Characterizing edges in signed and vector-valued graphs

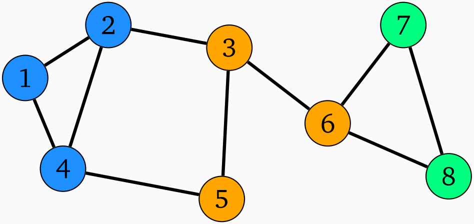
Géraud Le Falher, MAGNET team

April 16, 2018

Centre de Recherche en Informatique, Signal et Automatique de Lille

General context

- Machine Learning: automatically extract patterns from data and exploit them on future data.
- Here: relationship between data point; through a common structure called graph.



- Graphs support tasks such as community detection [For10], node classification [ST14], link prediction [MBC16] and influence maximization [KKT15].

Problem: predicting edge semantics

Various types of graph: directed or not, with or without node attributes and especially with more than one edge semantics

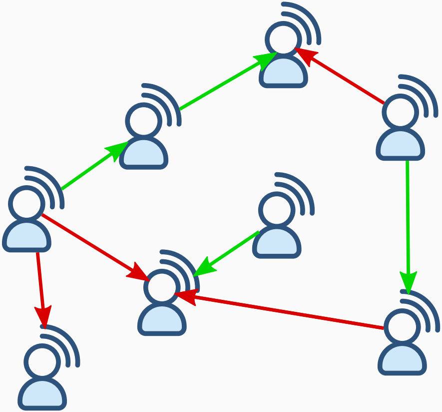
However, those edge semantics might be:

- costly to obtain
- too numerous
- missing

The task we address is thus predicting edges semantics

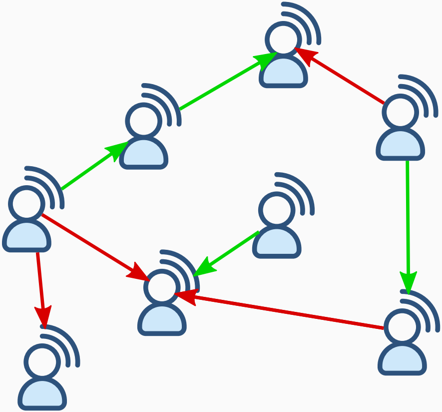
Type of graph – Wikipedia vote

Votes on Wikipedia: nodes are editors and edges represent votes for or against being promoted to administrators



Type of graph – Wikipedia vote

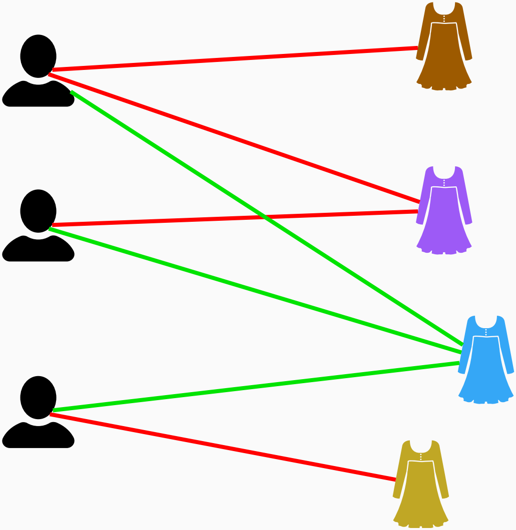
Votes on Wikipedia: nodes are editors and edges represent votes for or against being promoted to administrators



- Citations: *i* cites the work of *j* to praise it or criticise it.
- Slashdot: *i* considers *j* as a friend or foe.
- Epinion: *i* trusts or not the reviews made by *j*.

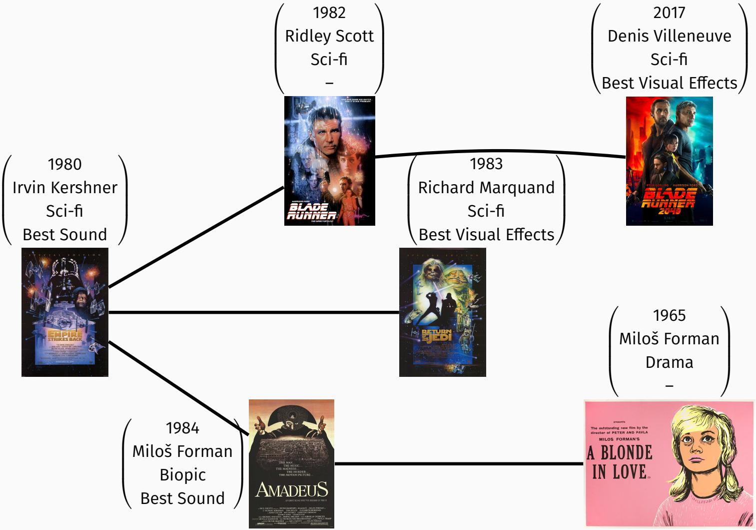
Type of graph – Bipartite purchase

Bipartite purchase network: nodes are customers and products, edges are reviews



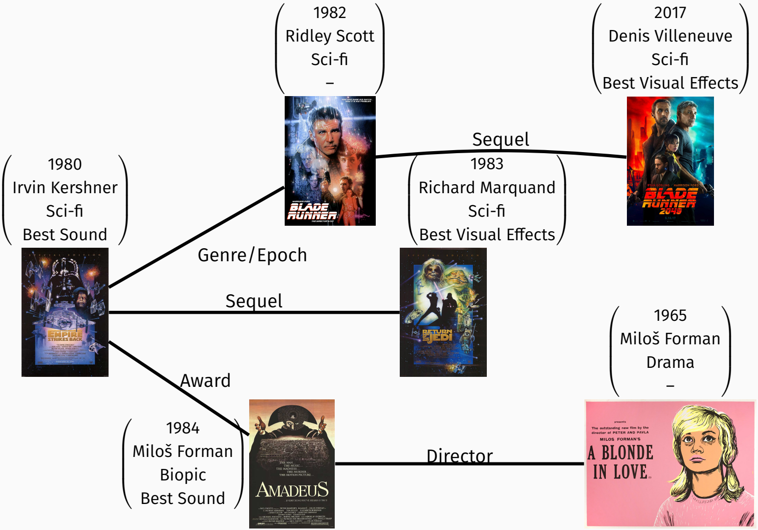
Type of graph – Co-purchase

Attributed co-purchase network: nodes are products and their characteristic, edges denotes “frequently purchased together”



Type of graph – Co-purchase

Attributed co-purchase network: nodes are products and their characteristic, edges denotes “frequently purchased together”



Thesis statement

There exist efficient and accurate methods to predict edge semantics in various types of graph, relying only on the graph topology or also on node attributes.

Outline

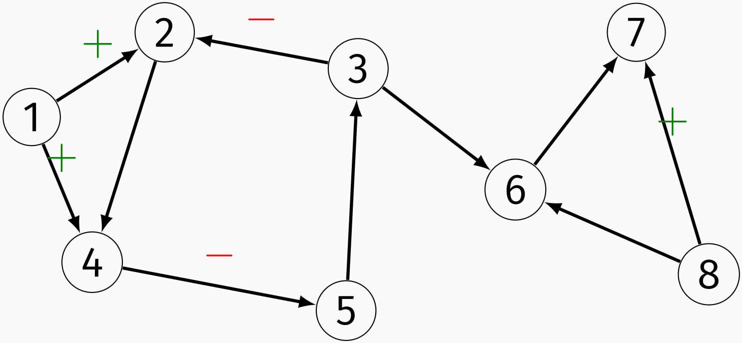
Applications

Predicting edge semantics in directed signed graphs is applicable to practical, real world problems:

- “Frenemy” detection [Yan+12];
- Automatic moderation of large scale online interactions
- Cyber bullying prevention, at school or in online games [JG08].

Problem statement

Given the topology of a directed graph and the signs of some edges: predict the remaining signs → batch binary classification



Existing approaches

Adjacency matrix completion [Chi+14; Wan+17]

PageRank-like node scores [WAS16]

Logistic regression trained on local edge features [LHK10; SM15]

Our two methods based on a generative model

Based on a simple sign generative model

Two methods for estimating the parameters of the model:

1. **BLC** is a local counting method
2. **Label Prop.** is a global label propagation method run on a transformed graph

Signs generative model

$p_u \in [0, 1]$ models the tendency of a node u to send positive links

$q_v \in [0, 1]$ models the tendency of a node v to receive positive links

The sign of an edge $u \to v$ is decided by their combination

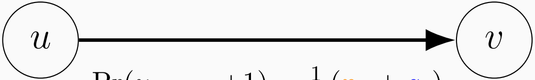
Signs generative model

$p_u \in [0, 1]$ models the tendency of a node u to send positive links

$q_v \in [0, 1]$ models the tendency of a node v to receive positive links

The sign of an edge $u \to v$ is decided by their combination

$(p_u, q_u) \sim \mu(p, q)$ $\quad\quad$ $(p_v, q_v) \sim \mu(p, q)$



$$\Pr(y_{u,v} = +1) = \frac{1}{2}\left(p_u + q_v\right)$$

The Bayes optimal prediction for the sign of $u \to v$ is thus

$$\begin{cases} +1 & \text{if } \frac{1}{2}\left(p_u + q_v\right) \geq \frac{1}{2} \\ -1 & \text{otherwise} \end{cases}$$

Compute estimators in linear time on the training set:

- $\widehat{p}_u$: fraction of training edges outgoing from u that are positive
- $\widehat{q}_v$: fraction of training edges incoming to v that are positive
- $\widehat{\tau}$: fraction of training edges that are positive

Predict the sign of $u \to v$ as:

$$\begin{cases} +1 & \text{if } \left(\widehat{p}_u + \widehat{q}_v - \widehat{\tau}\right) \geq \frac{1}{2} \\ -1 & \text{otherwise} \end{cases}$$

$\widehat{\tau}$ is needed because $\widehat{p}_u$ not only estimates p_u but also depends on q_v $\forall v \in \mathcal{N}_{\mathrm{out}}(u)$.

BLC GUARANTEE

Having $Q = \frac{1}{2\varepsilon^2} \ln \frac{4|V|}{\delta}$ samples per node and per direction,

$$\left| \left[\widehat{p}_u + \widehat{q}_v - \widehat{\tau} \right] - \left[\frac{p_u + q_v}{2} \right] \right| \leq 8\varepsilon$$

holds with probability at least $1 - 10\delta$ simultaneously for all non-queried edges $(u, v) \in E$ such that $d_{\mathrm{out}}(u), d_{\mathrm{in}}(v) \geq Q$.

Correct prediction as long as $\frac{p_u+q_v}{2}$ is bounded away from $\frac{1}{2}$.

BLC GUARANTEE DERIVATION

$\widehat{p}_u$ is not a correct estimation of p_u, but concentrates around its expectation:

$$\begin{aligned}\tfrac{1}{d_{\mathrm{out}}(u)} \sum_{v \in \mathcal{N}_{\mathrm{out}}(u)} \Pr\left(y_{u,v} = 1\right) &= \tfrac{1}{d_{\mathrm{out}}(u)} \sum_{v \in \mathcal{N}_{\mathrm{out}}(u)} \frac{p_u + q_v}{2} \\ &= \frac{1}{2}\left(p_u + \frac{1}{d_{\mathrm{out}}(u)} \sum_{v \in \mathcal{N}_{\mathrm{out}}(u)} q_v\right) = \frac{1}{2}\left(p_u + \overline{q}_u\right)\end{aligned}$$

where $\overline{q}_u$, being a sample mean of i.i.d. $[0, 1]$-valued random variables independently drawn from the prior marginal $\int_0^1 \mu(p, \cdot)dp$, concentrates around its expectation μ_q.

$$\widehat{p}_u \approx \frac{1}{2}\left(p_u + \mu_q\right)$$

BLC GUARANTEE DERIVATION

By a similar reasoning,

$$\widehat{q}_v \approx \frac{1}{2}\left(q_v + \overline{p}_v\right) \approx \frac{1}{2}\left(q_v + \mu_p\right)$$

Likewise, the expected fraction of positive edges is close to:

$$\frac{1}{2}\left(\mu_p + \mu_q\right)$$

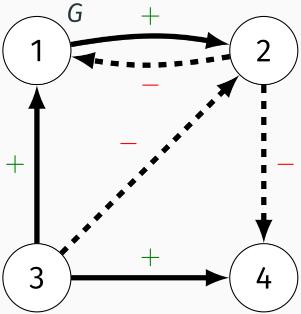
Thus the empirical fraction of positive edges $\widehat{\tau}$ satisfies:

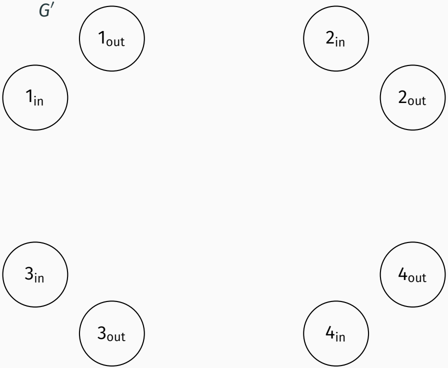
$$\widehat{\tau} \approx \frac{1}{2}\left(\mu_p + \mu_q\right)$$

and finally

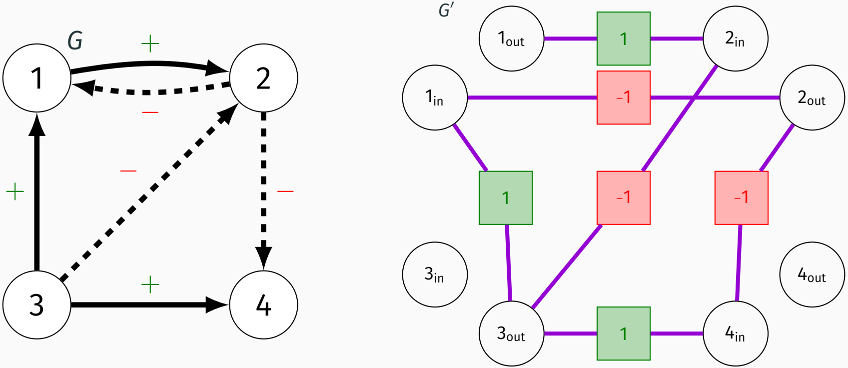
$$\widehat{p}_u + \widehat{q}_v - \widehat{\tau} \approx \frac{p_u + q_v}{2}$$

Transforming G into G' turns the problem into node classification



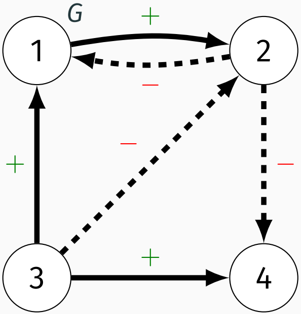


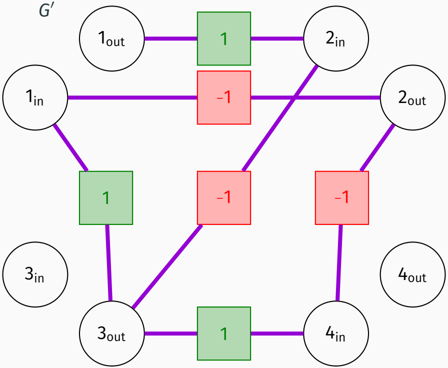
Transforming G into G' turns the problem into node classification



Label Prop. algorithm

Transforming G into G' turns the problem into node classification





Label propagation on this new graph approximates the maximum likelihood estimator of p and q

Both operations are $O(|E|)$

Experiments: Citations dataset

Table 1: 100× Matthews Correlation Coefficient results on Citations as the training set size increases

	5%	10%	15%	20%	25%	Global	time (ms)
BLC	**19.8**	**28.0**	**33.1**	**37.1**	39.7		**<1**
Label Prop.	**24.2**	**31.7**	**36.1**	**38.9**	**41.1**	✓	**19**
RankNodes [WAS16]	17.5	25.1	31.2	35.2	37.8	✓	157
MaxNorm [Wan+17]	1.2	12.6	22.2	30.3	36.5	✓	23229
LowRank [Chi+14]	12.4	17.9	22.0	25.7	29.0	✓	3222
Bayesian [SM15]	15.2	25.5	32.0	36.7	**39.8**		4787
16 Triads [LHK10]	11.4	17.2	21.0	24.3	27.0		7

Summary

- We add a simple sign generative model to directed graphs

Summary

- We add a simple sign generative model to directed graphs
- Estimating its parameters gives us two prediction methods, the local BLC and the global Label Prop.

Summary

- We add a simple sign generative model to directed graphs
- Estimating its parameters gives us two prediction methods, the local BLC and the global Label Prop.
- They are efficient: linear time and BLC is trivially parallelizable

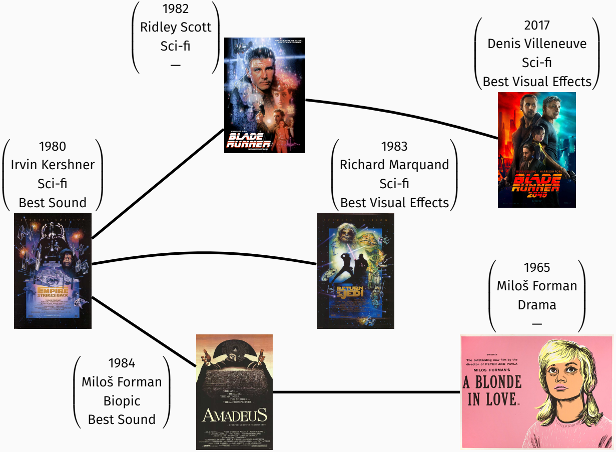
Summary

- We add a simple sign generative model to directed graphs
- Estimating its parameters gives us two prediction methods, the local BLC and the global Label Prop.
- They are efficient: linear time and BLC is trivially parallelizable
- They are accurate: BLC has theoretical guarantees and both are competitive in practice

Outline

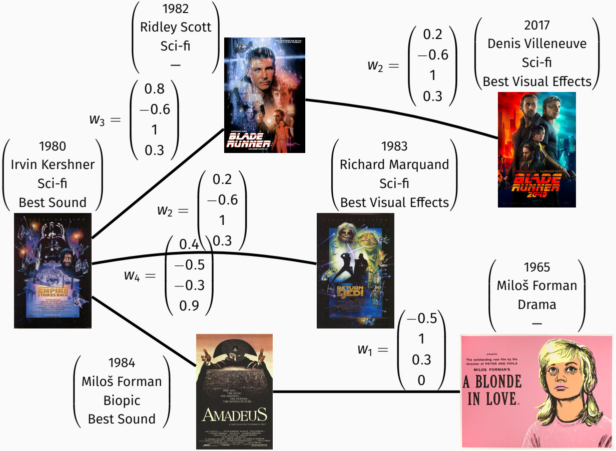
Problem

A graph is given, where nodes with attributes are connected for one of k reasons, but all reasons are unknown



Problem

A graph is given, where nodes with attributes are connected for one of k reasons, but all reasons are unknown



Find k semantic directions and associate to every edge the "most explanatory" direction among those k

Problem features and plan

Still predicting edge semantics, two big differences:

- more than 2 edge semantics $\rightarrow$ k-multilayer graphs [Kiv+14]
- no label are provided at any point $\rightarrow$ unsupervised problem

Problem features and plan

Still predicting edge semantics, two big differences:

- more than 2 edge semantics $\rightarrow$ k-multilayer graphs [Kiv+14]
- no label are provided at any point $\rightarrow$ unsupervised problem

1. Formalize an interpretable linear objective function

Problem features and plan

Still predicting edge semantics, two big differences:

- more than 2 edge semantics $\rightarrow$ k-multilayer graphs [Kiv+14]
- no label are provided at any point $\rightarrow$ unsupervised problem

1. Formalize an interpretable linear objective function
2. Optimize it with a scalar and matrix formulations

Problem features and plan

Still predicting edge semantics, two big differences:

- more than 2 edge semantics $\rightarrow$ k-multilayer graphs [Kiv+14]
- no label are provided at any point $\rightarrow$ unsupervised problem

1. Formalize an interpretable linear objective function
2. Optimize it with a scalar and matrix formulations
3. Experiment on synthetic data

Formalization – input

An undirected graph $G = (V, E)$

A number of semantic directions $k \in \mathbb{N}$

For each node $u \in V$, a unit norm profile $x_u \in [-1, 1]^d$

Formalization – linear edge scoring

Combine node profiles into an edge profile by entrywise product $\rightarrow$ signed similarity

x_u	x_v	$x_u \circ x_v$
0.9	0.8	0.72
-0.8	-0.9	0.72
0.9	-0.8	-0.72
0.0	0.5	0.0

Formalization – linear edge scoring

Combine node profiles into an edge profile by entrywise product $\rightarrow$ signed similarity

Then score it with semantic direction w_ℓ:

$$(x_u \circ x_v)^T w_\ell$$

For instance, given x_u and x_v, and having found w_1 and w_2:

x_u	x_v	$x_u \circ x_v$	w_1	w_2
0.9	0.8	0.72	0.6	0.1
−0.8	−0.9	0.72	0.1	0.2
0.9	−0.8	−0.72	−0.7	0.6
0.0	0.5	0.0	0.1	0.6

we get that

$$(x_u \circ x_v)^T w_1 = 1.008 \quad \text{and} \quad (x_u \circ x_v)^T w_2 = -0.216$$

Formalization – objective function

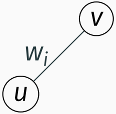
Associate the "most explanatory" direction to an edge u, v:

$$\max_{\ell \in \{1,\dots,k\}} \left(x_u \circ x_v\right)^T w_\ell \quad \text{Let } w_\ell = w(u, v)$$

Find the overall best k directions:

$$\operatorname*{arg\,max}_{\{w_1,\dots,w_k\} \subset \mathbb{B}^d} \sum_{u,v \in E} \max_{\ell \in \{1,\dots,k\}} \left(x_u \circ x_v\right)^T w_\ell$$

First topological regularization



For a single edge, the score is maximal when $x_u \circ x_v$ is collinear with w_i

In general, we expect node profiles to be (close to) a linear combination of their incident directions

$$\mathcal{L}_{\text{node}} = \sum_{u \in V} \left\| x_u - \left(\sum_{v \in \mathcal{N}(u)} a_{uv} w(u, v) + b_u \right) \right\|_2^2$$

Add more parameters ($\{a_{uv}\}_{u,v \in E} \in \mathbb{R}$ and $\{b_u\}_{u \in V} \in \mathbb{R}^d$) but restrict the search space

Second topological regularization

In high dimension, profiles cannot be too spread out

Each node can only be involved in $k_{\mathrm{local}} < k$ directions

$$\mathcal{L}_{\mathrm{local}} = \sum_{u \in V} \left\| \sum_{v \in \mathcal{N}(u)} c_{uv} \right\|_1 \text{ where } c_{uv} \in \mathbb{R}^d \text{ is an indicator vector}$$

c_{uv} i^{th} component is based on each direction score:

$$c_{uv;i} = \frac{\exp(\beta (x_u \circ x_v)^T w_i)}{\sum_{j=1}^{k} \exp(\beta (x_u \circ x_v)^T w_j)}$$

Three scalar solutions

1. Baseline: run k-**means** on the $|E|$ edge profiles $x_u \circ x_v$
2. Improvement: replace the euclidean distance in the **Lloyd** algorithm by our scoring function
3. Convexify the objective function by replacing max by softmax:

$$\operatorname*{arg\,max}_{\{w_1,\ldots,w_k\}\subset\mathbb{B}^d} \sum_{u,v\in E} \frac{1}{\beta} \log\left(\sum_{\ell=1}^{k} \exp\left(\beta\left(x_u \circ x_v\right)^T w_\ell\right)\right)$$

Combine it with the two convexified regularization terms, and optimize the difference by gradient descent, each step taking $O(|E|)$ to compute.

A more expressive matrix formulation

Find a direction w_{uv} for each edge:

$$\sum_{u,v \in E} \left(x_u \circ x_v\right)^T w_{uv} = \left\langle S^T, W \right\rangle_F$$

Make sure those directions are linear combination of k "base" directions $\rightarrow$ low rank matrix

$$\min_{W \in \mathbb{M}^{d \times |E|}} -\left\langle S^T, W \right\rangle_F + \text{rank}(W)$$

Two matrix optimization methods

4. Relax low rank by nuclear norm and ensure the norm of W columns are not too large.

$$\min_{\substack{W \in \mathbb{R}^{d \times |E|} \\ \|W\|_* \leq \delta}} \left\langle \mu W - S^T, W \right\rangle_F$$

   Optimized by the **Frank–Wolfe** [FW56] method, only requiring top singular vector at each iteration (computed in $O(d|E|)$).

5. Make the rank constraint **Explicit** and optimize the bi-convex expression

$$\min_{P \in \mathbb{R}^{d \times k}, Q \in \mathbb{R}^{|E| \times k}} -\left\langle S^T, PQ^T \right\rangle_F$$

   by alternating optimization

Contrasting scalar and matrix formulations

	Scalar (Combined)	Matrix (Frank–Wolfe)
time complexity	$O(\|E\|)$	$O(\|E\|)$
number of parameters	dk	$d\|E\|$
convex	no	yes
soft clustering	no	yes

Synthetic experiments

Generate a graph, 500 user profiles, k directions and assign one to each of the 1 300 edges

Evaluate how well we recover this planted assignment with Adjusted Mutual Information

$\mathcal{D}_k$	k-means	Lloyd	Combined	Frank Wolfe	Explicit
default	.818	*.873**	**.893***	.381	.893
$k = 5$	.836	*.838*	**.875***	.213	.875
$k = 9$	.803	*.881**	**.894***	.421	.894
$n_o = 6$	.813	*.824**	**.856***	.378	.855
$n_o = 12$	*.827*	.823	**.852***	.370	.851
$k_{\mathrm{local}} = 4$	.772	*.814**	**.853***	.320	.853
$d = 77$	.905	*.933**	**.941***	.222	.931

Summary

- Given a graph with node profiles, find k semantic directions explaining the existing connections

Summary

- Given a graph with node profiles, find *k* semantic directions explaining the existing connections
- Linear model for performance and interpretability, two formulations:
    - difference of convex scalar functions
    - Matrix convex expression suitable for Frank–Wolfe algorithm

Outline

Contributions

Efficient and accurate methods to predict edge semantics in various kind of graphs, relying only on the graph topology or also on node attributes.

	I	II	III
graph	directed, 2 semantics	attributed, k semantics	undirected, 2 semantics
learning setting	batch supervised	unsupervised	active supervised
approach	estimate parameters	matrix optimization	find spanning structure
efficient	$O(\|E\|)$, parallel	$O(dT\|E\|)$	$O(\|E\| \log \|E\|)$, conjectured
accurate	close to Bayes predictor, competitive in practice	convex problem: global max (but no ground truth)	Close to Correlation Clustering bound

Perspectives

Two ways relation with representation learning [Wil17] in graphs:

- Node embedding can be used as node attributes
- Give *k* adjacency matrices, use tensor factorization instead of matrix factorization for node embedding

References I

- S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3, pp. 75–174, 2010.
- A. Subramanya *et al.*, "Graph-based semi-supervised learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 8, no. 4, pp. 1–125, 2014.
- V. Martínez *et al.*, "A survey of link prediction in complex networks," *ACM Computer Survey*, vol. 49, no. 4, 69:1–69:33, 2016.
- D. Kempe *et al.*, "Maximizing the spread of influence through a social network," *Theory of Computing*, vol. 11, no. 4, pp. 105–147, 2015.

References II

S.-H. Yang *et al.*, "Friend or frenemy?" In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval - SIGIR '12*, 2012, p. 555.

J. Juvonen *et al.*, "Extending the school grounds?—bullying experiences in cyberspace," *Journal of School Health*, vol. 78, no. 9, pp. 496–505, 2008.

K.-y. Chiang *et al.*, "Prediction and clustering in signed networks: a local to global perspective," *Journal of Machine Learning Research*, vol. 15, pp. 1177–1213, 2014.

J. Wang *et al.*, "Online matrix completion for signed link prediction," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining - WSDM '17*, 2017, pp. 475–484.

References III

Z. Wu *et al.*, “The troll-trust model for ranking in signed networks,” in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, 2016, pp. 447–456.

J. Leskovec *et al.*, “Predicting positive and negative links in online social networks,” in *Proceedings of the 19th international conference on World wide web - WWW ’10*, 2010, p. 641.

D. Song *et al.*, “Link sign prediction and ranking in signed directed social networks,” *Social Network Analysis and Mining*, vol. 5, no. 1, p. 52, 2015.

M. Kivela *et al.*, “Multilayer networks,” en, *Journal of Complex Networks*, vol. 2, no. 3, pp. 203–271, 2014.

References IV

- M. Frank *et al.*, “An algorithm for quadratic programming,” *Naval Research Logistics Quarterly*, vol. 3, no. 1-2, pp. 95–110, 1956.
- William L. Hamilton and Rex Ying and Jure Leskovec, “Representation learning on graphs: methods and applications,” *Bulletin of the Technical Committee on Data Engineering*, vol. 40, no. 4, pp. 52–74, 2017.
- P. Auer *et al.*, “Adaptive and self-confident on-line learning algorithms,” *Journal of Computer and System Sciences*, vol. 64, no. 1, pp. 48–75, 2002.
- N. Cesa-Bianchi *et al.*, “A correlation clustering approach to link classification in signed networks,” in *Proceedings of the 25th Annual Conference on Learning Theory*, vol. 23, 2012, pp. 1–20.

Thank you!

Questions?

Online setting of sign prediction in directed graphs

The signs are **adversarial** rather than generated by our model.

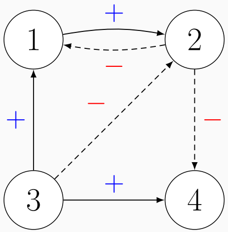
At each round, the learner is asked to predict one label, which is then revealed to him and the procedure repeats.

Labeling regularity

Letting Y be the vector of all labels, $\Psi_{\text{out}}(i, Y)$ is the number of least used label outgoing from i, and $\Psi_{\text{out}}(Y) = \sum_{i \in V} \Psi_{\text{out}}(i, Y)$.

Likewise for incoming edges, $\Psi_{\text{in}}(Y) = \sum_{j \in V} \Psi_{\text{in}}(j, Y)$ and finally $\Psi_G(Y) = \min\left\{\Psi_{\text{in}}(Y), \Psi_{\text{out}}(Y)\right\}$.

node i	1	2	3	4	
$\Psi_{\text{out}}(i, Y)$	0	0	1	0	$\Psi_{\text{out}}(Y) = 1$
$\Psi_{\text{in}}(i, Y)$	1	0	0	1	$\Psi_{\text{in}}(Y) = 2$



Online algorithm and bounds

- Our algorithm is a combination of Randomized Weighted Majority instances built on top of each other.

Online algorithm and bounds

- Our algorithm is a combination of Randomized Weighted Majority instances built on top of each other.
- On average, it makes $\Psi_G(Y) + O\left(\sqrt{|V|\Psi_G(Y)} + |V|\right)$ mistakes.

Online algorithm and bounds

- Our algorithm is a combination of Randomized Weighted Majority instances built on top of each other.
- On average, it makes $\Psi_G(Y) + O\left(\sqrt{|V|\Psi_G(Y)} + |V|\right)$ mistakes.
- On the lower side, for any directed graph G and any integer K, there exists a labeling Y forcing at least $\frac{K}{2}$ mistakes to any online algorithms, while $\Psi_G(Y) \leq K$.

Online algorithm, 1. RWM node instances

For each node i, we predict the sign of edge outgoing from i by relying on two constant experts, always predicting -1 or always predicting $+1$. The best one will make $\Psi_{\mathrm{out}}(i, Y)$ mistakes. We combine them in a Randomized Weighted Majority algorithm (RWM) instance associated with i, call it $RWM_{out}(i)$. The instance expected number of mistakes is therefore [ACG02], denoting by $M(i, j)$ the indicator function of a mistake on edge (i, j)

$$\sum_{j \in \mathcal{E}_{\mathrm{out}}(i)} \mathbb{E}\, M(i, j) = \Psi_{\mathrm{out}}(i, Y) + O\left(\sqrt{\Psi_{\mathrm{out}}(i, Y)} + 1\right)$$

We use the same technique to predict incoming edges of each node j, the instance $RWM_{in}(j)$ having the following average number of mistakes

$$\sum_{i \in \mathcal{E}_{\mathrm{in}}(j)} \mathbb{E}\, M(i, j) = \Psi_{\mathrm{in}}(j, Y) + O\left(\sqrt{\Psi_{\mathrm{in}}(j, Y)} + 1\right)$$

Online algorithm, 2. combining instances

We then define two meta experts: RWM_{out}, which predicts $y_{i,j}$ as $RWM_{out}(i)$, and RWM_{in}, which predicts $y_{i,j}$ as $RWM_{in}(j)$. Summing over all nodes, the number of mistakes of these two experts satisfy

$$\sum_{i \in V} \sum_{j \in \mathcal{E}_{\mathrm{out}}(i)} \mathbb{E}\, M(i,j) = \Psi_{\mathrm{out}}(Y) + O\left(\sqrt{|V|\Psi_{\mathrm{out}}(Y)} + |V|\right)$$

$$\sum_{j \in V} \sum_{i \in \mathcal{E}_{\mathrm{in}}(j)} \mathbb{E}\, M(i,j) = \Psi_{\mathrm{in}}(Y) + O\left(\sqrt{|V|\Psi_{\mathrm{in}}(Y)} + |V|\right)$$

Online algorithm, 3. final prediction

Our final predictor is a RWM combination of RWM_{out} and RWM_{out}, whose expected number of mistakes is

$$\begin{aligned}\sum_{(i,j)\in E} \mathbb{E}\, M(i,j) &= \Psi_G(Y) + O\Bigg(\sqrt{|V|\Psi_G(Y)} + |V| \\ &\qquad + \sqrt{\Big(\Psi_G(Y) + |V| + \sqrt{|V|\Psi_G(Y)}\Big)}\Bigg) \\ &= \Psi_G(Y) + O\Big(\sqrt{|V|\Psi_G(Y)} + |V|\Big)\end{aligned}$$

Datasets properties

Table 2: Dataset properties.

Dataset	$\lvert V\rvert$	$\lvert E\rvert$	$\frac{\lvert E\rvert}{\lvert V\rvert}$	$\frac{\lvert E^+\rvert}{\lvert E\rvert}$	$\frac{\Psi_{G''}(Y)}{\lvert E\rvert}$	$\frac{\Psi_G(Y)}{\lvert E\rvert}$
Citations	4 831	39 452	8.1	72.33%	.076	.191
Wikipedia	7 114	103 108	14.5	78.79%	.063	.142
Slashdot	82 140	549 202	6.7	77.40%	.059	.143
Epinion	131 580	840 799	6.4	85.29%	.031	.074
Wik. Edits	138 587	740 106	5.3	87.89%	.034	.086

$$\Psi_{G''}(Y) = \min_{\boldsymbol{p},\boldsymbol{q}\in[0,1]^{|V|}} \sum_{(i,j)\in E} \left(\frac{1+y_{i,j}}{2} - \frac{p_i+q_j}{2} \right)^2$$

Data generation

- create a random graph and pick for each a set of k_{local} directions.
- pick for each edge (u, v) a direction index $y_{uv} \in [\![k]\!]$ among the ones shared by its endpoint.
- draw the k directions at random
- optimize the profiles so as to maximize the edges goodness, minimize the term $\mathcal{L}_{\mathrm{node}}$ and enforces as much as possible that for every edge $(u, v) \in E$, $\mathcal{E}(u, v) = y_{uv}$.

(take a look a Liva comment about Lagrangian relaxation of integer program...)

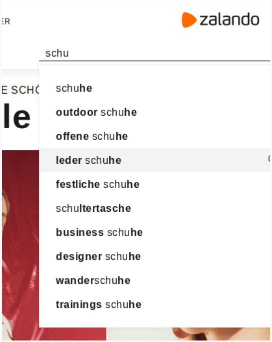
Experiments: Epinion dataset

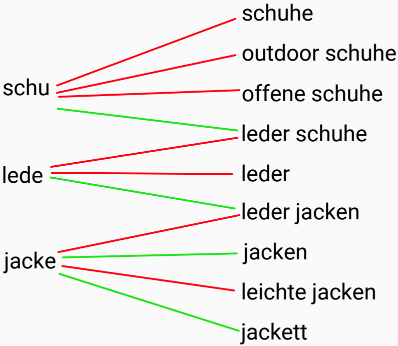
Table 3: 100× MCC results on Epinion as $|E_0|$ grows

	Global	3%	9%	15%	20%	25%	time (r
BLC		41.39	53.23	57.76	60.06	61.93	
Label Prop.	✓	**51.47**	**58.43**	**61.41**	**63.14**	**64.47**	**12**
RankNodes [WAS16]	✓	**52.04**	**60.21**	**62.69**	**64.13**	**65.22**	**23**
LowRank [Chi+14]	✓	36.84	43.95	48.61	51.43	54.51	1215
Bayesian [SM15]		31.00	48.24	56.88	61.49	64.45	1168
16 Triads [LHK10]		34.42	49.94	54.56	56.96	58.73	1

Problem

- Our sign generative model is not suitable for undirected graphs. Yet such graphs are important, for instance for recommendation in bipartite graphs.

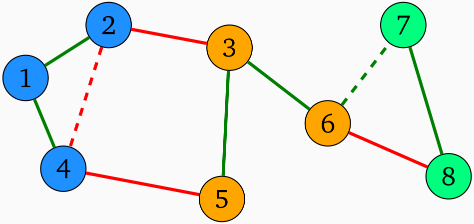




- We consider the active setting, where the learner first selects a training set, queries its signs and predicts the remaining edges.

New bias

Assume that nodes belongs to K latent groups, and that signs are governed by those groups, modulo some irregularities.



This is motivated:

in social networks by the balance theory (my friend's enemy is my enemy)

in other graphs by assortative/disassortative patterns

Connection with Correlation Clustering

- Recovering those K groups in the presence of noise is the well studied Correlation Clustering problem (CC).
- It's a NP-hard problem in general, but with effective solution on regular cases
- The minimal objective value of CC when $K = 2$ is a bound on the number of mistakes for any active algorithms [Ces+12].

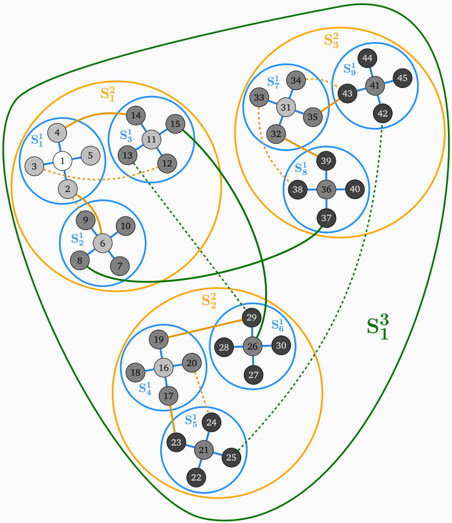
Solution: spanning structure

- An interesting case is $K = 2$ (strong balance), if there was no noise, the sign of any edge would the parity of any path between its endpoint.
- Since the noise is uniform (with probability q), the longer such path, the more likely we make an error by predicting using parity of observed edges. Overall, the expected number of errors is:

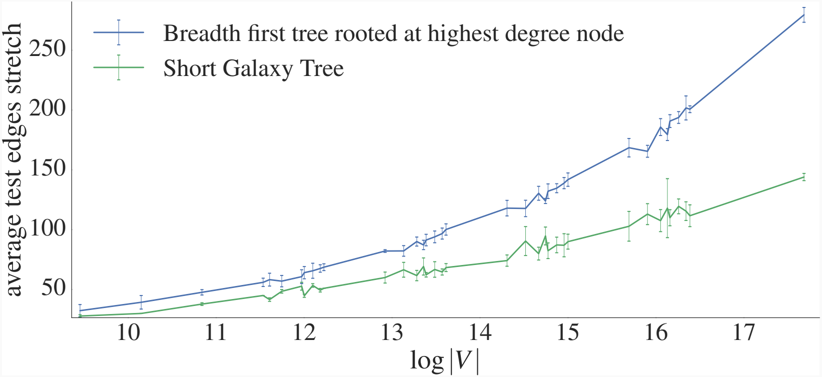
$$q\left(|E| + \sum_{(u,v)\in E_{\text{test}}} |\mathrm{path}^T(u,v)|\right)$$

- Thus we look for a spanning structure with few edges that aims to minimize the stretch.

Construction



Experiments - Stretch on grid graphs



Experiments - MCC on synthetic PA graphs