RealGraph: User Interaction Prediction at Twitter

Krishna Kamath, Aneesh Sharma, Dong Wang, Zhijun Yin
Twitter, Inc.
@krishna_kamath @aneeshs @dongwang218 @zjyin

ABSTRACT

A common requirement for personalization in social networks is to estimate relationship strength for existing ties of a given user. In this work, we provide a framework to compute relationship strength for ties based on directed interactions between users. The proposed framework, called RealGraph, produces a directed and weighted graph where the nodes are Twitter users, and the edges are labeled with interactions between a directed pair of users. Further, each directed edge also has a weight that is the probability of *any* interaction going from the edge source to the edge destination in the future. The framework learns a logistic regression based model using historical data and then scores the edge features using the model to produce the weight.

We provide several applications of RealGraph at Twitter: it is used to compute better user recommendations, improve the relevance of user search results, and provide enhanced performance on any task that can benefit from separating strong ties from weak ones. Finally, we also provide an evaluation of the RealGraph based on both the effectiveness of the learning methodology and its performance from an application standpoint.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining

General Terms

Algorithm

Keywords

Social Network Analysis

1 Introduction

Online social networks enable users worldwide to *connect* to each other, with the connections taking a variety of forms. On Twitter, for instance, users can connect to each other not only via *following* each other (which enables them to receive content from each other) but also via a variety of interactions. For instance, users can *click* on each other's content (i.e., their Tweets), *retweet* other's tweets (propagating it to their followers), *favorite* the tweets, and in a number of other ways. These interactions provide a window into understanding *tie strength* on Twitter, i.e. we want to assign a quantitative "strength" value for each (directional) pairwise connection on Twitter. This paper details a framework we have built at Twitter for computing tie strength.

In addition to an increased understanding of user behavior, tie strength has a wide range of applications at Twitter: it helps compute better user recommendations, improves the relevance of user search results, and provides enhanced performance on any task that can benefit from separating close ties from weak ones. For example, Twitter users often search for a direct connection (for instance, to go to their profile pages) and surfacing users that they interact more with increases their satisfaction and usage of the product.

The problem of computing tie strength on Twitter has a few characteristics that make it challenging. First, as noted earlier, there are a variety of interactions between users that might have very different effects on tie strength: if a user follows two accounts, but retweets one's tweets and favorites the other's tweets, can we tell which tie is stronger? Second, Twitter has more than 250 million active users with billions of following edges, and billions of interactions take place every day. The framework needs to be capable of handling such a scale. Finally, there is the question of how would one interpret the computed tie strength. Ideally, we would like to make the computed weight a general and easily interpretable metric so that it is applicable for a variety of use cases.

We present a framework named *RealGraph* that we have built to measure tie strength at Twitter. The RealGraph is a directed, edge-labeled, weighted graph where the nodes are Twitter users, and the edges are labeled with interactions (from a fixed, extensible set) between a (directed) pair of users. Each edge also has a weight that is interpretable as tie strength and defined as the probability of *any* interaction going from the edge source to the edge destination in the future. The weight is learned using the edge labels as features and using historical interaction data for training. We emphasize that the RealGraph weight is generally applicable for the following reasons: (i) it is readily interpretable in many different settings: going back to the user search example, note that we in fact do want to prefer suggestions that the user would like to interact with, (ii) it is one measure of tie strength that looks forward as opposed to just being a historical summary, and (iii) it provides a quantitative measure of tie strength (via probability of future interaction) that can allow applications to use the score directly. For instance, one can use probability of interaction along with a thresholding mechanism as a relevance filter.

UEO '14 New York, New York USA

We also note that the framework is built such that the edge labels and the training pipeline can be adaped to learn a custom weight for an application.

Since it was built in late 2011, RealGraph has been used in a variety of applications inside Twitter. The first application (and original motivation) of RealGraph was in Twiter's user recommendation system, Who To Follow [5]. As detailed in previous work, a basic building block Who To Follow is random walk algorithm that runs on the Twitter social graph. The RealGraph output can be easily consumed by these algorithms to guide the random walks according to the weights. It biases the suggestions towards users that have stronger ties with their followers. We also note a related application for the RealGraph that is useful outside recommendations: it can be used as an effective pre-processing mechanism for finding the top-k connections of a user, which can provide an easy scaling scheme for applications (countering the heavily skewed distribution) while limiting the information loss. Finally, a variety of relevance products (such as search on Twitter, and the Discover page) use the RealGraph weights for personalized scoring of results.

The rest of this paper is organized as follows. In Section 2, we present the overview of the RealGraph framework, and discuss the edge features and related applications. In Section 3, we provide an evaluation of the RealGraph framework. In Section 4, we discuss some related work. We conclude the paper in Section 5.

2 Framework

We now detail the implementation of the RealGraph framework at Twitter. Our framework is mainly implemented in Pig [8] on Hadoop. Running on Hadoop enables our system to scale to several hundreds million of users, which is crucial requirement for Twitter. As shown in Figure 1, we have a "pipeline" of jobs, which consists of essentially three components: graph (and feature) generation, feature scoring and applications. The graph generation step is detailed in Section 2.1. Recall that the graph vertices correspond to users. For each user, we obtain a set of outgoing edges are them using a combination of people they are following, people in their address book[1], and users who they interacted with. Further, each edge is labeled with features that contain information such as when and how often an interaction happens from the source user to the destination user. These edge features and the corresponding aggregated vertex features are used in the model scoring step in Section 2.2, where we apply a logistic regression model that has been trained separately. The learned model is applied to each graph edge using both the edge and vertex features to compute the weight, i.e., the probability of future interactions on this edge. In Section 2.3, we present an example of Hadoop computations that apply RealGraph weights for recommendation, and search.

2.1 Graph Generation

There are three ways for an edge from user A to user B to be added to the RealGraph in any given time period: (i) if A follows B, (ii) if B is in A's phone or email address book (again, only if the requisite permissions have been granted by the users), and (iii) if A interacted with B. Currently, we compute the RealGraph on a daily basis, so we add these edges daily. Then, we merge them with the previous day's RealGraph, resulting an aggregate version. We note that to prevent this data from snowballing, we decay historical interaction values and remove an edge if the most recent interaction is too old (this is adjustable with a parameter). The details of graph edges with various edge features are describe in Section 2.1.1. We also associate user features on graph vertices, which are presented in Section 2.1.2.

2.1.1 Edge Features

We distinguish three variants of the following relationship for RealGraph: a one-directional follow (the most common), two-directional follows (where both users follow each other), and an SMS-follow where A not only follows B, but has opted in to receive all of B's tweets as either SMS or in-application notifications. Furthermore, we not only store a Boolean value on each of these features but also store a number of quantities. For follow edges, we store the number of days since the edge was created. The address book edge is also annotated with this quantity.

The interaction edges are treated in a slightly different manner. We collect two kinds of interactions from a user A to B: (1) Visible interactions (from B's viewpoint): A retweets B's tweet, A favorites B's tweet, A mentions B (via their handle), or A messages B, and (2) Implicit interactions: A clicks on B's tweet or on a link within the tweet, A visits B's profile page.

For each of these interactions, we collect several time-series related values that aim to capture the frequency, intensity and recency of each interaction when it happens:

- Non-zero days: the number of days when such an interaction happens
- Mean and variance: the mean and variance of interaction counts (computed over non-zero interaction days)
- Decayed count: a daily exponentially-decayed interaction count (EWMA)[2]
- Days since last interaction: number of days since the last interaction of this type
- Elapsed days: number of days since the first interaction of this type happened

Finally, we mention a few other edge features that are in addition to the above. First, we have a feature for the number of different non-zero interaction types for an edge as we have found diversity of historical interactions between two users to be a good indicator. We also have a feature for the number of common friends (users that both follow and are followed by A and B) to measure closeness in terms of the graph. We also add a few topic-related edge features on each edge. For computing these, we use the 300 topic taxonomy and high precision topic modeling system to assign interested-in and known-for topics for each user as describe in [10]. This results in several edge features such as the number of common topics between source user's interested-in and destination user's known-for etc.

2.1.2 User Features

For each edge feature type described in previous section, we aggregate the values and use it for the source user as sending-feature and for the destination user as received-feature. For example, the EWMA of a user's total retweets is the sum of EWMA of retweet interactions on all its outgoing edges. On

[1]We only add address book edges where *both* ends of the edge have allowed Twitter permission to use this information.

[2]When this value becomes too small for an interaction type, the feature is removed. This in turn removes the edge if this is the only feature.

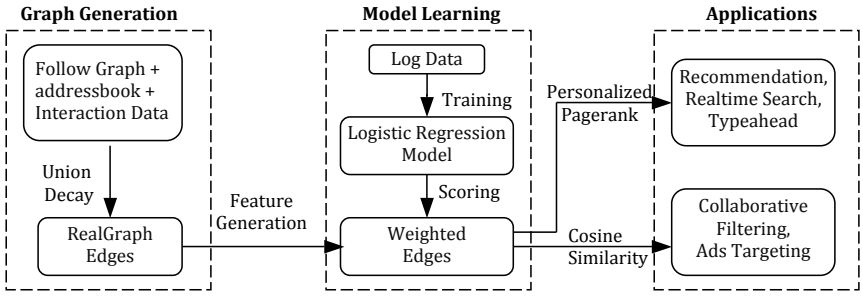

Figure 1: Twitter RealGraph Framework.

the other hand, the mean of mentions is the daily average of tweets containing the user's name. We also include several features for each user's activity and reputation. These include number of tweets in the last week, language, country, number of followers, number of people they follow, and PageRank on the follow graph.

2.2 Model Training and Scoring

We use the PigML pipeline at Twitter [6] to periodically train a logistic regression model and use it daily to score the updated RealGraph edges. Given an edge, we use edge features (Section 2.1.1) and vertex features of both the source user and the destination user (Section 2.1.2) for classification. The target for learning is to predict interactions in a period given feature values before beginning of that period (and taken from a period of the same length). To remove the effect of fluctuations in users' activity, we apply two filters when creating the training dataset: an edge is eligible for becoming a training instance only if the destination user wrote at least one tweet in the test period, and if the source user had created at least one interaction in the test period.

The training task is set up as a binary classification problem: the training label for an edge is set to 1 or -1 depending on the existence of any interaction. Thus, the training task is to learn what combination of features in a period can predict whether there will be *any* interaction on the same edge in the future. This particular setup exists for two reasons: (i) we still learn to combine and distinguish between features of different kinds, which was an original goal, and (ii) predicting the union of interactions is very general, which captures a diversity of use cases. To evaluate the effectiveness of features, we group features into 10 groups and calculate the incremental AUC improvement. The details are in Section 3.

2.3 Applications

We briefly describe a few applications of the RealGraph weights. Recall that since the RealGraph weight corresponds to the probability of future interactions on an edge, it provides a quantitative measure of tie strength between users. This helps differentiate a user's followings, so that edges that do not see interactions can be downgraded while at the same time, edges that see large and repeated interactions are upgraded. This local graph "pruning" makes it easier for graph based algorithms to focus on high quality edges.

We now note a few graph algorithms at Twitter that use the RealGraph weights. As mentioned earlier, user recommendations at Twitter are driven by personalized algorithms such as personalized PageRank that run on the social graph [5]. We also compute and store a large set of accounts for each user that have high personalized PageRank for the user (called the *Circle of Trust*), which is then used by a variety of systems such as Search and Discover to enhance relevance. We can achieve better personalization by incorporating RealGraph weights in these personalized PageRank computations (which are implemented via random walks). This is made seamless by the fact that our personalized PageRank computations now run on Hadoop [5]. In addition to benefitting user recommendations, better personalization for Circle of Trust also enhances the quality of our personalized tweet search [3] as tweets from a user's Circle of Trust are ranked higher in the search result page. Another application of Circle of Trust is TypeAhead (also known as auto-complete) for Twitter user search that has also seen improvements in click rate for RealGraph weight-enhanced Circle of Trust.

We also mention the problem of discovering similar users on Twitter [4] that has applications in ads targeting, collaborative filtering and community detection. We say that two Twitter users are considered similar if they are followed by a similar set of users. As mentioned in previous work [4], our similarity computation also takes interactions into account by running the similarity computation (cosine similarity) on the RealGraph. We refer to the original work for details about the similarity framework.

3 Evaluation

In this section, we provide an evaluation for the performance of RealGraph. We can perform two kinds of evaluation for the RealGraph: (i) a self-evaluation that measures the effectiveness of RealGraph in its own learning task, and (ii) an evaluation from an application standpoint in measuring the effectiveness of the RealGraph in enhancing quality for the metric. We present results from both of these evaluations. First, we evaluate the RealGraph learning effectiveness by splitting the features into several groups and then computing the area under curve (AUC) for each group using a forward stage-wise procedure in Section 3.1. Then, we evaluate the quality of the RealGraph by conducting a small user survey for the application of determining precision of top-k edges in Section 3.2.

3.1 Self-evaluation

In this evaluation, we use historical interactions to build RealGraph models as described in Section 2 and then use the AUC metric to evaluate these models. Given a week of interaction data we build a RealGraph model using features described before with aggregated RealGraph edges for the

Feature group	AUC
Historical interaction ewma/mean	0.830
Historical interaction non_zero_days	0.849
Historical interaction type	0.849
User reputation	0.852
Social graph features	0.853
Topic similarity	0.854
User activity	0.876
User country	0.877
Address book	0.877

Table 1: Performance of incrementally adding various feature groups for RealGraph model

first day of the week. As before, we split the daily interactions for the remaining days of the week into two types: (i) visible interactions (including retweet, favorite, mention, direct message) and; (ii) implicit interactions (including tweet clicks, link clinks, profile clicks). We sub-sample visible interactions by giving them 5x weight of implicit interactions. Recall that all the edges in the aggregated RealGraph for the first day are labeled as 1 if the edge saw any interaction in the following week and -1 if it did not. We do not perform any explicit balancing of positive and negative training instances. Using this data we train a model using stochastic gradient descent logistic regression with L_2 regularization and determine the AUC for the model.

The results of this evaluation are shown in Table 1. The first two groups uses different time series for historical features like Retweets, Favorites and Messages. The next group adds implicit interaction and edge type (follow edge, bidirectional, or SMS). We then used features based on the reputation of the user and the features extracted from the network like total follows, and total followers. We also evaluated the impact of adding features related to topics that a user is associated with. For a given edge we used the number of overlapping topics as the feature value. Next we used features obtained from the summary of user's activity like number of posted tweets, number of sent favorites and user's country. The last set of features is obtained from user's address books. The improvement of AUC as we added new features is shown in the second column. As is clear, adding vertex features has been quite valuable. Further, we note that consistency of edge features (reflect by non-zero-days) is also a good predictor of future interactions. Finally, user activity is certainly related to enhanced user activity, so the likelihood of interactions also goes up.

3.2 Application Evaluation

In this section, we provide results from a small survey of Twitter users that evaluates the performance of RealGraph on the precision of identifying their top followings. For this survey we recruited users who accessed Twitter on web and used English as their primary tweeting language. We also selected only those users who used the service regularly and hence knew users that they are following well.

For every target user, we identified the top-20 Twitter accounts they follow using RealGraph score. From this set we selected 10 users using stratified sampling as follows: 4 for top 5, 2 from 6-10, 2 from 11-15 and 2 from 16-20. The survey responders were then shown these selected users and were asked to label them into three categories by asking: "Below are ten people you follow. Please tell us how many Tweets from each person you find interesting: (i) All or almost all of the Tweets; (ii) Some of the Tweets; and (iii) A few or none of the Tweets." The responses were not mandatory and users could choose to skip questions.For this survey we were interested in evaluating the precision of the model and hence concentrated on top-20 users only.

We ran this survey on twitter.com for 14 days and collected 10,982 responses for the survey. The survey had a click through rate of 3.1%. We observed that on average people marked 40% of the users in their list in first category, 42% in second category and remaining in the third category. So overall people found than 82% of Twitter followers identified by RealGraph have reasonable precision. We emphasize that this is a preliminary result and is best interpreted qualitatively. Further, the effectiveness of RealGraph is best evaluated separately for each application, which is beyond the scope of this paper.

4 Related Work

We provide a brief survey of relevant work here. We are not aware of any directly comparable work that aims to serve as a general personalization framework. The closest work to ours is [2], where the authors build a logistic edge weight model that is optimized such that personalized PageRank based on the model weights would rank future friends higher than non-friends. We note that our work differs in the basic goal in that the resulting edge weights are not only used to rank users and their tweets, but more importantly they serve as a general personalization framework for search, recommendation and targeting at Twitter.

Predicting user engagement has also been studied for web search, sponsored search advertising, and display advertising. Using interaction log data, [1] develop a model of search success based on realistic web search tasks, and train effective models for predicting search success. [7] describes practical application of CTR prediction system for sponsored search advertising in an industrial setting. To overcome conversion sparsity in display ads, [9] trains a high-dimensional models on proxy populations and then transfers that knowledge to the real prediction target using a lower-dimensional stacked ensemble model. None of these are directly applicable to our setting.

5 Conclusion

In this paper, we presented the user interaction prediction framework, named RealGraph, developed at Twitter. RealGraph consumes heterogeneous interaction data to effectively predict potential user interaction in the future. The prediction score of the user interaction can also be interpreted as connection strength, which enables a diverse set of applications to use the RealGraph. The successful deployment of these applications makes RealGraph an essential component of graph processing tools at Twitter.

We also note that there are many directions for future work on the RealGraph. First, extending the framework to consume real time signals would be of immediate interest as RealGraph is currently updated only in batch mode. Further, an online learning approach might be a nice fit for processing real time updates. Second, modeling of time decay effect can be improved as some user interactions are much more time-sensitive than others, so incorporating the temporal pattern for features could be quite useful. Third, it would be interesting to explore dependency among the users for better prediction performance. For example, one user might always interact with another two users simultaneously if those two have similar attributes.

6 References

[1] M. Ageev, Q. Guo, D. Lagun, and E. Agichtein. Find it if you can: a game for modeling different types of web search success using interaction data. In *SIGIR*, pages 345–354, 2011.

[2] L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. In *WSDM*, pages 635–644, 2011.

[3] M. Busch, K. Gade, B. Larson, P. Lok, S. Luckenbill, and J. Lin. Earlybird: Real-time search at twitter. In *ICDE*, pages 1360–1369, 2012.

[4] A. Goel, A. Sharma, D. Wang, and Z. Yin. Discovering similar users on twitter. In *11th Workshop on Mining and Learning with Graphs*, 2013.

[5] P. Gupta, A. Goel, J. Lin, A. Sharma, D. Wang, and R. Zadeh. Wtf: the who to follow service at twitter. In *WWW*, pages 505–514, 2013.

[6] J. Lin and A. Kolcz. Large-scale machine learning at twitter. In *SIGMOD Conference*, pages 793–804, 2012.

[7] H. B. McMahan, G. Holt, D. Sculley, M. Young, D. Ebner, J. Grady, L. Nie, T. Phillips, E. Davydov, D. Golovin, S. Chikkerur, D. Liu, M. Wattenberg, A. M. Hrafnkelsson, T. Boulos, and J. Kubica. Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2013.

[8] C. Olston, B. Reed, U. Srivastava, R. Kumar, and A. Tomkins. Pig latin: a not-so-foreign language for data processing. In *SIGMOD Conference*, pages 1099–1110, 2008.

[9] C. Perlich, B. Dalessandro, T. Raeder, O. Stitelman, and F. J. Provost. Machine learning for targeted display advertising: transfer learning in action. In *Machine Learning*, volume 95, pages 103–127, 2014.

[10] S. Yang, A. Kolcz, A. Schlaikjer, and P. Gupta. Large-scale high-precision topic modeling on twitter. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, New York, NY, USA, 2014. ACM.