Trust Prediction

...

Overview

- In social networks or online platforms, trust is the **expectation** of an entity (e.g. user) on another entity.
- Generally, the expectation is limited to a certain **domain**.
- For example : In domain of health, a person A will trust doctor more than his/her parents.

Why Trust Prediction is hard problem ?

1. Sparsity
2. The process of formation of trust is not entirely dependent on online activity.
3. Lack of ground truth data. Generally binary data is available.

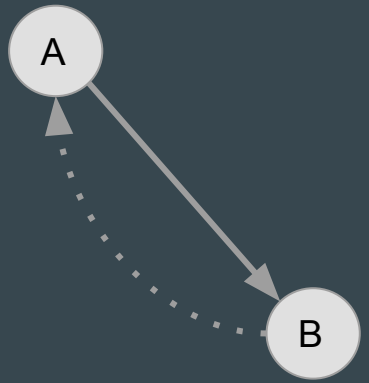
Trust Propagation

Multiple factors can influence trust propagation through social networks :

- Reciprocity
- Transitivity
- Weak Dependency and Homophily
- Social Dimensions and Heterogeneity

Reciprocity

When A places trust in B he/she expects B to be committed. The very act of trusting engenders a weak sense of expectation from B to A.



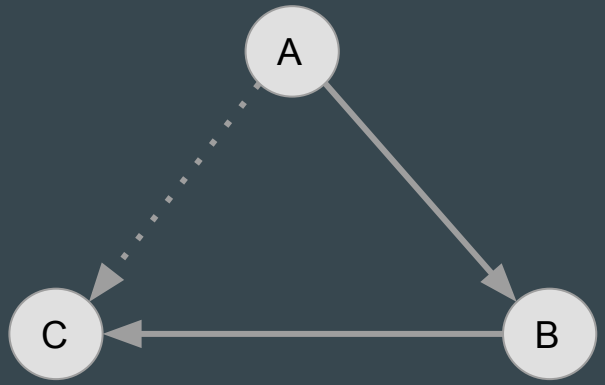
Transitivity

Consider a situation :

A trusts B

B trusts C

Existence of indirect relationship between A and C increases the probability of formation of direct trust relationship from A to C.

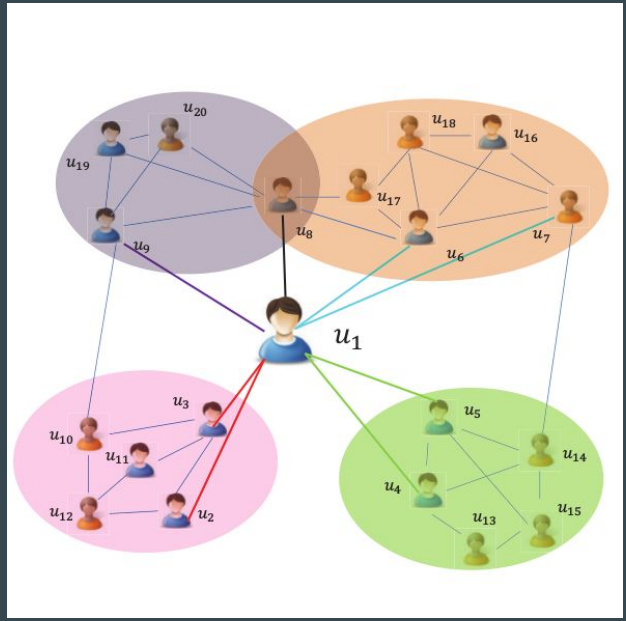


Homophily

Attributes of nodes or users also play a significant role in inducing trust relationships. Nodes with similar attributes have higher probability of formation of trust relationships.

Social Dimensions

Social dimensions or communities' trend impact the decision making process at individual level. However, an user belongs to multiple communities. Hence, trust relations are generally a composition of multiple relations.



Related Works :

1. hTrust : models the homophily effect in trust prediction
2. Recommendation with Social Dimensions : models the heterogeneity phenomenon for product recommendation
3. DeepWalk, Node2Vec : embedding based approaches for link prediction

Possible Extensions : Node2vec

1. Use node2vec to learn representations of nodes and use them to calculate homophily coefficients.
2. Extend node2vec for learning node representation:
    a. Extend the sampling function to include notion of homophily, weak dependency and heterogeneity
    b. Also extend the objective function in order to fit with new sampling process. Use a completely probabilistic model instead of MF models. Derive an objective function to model trust relationships along with novel sampling algorithms. Objective function will iterate over pair of nodes and evaluate loss.

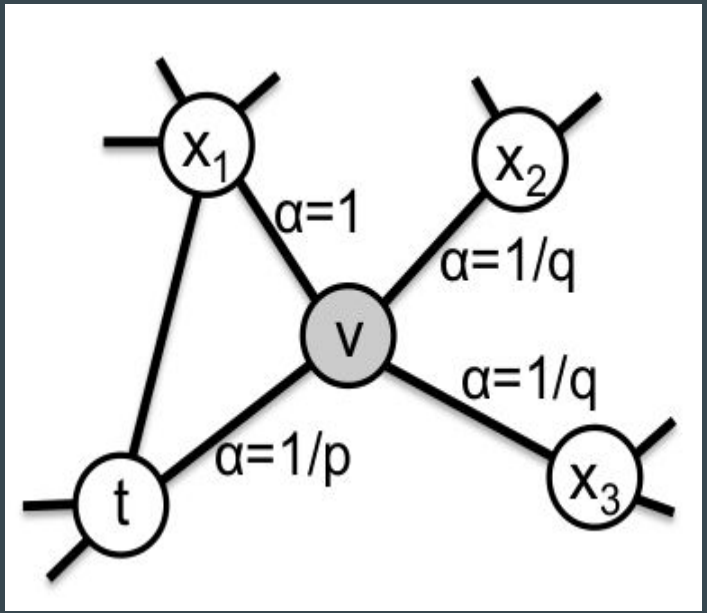
Extension 1

Important features of node2vec :

1. Uses flexible random walks to sample sequence of nodes
2. Sequence of nodes is treated as sequence of words
3. Uses skip-gram model's objective function to learn representations.
4. Sampling strategies discussed in paper :
    a. Breadth First Sampling
    b. Depth First Sampling
    c. Flexible Random Walk

Extension 1 ...

1. BFS : Neighbourhood restricted to immediate neighbours. Models homophily hypothesis.
2. DFS : Neighbourhood consists of nodes sequentially sampled at increasing distances. Models structural equivalence hypothesis.
3. Flexible random walk



Extension 2

Disadvantages of current sampling method and objective function:

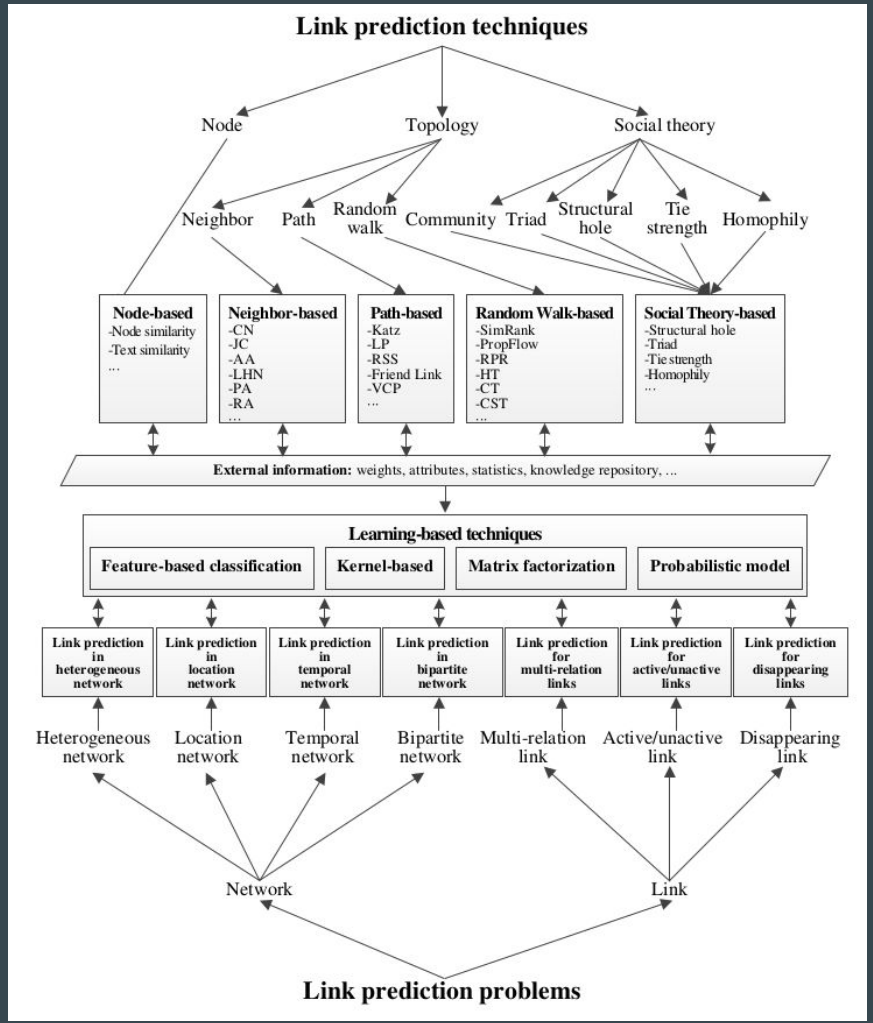
1. Nodes at 1-hop neighbourhood from the current node are given equal preference. Finer distinction between sampling probabilities of nodes can help to improve the sampling process.
2. Homophily is neither modeled in the current objective function nor sampling algorithm.
3. The objective function of node2vec is designed to learn generic representations of nodes. Instead for trust prediction there is a need to learn the trust values of edges as function of two nodes.
4. Node2vec's link prediction is symmetric. It doesn't distinguish between trustor and trustee's representations.

Trust prediction as link prediction problem

Related works :

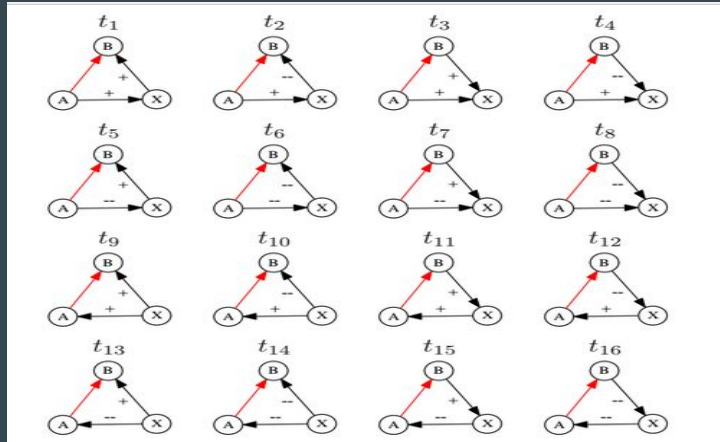
1. Leskovec, Jure, Daniel Huttenlocher, and Jon Kleinberg. "Signed networks in social media." *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 2010.
2. Leskovec, Jure, Daniel Huttenlocher, and Jon Kleinberg. "Predicting positive and negative links in online social networks." *Proceedings of the 19th international conference on World wide web*. ACM, 2010.
3. Backstrom, Lars, and Jure Leskovec. "Supervised random walks: predicting and recommending links in social networks." *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 2011.
4. Wang, Suhang, et al. "Signed network embedding in social media." *Proceedings of the 2017 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, 2017.
5. Yuan, Shuhan, Xintao Wu, and Yang Xiang. "SNE: Signed Network Embedding." *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, Cham, 2017.

Wang, Peng, et al. "Link prediction in social networks: the state-of-the-art." *Science China Information Sciences* 58.1 (2015): 1-38.



Leskovec, Jure, Daniel Huttenlocher, and Jon Kleinberg. "Signed networks in social media." *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 2010.

1. Compares social balance theory and social status theory
2. Calculates overrepresented and underrepresented triads.
3. Nodes with more common neighbours tend to have positive links.



t_i	count	$P(+)$	s_g	s_r	B_g	B_r	S_g	S_r
t_1	178,051	0.97	95.9	197.8	✓	✓	✓	✓
t_2	45,797	0.54	-151.3	-229.9	✓	✓	✓	○
t_3	246,371	0.94	89.9	195.9	✓	✓	○	✓
t_4	25,384	0.89	1.8	44.9	○	○	✓	✓
t_5	45,925	0.30	18.1	-333.7	○	✓	✓	✓
t_6	11,215	0.23	-15.5	-193.6	○	○	✓	✓
t_7	36,184	0.14	-53.1	-357.3	✓	✓	✓	✓
t_8	61,519	0.63	124.1	-225.6	✓	○	✓	✓
t_9	338,238	0.82	207.0	-239.5	✓	○	✓	✓
t_{10}	27,089	0.20	-110.7	-449.6	✓	✓	✓	✓
t_{11}	35,093	0.53	-7.4	-260.1	○	○	✓	✓
t_{12}	20,933	0.71	17.2	-113.4	○	✓	✓	✓
t_{13}	14,305	0.79	23.5	24.0	○	○	✓	✓
t_{14}	30,235	0.69	-12.8	-53.6	○	○	✓	○
t_{15}	17,189	0.76	6.4	24.0	○	○	○	✓
t_{16}	4,133	0.77	11.9	-2.6	✓	○	✓	○
	Number of correct predictions				8	7	14	13

Figure 2. Top: All contexts $(A, B; X)$. Red edge is the edge that closes the triad. Bottom: Surprise values and predictions based on the competing theories of structural balance and status. t_i refers to triad contexts above; *Count*: number of contexts t_i; $P(+)$: prob. that closing red edge is positive; s_g: surprise of edge initiator giving a positive edge; s_r: surprise of edge destination receiving a positive edge; B_g: consistency of balance with generative surprise; B_r: consistency of balance with receptive surprise; S_g: consistency of status with generative surprise; S_r: consistency of status with receptive surprise.

Wang, Suhang, et al. "Signed network embedding in social media." *Proceedings of the 2017 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, 2017.

1. Models social balance theory
2. Applicable for signed undirected networks
3. Learns f(x,y) using neural network
4. Logistic regression classifier is trained on training dataset and used for prediction on test dataset.

$$\min_{\mathbf{X},\mathbf{x}_0,\theta} \frac{1}{C}\Big[\sum_{(\mathbf{x}_i,\mathbf{x}_j,\mathbf{x}_k)\in\mathcal{P}} \max(0, f(\mathbf{x}_i,\mathbf{x}_k)+\delta-f(\mathbf{x}_i,\mathbf{x}_j)) + \sum_{(\mathbf{x}_i,\mathbf{x}_j,\mathbf{x}_0)\in\mathcal{P}_0} \max(0, f(\mathbf{x}_i,\mathbf{x}_0)+\delta_0-f(\mathbf{x}_i,\mathbf{x}_j))\Big] + \alpha\big(\Re(\theta)+\|\mathbf{X}\|_F^2+\|\mathbf{x}_0\|_2^2\big),$$

where $C = |\mathcal{P}| + |\mathcal{P}_0|$ is the size of the training data and $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$ is the low-dimensional representation of the m nodes, and θ is a set of parameters to define the similarity function f. $\Re(\theta)$ is the regularizer to avoid overfitting and α is a parameter to control the contribution of the regularizers.

Experiment

$$\underset{\theta}{\text{minimize}} \sum_{(p^+,p^-)\sim D} [max(0, \lambda - g(p^+) + g(p^-)) + (\alpha_1 H(p^+) + \alpha_2 I(p^+))||g(p^+) - 1)||] + \alpha_3||\theta||$$

where

p^+ is sampled from node2vec sampling algorithm

p^- is sampled randomly

$H(p^+)$ is item based homophily coefficient or cosine similarity

$I(p^+)$ is category based homphily or Jaccard similarity

Learning g(p+) is important part. Next slide discusses preliminary results using htrust as baseline and g(p+) as Hadamard operator W.(f(u)*f(v)). Two sets of embeddings are learned for each node.

Performance Metric

For measuring performance F1-score or accuracy is not used as prediction task is highly imbalanced.

Consider a list E consisting of positive pairs of links, chronologically sorted based on time of formation of links.
Assume A as the training set of relations, N as the test set of relations and B as the set of negative pairs or pairs without any link.
Let S be the noisy pairs obtained from *x%* of B.

PS = N U S

Using link prediction algorithm the likelihood of each pair in PS is calculated and sorted accordingly. Set D is obtained by selecting first |P| pairs.

Performance = $|D \cap N|/|N|$

Similar to Precision@k

HTrust Performance:

hTrust			
lambda = 10	alpha = 0.01	beta = 0.01	
U [0.0, 0.1]	V [0.0, 0.9]	factor=6.0	
train_p	test_p	n_iter	accuracy
0.7	0.3	60	31.23
0.7	0.3	80	33.45
0.7	0.3	100	33.71
0.65	0.3	60	21.09
0.65	0.3	80	22.23
0.65	0.3	100	22.58
0.6	0.3	60	19.52
0.6	0.3	80	16.82
0.6	0.3	100	19.52

Exp1 Performance:

Train percentage: 70% and test percentage: 30%
alpha1: 0.1
alpha2: 0.1
alpha3: 0.1

Average performance: 44.94%

From other experiments it is found that:

Using only l2-regularization without any homophily term yields an average performance of 44.44% which indicates that homophily terms in current formulation are not very helpful.

Without any regularization term average performance is 39.33% which seems to indicate that node2vec sampling algorithm and/or negative sampling is very helpful in link prediction task.

Using leaky relu units on hadamard operator drops the average performance by around 4%.