Finding the Bias and Prestige of Nodes in Networks based on Trust Scores

Arnab Bhattacharya
arnabb@cse.iitk.ac.in

Department of Computer Science and Engineering,
Indian Institute of Technology, Kanpur,
India

4th-5th October, 2012
Workshop on Social Computing (WoSc 2012)
IIT Kharagpur
About

- Assistant Professor in Computer Science and Engineering department at IIT Kanpur
- Data Mining, Databases, Bioinformatics
About

- Assistant Professor in Computer Science and Engineering department at IIT Kanpur
- Data Mining, Databases, Bioinformatics
- Data Networks (please read as Social Networks)
- Trust and Bias in social platforms
Trust Networks

- A network where entities indicate trust of other entities by rating each other
- *Positive* ratings represent trust, friend, etc.
- *Negative* ratings represent mistrust, foe, etc.
- Example
  - P2P networks
  - Epinions rating reviews
  - Slashdot rating comments
A network where entities indicate trust of other entities by rating each other

Positive ratings represent trust, friend, etc.

Negative ratings represent mistrust, foe, etc.

Example

- P2P networks
- Epinions rating reviews
- Slashdot rating comments

Basic question: How to rank nodes?
Ranking

- Ranking on a graph
  - HITS, Pagerank
- Demands only positive edge weights
Ranking on a graph
  - HITS, Pagerank
Demands only positive edge weights
Ranking on a signed graph
  - Eigentrust, Pagetrust
Solution does not guarantee convergence
Removes negative weights
Neutral Ratings

- Neutral ratings are marked with edge weight 0
- Very different from a no-edge
- Consider node A with only 1 negative in-link
- Consider node B with 1 negative in-link and 100 neutral in-links
- Node B has more prestige than node A
Prestige or Deserve of a node is the “true” rating it deserves
• Prestige or Deserve of a node is the “true” rating it deserves
• True rating is the average of all incoming ratings after removing the bias
Prestige or Deserve of a node is the “true” rating it deserves

True rating is the average of all incoming ratings after removing the bias

Bias is the propensity of a node to trust/mistrust
Prestige or Deserve of a node is the “true” rating it deserves

True rating is the average of all incoming ratings after removing the bias

Bias is the propensity of a node to trust/mistrust

Bias is the average difference of the weight that a node assigns to another to the actual rating of that other node
Prestige or Deserve of a node is the “true” rating it deserves

True rating is the average of all incoming ratings after removing the bias

Bias is the propensity of a node to trust/mistrust

Bias is the average difference of the weight that a node assigns to another to the actual rating of that other node

Thus, deserve is the expected weight of an incoming edge coming from an unbiased node

How to identify and remove bias from nodes?
Removing Bias

- If a node weights another node which has negative rating positively, then it is positively biased.
- Thus, the amount of this positive bias should be removed from the positive edge weights.
- However, if this node weights some other negatively, then nothing should be done to that edge weight.

![Diagram showing node A with edge weights: +1, -1, +1, +1, -1. Resulting graph shows A with edge weights: +1, -1, -1, -1, -1, with 1-b.]
Effect of Bias

- An auxiliary variable to capture the effect of bias

\[ X_{kj} = \begin{cases} 
0 & \text{if } (bias(k) \times w_{kj}) \leq 0 \\
|bias(k)| & \text{otherwise}
\end{cases} \]

- If edge weight and bias are oppositely signed, then no correction is needed
- Otherwise, the amount of bias needs to be corrected

\[ w'_{kj} = w_{kj}(1 - X_{kj}) \]
Bias of a node $i$ is defined in terms of deserve of all its neighbours $j$, i.e., where $ij$ is an edge

$$bias(i) = \frac{1}{2|d^o(i)|} \sum_{j \in d^o(i)} (w_{ij} - deserve(j))$$
Bias of a node $i$ is defined in terms of deserve of all its neighbours $j$, i.e., where $ij$ is an edge

$$bias(i) = \frac{1}{2|d^o(i)|} \sum_{j \in d^o(i)} (w_{ij} - \text{deserve}(j))$$

Deserve of a node $j$ is defined in terms of (corrected) edge weights of all its neighbours $i$, i.e., where $ij$ is an edge

$$\text{deserve}(j) = \frac{1}{|d^i(j)|} \sum_{k \in d^i(j)} (w_{kj}(1 - X_{kj}))$$
Computing Bias and Deserve

- Iterative algorithm
- Start with random values of deserve
- Compute bias using these
- In next iteration, update deserve
- Then update bias again
- So on
Properties of the Algorithm

- Converges to a **unique** solution no matter what the starting values are.
Properties of the Algorithm

- Converges to a unique solution no matter what the starting values are
- Convergence rate is exponential

\[ |b^\infty(i) - b^t(i)| \leq \frac{1}{2^t} \]

where \( b^\infty(i) \) represents the final bias of a node \( i \) and \( b^t(i) \) represents the bias after \( t \) iterations
Properties of the Algorithm

- Converges to a unique solution no matter what the starting values are
- Convergence rate is exponential

\[ \left| b^\infty(i) - b^t(i) \right| \leq \frac{1}{2^t} \]

where \( b^\infty(i) \) represents the final bias of a node \( i \) and \( b^t(i) \) represents the bias after \( t \) iterations

- So, if error tolerance is \( \epsilon \), the number of iterations needed is only logarithmic, i.e., \( \log_2(1/\epsilon) \)
Properties of the Algorithm

- Converges to a unique solution no matter what the starting values are.
- Convergence rate is exponential.

\[ |b^\infty(i) - b^t(i)| \leq \frac{1}{2^t} \]

where \( b^\infty(i) \) represents the final bias of a node \( i \) and \( b^t(i) \) represents the bias after \( t \) iterations.

- So, if error tolerance is \( \epsilon \), the number of iterations needed is only logarithmic, i.e., \( \log_2(1/\epsilon) \).
- Running time needed per iteration is \( O(m) \) for a graph with \( m \) edges.
Future Directions

- Can model *influential nodes* by connecting it to a clique of large number of nodes with high deserve values.
Future Directions

- Can model *influential nodes* by connecting it to a clique of large number of nodes with high deserve values
- Handling *malicious nodes* is not easy
- A node can maintain its bias at 0 even though it is not truthful by rating positive nodes negatively and negative nodes positively
- Higher order difference such as variance may capture this
Future Directions

- Can model *influential nodes* by connecting it to a clique of large number of nodes with high deserve values
- Handling *malicious nodes* is not easy
- A node can maintain its bias at 0 even though it is not truthful by rating positive nodes negatively and negative nodes positively
- Higher order difference such as variance may capture this
- *Collusion attacks* are harder
- Nodes in a group rate each other highly but all other nodes badly
Future Directions

- Can model *influential nodes* by connecting it to a clique of large number of nodes with high deserve values
- Handling *malicious nodes* is not easy
- A node can maintain its bias at 0 even though it is not truthful by rating positive nodes negatively and negative nodes positively
- Higher order difference such as variance may capture this
- *Collusion attacks* are harder
- Nodes in a group rate each other highly but all other nodes badly
- Should model better edge features for *recommendation* purposes
Future Directions

- Can model *influential nodes* by connecting it to a clique of large number of nodes with high deserve values.
- Handling *malicious nodes* is not easy.
- A node can maintain its bias at 0 even though it is not truthful by rating positive nodes negatively and negative nodes positively.
- Higher order difference such as variance may capture this.
- *Collusion attacks* are harder.
- Nodes in a group rate each other highly but all other nodes badly.
- Should model better edge features for *recommendation* purposes.

THANK YOU!