CS:4980:0004
MINING AND LEARNING ON LARGE NETWORKS

Joyce Yang

Lecture 24: Link Analysis

Fall 2020

Outline

- J. Leskovec, D. Huttenlocher, J. Kleinberg. Predicting Positive and Negative Links in Online Social Networks. (Links to an external site.) WWW 2010.

- L. Backstrom, J. Leskovec. Supervised Random Walks: Predicting and Recommending Links in Social Networks. (Links to an external site.) WSDM, 2011.

Predicting Positive and Negative Links in Online Social Networks

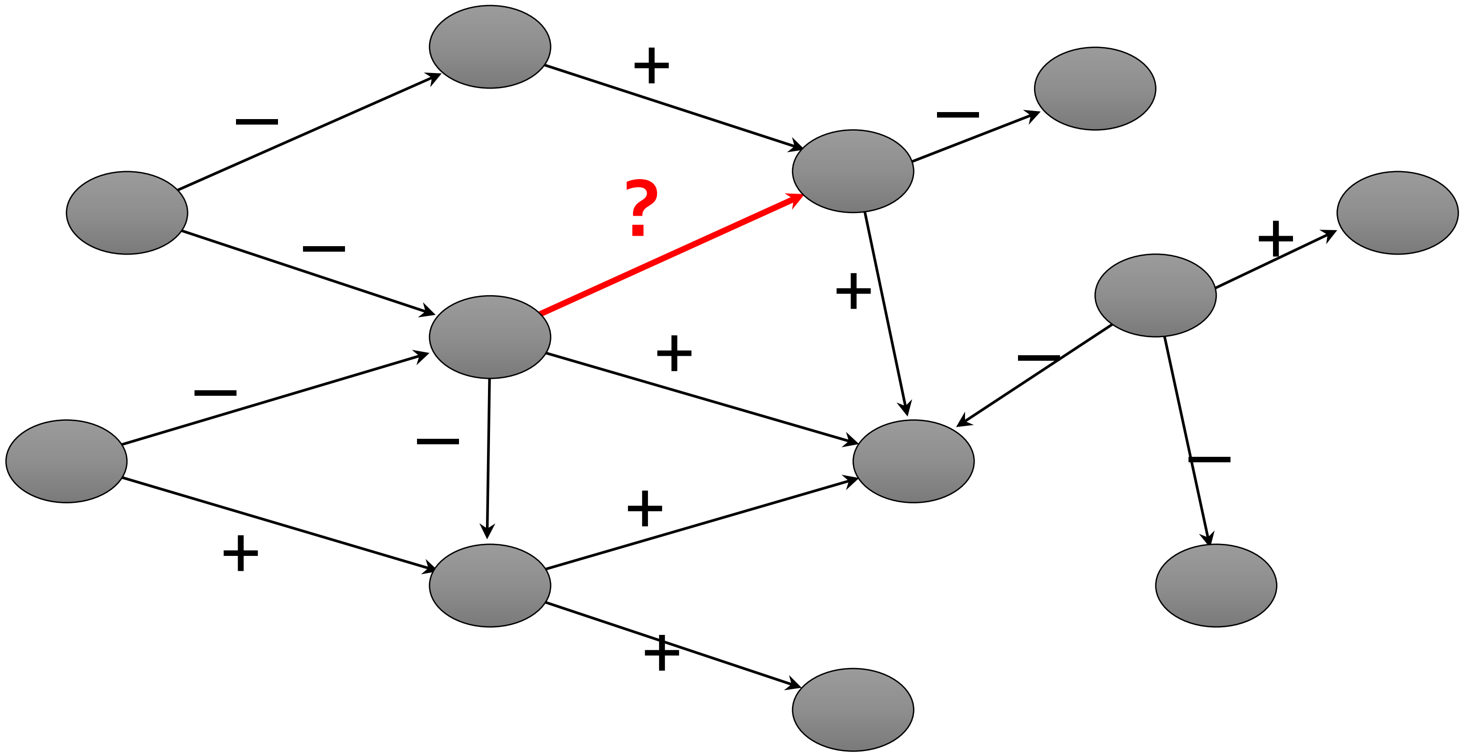
Sign of Links

Sign :

Positive or Negative

Task:

Edge Sign Prediction



Edge Sign Prediction

Suppose we are given a social network with signs on all its edges, but the sign on the edge from node u to node v, denoted s(u, v), has been “hidden.”

How reliably can we infer this sign s(u, v) using the information provided by the rest of the network?

Goals

- Achieve Better Performance on Task
- Find General Principles Across Datasets
- Prove theories of link signs from social psychology
  - Theories of *balance* and *status*.

Theories of Balance and Status

- **Balance** is a theory based on the principles that "the enemy of my friend is my enemy," "the friend of my enemy is my enemy," and variations on these.

- **Status** is a theory of signed link formation based on an implicit ordering of the nodes
  - Higher status / lower status

Dataset

	Epinions	Slashdot	Wikipedia
Nodes	119,217	82,144	7,118
Edges	841,200	549,202	103,747
+ edges	85.0%	77.4%	78.7%
− edges	15.0%	22.6%	21.2%

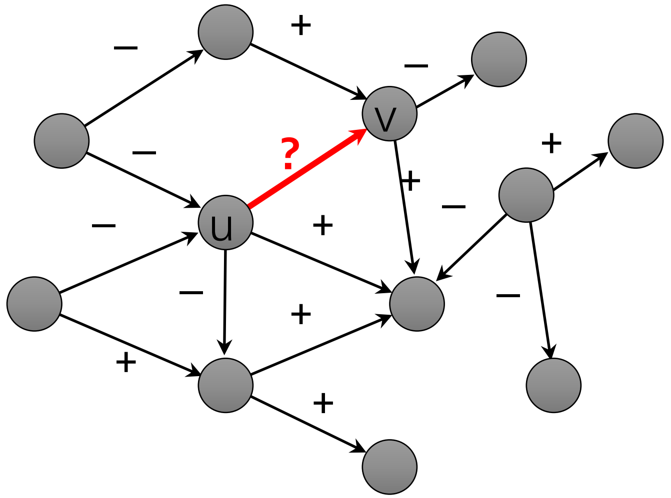
Edge Sign Prediction

Machine Learning formulation:

- Predict sign of edge (u,v)
- Class label:
  - +1: positive edge
  - -1: negative edge
- Learning method:
  - Logistic regression

$$P(+|x) = \frac{1}{1 + e^{-(b_0 + \sum_i^n b_i x_i)}}$$

- Evaluation:
  - Accuracy and ROC curves

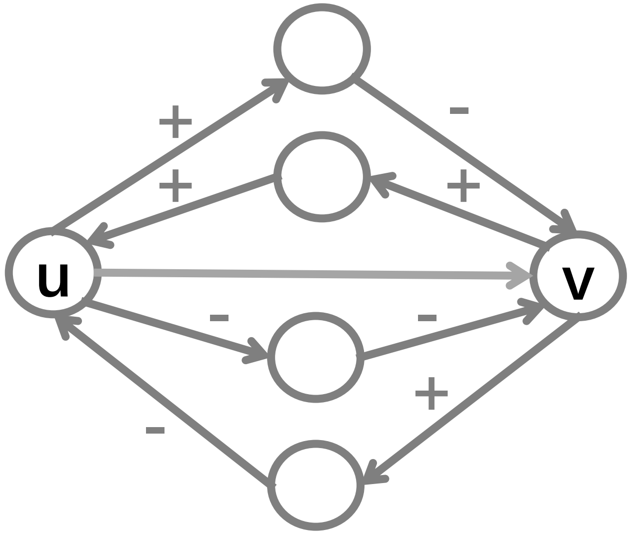


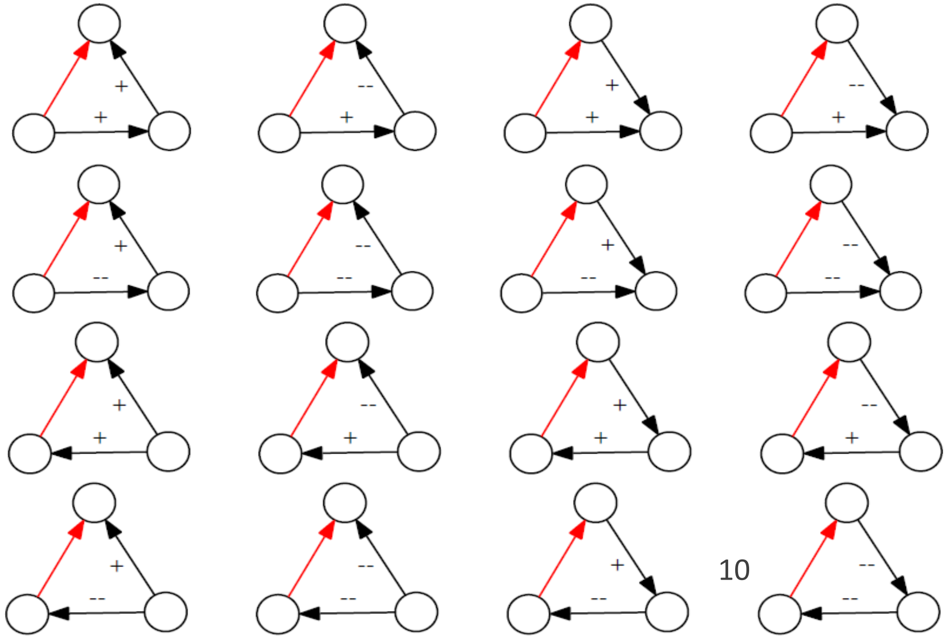
- Dataset:
  - Original: 80% +edges
  - Balanced: 50% +edges
- Features for learning:
  - Next slide

Features for Learning

For each edge (u,v) create features:

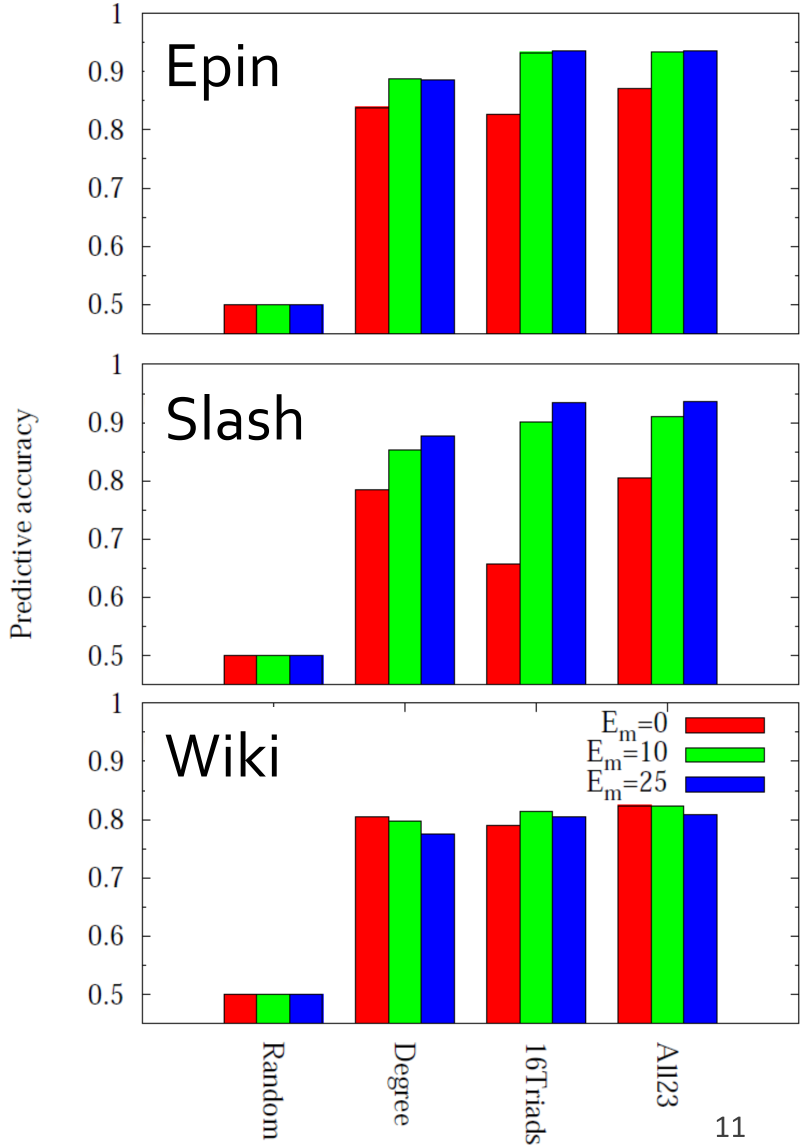
- Node degree (7 features):
  - Signed degree:
    - $d^{+}_{in}(v)$, $d^{-}_{in}(v)$
    - $d^{+}_{out}(u)$, $d^{-}_{out}(u)$
  - Total degree:
    - $d_{out}(u)$, $d_{in}(v)$
  - Embeddedness
    of edge (u,v) – C(u,v)
- Triad counts (16):
  - Counts of signed triads
    edge u→v takes part in





Result

- Error rates:
  - Epinions: 6.5%
  - Slashdot: 6.6%
  - Wikipedia: 19%
- Signs can be modeled from local network structure alone
  - Trust propagation model of [Guha et al. '04] has 14% error on Epinions
- Triad features perform less well for less embedded edges
- Wikipedia is harder to model:
  - Votes are publicly visible



Connection to Theories

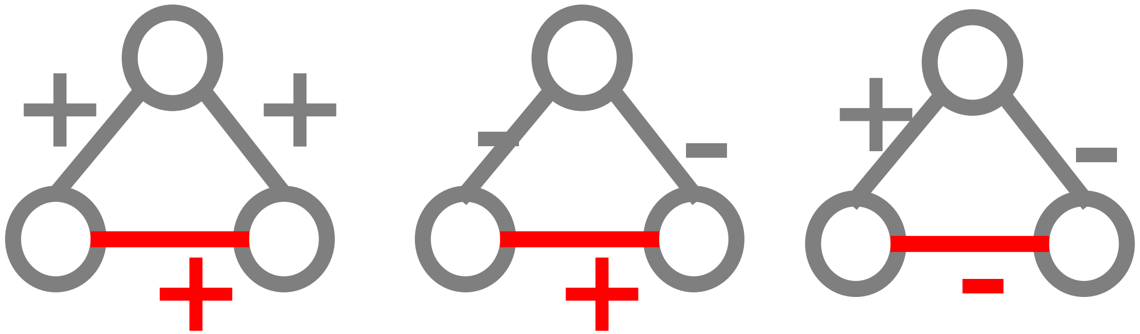
- Our goal is not just to predict signs but also to derive insights into usage of signed edges
- Connection to theories from social psychology:
  - Structural balance
  - Theory of status

  which both give predictions on the sign of the edge (u,v) based on the triad it is embedded into

Theory of Structural Balance

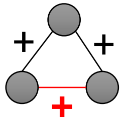
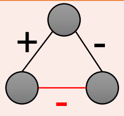
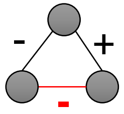
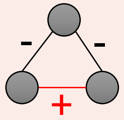
Consider edges as undirected

- Start with intuition [Heider '46]:
  - Friend of my friend is my friend
  - Enemy of enemy is my friend
  - Enemy of friend is my enemy
- Look at connected triples of nodes that are consistent with this logic:



Balance Theory

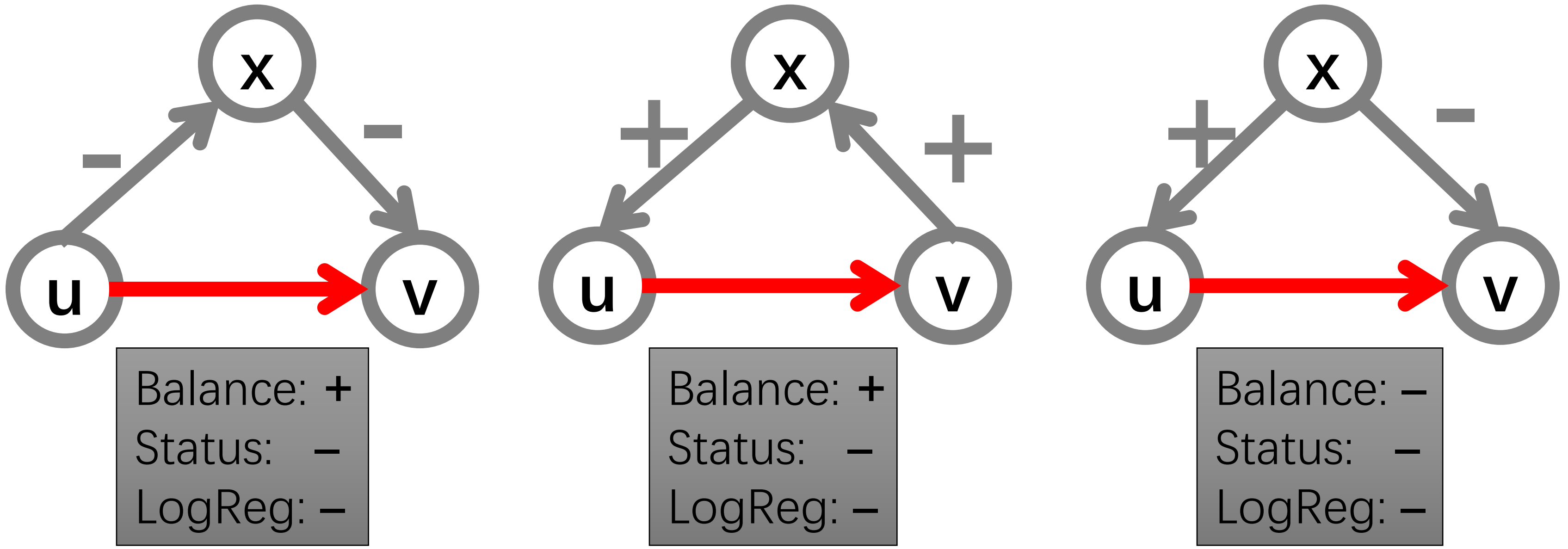
- Balance based and learned coefficients:

Feature	Balance theory	Epin	Slashdot	Wiki
const	0	0.43	1.49	0.04
	1	0.05	0.04	0.05
	-1	-0.11	-0.24	-0.16
	-1	-0.21	-0.35	-0.14
	1	-0.01	-0.03	-0.05

Model if signs would be created purely based on **Balance theory**

Theory of Status

- Status theory [Davis-Leinhardt ‘68, Guha et al. ’04, Leskovec et al. ‘10]
  - Link u $\xrightarrow[-]{+}$ v means: v has higher status than u
  - Link u $\rightarrow$ v means: v has lower status than u
  - Based on signs/directions of links from/to node x make a prediction
- Status and balance can make different predictions:

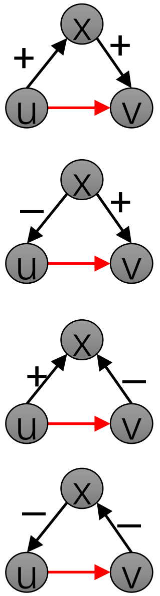


Status Theory

- Status based and learned coefficients:

Feature	Status theory	Epin	Slashdot	Wiki
const	0	-0.68	-1.39	-0.30
u < x < v	1	0.11	0.05	0.03
u > x > v	-1	-0.10	-0.11	-0.19
u < x > v	0	0.06	0.16	0.03
u > x < v	0	-0.01	0.04	0.05

Model if signs would be created purely based on **Status theory**



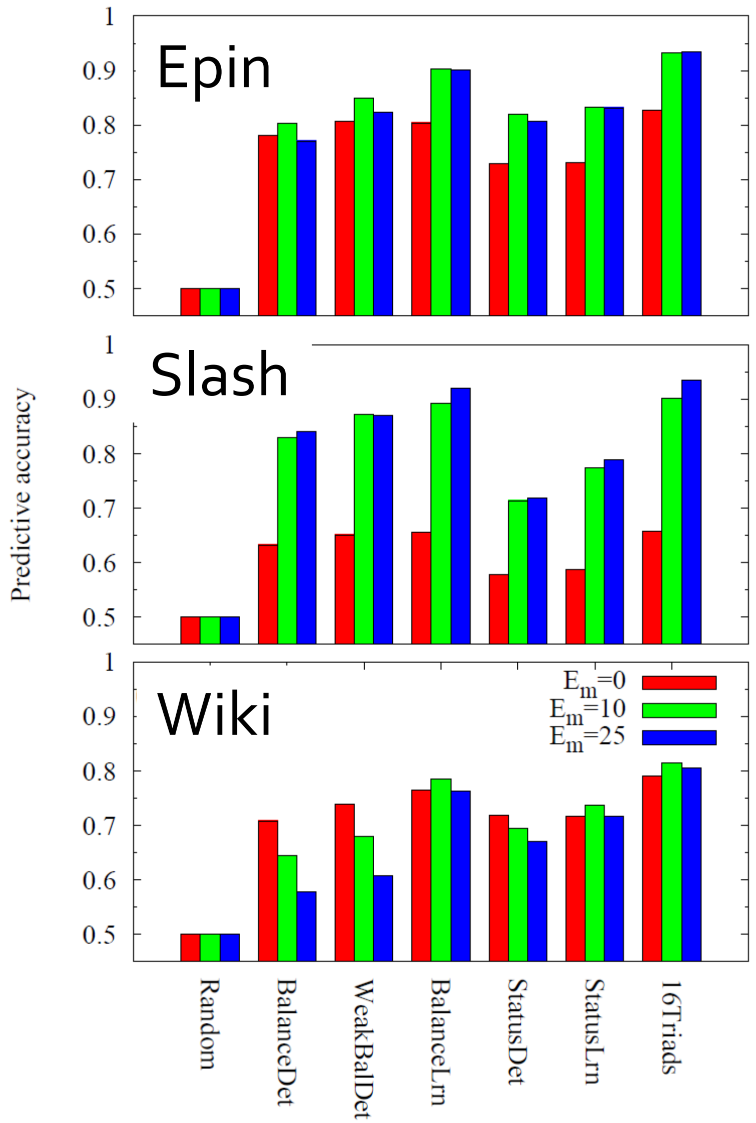
Triads where u > x > v

Complete model

Feature	Bal	Stat	**Epin**	**Slashd**	**Wikip**
const			-0.2	0.02	-0.2
$\bullet \xrightarrow{+} \bullet \xrightarrow{+} \bullet$	1	1	**0.5**	**0.9**	**0.3**
$\bullet \xrightarrow{+} \bullet \xrightarrow{-} \bullet$	-1	0	**-0.5**	**-0.9**	**-0.4**
$\bullet \xrightarrow{-} \bullet \xrightarrow{+} \bullet$	-1	0	**-0.4**	**-1.1**	**-0.3**
$\bullet \xrightarrow{-} \bullet \xrightarrow{-} \bullet$	1	-1	**-0.7**	**-0.6**	**-0.8**
$\circ \xrightarrow{+} \circ \xleftarrow{+} \circ$	1	0	0.3	0.4	0.05
$\circ \xrightarrow{+} \circ \xleftarrow{-} \circ$	-1	1	-0.01	-0.1	-0.01
$\bullet \xrightarrow{-} \bullet \xleftarrow{+} \bullet$	-1	-1	**-0.9**	**-1.2**	**-0.2**
$\circ \xrightarrow{-} \circ \xleftarrow{-} \circ$	1	0	0.04	-0.07	-0.03
$\circ \xleftarrow{+} \circ \xrightarrow{+} \circ$	1	0	0.08	0.4	0.1
$\bullet \xleftarrow{+} \bullet \xrightarrow{-} \bullet$	-1	-1	**-1.3**	**-1.1**	**-0.4**
$\circ \xleftarrow{-} \circ \xrightarrow{+} \circ$	-1	1	-0.1	-0.2	0.05
$\circ \xleftarrow{-} \circ \xrightarrow{-} \circ$	1	0	0.08	-0.02	-0.1
$\circ \xleftarrow{+} \circ \xleftarrow{+} \circ$	1	-1	-0.09	-0.09	-0.01
$\circ \xleftarrow{+} \circ \xleftarrow{-} \circ$	-1	0	-0.05	-0.3	-0.02
$\circ \xleftarrow{-} \circ \xleftarrow{+} \circ$	-1	0	-0.04	-0.3	0.05
$\circ \xleftarrow{-} \circ \xleftarrow{-} \circ$	1	1	-0.02	0.2	-0.2

Learned vs. Deterministic

- Deterministic models compare well to Learned models
- Epinions and Slashdot:
  - More embedded edges are easier to predict
- Wikipedia:
  - Status outperforms balance
- Learned balance performs nearly as well as the full model



Generalization

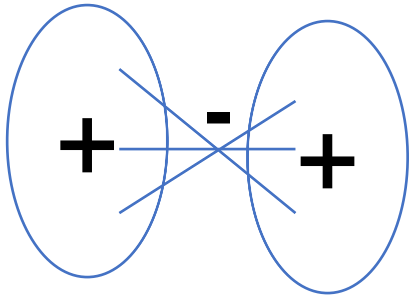
- Do people use these very different linking systems by obeying the same principles?
  - How generalizable are the results across the datasets?
    - Train on row “dataset”, predict on “column”

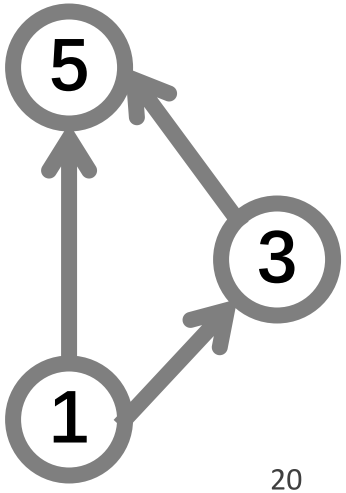
All23	Epinions	Slashdot	Wikipedia
Epinions	0.9342	0.9289	0.7722
Slashdot	0.9249	0.9351	0.7717
Wikipedia	0.9272	0.9260	0.8021

- Almost perfect generalization of the models even though networks come from very different applications

Local to Global Structure

- Both theories make predictions about the global structure of the network
- Structural balance – Factions
  - Put nodes into groups such that the number of in group "+" and between group "-" edges is maximized
- Status theory – Global Status
  - Flip direction and sign of negative edges
  - Assign each node a unique status value so that most edges point from low to high





Local to Global Structure

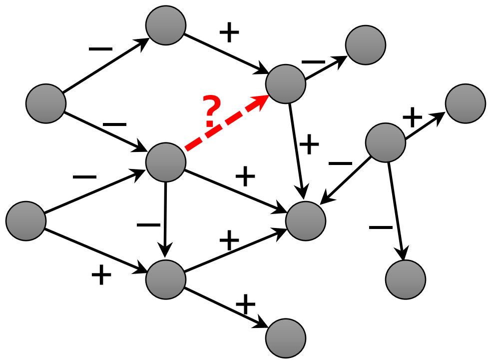
- Fraction of edges of the network that satisfy Balance and Status?

- Observations:
  - No evidence for global balance beyond the random baselines
    - Real data is 80% consistent vs. 80% consistency under random baseline
  - Evidence for global status beyond the random baselines
    - Real data is 80% consistent, but 50% consistency under random baseline

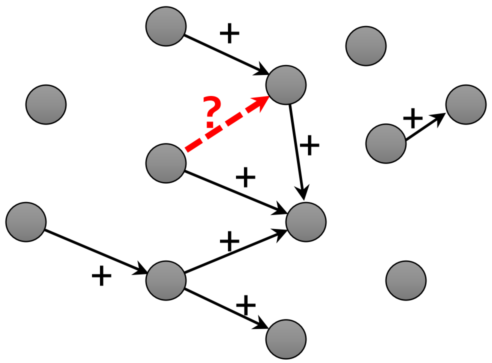
Does negative information help?

- Suppose we are only interested in predicting whether there is a trust edge or no edge
- Does knowing negative edges help?

YES!



Vs.



Features	Epinions	Slashdot	Wikipedia
Positive edges	0.5612	0.5579	0.6983
Positive and negative edges	0.5911	0.5953	0.7114

Conclusion

- Signed networks provide insight into how social computing systems are used:
  - Status vs. Balance
- Sign of relationship can be reliably predicted from the local network context
  - ~90% accuracy sign of the edge
- More evidence that networks are globally organized based on status
- People use signed edges consistently regardless of particular application
  - Near perfect generalization of models across datasets
- Negative information helps in predicting positive edges

Future work

- Better performance in basic sign prediction problem
- Strengthening the connections between local structure and global structure for signed links
- The global usage of positive and negative relationship

Supervised Random Walks: Predicting and Recommending Links in Social Networks.

Link Prediction

Challenge:

how to effectively combine the information from the network structure with rich node and edge attribute data?

Solution:

Supervised Random Walks

Challenges

- Real networks are sparse
    - No prediction = perfect accuracy
- How important is network structure or node features?

Supervised Random Walks

Supervised Random Walks - Bias a PageRank-like random walk

- Edge strengths – random walk transition probabilities

Possible Approaches:

- classification task
- rank nodes of the network

Both have problems => Combine two approaches

Problem Formulation

- Given:
  - Graph *G(V,E)*, node *s*, a set of candidates *C* (*d or l*)
  - Node features
- Logic:
  - Compute strength for edges
  - Apply random walk with restarts run from s
- Task:
  - learn the parameters w of function $f_w(\psi_{uv})$ that assigns each edge a transition probability a_{uv} .

Optimization Problem

constraints

$$\min_{w} F(w) = ||w||^2 + \boxed{\lambda \sum_{d \in D, l \in L} h(p_l - p_d)}$$

Optimization Problem

Random walk stochastic transition matrix

$$Q'_{uv} = \begin{cases} \frac{a_{uv}}{\sum_w a_{uw}} & \text{if } (u,v) \in E, \\ 0 & \text{otherwise} \end{cases}$$

Final random walk transition probability matrix

$$Q_{uv} = (1-\alpha)Q'_{uv} + \alpha \mathbf{1}(v = s).$$

Stationary Distribution

$$p^T = p^T Q$$

Derivative:

$$\frac{\partial F(w)}{\partial w} = 2w + \sum_{l,d} \frac{\partial h(p_l - p_d)}{\partial w}$$

$$= 2w + \sum_{l,d} \boxed{\frac{\partial h(\delta_{ld})}{\partial \delta_{ld}}} \boxed{\left(\frac{\partial p_l}{\partial w} - \frac{\partial p_d}{\partial w}\right)}$$

Hard to compute

Simple, common loss function

Optimization Problem

$$p^T = p^T Q \xrightarrow{\text{rewritten}} p_u = \sum_j p_j Q_{ju}$$

Derivative:

$$\begin{aligned}\frac{\partial F(w)}{\partial w} &= 2w + \sum_{l,d} \frac{\partial h(p_l - p_d)}{\partial w} \\ &= 2w + \sum_{l,d} \boxed{\frac{\partial h(\delta_{ld})}{\partial \delta_{ld}}} \boxed{\left(\frac{\partial p_l}{\partial w} - \frac{\partial p_d}{\partial w}\right)}\end{aligned}$$

Hard to compute

Simple, common loss function

$$\frac{\partial p_u}{\partial w} = \sum_j Q_{ju} \frac{\partial p_j}{\partial w} + p_j \frac{\partial Q_{ju}}{\partial w}$$

Optimization Problem

Initialize PageRank scores p and partial derivatives $\frac{\partial p_u}{\partial w_k}$:

  foreach $u \in V$ **do** $p_u^{(0)} = \frac{1}{|V|}$

  foreach $u \in V, k = 1, \ldots, |w|$ **do** $\frac{\partial p_u}{\partial w_k}^{(0)} = 0$

$t = 1$

while *not converged* **do**

  foreach $u \in V$ **do**

    $p_u^{(t)} = \sum_j p_j^{(t-1)} Q_{ju}$

  $t = t + 1$

$t = 1$

foreach $k = 1, \ldots, |w|$ **do**

  while *not converged* **do**

    foreach $u \in V$ **do**

      $\frac{\partial p_u}{\partial w_k}^{(t)} = \sum_j Q_{ju} \frac{\partial p_j}{\partial w_k}^{(t-1)} + p_j^{(t-1)} \boxed{\frac{\partial Q_{ju}}{\partial w_k}}$

    $t = t + 1$

return $\frac{\partial p_u}{\partial w}^{(t-1)}$

Algorithm 1: Iterative power-iterator like computation of PageRank vector p and its derivative $\frac{\partial p_u}{\partial w}$.

$$Q'_{uv} = \begin{cases} \frac{a_{uv}}{\sum_w a_{uw}} & \text{if } (u,v) \in E, \\ 0 & \text{otherwise} \end{cases}$$

$$\frac{\partial Q_{ju}}{\partial w} = (1-\alpha) \frac{\frac{\partial f_w(\psi_{ju})}{\partial w} \left(\sum_k f_w(\psi_{jk}) \right) - f_w(\psi_{ju}) \left(\sum_k \frac{\partial f_w(\psi_{jk})}{\partial w} \right)}{\left(\sum_k f_w(\psi_{jk}) \right)^2}$$

Experiments

- General consideration
  - Choice of the loss function
  - Choice of edge strength function
  - Choice of α.
  - Regularization parameter
- Dataset: four co-authorship networks and Facebook network of Iceland

Choice of the loss function

- Squared loss with margin b:

$$h(x) = \max\{x + b, 0\}^2$$

- Huber loss with margin b and window $z > b$:

$$h(x) = \begin{cases} 0 & \text{if } x \le -b, \\ (x+b)^2/(2z) & \text{if } -b < x \le z - b, \\ (x+b) - z/2 & \text{if } x > z - b \end{cases} \tag{7}$$

- Wilcoxon-Mann-Whitney (WMW) loss with width b (Proposed to be used when one aims to maximize AUC [32]):

$$h(x) = \frac{1}{1 + exp(-x/b)}$$

Choice of edge strength function

- Exponential edge strength: $a_{uv} = \exp(\psi_{uv} \cdot w)$
- Logistic edge strength: $a_{uv} = (1 + \exp(-\psi_{uv} \cdot w))^{-1}$

Choice of α.

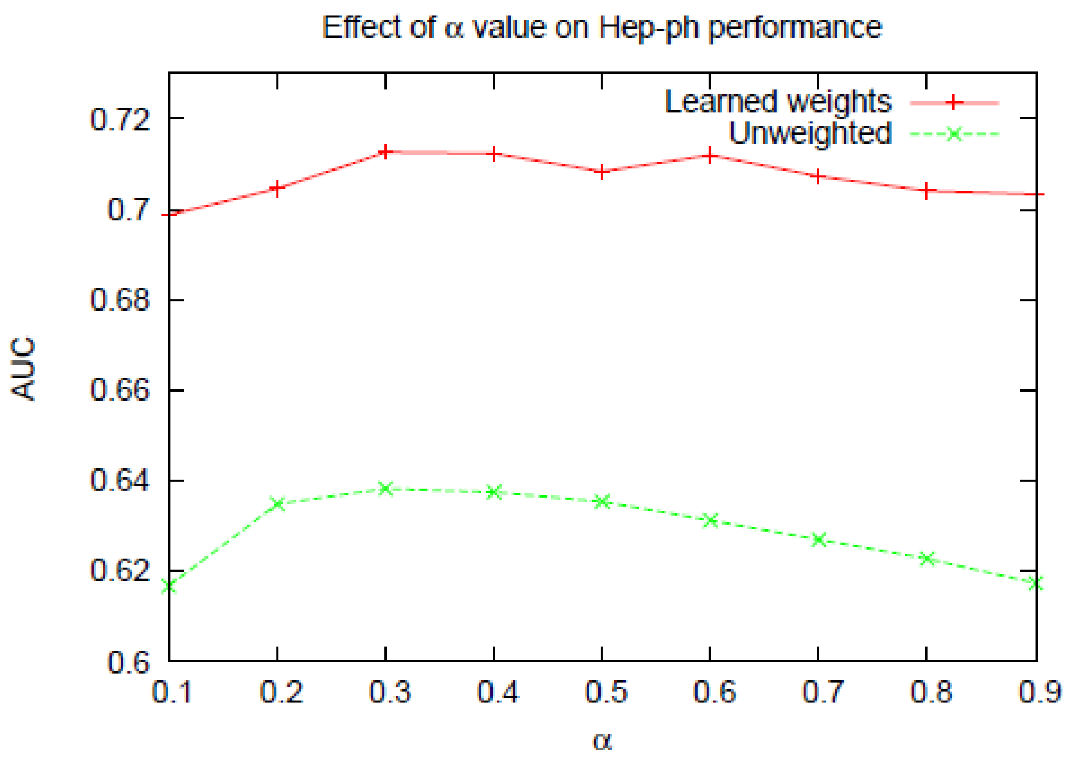


Figure 5: Impact of random walk restart parameter α.

Regularization parameter λ.

- Overfitting is not an issue in model as the number of parameters w is relatively small.
- Setting = 1 gives best performance.

Evaluation - Parameter estimation.

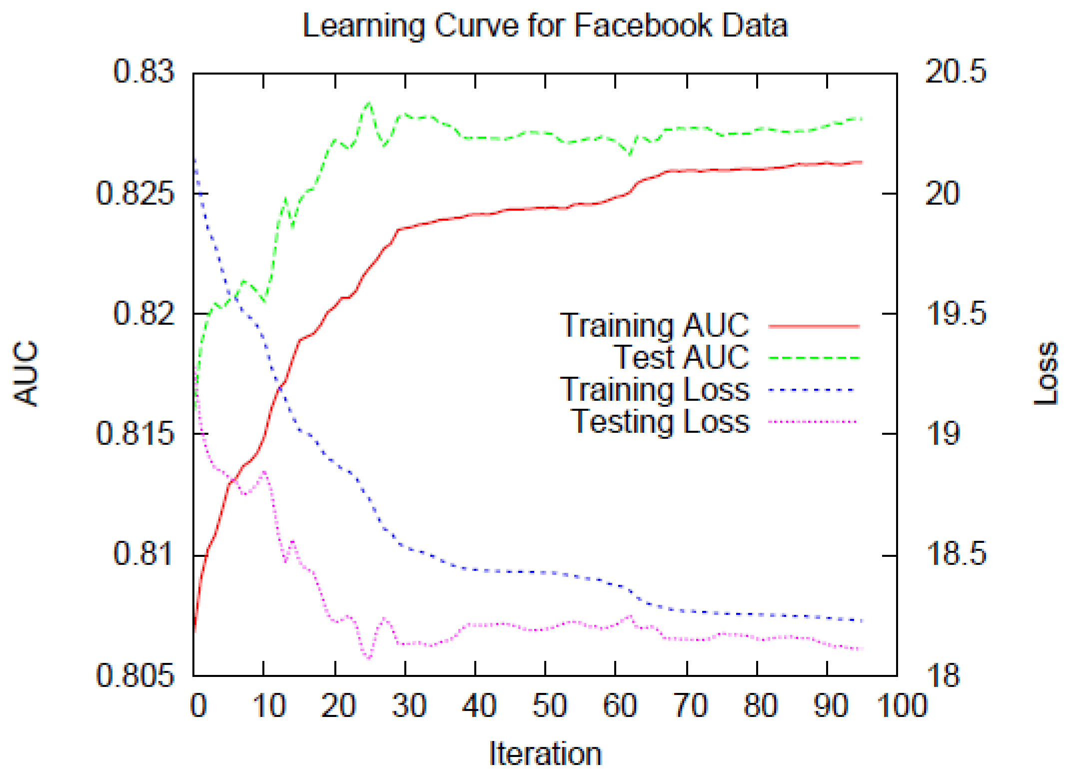


Figure 7: Performance of Supervised Random Walks as a function of the number of steps of parameter estimation procedure.

Evaluation - Comparsion

- unsupervised link prediction methods:
  - plain Random Walk with Restarts
  - Adamic-Adar score
  - number of common friends
  - node degree
- Supervised:
  - Decision Trees
  - logistic regression

- **Network features:** unweighted random walk scores, Adamic-Adar score, number of common friends, and degrees of nodes s and the potential target c
- **Node features:** average of the edge features for those edges' incident to the nodes s and c
- **Path features:** averaged edge features over all paths between seed s and the potential destination c.

Result

Learning Method	AUC	Prec@20
Random Walk with Restart	0.63831	3.41
Adamic-Adar	0.60570	3.13
Common Friends	0.59370	3.11
Degree	0.56522	3.05
DT: Node features	0.60961	3.54
DT: Network features	0.59302	3.69
DT: Node+Network	0.63711	3.95
DT: Path features	0.56213	1.72
DT: All features	0.61820	3.77
LR: Node features	0.64754	3.19
LR: Network features	0.58732	3.27
LR: Node+Network	0.64644	3.81
LR: Path features	0.67237	2.78
LR: All features	0.67426	3.82
SRW: one edge type	0.69996	4.24
SRW: multiple edge types	**0.71238**	**4.25**

Table 2: Hep-Ph co-authorship network. DT: decision tree, LR: logistic regression, and SRW: Supervised Random Walks.

Learning Method	AUC	Prec@20
Random Walk with Restart	0.81725	6.80
Degree	0.58535	3.25
DT: Node features	0.59248	2.38
DT: Path features	0.62836	2.46
DT: All features	0.72986	5.34
LR: Node features	0.54134	1.38
LR: Path features	0.51418	0.74
LR: All features	0.81681	7.52
SRW: one edge type	**0.82502**	6.87
SRW: multiple edge types	**0.82799**	**7.57**

Table 3: Results for the Facebook dataset.

Result

Dataset	AUC		Prec@20	
	SRW	LR	SRW	LR
Co-authorship Astro-Ph	**0.70548**	0.67639	2.55	2.15
Co-authorship Cond-Mat	**0.74173**	0.71672	2.54	2.61
Co-authorship Hep-Ph	**0.71238**	0.67426	4.18	3.82
Co-authorship Hep-Th	**0.72505**	0.69428	2.59	2.61
Facebook (Iceland)	**0.82799**	0.81681	7.57	7.52

Table 4: Results for all datasets. We compare favorably to logistic features as run on all features. Our Supervised Random Walks (SRW) perform significantly better than the baseline in all cases on ROC area. The variance is too high on the Top20 metric, and the two methods are statistically tied on this metric.

Conclusion

- Experiments on Facebook and coauthorship networks demonstrate good generalization and overall performance of Supervised Random Walks.
- Supervised RandomWalks are not limited to link prediction, and can be applied to many other problems that require learning to rank nodes in a graph, like recommendations, anomaly detection, missing link, and expertise search and ranking.

Thanks