# Learning and Inference in Massive Social Networks

## [Extended Abstract] *

Shawndra Hill
University of Pennsylvania
3730 Walnut Street
Philadelphia, PA 19104
shawndra@wharton.upenn.edu

Foster Provost
New York University
44 W 4th Street
New York, NY 10012
fprovost@stern.nyu.edu

Chris Volinsky
AT&T Labs Research
180 Park Avenue
Florham Park, NJ 07932
volinsky@research.att.com

## Keywords

Social networks, collective inference, viral marketing

## 1. INTRODUCTION

Researchers and practitioners increasingly are gaining access to data on explicit social networks. For example, telecommunications and technology firms record data on consumer networks (via phone calls, emails, voice-over-IP, instant messaging), and social-network portal sites such as MySpace, Friendster and Facebook record consumer-generated data on social networks. Inference for fraud detection [5, 3, 8], marketing [9], and other tasks can be improved with learned models that take social networks into account and with collective inference [12], which allows inferences about nodes in the network to affect each other. However, these social-network graphs can be huge, comprising millions to billions of nodes and one or two orders of magnitude more links.

This paper studies the application of collective inference to improve prediction over a massive graph. Faced initially with a social network comprising hundreds of millions of nodes and a few billion edges, our goal is: *to produce an approximate consumer network that is orders of magnitude smaller, but still facilitates improved performance via collective inference.* We introduce a sampling technique designed to reduce the size of the network by many orders of magnitude, but to keep linkages that facilitate improved prediction via collective inference.

In short, the sampling scheme operates as follows: (1) choose a set of nodes of interest; (2) then, in analogy to snowball sampling [14], grow local graphs around these nodes, adding their social networks, their neighbors' social networks, and so on; (3) next, prune these local graphs of edges which are expected to contribute little to the collective inference; (4) finally, connect the local graphs together to form a graph with (hopefully) useful inference connectivity.

We apply this sampling method to assess whether collective inference can improve learned targeted-marketing models for a social network of consumers of telecommunication services. Prior work [9] has shown improvement to the learning of targeting models by including social-neighborhood information—in particular, information on existing customers in the immediate social network of a potential target. However, the improvement was restricted to the "network neighbors", those targets linked to a prior customer thought to be good candidates for the new service. Collective inference techniques may extend the predictive influence of existing customers beyond their immediate neighborhoods. For the present work, our motivating conjecture has been that this influence can improve prediction for consumers who are not strongly connected to existing customers. Our results show that this is indeed the case: collective inference on the approximate network enables significantly improved predictive performance for non-network-neighbor consumers, and for consumers who have few links to existing customers.

In the rest of this extended abstract we motivate our approach, describe our sampling method, present results on applying our approach to a large real-world target marketing campaign in the telecommunications industry, and finally discuss our findings.

## 2. COLLECTIVE INFERENCE IN MASSIVE NETWORKS

Relational collective inference is simultaneous statistical inference about the values of variables for a set of interconnected data items (nodes in the network). In particular, we are interested in collective inference where the value of the same variable is estimated for linked nodes [10]. Various techniques have been used for univariate collective inference on networked data [12], for example, Gibbs sampling [7], relaxation labeling [11], loopy belief propagation [13], graph-cut methods [1], and iterative classification [2]. To our knowledge, collective inference has not been applied previously to a massive social network. Why not?

Let us consider a univariate consumer network, represented by the graph $G$, where the nodes $N(G)$ are consumers and the edges $E(G)$ are links between consumers. This is similar to the consumer network modeled by Domingos and Richardson [4], where customers are linked if they rate the same movie. We link customers by social ties—in our case, by our knowledge that the two consumers have communicated. We weight the network by aggregating amount of communications between consumers over some time period, i.e., the aggregation defines a *weight* on each edge, $\mathrm{w}_G(e)$ for each edge each edge $e \in E(G)$. For our results below, the weights are based on the sums of durations of communications between nodes.

Usually, collective inference methods are used with a model that is applied to each node $n \in N(G)$, and that for each $n$ accesses some information about each node linked to $n$. Thus, the computational complexity of the methods is at least $O(E(G))$. In principle, this is quadratic in the number of nodes, but in practice the average degree of a massive social network will be much less than $\|N(G)\|$. Unfortunately, even for node degrees averaging in the hundreds, massive consumer networks will have $10^9$ edges at least, and for many modelers, operating on this graph in main memory will be infeasible. In addition, there is a cost ($O(N(G))$) of repeatedly applying a learned model. What's more, collective inference methods generally cycle through the set of nodes dozens, hundreds, or thousands of times.

# 3. BUILDING AN APPROXIMATE CONSUMER NETWORK

Unfortunately, we cannot solve the problem with a simple random sampling of the network, because random sampling of a small set of either nodes or edges will destroy the very connectivity structure that we would like to take advantage. On the other hand, network-based sampling methods, such as snowball sampling, also can introduce bias [14]; however, for prediction we often are willing to tolerate or even take advantage of bias [6].

We build an approximate consumer network based on three main assumptions (which will become more precise when we describe the procedure). First, we can select a set of potential "targets" of interest based on other characteristics (for example, a traditional targeted marketing model or even a network-based marketing model such as that of Hill et al. [9]; similar techniques would apply for other applications, such as fraud detection [5]). Second, existing-customer influence does not propagate arbitrarily far through the network, but some long-distance propagation is helpful. Third, stronger social ties will be more important for propagating influence. Thus, the approximate network is constructed by the following steps.

***Step 1.*** **Selecting seed nodes:** Select a sample $T_0$ of size $t$ from all targets $T$. These are the *targets of interest.* For our results the sample is a random sample, but it also could be (for example) the top $k$ targets on a list ranked by a traditional model.

***Step 2.*** **Build local subgraphs:** Add all edges between targets of interest and other consumers, and if necessary add the corresponding consumers' nodes, for consumers that have explicitly communicated with nodes belonging to $T_0$.

**Definition 3.1: First-order graph $G_1$, the local neighborhood.** To construct $G_1$, let $t$ be the number of targets of interest in $N(G)$. For each node $n = 1, ..., t$, let $k_n$ be the number of neighbors for node $n$. We will create a graph $G_{1i}$ as follows. For $i = 0, ..., k_n$, add node $i$ to $N(G_{1i})$ and edge $e_{ni}$ weighted by $w_{ni}(e_{ni})$ to $E(G_1)$, where $e_{ni}$ connects the seed node $n$ to the neighboring node $i$. We call $G_{1i}$ the *local neighborhood of* $n$.

Moving beyond the $G_1$ local neighborhood for each of the sample targets, we add $G_1$ local neighborhood of each neighbor belonging to the target of interest's $G_1$ local neighborhood. We call this the second-order graph, $G_2$. The graph used for analysis is the union of the $G_2$ subgraphs for all sample targets.

**Definition 3.2: Second-order graph $G_2$.** To construct $G_2$, let $t$ be the number of targets of interest in $N(G)$. For each node $n = 1, ..., t$, let $k_n$ be the number of neighbors for node $n$. We will create a graph $G_{2i}$ as follows. For $i = 1, ..., k_n$, add all nodes in $N(G_{1i})$ and all edges in $E(G_{1i})$ to $G_{2i}$. In other words, for each node $n$ take the union of all $G_1$ subgraphs of the neighbors to which they are directly connected.

***Step 3.*** **Top-$k$ pruning:** Remove less-informative nodes and edges. For a given consumer $n$, local neighbors $i$ are ranked by their connecting edge weight $w_{ni}(e_{ni})$ and the top $k$ are kept.

We determine $k$ via (nested) cross-validation. A sensitivity analysis shows that the number of nodes $k$ kept in the network is a significant factor for predictive performance.

## 3.1 Estimating adoption probability

Once the subgraph is constructed, we can apply collective inference to the graph. Using as our inference model a univariate Gaussian random field over the estimated probabilities of adoption, with existing customers' probabilities fixed at unity and the rest at small constants, we apply relaxation labeling for collective inference. This procedure is described in detail by Macskassy and Provost as the weighted-vote relational neighbor (wvRN) classifier [12]. The resulting estimated probabilities of adoption can then be included in a multivariate model for predicting the conditional likelihood of adoption. We use a multiple logistic regression based on other variables, such as other network attributes or traditional demographic and prior relationship variables [9].

# 4. RESULTS

Detailed results are available in the full paper. In this abstract we will show that indeed collective inference can be applied effectively to a massive consumer network.

## 4.1 Data

The network approximation procedure was applied to a consumer network of approximately hundreds of millions of nodes and a few billion edges. Estimating that the effort would yield about 1 million unique nodes when generating the approximate network, we chose 1800 seed nodes, via a random, stratified sample of 900 network-neighbor and 900 non-network-neighbor targets with a 50/50 split on the class label, viz., whether or not the consumer will adopt in a future time period. These will be the training/testing nodes for the results below. Building the second-order subgraphs results in a graph with approximately 1 million nodes and 4.5 million edges—a thousand-fold reduction in the size of the network. (We consider further pruning below).

## 4.2 A collective inference oracle

Before presenting the main results—applying collective inference to the approximate network—let's discuss a point of comparison for a collective inference procedure: how well could we do if an oracle were to tell us the "truth" about the future adoption of all the other nodes in the network. (For our model, the only thing that matters is the truth about whether or not a node's neighbors will adopt.) From these data we can construct an oracle-based "leave-one-out" method, as follows.

One at a time, each target (testing) node was removed from the network; for its neighbors the oracle tells us whether or not they in fact will adopt. We then apply the univariate estimation model (wvRN) to estimate the probability of adoption.[1] We now can assess: if the collective inference could perfectly predict future adoption for neighbors, (how much) would that add value to the prediction of adoption for the target node?

When added to the multivariate logistic regression model using a large set of network-based attributes, this oracle-based score was a statistically significant predictor. When using forward selection over these attributes, the oracle-based score was selected as part of the best model. Predictive performances are reported in the next section; the reader should keep in mind that these particular "leave-one-out" results are based on this unrealizable oracle procedure, and therefore are useful only for comparison to the actual collective inference results.

## 4.3 Modeling with collective inference on the approximate network

Table 1 compares three models, showing areas under ROC curves (AUC) based on 10-fold cross validation. First, using no collective inference a multiple logistic regression model was built using network-based variables constructed from the $G_1$ local neighborhood, following Hill et al. [9]. We see that we can get a substantial lift in performance, especially for the consumers who do not have a customer in their local neighborhood (non-NN). The second row shows the performance of the oracle-based procedure, showing a significant increase in predictive performance would be possible if the future behavior of the rest of the network were known. The final row shows that the collective inference procedure also increases predictive performance, in particular for the non-NN consumers—which is in line with our initial conjecture that collective inference would be useful for those consumers not strongly connected to existing customers.

| Attribute | NN | non-NN |
|---|---|---|
| **All $G_1$ local** | 0.61 | .71 |
| **All $G_1$ local + leave-one-out** | 0.63 | 0.74 |
| **All $G_1$ local + CI** | 0.62 | 0.74 |

**Table 1:** ***CI AUC analysis. I calculate each attribute's ranking ability by using AUC. The CI attribute alone is statistically significant for both network and non-network-neighbors. It is also significant when combined with the $G_1$ local neighborhood model.***

Deeper investigation reveals that the collective inference score indeed helps relatively more for those *network-neighbor* targeted consumers connected to fewer existing customers and most for the non-network-neighbor customers (connected to zero existing customers by definition). Table 2 shows the AUC and accuracy improvements for non network-neighbors, and network-neighbors separated into two groups, those that communicated with only 1 or 2 existing customers and those who communicated with 3 or more existing customers (see Table 2).

### 4.3.1 Pruning

| non-NN | | NN 1-2 | | NN >=3 | |
|---|---|---|---|---|---|
| **AUC** | **ACC** | **AUC** | **ACC** | **AUC** | **ACC** |
| 0.71 | 0.67 | 0.57 | 0.55 | 0.55 | 0.66 |
| 0.74 | 0.69 | 0.59 | 0.58 | 0.54 | 0.65 |

**Table 2:** ***AUC and significance analysis for network-neighbors split into high and low categories. The top row is without the CI score and the bottom row is the model including the CI score. The first set of columns correspond to non-network-neighbors and the middle columns correspond to the network-neighbors with links to one or two network neighbors and the last columns correspond to network neighbors with connections to three or more existing customers***

So far we have ignored one step in the creation of the approximate network: the pruning of low-information links (and corresponding nodes). Table 3 shows that pruning not only reduces further the size of the approximate network, it also improves the predictive performance using collective inference. Our findings are consistent with work in fraud detection [8] where pruned graphs were proven useful for identifying repetitive defaulters of a telecommunication service.

We chose $k$ based on a nested 10–fold cross validation, where 10-fold cross-validation on the training cases are used to pick $k$ for each hold-out sample. This section compares the results using top-k pruning to the results using collective inference without pruning. Working at $k$=25 reduces the number of nodes in the graph by 90%, or an order of magnitude, and gives results equally as good as operating on the entire graph. The method performs best around $k$=100. At $k$=100, approximately 29% of the edges remain.

| Attribute | NN | non-NN |
|---|---|---|
| **All $G_1$ local** | 0.61 | .71 |
| **All $G_1$ local + leave-one-out** | 0.63 | 0.74 |
| **CI** | .57 | .60 |
| **All $G_1$ local + CI** | 0.62 | 0.74 |
| **All $G_1$ local + CI pruned** | 0.63 | 0.75 |

**Table 3:** ***Pruning AUC and significance analysis. Pruning helps ranking for both the network neighbor and non-network-neighbor case.***

Singh and Getoor used a similar pruning strategy [15], and demonstrated its effectiveness in achieving reasonably good performance on a set of prediction problems using NASDAQ and NYSE businesses and on a bibliographic network. Here, we demonstrate that pruning uninformative nodes and edges not only maintains good performance relative to that achieved with larger graphs, but also can improve performance while greatly reducing the size of the approximate consumer network.

### 4.3.2 Combining Evidence

Finally, we also included traditional consumer attributes in our multiple logistic regression models, with and without the collective inference score. Again, as shown in Table 4, including the collective inference score improved the predic-

[1] Technically, we only use these probabilities as scores for ranking or as predictors in a subsequent model, rather than as true probabilities in a utility-based decision procedure.

tions. Specifically, first we used sequential forward selection to find the best model based on only traditional consumer attributes and local network attributes. Then, including the collective inference score in this model improved ranking performance (as judged by AUC) from 0.69 to 0.72 for the network neighbors and from 0.73 to 0.77 for the non-network-neighbor targets.

| **Attribute** | **NN** | **non-NN** |
|---|---|---|
| **All trad** | 0.68 | .72 |
| **All trad + All $G_1$ local** | 0.69 | .73 |
| **All trad + All $G_1$ local + CI prune** | 0.72 | 0.77 |

**Table 4:** ***Combining evidence AUC and significance analysis. I calculate each attributes ranking ability by using AUC. The CI attribute alone is statistically significant for both network and non network-neighbors. It is also significant when combined with the $G_1$ local neighborhood model.***

## 5. DISCUSSION

Starting with a social network with billions of nodes and edges, we chose a set of 1800 target nodes and build an approximate social network with about 1 million nodes. On this network it was feasible to perform univariate collective inference, based on the past adoption of other nodes in the network. This collective inference yielded marked improvements in the prediction of future adoption for the target nodes, when added to a variety of alternative models.

We have not yet explored to see just how much "depth" is necessary in the paths to existing customers. For example, how much value is added by combining the second-order networks for the individual targets into the larger graphs. We also have not explored to see how much value could be added by building multivariate relational models, including collective inference, rather than doing univariate collective inference separately and adding it to a non-relational logistic regression. In either of these cases, positive answers would only strengthen our conclusions that massive consumer networks can be approximated by much, much smaller networks, and still facilitate improved prediction through collective inference.

## 6. REFERENCES

[1] B. Balasundaram, S. Butenko, I. Hicks, and S. Sachdeva, *Clique relaxations in social network analysis: The maximum k-plex problem*, Tech. report, 2006.

[2] J. Besag, *On the statistical analysis of dirty pictures*, Journal of the Royal Statistical Society B **48** (1986), 259–302.

[3] C. Cortes, D. Pregibon, and C. Volinsky, *Computational methods for dynamic graphs*, Journal of Computational and Graphical Statistics **12** (2003), 950–970.

[4] P. Domingos and M. Richardson, *Mining the network value of customers*, Proc. of the 7th Intl. Conf. on Knowledge Discovery and Data Mining (San Francisco, CA), ACM Press, 2001, pp. 57–66.

[5] T. Fawcett and F. Provost, *Adaptive fraud detection*, Data Mining and Knowledge Discovery **1** (1997), no. 3, 291–316.

[6] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer, *Learning probabilistic relational models*, In Proc. of the 16th International Joint Conf. on Artificial Intelligence, 1999, pp. 1300–1309.

[7] S. Geman and D. Geman, *Stochastic relaxation, gibbs distributions and the bayesian restoration of images.*, IEEE Transactions on Pattern Analysis and Machine Intellingence **6** (1984), 721741.

[8] S. Hill, D. Agarwal, R. Bell, and C. Volinsky, *Building an effective representation of a dynamic network*, Journal of Computational and Graphical Statistics **15** (2006), no. 3, 584–608.

[9] S. Hill, F. Provost, and C. Volinsky, *Network-based marketing: Identifying likely adopters via consumer networks*, Statistical Science **22** (2006), no. 2, 256–276.

[10] D. Jensen, J. Neville, and B. Gallagher, *Why collective inference improves relational classification*, Proc. of the 10th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining, 2004.

[11] S. Li, W. Soh, and E. Teoh, *Relaxation labeling using augmented lagrange-hopfield method*, 1998.

[12] S. Macskassy and F. Provost, *Classification in networked data: A toolkit and a univariate case study*, Journal of Machine Learning Research (forthcoming 2007).

[13] J. Pearl, *Probabilistic reasoning in intelligent systems*, Morgan Kaufmann, 1988.

[14] M.J. Salganik and D.D. Heckathorn, *Sampling and estimation in hidden populations using respondent-driven sampling*, Sociological Methodology (2004), no. 34, 193–239.

[15] L. Singh, L. Getoor, and L. Licamele, *Pruning social networks using structural properties and descriptive attributes*, Proc. of the 5th IEEE International Conference on Data Mining (Washington, DC, USA), IEEE Computer Society, 2005, pp. 773–776.