 Your GUIDE to easy rust PROOFING	D7 Beanie BABIES collector's GUIDE	D8 SAFETY GUIDE for CHILD PROOFING your HOME
BABY	0	$\sqrt{3}$	0	$\sqrt{5}$	$\sqrt{2}$	0	$\sqrt{2}$	0
CHILD	0	$\sqrt{3}$	$\sqrt{2}$	0	0	0	0	$\sqrt{5}$
GUIDE	0	0	0	0	0	$\sqrt{2}$	$\sqrt{2}$	$\sqrt{5}$
HEALTH	0	0	0	$\sqrt{5}$	0	0	0	0
HOME	0	$\sqrt{3}$	$\sqrt{2}$	0	0	0	0	$\sqrt{5}$
INFANT	$\sqrt{2}$	0	0	$\sqrt{5}$	0	0	0	0
PROOFING	0	0	0	0	$\sqrt{2}$	$\sqrt{2}$	0	$\sqrt{5}$
SAFETY	0	0	$\sqrt{2}$	$\sqrt{5}$	0	0	0	$\sqrt{5}$
TODDLER	$\sqrt{2}$	0	0	$\sqrt{5}$	0	0	0	0

Latent Semantic Indexing – LSI

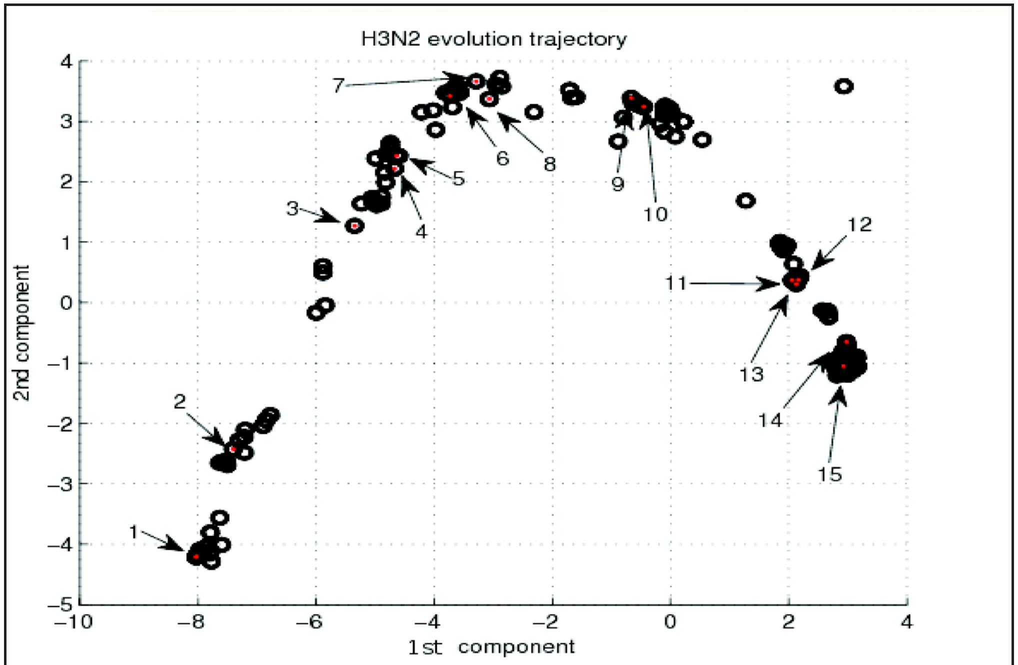


- Stay length-independent: compare using just angles.

NEED VOLUNTEER FOR NEXT WEEK

- Previous material is topic of first paper.
- Material pretty straightforward.

Model Avian Influenza Virus



from Lam&Boley 2011

Number	Vaccine strain
1	A/Aichi/1968
2	A/Port Chalmers/1/1973
3	A/Philippines/2/1982
4	A/leningrad/360/1986
5	A/Shanghai/11/1987
6	A/Beijing/353/1989
7	A/Shangdong/9/1993
8	A/Johannesburg/33/1994
9	A/Sydney/5/1997
10	A/Moscow/10/1999
11	A/Fujian/411/2002
12	A/California/7/2004
13	A/Wisconsin/67/2005
14	A/Brisbane/10/2007
15	A/Perth/16/2009

- Evolution is a flow, naturally falls in chronological order.
- Without vaccine, picture is more a random cloud of points.
- Suggests vaccine use does affect evolution of virus.

Model Avian Influenza Virus

(Lam et al., 2012)

- Avian Flu Virus characterized by the HA protein, which the virus uses to penetrate the cell.
- The protein is described by a string of 566 symbols, each representing one of 20 Amino Acids.
- Embed in high dimensional Euclidean space by replacing each Amino Acid with a string of 20 bits:
  - E.g. 3rd amino acid $= \rightarrow$ 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
- Result is a vector of length $566 \cdot 20 = 11230$.
- Use PCA to reduce dimensions from 11320 to 6.
- Use first 2 components to track evolution of this protein in a simple visual way.

Non-symmetric Eigenvalue Problem

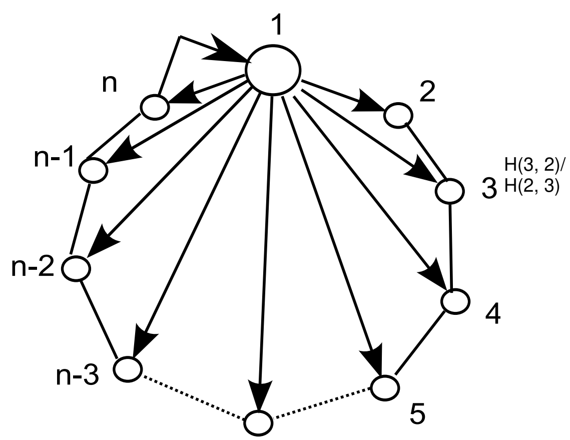
- Applies in random walk (Markov chain) over a graph (even undirected)
- Matrix non always diagonalizable
- Perron-Frobenius theory: A matrix A with all positive entries has a simple eigenvalue $\lambda > 0$ of largest absolute value.
  - For the matrix P of transition probabilities in a random walk, the row sums are 1, $\lambda = 1$, and the eigenvector is the vector of stationary probabilities.
  - A similar result holds for a non-negative matrix A if A^k is all positive for some integer power $k > 0$.

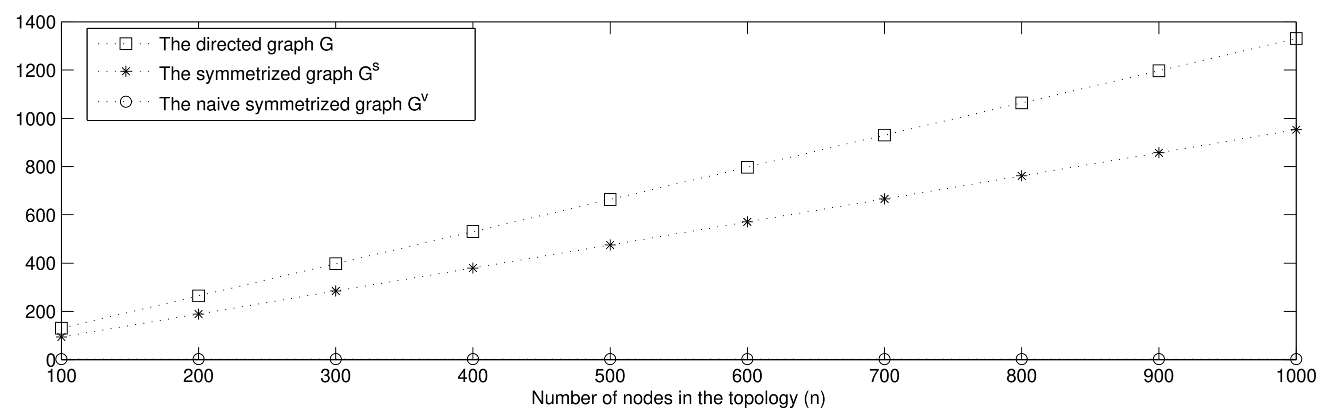
Outline

- Organization and Administrative
- Linear Algebra Basics
  - Linear Equations, matrix inversion
  - Least Squares Approximation
  - Symmetric Eigenvalue Problem
  - Singular Value Decomposition (SVD)
  - Application of SVD
  - Non-symmetric matrix eigenvalue: Perron Frobenius
- Graph Analysis
  - Clustering and Graph Partitioning
  - Principal Direction Divisive Partitioning
  - Spectral Partitioning
  - Importance Measures
  - Network Link Analysis: Pagerank, HITS

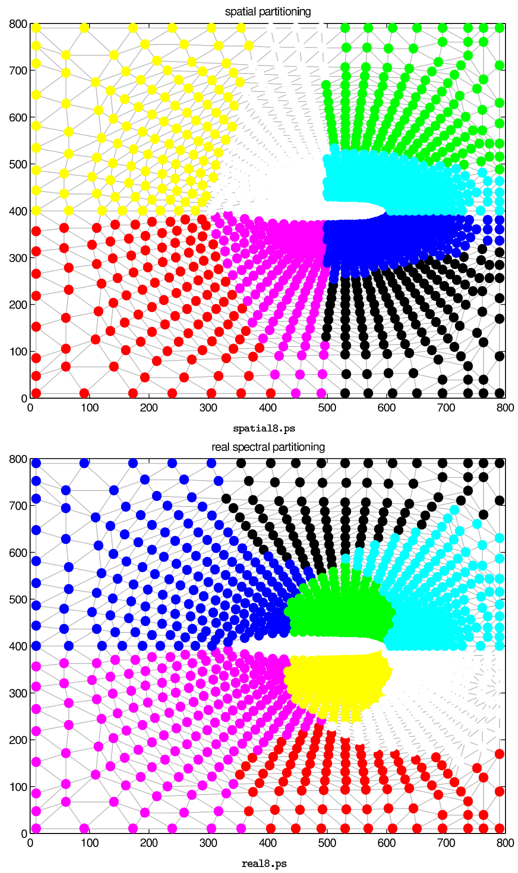
Spectral Methods on Graphs

- Model an undirected graph by a random walk.
- Recurring Node Probability $\Longleftrightarrow$ Pagerank [as in google].
- Average round-trip commute time is a metric-squared
- Can embed graph in Euclidean space preserving distances.
- Principal Direction splitting on embedding is equivalent to two-way Spectral Graph Partitioning.
- Can be extended to directed graphs (e.g., commute times still a metric). (Boley et al., 2011)

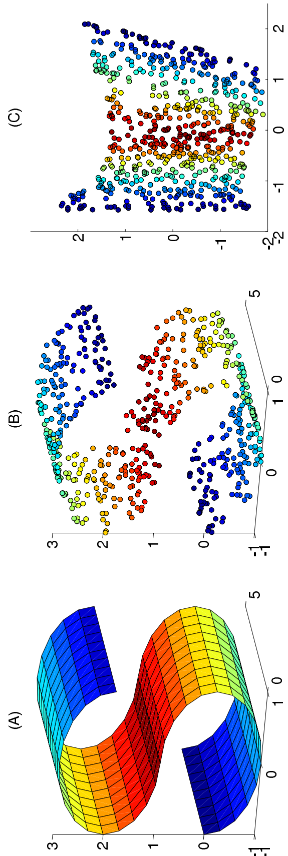




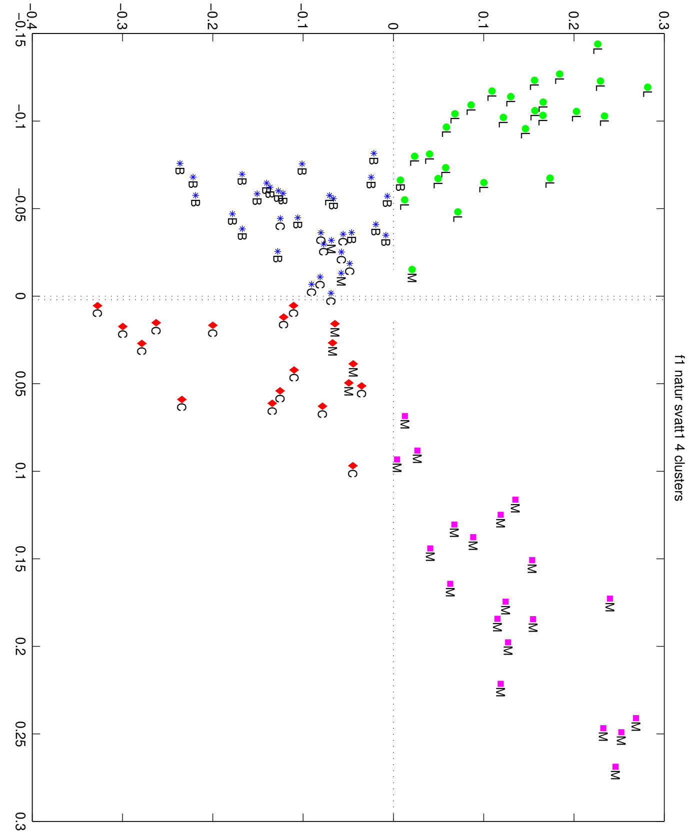
Graph Examples



graph partitioning



LLE



clustering

Graph Analysis: Random Walk Model

- Many properties of a graph can be obtained or estimated from properties of the so-called Fundamental Tensor derived using Random Walk model.
  - average hitting times, commute times.
  - distances or affinities between nodes.
  - betweenness measures.
  - importance/centrality measures.
  - bottlenecks in computer communication networks, road networks.
  - influence propagation.
- Much existing theory is for undirected graphs
- Some can be extended to directed graphs.
- Much of this material is from (Boley et al., 2010; Boley et al., 2018; Golnari et al., 2019).

Undirected vs Directed graphs

Undirected Graph

- social networks:
  friends and contact lists
- passive electrical networks
- recommender systems:
  e.g. bipartite graph:
  users $\leftrightarrow$ movies.
- the internet, computer communication networks.

Directed Graph

- the WWW: random walk on relaxed graph yields pagerank.
- road network with one-way streets.
- wireless device network with mix of high and low-powered devices.
- propagation of influence or trust in social networks.

Graphs-Outline

- Graphs Matrices & Laplacians
- Fundamental Tensor
- Average Hitting and Commute Times (*Directed Graph*)
- Centrality: Embedding in Euclidean Space
- Tensor Applications
- Classic Interpretations of unidrected graphs: Electric Resistances
- Classic: Spectral Graph Partitioning: Cuts
- Classic: Cheeger-like bounds.

Basics: Graphs and Matrices

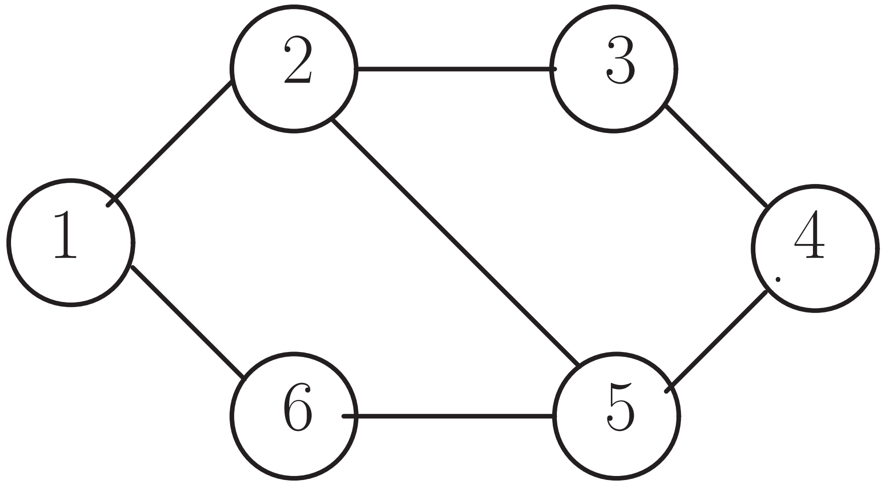
- Graph represented by
  - Adjacency Matrix A s.t. $a_{ij} \neq 0$ when $\exists$ an edge $i \to j$.
  - Markov chain transition matrix P s.t. p_{ij} = probability of transition from node i to node j.
  - Undirected graph $\Longleftrightarrow$ symmetric adjacency matrix $\Longleftrightarrow$ reversible Markov chain.
  - Assume no absorbing states $\Longleftrightarrow$ strongly connected.
- Related Quantities
  - $\mathbf{d} = A \cdot \mathbf{1}$ vector of node (out) degrees,
  - $D = \mathrm{diag}(\mathbf{d})$ = diagonal matrix of degrees,
  - $\boldsymbol{\pi}$ = vector of stationary probabilities, s.t. $\boldsymbol{\pi}^{\mathrm{T}} P = \boldsymbol{\pi}^{\mathrm{T}}$,
  - Π = diagonal matrix of stationary probabilities,
  - $Z = (I - P + \mathbf{1}\boldsymbol{\pi}^{\mathrm{T}})^{-1}$ = Fundamental Matrix (Grinstead & Snell, 2006).

Alternative Laplacians

Laplacians lead to many graph properties (many for undirected graphs)

- $L^{\mathrm{a}} = D - A = D(I - P)$ "combinatorial," based on node degrees.
  - Matrix Tree Theorem $\rightarrow$ number of spanning 'trees' anchored at each node (DiGraphs too) (Brualdi & Ryser, 1991; Chebotarev & Shamis, 2006)
  - smallest graph cut relative to number of nodes in each half (Shi & Malik, 2000; Spielman & Teng, 1996; von Luxburg, 2007).
- $L = \Pi(I - P)$ "Random Walk" $= L^{\mathrm{a}} \cdot$ vol*2 if undirected.
  - pseudo-inverse leads to average commute times/resistances (Doyle & Snell, 1984; Chandra et al., 1989; Klein & Randic, 1993; Boley et al., 2011).
  - pseudo-inverse leads to metric embedding in $\mathbb{R}^n$ (Gower & Legendre, 1986; Fouss et al., 2007).
- $L^{\mathrm{p}} = I - P = I - D^{-1}A = D^{-1}L^{\mathrm{a}}$ "normalized"
  - smallest graph cut relative to number of edges in each half (von Luxburg, 2007).
  - Consensus dynamics over nodes of a graph: $\dot{\mathbf{x}} = -L\mathbf{x}$ (DiGraphs too). (Olfati-Saber et al., 2004, 2006), (Bamieh et al., 2008), (Young et al., 2010, 2011).
- $\mathcal{L} = D^{1/2} L^{\mathrm{p}} D^{-1/2} = D^{-1/2} L^{\mathrm{a}} D^{-1/2}$ = symmetrized normalized Laplacian.
  - shares same eigenvalues as $L^{\mathrm{p}} = I - P$.

Example – Undirected Graph



$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \qquad \mathbf{d} = \begin{pmatrix} 2 \\ 3 \\ 2 \\ 2 \\ 3 \\ 2 \end{pmatrix} \qquad \boldsymbol{\pi} = \frac{1}{14} \cdot \begin{pmatrix} 2 \\ 3 \\ 2 \\ 2 \\ 3 \\ 2 \end{pmatrix}$$

Laplacians

- $L^{a} = \begin{pmatrix} 2 & -1 & 0 & 0 & 0 & -1 \\ -1 & 3 & -1 & 0 & -1 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & -1 & 0 & -1 & 3 & -1 \\ -1 & 0 & 0 & 0 & -1 & 2 \end{pmatrix} = 14 * L$
- Number of spanning 'trees': det$(L^{a}_{[2:6],[2:6]}) = 15$.
- Eigenvalues are 0, 1, 2, 3, 3, 5.
- Eigenvector corresp. to 1 (Fiedler vector): $(1, 0, -1, -1, 0, 1)/2$.
  Used in Spectral Graph Partitioning.
- Volume $=$ number of edges $= {}^{1}\!/_{2}\mathrm{trace}(L^{a}) = 7$.

Fundamental Tensor: Number of Visits

$\boldsymbol{N}(i,j,k)$ for all i, j, k.
$=$ average number of visits to j starting from i before reaching k.

Fundamental Tensor: Number of Visits

- Partition $P = \left[\begin{array}{c|c} P_{11} & \mathbf{p}_{12} \\ \hline \mathbf{p}_{21}^T & p_{nn} \end{array}\right]$.

- If last row replaced with $[\mathbf{0}^T, 1]$, then $[P_{11}^k]_{ij}$ is the probability of being in node j starting in node i at the $k-th$ step, before reaching n.

- $[I + P_{11} + P_{11}^2 + \cdots]_{ij} = [(I - P_{11})^{-1}]_{ij} \stackrel{\text{def}}{=} \mathbf{N}(i,j,n)$
  $= \#$ visits to j starting from i before reaching n.

- $(I - P_{11})^{-1} = [\Pi_{1,\ldots,n-1}^{-1} \underbrace{\Pi_{1,\ldots,n-1}(I - P_{11})}_{L_{11}}]^{-1} = L_{11}^{-1}\Pi_{1,\ldots,n-1}$.

- Since $L \cdot \mathbf{1} = \mathbf{0}$, $\mathbf{1}^T L = \mathbf{0}^T$, can write $(I - P_{11})^{-1}$ in terms of $M \stackrel{\text{def}}{=} L^+$ to yield
  $\mathbf{N}(i,j,n) = (m_{ij} + m_{nn} - m_{in} - m_{nj})\pi_j$ (Lemma ??).

- Choice of destination node n is arbitrary, so have Tensor:
  $\mathbf{N}(i,j,k) = (m_{ij} + m_{kk} - m_{ik} - m_{kj})\pi_j$ for all i, j, k.
  $=$ average number of visits to j starting from i before reaching k.

Tensor

$\boldsymbol{N} =$

```
k=1
 ( 0    0    0    0    0    0   )
 ( 0    2.20 1.33 1.20 1.60 0.53)
 ( 0    2.00 2.67 2.00 2.00 0.67)
 ( 0    1.80 2.00 2.80 2.40 0.80)
 ( 0    1.60 1.33 1.60 2.80 0.93)
 ( 0    0.80 0.67 0.80 1.40 1.47)

k=2
 (1.47  0    0.13 0.27 0.60 0.93)
 ( 0    0    0    0    0    0   )
 (0.13  0    1.47 0.93 0.60 0.27)
 (0.27  0    0.93 1.87 1.20 0.53)
 (0.40  0    0.40 0.80 1.80 0.80)
 (0.93  0    0.27 0.53 1.20 1.87)

k=3
 (2.67 2.00  0   0.67 2.00 2.00)
 (1.33 2.20  0   0.53 1.60 1.20)
 ( 0    0    0    0    0    0   )
 (0.67 0.80  0   1.47 1.40 0.80)
 (1.33 1.60  0   0.93 2.80 1.60)
 (2.00 1.80  0   0.80 2.40 2.80)

k=4
 (2.80 2.40 0.80  0   1.80 2.00)
 (1.60 2.80 0.93  0   1.60 1.33)
 (0.80 1.40 1.47  0   0.80 0.67)
 ( 0    0    0    0    0    0   )
 (1.20 1.60 0.53  0   2.20 1.33)
 (2.00 2.00 0.67  0   2.00 2.67)

k=5
 (1.87 1.20 0.53 0.27  0   0.93)
 (0.80 1.80 0.80 0.40  0   0.40)
 (0.53 1.20 1.87 0.93  0   0.27)
 (0.27 0.60 0.93 1.47  0   0.13)
 ( 0    0    0    0    0    0   )
 (0.93 0.60 0.27 0.13  0   1.47)

k=6
 (1.47 1.40 0.80 0.67 0.80  0   )
 (0.93 2.80 1.60 1.33 1.60  0   )
 (0.80 2.40 2.80 2.00 1.80  0   )
 (0.67 2.00 2.00 2.67 2.00  0   )
 (0.53 1.60 1.20 1.33 2.20  0   )
 ( 0    0    0    0    0    0   )
```

Get Pseudo-Inv of Laplacian

1. Compute normalized Laplacian $L = I - P$.
2. Compute inverse of the upper $(n-1) \times (n-1)$ part: $I - P_{11}$
3. Solve for the stationary probabilites: $(\pi_1, \ldots, \pi_{n-1}) = -(L^{\mathrm{p}}_{11})^{-1} \ell^{\mathrm{p}}_{12} \pi_n$;
4. Form random walk Laplacian $\mathbf{L} = \textsc{Diag}(\boldsymbol{\pi}) \cdot L = \mathbf{\Pi}(I - P)$.
5. Compute the inverse of $\mathbf{L}_{11}^{-1} = (I - P_{11})^{-1} \mathbf{\Pi}_1^{-1}$
6. Compute desired pseudoinverse $\mathbf{M}$

   $$\mathbf{M} = \begin{pmatrix} R_{\mathbf{1}} \\ \frac{-1}{n}\mathbf{1}^T \end{pmatrix} \mathbf{L}_{11}^{-1} \left(R_{\mathbf{1}},\ \frac{-1}{n}\mathbf{1} \right),$$

   where $R_{\mathbf{1}} = (I_{n-1} - \frac{1}{n}\mathbf{1}\mathbf{1}^T)$.
7. $\mathbf{N}(i,j,k) = (m_{ij} + m_{kk} - m_{ik} - m_{kj})\pi_j$ for all i, j, k.

Get Pseudo-Inv of Laplacian

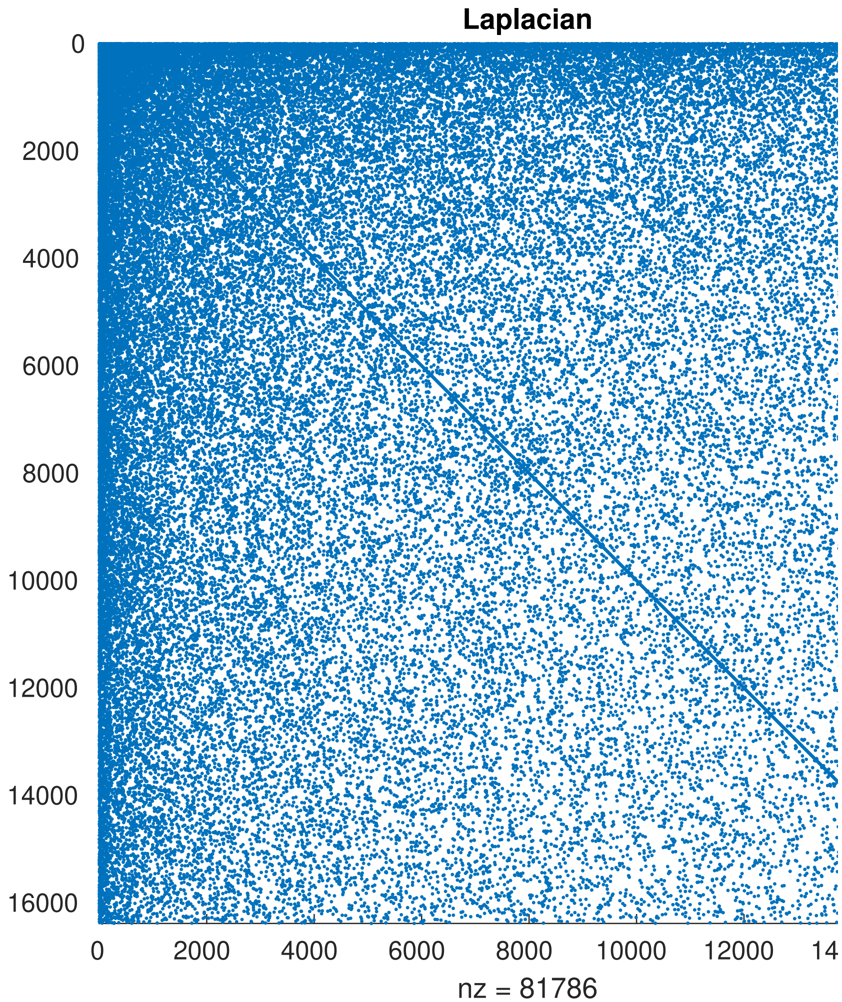
1. Compute normalized Laplacian $L = I - P$.
2. Compute inverse of the upper $(n-1) \times (n-1)$ part: $I - P_{11}$ Expensive
3. Solve for the stationary probabilites: $(\pi_1, \ldots, \pi_{n-1}) = -(L^{\mathrm{p}}_{11})^{-1} \ell^{\mathrm{p}}_{12} \pi_n$;
4. Form random walk Laplacian $\mathbf{L} = \textsc{Diag}(\boldsymbol{\pi}) \cdot L = \mathbf{\Pi}(I - P)$.
5. Compute the inverse of $\mathbf{L}_{11}^{-1} = (I - P_{11})^{-1} \mathbf{\Pi}_1^{-1}$
6. Compute desired pseudoinverse $\mathbf{M}$

$$\mathbf{M} = \begin{pmatrix} R_{\mathbf{1}} \\ \frac{-1}{n} \mathbf{1}^T \end{pmatrix} \mathbf{L}_{11}^{-1} \left(R_{\mathbf{1}}, \ \frac{-1}{n} \mathbf{1} \right),$$

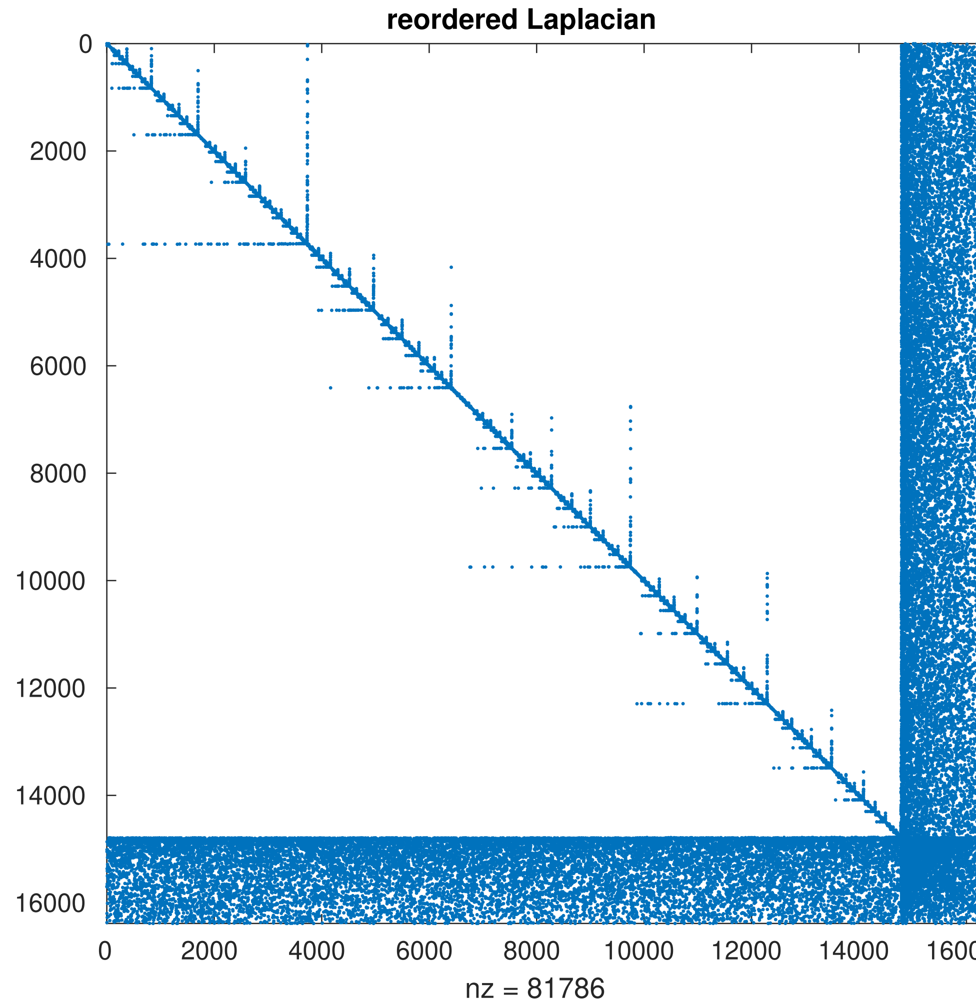
   where $R_{\mathbf{1}} = (I_{n-1} - \frac{1}{n} \mathbf{1}\mathbf{1}^T)$.

7. $\mathbf{N}(i, j, k) = (m_{ij} + m_{kk} - m_{ik} - m_{kj}) \pi_j$ for all i, j, k.

Re-order Laplacian for Small World Graphs



original order



approx minimum degree ordering

Cost for Small World Graphs

number of			time in csec	
vertices	edges	LU fill	LU	backsolve
1,024	4,059	20,620	5	2
2,048	8,140	66,851	2	< 1
4,096	16,314	205,826	4	< 1
8,192	32,671	763,440	12	1
16,384	65,402	2,804,208	56	5
32,768	130,884	10,740,194	250	19
65,536	261,882	43,504,911	1,363	82
131,072	523,920	168,455,437	7,989	328

- Double the size $\Longrightarrow$
  LU cost grows by about a factor of 5 instead of a factor of $2^3 = 8$.

Hitting and Commute Times

Adding up previous gives

- $\mathbf{H}(i,k) = \sum_j \mathbf{N}(i,j,k) = m_{kk} - m_{ik} + \sum_j (m_{ij} - m_{kj})\pi_j$
- $\mathbf{C}(i,k) = \mathbf{H}(i,k) + \mathbf{H}(k,i) = m_{kk} + m_{ii} - m_{ik} - m_{ki}$.
- Above holds also for strongly connected directed graphs (arbitrary Markov chain with no transient states).
- Could add along other dimensions to get betweenness measures, etc.

Commute Times

- Pseudo inverse of $L = L^{\mathrm{a}}/14$ is a Gram matrix:

$$M = L^{+} = \frac{7}{90} \cdot \begin{pmatrix} \mathbf{83} & -1 & -37 & -43 & -19 & 17 \\ -1 & \mathbf{47} & -1 & -19 & -7 & -19 \\ -37 & -1 & \mathbf{83} & 17 & -19 & -43 \\ -43 & -19 & 17 & \mathbf{83} & -1 & -37 \\ -19 & -7 & -19 & -1 & \mathbf{47} & -1 \\ 17 & -19 & -43 & -37 & -1 & \mathbf{83} \end{pmatrix}$$

- $\Longrightarrow$ expected commute times in random walk $[(\ell_2 \text{ metric})^2]$

$$\mathbf{C} = \begin{bmatrix} \mathrm{diag}(L^{+}) \cdot \mathbf{1}^{\mathrm{T}} \\ + \mathbf{1} \cdot \mathrm{diag}(L^{+}) \\ - L^{+} - (L^{+})^{\mathrm{T}} \end{bmatrix} = \frac{14}{15} \cdot \begin{pmatrix} 0 & 11 & 20 & 21 & 14 & 11 \\ 11 & 0 & 11 & 14 & 9 & 14 \\ 20 & 11 & 0 & 11 & 14 & 21 \\ 21 & 14 & 11 & 0 & 11 & 20 \\ 14 & 9 & 14 & 11 & 0 & 11 \\ 11 & 14 & 21 & 20 & 11 & 0 \end{pmatrix}.$$

- Red numbers: average extra cost of detour thru given node.

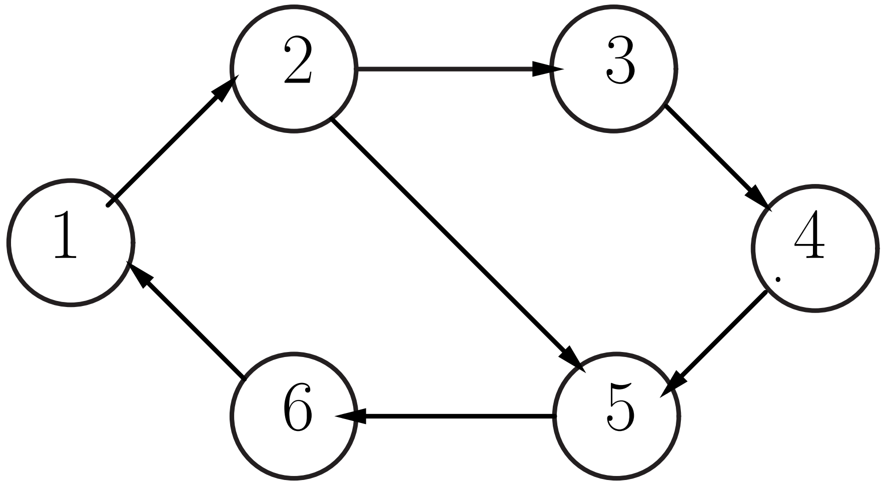
Embedding

- $L^+ = \mathbf{S}^T\mathbf{S}$ with

$$\mathbf{S} = \begin{array}{c} \begin{matrix} \mathbf{s}_1 & \mathbf{s}_2 & \mathbf{s}_3 & \mathbf{s}_4 & \mathbf{s}_5 & \mathbf{s}_6 \end{matrix} \\ \begin{pmatrix} 2.5408 & -.0306 & -1.1326 & -1.3163 & -.5816 & .52040 \\ 0 & 1.9117 & -.0588 & -.7941 & -.2941 & -.7647 \\ 0 & 0 & 2.2736 & -.0947 & -.9473 & -1.2315 \\ 0 & 0 & 0 & 2.02070 & -.5774 & -1.4434 \\ 0 & 0 & 0 & 0 & 1.4142 & -1.4142 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

- For all i, j, $\|\mathbf{s}_i - \mathbf{s}_j\|_2^2 = C_{ij}$.
- Since $L^+\mathbf{1} = \mathbf{0}$, the columns of $\mathbf{S}$ are already centered.
- Previous red numbers are distance^2 from origin = Centrality [83, 47, 83, 83, 47, 83] $\times$ $(7/90)$.

Example – Directed Graph



$$P = \begin{pmatrix} 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 \\ 1.0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{d} = \begin{pmatrix} 1 \\ 2 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \quad \not\propto \quad \boldsymbol{\pi} = \begin{pmatrix} 0.2 \\ 0.2 \\ 0.1 \\ 0.1 \\ 0.2 \\ 0.2 \end{pmatrix}$$

Laplacian from Probabilities

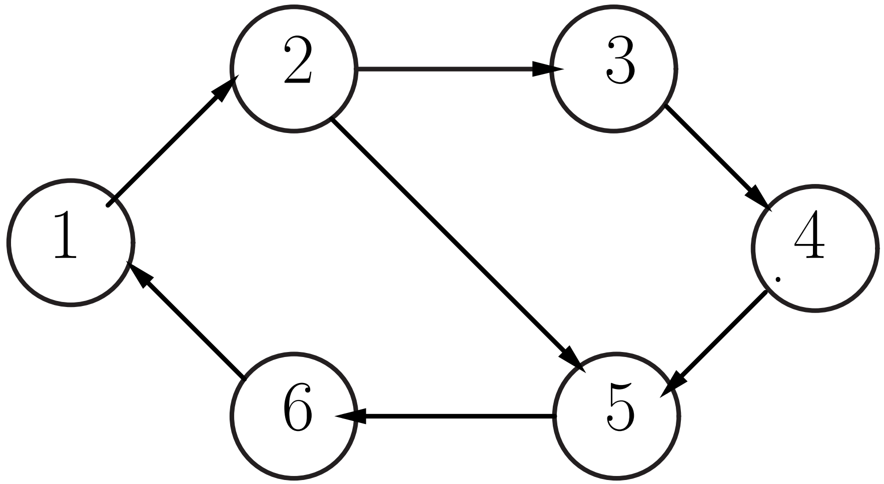
- Obtain Tensor & commute times same way, using $L = \Pi - \Pi P$:

$$L = \begin{pmatrix} 0.2 & -0.2 & 0 & 0 & 0 & 0 \\ 0 & 0.2 & -0.1 & 0 & -0.1 & 0 \\ 0 & 0 & 0.1 & -0.1 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & -0.1 & 0 \\ 0 & 0 & 0 & 0 & 0.2 & -0.2 \\ -0.2 & 0 & 0 & 0 & 0 & 0.2 \end{pmatrix}, \begin{matrix}\text{null} \\ \text{vec}\end{matrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

$$M = L^{+} = \frac{5}{6} \begin{pmatrix} \mathbf{3} & 2 & 0 & -2 & -1 & -2 \\ -2 & \mathbf{3} & 1 & -1 & 0 & -1 \\ -3 & -4 & \mathbf{6} & 4 & -1 & -2 \\ -1 & -2 & -4 & \mathbf{6} & 1 & 0 \\ 1 & 0 & -2 & -4 & \mathbf{3} & 2 \\ 2 & 1 & -1 & -3 & -2 & \mathbf{3} \end{pmatrix}$$

Laplacians: only $\Pi - \Pi P$ has null vector $(1, \ldots, 1)$ on both sides.

Hitting & Commute Times



H (hitting times)

$$\begin{pmatrix} 0 & \mathbf{1} & \mathbf{6} & \mathbf{7} & \mathbf{3} & \mathbf{4} \\ 4 & 0 & 5 & 6 & 2 & 3 \\ 4 & 5 & 0 & 1 & 2 & 3 \\ 3 & 4 & 9 & 0 & 1 & 2 \\ 2 & 3 & 8 & 9 & 0 & 1 \\ 1 & 2 & 7 & 8 & 4 & 0 \end{pmatrix}$$

C (commute times)

$$\begin{pmatrix} 0 & \mathbf{5} & \mathbf{10} & \mathbf{10} & \mathbf{5} & \mathbf{5} \\ 5 & 0 & 10 & 10 & 5 & 5 \\ 10 & 10 & 0 & 10 & 10 & 10 \\ 10 & 10 & 10 & 0 & 10 & 10 \\ 5 & 5 & 10 & 10 & 0 & 5 \\ 5 & 5 & 10 & 10 & 5 & 0 \end{pmatrix}$$

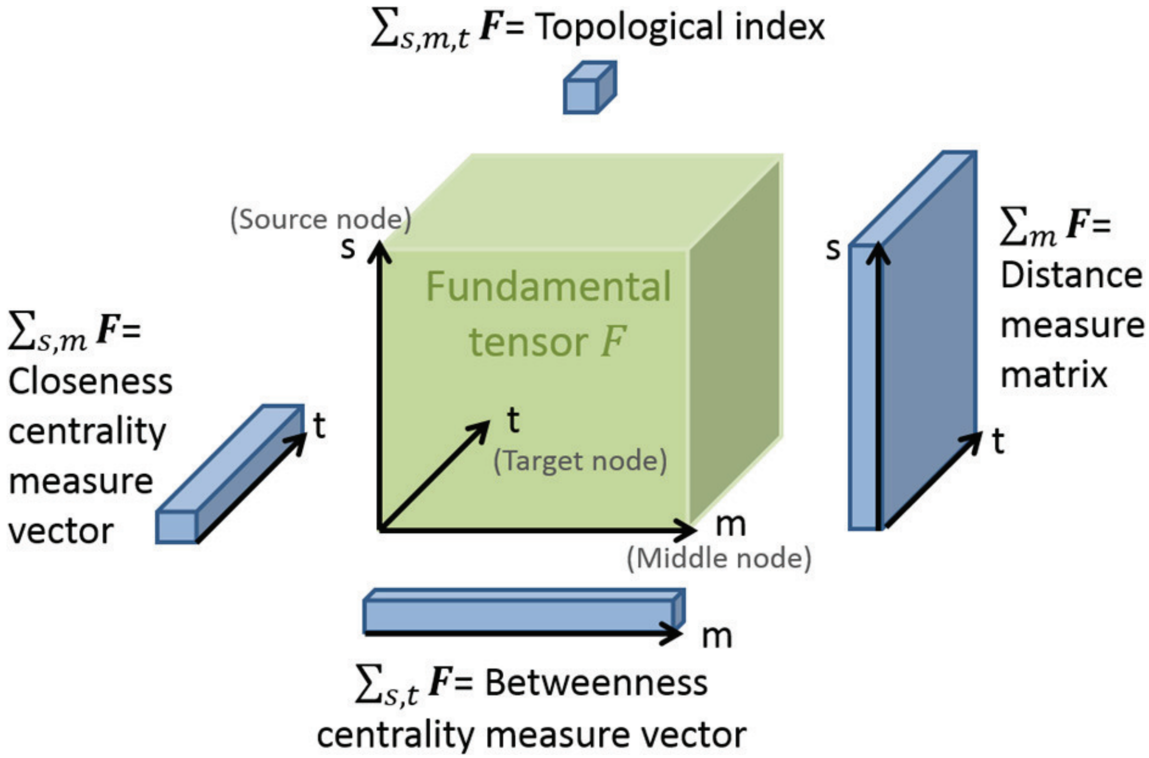
- Only nodes 3, 4 are peripheral. Others are all equally important.
- Same reflected in average commute times from node 2.

Tensor

$\boldsymbol{N} =$

```
k=1
 (  0   0   0   0   0   0)
 (  0 1.0 0.5 0.5 1.0 1.0)
 (  0   0 1.0 1.0 1.0 1.0)
 (  0   0   0 1.0 1.0 1.0)
 (  0   0   0   0 1.0 1.0)
 (  0   0   0   0   0 1.0)

k=2
 (1.0   0   0   0   0   0)
 (  0   0   0   0   0   0)
 (1.0   0 1.0 1.0 1.0 1.0)
 (1.0   0   0 1.0 1.0 1.0)
 (1.0   0   0   0 1.0 1.0)
 (1.0   0   0   0   0 1.0)

k=3
 (2.0 2.0   0   0 1.0 1.0)
 (1.0 2.0   0   0 1.0 1.0)
 (  0   0   0   0   0   0)
 (2.0 2.0   0 1.0 2.0 2.0)
 (2.0 2.0   0   0 2.0 2.0)
 (2.0 2.0   0   0 1.0 2.0)

k=4
 (2.0 2.0 1.0   0 1.0 1.0)
 (1.0 2.0 1.0   0 1.0 1.0)
 (  0   0 1.0   0   0   0)
 (  0   0   0   0   0   0)
 (2.0 2.0 1.0   0 2.0 2.0)
 (2.0 2.0 1.0   0 1.0 2.0)

k=5
 (1.0 1.0 0.5 0.5   0   0)
 (  0 1.0 0.5 0.5   0   0)
 (  0   0 1.0 1.0   0   0)
 (  0   0   0 1.0   0   0)
 (  0   0   0   0   0   0)
 (1.0 1.0 0.5 0.5   0 1.0)

k=6
 (1.0 1.0 0.5 0.5 1.0   0)
 (  0 1.0 0.5 0.5 1.0   0)
 (  0   0 1.0 1.0 1.0   0)
 (  0   0   0 1.0 1.0   0)
 (  0   0   0   0 1.0   0)
 (  0   0   0   0   0   0)
```

Tensor Applications



Tensor Applications

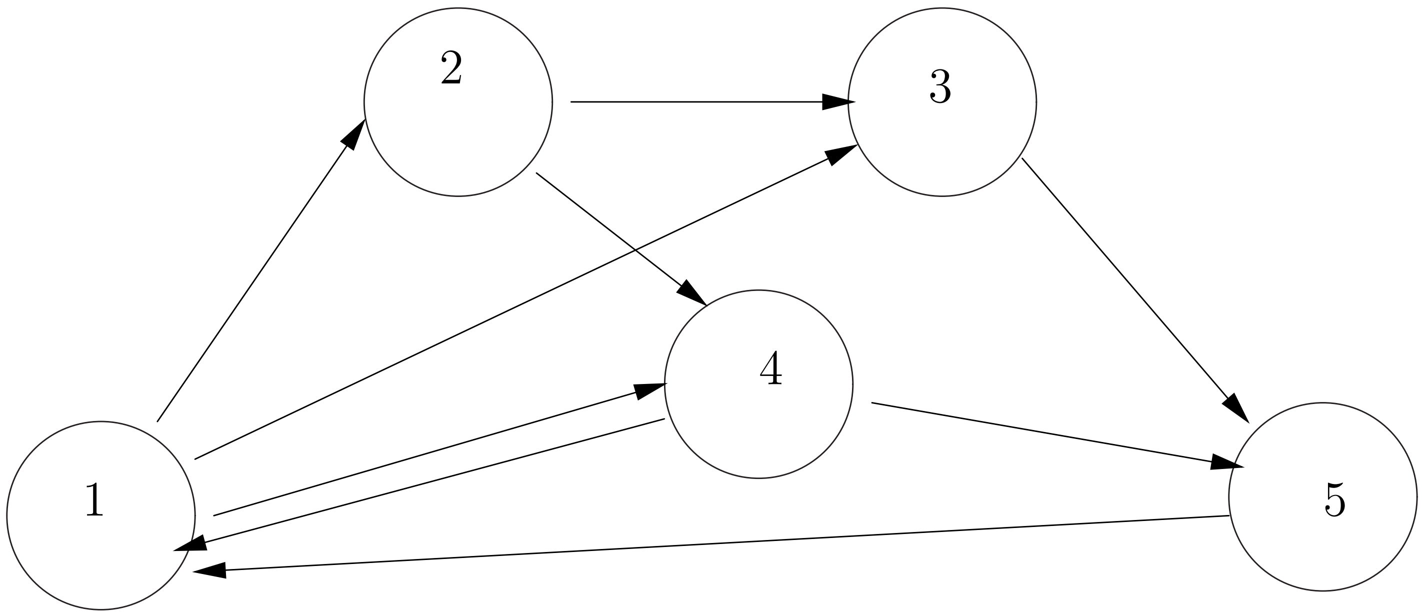
- Closeness(k): $\sum_{ij} \boldsymbol{N}(i,j,k)$
  39.4, 20.6, 39.4, 39.4, 20.6, 39.4 for undirected graph
  14.0, 15.0, 35.0, 31.0, 12.0, 13.0: for directed graph

- Betweenness(j): $\sum_{ik} \boldsymbol{N}(i,j,k)$
  28.4, 42.6, 28.4, 28.4, 42.6, 28.4 for undirected graph
  24.0, 24.0, 12.0, 12.0, 24.0, 24.0 for directed graph

- Probability of passage (i,j,k): $\widetilde{\boldsymbol{N}}(i,j,k) = \boldsymbol{N}(i,j,k)/\boldsymbol{N}(j,j,k)$
  [or zero if $i = k$ or $j = k$].

- node load: chance of passage averaged over all start/end nodes
  $\sum_{ik} \widetilde{\boldsymbol{N}}(i,j,k) \;/\; (n-1)^2$.
  0.54, 0.72, 0.54, 0.54, 0.72, 0.54, for undirected graph
  0.64, 0.60, 0.48, 0.48, 0.72, 0.68 for directed graph

Example – Directed Graph



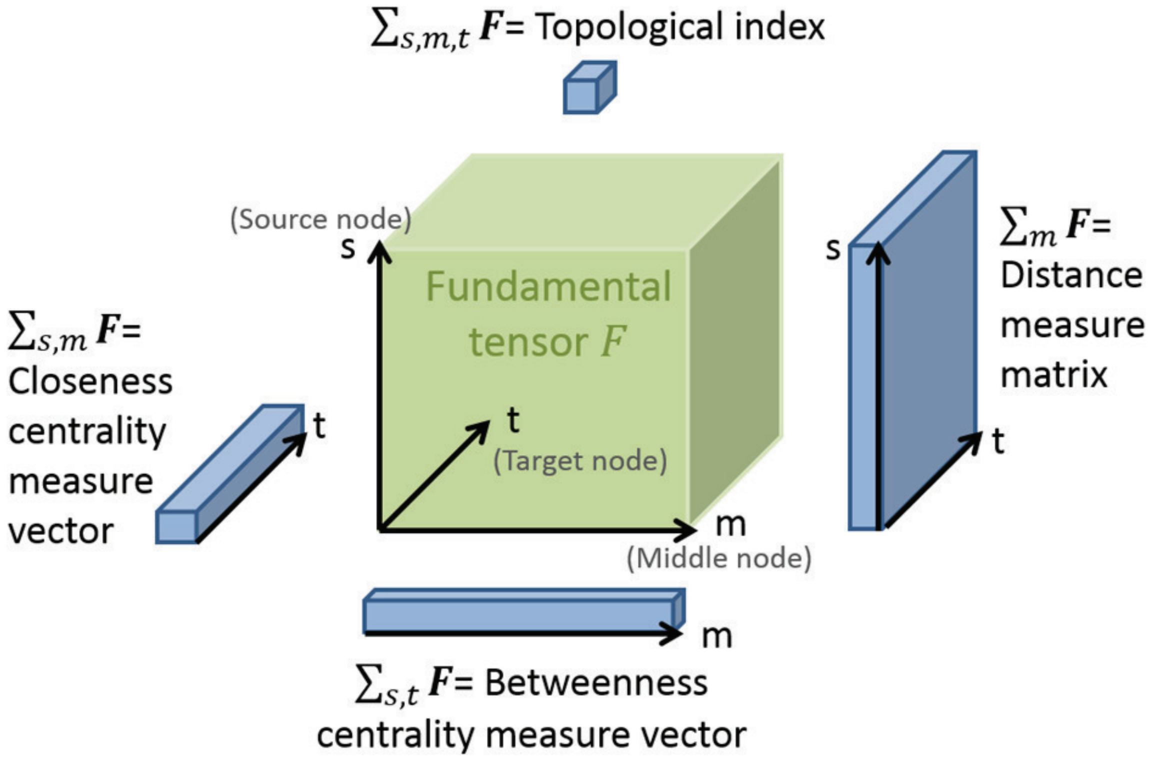

```
P =                                                      D =    π =
 (   0      .333333 .333333 .333333   0    )   (3)    (12)
 (   0        0     .500000 .500000   0    )   (2)    ( 4)    1
 (   0        0       0       0     1.00000)   (1)    ( 6) * --
 (.500000     0       0       0     .500000)   (2)    ( 6)   37
 (   1        0       0       0       0    )   (1)    ( 9)
```

Laplacian from Probabilities

- Obtain Tensor & commute times same way, using $L = \Pi - \Pi P$:

```
L =         (     12      -4      -4      -4      0)
       1    (      0       4      -2      -2      0)
      --    (      0       0       6       0     -6)
      37    (     -3       0       0       6     -3)
            (     -9       0       0       0      9)

M =         (      8      -1      -1      -1     -5)
      37    (    -10      26      -4      -4     -8)
     ---    (     -2     -11      19     -11      5)
     180    (      0      -9      -9      21     -3)
            (      4      -5      -5      -5     11)
```

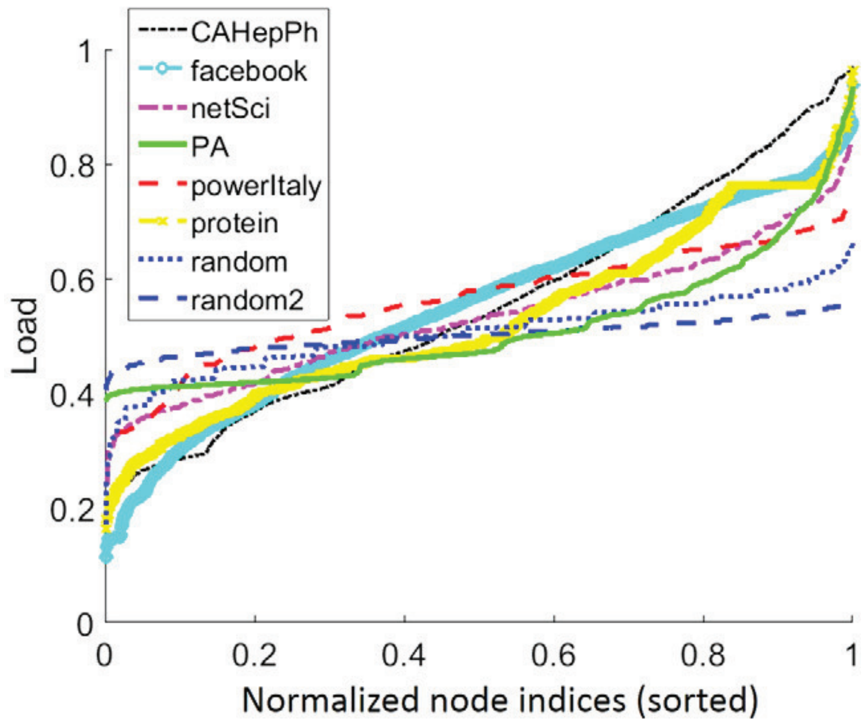
Tensor Applications



Tensor Applications

- Closeness(k): $\sum_{ij} \boldsymbol{N}(i,j,k)$
  (7.25000 30.5000 18.8333 21.3333 9.44444)
- Betweenness(j): $\sum_{ik} \boldsymbol{N}(i,j,k)$
  (28.3333 9.44444 14.1666 14.1666 21.2500)
- Probability of passage (i,j,k): $\widetilde{\boldsymbol{N}}(i,j,k) = \boldsymbol{N}(i,j,k)/\boldsymbol{N}(j,j,k)$
  [or zero if $i = k$ or $j = k$].
- node load: chance of passage averaged over all start/end nodes
  $\sum_{ik} \widetilde{\boldsymbol{N}}(i,j,k) \;/\; (n-1)^2$.
  (.796875 .475000 .572916 .531250 .748958)
- prob of passing 2 when starting from 1 before reaching 5:
  .400 (full network); .500 (if 4 removed); .333 (if 4 is to be avoided)

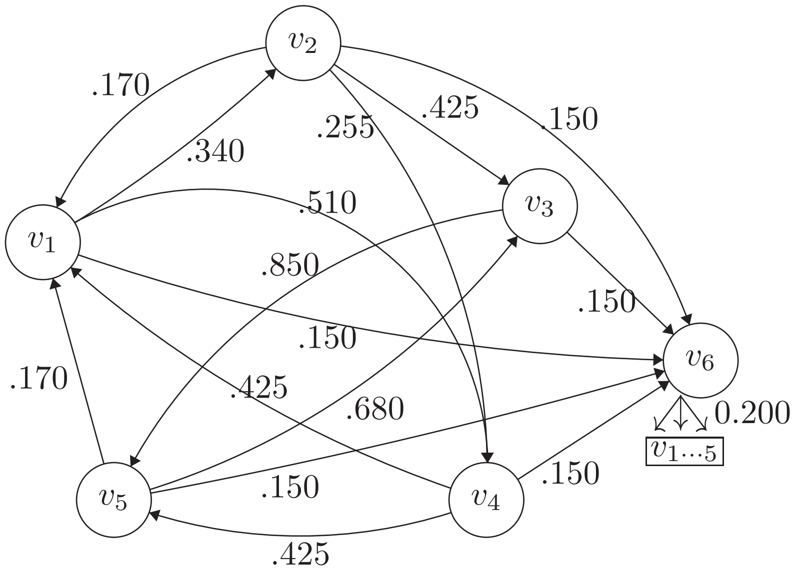
Load examples



- High Energy Physics Collaboration Network
- sampled Facebook network
- netSci: co-author network of network scientists
- Italian power grid
- protein-protein interaction network
- synthetic: Preferential Attachment
- synthetic: random Erdös-Rényi (8 or 40 initial links)

- Load is most balanced in ER network, less so in PA.
- Co-author networks: more imbalanced.
- power network: more balanced load.

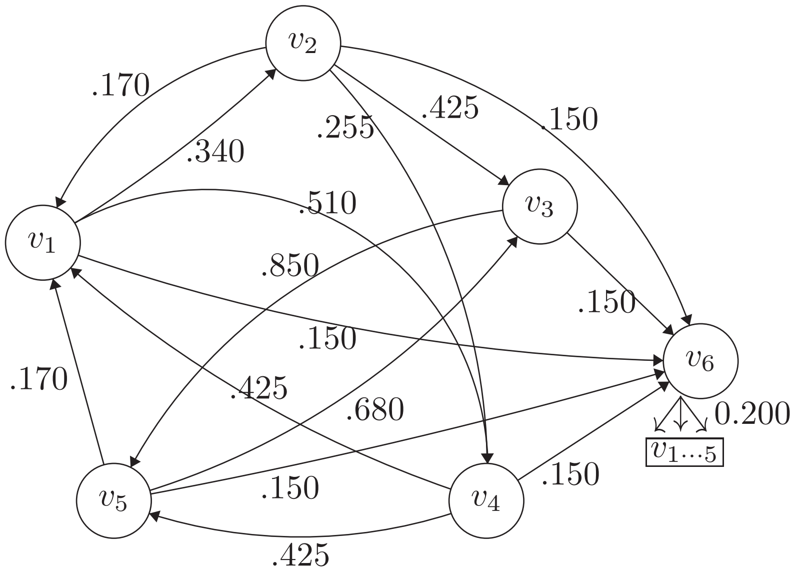
Influence Propagation



- Network shows how much agent i trusts agent j, for $i, j{=}1, \ldots, 5$ after direct interaction.
- Node 6 is extra evaporation node to control influence locality.

- Seek to predict trust between new pairs

Influence Propagation



- Network shows how much agent i trusts agent j, for i, j=1, ..., 5 after direct interaction.
- Node 6 is extra evaporation node to control influence locality.

- Seek to predict trust between new pairs: predicted trust of j by i = $\mathbf{PHT}(i,j) = \boldsymbol{N}(i,j,6)/\boldsymbol{N}(j,j,6)$. $\boxed{\text{A}}$
- Example: from point of view of agent 4:

$$\mathbf{PHT}(4, 1{:}5) = (0.60,\ 0.29,\ 0.53,\ 1.00,\ 0.66),$$

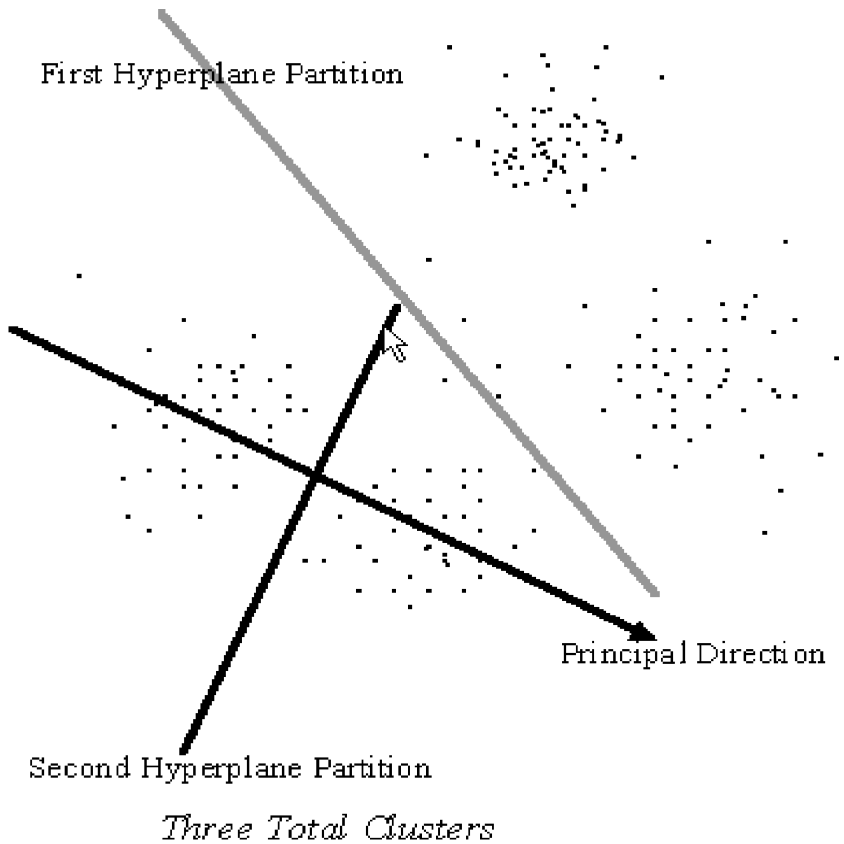
- If agent 2 is a bad actor to be avoided then we can use Fundamental Tensor to compute trust values based on avoiding 2:

$$\mathbf{PHT}_2(4, 1{:}5) = (0.60,\ 0.00,\ 0.39,\ 1.00,\ 0.54)$$

[plug $\widehat{\boldsymbol{N}} = \boldsymbol{N}(:,:,6) - \frac{\boldsymbol{N}(:,2,6)}{\boldsymbol{N}(2,2,6)}\boldsymbol{N}(2,:,6)$ into $\boxed{\text{A}}$.

- Now most trustworthy node for 4 is node 1 instead of node 5.

Principal Direction Divisive Partitioning



(Boley, 1998)

Divisive Partitioning for Unsupervised Clustering

- Unsupervised, as opposed to Supervised:
  - No predefined categories;
  - No previously classified training data;
  - No a-priori assumptions on the number of clusters.
- Top-down Hierarchical:
  - Imposes a tree hierarchy on unstructured data;
  - Tree is source for some taxomonic information for dataset;
  - Tree is generated from the root down.
  - Result is Principal Direction Divisive Partitioning. (Boley, 1998)
- Multiway Clustering – variations using similar ideas.
  - Project onto first k principal directions. Result: each data sample is represented by k components.
  - Apply classical k-means clustering to projected data.
  - Used for both Graph Partitioning and Data Clustering. (Dhillon, 2001)
- Empirically Best Approach: a hybrid method:
  - Use Divisive Partitioning first (deterministic).
  - Refine with K-means (avoids initialization issues). (Savaresi & Boley, 2004)

Conclusions

- Many different types of data, many highly unstructured.
- Extracting patterns or connections in data involves somehow reducing the volume of data one must look at.
- Data Reduction is an old paradigm that has been updated for the modern digital age.
- Methods discussed here started with classical PCA - SVD based approaches (e.g., assuming independent gaussian noise).
- Connections and pair-wise correlations modeled by graphs.
- Graphs modeled by random walks, counting subgraphs, min-cut/max-flow, models, ...
- Sparse representations: wide variety of sparse approximations: low fill, short basis, non-negative basis, non-squared loss function, count violations of some constraints, low rank (nuclear norm $= L1$-norm on the singular values), ...
- Leads to need for scalable solvers for very large convex programs.

Conclusions

- Many different types of data, many highly unstructured.
- Extracting patterns or connections in data involves somehow reducing the volume of data one must look at.
- Data Reduction is an old paradigm that has been updated for the modern digital age.
- Methods discussed here started with classical PCA - SVD based approaches (e.g., assuming independent gaussian noise).
- Connections and pair-wise correlations modeled by graphs.
- Graphs modeled by random walks, counting subgraphs, min-cut/max-flow, models, ...
- Sparse representations: wide variety of sparse approximations: low fill, short basis, non-negative basis, non-squared loss function, count violations of some constraints, low rank (nuclear norm $= L1$-norm on the singular values), ...
- Leads to need for scalable solvers for very large convex programs.

References

Bamieh, B., Jovanovic, M., Mitra, P., & Patterson, S. (2008). Effect of topological dimension on rigidity of vehicle formations: Fundamental limitations of local feedback. *Proc. CDC* (pp. 369–374). Cancun, Mexico.

Boley, D., Buendia, A., & Golnari, G. (2018). Random walk laplacian and network centrality measures. arXiv 1808.02912.

Boley, D., Ranjan, G., & Zhang, Z.-L. (2010). Commute times for a directed graph using an asymmetric laplacian. U of Mn CS&E Dept report TR 10-005, Univ. of Minn., Dept. of Comp uter Science and Eng. (a revised version is to appear in LAA 2011).

Boley, D., Ranjan, G., & Zhang, Z.-L. (2011). Commute times for a directed graph using an asymmetric Laplacian. *Linear Algebra and Appl.*, *435*, 224–242.

Boley, D. L. (1998). Principal direction divisive partitioning. *Data Mining and Knowledge Discovery*, *2*, 325–344.

Brualdi, R. A., & Ryser, H. J. (1991). *Combinatorial matrix theory*. Cambridge Univ. Press.

Chandra, A., Raghavan, P., Ruzzo, W., Smolensky, R., & Tiwari, P. (1989). The electrical resistance of a graph captures its commute and cover times. *Proc. of Annual ACM Symposium on Theory of Computing* (pp. 574–586).

Chebotarev, P., & Shamis, E. (2006). Matrix-forest theorems.

Dhillon, I. S. (2001). Co-clustering documents and words using bipartite spectral graph partitioning. *KDD* (pp. 269–274).

Doyle, P., & Snell, J. (1984). *Random walks and electric networks*. The Math. Assoc. of Am. front.math.ucdavis.edu/math.PR/0001057.

Fouss, F., Pirotte, A., Renders, J., & Saerens, M. (2007). Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Trans. on Knowledge and Data Engineering*, *19*, 355–369.

Golnari, G., Zhang, Z.-L., & Boley, D. (2019). Markov fundamental tensor and its applications to network analysis. *Linear Algebra and Appl.*, *564*, 126–158.

Gower, J., & Legendre, P. (1986). Metric and euclidean properties of dissimilarities coefficients. *J. Classification*, *3*, 5–48.

Grinstead, C. M., & Snell, J. L. (2006). *Introduction to probability*. American Mathematical Society. 2nd edition, www.dartmouth.edu/~chance/teaching_aids/books_articles/probability_book/book.html.

Klein, D., & Randic, M. (1993). Resistance distance. *J. Math. Chemistry*, *12*, 81–95.

Lam, H. C., Sreevatsan, S., & Boley, D. (2012). Analyzing influenza virus sequences using binary encoding approach. *Scientific Programming*, *20*, 3–13.

Olfati-Saber, R., Murray, R. M., A, & B (2004). Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. Auto. Contr.*, *49*, 1520–1533.

Savaresi, S. M., & Boley, D. (2004). A comparative analysis on the bisecting K-means and the PDDP clustering algorithms. *Intelligent Data Analysis*, *8*, 345–362.

Shi, J., & Malik, H. (2000). Normalized cuts and image segmentation. *IEEE Trans PAMI*, *22*, 888–905.

Spielman, D. A., & Teng, S.-H. (1996). Spectral partitioning works:planar graphs and finite element meshes. *37th Annual Symposium on Foundations of Computer Science*. IEEE Computer Soc. Press.

von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, *17*, 395–416.

Young, G. F., Scandovi, L., & Leonard, N. (2010). Robustness of noisy consensus dynamics with directed communication. *Proc. ACC* (pp. 6312–6317).