Mining signed networks: theory and applications

Aristides Gionis,[1,2] Antonis Matakos,[1] Bruno Ordozgoiti,[1] Han Xiao[1]

[1]Aalto University, [2]KTH Royal Institute of Technology

THE WEB CONFERENCE

April 20-24

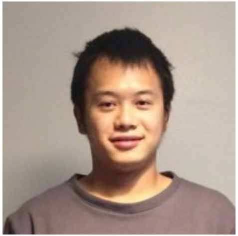
TAIPEI

2020

Slides can be found at

bit.ly/www2020-signed

Contact us

Aris — argioni@kth.se

Antonis, Bruno, Han — {antonis.matakos, bruno.ordozgoiti, han.xiao}@aalto.fi

Outline

Introduction

Signed graphs

Graphs with edge signs,
either *positive* or *negative*

Signed graphs: motivation

Human interaction

friendly or antagonistic

Image source: pxfuel.com

Signed graphs: motivation

Online social media

- a graph of users (Twitter, Facebook, etc.)
- Users may express like or dislike towards others
- Can be used to study online polarization

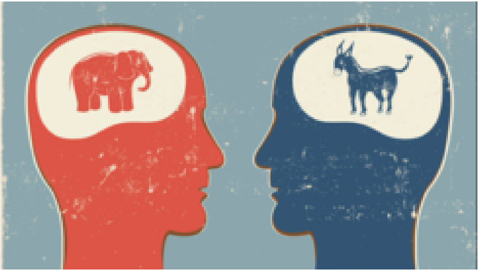
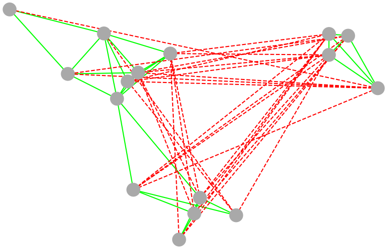


Image source: iStockphoto.com

Signed graphs: motivation

Groups of humans

- Examples: tribes, political parties, countries, etc.
- Country relations during war
- Political polarization



New Guinea Highland Tribes graph Read (1954)

Signed graphs: motivation

Human language

- a graph of words
- that captures synonym or antonym relations

“Happy”

Image source: thesaurus.com

Signed graphs: motivation

Molecular biology

- a graph of proteins
- one protein activate or inhibit the functioning of another

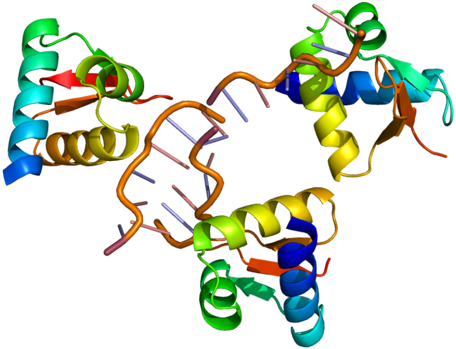


Image source: commons.wikimedia.org

Signed graphs: motivation

Finance

- a graph of securities (tradable assets)
- a security *correlates* positively/negatively with another.
- "Correlate" means the joint movement of price

Image source: vecteezy.com

Signed graphs: motivation

Linear algebra

- Speed up linear program solver by finding an embedded network matrix.

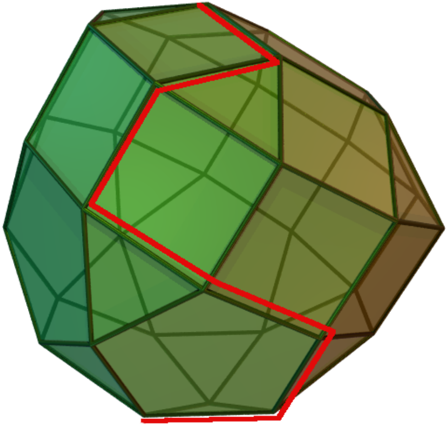


Image source: wiki.ubc.ca

Theory of signed networks

Outline

We will discuss:

Outline

We will discuss:

- Balance

Outline

We will discuss:

- Balance
- Spectrum

Outline

We will discuss:

- Balance
- Spectrum
- Frustration

Signed networks

Signed networks (or graphs): each edge labeled $+$ or $-$.

Definitions:

- $G = (V, E^+, E^-)$,
- $G = (V, E, \sigma), \sigma : E \to \{-, +\}$.

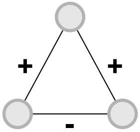
Signed networks

Signed networks (or graphs): each edge labeled $+$ or $-$.

Definitions:

- $G = (V, E^+, E^-)$,
- $G = (V, E, \sigma), \sigma : E \to \{-, +\}$.

Adjacency matrix: $A = A_{E^+} - A_{E^-}$

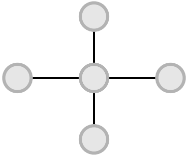


$$\mapsto \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & -1 \\ 1 & -1 & 0 \end{pmatrix}$$

Expressiveness of signed graphs

Signed graphs can be quite expressive.

Example: Star graph

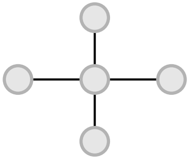


- Number of possible graphs:
- Number of non-isomorphic graphs:

Expressiveness of signed graphs

Signed graphs can be quite expressive.

Example: Star graph

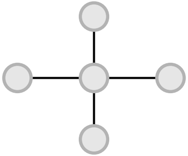


- Number of possible graphs: $2^{|E|}$
- Number of non-isomorphic graphs:

Expressiveness of signed graphs

Signed graphs can be quite expressive.

Example: Star graph



- Number of possible graphs: $2^{|E|}$
- Number of non-isomorphic graphs: $|E|$

Differences in signed graphs

Shortest paths

Signed graphs can be quite different... Consider e.g. shortest paths.

How do we even define path length in signed graphs?

Signed graphs can be quite different... Consider e.g. shortest paths.
How do we even define path length in signed graphs?

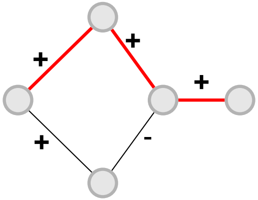
Proposal: distinguish positive and negative paths (by product of edge signs).

Differences in signed graphs

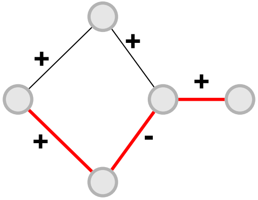
Shortest paths

Signed graphs can be quite different... Consider e.g. shortest paths.
How do we even define path length in signed graphs?

Proposal: distinguish positive and negative paths (by product of edge signs).



Positive path



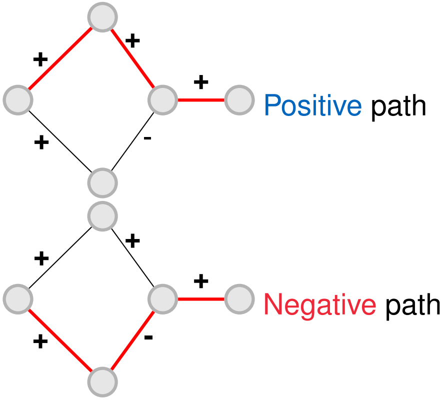
Negative path

Differences in signed graphs

Shortest paths

Signed graphs can be quite different... Consider e.g. shortest paths.
How do we even define path length in signed graphs?

Proposal: distinguish positive and negative paths (by product of edge signs).



Finding all shortest simple signed paths is **NP**-complete!

If repetitions are allowed, $\mathcal{O}(|E| \log\log \frac{D}{d})$ algorithm (Hansen, 1984).

Densest subgraph problem in unsigned graphs:

$$\max_{x \in \{0,1\}^n} \frac{x^T A x}{x^T x}$$

Polynomial-time solvable (Goldberg, 1984).

Densest subgraph problem in unsigned graphs:

$$\max_{x \in \{0,1\}^n} \frac{x^T A x}{x^T x}$$

Polynomial-time solvable (Goldberg, 1984).

Densest subgraph problem in signed graphs:

$$\max_{x \in \{-1,0,1\}^n} \frac{x^T A x}{x^T x}$$

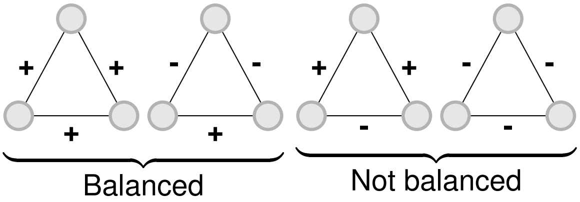
NP-hard ! (Bonchi et al., 2019; Tsourakakis et al., 2019).

Balance

Motivation

Motivation: **balance** in social networks (Harary, 1953).

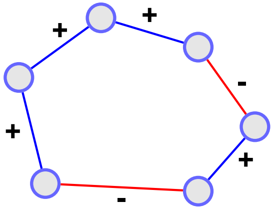
“*The friend of a friend is a friend*” (or “*the enemy of a friend is an enemy*”).



The four possible non-isomorphic signed triangles.

Motivation

Balance applies to cycles of any length.



$$+ \times - \times + \times - \times + \times + = +.$$

Definition of balanced cycle

A cycle is balanced if the product of its signs is positive.

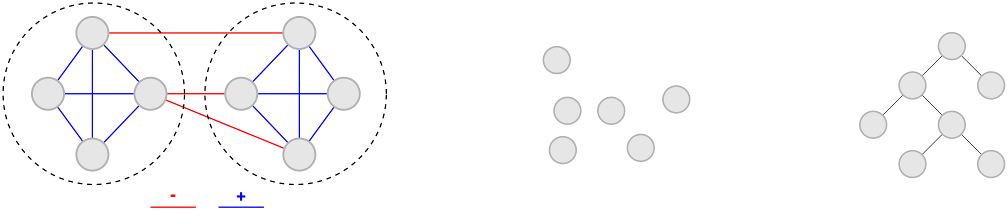
Motivation

Characterizations of balance

G is balanced iff

- There are no negative (unbalanced) cycles.

Some balanced graphs



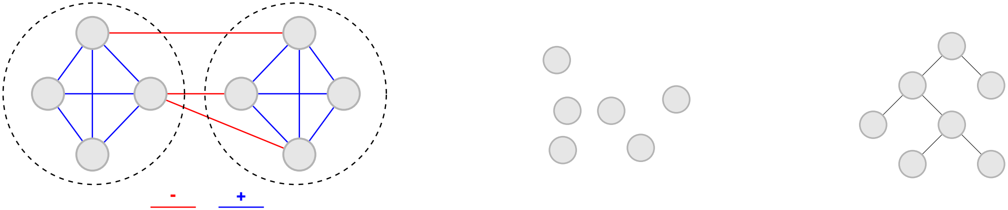
Motivation

Characterizations of balance

G is balanced iff

- There are no negative (unbalanced) cycles.
- There exists a sign-compliant partition: $V = V_1 \cup V_2$, all **+** edges within sets, all **-** edges between sets.

Some balanced graphs



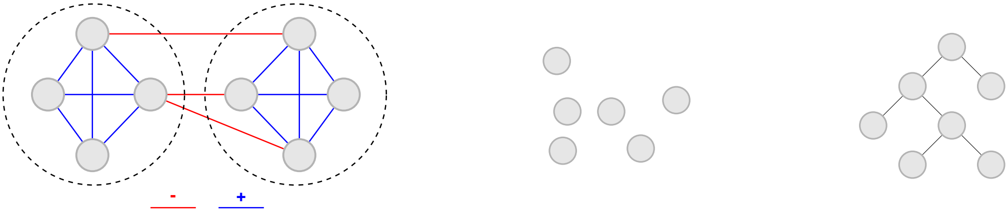
Motivation

Characterizations of balance

G is balanced iff

- There are no negative (unbalanced) cycles.
- There exists a sign-compliant partition: $V = V_1 \cup V_2$, all **+** edges within sets, all **-** edges between sets.
- All paths between any pair u, v have same sign.

Some balanced graphs



Measures of partial balance

How can we measure partial balance?

- Fraction of balanced cycles
  (Cartwright and Harary, 1956; Giscard et al., 2017)
  - Fraction of balanced triangles
    (Terzi and Winkler, 2011) (example in next slide)
- Spectral methods (discussed later on)

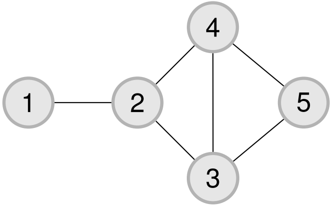
Check Aref and Wilson (2018) for an overview of partial measures of balance.

Measures of partial balance

Example: fraction of balanced triangles

Reminder: counting triangles in unsigned graphs:

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

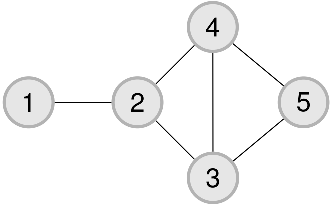


Measures of partial balance

Example: fraction of balanced triangles

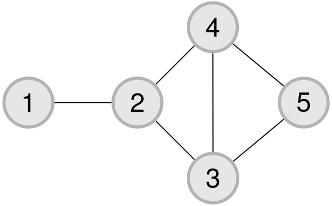
Reminder: counting triangles in unsigned graphs:

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}, \; A^2 = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 3 & 1 & 1 & 2 \\ 1 & 1 & 3 & 2 & 1 \\ 1 & 1 & 2 & 3 & 1 \\ 0 & 2 & 1 & 1 & 2 \end{pmatrix}$$



Reminder: counting triangles in unsigned graphs:

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}, \; A^2 = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 3 & 1 & 1 & 2 \\ 1 & 1 & 3 & 2 & 1 \\ 1 & 1 & 2 & 3 & 1 \\ 0 & 2 & 1 & 1 & 2 \end{pmatrix}, \; A^3 = \begin{pmatrix} 0 & 3 & 1 & 1 & 2 \\ 3 & 2 & 6 & 6 & 2 \\ 1 & 6 & 4 & 5 & 5 \\ 1 & 6 & 5 & 4 & 5 \\ 2 & 2 & 5 & 5 & 2 \end{pmatrix}$$

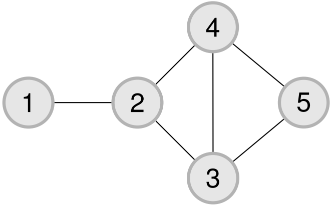


Measures of partial balance

Example: fraction of balanced triangles

Reminder: counting triangles in unsigned graphs:

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}, \quad A^2 = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 3 & 1 & 1 & 2 \\ 1 & 1 & 3 & 2 & 1 \\ 1 & 1 & 2 & 3 & 1 \\ 0 & 2 & 1 & 1 & 2 \end{pmatrix}, \quad A^3 = \begin{pmatrix} 0 & 3 & 1 & 1 & 2 \\ 3 & 2 & 6 & 6 & 2 \\ 1 & 6 & 4 & 5 & 5 \\ 1 & 6 & 5 & 4 & 5 \\ 2 & 2 & 5 & 5 & 2 \end{pmatrix}$$



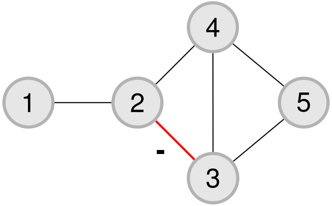
$$A^3_{ii} = 2 \times \#(\text{3-cycles adjacent to vertex } i).$$

Measures of partial balance

Example: fraction of balanced triangles

Counting triangles in signed graphs:

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

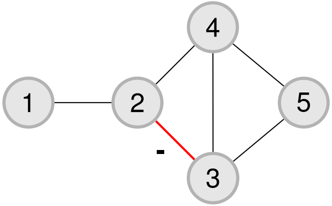


Measures of partial balance

Example: fraction of balanced triangles

Counting triangles in signed graphs:

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}, A^2 = \begin{pmatrix} 1 & 0 & -1 & 1 & 0 \\ 0 & 3 & 1 & -1 & 0 \\ -1 & 1 & 3 & 0 & 1 \\ 1 & -1 & 0 & 3 & 1 \\ 0 & 0 & 1 & 1 & 2 \end{pmatrix}$$

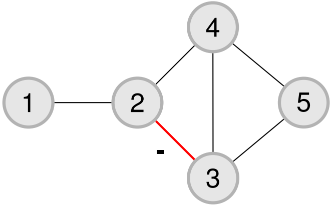


Measures of partial balance

Example: fraction of balanced triangles

Counting triangles in signed graphs:

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}, A^2 = \begin{pmatrix} 1 & 0 & -1 & 1 & 0 \\ 0 & 3 & 1 & -1 & 0 \\ -1 & 1 & 3 & 0 & 1 \\ 1 & -1 & 0 & 3 & 1 \\ 0 & 0 & 1 & 1 & 2 \end{pmatrix}, A^3 = \begin{pmatrix} 0 & 3 & 1 & -1 & 0 \\ 3 & -2 & -4 & 4 & 0 \\ 1 & -4 & 0 & 5 & 3 \\ -1 & 4 & 5 & 0 & 3 \\ 0 & 0 & 3 & 3 & 2 \end{pmatrix}$$

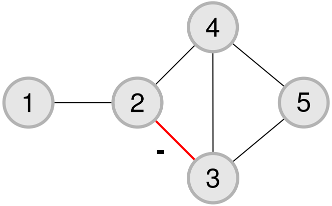


Measures of partial balance

Example: fraction of balanced triangles

Counting triangles in signed graphs:

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}, A^2 = \begin{pmatrix} 1 & 0 & -1 & 1 & 0 \\ 0 & 3 & 1 & -1 & 0 \\ -1 & 1 & 3 & 0 & 1 \\ 1 & -1 & 0 & 3 & 1 \\ 0 & 0 & 1 & 1 & 2 \end{pmatrix}, A^3 = \begin{pmatrix} 0 & 3 & 1 & -1 & 0 \\ 3 & -2 & -4 & 4 & 0 \\ 1 & -4 & 0 & 5 & 3 \\ -1 & 4 & 5 & 0 & 3 \\ 0 & 0 & 3 & 3 & 2 \end{pmatrix}$$



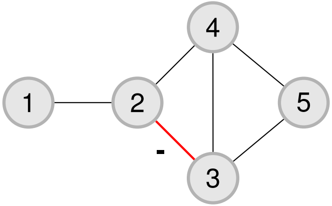
$$A^3_{ii} = 2 \times (\#\text{balanced 3-cycles} - \#\text{unbalanced 3-cyles}).$$

Measures of partial balance

Example: fraction of balanced triangles

Counting triangles in signed graphs:

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}, \; A^2 = \begin{pmatrix} 1 & 0 & -1 & 1 & 0 \\ 0 & 3 & 1 & -1 & 0 \\ -1 & 1 & 3 & 0 & 1 \\ 1 & -1 & 0 & 3 & 1 \\ 0 & 0 & 1 & 1 & 2 \end{pmatrix}, \; A^3 = \begin{pmatrix} 0 & 3 & 1 & -1 & 0 \\ 3 & -2 & -4 & 4 & 0 \\ 1 & -4 & 0 & 5 & 3 \\ -1 & 4 & 5 & 0 & 3 \\ 0 & 0 & 3 & 3 & 2 \end{pmatrix}$$



$A^3_{ii} = 2 \times (\#\text{balanced 3-cycles} - \#\text{unbalanced 3-cyles})$. Thus,

$$\frac{Tr(A^3) + Tr(|A|^3)}{2\,Tr(|A|^3)} = \text{fraction of balanced triangles.}$$

(Terzi and Winkler, 2011)

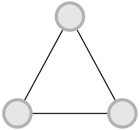
Note: $|A|$ is the adj. matrix of the *underlying* (unsigned) graph.

Spectrum

Spectral theory

Review of unsigned spectral theory:

Laplacian: $L = D - A$

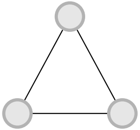


$$L = \begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{pmatrix}$$

Spectral theory

Review of unsigned spectral theory:

Laplacian: $L = D - A$



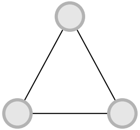
$$Lv_1 = \begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = 0$$

- $\lambda_{min}(L) = 0$ (Multiplicity of 0 = n. of connected components)

Spectral theory

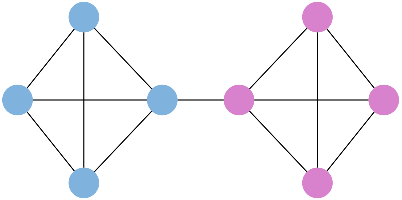
Review of unsigned spectral theory:

Laplacian: $L = D - A$



$$Lv_1 = \begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = 0$$

- $\lambda_{min}(L) = 0$ (Multiplicity of 0 = n. of connected components)
- Eigenvector v_2 gives a "good" partition (Cheeger inequality).



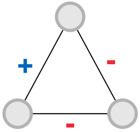
$$v_2 \approx \begin{pmatrix} -0.25 \\ -0.38 \\ -0.38 \\ -0.38 \\ 0.38 \\ 0.38 \\ 0.38 \\ 0.25 \end{pmatrix}, \lambda_2(L) \approx 0.35.$$

Spectral theory

Signed spectral theory:

Laplacian: $L = D - A$

Unsigned	Signed
L is positive semidefinite	
$D_{ii} = \sum_j A_{ij}$	$D_{ii} = \sum_j \lvert A_{ij} \rvert$
$\lambda_{min}(L) = 0$	$\lambda_{min}(L) \geq 0$



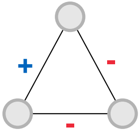
$$L = \begin{pmatrix} 2 & -1 & 1 \\ -1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix}$$

Spectral theory

Signed spectral theory:

Laplacian: $L = D - A$

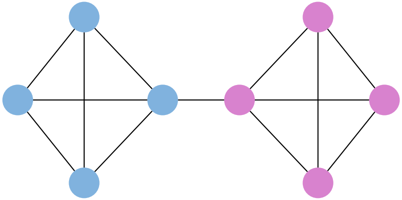
Unsigned	Signed
L is positive semidefinite	
$D_{ii} = \sum_j A_{ij}$	$D_{ii} = \sum_j \lvert A_{ij} \rvert$
$\lambda_{min}(L) = 0$	$\lambda_{min}(L) \geq 0$



$$Lv_1 = \begin{pmatrix} 2 & -1 & 1 \\ -1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix} = 0$$

Spectral theory

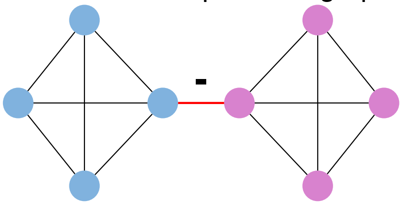
Consider our previous graph.



$$v_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \lambda_{min}(L) = 0.$$

Spectral theory

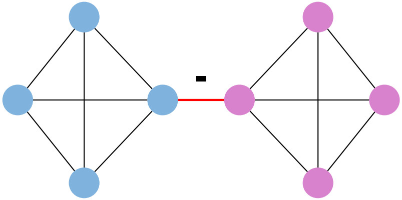
Consider our previous graph. Flip sign of one edge:



$$v_1 = \begin{pmatrix} -1 \\ -1 \\ -1 \\ -1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \lambda_{min}(L) = 0.$$

Spectral theory

Consider our previous graph. Flip sign of one edge:



$$v_1 = \begin{pmatrix} -1 \\ -1 \\ -1 \\ -1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \lambda_{min}(L) = 0.$$

This graph is **balanced**!

Spectral characterizations of balance

1. Connected and $\lambda_{min} = 0$ (or one zero-eigval per connected component).

Spectral theory

A taste of spectral analysis:

Lemma (Hou et al., 2003)

$\lambda_{max}(L(G)) \leq \lambda_{max}(L(G^{-}))$, where G^{-} is the all-negative graph.

Spectral theory

A taste of spectral analysis:

Lemma (Hou et al., 2003)

$\lambda_{max}(L(G)) \leq \lambda_{max}(L(G^-))$, where G^- is the all-negative graph.

Easy: $L(G^-)$ is the *signless Laplacian* of the underlying graph ($D_{|G|} + A_{|G|}$).So,

$x^T Lx =$

Spectral theory

A taste of spectral analysis:

Lemma (Hou et al., 2003)

$\lambda_{max}(L(G)) \leq \lambda_{max}(L(G^-))$, where G^- is the all-negative graph.

Easy: $L(G^-)$ is the *signless Laplacian* of the underlying graph ($D_{|G|} + A_{|G|}$).So,

$x^T L x = \sum_{(v_i, v_j) \in E} (x_i - \sigma(v_i, v_f) x_j)^2$

Spectral theory

A taste of spectral analysis:

Lemma (Hou et al., 2003)

$\lambda_{max}(L(G)) \leq \lambda_{max}(L(G^-))$, where G^- is the all-negative graph.

Easy: $L(G^-)$ is the *signless Laplacian* of the underlying graph ($D_{|G|} + A_{|G|}$).So,

$x^T L x = \sum_{(v_i,v_j)\in E}(x_i - \sigma(v_i, v_f)x_j)^2 \leq \sum_{(v_i,v_j)\in E}(|x_i| + |x_j|)^2.$

Spectral theory

A taste of spectral analysis:

Lemma (Hou et al., 2003)

$\lambda_{max}(L(G)) \leq \lambda_{max}(L(G^-))$, where G^- is the all-negative graph.

Easy: $L(G^-)$ is the *signless Laplacian* of the underlying graph $(D_{|G|} + A_{|G|})$.So,

$$x^T Lx = \sum_{(v_i,v_j)\in E}(x_i - \sigma(v_i, v_f)x_j)^2 \leq \sum_{(v_i,v_j)\in E}(|x_i| + |x_j|)^2.$$

Lemma (Hou et al., 2003)

$\lambda_{max}(L(G)) \leq 2(n-1)$, where n is the number of vertices.

Spectral theory

A taste of spectral analysis:

Lemma (Hou et al., 2003)

$\lambda_{max}(L(G)) \leq \lambda_{max}(L(G^{-}))$, where G^{-} is the all-negative graph.

Easy: $L(G^{-})$ is the *signless Laplacian* of the underlying graph $(D_{|G|} + A_{|G|})$.So,

$x^T L x = \sum_{(v_i,v_j)\in E}(x_i - \sigma(v_i, v_f)x_j)^2 \leq \sum_{(v_i,v_j)\in E}(|x_i| + |x_j|)^2$.

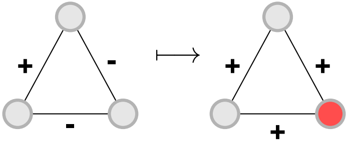
Lemma (Hou et al., 2003)

$\lambda_{max}(L(G)) \leq 2(n-1)$, where n is the number of vertices.

Easy: $\lambda_{max}(G) = \lambda_{max}(D - A) \leq \lambda_{max}(D_G) + \lambda_{max}(-A_G) \leq n - 1 + n - 1$

Switching

Switch $S \subset V$: flip edges between S and $V \setminus S$.

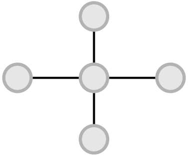


The spectrum is invariant w.r.t. switching.

$$A' = SAS^{-1}, \text{ where } S = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}.$$

Spectral theory

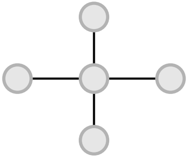
Example: Star graph



- Number of possible graphs: $2^{|E|}$
- Number of non-isomorphic graphs: $|E|$
- Number of distinct spectra: ?

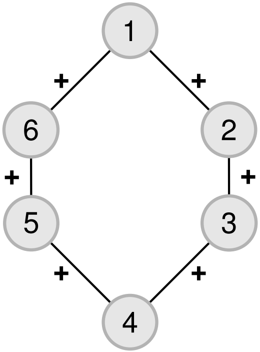
Spectral theory

Example: Star graph



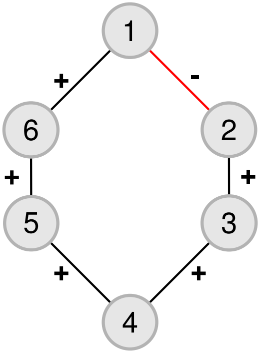
- Number of possible graphs: $2^{|E|}$
- Number of non-isomorphic graphs: $|E|$
- Number of distinct spectra: **Just the one!**

Example: cycle



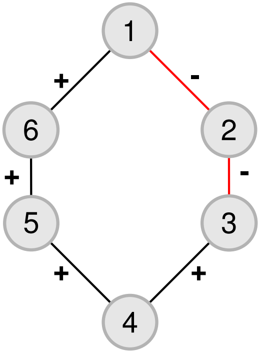
- Spectrum: $(0, 1, 1, 3, 3, 4)$
-
-

Example: cycle



-
- Spectrum: $(0.27, 0.27, 2, 2, 3.73, 3.73)$
-

Example: cycle

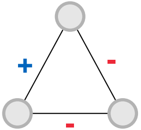


-
-
- Spectrum: $(0, 1, 1, 3, 3, 4)$

Switching

Spectral characterizations of balance

1. Connected and $\lambda_{min} = 0$ (or one zero-eigval per connected component).



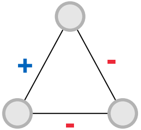
$Lv_1 = 0$

What's more: $\lambda_{min}(G) \leq \lambda_{max}(H)$, where H is the smallest subgraph to remove to make G balanced Li and Li (2016).

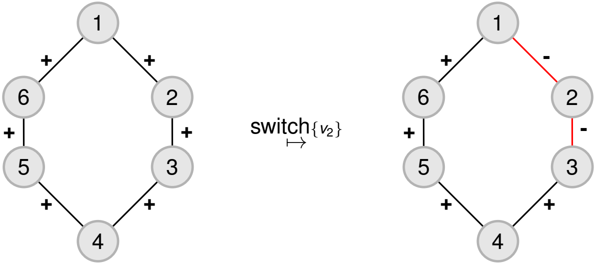
Switching

Spectral characterizations of balance

1. Connected and $\lambda_{min} = 0$ (or one zero-eigval per connected component).
2. Spectrum of G = spectrum of $|G|$ (**underlying** graph) (Acharya, 1980)



$Lv_1 = 0$

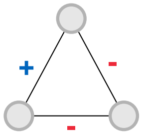


What's more: $\lambda_{min}(G) \leq \lambda_{max}(H)$, where H is the smallest subgraph to remove to make G balanced Li and Li (2016).

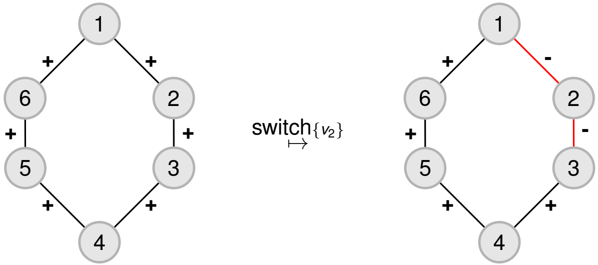
Switching

Spectral characterizations of balance

1. Connected and $\lambda_{min} = 0$ (or one zero-eigval per connected component).
2. Spectrum of G = spectrum of $|G|$ (**underlying** graph) (Acharya, 1980)
3. Switching equivalent to all-positive.



$Lv_1 = 0$



What's more: $\lambda_{min}(G) \leq \lambda_{max}(H)$, where H is the smallest subgraph to remove to make G balanced Li and Li (2016).

Frustration

Frustration

Frustration number (index) : N. of vertices (edges) that need to be removed to achieve balance.

Frustration

Frustration number (index) : N. of vertices (edges) that need to be removed to achieve balance.

Terrible nomenclature, if you ask me...

Proposal: vertex frustration f_v, edge frustration f_e.

Frustration

Frustration number (index) : N. of vertices (edges) that need to be removed to achieve balance.

Terrible nomenclature, if you ask me...

Proposal: vertex frustration f_v, edge frustration f_e.

Spectral frustration inequalities

$$\lambda_{min}(L) \leq f_v \leq f_e.$$

Spectral frustration inequalities

Spectral frustration inequalities

(Belardo, 2014)

$$\lambda_{min}(L) \leq f_v \leq f_e.$$

Spectral frustration inequalities

Spectral frustration inequalities

(Belardo, 2014)

$$\lambda_{min}(L) \leq f_v \leq f_e.$$

First inequality: start with and easy bound:

$$\lambda_{min}(L) \leq f_e$$

(try $\mathbf{1}_\pm^T L \mathbf{1}_\pm$, where $\mathbf{1}_\pm =$ partition indicator.)

Spectral frustration inequalities

Spectral frustration inequalities

(Belardo, 2014)

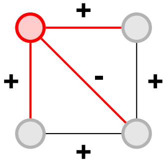
$$\lambda_{min}(L) \leq f_v \leq f_e.$$

First inequality: start with and easy bound:

$$\lambda_{min}(L) \leq f_e$$

(try $\mathbf{1}_\pm^T L \mathbf{1}_\pm$, where $\mathbf{1}_\pm$ = partition indicator.)

Second inequality: trivial.



Frustration

How hard is vertex frustration?

Frustration

How hard is vertex frustration?

Finding f_v is finding a **good submatrix...**

Frustration

How hard is vertex frustration?

Finding f_v is finding a **good submatrix**...

A good submatrix is hard to find (Bartholdi III, 1982)

It is **NP**-hard to find a submatrix with property π, where π is non-trivial, hereditary and holds for permutation matrices.

Frustration

How hard is vertex frustration?

Finding f_v is finding a **good submatrix...**

A good submatrix is hard to find (Bartholdi III, 1982)

It is **NP**-hard to find a submatrix with property π, where π is non-trivial, hereditary and holds for permutation matrices.

π: There exists $S = \mathit{diag}(\mathbf{1}_{\pm})$ such that $SAS = A^{+}$ (that is, the submatrix A switches to all positive).

In particular, finding f_v is finding submatrix of the adj. matrix satisfying π.

Frustration

How hard is vertex frustration?

Dual: find largest balanced subgraph.

> Admits **2-approximation** on complete graphs (Bai and Wu, 2012).

Other than that, not that much is known.

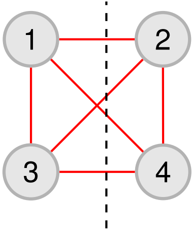
Frustration

How hard is edge frustration?

Frustration

How hard is edge frustration?

Consider an **all-negative** signed graph. Finding the minimum f_e is finding the MAXCUT, and thus **NP**-complete.



Corollary: **all-negative** is balanced $\Leftrightarrow$ it is bipartite.

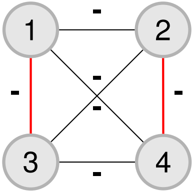
Frustration

How hard is edge frustration?

Frustration

How hard is edge frustration?

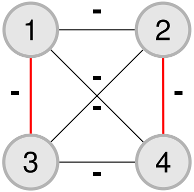
Every signed graph $G = (V, E)$ contains a balanced subgraph with at least $\frac{|E|}{2} + \frac{|V|-1}{4}$ edges.



Frustration

How hard is edge frustration?

Every signed graph $G = (V, E)$ contains a balanced subgraph with at least $\frac{|E|}{2} + \frac{|V|-1}{4}$ edges.

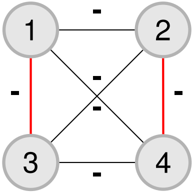


f_e is (UG)-hard to approximate to any constant.

Frustration

How hard is edge frustration?

Every signed graph $G = (V, E)$ contains a balanced subgraph with at least $\frac{|E|}{2} + \frac{|V|-1}{4}$ edges.

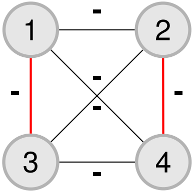


f_e is (UG)-hard to approximate to any constant.

- FPT: find subgraph of $|E| - k$ in $\mathcal{O}(2^k m^2)$ (Hüffner et al., 2007).

Frustration

How hard is edge frustration?



Every signed graph $G = (V, E)$ contains a balanced subgraph with at least $\frac{|E|}{2} + \frac{|V|-1}{4}$ edges.

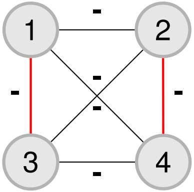
f_e is (UG)-hard to approximate to any constant.

- FPT: find subgraph of $|E| - k$ in $\mathcal{O}(2^k m^2)$ (Hüffner et al., 2007).
- FPT: decide if subgraph of $\frac{|E|}{2} + \frac{|V|-1}{4} + \frac{k}{4}$ in $\mathcal{O}(8^k (kn)^{\mathcal{O}(1)})$ (Crowston et al., 2013).

Frustration

How hard is edge frustration?

Every signed graph $G = (V, E)$ contains a balanced subgraph with at least $\frac{|E|}{2} + \frac{|V|-1}{4}$ edges.



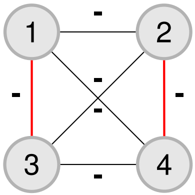
f_e is (UG)-hard to approximate to any constant.

- FPT: find subgraph of $|E| - k$ in $\mathcal{O}(2^k m^2)$ (Hüffner et al., 2007).
- FPT: decide if subgraph of $\frac{|E|}{2} + \frac{|V|-1}{4} + \frac{k}{4}$ in $\mathcal{O}(8^k (kn)^{\mathcal{O}(1)})$ (Crowston et al., 2013).
- Dual: $\mathcal{O}(\sqrt{\log n})$ (Agarwal et al., 2005), (MinUnCut problem).

Frustration

How hard is edge frustration?

Every signed graph $G = (V, E)$ contains a balanced subgraph with at least $\frac{|E|}{2} + \frac{|V|-1}{4}$ edges.



f_e is (UG)-hard to approximate to any constant.

- FPT: find subgraph of $|E| - k$ in $\mathcal{O}(2^k m^2)$ (Hüffner et al., 2007).
- FPT: decide if subgraph of $\frac{|E|}{2} + \frac{|V|-1}{4} + \frac{k}{4}$ in $\mathcal{O}(8^k (kn)^{\mathcal{O}(1)})$ (Crowston et al., 2013).
- Dual: $\mathcal{O}(\sqrt{\log n})$ (Agarwal et al., 2005), (MinUnCut problem).
- Dual: $\mathcal{O}(k \log k)$ (Avidor and Langberg, 2007).

Frustration

Reduction example #1

MinUnCut problem

Given a graph $G = (V, E)$ find a cut that minimizes the number of uncut edges.

Frustration

Reduction example #1

MinUnCut problem

Given a graph $G = (V, E)$ find a cut that minimizes the number of uncut edges.

MinUnCut problem (alternative formulation)

Consider boolean variables $b_1, \ldots, b_n$ and a set of constraints of the form $b_i \oplus b_j = 0$ and $b_i \oplus b_j = 1$. The goal is to minimize the number of unsatisfied constraints.

MINUNCUT problem

Given a graph $G = (V, E)$ find a cut that minimizes the number of uncut edges.

MINUNCUT problem (alternative formulation)

Consider boolean variables $b_1, \ldots, b_n$ and a set of constraints of the form $b_i \oplus b_j = 0$ and $b_i \oplus b_j = 1$. The goal is to minimize the number of unsatisfied constraints.

Reduction to f_e:

- $b_i \mapsto v_i$,
- $b_i \oplus b_j = 1 \mapsto$ 

- $b_i \oplus b_j = 0 \mapsto$ 

$\mathcal{O}(\sqrt{\log n})$-approximation (Agarwal et al., 2005).

EdgeBipartization problem

Given a graph $G = (V, E)$ delete the minimum number of edges to make it bipartite.

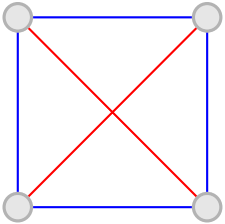
Frustration

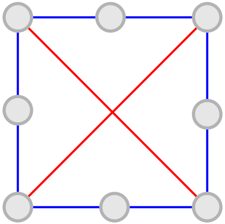
Reduction example #2

EDGEBIPARTIZATION problem

Given a graph $G = (V, E)$ delete the minimum number of edges to make it bipartite.

Reduction of f_e: insert a vertex into every **+** edge.

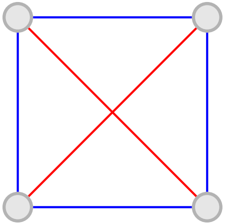


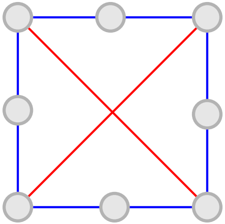


EDGEBIPARTIZATION problem

Given a graph $G = (V, E)$ delete the minimum number of edges to make it bipartite.

Reduction of f_e: insert a vertex into every **+** edge.





$\mathcal{O}(k \log k)$-approximation (Avidor and Langberg, 2007).

Other concepts

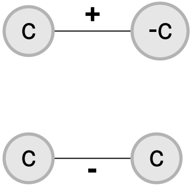
Signed graph coloring

$$
\begin{array}{rrll r}
\text{Coloring:} & \kappa : V \mapsto \mathbb{N} & \text{s.t.} & \forall (u,v) \in E & \kappa(u) \neq \kappa(v) \\
\text{Signed coloring:} & \kappa_{\pm} : V \mapsto \mathbb{Z} & \text{s.t.} & \forall (u,v) \in E & \kappa_{\pm}(u) \neq \sigma(uv)\kappa_{\pm}(v)
\end{array}
$$

Signed graph coloring

Coloring: $\kappa : V \mapsto \mathbb{N}$ s.t. $\forall (u,v) \in E$ $\kappa(u) \neq \kappa(v)$

Signed coloring: $\kappa_{\pm} : V \mapsto \mathbb{Z}$ s.t. $\forall (u,v) \in E$ $\kappa_{\pm}(u) \neq \sigma(uv)\kappa_{\pm}(v)$



Motivation: switching-friendly.

If G is balanced, $\chi(G) = \chi(|G|)$.

(Zaslavsky, 1982)

(Máčajová et al., 2014)

Signed graph coloring

Some results.

- $\chi(G)$: chromatic number of G.
- $|G|$: underlying graph.

Signed graph coloring

Some results.

- $\chi(G)$: chromatic number of G.
- $|G|$: underlying graph.

Chromatic number of a signed graph (Máčajová et al., 2014)

$\chi(G) \leq 2\chi(|G|) - 1.$

Signed graph coloring

Some results.

- $\chi(G)$: chromatic number of G.
- $|G|$: underlying graph.

Chromatic number of a signed graph (Máčajová et al., 2014)

$\chi(G) \leq 2\chi(|G|) - 1$.

2-colorable signed graphs (Máčajová et al., 2014)

$\chi(G) \leq 2$ iff G is antibalanced (switches to all-negative).

Signed graph coloring

Some results.

- $\chi(G)$: chromatic number of G.
- $|G|$: underlying graph.

Chromatic number of a signed graph (Máčajová et al., 2014)

$\chi(G) \leq 2\chi(|G|) - 1$.

2-colorable signed graphs (Máčajová et al., 2014)

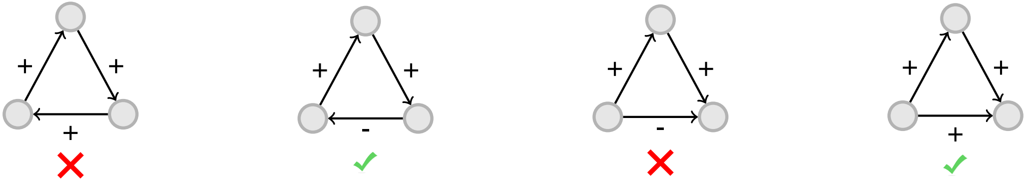
$\chi(G) \leq 2$ iff G is antibalanced (switches to all-negative).

Brooks theorem for signed graphs (Máčajová et al., 2014)

If G is not a balanced complete graph, a balanced odd circuit, or an unbalanced even circuit, then $\chi(G) \leq \Delta(G)$.

Status

Interpretation of signs in directed graphs: Status.



If Bob looks up to Alice and Alice looks up to Charlie,

then Charlie probably looks down on Bob.

What else

Some concepts we have overlooked.

- Signed-graphic matroid
- Matrix-tree theorem and the chromatic polynomial
- Two-graphs
- Full graphs
- Gain graphs

The bible: Zaslavsky's bibliography (Zaslavsky, 2012)

Problems and applications

Outline

Subgraph mining

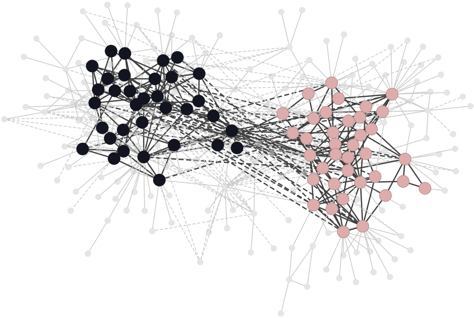
Subgraph mining

Goal

Find interesting subgraphs in a signed graph G

Some definitions of “interesting”:

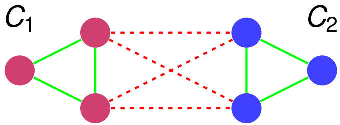
- maximal balanced subgraph
- polarized subgraph



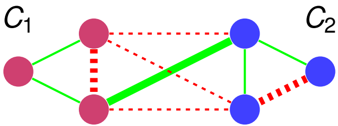
US Congress network (Bonchi et al., 2019)

Subgraph mining: balanced graphs vs. polarized graphs

Balanced graphs

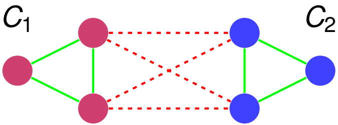


Polarized graphs: "noisy" edges are allowed

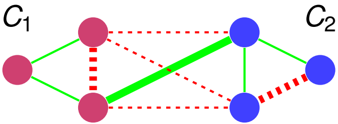


Subgraph mining: balanced graphs vs. polarized graphs

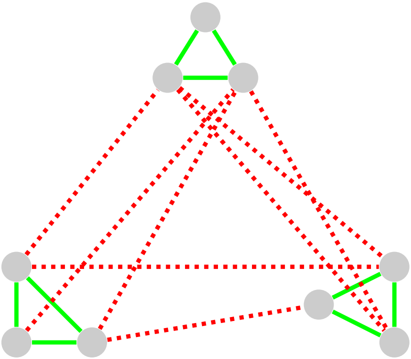
Balanced graphs



Polarized graphs: "noisy" edges are allowed



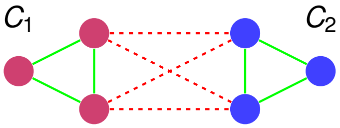
Polarized graphs: more than two groups



Finding maximum balanced subgraph (MBS)

Problem definition

- Input: a signed graph $G = (V, E^+, E^-)$
- Output: a maximum-cardinality vertex subset $U \subseteq V$ such that $G(U)$ is balanced

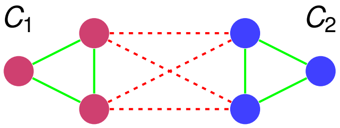


A balanced graph

Finding maximum balanced subgraph (MBS)

Problem definition

- Input: a signed graph $G = (V, E^+, E^-)$
- Output: a maximum-cardinality vertex subset $U \subseteq V$ such that $G(U)$ is balanced



A balanced graph

- An equivalent problem: remove the minimum number of vertices such that the remaining graph is balanced
- Solution size of MBS $=$ frustration index (defined in theory part)
- Edge-version of MBS: a balanced subgraph with maximum number of edges
- All these problems are **NP**-hard

Maximum balanced subgraph: applications in LP optimization

- LP in standard form

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax \le b \end{aligned}$$

- Each vertex represents a row in A
- $(i, j) \in E^-$ if $A_{ik} = A_{jk} \neq 0$ for some k
- $(i, j) \in E^+$ if $A_{ik} = -A_{jk} \neq 0$ for some k
- A balanced subgraph $\rightarrow$ a network sub-matrix
- Extracting network sub-matrices speed up LP solvers

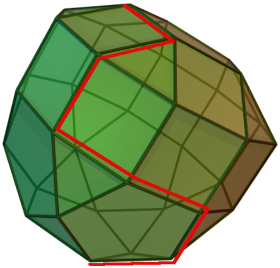


Image source: wiki.ubc.ca

Maximum balanced subgraph: applications in finance

- Can be used to find a risk-limiting strategy for portfolio definition Harary et al. (2002)
- Each node represents a security (a tradable asset)
- $(i, j) \in E^+$ if the value of two securities move in *same direction*
- $(i, j) \in E^-$ if the value of two securities move in *opposite direction*
- A balanced subgraph $\rightarrow$ a set of assets whose values are predictable $\rightarrow$ low investment risk

Image source: vecteezy.com

Maximum balanced subgraph: applications in political science

- Setting: find political parties with opposite political leaning
- Each node represents a politician
- Edges capture supportive/opposive relations
- Further applications:
  - understanding the nature of polarization
  - reducing polarization via applied intervention

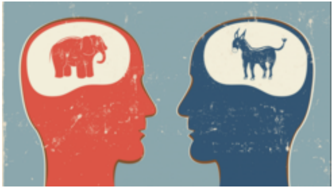


Image source: iStockphoto.com

MBS: Overview of existing results

Paper	graph type	approximation ratio	running time
Gülpinar et al. (2004)	general	—	$\mathcal{O}(m)$
Figueiredo et al. (2011)	general	exact	branch & bound
Bai and Wu (2012)	complete	2	$\mathcal{O}(n^3)$
Bai and Wu (2012)	complete	exact	$\mathcal{O}(1.28^k + n^3)$
Figueiredo and Frota (2014)	general	—	$\mathcal{O}(m)$
Ordozgoiti et al. (2020)	general	—	$\mathcal{O}(n^2)$ per iteration

- — indicates approximation “not known”, i.e., the algorithm is a heuristic
- k is the number of nodes to delete

Spanning-tree heuristic for MBS

Notations:

- Negative graph G^-: induced subgraph on the negative edges in G
- Positive graph G^+: induced subgraph on the positive edges in G
- $I(G)$: any maximal independent set of G

Spanning-tree heuristic for MBS

Notations:

- Negative graph G^-: induced subgraph on the negative edges in G
- Positive graph G^+: induced subgraph on the positive edges in G
- $I(G)$: any maximal independent set of G

High-level idea (Gülpinar et al., 2004)

1. Find a spanning tree T on G
2. Find a switch W such that T^W is all positive
3. Switch G by W, yielding G^W
4. Return $I(G^W)^-$

Spanning-tree heuristic: maximal independent set on G^-

Intuition 1

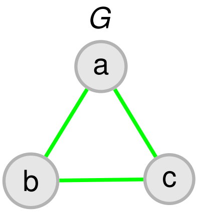
Any maximal independent set on G^- is balanced in G.

G G^- $\mathrm{I}(G^-)$

Spanning-tree heuristic: maximal independent set on G^-

Intuition 1

Any maximal independent set on G^- is balanced in G.



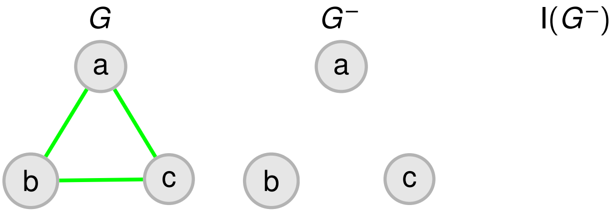
G^-

$\mathrm{I}(G^-)$

Spanning-tree heuristic: maximal independent set on G^-

Intuition 1

Any maximal independent set on G^- is balanced in G.

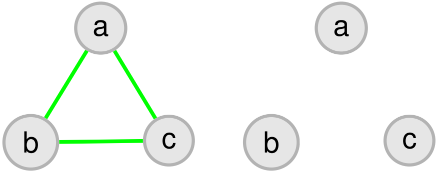


Spanning-tree heuristic: maximal independent set on G^-

Intuition 1

Any maximal independent set on G^- is balanced in G.

G $\qquad$ G^- $\qquad$ $\mathrm{I}(G^-)$



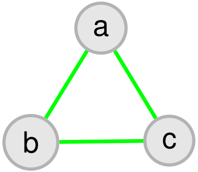
$\{a, b, c\}$

Spanning-tree heuristic: maximal independent set on G^-

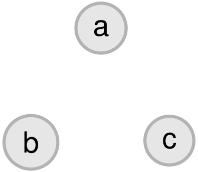
Intuition 1

Any maximal independent set on G^- is balanced in G.

G

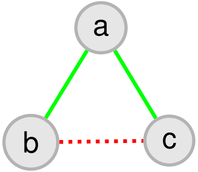


G^-



$\mathrm{I}(G^-)$

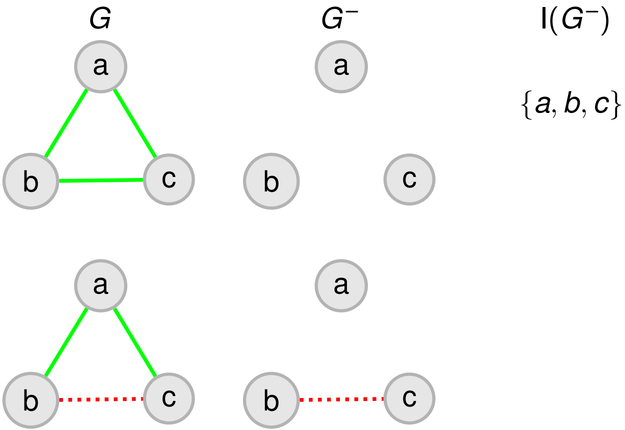
$\{a, b, c\}$



Spanning-tree heuristic: maximal independent set on G^-

Intuition 1

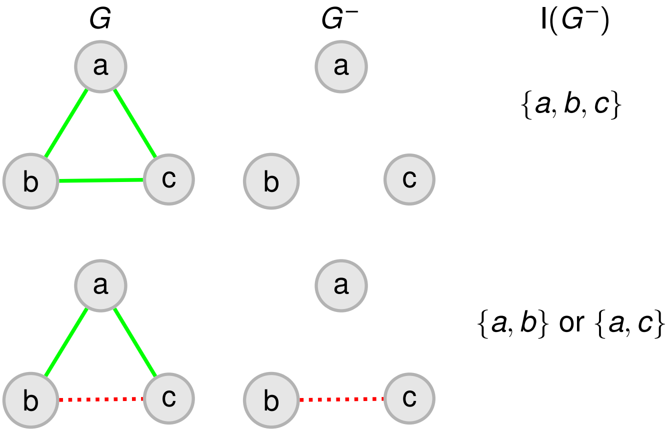
Any maximal independent set on G^- is balanced in G.



Spanning-tree heuristic: maximal independent set on G^-

Intuition 1

Any maximal independent set on G^- is balanced in G.



Spanning-tree heuristic: maximal independent set on G^-

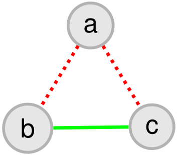
Quiz: Can we solve MBS optimally by maximizing $|I(G^-)|$?

Spanning-tree heuristic: maximal independent set on G^-

Quiz: Can we solve MBS optimally by maximizing $|I(G^-)|$?

No, a counter-example:

G



G^-

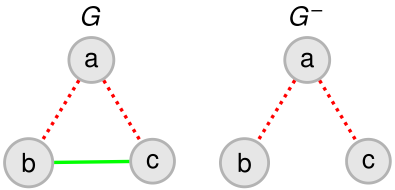
$\mathrm{I}(G^-)$ of maximum size

Expected solution: $\{a, b, c\}$

Spanning-tree heuristic: maximal independent set on G^-

Quiz: Can we solve MBS optimally by maximizing $|I(G^-)|$?

No, a counter-example:



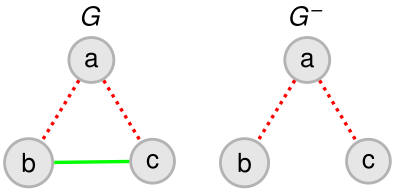
I(G^-) of maximum size

Expected solution: $\{a, b, c\}$

Spanning-tree heuristic: maximal independent set on G^-

Quiz: Can we solve MBS optimally by maximizing $|I(G^-)|$?

No, a counter-example:



$\mathrm{I}(G^-)$ of maximum size

$\{b, c\}$ ✗

Expected solution: $\{a, b, c\}$

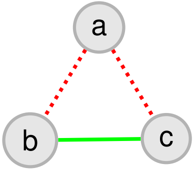
Spanning-tree heuristic: switch

Intuition 2

Switch G to expand size of $I(G^{-})$

If we switch on $\{a\}$:

G



Spanning-tree heuristic: switch

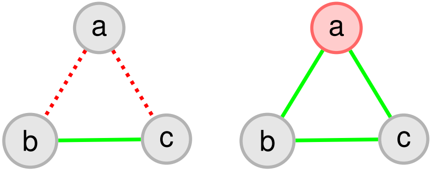
Intuition 2

Switch G to expand size of $I(G^-)$

If we switch on $\{a\}$:

new! $\downarrow$

G $\qquad$ $G^W, W = \{a\}$



Spanning-tree heuristic: switch

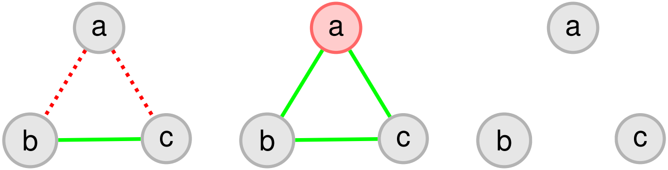
Intuition 2

Switch G to expand size of $I(G^-)$

If we switch on $\{a\}$:

new! $\downarrow$

G $\quad$ $G^W, W = \{a\}$ $\quad$ $(G^W)^-$



Spanning-tree heuristic: switch

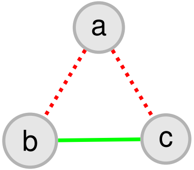
Intuition 2

Switch G to expand size of $I(G^{-})$

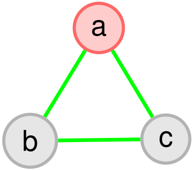
If we switch on $\{a\}$:

new! $\downarrow$

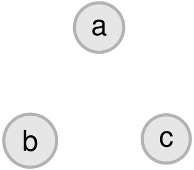
G



$G^W,\ W = \{a\}$



$(G^W)^{-}$



I(G^{-}) of maximum size

$\{a, b, c\}$ ✓

Spanning-tree heuristic: combining the previous ideas

An equivalent form of MBS

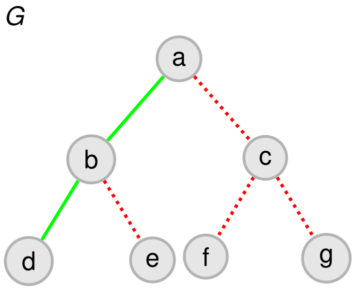
Find a switch W s.t. $\left|I\left(G^W\right)^-\right|$ is maximized (an **NP**-hard problem)

Spanning-tree heuristic: combining the previous ideas

An equivalent form of MBS

Find a switch W s.t. $\left|I\left(G^W\right)^-\right|$ is maximized (an **NP**-hard problem)

A tree is always balanced, i.e., there exists some W s.t. T^W is all positive



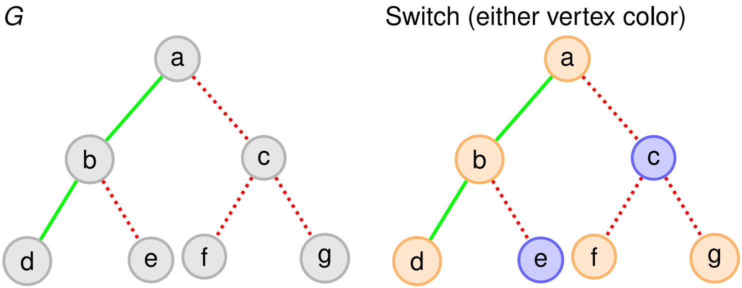
Quiz: How to find a switch that makes a tree all positive? Hint: use BFS

Spanning-tree heuristic: combining the previous ideas

An equivalent form of MBS

Find a switch W s.t. $\left|I\left(\left(G^{W}\right)^{-}\right)\right|$ is maximized (an **NP**-hard problem)

A tree is always balanced, i.e., there exists some W s.t. T^{W} is all positive



Quiz: How to find a switch that makes a tree all positive? Hint: use BFS

Spanning-tree heuristic for MBS

Algorithm (Gülpinar et al., 2004)

1. Find a spanning tree T on G # a tree is an easy case to solve

Spanning-tree heuristic for MBS

Algorithm (Gülpinar et al., 2004)

1. Find a spanning tree T on G # a tree is an easy case to solve
2. Find a switch W that makes T^W all positive # expands the solution size

Spanning-tree heuristic for MBS

Algorithm (Gülpinar et al., 2004)

1. Find a spanning tree T on G # a tree is an easy case to solve
2. Find a switch W that makes T^W all positive # expands the solution size
3. Use W to switch G, yielding G^W

Spanning-tree heuristic for MBS

Algorithm (Gülpinar et al., 2004)

1. Find a spanning tree T on G # a tree is an easy case to solve
2. Find a switch W that makes T^W all positive # expands the solution size
3. Use W to switch G, yielding G^W
4. Return maximal independent set on $(G^W)^-$ # $I\,(G^W)^-$ is balanced

A spectral method for MBS

Finding large balanced subgraphs in signed networks (Ordozgoiti et al., 2020)

Intuition

A balanced subgraph has $\lambda_{min}(L) = 0$.

Can we remove as few vertices as possible, such that $\lambda_{min}(L)$ of the remaining (connected) graph is zero?

A spectral method for MBS

Finding large balanced subgraphs in signed networks (Ordozgoiti et al., 2020)

Intuition

A balanced subgraph has $\lambda_{min}(L) = 0$.
Can we remove as few vertices as possible, such that $\lambda_{min}(L)$ of the remaining (connected) graph is zero?

A greedy algorithm: remove vertex v_i that improves balance the most, that is:

$$i = \arg\min_j \lambda_{min}(L \backslash v_j).$$

Intuition

A balanced subgraph has $\lambda_{min}(L) = 0$.
Can we remove as few vertices as possible, such that $\lambda_{min}(L)$ of the remaining (connected) graph is zero?

A greedy algorithm: remove vertex v_i that improves balance the most, that is:

$$i = \arg\min_j \lambda_{min}(L \backslash v_j).$$

Perturbation bound for fast computation: if $\mathbf{v}$ is the eigenvector for λ_{min} then

$$\lambda_1(L \backslash \mathbf{v}_i) \leq \frac{\lambda_1(L)(1 - 2\mathbf{v}_i^2) - \sum_{j \in N(i)} \mathbf{v}_j^2 + \mathbf{v}_i^2 d(i)}{1 - \mathbf{v}_i^2}.$$

A spectral method for MBS

Finding large balanced subgraphs in signed networks (Ordozgoiti et al., 2020)

Further speed-ups:

- Remove multiple vertices at once (extending perturbation bound)
- Estimate eigen-pair with gradient methods (LOBPCG)
- Low-rank updates
- Sampling: Solve the problem for $\sqrt{n}$ random subgraphs of size $\sqrt{n}$

A spectral method for MBS

Finding large balanced subgraphs in signed networks (Ordozgoiti et al., 2020)

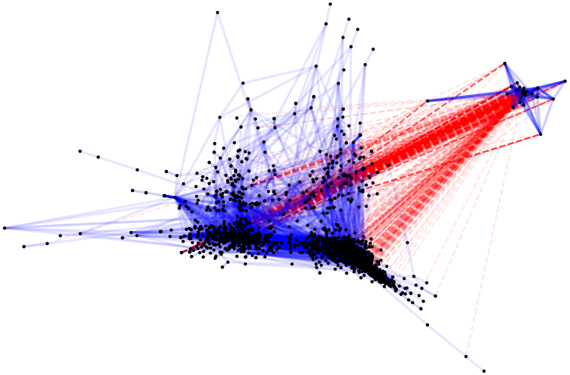
	HighlandTribes		Cloister		Congress		Bitcoin		TwitterReferendum	
method	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert V\rvert$	$\lvert E\rvert$
Timbal	**13**	35	**10**	**33**	**208**	**452**	**4 208**	**10 158**	**8 944**	**166 243**
Grasp	10	18	6	11	115	145	2 167	3 686	5 425	49 105
Ggmz	10	21	5	7	153	238	1 388	1 683	2 501	2 821
Eigen	12	**37**	8	27	11	16	7	17	132	6 140
	WikiElections		Slashdot		WikiConflict		WikiPolitics		Epinions	
Timbal	**3 786**	**18 550**	**42 205**	**96 460**	**48 136**	**356 204**	**63 252**	**218 360**	**62 010**	**169 894**
Grasp	1 752	4 416	23 289	40 511	18 576	82 726	31 561	81 557	28 189	63 250
Ggmz	713	771	16 389	17 867	6 137	9 145	23 342	37 098	21 009	25 013
Eigen	11	41	35	491	11	28	10	45	6	14

A spectral method for MBS

Finding large balanced subgraphs in signed networks (Ordozgoiti et al., 2020)

A graph of traders in a Bitcoin exchange

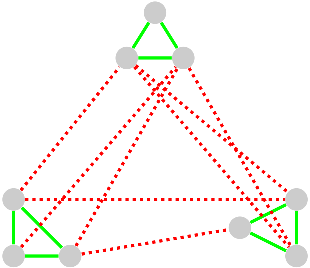
with trust / distrust relations



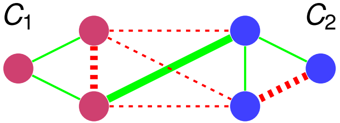
Polarized subgraph detection

Polarized subgraphs as an extension of balanced subgraphs

- Can have more than two components
- Permits the presence of noisy edges:
  - Positive edges between C_1 and C_2
  - Negative edges within C_1 or C_2



More than two components



With “noisy” edges (drawn in thick lines)

Polarized subgraph detection: problem dimensions

Polarized subgraph detection: problem dimensions

- What measure of polarization?

Polarized subgraph detection: problem dimensions

- What measure of polarization?
- How many groups inside a polarized subgraph?

  2-way or k-way polarized subgraph?

Polarized subgraph detection: problem dimensions

- What measure of polarization?
- How many groups inside a polarized subgraph?
  2-way or k-way polarized subgraph?
- How many polarized subgraphs to find: one or multiple?

Polarized subgraph detection: problem dimensions

- What measure of polarization?
- How many groups inside a polarized subgraph?

  2-way or k-way polarized subgraph?
- How many polarized subgraphs to find: one or multiple?
- Are seed nodes given? local or global community detection?

Polarized subgraph detection: problem dimensions

- What measure of polarization?
- How many groups inside a polarized subgraph?
  2-way or k-way polarized subgraph?
- How many polarized subgraphs to find: one or multiple?
- Are seed nodes given? local or global community detection?

Paper	num. groups	num. subgraphs	local / global	approximation guarantee
Chu et al. (2016)	k	≥ 1	global	-
Bonchi et al. (2019)	2	1	global	$\sqrt{n}$
Xiao et al. (2020)	2	≥ 1	local	$\sqrt{\text{OPT}}$

Polarized subgraph detection: multiple k-way subgraphs

Finding gangs in war from signed networks (Chu et al., 2016)

- Intuition of the polarization measure:
  1. In each group, many positive edges
  2. Between two groups, many negative edges
  3. Between two groups, few nodes are shared

Polarized subgraph detection: multiple k-way subgraphs

Finding gangs in war from signed networks (Chu et al., 2016)

- Intuition of the polarization measure:
  1. In each group, many positive edges
  2. Between two groups, many negative edges
  3. Between two groups, few nodes are shared
- Objective in matrix form:

$$tr(X^T A^+ X) + \alpha \bar{tr}(X^T A^- X) - \beta \bar{tr}(X^T X)$$

where $X \in \{0, 1\}^{n \times k}$, e.g., each column maps to a group

Polarized subgraph detection: multiple k-way subgraphs

Finding gangs in war from signed networks (Chu et al., 2016)

- Intuition of the polarization measure:
  1. In each group, many positive edges
  2. Between two groups, many negative edges
  3. Between two groups, few nodes are shared
- Objective in matrix form:

$$tr(X^T A^+ X) + \alpha \bar{tr}(X^T A^- X) - \beta \bar{tr}(X^T X)$$

where $X \in \{0, 1\}^{n \times k}$, e.g., each column maps to a group

- Main idea of algorithm:
  - Relax X to be continuous
  - The relaxed problem is non-convex $\rightarrow$ multiple local optimums
  - Each local optimum corresponds to a polarized subgraph
  - Solved by gradient-descent methods

Polarized subgraph detection: single 2-way subgraph

Discovering polarized communities in signed networks (Bonchi et al., 2019)

- Intuition of the polarization measure:
  1. In each group, many positive edges
  2. Between two groups, many negative edges
  3. The subgraph is dense in terms of the number of nodes

Polarized subgraph detection: single 2-way subgraph

Discovering polarized communities in signed networks (Bonchi et al., 2019)

- Intuition of the polarization measure:
  1. In each group, many positive edges
  2. Between two groups, many negative edges
  3. The subgraph is dense in terms of the number of nodes

- Objective in matrix form:

$$\max_{x} \frac{x^T A x}{x^T x} \qquad \text{(\textbf{NP}-hard problem)}$$

where $x \in \{-1, 0, 1\}^n$ is used to encode the subgraph

Polarized subgraph detection: single 2-way subgraph

Discovering polarized communities in signed networks (Bonchi et al., 2019)

- Intuition of the polarization measure:
  1. In each group, many positive edges
  2. Between two groups, many negative edges
  3. The subgraph is dense in terms of the number of nodes

- Objective in matrix form:

$$\max_x \frac{x^T A x}{x^T x} \qquad \text{(\textbf{NP}-hard problem)}$$

  where $x \in \{-1, 0, 1\}^n$ is used to encode the subgraph

- Spectral algorithms:
  - Relax x to be continuous
  - The relaxed problem is solved by finding the leading eigenvector
  - Randomized $\sqrt{n}$-approximation based on rounding the leading eigenvector

Polarized subgraph detection: local 2-way subgraph

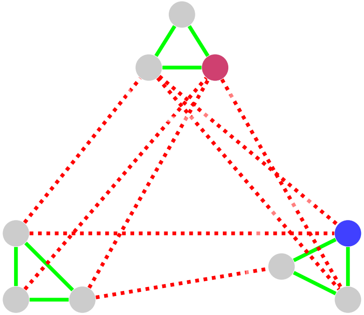
Searching for polarization in signed graphs: a local spectral approach (Xiao et al., 2020)

- Input: a signed graph and two sets of seed nodes
- Output: a polarized subgraph that contains the seeds

Polarized subgraph detection: local 2-way subgraph

Searching for polarization in signed graphs: a local spectral approach (Xiao et al., 2020)

- Input: a signed graph and two sets of seed nodes
- Output: a polarized subgraph that contains the seeds

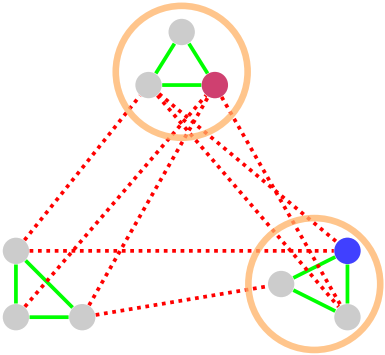


First seed set = {●}, Second seed set = {●}

Polarized subgraph detection: local 2-way subgraph

Searching for polarization in signed graphs: a local spectral approach (Xiao et al., 2020)

- Input: a signed graph and two sets of seed nodes
- Output: a polarized subgraph that contains the seeds

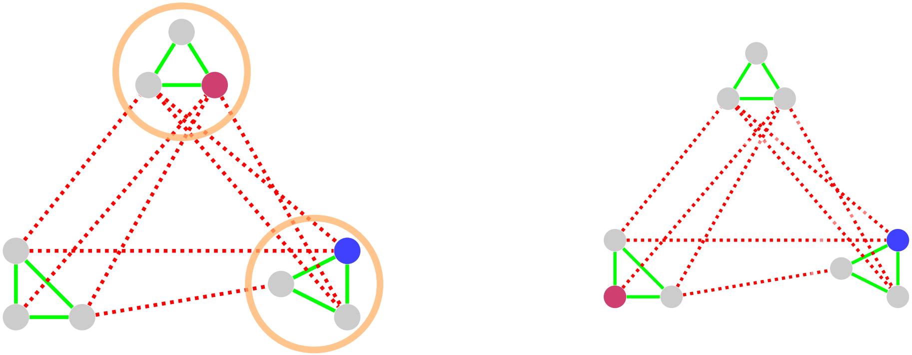


First seed set = { }, Second seed set = { }

Polarized subgraph detection: local 2-way subgraph

Searching for polarization in signed graphs: a local spectral approach (Xiao et al., 2020)

- Input: a signed graph and two sets of seed nodes
- Output: a polarized subgraph that contains the seeds

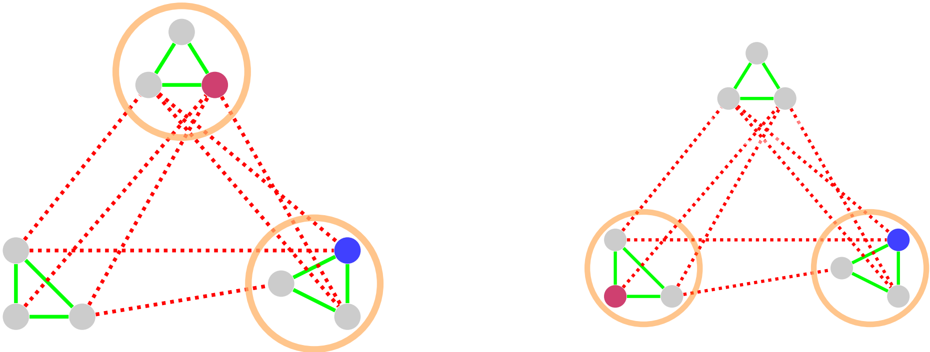


First seed set $= \{\bullet\}$, Second seed set $= \{\bullet\}$

Polarized subgraph detection: local 2-way subgraph

Searching for polarization in signed graphs: a local spectral approach (Xiao et al., 2020)

- Input: a signed graph and two sets of seed nodes
- Output: a polarized subgraph that contains the seeds



First seed set = {●}, Second seed set = {●}

Polarized subgraph detection: local 2-way subgraph

Searching for polarization in signed graphs: a local spectral approach (Xiao et al., 2020)

- Intuition of the polarization measure:
  1. In each group, few negative edges
  2. Between two groups, few positive edges
  3. The subgraph is sparse-connected with the remaining graph

  Combined: the *fraction* of the above edges in the subgraph is small

Polarized subgraph detection: local 2-way subgraph

Searching for polarization in signed graphs: a local spectral approach (Xiao et al., 2020)

- Intuition of the polarization measure:
  1. In each group, few negative edges
  2. Between two groups, few positive edges
  3. The subgraph is sparse-connected with the remaining graph

  Combined: the *fraction* of the above edges in the subgraph is small
- Problem in matrix form:

$$\begin{array}{cc} \min_{x \in \{-1,0,1\}^n} & \frac{x^T L x}{x^T D x} \\ \text{s.t.} & s^T D x \geq \kappa \end{array}$$

  where s encodes the seed information and x correlates with s

Polarized subgraph detection: local 2-way subgraph

Searching for polarization in signed graphs: a local spectral approach (Xiao et al., 2020)

- Intuition of the polarization measure:
  1. In each group, few negative edges
  2. Between two groups, few positive edges
  3. The subgraph is sparse-connected with the remaining graph

  Combined: the *fraction* of the above edges in the subgraph is small
- Problem in matrix form:

$$
\begin{array}{cc}
\min_{x \in \{-1,0,1\}^n} & \frac{x^T L x}{x^T D x} \\
\text{s.t.} & s^T D x \geq \kappa
\end{array}
$$

  where s encodes the seed information and x correlates with s
- A spectral algorithm:
  - Relax x to be continuous
  - The relaxed problem is a locally-biased eigen problem
  - An approximation algorithm based on rounding the locally-based eigenvector.

Polarized subgraph detection: local 2-way subgraph

Searching for polarization in signed graphs: a local spectral approach (Xiao et al., 2020)

Case study: overlapping communities in English-word graph

- Dataset: a graph of English words
- Word relations: synonym and antonym

Polarized subgraph detection: local 2-way subgraph

Searching for polarization in signed graphs: a local spectral approach (Xiao et al., 2020)

Case study: overlapping communities in English-word graph

- Dataset: a graph of English words
- Word relations: synonym and antonym

Two meanings of "fair"

- "without cheating", e.g., a fair game
- "not excessive", e.g., fair amount of time

Polarized subgraph detection: local 2-way subgraph

Searching for polarization in signed graphs: a local spectral approach (Xiao et al., 2020)

Case study: overlapping communities in English-word graph

- Dataset: a graph of English words
- Word relations: synonym and antonym

Two meanings of "fair"

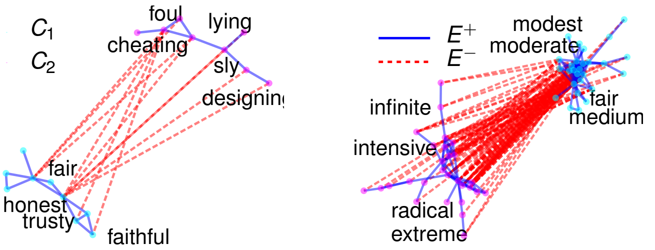
- "without cheating", e.g., a fair game
- "not excessive", e.g., fair amount of time

Question

Can we find the synonyms and antonyms w.r.t each meaning of "fair"?

Polarized subgraph detection: local 2-way subgraph

Searching for polarization in signed graphs: a local spectral approach (Xiao et al., 2020)



(a) *fair as without cheating*

$S_1 = \{fair, honest\}$

$S_2 = \{cheating\}$

(b) *fair as not excessive*

$S_1 = \{fair, modest\}$

$S_2 = \{extreme\}$

Subgraph mining: summary

- Types of subgraphs are discussed:
  - Maximal balanced subgraph:
    - studied mainly in theoretical computer science
  - Polarized subgraph:
    - not necessarily balanced, with different objective functions
- Applications: from solving linear program solver to analyzing societal debates
- An emerging and timely topic when polarization is harming our society !

Coffee break (5 minutes).

Outline

Introduction

Theory of signed networks

Problems and applications

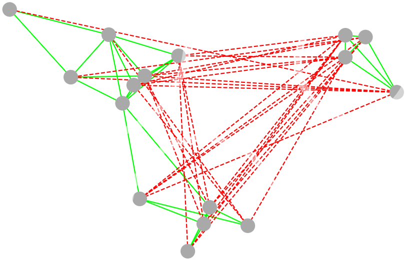
- Subgraph mining
- Graph partitioning
- Correlation clustering
- Link prediction and link classification
- Network dynamics
- Graph embedding and representation learning

Conclusions and future directions

Graph partitioning

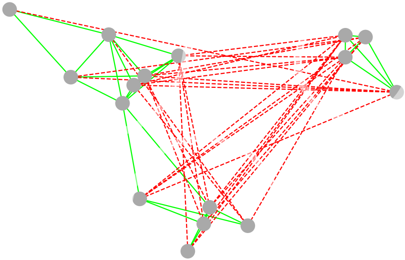
Graph partitioning

- General goal: partition all vertices into disjoint sets



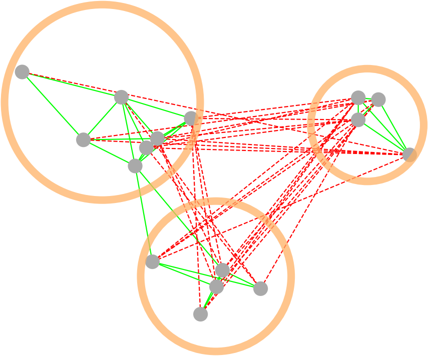
Graph partitioning

- General goal: partition all vertices into disjoint sets
- such that
  - edges *inside* each partition are mostly positive
  - edges *between* two partitions are mostly negative



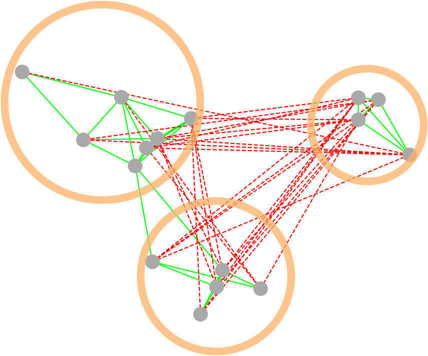
Graph partitioning

- General goal: partition all vertices into disjoint sets
- such that
  - edges *inside* each partition are mostly positive
  - edges *between* two partitions are mostly negative



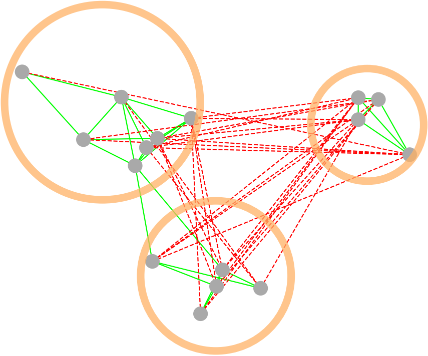
Graph partitioning

- General goal: partition all vertices into disjoint sets
- such that
  - edges *inside* each partition are mostly positive
  - edges *between* two partitions are mostly negative
- Example applications:
  - Identifying proteins that have synergetic interaction
  - Finding cohesive political parties or social-media user groups



Graph partitioning

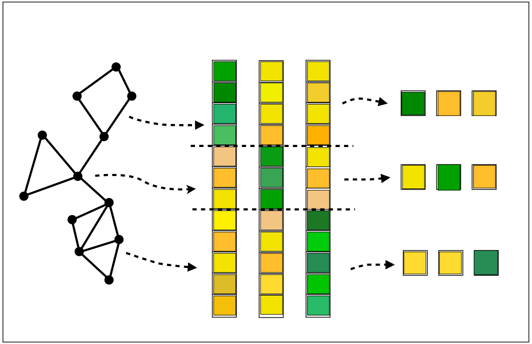
- General goal: partition all vertices into disjoint sets
- such that
  - edges *inside* each partition are mostly positive
  - edges *between* two partitions are mostly negative
- Example applications:
  - Identifying proteins that have synergetic interaction
  - Finding cohesive political parties or social-media user groups
- Our focus: spectral methods



Recap: spectral methods on unsigned graphs

Heuristics for k-way partitioning

1. Construct some Laplacian matrix of the graph
2. $[\mathbf{v}_1, \ldots, \mathbf{v}_k] \leftarrow$ the first k eigenvectors of the Laplacian
3. Run a feature-based clustering algorithm such as k-means



Spectral clustering methods on signed graphs

Similar skeleton for k-way partitioning

- Formulate some Laplacian L
- $[u_1, \ldots, u_k] \leftarrow$ the leading k eigenvectors of L, corresponding to smallest eigenvalues)
- Run a feature-based clustering algorithm on $[u_1, \ldots, u_k]$

The Laplacian matrix matters

Spectral clustering methods on signed graphs

Similar skeleton for k-way partitioning

- Formulate some Laplacian L
- $[u_1, \ldots, u_k] \leftarrow$ the leading k eigenvectors of L, corresponding to smallest eigenvalues)
- Run a feature-based clustering algorithm on $[u_1, \ldots, u_k]$

The Laplacian matrix matters

- Arithmetic-mean Laplacian (Kunegis et al., 2010)
- Geometric-mean Laplacian (Mercado et al., 2016)
- Power-mean Laplacian (Mercado et al., 2019)

"Classic" Laplacian

Spectral analysis of signed graphs for clustering, prediction and visualization (Kunegis et al., 2010)

Problem

Find a partition of (X, Y) over all nodes that

$$\text{minimizes} \quad \text{cut}(X, Y) \left(\frac{1}{|X|} + \frac{1}{|Y|} \right)$$

where

$$\text{cut}(X, Y) =) \left(2\,\text{cut}^{+}(X, Y) + \text{cut}^{-}(X) + \text{cut}^{-}(Y)\right)$$

The objective is called signed ratio cut

“Classic” Laplacian

Spectral analysis of signed graphs for clustering, prediction and visualization (Kunegis et al., 2010)

Define

$$L = D - A$$

An equivalent problem in matrix form

$$\begin{array}{lll} \text{minimize} & u^T L u & \\ \text{s.t.} & u_i = \frac{1}{2}\left(\sqrt{|X|/|Y|} + \sqrt{|Y|/|X|}\right) & \text{if } i \in X \\ & u_i = -\frac{1}{2}\left(\sqrt{|X|/|Y|} + \sqrt{|Y|/|X|}\right) & \text{if } i \in Y \end{array}$$

“Classic” Laplacian

Spectral analysis of signed graphs for clustering, prediction and visualization (Kunegis et al., 2010)

Define

$$L = D - A$$

An equivalent problem in matrix form

$$\begin{aligned}
\text{minimize} \quad & u^T L u \\
\text{s.t.} \quad & u_i = \frac{1}{2}\left(\sqrt{|X|/|Y|} + \sqrt{|Y|/|X|}\right) \quad \text{if } i \in X \\
& u_i = -\frac{1}{2}\left(\sqrt{|X|/|Y|} + \sqrt{|Y|/|X|}\right) \quad \text{if } i \in Y
\end{aligned}$$

- If u is relaxed that $u^T u = 1$,

  the problem is equivalent to standard eigen-problem

“Classic” Laplacian

Spectral analysis of signed graphs for clustering, prediction and visualization (Kunegis et al., 2010)

If normalized Laplacian is used

$$\mathcal{L} = I - D^{-1/2} A D^{-1/2},$$

“Classic” Laplacian

Spectral analysis of signed graphs for clustering, prediction and visualization (Kunegis et al., 2010)

If normalized Laplacian is used

$$\mathcal{L} = I - D^{-1/2} A D^{-1/2},$$

the eigen-problem on $\mathcal{L}$ corresponds to minimizing signed normalized cut:

$$\mathrm{cut}(X, Y)\left(\frac{1}{vol(X)} + \frac{1}{vol(Y)}\right)$$

Geometric-mean Laplacian: decomposing classic Laplacian

Spectral clustering using geometric mean Laplacian (Mercado et al., 2016)

Classic Laplacian can be re-written:

$$
\begin{aligned}
L &= D - A \\
&= D - (A^{+} - A^{-}) \\
&= (D^{+} - A^{+}) + (D^{-} + A^{-}) \\
&= L^{+} + Q^{-}
\end{aligned}
$$

Geometric-mean Laplacian: decomposing classic Laplacian

Spectral clustering using geometric mean Laplacian (Mercado et al., 2016)

Classic Laplacian can be re-written:

$$\begin{aligned} L &= D - A \\ &= D - (A^+ - A^-) \\ &= (D^+ - A^+) + (D^- + A^-) \\ &= L^+ + Q^- \end{aligned}$$

- Laplacian on G^+: $L^+ = D^+ - A^+$ "Positive Laplacian"
- Laplacian on G^-: $Q^- = D^- + A^-$ "Negative Laplacian"

Geometric-mean Laplacian: decomposing classic Laplacian

Spectral clustering using geometric mean Laplacian (Mercado et al., 2016)

Classic Laplacian can be re-written:

$$\begin{aligned} L &= D - A \\ &= D - (A^+ - A^-) \\ &= (D^+ - A^+) + (D^- + A^-) \\ &= L^+ + Q^- \end{aligned}$$

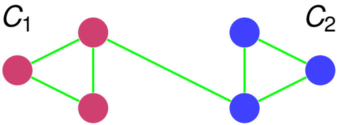
- Laplacian on G^+: $L^+ = D^+ - A^+$ "Positive Laplacian"
- Laplacian on G^-: $Q^- = D^- + A^-$ "Negative Laplacian"
- L is twice the arithmetic mean of L^+ and Q^- and thus, is called *arithmetic-mean Laplacian*

Geometric-mean Laplacian: decomposing classic Laplacian

Spectral clustering using geometric mean Laplacian (Mercado et al., 2016)

G^+: assortative case

G^+ is well-separated
$\rightarrow \lambda_2(L^+)$ is small
$\rightarrow v_2(L^+)$ is informative



If G^+ is disconnected, $\lambda_2(L^+) = 0$

Geometric-mean Laplacian: decomposing classic Laplacian

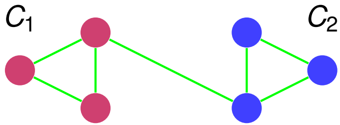
Spectral clustering using geometric mean Laplacian (Mercado et al., 2016)

G^+: assortative case

G^+ is well-separated

$\rightarrow \lambda_2(L^+)$ is small

$\rightarrow v_2(L^+)$ is informative



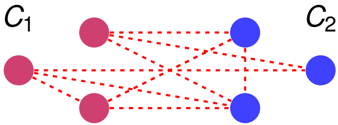
If G^+ is disconnected, $\lambda_2(L^+) = 0$

G^-: Disassortative case

G^- is close to bipartite

$\rightarrow \lambda_1(Q^-)$ is small

$\rightarrow v_1(Q^-)$ is informative



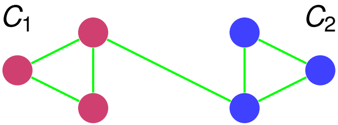
if G^- is bipartite, $\lambda_1(Q^-) = 0$

Geometric-mean Laplacian: decomposing classic Laplacian

Spectral clustering using geometric mean Laplacian (Mercado et al., 2016)

G^+: assortative case

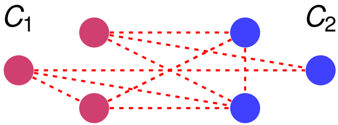
G^+ is well-separated
$\rightarrow \lambda_2(L^+)$ is small
$\rightarrow v_2(L^+)$ is informative



If G^+ is disconnected, $\lambda_2(L^+) = 0$

G^-: Disassortative case

G^- is close to bipartite
$\rightarrow \lambda_1(Q^-)$ is small
$\rightarrow v_1(Q^-)$ is informative



if G^- is bipartite, $\lambda_1(Q^-) = 0$

The above intuition also applies to normalized Laplacians:

$$L^+_{sym} = (D^+)^{-1/2} L^+ (D^+)^{-1/2} \qquad Q^-_{sym} = (D^-)^{-1/2} Q^- (D^-)^{-1/2}$$

Geometric-mean Laplacian

Spectral clustering using geometric mean Laplacian (Mercado et al., 2016)

- Definition: Geometric mean of two matrices A and B
$$A \# B = A^{1/2}(A^{-1/2} B A^{-1/2})^{1/2} A^{1/2}$$

Geometric-mean Laplacian

Spectral clustering using geometric mean Laplacian (Mercado et al., 2016)

- Definition: Geometric mean of two matrices A and B
$$A\#B = A^{1/2}(A^{-1/2}BA^{-1/2})^{1/2}A^{1/2}$$
- Geometric-mean Laplacian (normalized)

$$L_{GM} = L^{+}_{sym}\#Q^{-}_{sym}$$

Geometric-mean Laplacian

Spectral clustering using geometric mean Laplacian (Mercado et al., 2016)

- Definition: Geometric mean of two matrices A and B
$$A \# B = A^{1/2}(A^{-1/2}BA^{-1/2})^{1/2}A^{1/2}$$
- Geometric-mean Laplacian (normalized)
$$L_{GM} = L^{+}_{sym} \# Q^{-}_{sym}$$
- Arithmetic-mean Laplacian (normalized): $L_{AM} = L^{+}_{sym} + Q^{-}_{sym}$

Geometric-mean Laplacian

Spectral clustering using geometric mean Laplacian (Mercado et al., 2016)

- Definition: Geometric mean of two matrices A and B
$$A \# B = A^{1/2}(A^{-1/2} B A^{-1/2})^{1/2} A^{1/2}$$
- Geometric-mean Laplacian (normalized)
$$L_{GM} = L^{+}_{sym} \# Q^{-}_{sym}$$
- Arithmetic-mean Laplacian (normalized): $L_{AM} = L^{+}_{sym} + Q^{-}_{sym}$
- Spectral properties: If x is an eigenvector *shared* by L^{+}_{sym} and Q^{-}_{sym} with eigenvalue λ and μ respectively, then:

$$L_{GM}\, x = \sqrt{\lambda\mu}\; x \qquad \text{Geometric mean of } \lambda \text{ and } \mu$$
$$L_{AM}\, x = (\lambda + \mu)\; x \qquad 2 \times \text{ Arithmetic mean of } \lambda \text{ and } \mu$$

Scalar interpretation

Given two numbers a and b:

- $a + b$ is small if a and b are small
- $\sqrt{ab}$ is small if a or b is small

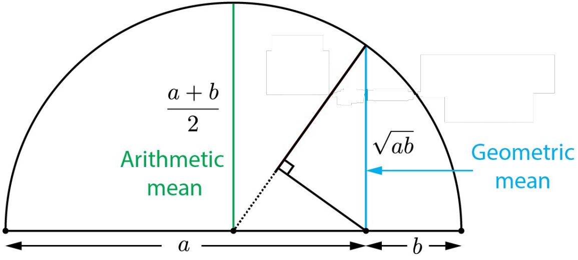


Image source: twitter.com/divbyzero

Algorithmic interpretation

- Recall: spectral clustering takes eigenvectors whose corresponding eigenvalues are small $\rightarrow$ ordering of eigenvectors matter !
- Desire: the most informative eigenvectors has the smallest eigenvalues.

Algorithmic interpretation

- Recall: spectral clustering takes eigenvectors whose corresponding eigenvalues are small $\rightarrow$ ordering of eigenvectors matter !
- Desire: the most informative eigenvectors has the smallest eigenvalues.
- Assume x is an informative eigenvector *shared* by L_{AM} and L_{GM}
- λ and μ are the corresponding eigenvalues of L^{+}_{sym} and Q^{-}_{sym} respectively

Algorithmic interpretation

- Recall: spectral clustering takes eigenvectors whose corresponding eigenvalues are small $\rightarrow$ ordering of eigenvectors matter !
- Desire: the most informative eigenvectors has the smallest eigenvalues.
- Assume x is an informative eigenvector *shared* by L_{AM} and L_{GM}
- λ and μ are the corresponding eigenvalues of L^+_{sym} and Q^-_{sym} respectively

To make x among the top, L_{AM} and L_{GM} asks different things

- $L_{AM}\, x = (\lambda + \mu)\, x$: both λ and μ are small $\leftarrow$ a stringent requirement
- $L_{GM}\, x = \sqrt{\lambda\mu}\, x$: either λ or μ is small $\leftarrow$ a loose requirement (better ✓)

Geometric-mean Laplacian: motivation

Spectral clustering using geometric mean Laplacian (Mercado et al., 2016)

Case 1: small eigenvalue of L^+_{sym} $\rightarrow$ assortative structures in G^+

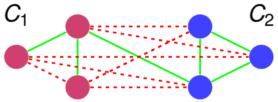
Case 2: small eigenvalue of Q^-_{sym} $\rightarrow$ disassortative structures in G^-

Geometric-mean Laplacian: motivation

Spectral clustering using geometric mean Laplacian (Mercado et al., 2016)

Case 1: small eigenvalue of L^+_{sym} $\rightarrow$ assortative structures in G^+

Case 2: small eigenvalue of Q^-_{sym} $\rightarrow$ disassortative structures in G^-



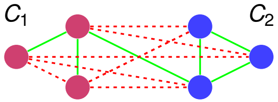
Case 1+2: L_{AM} ✓ L_{GM} ✓ (NOT always holds in practice)

Geometric-mean Laplacian: motivation

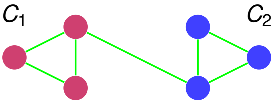
Spectral clustering using geometric mean Laplacian (Mercado et al., 2016)

Case 1: small eigenvalue of $L^+_{sym} \rightarrow$ assortative structures in G^+

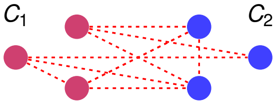
Case 2: small eigenvalue of $Q^-_{sym} \rightarrow$ disassortative structures in G^-



Case 1+2: L_{AM} ✓ L_{GM} ✓ (NOT always holds in practice)



Case 1: L_{AM} ✗ L_{GM} ✓



Case 2: L_{AM} ✗ L_{GM} ✓

Power-mean Laplacian

Spectral clustering of signed graphs via matrix power means (Mercado et al., 2019)

Power mean Laplacian

$$L_p = \left(\frac{{L^+_{sim}}^p + {Q^-_{sim}}^p}{2} \right)^{1/p}$$

where $p \in \mathbb{R}$

p	$m_p(a,b)$	name
$p \to \infty$	$\max\{a,b\}$	maximum
$p = 1$	$(a+b)/2$	arithmetic mean
$p = 0$	$\sqrt{ab}$	geometric mean
$p = -1$	$2(\frac{1}{a}+\frac{1}{b})^{-1}$	harmonic mean
$p \to -\infty$	$\min\{a,b\}$	minimum

Power-mean Laplacian

Spectral clustering of signed graphs via matrix power means (Mercado et al., 2019)

Power mean Laplacian

$$L_p = \left(\frac{{L^+_{sim}}^p + {Q^-_{sim}}^p}{2} \right)^{1/p}$$

where $p \in \mathbb{R}$

p	$m_p(a,b)$	name
$p \to \infty$	$\max\{a,b\}$	maximum
$p = 1$	$(a+b)/2$	arithmetic mean
$p = 0$	$\sqrt{ab}$	geometric mean
$p = -1$	$2(\frac{1}{a}+\frac{1}{b})^{-1}$	harmonic mean
$p \to -\infty$	$\min\{a,b\}$	minimum

L_p is a generalization of L_{AM} and L_{GM}

$p = 1 \quad \rightarrow \quad L_1 = L_{AM}$

$p = 0 \quad \rightarrow \quad L_0 = L_{GM}$

$p = -1 \quad \rightarrow$ harmonic-mean Laplacian

Power-mean Laplacian: role of the parameter p

In one sentence:

- larger p $\rightarrow$ more stringent clustering recovery condition
- smaller p $\rightarrow$ less stringent clustering recovery condition

Clustering recovery condition:

the most informative eigenvectors have the smallest eigenvalues

Power-mean Laplacian: role of the parameter p

Spectral clustering of signed graphs via matrix power means (Mercado et al., 2019)

In one sentence:

- larger p $\rightarrow$ more stringent clustering recovery condition
- smaller p $\rightarrow$ less stringent clustering recovery condition

Clustering recovery condition:

the most informative eigenvectors have the smallest eigenvalues

Extreme cases:

- $p = \infty$ requires G^+ AND G^- are informative (less diserable)
- $p = -\infty$ requires G^+ OR G^- is informative (more diserable)

Power-mean Laplacian: analysis under stochastic block model

Spectral clustering of signed graphs via matrix power means (Mercado et al., 2019)

Definition: Signed Stochastic Block Model (SSBM)

- k partitions of equal size
- p_{in}^{+} (p_{in}^{-}): probability of positive (negative) edge inside a cluster
- p_{out}^{+} (p_{out}^{-}): probability of positive (negative) edge between clusters

Power-mean Laplacian: analysis under stochastic block model

Spectral clustering of signed graphs via matrix power means (Mercado et al., 2019)

Definition: Signed Stochastic Block Model (SSBM)

- k partitions of equal size
- p_{in}^{+} (p_{in}^{-}): probability of positive (negative) edge inside a cluster
- p_{out}^{+} (p_{out}^{-}): probability of positive (negative) edge between clusters

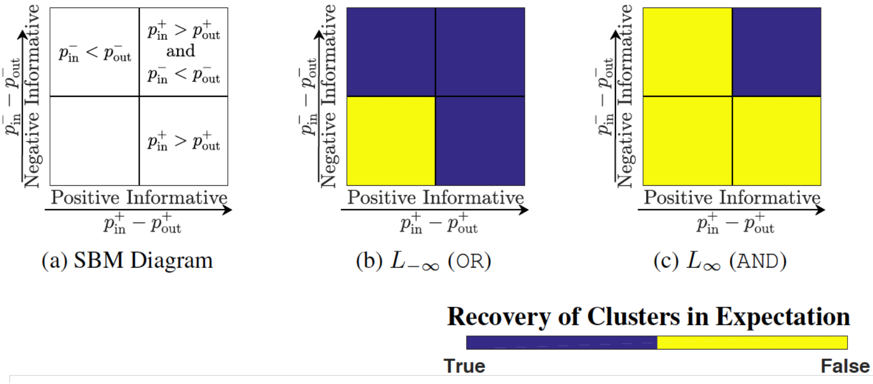
- G^{+} is informative if $p_{\text{in}}^{+} - p_{\text{out}}^{+} > 0$ and is large

  $\rightarrow$ assortative structures
- G^{-} is informative if $p_{\text{in}}^{-} - p_{\text{out}}^{-} < 0$and is small

  $\rightarrow$ disassortative structures

Power-mean Laplacian: recovery probability under SSBM $p = \infty$ or $-\infty$

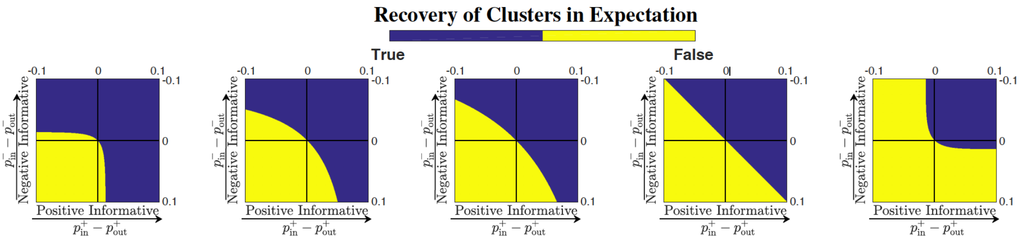
Spectral clustering of signed graphs via matrix power means (Mercado et al., 2019)



The larger the blue areas, the more likely to recover the ground-truth

Power-mean Laplacian: recovery probability under SSBM

Spectral clustering of signed graphs via matrix power means (Mercado et al., 2019)

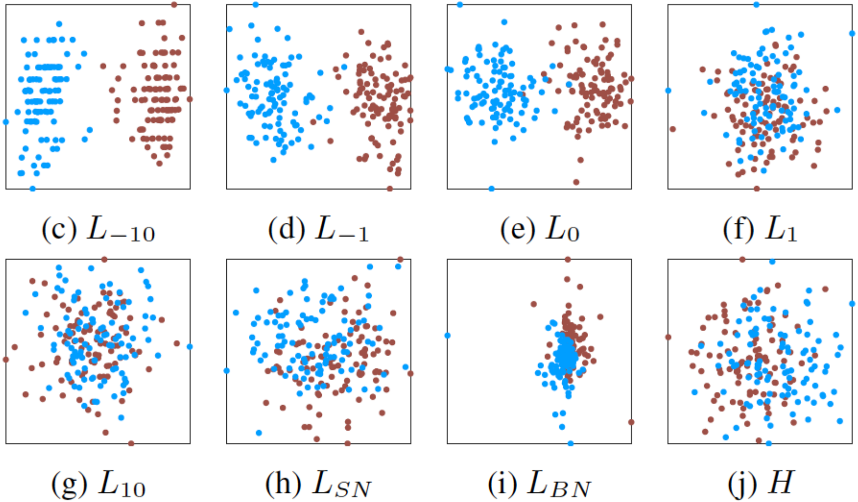


(a) $\mathcal{L}_{-10}$ (b) $\mathcal{L}_{-1}$ (c) $\mathcal{L}_0$ (d) $\mathcal{L}_1$ (e) $\mathcal{L}_{10}$

HM GM AM

Power-mean Laplacian: node embedding example ($k = 2$)

Spectral clustering of signed graphs via matrix power means (Mercado et al., 2019)



(c) L_{-10} (d) L_{-1} (e) L_0 (f) L_1

(g) L_{10} (h) L_{SN} (i) L_{BN} (j) H

Power-mean Laplacian: computing eigenvectors

Spectral clustering of signed graphs via matrix power means (Mercado et al., 2019)

- Materializing the power-mean Laplacian can result in a dense matrix

  not scalable to large graphs
- Without conducting the above step, the leading eigenvectors can be computed efficiently
- The algorithm combines the power method with a Krylov subspace approximation technique (Mercado et al., 2018)

Graph partitioning: summary

- Main focus: spectral methods and the choice of Laplacian matrix
  - Arithmetic-mean Laplacian L_{AM}
  - Geometric-mean Laplacian L_{GM}
  - Power-mean Laplacian L_p (a generalization of the above)
  - $(L_{AM} = L_1) \; < \; (L_{GM} = L_0) \; < \; (L_{p<0})$
  - $A \; < \; B$ means A is stricter ("worse") than B in terms of clustering recovery condition
- Uncovered topics:
  - Probabilistic methods (Jiang, 2015; Yang et al., 2017)
  - Modularity-based methods (Gómez et al., 2009; Traag and Bruggeman, 2009; Li et al., 2014)
  - Ad-hoc methods (Chiang et al., 2012; Wu et al., 2016a)

Outline

Correlation clustering

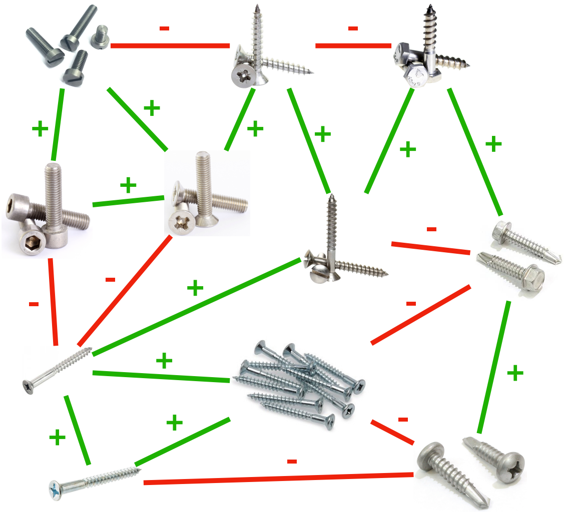
Data clustering — background

- Data clustering: a fundamental problem in machine learning
- Intuitively: we want to partition a dataset into clusters so that similar objects are assigned to the same cluster
- Extensively-studied problem, many different settings, objectives, applications
- Euclidean setting: data are represented as Euclidean points
  - minimize an objective function such as k-means ($\sum_i \min_j ||x_i - c_j||_2^2$), k-median ($\sum_i \min_j ||x_i - c_j||_2$) or k-center ($\max_i \min_j ||x_i - c_j||_2$)
- Graph setting: data are represented as a graph
  - edges represent affinity, e.g., friends in a social network
  - often a similarity value is available, e.g., connection strength
  - optimize an objective function such as normalized edge cut across clusters (minimize) or edge density within clusters (maximize)

Correlation clustering — motivation

- In the graph setting described above, edges are positive
  - presence of an edge suggests that nodes should be clustered together
  - absence of an edge suggests that nodes should be assigned to different clusters
- In some cases, we may have a local prediction whether two objects should be assigned to the same cluster or not
  - positive edge : the two objects should be clustered together
  - negative edge : the two objects should be assigned to different clusters
  - no edge : no information
- We obtain a signed network !

Correlation clustering — motivation

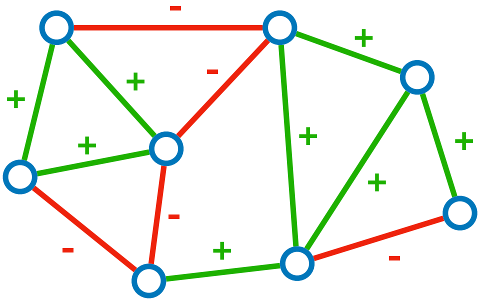


- example: a dataset of images, e.g., screws of different types
- a machine-learning program, which, given two images, outputs whether the images depict the same type of screws
- we obtain a signed network
- we want to cluster the images so that same-type screws are assigned in the same cluster

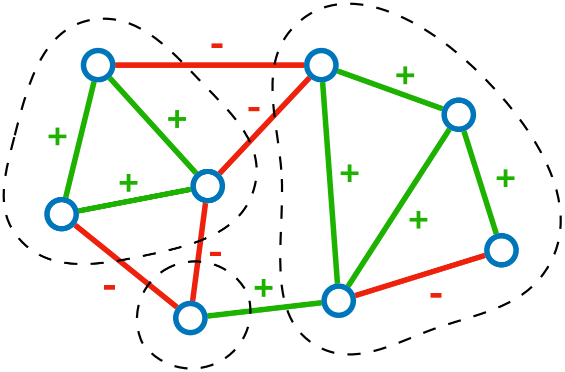
Correlation clustering — motivation

- Due to noise in the data and classification errors in the network construction, we cannot expect to achieve perfect agreement
- We need an objective function to capture the consistency of the resulting clustering with the input signed network

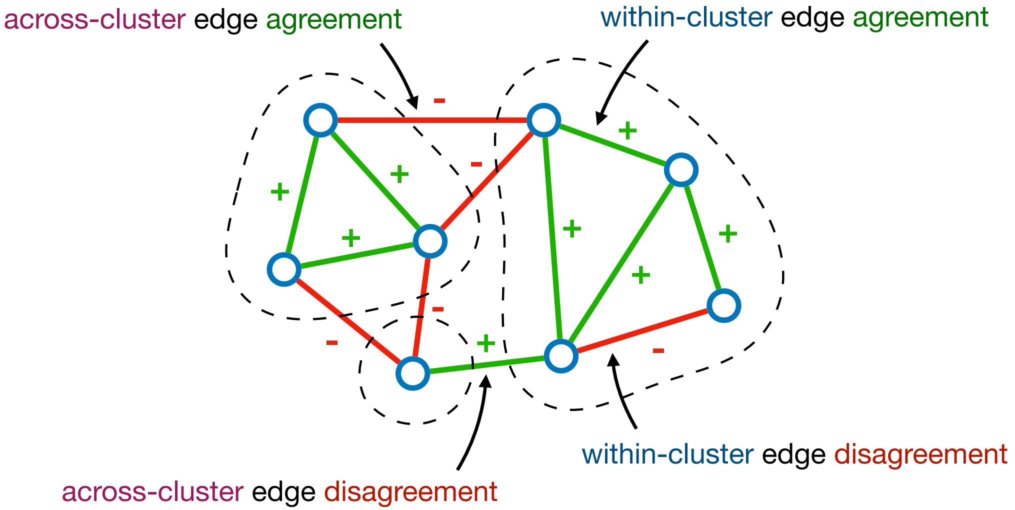
Correlation clustering — edge agreements and disagreements



Correlation clustering — edge agreements and disagreements



Correlation clustering — edge agreements and disagreements



Correlation clustering — problem formulation

Given a signed network $G = (V, E^+, E^-)$, find a partitioning $\mathcal{C} = \{C_1, \ldots, C_k\}$ of the graph vertices (i.e., $\bigcup_{i=1}^k C_i = V$ and $C_i \cap C_j = \varnothing$, for all $i \neq j$),

so as to

Variant 1 : [Maximize agreements]

$$\max \quad a(\mathcal{C}) = \sum_{i,j} \mathbb{I}\{(i,j) \in E^+\}\, \mathbb{I}\{c(i) = c(j)\} + \sum_{i,j} \mathbb{I}\{(i,j) \in E^-\}\, \mathbb{I}\{c(i) \neq c(j)\}$$

Variant 2 : [Minimize disagreements]

$$\min \quad d(\mathcal{C}) = \sum_{i,j} \mathbb{I}\{(i,j) \in E^+\}\, \mathbb{I}\{c(i) \neq c(j)\} + \sum_{i,j} \mathbb{I}\{(i,j) \in E^-\}\, \mathbb{I}\{c(i) = c(j)\}$$

Correlation clustering — problem formulation

Given a signed network $G = (V, E^+, E^-)$, find a partitioning $\mathcal{C} = \{C_1, \ldots, C_k\}$ of the graph vertices (i.e., $\bigcup_{i=1}^k C_i = V$ and $C_i \cap C_j = \emptyset$, for all $i \neq j$), so as to

Variant 1 : [Maximize agreements]

$$\max \quad a(\mathcal{C}) = \sum_{i,j} \mathbb{I}\{(i,j) \in E^+\}\, \mathbb{I}\{c(i) = c(j)\} + \sum_{i,j} \mathbb{I}\{(i,j) \in E^-\}\, \mathbb{I}\{c(i) \neq c(j)\}$$

Variant 2 : [Minimize disagreements]

$$\min \quad d(\mathcal{C}) = \sum_{i,j} \mathbb{I}\{(i,j) \in E^+\}\, \mathbb{I}\{c(i) \neq c(j)\} + \sum_{i,j} \mathbb{I}\{(i,j) \in E^-\}\, \mathbb{I}\{c(i) = c(j)\}$$

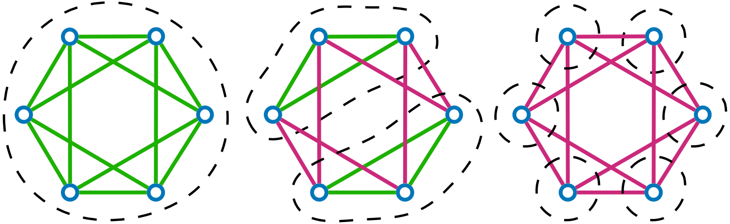
Majority of research focuses on the minimization variant.

Correlation clustering — number of clusters

An important observation

- the problem formulation does not (need to) specify the number of clusters
- optimal k depends on input network, and does not have trivial minimizers

  e.g.,



$k = 1$ $k = 2$ $k = n$

- the optimal solution in each of the above cases is the most intuitive one

Correlation clustering — hardness

Both formulations (max-agree and min-disagree) are **NP**-hard

The min-disagree problem is

- **NP**-hard for complete unweighted graphs — Bansal et al. (2004)
  reduction from "partition into triangles"
- **APX**-hard for general (un)weighted graphs — Demaine et al. (2006)
  reduction from multiway cut

Correlation clustering — existing approximation algorithms

Overview of results for the min-disagree problem

Paper	graph type	approximation ratio	deterministic /randomized	running time
Bansal et al. (2004)	complete	large constant	deterministic	$\mathcal{O}(n^2)$
Demaine et al. (2006)	general	$\mathcal{O}(\log n)$	deterministic	LP
Ailon et al. (2005)	complete	2.5	randomized	LP
Ailon et al. (2005)	complete	3	randomized	$\mathcal{O}(m)$
Chawla et al. (2015)	complete	$2.06 - \epsilon$	deterministic	LP
Giotis and Guruswami (2005)[1]	complete	PTAS	randomized	combinatorial
Coleman et al. (2008)[2]	complete[3]	2	deterministic	combinatorial

[1] For fixed k; Recall that the problem is **APX**-hard when k is not fixed

[2] For $k = 2$ (2-correlation-clustering)

[3] Algorithm applicable to general graphs, but analysis for complete graphs

Correlation clustering — the KwikCluster (or Pivot) algorithm

$\textsc{KwikCluster}(G = (V, E^+, E^-))$

```
If V = ∅ then return ∅
Pick random pivot i ∈ V.
Set C = {i}, V' = ∅.

For all j ∈ V, j ≠ i:
    If (i, j) ∈ E⁺ then
        Add j to C
    Else (If (i, j) ∈ E⁻)
        Add j to V'

Let G' be the subgraph induced by V'.

Return C ∪ KwikCluster(G') .
```

Correlation clustering — the KwikCluster (or Pivot) algorithm

$\textsc{KwikCluster}(G = (V, E^+, E^-))$

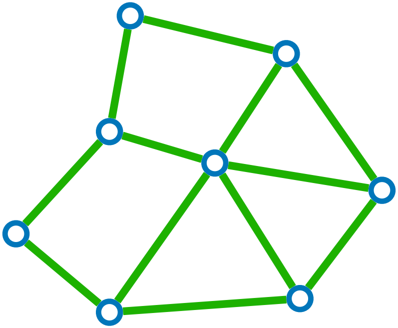
```
If V = ∅ then return ∅
Pick random pivot i ∈ V.
Set C = {i}, V′ = ∅.

For all j ∈ V, j ≠ i:
    If (i, j) ∈ E+ then
        Add j to C
    Else (If (i, j) ∈ E−)
        Add j to V′

Let G′ be the subgraph induced by V′.

Return C ∪ KwikCluster(G′) .
```

- The Pivot algorithm

  Ailon et al. (2005)

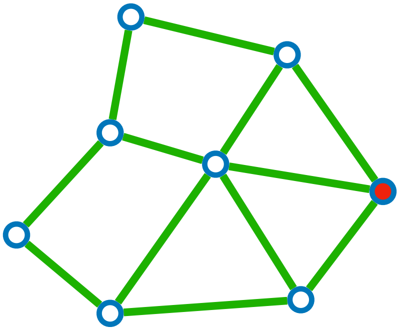
+ An elegant randomized algorithm
+ Approximation ratio 3
+ Running time $\mathcal{O}(m)$
– It assumes a complete graph
– It assumes an unweighted graph

The PIVOT algorithm — example



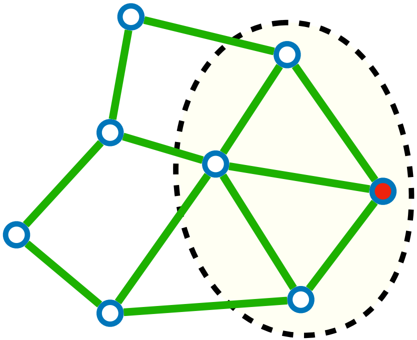
A complete graph: positive edges shown, negative edges not shown

The PIVOT algorithm — example



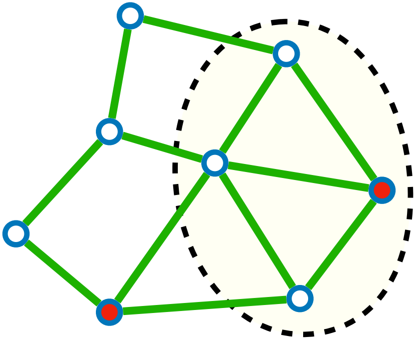
a pivot is selected uniformly at random

The Pivot algorithm — example



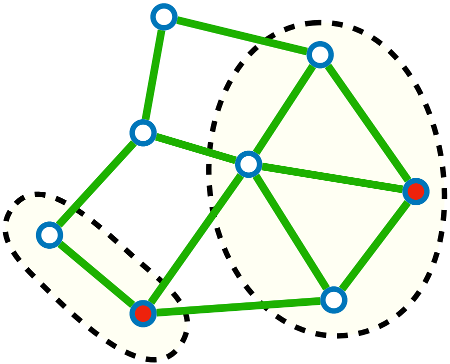
a cluster is formed with the pivot and all its positive neighbors

The PIVOT algorithm — example



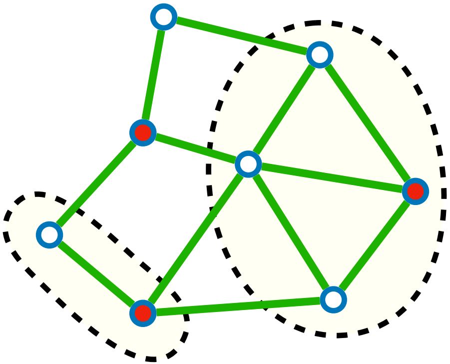
a new pivot is selected from the remaining of the graph vertices

The PIVOT algorithm — example



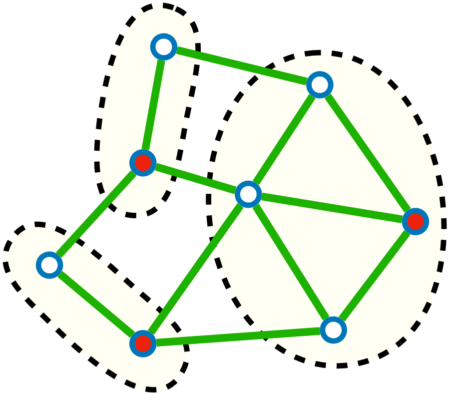
a second cluster is formed with the pivot and all its positive neighbors

The PIVOT algorithm — example



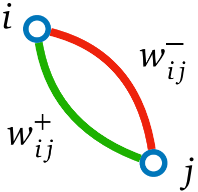
and the process continues ...

The Pivot algorithm — example



... until the whole graph is consumed.

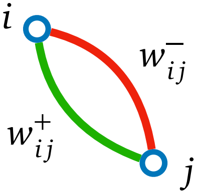
Weighted signed networks



We want to extend the methods to weighted signed networks $G = (V, w^+, w^-)$

- w^+_{ij} : weight of positive edge (i, j)
- w^-_{ij} : weight of negative edge (i, j)
- unweighted case : $w^+_{ij}, w^-_{ij} \in \{0, 1\}$
- weighted case : $w^+_{ij}, w^-_{ij} \in \mathbb{R}_{\geq 0}$

Weighted signed networks



We want to extend the methods to weighted signed networks $G = (V, w^+, w^-)$

- w_{ij}^+ : weight of positive edge (i, j)
- w_{ij}^- : weight of negative edge (i, j)
- unweighted case : $w_{ij}^+, w_{ij}^- \in \{0, 1\}$
- weighted case : $w_{ij}^+, w_{ij}^- \in \mathbb{R}_{\geq 0}$

Interesting cases :

- probability constraints : $w_{ij}^+ + w_{ij}^- = 1$, for all $i, j \in V$
- triangle inequality : $w_{ik}^- \leq w_{ij}^- + w_{jk}^-$, for all $i, j, k \in V$

The PIVOT algorithm on weighted signed networks

1. Consider a weighted signed networks $G = (V, w^+, w^-)$
2. Assume probability constraints $w_{ij}^+ + w_{ij}^- = 1$, for all $i, j \in V$
3. Form unweigted $G_u = (V, E^+, E^-)$ by taking “majority” on each edge
4. Apply PIVOT on G_u
5. Return solution of PIVOT on G_u, as the solution for G

Theoretical properties of the above algorithm

- 5 approximation, with probability constraints
- 2 approximation, with probability constraints and triangle inequality

Using PIVOT for LP rounding

LP relaxation Ailon et al. (2005)

$$
\begin{aligned}
\text{maximize} \quad & \sum_{ij} \left(x_{ij}^{+} w_{ij}^{-} + x_{ij}^{-} w_{ij}^{+} \right) \\
\text{such that} \quad & x_{ik}^{-} \leq x_{ij}^{-} + x_{jk}^{-}, \text{ for all } i, j, k \in V \\
& x_{ij}^{+} + x_{ij}^{-} = 1, \text{ for all } i, j \in V \\
& x_{ij}^{+}, x_{ij}^{-} \geq 0, \text{ for all } i, j \in V
\end{aligned}
$$

- Notice that if $x_{ij}^{-} \in \{0, 1\}$, then x_{ij}^{-} define an equivalence class (clustering)

Using PIVOT for LP rounding

LP-KWIKCLUSTER(V, x^+, x^-)

A recursive algorithm for rounding the LP for weighted CORRELATION-CLUSTERING. *Given an LP solution* $x^+ = \{x^+_{ij}\}_{i<j}$, $x^- = \{x^-_{ij}\}_{i<j}$, *returns a clustering of the vertices*

```
If V = ∅ then return ∅
Pick random pivot i ∈ V.
Set C = {i}, V' = ∅.

For all j ∈ V, j ≠ i :
    With probability x⁺_ij
        Add j to C.
    Else (With probability x⁻_ij = 1 − x⁺_ij)
        Add j to V'.

Return clustering
  {C} ∪ LP-KwikCluster(V', x⁺, x⁻).
```

Ailon et al. (2005)

1. Solve the LP relaxation
2. Use the PIVOT for randomized rounding of the LP solution

- ▶ 2.5-approximation, with probability constraints
- ▶ 2-approximation, with probability & triangle inequality constraints
- – Expensive; requires solving an LP

Correlation clustering — summary

- Signed graphs have been studied in theoretical computer science in the context of correlation clustering
- A wealth of theoretical results for different problem settings
- Several applications, e.g., clustering aggregation
- Many other problem variants not discussed here

  overlapping, on-line, bipartite, chromatic, local, . . .
- See also KDD tutorial by Bonchi et al. (2014)

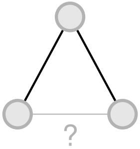
Outline

Link prediction and link classification

Link prediction and link classification

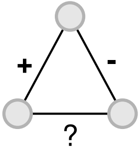
In unsigned networks:

- Link prediction:
  - Is an unobserved edge likely to exist?
  - Is a non-existing edge likely to materialize?



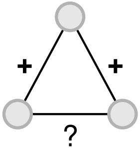
In signed networks:

- Link prediction:
  - Is an unobserved edge of a particular sign likely to exist?
- Link classification:
  - What sign an existing edge is likely to have?
    Task also known as sign prediction or label prediction

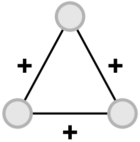


Link classification

Methods based on balance

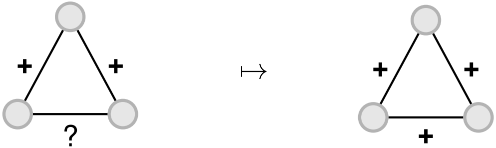


$\mapsto$

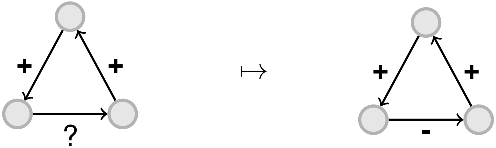


Link classification

Methods based on balance

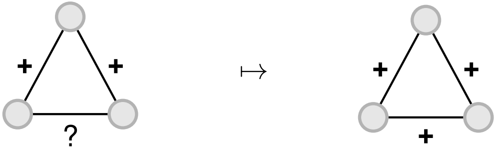


Methods based on status

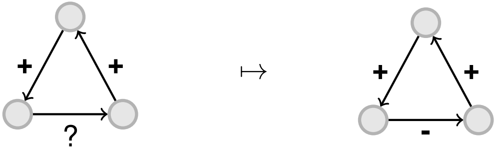


Link classification

Methods based on balance



Methods based on status



Methods based on machine learning

$$G \mapsto x \in \mathbb{R}^d,\ f : \mathbb{R}^d \to \{\textbf{-}, \textbf{+}\}$$

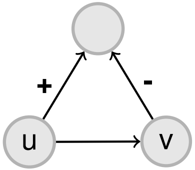
Link classification

Predicting positive and negative links in online social networks (Leskovec et al., 2010)

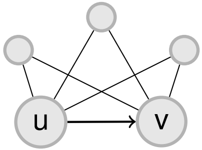
Task: predict sign of (u, v)

Feature extraction:

- In & out-degree: $d_{in}^{+}(u), d_{in}^{-}(v), \ldots$
- Count vector of directed triads: forward / backward, **+** / **-**
- Embeddedness (number of common neighbours of u, v)



FBpn triad



Embeddedness = 3

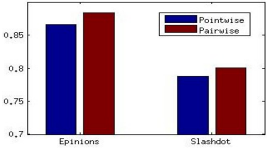
Link classification

Link label prediction in signed social networks

(Agrawal et al., 2013)

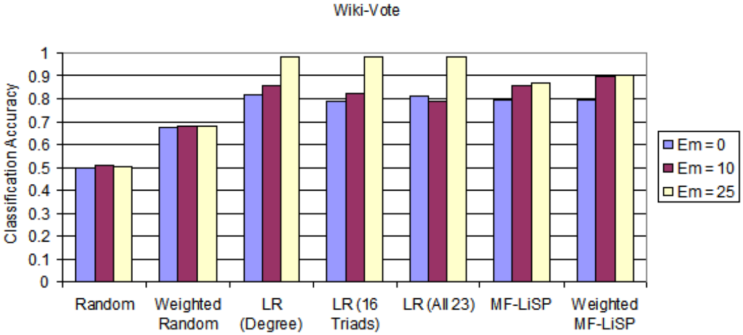
Given incomplete $n \times n$ adjacency matrix A, learn $X = UV$ to minimize

$$\frac{1}{n}\sum_{i=1}^{n}\frac{1}{|Q_i|}\sum_{(j,k)\in Q_i}((A_{ij}-A_{ik})-(X_{ij}-X_{ik}))_{+}+\frac{\lambda}{2}\|U\|_F^2\|V\|_F^2$$



Pairwise vs. pointwise

Better for edges of low embeddedness (Leskovec et al., 2010)



Link classification

Exploiting longer cycles for link prediction in signed networks (Chiang et al., 2011)

Measure of imbalance:

$$\mu_k(G) = \sum_{i=3}^{k} \beta_i \sum_{\sigma \in Cycles_i} \mathbb{I}\{\sigma \text{ is balanced}\}$$

Prediction:

$$sign(i,j) = sign(\mu_k(G \cup (i,j,-)) - \mu_k(G \cup (i,j,+)))$$

Key insight:

$$sign(\mu_k(G \cup (i,j,-)) - \mu_k(G \cup (i,j,+))) = sign(\sum_{t=3}^{k} \beta_t A_{i,j}^{t-1})$$

So we just need to compute powers of the adjacency matrix !

Two predictors: balance-based and feature based

Learning bounds for link classification

- Link classification methods discussed in previous slides focus on empirical performance
- Link classification has also been studied in the context of computational learning theory
- The goal is to derive theoretical bounds on the number of mistakes a learning algorithm can make
- Choice of computational learning model is important
- Several different models have been studied

Link classification in computational learning theory

- The link-classification task, at high level:
  1. Observe a graph $G = (V, E)$ and the signs of a subset $E_0 \subseteq E$
  2. Build model $f : E \to \{-1, 1\}$
  3. Use f to predict the signs of the rest of the edges $E \setminus E_0$
- Goal: bound the number of mistakes on $E \setminus E_0$
- Questions: What are the assumptions to ensure a small number of mistakes?

  Why observing a subset of edges allows to predict the rest of the edges?
- Observation: Predictions are easy on a perfectly balanced graph

  The closer is a graph to be balanced, the easier is to make predictions

Mistake bounds for link classification

- Consider: What should a mistake bound depend on?

  $|V|$, $|E|$, $|E_0|$, but also ... a measure of balance of G

- such a measure is $\Delta(G)$: cost of optimal correlation clustering of G

  e.g., $\Delta(G) = 0$ if G is perfectly balanced

  Why? What is the intuition?

Mistake bounds for link classification

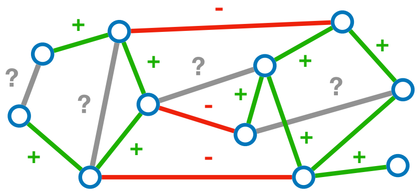
- Consider: What should a mistake bound depend on?

  $|V|$, $|E|$, $|E_0|$, but also ... a measure of balance of G

- such a measure is $\Delta(G)$: cost of optimal correlation clustering of G

  e.g., $\Delta(G) = 0$ if G is perfectly balanced

Why? What is the intuition?



- small $\Delta(G)$ indicates clear community structure
- a small set of edges E_0 suffices to reveal the community structure
- rest of edges $E \setminus E_0$ can be predicted, and small $\Delta(G)$ ensures few mistakes

Batch transductive learning

A correlation clustering approach to link classification in signed networks (Cesa-Bianchi et al., 2012)

Learning algorithm

1. Observe signs on a subset of edges $E_0 \subseteq E$
2. Solve correlation-clustering problem on $G = (V, E_0)$ to obtain solution f
3. Use solution f to predict the rest of the edges

Batch transductive learning

A correlation clustering approach to link classification in signed networks (Cesa-Bianchi et al., 2012)

Learning algorithm

1. Observe signs on a subset of edges $E_0 \subseteq E$
2. Solve correlation-clustering problem on $G = (V, E_0)$ to obtain solution f
3. Use solution f to predict the rest of the edges

(Assumptions) If

- E_0 is a random subset of E
- correlation clustering on $G = (V, E_0)$ can be approximated within factor α
- (only for simplicity of exposition, $|E_0| = m/2$)

Batch transductive learning

A correlation clustering approach to link classification in signed networks (Cesa-Bianchi et al., 2012)

Learning algorithm

1. Observe signs on a subset of edges $E_0 \subseteq E$
2. Solve correlation-clustering problem on $G = (V, E_0)$ to obtain solution f
3. Use solution f to predict the rest of the edges

(Assumptions) If

- E_0 is a random subset of E
- correlation clustering on $G = (V, E_0)$ can be approximated within factor α
- (only for simplicity of exposition, $|E_0| = m/2$)

then, the number of mistakes on $E \setminus E_0$ can be bounded by $\Delta\alpha + \sqrt{|E||V| \log |V|}$

- An application of a result by El-Yaniv and Pechyony (2009)
- Recall $\alpha = \mathcal{O}(\log |V|)$ (Demaine et al., 2006)

Active learning

A correlation clustering approach to link classification in signed networks (Cesa-Bianchi et al., 2012)

- As before, but learner selects subset $E_0 \subseteq E$
- Adversary returns labeling on E_0
- Learner makes predictions on $E \setminus E_0$

- As before, but learner selects subset $E_0 \subseteq E$
- Adversary returns labeling on E_0
- Learner makes predictions on $E \setminus E_0$

One particular case of interest, p-random model :

- The observed labeling is obtained by a graph that has 2 clusters and $\Delta = 0$, and where each edge is flipped with probability p

Active-learning algorithm

1. Select a spanning tree $T = (V_T, E_T)$, and query the labels of E_T
2. Predict label of $(u, v) \in E \setminus E_0$ as the product of signs on $path_T(u, v)$

Active-learning algorithm

1. Select a spanning tree $T = (V_T, E_T)$, and query the labels of E_T
2. Predict label of $(u, v) \in E \setminus E_0$ as the product of signs on $path_T(u, v)$

Number of mistakes :

$$M_T \leq |E_{flip}| + \sum_{e' \in E \setminus E_T} \sum_{e \in E_T} \mathbb{I}\left\{e \in path_T(e')\right\} \mathbb{I}\left\{e \in E_{flip}\right\}$$

Active learning

A correlation clustering approach to link classification in signed networks (Cesa-Bianchi et al., 2012)

Active-learning algorithm

1. Select a spanning tree $T = (V_T, E_T)$, and query the labels of E_T
2. Predict label of $(u, v) \in E \setminus E_0$ as the product of signs on $path_T(u, v)$

Number of mistakes :

$$M_T \leq |E_{flip}| + \sum_{e' \in E \setminus E_T} \sum_{e \in E_T} \mathbb{I}\{e \in path_T(e')\} \, \mathbb{I}\{e \in E_{flip}\}$$

- M_T related to average stretch of the spanning tree T, which we can approximate within $\mathcal{O}(\log^2 |V| \log\log |V|)$ (Elkin et al., 2008)

Active-learning algorithm

1. Select a spanning tree $T = (V_T, E_T)$, and query the labels of E_T
2. Predict label of $(u, v) \in E \setminus E_0$ as the product of signs on $path_T(u, v)$

Number of mistakes :

$$M_T \leq |E_{flip}| + \sum_{e' \in E \setminus E_T} \sum_{e \in E_T} \mathbb{I}\left\{e \in path_T(e')\right\} \mathbb{I}\left\{e \in E_{flip}\right\}$$

- M_T related to average stretch of the spanning tree T, which we can approximate within $\mathcal{O}(\log^2 |V| \log\log |V|)$ (Elkin et al., 2008)

Thus,

$$\mathbb{E}\left[M_T\right] \leq p|E|\,\mathcal{O}(\log^2 |V| \log\log |V|)$$

Link prediction and link classification — summary

- Predicting new links and their signs
- Different models, assumptions, approaches
- Empirical as well as theoretical work
- We reviewed only a small sample of work

Outline

Network dynamics

Network Dynamics in Signed Graphs

Most works in this area model the evolution of social networks to achieve balance

- A large network is almost surely imbalanced
- Gradual evolution to a more stable state
- Social dynamics $\rightarrow$ tendency to reduce imbalanced triads

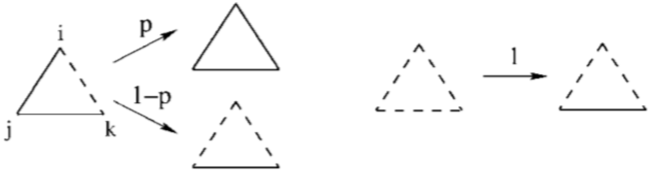
Possible outcomes of dynamics:

- "Paradise" state: all positive signature
- Bipolar state: two antagonistic factions where all interior edges are positive and all exterior negative
- "Jammed" state: system is trapped in local optimum

Analysis often relies on statistical physics. Therefore many works consider complete graphs

T. Antal, P. L. Krapivsky, S. Redner: "Dynamics of social balance on networks" (2005)

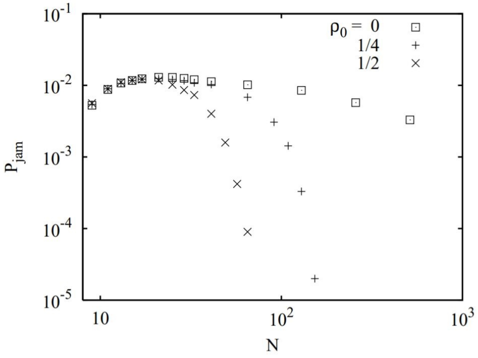
- Local dynamics: Pick random unbalanced triad and flip sign of one link



- New imbalances $\rightarrow$ interaction cascade
- Infinite network:
  1. For $p < 1/2$ quasistationary dynamics
  2. For $p \geq 1/2$ paradise state
- Finite network:
  1. For $p < 1/2$ bipolar state
  2. For $p \geq 1/2$ paradise state

Constrained triad dynamics

- Select random link
  1. flip sign if # of imbalanced triads decreases
  2. flip w.p. $1/2$ if # of imbalanced triads stays the same
  3. don't flip if # of imbalanced triads increases
- Corresponds to Ising model with a three-spin interaction between links of triad
- In $\ln N$ time reaches balanced state or gets trapped in jammed state



Probability of reaching a jammed state P_{jam} as a function of N for several values of ρ_0.

Image

source: Antal et al.

Marvel S., Strogatz S., Kleinberg J.: "Energy Landscape of Social Balance" (2008)

Investigate Jammed states introduced by Antal et al.

Definition (Energy)

The energy of a social network is defined as:

$$U = -\frac{1}{\binom{n}{3}} \sum s_{ij} s_{jk} s_{ik}$$

- Product of edge signs is positive for balanced triangles, negative for unbalanced
- The "correction" of an edge sign to achieve balance relaxes the energy of the system

Contributions:

- Proof that jammed states at most halfway up the energy spectrum
- Show that jammed states have a natural modular structure

K. Kulakowski, P. Gawronski, P. Gronek: "The Heider balance - a continuous approach"(2005)

Real numbers in $[-R, R]$ instead of ± 1 to describe social distance

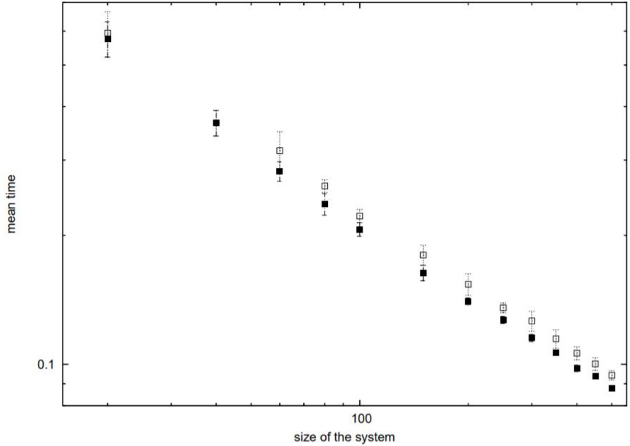
- Represent a completely connected social network using $n \times n$ matrix X of edge strengths
- Dynamics expressed in terms of differential equations

Definition (Evolution of social relationships)

$$\frac{dX}{dt} = X^2 \quad \Longleftrightarrow \quad \frac{dx_{ij}}{dt} = \sum_k x_{ik} x_{kj}$$

- x_{ij} pushed in a positive or negative direction so as to achieve balance in the triangle i, j, k
- Converges to either paradise state or to two opposing factions

Reaching Heider balance



Time to reach HB as dependent on the system size N for $R = 5.0$ (open squares) and $R = \infty$ (black squares).

Image source: Kulakowski et al.

S. Marvel, J. Kleinberg, R. Kleinberg, S. Strogatz: "Continuous-time model of structural balance"(2011)

The paper tackles the open questions presented in the paper of Kulakowski et al.

Analytical solution to the model:

- initial matrix $X(0)$ is real and symmetric so it admits eigendecomposition: $QD(0)Q^T, D(0) = Diag(\lambda_1, \dots, \lambda_n)$
- solution is obtained using matrix recatti equations as $D(t) = Diag(\ell_1(t), \dots, \ell_n(t))$ where $\ell_k(t) = \frac{\ell_k}{1-\ell_k t}$

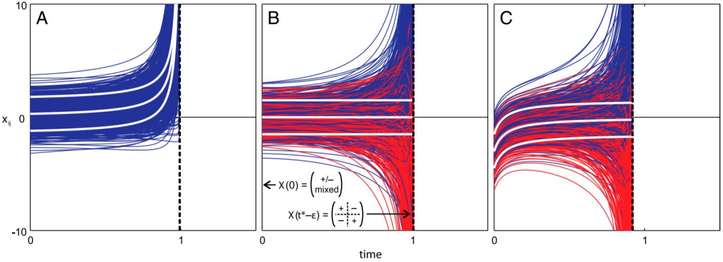
What are the conditions for reaching balance?

Definition (Conditions)

- $\lambda_1 > 0$ *(holds with $p \to 1$ as $n \to \infty$ in finite time)*
- $\lambda_1 \neq \lambda_2$ *(holds with $p = 1$)*
- *all components of first eigenvector are nonzero* *(holds with $p = 1$)*

Analysis for classes of random matrices

- Consider matrices where each entry is drawn from distribution with bounded support, symmetric about μ
- As μ varies:
  1. $\mu > 0$: the system evolves to all-positive signature
  2. $\mu \leq 0$: the system evolves to two all-positive cliques connected by negative edges



Representative large-n plots of the model for (A) $\mu > 0$ ($\mu = 3/10$ in the plot shown), (B) $\mu = 0$, and (C) $\mu < 0$ ($\mu = -3$ in the plot shown).

Image source: Marvel et al.

C. Altafini: "Dynamics of Opinion Forming in Structurally Balanced Social Networks" (2012)

Main assumption: Outcome of opinion forming process overlaps with bipartition of the network Opinion vector $x \in \mathbb{R}^n$, opinion formation process $f(\cdot)$

Definition (Influence)

$F_{ij}(x) = \frac{\partial f_i(x)}{\partial x_j}, i, j = 1, \ldots, n \quad \rightarrow sgn(F_{ij}(x)) = sgn(A_{ij})$

Definition (Monotonicity(Kamke condition))

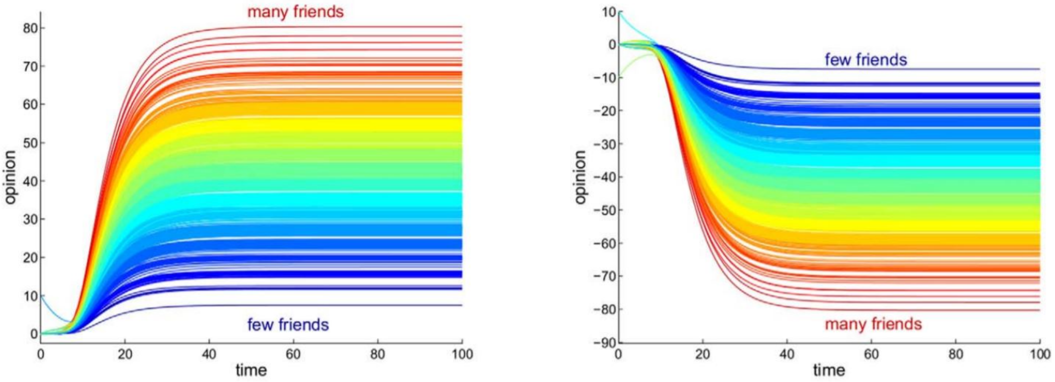
$x = f(x)$ is monotone with respect to order σ iff :

$$\sigma_i \sigma_j F_{ij}(x) \geq 0 \quad x \in \mathbb{R}^n, \quad \forall i, j = 1, \ldots, n \quad i \neq j$$

Monotone system $\iff$ corresponding graph is structurally balanced

Results(1)

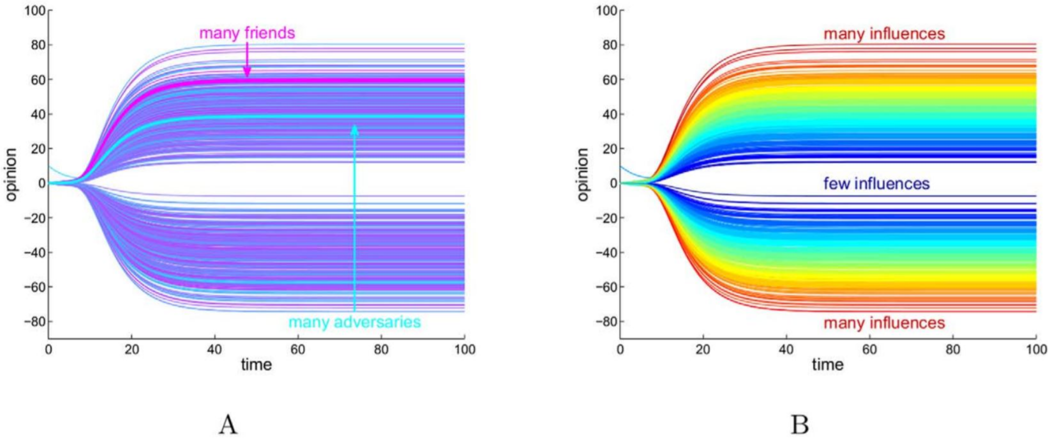
Cooperative case:



A B

Results(2)

Antagonistic case:



Image

source: Altafini et al

Outline

Graph embedding and representation learning

Signed graphs in neural networks

Mining signed social networks substantially different from unsigned networks

- Homophily and social influence applicable in different form
- Simple extension of unsigned embedding algorithms is not possible

The following papers propose some approaches.

- We first show a paper that operates within the classical embeddings framework.
- A more recent paper by Derr et al. extends graph convolutional networks for signed graphs

Attributed Signed Network Embedding, Wang et al. (2017)

Leverages both user attributes and signed links for learning network embeddings

First they perform an analysis of links and attributes

- Users more likely to have similar attributes with friends than with foes
- Users more likely to be similar with foes than users without connections

Average user attributes similarities in $\mathcal{P}$, $\mathcal{N}$ and $\mathcal{R}$

	Epinions		Slashdot	
	CA	COSINE	CA	COSINE
$\mathcal{P}$	75.93	0.0650	21.24	0.0332
$\mathcal{N}$	67.16	0.0540	16.64	0.0289
$\mathcal{R}$	27.30	0.0168	10.34	0.0162

Image source: Wang et al.

Decomposing the adjacency matrix

The first step is to decompose the adjacency matrix $\mathbf{A}$:

$$\min_{\mathbf{U},\mathbf{H}} \|\mathbf{W} \odot (\mathbf{A} - \mathbf{U}\mathbf{H}\mathbf{U}^T)\|_F^2$$

- $\mathbf{U}$ is the latent user feature matrix for $\mathbf{A}$
- $\mathbf{W}_{ij} = 1$ if $\mathbf{A}_{ij} \neq 0$, $\mathbf{W}_{ij} = 0$ otherwise
- $\mathbf{H}$ "interaction matrix", captures similarity of user latent features

Incorporating structural balance

Based on Euclidean distance

For three users (u_i, u_j, u_k):

- if $\mathbf{A}_{ij} = 1$ and $\mathbf{A}_{ik} = 0$ then it should be $\|u_i - u_j\|_2^2 \leq \|u_i - u_k\|_2^2$
- if $\mathbf{A}_{ij} = -1$ and $\mathbf{A}_{ik} = 0$ then it should be $\|u_i - u_j\|_2^2 \geq \|u_i - u_k\|_2^2$
- include all these triplets in $\mathcal{H}$

$$\min_U \sum_{(u_i,u_j,u_k)\in\mathcal{H}} \max(0, \|u_i - u_j\|_2^2 - \|u_i - u_k\|_2^2)$$

rewrite it as:

$$\min Tr(\sum_{(u_i,u_j,u_k)\in\mathcal{H}} I_{ijk}\mathbf{M}^{ijk}\mathbf{U}\mathbf{U}^T)$$

where $I_{ijk} = 1$ if $\|u_i - u_j\|_2^2 > \|u_i - u_k\|_2^2$, 0 otherwise and appropriately defined $\mathbf{M} \in \{-1, 0, 1\}^{n\times n}$

Learning user attributes

Project $\mathbf{U}$ to user attribute matrix $\mathbf{X}$ using projection $\mathbf{P}$

$$\min_{\mathbf{U},\mathbf{V},\mathbf{P}^T\mathbf{P}=\mathbf{I}} \|\mathbf{X} - \mathbf{U}\mathbf{P}\mathbf{V}^T\|_F^2$$

Adding user similarities encoded in the laplacian $\mathbf{L} = \mathbf{D} - \mathbf{S}$, where $\mathbf{S}$ is kernel of $\mathbf{X}$:

$$\min_{\mathbf{U},\mathbf{P}} Tr(\mathbf{P}^T\mathbf{U}^T\mathbf{L}\mathbf{U}\mathbf{P})$$

Proposed model

Putting everything together, along with regularization:

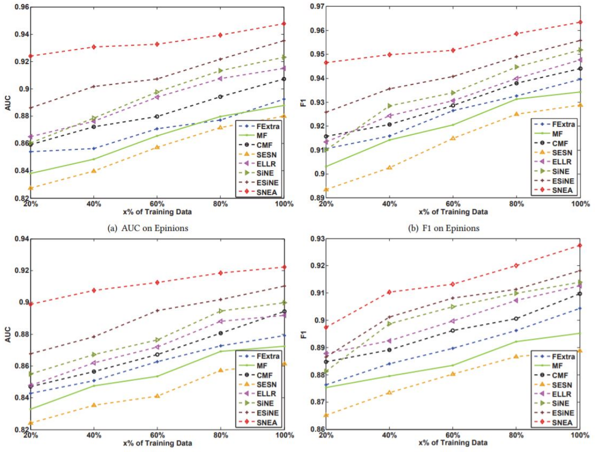
$$\min_{\mathbf{U},\mathbf{P},\mathbf{V},\mathbf{H}} \|\mathbf{W} \odot (\mathbf{A} - \mathbf{UHU}^T)\|_F^2 + \alpha Tr\Big(\sum_{(u_i,u_j,u_k)\in\mathcal{H}} I_{ijk}\mathbf{M}^{ijk}\mathbf{UU}^T\Big) + \beta\|\mathbf{X} - \mathbf{UPV}^T\|_F^2 +$$

$$+\gamma Tr(\mathbf{P}^T\mathbf{U}^T\mathbf{LUP}) + \lambda(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2\|\mathbf{H}\|_F^2)$$

The objective function is optimized using gradient descent

Performance analysis: Link prediction

- Fextra: learns degree-based and triad-based features
- MF and CMF: matrix factorization-based, MF: links, CMF: links and attributes
- SESEN: Spectral analysis for signed graphs
- ELLR: ranking-based signed network embedding algorithm
- SiNE: deep learning framework that utilizes structural balance theory, ESiNE extension that aggregates also attributes



(a) AUC on Epinions
(b) F1 on Epinions
(c) AUC on Slashdot
(d) F1 on Slashdot

Image source: Wang et al.

Graph convolutional networks

A graph convolutional network(GCN) is a neural network that operates on graphs

- Learns node-level representations
- Convolutional layers learn different feature representations of nodes
- Aggregation function on features of each layer to form next layer's features
  - First layer encodes information about the neighborhood of every node
  - Second layer encodes 2-hop neighborhood etc...

Useful for node classification, link prediction, community detection, visualization ...

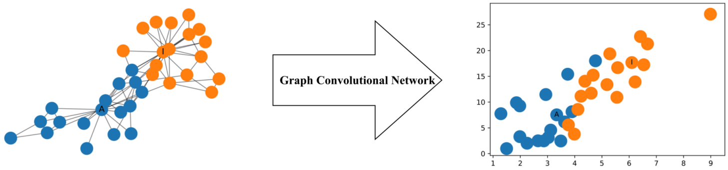


Image source: https://towardsdatascience.com

Signed graph convolutional networks, Derr et al (2018)

Two different aggregations based on structural balance:

- Unbalanced paths have an odd number of negative links
- Balanced paths have an even number of negative links

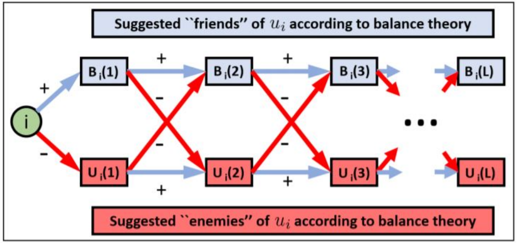


Image source: Derr et al.

Balanced and unbalanced sets $B_i(l+1)$ and $U_i(l+1)$ can be obtained from $B_i(l)$ and $U_i(l)$

Extending unsigned GCNs

Rather than maintaining a single representation for each node, keep both friends and enemies.

$$\mathbf{h}_i^{B(l)} = \sigma(\mathbf{W}^{B(l)}[\Sigma_{j \in \mathcal{N}_i^+} \frac{\mathbf{h}_j^{B(l-1)}}{|\mathcal{N}_i^+|}, \Sigma_{k \in \mathcal{N}_i^-} \frac{\mathbf{h}_k^{U(l-1)}}{|\mathcal{N}_i^{!}|}, \mathbf{h}_i^{B(l-1)}]$$

$$\mathbf{h}_i^{U(l)} = \sigma(\mathbf{W}^{U(l)}[\Sigma_{j \in \mathcal{N}_i^+} \frac{\mathbf{h}_j^{U(l-1)}}{|\mathcal{N}_i^+|}, \Sigma_{k \in \mathcal{N}_i^-} \frac{\mathbf{h}_k^{B(l-1)}}{|\mathcal{N}_i^{!}|}, \mathbf{h}_i^{U(l-1)}]$$

- $\mathbf{h}$ represents node features
- $\sigma()$ is a non-linear activation function
- $\mathbf{W}^{B(l)}$ and $\mathbf{W}^{U(l)}$ are linear transformation matrices responsible for "friends" and "enemies"

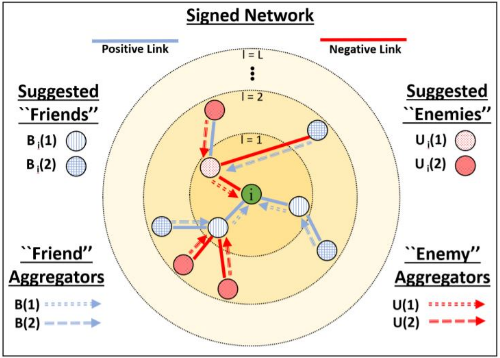


Image source: Derr et al.

Application: Link prediction

- SSE: spectral clustering algorithm based on signed Laplacian
- SiNE: deep learning framework that utilizes structural balance theory
- SIDE: random walk based method that utilizes structural balance theory
- SGCN-2: Proposed method. SGCN-1 and SGCN-1+ are structural balance-agnostic variants(only consider first aggregation layer)

LINK SIGN PREDICTION RESULTS WITH AUC.

Embedding Method	Bitcoin-Alpha	Bitcoin-OTC	Slashdot	Epinions
SSE	0.764	0.803	0.769	0.822
SiNE	0.778	0.814	0.792	0.849
SIDE	0.630	0.618	0.547	0.571
SGCN-1	0.780	0.818	0.784	0.663
SGCN-1+	0.785	0.817	**0.804**	0.722
SGCN-2	**0.796**	**0.823**	**0.804**	**0.864**

LINK SIGN PREDICTION RESULTS WITH F1.

Embedding Method	Bitcoin-Alpha	Bitcoin-OTC	Slashdot	Epinions
SSE	0.898	0.923	0.820	0.901
SiNE	0.888	0.878	0.854	0.914
SIDE	0.738	0.750	0.646	0.711
SGCN-1	0.910	0.918	0.853	0.851
SGCN-1+	0.912	0.923	**0.865**	0.893
SGCN-2	**0.917**	**0.925**	0.864	**0.933**

Conclusions and future directions

Conclusions

- Signed networks differ in terms of basic concepts, properties and present unique computational challenges
- In this tutorial we gave an overview of the literature in mining signed networks
  - We discussed the most important theoretical concepts
  - We discussed some common applications
- Rapidly growing research area
- In the remaining time we will mention:
  - Topics not covered
  - Open problems

Topics not covered in this tutorial

Signed networks have been the focus of many research works in the last few years

- Impossible to cover all topics in one tutorial...

Some important problem areas not covered in this tutorial:

- Node ranking (Wu et al. (2016b); Jung et al. (2016); Chung et al. (2013))
- Community detection (Gao et al. (2016); Li et al. (2014); Anchuri and Magdon-Ismail (2012))
- Dense subgraph discovery (Cadena et al. (2016))
- Recommendations (Tang et al. (2016); Victor et al. (2011); Ma et al. (2009))
- Many other papers in these areas not listed here...

Future directions

Fundamental combinatorial optimization problems still open. E.g.

- Approximability of vertex frustration and MBS.
- Approximability of polarized communities: $\max_{x \in \{-1,0,1\}} \frac{x^T A x}{x^T x}$.

For more theoretically-oriented challenges, check open problems in spectral theory listed by Belardo et al. (2019).

Graph construction in social media

- A lot of information under-used!
- Primary task: inferring edge signs (a link classification problem)
- Can we leverage multi-media information, e.g. user posts?
- Some attempt using crowd-sourcing, e.g. Kumar et al. (2018)
- Automatic way of doing this, e.g. using NLP techniques?

Graph summarization

General idea Koutra et al. (2018)

- Find a short representation of the input graph,
- often in the form of a summary or sparsified graph,
- which reveals patterns in the original data and preserves specific structural or other properties, depending on the application domain.

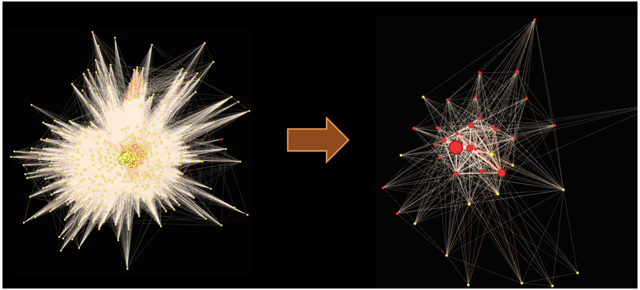
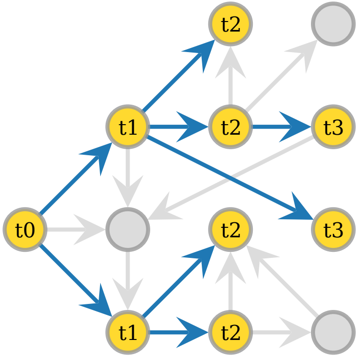


Image source: Koutra et al. (2018)

Propagation process modeling

- Real-world phenomena
  - Human disease
  - Computer virus
  - piece of information
- Is the propagation dynamics on signed graphs different from that on unsigned graphs?
- Inference tasks:
  - Source identification
  - Cascade reconstruction
  - Cascade prediction (size/popularity, node activation states)
  - Network structure inference



Local community detection

General setting

- Input: a graph and some seed nodes
- Output: a community relevant to the seed nodes

A rarely-touched topic in the context of signed graphs

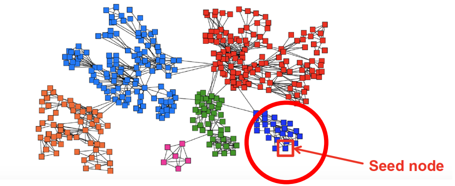


Image source:
http://web.stanford.edu/class/cs246/

Thank you and stay healthy !

References I

Acharya, B. D. (1980). Spectral criterion for cycle balance in networks. *Journal of Graph Theory*, 4(1):1–11.

Agarwal, A., Charikar, M., Makarychev, K., and Makarychev, Y. (2005). O (sqrt log n) approximation algorithms for min uncut, min 2cnf deletion, and directed cut problems. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 573–581.

Agrawal, P., Garg, V. K., and Narayanam, R. (2013). Link label prediction in signed social networks. In *Twenty-Third International Joint Conference on Artificial Intelligence*.

Ailon, N., Charikar, M., and Newman, A. (2005). Aggregating inconsistent information: ranking and clustering. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 684–693.

References II

Anchuri, P. and Magdon-Ismail, M. (2012). Communities and balance in signed networks: A spectral approach. In *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 235–242.

Aref, S. and Wilson, M. C. (2018). Measuring partial balance in signed networks. *Journal of Complex Networks*, 6(4):566–595.

Avidor, A. and Langberg, M. (2007). The multi-multiway cut problem. *Theoretical Computer Science*, 377(1-3):35–42.

Bai, C.-H. and Wu, B. Y. (2012). Finding the maximum balanced vertex set on complete graphs. In *THE 29TH WORKSHOP ON COMBINATORIAL MATHEMATICS AND COMPUTATION THEORY*. Citeseer.

Bansal, N., Blum, A., and Chawla, S. (2004). Correlation clustering. *Machine learning*, 56(1-3):89–113.

References III

Bartholdi III, J. J. (1982). A good submatrix is hard to find. *Operations Research Letters*, 1(5):190–193.

Belardo, F. (2014). Balancedness and the least eigenvalue of Laplacian of signed graphs. *Linear Algebra and its Applications*, 446:133–147.

Belardo, F., Cioabă, S. M., Koolen, J. H., and Wang, J. (2019). Open problems in the spectral theory of signed graphs. *arXiv preprint arXiv:1907.04349*.

Bonchi, F., Galimberti, E., Gionis, A., Ordozgoiti, B., and Ruffo, G. (2019). Discovering polarized communities in signed networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 961–970.

Bonchi, F., Garcia-Soriano, D., and Liberty, E. (2014). Correlation clustering: from theory to practice. In *KDD*.

References IV

Cadena, J., Vullikanti, A. K., and Aggarwal, C. C. (2016). On dense subgraphs in signed network streams. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 51–60. IEEE.

Cartwright, D. and Harary, F. (1956). Structural balance: a generalization of heider's theory. *Psychological review*, 63(5):277.

Cesa-Bianchi, N., Gentile, C., Vitale, F., and Zappella, G. (2012). A correlation clustering approach to link classification in signed networks. In *Annual Conference on Learning Theory*, volume 23, pages 34–1.

Chawla, S., Makarychev, K., Schramm, T., and Yaroslavtsev, G. (2015). Near optimal LP-rounding algorithm for correlation clustering on complete and complete k-partite graphs. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 219–228.

References V

Chiang, K.-Y., Natarajan, N., Tewari, A., and Dhillon, I. S. (2011). Exploiting longer cycles for link prediction in signed networks. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1157–1162.

Chiang, K.-Y., Whang, J. J., and Dhillon, I. S. (2012). Scalable clustering of signed networks using balance normalized cut. In *Proceedings of the 21st ACM international conference on Information and knowledge management (CIKM)*, pages 615–624.

Chu, L., Wang, Z., Pei, J., Wang, J., Zhao, Z., and Chen, E. (2016). Finding gangs in war from signed networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1505–1514. ACM.

Chung, F., Tsiatas, A., and Xu, W. (2013). Dirichlet pagerank and ranking algorithms based on trust and distrust. *Internet Mathematics*, 9(1):113–134.

References VI

Coleman, T., Saunderson, J., and Wirth, A. (2008). A local-search 2-approximation for 2-correlation-clustering. In *European Symposium on Algorithms*, pages 308–319.

Crowston, R., Gutin, G., Jones, M., and Muciaccia, G. (2013). Maximum balanced subgraph problem parameterized above lower bound. *Theoretical Computer Science*, 513:53–64.

Demaine, E. D., Emanuel, D., Fiat, A., and Immorlica, N. (2006). Correlation clustering in general weighted graphs. *Theoretical Computer Science*, 361(2-3):172–187.

El-Yaniv, R. and Pechyony, D. (2009). Transductive rademacher complexity and its applications. *Journal of Artificial Intelligence Research*, 35:193–234.

Elkin, M., Emek, Y., Spielman, D. A., and Teng, S.-H. (2008). Lower-stretch spanning trees. *SIAM Journal on Computing*, 38(2):608–628.

Figueiredo, R. and Frota, Y. (2014). The maximum balanced subgraph of a signed graph: Applications and solution approaches. *European Journal of Operational Research*, 236(2):473–487.

Figueiredo, R. M., Labbé, M., and De Souza, C. C. (2011). An exact approach to the problem of extracting an embedded network matrix. *Computers & operations research*, 38(11):1483–1492.

Gao, M., Lim, E.-P., Lo, D., and Prasetyo, P. K. (2016). On detecting maximal quasi antagonistic communities in signed graphs. *Data mining and knowledge discovery*, 30(1):99–146.

Giotis, I. and Guruswami, V. (2005). Correlation clustering with a fixed number of clusters. *arXiv preprint cs/0504023*.

Giscard, P.-L., Rochet, P., and Wilson, R. C. (2017). Evaluating balance on social networks from their simple cycles. *Journal of Complex Networks*, 5(5):750–775.

Goldberg, A. V. (1984). *Finding a maximum density subgraph*. University of California Berkeley.

Gómez, S., Jensen, P., and Arenas, A. (2009). Analysis of community structure in networks of correlated data. *Physical Review E*, 80(1):016114.

Gülpinar, N., Gutin, G., Mitra, G., and Zverovitch, A. (2004). Extracting pure network submatrices in linear programs using signed graphs. *Discrete Applied Mathematics*, 137(3):359–372.

Hansen, P. (1984). Shortest paths in signed graphs. In *North-Holland mathematics studies*, volume 95, pages 201–214. Elsevier.

Harary, F. (1953). On the notion of balance of a signed graph. *The Michigan Mathematical Journal*, 2(2):143–146.

Harary, F., Lim, M.-H., and Wunsch, D. C. (2002). Signed graphs for portfolio analysis in risk management. *IMA Journal of management mathematics*, 13(3):201–210.

References IX

Hou, Y., Li, J., and Pan, Y. (2003). On the Laplacian eigenvalues of signed graphs. *Linear and Multilinear Algebra*, 51(1):21–30.

Hüffner, F., Betzler, N., and Niedermeier, R. (2007). Optimal edge deletions for signed graph balancing. In *International Workshop on Experimental and Efficient Algorithms*, pages 297–310.

Jiang, J. Q. (2015). Stochastic block model and exploratory analysis in signed networks. *Physical Review E*, 91(6):062805.

Jung, J., Jin, W., Sael, L., and Kang, U. (2016). Personalized ranking in signed networks using signed random walk with restart. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 973–978. IEEE.

Koutra, D., Vreeken, J., and Bonchi, F. (2018). Summarizing graphs at multiple scales: New trends. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 1097–1097.

References X

Kumar, S., Hamilton, W. L., Leskovec, J., and Jurafsky, D. (2018). Community interaction and conflict on the web. In *Proceedings of the 2018 World Wide Web Conference*, pages 933–943.

Kunegis, J., Schmidt, S., Lommatzsch, A., Lerner, J., De Luca, E. W., and Albayrak, S. (2010). Spectral analysis of signed graphs for clustering, prediction and visualization. In *Proceedings of the 2010 SIAM International Conference on Data Mining*, pages 559–570.

Leskovec, J., Huttenlocher, D., and Kleinberg, J. (2010). Predicting positive and negative links in online social networks. In *Proceedings of the 19th international conference on World wide web*, pages 641–650.

Li, H. S. and Li, H. H. (2016). A note on the least (normalized) laplacian eigenvalue of signed graphs. *Tamkang Journal of Mathematics*, 47(3):271–278.

Li, Y., Liu, J., and Liu, C. (2014). A comparative analysis of evolutionary and memetic algorithms for community detection from signed social networks. *Soft Computing*, 18(2):329–348.

Ma, H., Lyu, M. R., and King, I. (2009). Learning to recommend with trust and distrust relationships. In *Proceedings of the 3rd ACM conference on Recommender systems*, pages 189–196.

Máčajová, E., Raspaud, A., and Škoviera, M. (2014). The chromatic number of a signed graph. *arXiv preprint arXiv:1412.6349*.

Mercado, P., Gautier, A., Tudisco, F., and Hein, M. (2018). The power mean laplacian for multilayer graph clustering. *arXiv preprint arXiv:1803.00491*.

Mercado, P., Tudisco, F., and Hein, M. (2016). Clustering signed networks with the geometric mean of laplacians. In *Advances in Neural Information Processing Systems*, pages 4421–4429.

References XII

Mercado, P., Tudisco, F., and Hein, M. (2019). Spectral clustering of signed graphs via matrix power means. *arXiv preprint arXiv:1905.06230*.

Ordozgoiti, B., Matakos, A., and Gionis, A. (2020). Finding large balanced subgraphs in signed networks. *arXiv preprint arXiv:2002.00775*.

Read, K. E. (1954). Cultures of the central highlands, new guinea. *Southwestern Journal of Anthropology*, 10(1):1–43.

Tang, J., Aggarwal, C., and Liu, H. (2016). Recommendations in signed social networks. In *Proceedings of the 25th International Conference on World Wide Web*, pages 31–40.

Tang, J., Chang, S., Aggarwal, C., and Liu, H. (2015). Negative link prediction in social media. In *Proceedings of the eighth ACM international conference on web search and data mining*, pages 87–96.

References XIII

Terzi, E. and Winkler, M. (2011). A spectral algorithm for computing social balance. In *International Workshop on Algorithms and Models for the Web-Graph*, pages 1–13.

Traag, V. A. and Bruggeman, J. (2009). Community detection in networks with positive and negative links. *Physical Review E*, 80(3):036115.

Tsourakakis, C. E., Chen, T., Kakimura, N., and Pachocki, J. (2019). Novel dense subgraph discovery primitives: Risk aversion and exclusion queries. *arXiv preprint arXiv:1904.08178*.

Victor, P., Cornelis, C., De Cock, M., and Teredesai, A. M. (2011). Trust-and distrust-based recommendations for controversial reviews. *IEEE Intelligent Systems*, 26(1):48–55.

Wu, J., Zhang, L., Li, Y., and Jiao, Y. (2016a). Partition signed social networks via clustering dynamics. *Physica A: Statistical Mechanics and its Applications*, 443:568–582.

References XIV

Wu, Z., Aggarwal, C. C., and Sun, J. (2016b). The troll-trust model for ranking in signed networks. In *Proceedings of the Ninth ACM international conference on Web Search and Data Mining*, pages 447–456. ACM.

Xiao, H., Ordozgoiti, B., and Gionis, A. (2020). Searching for polarization in signed graphs: a local spectral approach. *arXiv preprint arXiv:2001.09410*.

Yang, B., Liu, X., Li, Y., and Zhao, X. (2017). Stochastic blockmodeling and variational bayes learning for signed network analysis. *IEEE Transactions on Knowledge and Data Engineering*, 29(9):2026–2039.

Zaslavsky, T. (1982). Signed graph coloring. *Discrete Mathematics*, 39(2):215–228.

Zaslavsky, T. (2012). A mathematical bibliography of signed and gain graphs and allied areas. *The Electronic Journal of Combinatorics*, pages DS8–Dec.