

Predicting Trust and Distrust in Social Networks

Thomas DuBois, Jennifer Golbeck, and Aravind Srinivasan
Presented by: Reese Moore

April 17, 2014

Overview

- ▶ Overview
- ▶ Introduction
- ▶ Proposed Algorithm
 - ▶ Path Probabilities
 - ▶ Modified Spring Embedding
- ▶ Testing Methodology
- ▶ Results
- ▶ Conclusions
- ▶ Summary

Introduction

The Internet is growing, and the problem of who to trust is increasingly important as more content is user generated.

Social networking on the Internet allows users to mark who they trust and distrust.

- ▶ Trust is transitive
- ▶ Distrust is *not* transitive.

Attempt to predict *both* trust and distrust in a social network

Proposed Algorithm

The algorithm proposed makes use of two independent processes

- ▶ Path Probabilities
- ▶ Modified Spring Embedding

The edge between two nodes in the social graph represents a two dimensional vector whose position indicates the amount of trust between its endpoints

Proposed Algorithm

Path Probabilities

For each pair of users (u, v) , an edge is placed between them with some probability that depends on the direct trust value between them $t_{u,v}$.

The trust between two people is inferred from the probability that they are connected in the resulting graph.

Formally,

- ▶ Choose a reversible mapping f from trust value to probabilities
- ▶ Construct a random graph G where edge (u, v) exists independently with probability $f(t_{u,v})$
- ▶ This graph gives inferred trust values $T_{u,v}$ where $f(T_{u,v})$ is the probability that a path from u to v exists

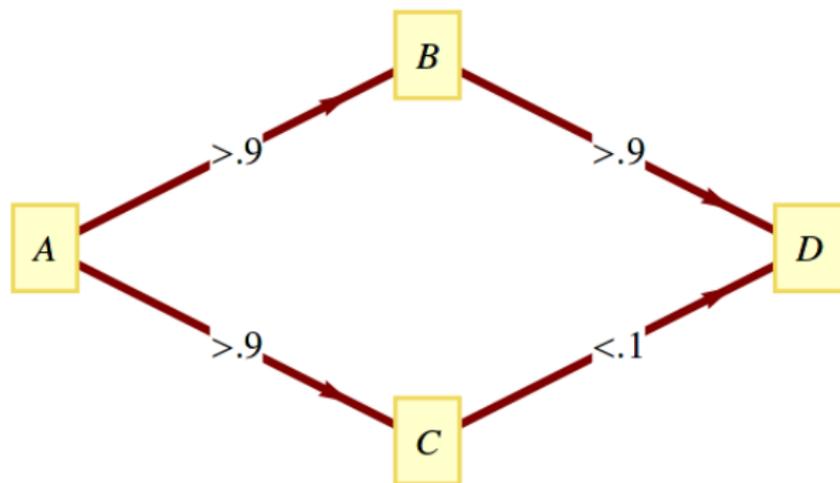
Proposed Algorithm

Path Probabilities

Path Probabilities works well for trust, but not distrust.

- ▶ Positive trust corresponds to edge probabilities
- ▶ Negative trust corresponds to the upper bound on path probabilities

Because paths are additive, this does not scale



Proposed Algorithm

Modified Spring Embedding

Spring embedding simulates the physics of springs

- ▶ Edges are treated as springs that pull nodes together
- ▶ Nodes repel one another
- ▶ Nodes are randomly laid out and simulated until
 - ▶ The system reaches a stable equilibrium
 - ▶ Some other condition is met

Spring embedding is modified to be used for trust inference

- ▶ The repelling force is only added between nodes connected by a negative edge
- ▶ Distance between nodes indicates trust

Testing Methodology

Datasets

Three datasets were used from the Stanford Large Network Dataset Collection¹

- ▶ Wikipedia moderator elections
- ▶ Slashdot user Friend or Foe
- ▶ Epinions

All of these datasets are biased towards positive trust

¹<http://snap.stanford.edu/data/>

Testing Methodology

For all of the datasets, some points are randomly selected and removed

- ▶ 500 in Wikipedia and Slashdot
- ▶ 1000 in Epinions

The remaining nodes become the training set

The removed nodes become the testing set

Testing Methodology

Tuning

System parameters were tuned using the training set

For Path Probabilities

- ▶ The probability corresponding to a positive edge $p = 0.05$

For Spring Embedding

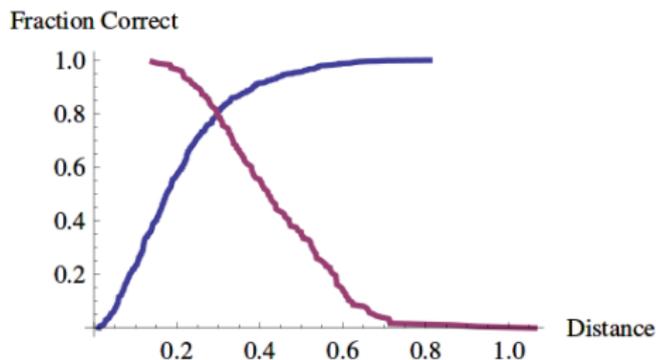
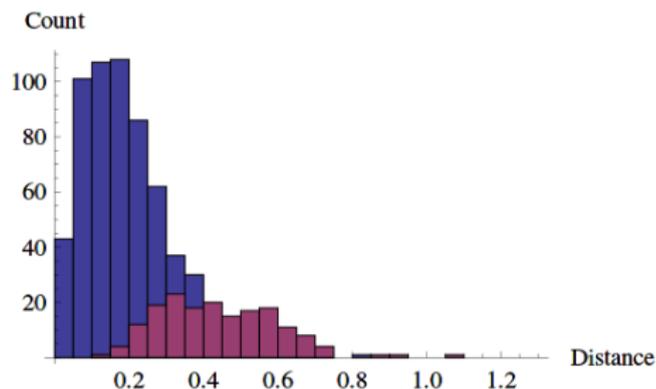
- ▶ An attractive force of d^2 for nodes at distance d
- ▶ A repelling force of $\frac{1}{d^2}$
- ▶ A 4-dimensional unit cube space

Testing Methodology

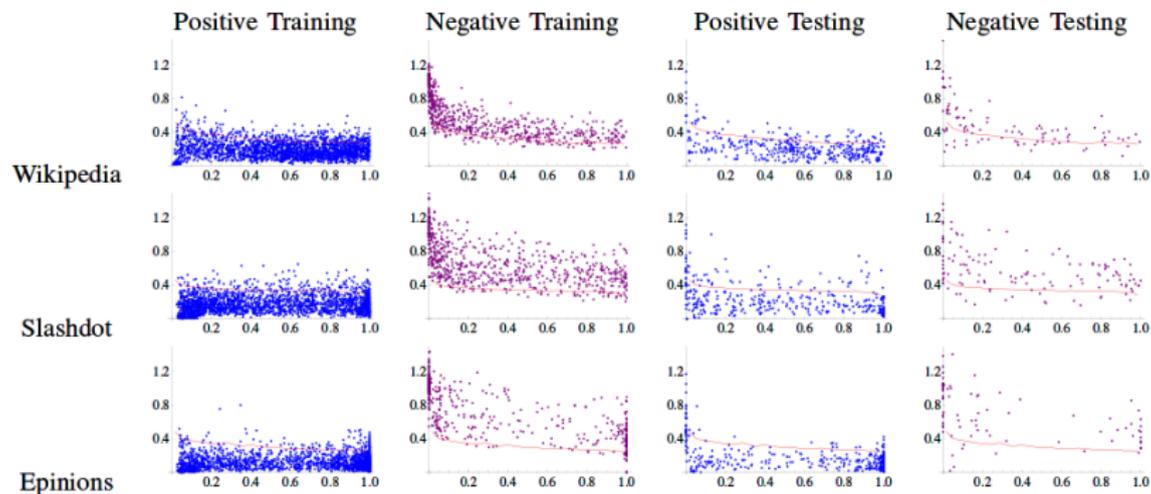
Training

Training data bucketed by path probability

For each interval, find embedded distance which minimizes the maximum ratio of mislabeled positive/negative edges



Results



For each run, a separator classifies positive and negative trust relationships

Results

| | Wikipedia | Slashdot | Epinions |
|-------------------------------------|-----------|----------|----------|
| Total Positive edges | 0.78 | 0.77 | 0.85 |
| Total Negative edges | 0.22 | 0.23 | 0.15 |
| Training edges correctly classified | 0.86 | 0.92 | 0.94 |
| Positive test edges correct | 0.81 | 0.81 | 0.89 |
| Negative test edges correct | 0.78 | 0.84 | 0.89 |
| Correct positive classifications | 0.93 | 0.94 | 0.98 |
| Correct negative classifications | 0.51 | 0.60 | 0.61 |
| Overall edges correctly classified | 0.81 | 0.82 | 0.89 |
| E_{10} edges correctly classified | 0.81 | 0.96 | 0.94 |
| E_{25} edges correctly classified | 0.81 | 0.96 | 0.95 |

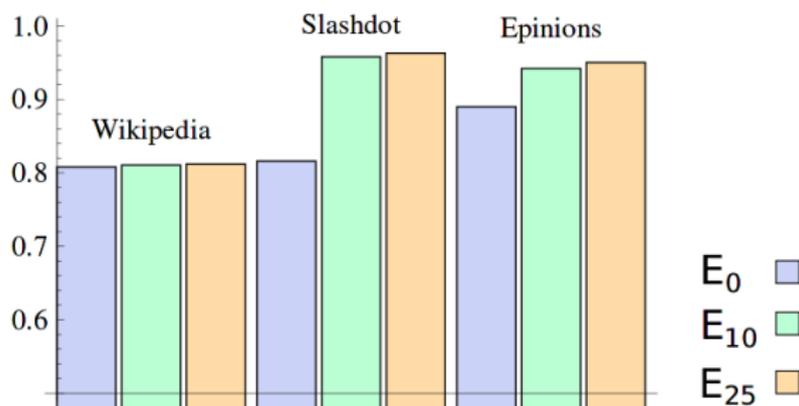
The fraction of correct classification for various criteria

Results

Embedded edges

Definition

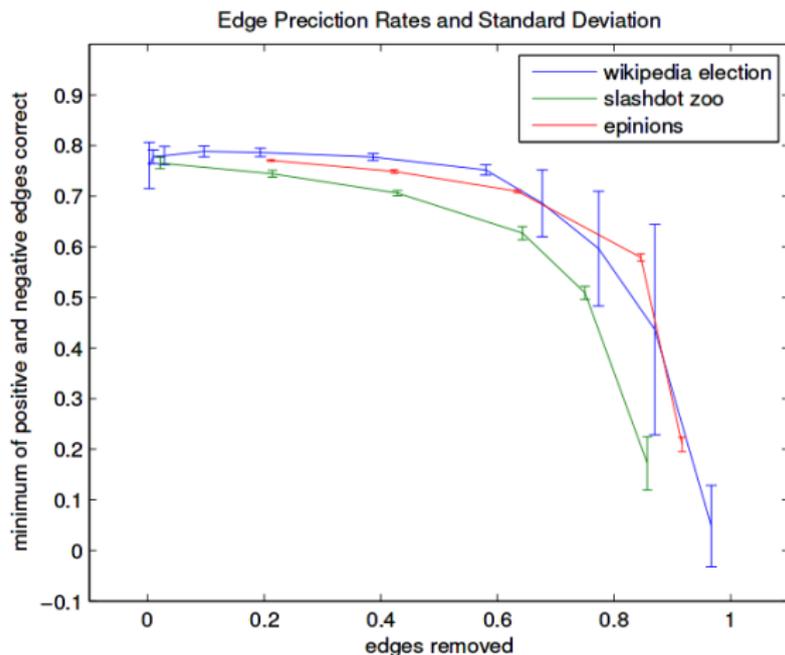
Embedded edges – Those sets $E_n \subseteq E$ of all edges which are a part of at least n undirected triangles



Overall accuracy for all edges, as well as E_{10} and E_{25}

Removed Edges

Opposite edges were merged into a single unidirectional edge, and edges were removed uniformly at random.



Accuracy rates as a function of edges removed

Conclusions

- ▶ The classifier is highly accurate (80% – 90%)
- ▶ Results show good self-consistency
- ▶ This algorithm is potentially useful in many applications
 - ▶ Sorting (Emails, Product Reviews, etc.)
 - ▶ Filtering (Online Discussions)
 - ▶ Aggregation

- ▶ Social networks are highly redundant
- ▶ Distrust is difficult to quantify as a trust value

Summary

This work attempts to infer *both* positive and negative trust in a social network.

This work presented a new algorithm for trust inference

- ▶ Path Probability model of the network
- ▶ Novel application of spring embedding by applying it to trust in social networks

Testing on real world data shows that

- ▶ The algorithm is successful as a classifier
- ▶ Social networks tend to have a very redundant structure

Questions?