1. Subgraph Analysis

2. Propagation Methods
   a) Background
   b) Normal Behavior
   c) Abnormal Behavior

3. Latent Factor Models
Given a graph, how can we find the “important” nodes?
Given the web, how can we find authoritative webpages?
Authoritative nodes and hubs: HITS

Authoritative nodes are pointed to by hubs

Hubs point to authoritative nodes

Authoritative Sources in a Hyperlinked Environment
Jon M. Kleinberg
JACM 1999
Authoritative nodes and hubs: HITS

Authoritative nodes are pointed to by hubs

Hubs point to authoritative nodes

Seems circular but still solvable!
Authoritative nodes and hubs: HITS

\[
\text{Authoritativness}(q) = \sum_{(p,q) \in E} \text{Hubness}(p)
\]

\[
\text{Hubness}(p) = \sum_{(p,q) \in E} \text{Authoritativness}(q)
\]

Seems circular but still solvable!

*We also keep authoritativness and hubness normalized
Authoritative nodes and hubs: HITS

Authoritativness \( (q) \) = \( \sum_{(p,q) \in E} \) Hubness \( (p) \)

Hubness \( (p) \) = \( \sum_{(p,q) \in E} \) Authoritativness \( (q) \)

Alternate updating both scores, repeatedly

*We also keep authoritativeness and hubness normalized
Authoritative nodes and hubs: HITS

Authoritative\textit{ness}(q) = \sum_{(p,q)\in E} \text{Hubness}(p)

Hubness(p) = \sum_{(p,q)\in E} \text{Authoritative\textit{ness}}(q)

Assume graph adjacent matrix $A$

- Authoritative\textit{ness} is the first \textbf{left singular vector} of $A$
- Hubness is the \textbf{first right singular vector} of $A$

Authoritative Sources in a Hyperlinked Environment
Jon M. Kleinberg
\textit{JACM} 1999
Authoritative nodes and hubs: HITS

Authoritiveness is the first left singular vector of $A$

Hubness is first right singular vector $A$
Authoritative nodes and hubs: HITS

Authoritativeness is the first left singular vector of $A$.
Hubness is first right singular vector $A$.

Authoritative Sources in a Hyperlinked Environment
Jon M. Kleinberg
JACM 1999
Authoritative nodes and hubs: HITS

Restrict graph by query “censorship”

Authorities:

eff.org  Electronic Frontier Foundation
eff.org/blueribbon.html  EFF Blue Ribbon Campaign
cdt.org  Center for Democracy and Tech
vtw.org  Voters Telecommunication Watch
aclu.org  American Civil Liberties Union
(Simplified) PageRank

Analyze random walk in graph
Anatomy of a Large-Scale Hypertextual Web Search Engine
Lawrence Page, Sergey Brin
WWW 1998

(Simplified) PageRank

Analyze random walk in graph
From a node, take each outgoing edge with equal probability.
(Simplified) PageRank

Analyze random walk in graph

From a node, take each outgoing edge with equal probability.

\[
\text{Rank}(p) = c \sum_{q \mid (p, q) \in E} \frac{\text{Rank}(q)}{\text{OutDegree}(q)}
\]

Anatomy of a Large-Scale Hypertextual Web Search Engine
Lawrence Page, Sergey Brin
WWW 1998
(Simplified) PageRank

$$\text{Rank}(p) = c \sum_{q \mid (p, q) \in E} \frac{\text{OutDegree}(q)}{R_{an}}$$

$A'$ is a column normalized adjacency matrix

$$A'_{p,q} \begin{cases} 
\frac{1}{\text{OutDegree}(q)} & \text{if } (p, q) \in E \\
0 & \text{if } (p, q) \notin E 
\end{cases}$$

Anatomy of a Large-Scale Hypertextual Web Search Engine
Lawrence Page, Sergey Brin
WWW 1998
(Simplified) PageRank

\[
\text{Rank}(p) = c \sum_{q \mid (p,q) \in E} \frac{\text{Rank}(q)}{\text{OutDegree}(q)}
\]

\(A'\) is a column normalized adjacency matrix

\[
A'_{p,q} = \begin{cases} 
1 & \text{if } (p,q) \in E \\
\frac{1}{\text{OutDegree}(q)} & \text{if } (p,q) \notin E \\
0 & \text{otherwise}
\end{cases}
\]

\[
\text{Rank} = cA'\text{Rank}
\]

Rank is the first eigenvector of \(cA'\)
PageRank

In “random walk,” jump to new node randomly with probability \((1-c)\)

\[
\text{Rank} = cA'\text{Rank} + \frac{1-c}{n} \mathbf{1}
\]

Anatomy of a Large-Scale Hypertextual Web Search Engine
Lawrence Page, Sergey Brin
WWW 1998
PageRank

In “random walk,” jump to new node randomly with probability (1-c)

\[ \text{Rank} = c A' \text{Rank} + \frac{1 - c}{n} \mathbf{1} \]
PageRank

In “random walk,” jump to new node randomly with probability $(1-c)$

$$\text{Rank} = cA'\text{Rank} + \frac{1 - c}{n} \mathbf{1}$$
PageRank

In “random walk,” jump to new node randomly with probability (1-c)

$$\text{Rank} = cA'\text{Rank} + \frac{1 - c}{n} \text{1}$$
PageRank

In “random walk,” jump to new node randomly with probability \((1-c)\)

\[
\text{Rank} = cA' \text{Rank} + \frac{1-c}{n} \mathbf{1}
\]

Rank is the first eigenvector of

\[
cA' + \frac{1-c}{n} \mathbf{1} \cdot \mathbf{1}^T
\]
PageRank

Random Walk with Restarts

\[ \overrightarrow{\text{Rank}} = cA'\overrightarrow{\text{Rank}} + \frac{1-c}{n} \mathbf{1} \]

Rank is the first eigenvector of

\[ cA' + \frac{1-c}{n} \mathbf{1} \cdot \mathbf{1}^T \]
### PageRank

<table>
<thead>
<tr>
<th>Web Page</th>
<th>PageRank (average is 1.0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Download Netscape Software</td>
<td>11589.00</td>
</tr>
<tr>
<td><a href="http://www.w3.org/">http://www.w3.org/</a></td>
<td>10717.70</td>
</tr>
<tr>
<td>Welcome to Netscape</td>
<td>8673.51</td>
</tr>
<tr>
<td>Point: It’s What You’re Searching For</td>
<td>7930.92</td>
</tr>
<tr>
<td>Web-Counter Home Page</td>
<td>7254.97</td>
</tr>
<tr>
<td>The Blue Ribbon Campaign for Online Free Speech</td>
<td>7010.39</td>
</tr>
<tr>
<td>CERN Welcome</td>
<td>6562.49</td>
</tr>
<tr>
<td>Yahoo!</td>
<td>6561.80</td>
</tr>
<tr>
<td>Welcome to Netscape</td>
<td>6203.47</td>
</tr>
<tr>
<td>Wusage 4.1: A Usage Statistics System For Web Servers</td>
<td>5963.27</td>
</tr>
<tr>
<td>The World Wide Web Consortium (W3C)</td>
<td>5672.21</td>
</tr>
<tr>
<td>Lycos, Inc. Home Page</td>
<td>4683.31</td>
</tr>
<tr>
<td>Starting Point</td>
<td>4501.98</td>
</tr>
<tr>
<td>Welcome to Magellan!</td>
<td>3866.82</td>
</tr>
<tr>
<td>Oracle Corporation</td>
<td>3587.63</td>
</tr>
</tbody>
</table>

Table 1: Top 15 Page Ranks: July 1996

Anatomy of a Large-Scale Hypertextual Web Search Engine
Lawrence Page, Sergey Brin
WWW 1998
### Practitioner’s Guide

<table>
<thead>
<tr>
<th>Method</th>
<th>Graph Type</th>
<th>Node Attributes</th>
<th>Edge Attributes</th>
<th>Seed Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>HITS</td>
<td>Directed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PageRank</td>
<td>Directed</td>
<td></td>
<td></td>
<td>Optional</td>
</tr>
<tr>
<td>Label Prop.</td>
<td>Undirected</td>
<td></td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>pMRF BP</td>
<td>Undirected</td>
<td></td>
<td></td>
<td>Preferred</td>
</tr>
<tr>
<td>EdgeExplain</td>
<td>Undirected</td>
<td>✔️</td>
<td></td>
<td>✔️</td>
</tr>
</tbody>
</table>
Semi-supervised Classification

Given a graph and labels for some nodes, can we learn the labels for the other nodes?
Semi-supervised Classification

Given a graph and labels for some nodes, can we learn the labels for the other nodes?

Huge research area with many different formulations

(we will present only a few)
Semi-supervised Classification

Given a graph and labels for some nodes, can we learn the labels for the other nodes?

Generally, learn labels $X$ so neighbors have the same label (homophily)
Semi-supervised Classification

\[ \min_{f} \sum_{i,j} w_{i,j} (f(i) - f(j))^2 \]

Edge weight \( w_{i,j} \)

\[ f(i) = \frac{1}{\sum_j w_{i,j}} \sum_j w_{i,j} f(j) \]

Can set label threshold for \( f(i) \)

Under this structure, closed form solution given in paper

Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions
Xiaojin Zhu, Zoubin Ghahramani, John Lafferty
ICML 2003
Semi-supervised Classification

Can view as a random walk:
- Go to neighbor with probability proportional to $w_{i,j}$
- Return node-prior with probability $p_v$
- Return no guess with probability $q_v$

$$\hat{f}(i) = p_i f(i) + q_i \frac{1}{2} + \frac{1}{\sum_j w_{i,j}} \sum_j w_{i,j} f(j)$$
Semi-supervised Classification

\[ \hat{y}_v = [p(A), p(B), p(C), p(\text{Don't Know})] \]

Observe prior labels \( y_v \) for some \( v \)

General prior \( r = [0,0,0,1] \)

\[ \min_{\hat{y}} \sum_{v,l} p_v (y_{v,l} - \hat{y}_{v,l})^2 + \sum_{u,v} w_{u,v} \| \hat{y}_u - \hat{y}_v \|^2 + \sum_v q_v \| \hat{y}_v - r \|^2 \]

New Regularized Algorithms for Transductive Learning
Partha Pratim Talukdar and Koby Crammer
ECML/PKDD 2009
Semi-supervised Classification

\[ \hat{y}_v = [p(A), p(B), p(C), p(Don't Know)] \]

Observe prior labels \( y_v \) for some \( v \)

General prior \( r = [0,0,0,1] \)

\( \min_{\hat{y}} \sum_{v,l} p_v (y_{v,l} - \hat{y}_{v,l})^2 + \sum_{u,v} w_{u,v} \| \hat{y}_u - \hat{y}_v \|^2 + \sum_{v} q_v \| \hat{y}_v - r \|^2 \)

New Regularized Algorithms for Transductive Learning
Partha Pratim Talukdar and Koby Crammer
ECML/PKDD 2009
Semi-supervised Classification

Can learn $\hat{y}$ through message passing of gradients

Predicted labels should match prior labels
Neighbors should have similar labels

Regularization

New Regularized Algorithms for Transductive Learning
Partha Pratim Talukdar and Koby Crammer
ECML/PKDD 2009
Semi-supervised Classifications: pMRF

Probabilistic interpretation with pairwise Markov random fields (pMRF)

Observe labels for nodes $\mathcal{L}$ and infer labels for nodes in $\mathcal{U}$

$$p(y_i \ \forall i \in \mathcal{U}) = \frac{1}{Z} \prod_i \phi_i(y_i) \prod_{(i,j) \in E} \psi_{i,j}(y_i, y_j)$$
Semi-supervised Classifications: pMRF

Probabilistic interpretation with pairwise Markov random fields (pMRF)

Observe labels for nodes $\mathcal{L}$ and infer labels for nodes in $\mathcal{U}$

$$p(y_i \ \forall i \in \mathcal{U}) = \frac{1}{Z} \prod_i \phi_i(y_i) \prod_{(i,j) \in E} \psi_{i,j}(y_i, y_j)$$

Prior belief for node $i$

Compatibility potentials between neighbors
Semi-supervised Classifications: pMRF

Probabilistic interpretation with pairwise Markov random fields (pMRF)

Observe labels for nodes $\mathcal{L}$ and infer labels for nodes in $\mathcal{U}$

$$p(y_i \ \forall i \in \mathcal{U}) = \frac{1}{Z} \prod_i \phi_i (y_i) \prod_{(i,j) \in E} \psi_{i,j} (y_i, y_j)$$

Compatibility potentials between neighbors

<table>
<thead>
<tr>
<th>$y_i$</th>
<th>$y_j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.8</td>
</tr>
<tr>
<td>B</td>
<td>0.2</td>
</tr>
</tbody>
</table>
Semi-supervised Classifications: pMRF

Probabilistic interpretation with pairwise Markov random fields (pMRF)

Observe labels for nodes $\mathcal{L}$ and infer labels for nodes in $\mathcal{U}$

$$p(y_i \ \forall i \in \mathcal{U}) = \frac{1}{Z} \prod_i \phi_i(y_i) \prod_{(i,j) \in E} \psi_{i,j}(y_i, y_j)$$

Use Loopy Belief Propagation [Pearl, 1982] to estimate most likely state.
Semi-supervised Classifications: pMRF

Use Loopy Belief Propagation [Pearl, 1982] to estimate most likely state.

$$p(y_i \ \forall i \in U) = \frac{1}{Z} \prod_i \phi_i(y_i) \prod_{(i,j) \in E} \psi_{i,j}(y_i, y_j)$$

Iteratively send messages between nodes to learn state.
Semi-supervised Classifications: pMRF

Use Loopy Belief Propagation [Pearl, 1982] to estimate most likely state.

\[
p(y_i \forall i \in U) = \frac{1}{Z} \prod_{i} \phi_i(y_i) \prod_{(i,j) \in E} \psi_{i,j}(y_i, y_j)
\]

Iteratively send messages between nodes to learn state.

Node \( y_i \) estimated by

\[
b_i(y_i) \leftarrow \alpha \phi_i(y_i) \prod_{(i,j) \in E} m_{j \rightarrow i}(y_i)
\]

Message from \( i \) to \( j \) given by

\[
m_{i \rightarrow j}(y_j) \leftarrow \sum_{y_i} \phi_i(y_i) \psi_{i,j}(y_i, y_j) \prod_{(i,k) \in E | k \neq j} m_{k \rightarrow i}(y_i)
\]
## Unifying Propagation Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Homophily</th>
<th>Heterophily</th>
<th>Convergence</th>
<th>Scalability</th>
</tr>
</thead>
<tbody>
<tr>
<td>RWR</td>
<td>✔️</td>
<td>✖️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>SSL</td>
<td>✔️</td>
<td>?</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>BP</td>
<td>✔️</td>
<td>✔️</td>
<td>?</td>
<td>✔️</td>
</tr>
</tbody>
</table>

Unifying Guilt-by-Association Approaches: Theorems and Fast Algorithms
Danai Koutra, Tai-You Ke, U Kang, Polo Chau, Hsing-Kuo Kenneth Pao, Christos Faloutsos
ECML/PKDD 2011
Unifying Propagation Methods

Can approximate all methods by:

<table>
<thead>
<tr>
<th>Method</th>
<th>matrix</th>
<th>unknown</th>
<th>known</th>
</tr>
</thead>
<tbody>
<tr>
<td>RWR</td>
<td>([\mathbf{I} - c\mathbf{AD}^{-1}]\times)</td>
<td>(x)</td>
<td>((1 - c) \mathbf{y})</td>
</tr>
<tr>
<td>SSL</td>
<td>([\mathbf{I} + \alpha(\mathbf{D} - \mathbf{A})]\times)</td>
<td>(x)</td>
<td>(\mathbf{y})</td>
</tr>
<tr>
<td>Gaussian BP = SSL</td>
<td>([\mathbf{I} + \alpha(\mathbf{D} - \mathbf{A})]\times)</td>
<td>(x)</td>
<td>(\mathbf{y})</td>
</tr>
</tbody>
</table>
1. Subgraph Analysis

2. Propagation Methods
   a) Background
   b) Normal Behavior
   c) Abnormal Behavior

3. Latent Factor Models
If we have incomplete labels of many types, can we infer the rest of the labels?

Idea:
Adjacent nodes only need to be agree on one label.

Explain friendships to infer user properties.
If we have incomplete labels of many types, can we infer the rest of the labels?

Idea:
Adjacent nodes only need to be agree on one label.
Explain friendships to infer user properties

If we have incomplete labels of many types, can we infer the rest of the labels?

Idea:
Adjacent nodes only need to be agree on one label.
Explain friendships to infer user properties

If we have incomplete labels of many types, can we infer the rest of the labels?

Idea:
Adjacent nodes only need to be agree on one label
For each user $u$ and property type $t$, we can observe 1 label from set $L(t)$, as given by vector $y_{u,t}$.

For each user $u$ and property type $t$, we can observe 1 label from set $L(t)$, as given by vector $y_{u,t}$.

For each user $u$ and property type $t$, we can observe 1 label from set $L(t)$, as given by vector $y_{u,t}$.

For each user $u$ and property type $t$, we can observe 1 label from set $L(t)$, as given by vector $y_{u,t}$.

For each user $u$ and property type $t$, we can observe 1 label from set $L(t)$, as given by vector $y_{u,t}$.

For each user $u$ and property type $t$, we can observe 1 label from set $L(t)$, as given by vector $y_{u,t}$.

For each user $u$ and property type $t$, we can observe 1 label from set $L(t)$, as given by vector $y_{u,t}$.

For each user $u$ and property type $t$, we can observe 1 label from set $L(t)$, as given by vector $y_{u,t}$.

For each user $u$ and property type $t$, we can observe 1 label from set $L(t)$, as given by vector $y_{u,t}$.

For each user $u$ and property type $t$, we can observe 1 label from set $L(t)$, as given by vector $y_{u,t}$.

For each user $u$ and property type $t$, we can observe 1 label from set $L(t)$, as given by vector $y_{u,t}$.

For each user $u$ and property type $t$, we can observe 1 label from set $L(t)$, as given by vector $y_{u,t}$.

For each user $u$ and property type $t$, we can observe 1 label from set $L(t)$, as given by vector $y_{u,t}$.

For each user $u$ and property type $t$, we can observe 1 label from set $L(t)$, as given by vector $y_{u,t}$.
Joint Inference of Multiple Label Types in Large Networks
Deepayan Chakrabarti, Stanislav Funiak, Jonathan Chang, Sofus A. Macskassy
ICML 2014

Sigmoid function so no additional benefit beyond being similar on one type

Probability vectors:
\[ \sum_{l \in L(t)} \hat{y}_{u,t,l} = 1 \]
Explain friendships to infer user properties

Recall (at 1) relative to Label Propagation

The plot shows lift in recall with respect to a fixed baseline of Label Propagation for graphs built with different number of friends. Increasing the number of friends increases recall up to a point, but then the extra friends introduce noise, which hurts accuracy. For example, in the case of Employer, the lift in recall is very significant when the number of friends is 100, but it drops significantly when the number of friends is increased to 400. This suggests that the number of friends should be balanced to achieve optimal recall.

Inclusion of group memberships is particularly attractive. Given the prior expectation of the impact of group memberships, this surprising result suggests that information from the group memberships actually turns out to be redundant. However, this gain largely disappears when the number of friends is increased further, indicating that the benefits of group memberships are not always consistent across different levels of friend connections.

Joint Inference of Multiple Label Types in Large Networks
Deepayan Chakrabarti, Stanislav Funiak, Jonathan Chang, Sofus A. Macskassy
ICML 2014
Joint Inference of Multiple Label Types in Large Networks
Deepayan Chakrabarti, Stanislav Funiak, Jonathan Chang, Sofus A. Macskassy
ICML 2014
1. Subgraph Analysis

2. Propagation Methods
   a) Background
   b) Normal Behavior
   c) Abnormal Behavior

3. Latent Factor Models
Find Fraud in HITS

Model a user’s following behavior:
- Out-degree
- Hubness
Find Fraud in HITS

Model a user’s following behavior:
- Out-degree
- Hubness

Surprising to have high out-degree but low hubness
Find Fraud in HITS

Model a user’s following behavior:
• Out-degree
• Hubness

Surprising to have high out-degree but low hubness
Find Fraud in HITS

Model a user’s followee behavior:
• In-degree
• Authoritiveness

Surprising to have high in-degree but low authoritativeness

CatchSync: Catching Synchronized Behavior in Large Directed Graphs
Meng Jiang, Peng Cui, Alex Beutel, Christos Faloutsos, Shiqiang Yang
KDD, 2014
Find Fraud in HITS

CatchSync: Catching Synchronized Behavior in Large Directed Graphs
Meng Jiang, Peng Cui, Alex Beutel, Christos Faloutsos, Shiqiang Yang
KDD, 2014
Find Fraud in HITS

CatchSync: Catching Synchronized Behavior in Large Directed Graphs
Meng Jiang, Peng Cui, Alex Beutel, Christos Faloutsos, Shiqiang Yang
KDD, 2014
Find Fraud in HITS

CatchSync: Catching Synchronized Behavior in Large Directed Graphs
Meng Jiang, Peng Cui, Alex Beutel, Christos Faloutsos, Shiqiang Yang
KDD, 2014
Find Fraud in HITS

Suspicious Behavior:
• Synchronized
• Abnormal

CatchSync: Catching Synchronized Behavior in Large Directed Graphs
Meng Jiang, Peng Cui, Alex Beutel, Christos Faloutsos, Shiqiang Yang
KDD, 2014
Find Fraud in HITS

CatchSync: Catching Synchronized Behavior in Large Directed Graphs
Meng Jiang, Peng Cui, Alex Beutel, Christos Faloutsos, Shiqiang Yang
KDD, 2014
Find Fraud in HITS

CatchSync: Catching Synchronized Behavior in Large Directed Graphs
Meng Jiang, Peng Cui, Alex Beutel, Christos Faloutsos, Shiqiang Yang
KDD, 2014
Find Fraud in HITS

- CatchSync
- CatchSync + SPOT: 0.813
- CatchSync: 0.751
- SPOT: 0.597
- OutRank: 0.412

CatchSync: Catching Synchronized Behavior in Large Directed Graphs
Meng Jiang, Peng Cui, Alex Beutel, Christos Faloutsos, Shiqiang Yang
KDD, 2014
Find Fraud in HITS

CatchSync: Catching Synchronized Behavior in Large Directed Graphs
Meng Jiang, Peng Cui, Alex Beutel, Christos Faloutsos, Shiqiang Yang
KDD, 2014
TrustRank

Given a small seed set of trustworthy pages, can we score how trustworthy all pages are?
TrustRank

Given a small seed set of trustworthy pages, can we score how trustworthy all pages are?

In PageRank random walk, jump to seed trustworthy nodes.
Given a small seed set of trustworthy pages, can we score how trustworthy all pages are?

In PageRank random walk, jump to seed trustworthy nodes.
Given a small seed set of trustworthy pages, can we score how trustworthy all pages are?

In PageRank random walk, jump to seed trustworthy nodes.
TrustRank

Given a small seed set of trustworthy pages, can we score how trustworthy all pages are?

In PageRank random walk, jump to seed trustworthy nodes
TrustRank

Given a small seed set of trustworthy pages, can we score how trustworthy all pages are?

In PageRank random walk, jump to seed trustworthy nodes

Seed Trustworthy vector \( \hat{t} \)

\[
\text{TrustRank} = cA'\text{TrustRank} + \frac{1 - c}{|t|} \hat{t}
\]
Given a small seed set of trustworthy pages, can we score how trustworthy all pages are?

In PageRank random walk, jump to seed trustworthy nodes

Seed Trustworthy vector $\vec{t}$

$$\text{TrustRank} = cA' \text{TrustRank} + \frac{1 - c}{|t|} \vec{t}$$

“Guilt” (Trust) by Association
Combating Web Spam with TrustRank
Zoltán Gyöngyi, Hector Garcia-Molina, Jan Pedersen
VLDB 2004
Distrust Rank

Given a small seed set of spam pages, can we find other spammy pages?
Distrust Rank

Given a small seed set of spam pages, can we find other spammy pages?

Spam pages are pointed to by spam pages

Propagating Trust and Distrust to Demote Web Spam
Baoning Wu, Vinay Goel, Brian D. Davison
WWW 2006
Propagating Trust and Distrust to Demote Web Spam
Baoning Wu, Vinay Goel, Brian D. Davison
WWW 2006

Distrust Rank

Given a small seed set of spam pages, can we find other spammy pages?

Spam pages are pointed to by spam pages

Random walk backward in graph
Distrust Rank

Given a small seed set of spam pages, can we find other spammy pages?

Spam pages are pointed to by spam pages

Random walk backward in graph

Propagating Trust and Distrust to Demote Web Spam
Baoning Wu, Vinay Goel, Brian D. Davison
WWW 2006
Distrust Rank

Random walk backward in graph

Reverse Normalized Adjacency Matrix $M$

$$M_{p,q} \begin{cases} 
1 / \text{InDegree}(q) & \text{if } (p, q) \in E \\
0 & \text{if } (p, q) \notin E
\end{cases}$$
Distrust Rank

Random walk backward in graph

Reverse Normalized Adjacency Matrix $M$

$$M_{p,q} = \begin{cases} 
1 / \text{InDegree}(q) & \text{if } (p, q) \in E \\
0 & \text{if } (p, q) \notin E 
\end{cases}$$

Seed spam pages in vector $\vec{s}$

DistrustRank = $cM$DistrustRank + $\frac{1 - c}{|\vec{s}|} \vec{s}$

Propagating Trust and Distrust to Demote Web Spam
Baoning Wu, Vinay Goel, Brian D. Davison
WWW 2006
SibylRank

Given a small seed set of normal social network accounts, can we find fake accounts?
Given a small seed set of normal social network accounts, can we find fake accounts?

Honest Users
Given a small seed set of normal social network accounts, can we find fake accounts?

Observation: It is difficult for fake users to connect to honest users.

SibylRank
SibylRank

Given a small seed set of normal social network accounts, can we find fake accounts?

Observation: It is difficult for fake users to connect to honest users.

Solution: Use Early-Termination Random Walks

Honest Users
SibylRank

Given a small seed set of normal social network accounts, can we find fake accounts?

Observation: It is difficult for fake users to connect to honest users.

Solution: Use Early-Termination Random Walks

Honest Users
SibylRank

Given a small seed set of normal social network accounts, can we find fake accounts?

Observation: It is difficult for fake users to connect to honest users.

Solution: Use Early-Termination Random Walks

Honest Users
SibylRank

Given a small seed set of normal social network accounts, can we find fake accounts?

Observation: It is difficult for fake users to connect to honest users.

Solution: Use Early-Termination Random Walks

Honest Users

Rarely make it to sybil accounts
## Practitioner’s Guide

<table>
<thead>
<tr>
<th>Method</th>
<th>Graph Type</th>
<th>Node Attributes</th>
<th>Edge Attributes</th>
<th>Seed Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>CatchSync</td>
<td>Directed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TrustRank</td>
<td>Directed</td>
<td></td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>Distrust Rank</td>
<td>Directed</td>
<td></td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>SibylRank</td>
<td>Directed</td>
<td></td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>NetProbe</td>
<td>Directed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FraudEagle</td>
<td>Bipartite</td>
<td></td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>SpEagle</td>
<td>Tripartite</td>
<td></td>
<td></td>
<td>✔️</td>
</tr>
</tbody>
</table>
Fraud in Online Auctions

- Auction sites: Attractive target for fraud
- 63% of complaints to Federal Internet Crime Complaint Center in U.S. in 2006
- Average loss per incident: \( = \$385 \)
- Often non-delivery fraud:

NetProbe: A Fast and Scalable System for Fraud Detection in Online Auction Networks
Shashank Pandit, Duen Horng Chau, Samuel Wang, Christos Faloutsos
WWW 2007
Fraud in Online Auctions

Individual features, e.g. geography, are too easy to fake!

Given a graph of user interactions, what does fraud look like and how can we catch it?
Fraud in Online Auctions

Each user gets a reputation score based on peer feedback:

Score = 70 + 1  
Score = -10 - 1
Fraud in Online Auctions

Each user gets a reputation score based on peer feedback:

Score = 70 + 1  
Score = -10 - 1

Fraudsters need to keep a high reputation score

How do they game the system?
Fraud in Online Auctions

Do they all just give each other positive reviews?

NetProbe: A Fast and Scalable System for Fraud Detection in Online Auction Networks
Shashank Pandit, Duen Horng Chau, Samuel Wang, Christos Faloutsos
WWW 2007
Fraud in Online Auctions

Do they all just give each other positive reviews?

No, because if one is caught they are all revealed.
Fraud in Online Auctions

Do they all just give each other positive reviews?

No, because if one is caught they are all revealed.
Fraud in Online Auctions

Fraudsters form near-bipartite core of 2 roles:
1. **Accomplices:**
   - Trade with honest, looks normal
2. **Fraudsters:**
   - Trade with accomplices
   - Fraud with honest

NetProbe: A Fast and Scalable System for Fraud Detection in Online Auction Networks
Shashank Pandit, Duen Horng Chau, Samuel Wang, Christos Faloutsos
WWW 2007
Fraud in Online Auctions

Use Belief Propagation!

<table>
<thead>
<tr>
<th>Neighbor State</th>
<th>Fraud</th>
<th>Accomplice</th>
<th>Honest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fraud</td>
<td>$\epsilon$</td>
<td>$1 - 2\epsilon$</td>
<td>$\epsilon$</td>
</tr>
<tr>
<td>Accomplice</td>
<td>0.5</td>
<td>$2\epsilon$</td>
<td>$0.5 - 2\epsilon$</td>
</tr>
<tr>
<td>Honest</td>
<td>$\epsilon$</td>
<td>$(1 - \epsilon)/2$</td>
<td>$(1 - \epsilon)/2$</td>
</tr>
</tbody>
</table>

Captures both homophily and heterophily
Fraud in Online Auctions

NetProbe: A Fast and Scalable System for Fraud Detection in Online Auction Networks
Shashank Pandit, Duen Horng Chau, Samuel Wang, Christos Faloutsos
WWW 2007
Fraud in Online Auctions

Initialize prior beliefs of fraudsters to $P(f)=1$

Initialize other nodes as unbiased

NetProbe: A Fast and Scalable System for Fraud Detection in Online Auction Networks
Shashank Pandit, Duen Horng Chau, Samuel Wang, Christos Faloutsos
WWW 2007
NetProbe: A Fast and Scalable System for Fraud Detection in Online Auction Networks
Shashank Pandit, Duen Horng Chau, Samuel Wang, Christos Faloutsos
WWW 2007
Fraud in Online Auctions

1. Initialize prior beliefs of fraudsters to $P(f) = 1$
2. Initialize other nodes as unbiased
3. At each iteration, for each node, compute messages to its neighbors
4. Continue till “convergence”
Fraud in Online Auctions

Initialize prior beliefs of fraudsters to $P(f)=1$

Initialize other nodes as unbiased

At each iteration, for each node, compute messages to its neighbors

Continue till “convergence”

Compute beliefs, use most likely state

NetProbe: A Fast and Scalable System for Fraud Detection in Online Auction Networks
Shashank Pandit, Duen Horng Chau, Samuel Wang, Christos Faloutsos
WWW 2007
Fraud in Online Auctions

Subgraphs found on eBay
(Red nodes are confirmed fraudsters)

NetProbe: A Fast and Scalable System for Fraud Detection in Online Auction Networks
Shashank Pandit, Duen Horng Chau, Samuel Wang, Christos Faloutsos
WWW 2007
Finding fraudulent reviews

Idea: Fraudsters give bad reviews to good products &
good reviews to bad products
Finding fraudulent reviews

Idea: Fraudsters give bad reviews to good products &
good reviews to bad products

Use Belief Propagation

<table>
<thead>
<tr>
<th>s: +</th>
<th>Products</th>
</tr>
</thead>
<tbody>
<tr>
<td>Users</td>
<td>Good</td>
</tr>
<tr>
<td>Honest Fraud</td>
<td>1-ε</td>
</tr>
<tr>
<td>Fraud</td>
<td>2ε</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>s: -</th>
<th>Products</th>
</tr>
</thead>
<tbody>
<tr>
<td>Users</td>
<td>Good</td>
</tr>
<tr>
<td>Honest Fraud</td>
<td>ε</td>
</tr>
<tr>
<td>Fraud</td>
<td>1-2ε</td>
</tr>
</tbody>
</table>
Found replicated bot reviews

Opinion Fraud Detection in Online Reviews by Network Effects
Leman Akoglu, Rishi Chandy, Christos Faloutsos
ICWSM 2013
Found replicated bot reviews

ha ha ha
@Muguiwara78
I want to be that close of J Lo!

Simple IQ test
@Muguiwara78
Good app.

Thanks
@Muguiwara78
I'm extremely satisfied

ha ha ha
@manul
I want to be that close of J Lo!

Simple IQ test
@manul
Good app.

Thanks
@manul
I'm extremely satisfied

Opinion Fraud Detection in Online Reviews by Network Effects
Leman Akoglu, Rishi Chandy, Christos Faloutsos
ICWSM 2013
Finding fraudulent reviews

Figure 8: Reviews of example bot members detected in SWM data for (from top to bottom) the same products (all from the same developer) (see §). Reviews shared among these users are highlighted with a (red) box. Replication of all 5-star reviews provides evidence for fraudulent activity. Also notice the reviews of each fraudster are written mostly on a single day.

Figure 9: Average rating of products under attack by a bot of fraudsters (see §, Fig. 7) drops significantly to $\sim 1$ (lowest possible rating) after removing their fake reviews.

Another way to compare the methods is to study the impact of the fraudsters they detect on the average ratings of products. In particular, we study the average rating of a product when fraudsters are included versus when they are excluded. Figure 10 gives the mean absolute change in average ratings of products after top users are removed by each method — these are the users with fraud scores greater than 0 for sIA and wv-RC, and with honesty scores less than 0 for aHITS. The figure shows that the rating changes are more significant for the high-rating products ([3-5]) for the removed fraudsters by sIA, while changes are more significant for low-rating ([1-2]) products for fraudsters removed by wv-RC. In fact when top fraudsters by sIA are removed, the average rating of high-rated products drop by more than 1 point on average. The removed random users, on the other hand, do not have as significant effects on the ratings.

Figure 10: \textit{FRAUD}\_\textit{EAGLE} top-scorers matter. Absolute change in average ratings of all products, after removing the reviews of top fraudsters detected by each method, along with change when as many random users are removed (averaged over 10 runs).

Computational complexity

Lemma 1 Proposed \textit{FRAUD}\_\textit{EAGLE} is scalable to large data, with computational complexity linear in network size.

Proof 1 In step 1 of \textit{FRAUD}\_\textit{EAGLE}, sIA performs message passing over the edges in a repeated fashion (see Outline 1 Line 12-21), with time complexity $O(|E|d^2t)$, where $|E|$ is the number of edges in the network, $d$ is the maximum domain size of a variable (i.e. number of classes, which is often small), and $t$ is the number of iterations until convergence. In our setting, domain sizes of both users and products is $d = |L_U| = |L_P| = 2$, and $t \sim |E|$ is often small ($t = 37$ on SWM data). Therefore, the time complexity is linear in the number of edges, i.e. network size.
Finding fraudulent reviews

Collective Opinion Spam Detection:
Bridging Review Networks and Metadata
Shebuti Rayana, Leman Akoglu
KDD 2015
Finding fraudulent reviews

Online Review System

Meta Data

Review network

Review Text

Behavioral Data

GREAT place to eat at!!!! The service is good but its a little to LOUD in there as its attached to the mall so I would think it would have been a little less noisy, the wait was way TOO LONG but that’s what you get on a Friday night. The food was a little BLAND!!!!! :

Behavioral Data

Spammer

Fake review

Target product

Collective Opinion Spam Detection:
Bridging Review Networks and Metadata
Shebubi Rayana, Leman Akoglu
KDD 2015

See the talk on
Tuesday @ 2:40PM!
(Level 2 – Room 2)
## Practitioner’s Guide

<table>
<thead>
<tr>
<th>Method</th>
<th>Graph Type</th>
<th>Node Attributes</th>
<th>Edge Attributes</th>
<th>Seed Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>HITS</td>
<td>Directed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PageRank</td>
<td>Directed</td>
<td></td>
<td></td>
<td>Optional</td>
</tr>
<tr>
<td>Label Prop.</td>
<td>Undirected</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>pMRF BP</td>
<td>Undirected</td>
<td></td>
<td></td>
<td>Preferred</td>
</tr>
<tr>
<td>EdgeExplain</td>
<td>Undirected</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Method</td>
<td>Graph Type</td>
<td>Node Attributes</td>
<td>Edge Attributes</td>
<td>Seed Labels</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------</td>
<td>-----------------</td>
<td>-----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>HITS</td>
<td>Directed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PageRank</td>
<td>Directed</td>
<td></td>
<td></td>
<td>Optional</td>
</tr>
<tr>
<td>Label Prop.</td>
<td>Undirected</td>
<td></td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>pMRF BP</td>
<td>Undirected</td>
<td></td>
<td></td>
<td>Preferred</td>
</tr>
<tr>
<td>EdgeExplain</td>
<td>Undirected</td>
<td>✔️</td>
<td></td>
<td>✔️</td>
</tr>
</tbody>
</table>

All of the node attributes are the seed labels for semi-supervised learning in EdgeExplain.
## Practitioner’s Guide

<table>
<thead>
<tr>
<th>Method</th>
<th>Graph Type</th>
<th>Node Attributes</th>
<th>Edge Attributes</th>
<th>Seed Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>CatchSync</td>
<td>Directed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TrustRank</td>
<td>Directed</td>
<td></td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>Distrust Rank</td>
<td>Directed</td>
<td></td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>SibylRank</td>
<td>Directed</td>
<td></td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>NetProbe</td>
<td>Directed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FraudEagle</td>
<td>Bipartite</td>
<td></td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>SpEagle</td>
<td>Tripartite</td>
<td></td>
<td></td>
<td>✔️</td>
</tr>
</tbody>
</table>
Key Points

Random walks ~

singular vectors: find important nodes, and communities

“Guilt-by-association” propagate labels (homophily, heterophily, and more)