ON THE TROLL-TRUST MODEL FOR EDGE SIGN PREDICTION IN SOCIAL NETWORKS

Géraud Le Falher(1), Nicolò Cesa-Bianchi(2), Claudio Gentile(3) and Fabio Vitale(1,4)

(1) Inria, Univ. Lille, CNRS UMR 9189 – CRISAL, France
(2) Università degli Studi di Milano, Italy
(3) University of Insubria, Italy
(4) Department of Computer Science, Aalto University, Finland

Problem setup and notations

Given a directed graph of binary interactions between users, we want a scalable method to predict the sign of an interaction, using only the topology and the sign of some edges in Eκ.

\[ s(i,j) = \{ +1, -1 \} \]

The triollness \( t(i) \) of \( i \) is its fraction of negative outgoing edges, its distrustworthiness \( u(i) \) is its fraction of negative incoming edges.

The Bayes optimal prediction for \( y \) holds with probability at least \( 1 - \delta \).

\[ \Pr(s(i,j) = y(i,j)) \leq \frac{e^{-\delta}}{2} \]

Batch algorithm: L. PROP.

Unfortunately, a good sampling requires a rather dense graph which is not usually the case in social media. In order to address this issue, first we transform the problem from edge to node classification.

Active algorithm: \( \text{BLC(tr, un)} \)

We use the complementary to 1 of triollness and distrustworthiness (estimated on the training set \( E_\text{tr} \)) as proxy for \( p_i \) and \( p_j \), and predict

\[ g(i,j) = \text{sgn}(\{ E(i,j) = +1 \} - \frac{1}{2}) \]

Conclusions

1. A simple generative model, justifying existing heuristics and providing a new principled predictor (\( \text{BLC(tr, un)} \)).
2. A maximum likelihood approximation by a label propagation algorithm (\( \text{L. Prop.} \)), leveraging a reduction from edge to node classification.
3. Extensive comparative experiments on real data against state of the art methods.

Generative model

Each node \( i \) has 2 parameters, drawn from an arbitrary distribution \( p_i \) over pairs in \([0,1]^2\):

- its tendency \( p_i \) to send positive edges (i.e. niceness) and
- its tendency \( q_i \) to receive positive edges (i.e. pleasantness)

\[ P(y_{i,j} = +1) = \frac{1}{2}(p_i + q_j) \]

The Bayes optimal prediction for \( y_{i,j} \) is thus

\[ g^*(i,j) = \text{sgn}(P(y_{i,j} = +1) - \frac{1}{2}) \]

We approximate \( g^*(i,j) \) by resorting to a maximum likelihood estimator of the parameters

\[ (p_i, q_j) \approx \arg \max_p \sum_{i,j \in E} \log P(y_{i,j} = +1) \]

Because the gradients of the log-likelihood function are not linear

\[ \frac{\partial \log P(y_{i,j})}{\partial p_i} = \frac{2}{y_{i,j} + \sum_k p_{i,k}} \]

we approximate them, which is equivalent to setting to zero the gradient w.r.t. \( p_i \) for \( y_{i,j} = -1 \), and for \( y_{i,j} = +1 \),

\[ f_E(p, q) = \sum_{i,j \in E} (1 - y_{i,j} \cdot \log (p_i + q_j)) \]

We then use the complementary to 1 of triollness and distrustworthiness (estimated on the training set \( E_\text{tr} \)) as proxy for \( p_i \) and \( p_j \), and predict

\[ g(i,j) = \text{sgn}(\{ E(i,j) = +1 \} - \frac{1}{2}) \]

Thus we need to subtract

\[ \frac{1}{2}(p_i + q_j) \]

as \( p_i \) and \( p_j \) concentrate around their mean \( u_i \) and \( u_j \).

We sample \( u_i \) outgoing and incoming edges for each node, estimate empirically \( g^*(i,j) \) and \( 1 - g^*(i,j) \) as the overall fraction of positive edges then finally predict remaining edges according to (1).

Setting \( Q = \frac{1}{2} \text{sgn}(\{ E(i,j) \}) \) we query \( \delta(\{ E(i,j) \}) \) edges.

Online setting

We also study an online setting where the signs are adversarial rather than generated by our model.

Letting \( y \) be the vector of all labels, \( \Psi_i = \{ y_{i,j} \} \) is the number of least used label outgoing from \( i \), and \( \Psi_i = \sum_j \Psi_i(\Psi_j) \). Likewise for incoming edges, \( \Psi_i = \sum_j \Psi_i(\Psi_j) \). Finally the regularity of a labeling \( y \) is \( \Psi_y(Y) = \sum_i \Psi_i(Y) \).

We devise an algorithm consisting of a combination of Randomized Weighted Majority instances built on top of each other that on average makes \( \Psi_y(Y) \) mistakes.

On the lower side, for any directed graph \( G \) and any integer \( k \), there exists a labeling \( Y \) forcing at least \( \frac{k}{2} \) mistakes to any online algorithms, while \( \Psi_y(Y) = k \).

References


Discussion

We presented two batch algorithms derived from our generative model. One is local (\( \text{BLC(tr, un)} \)) and therefore extremely scalable while being performant both in theory and in practice. The other (\( \text{L. Prop.} \)) propagates sign information along the graph which provides more accurate results while remaining faster than previous approaches.

Later, we would like to extend our results to weighted graphs and incorporate side information.