Retweet Modeling Using Conditional Random Fields

Huan-Kai Peng*, Jiang Zhu*, Dongzhen Piao*, Rong Yan† and Joy Ying Zhang*
*Carnegie Mellon University Silicon Valley, {huankai.peng,jiang.zhu,dongzhen.piao,joy.zhang}@sv.cmu.edu
†Facebook Inc, rong.yan@fb.com

***Abstract*—Among the most popular micro-blogging service, Twitter recently introduced their reblogging service called retweet to allow a user to repopulate another user's content for his followers. It quickly becomes one of the most prominent features on Twitter and an important mean for secondary content promotion. However, it remains unclear what motivates users to retweet and whether the retweeting decisions are predictable based on a user's tweeting history and social relationships. In this paper, we propose modeling the retweet patterns using conditional random fields with a three types of user-tweet features: content influence, network influence and temporal decay factor. We also investigate approaches to partition the social graphs and construct the network relations for retweet prediction. Our experiments demonstrate that CRF can improve prediction effectiveness by incorporating social relationships compared to the baselines that do not.**

***Keywords*-Social Network; Twitter; Conditional Random Fields;**

I. Introduction

Twitter, a microblogging service, has demonstrated its strength as an effective new medium. It allows users to freely spread realtime information through both the WWW and the mobile phone networks, where each piece of information may further be tagged for searching and grouping. Because of these features, Twitter has successfully distinguished itself from traditional media under a number of events (i.e., the 2008 U.S. Presidential Election [17], the 2009 Iran Election and Protest [6], [5], and the Haiti Earthquake [1]) by being more spontaneous, mobile, and disseminative. Much research has since been directed to twitter's potential applications, such as education [10], scientific communication [16], politics [8], and disaster response [22].

"Retweeting" is the most powerful mechanism to diffuse information via twitter. It is a conventional Twitter practice: when a user find a tweet worth sharing, he could copy the entire post and re-publish it to his followers. The information could thus reach beyond the network of the original author while the content remains relatively intact. Most existing studies on retweeting try to analyze retweeting behaviors and related factors. For retweeting behaviors, various motivations are explored in [9], while the propagation graph and statistics are studied in [7] and [14]. For retweeting-related factors, [27] and [20] found that retweeted and normal tweets are different in dimensions such as the inclusion of URL's and hashtags, publish time, wording, author publicity, and even the URL shortening service used.

Up to now, few work tries to build a model to predict the retweeting decisions of a targeted network. Such a fine-grained prediction can be used to estimate the spread of crucial informations, and will be beneficial to applications such as emergency response and viral marketing. One study toward this problem is [20], which builds a regressor to predict the aggregate number of retweets for a given tweet. However, since the number of followers for a Twitter user ranges from none to millions, using such an aggregate prediction is very difficult to estimate the information spread. The authors of [25] addresses the same problem as we do by means of constrained optimization using factor graphs in a generative manner. This approach, however, can be improved because historical retweeting decisions are labeled and plentiful. Therefore, since the problem only concerns predicting the labels given observation, a discriminative approach is proven to perform better then a generative one. Further, the perspectives about network structures and about how the approach should be adopted to larger networks are left unexplored.

In this paper, we aim to build a fine-grained predictive model for retweeting. Specifically, given a tweet, we would like to predict the retweeting decision of each users within a targeted network. Such a problem is challenging in several ways because (1) many local factors can contribute to an users's decision (2) dependencies exist between the multiple users' decisions that can further depend on a number of network factors, and (3) as the size of the network goes up, or when the structure of the network become more complicated, both prediction and runtime performances become major challenges.

To address the first two challenges, we utilize Conditional Radom Fields (CRF) to formulate the problem into a retweeting probability conditioned on the incoming tweet and targeted users. In our formulation, both the local effects of individual users and the network effects between user relationships are modeled explicitly using different sets of potential functions, where their weights are learned directly from data. To address the third problem and to make our method applicable for larger and more complicated networks, we take advantage of the retweet network's "small-world" nature. According, we combine CRF with network partitioning and separate training and prediction according to partitioned subnetworks. A series of experiments are conducted, and show that both of the above approaches can bring improvements in terms of prediction performance and runtime. We also closely look into a specific case to see how and why the proposed methods work.

The rest of this paper is structured as follows. In Section 2, we describe the Conditional Random Fields formulation as our base predictor. In Section 3, we describe retweet networks' properties as well as how to take advantage of them to improve the effectiveness of the base predictor. In Section 4, we describe the features used in this study. In Section 5, we present our experiment results. Finally, we conclude the paper in Section 6.

II. Retweet Modeling using Conditional Random Fields

We consider a *Twitter Network* $G = (U, E)$ such that each node represents a user $u \in U$ and each edge $(u, v) \in E$ represents the following relationship between u and v. We also introduce the *User Entities* $\mathbf{x} = \{x_1, \ldots, x_{|U|}\}$ such that each x_u corresponds to an $u \in U$. Finally, the binary *Retweet Decisions* $\mathbf{y} = \{y_1, \ldots, y_{|U|}\}$ represents the retweet decisions of all users.

Definition 2.1. **Network Decision Problem**. Given G, the *Network Decision Problem* is defined as predicting the decisions for all users in U given a new tweet r and the user entities $\mathbf{x}$. Equivalently, the problem is finding the *maximum a posteriori probability* (MAP) assigment:

$$\hat{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y}} \mathrm{P}(\mathbf{y}|r, \mathbf{x}). \tag{1}$$

There are three characteristics associated with this problem. First, there are dependencies between the y_i's. Second, many features coming from the interaction between r and $\mathbf{x}$ that can aid predicting the decisions. Third, rather than the joint distribution $\mathrm{P}(\mathbf{y}, r, \mathbf{x})$, we are only concerned with the conditional distribution $\mathrm{P}(\mathbf{y}|r, \mathbf{x})$. These characteristics motivate us to model the problem using Conditional Random Fields [15].

A. CRF-based Formulation

We assume that the decision of each user y_u follows the Markov property with respect to G in that

$$P(y_u|r, \mathbf{x}, \mathbf{y}_{U-\{u\}}) = P(y_u|r, \mathbf{x}, \mathbf{y}_{\mathcal{N}_u}) \tag{2}$$

where $U - \{u\}$ denotes the set of all users except user u; $\mathcal{N}_u$ denotes the set of neighbors of user u in E. Then the conditional probability of Equation (1) can be modeled using CRF as:

$$P(y|r, \mathbf{x}) = \frac{1}{Z(r)} \prod_{C \in \{\mathcal{C}_U, \mathcal{C}_E\}} \Phi_C(r, \mathbf{x}, \mathbf{y_C}) \tag{3}$$

where the graphical structure is depicted in Figure 1. In this equation, $Z(r)$ is a normalization term that ensures the probability sums to 1. $\mathcal{C}$ denotes the set of *Cliques* according to which the conditional distribution is factorized, which are grouped into *node cliques* $\mathcal{C}_U$ and *edge cliques* $\mathcal{C}_E$. The Φ_C's denote the *Potential Functions* defined over cliques, having the form

$$\Phi_C = \exp\left\{ \sum_{k}^{K(C)} \lambda_{ck} f_{ck}(r, \mathbf{y_C}, \mathbf{x_C}) \right\} \tag{4}$$

where the $f(.)$ and λ denote the features and their respective coefficients, and $K(.)$ denotes the number of features of a type.

B. Feature Definition

We group the features into three types: the first two defined for the nodes, the last defined for the edges.

Tweet Features $f_T^u(r, y_u)$ incorporate the features making the incoming tweet retweetable in its own right. These features do not depend on users: e.g., whether the tweet r contains an URL.

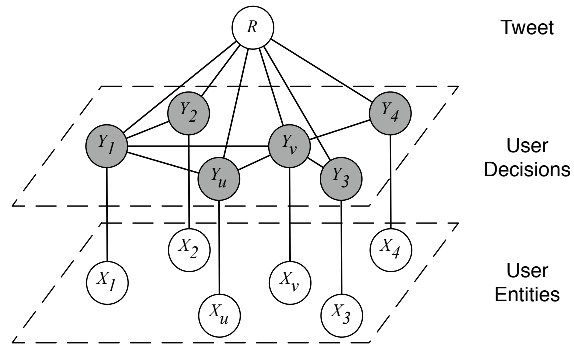

Figure 1. The graphical structure of the Retweet CRF

Each user still have a separate coefficient because such a factor can influent everyone's decision differently.

User Features $f_U^u(r, x_u, y_u)$ incorporate the perspectives that the incoming tweet is retweetable for an user u because it matches his own characteristics. These features therefore depend on the user u: e.g., the content similarity between u's historical retweets and tweet r.

Relationship Features $f_N^{u,v}(r, x_u, x_v, y_u, y_v)$ incorporate the perspectives whether the tweet is retweetable simultaneously for two users u and v because it matches some characteristics of this user-pair. These features therefore depend on the interactions between u and v: e.g., the content tri-similarity between u's retweets, v's retweets, and the incoming tweet r.

Based on these feature definitions, we can rewrite our CRF formulation as follows

$$\begin{aligned} &P(y|r, \mathbf{x}) = \\ &\quad \exp\left\{ \sum_u \left(\sum_k^{K(T)} \lambda_{Tk} f_{Tk}(r, y_u) + \sum_k^{K(U)} \lambda_{Uk} f_{Uk}(r, x_u, y_u) \right) \right. \\ &\quad \left. + \sum_{u,v} \left(\sum_k^{K(N)} \lambda_{Nk} f_{Nk}(r, x_u, x_v, y_u, y_v) \right) \right\}. \end{aligned} \tag{5}$$

C. Parameter Estimation

To learn the parameters $\theta = \{\lambda_k\}$, we obtain a set of historical tweets R and form a training set $\{\mathbf{y}^{(\mathbf{i})}, r^{(i)}\}_{i=1}^{|R|}$. We then maximize the regularized log-likelihood

$$\begin{aligned} l(\theta) &= \sum_{i=1}^{|R|} \log P(y^{(i)}|r^{(i)}, \mathbf{x}) - \sum_{k=1}^{K} \frac{\lambda_k^2}{2\sigma^2} \\ &= \sum_{i=1}^{|R|} \Big(\sum_u \big(\sum_k^{K(T)} \lambda_{Tk} f_{Tk}(r, y_u^{(i)}) + \sum_k^{K(U)} \lambda_{Uk} f_{Uk}(r, y_u^{(i)}, x_u) \big) \\ &\quad + \sum_{u,v}^{K(N)} \lambda_{Nk} f_{Nk}(r, y_u^{(i)}, y_v^{(i)}, x_u, x_v) \Big) - \sum_{k=1}^{K} \frac{\lambda_k^2}{2\sigma^2} \end{aligned} \tag{6}$$

by solving $\frac{dl(\theta)}{d\theta} = 0$. Since the negative ${\lambda_k}^2$ term makes $l(\theta)$ concave, it be efficiently approximated by second-order methods such as L-BFGS [4].

III. NETWORK STRUCTURE OF CRF

The key challenge of CRF-based modeling is having the graph structure fit the target network. An overly simplified structure is not expressive enough to capture the characteristics of the target network [21], while an overly complex structure with excessive edges are very expensive to train and infer. Previously when CRF are applied on Natural Language Processing and Computer Vision, simpler structures like linear-chain [21] or tree [3] deliver good results because they fit the sequential and hierarchical nature of text and images. Twitter networks, however, contain cycles, which is one of the most challenging structures for CRF and usually leads to intractably runtime for large networks. To make our model practical, it is necessary to investigate the properties of the retweet network.

A. The Small-World Twitter

Watz and Strogatz [24] discovered the *small-world* property embedded in many man-made networks, such as the US western power grids and the collaboration between film actors. Later, Adamic [2] found that the links between all sites in the World Wide Web also form a small-world network. Such a property is define on a graph that is (1) highly clustered and (2) the *Average Path Length* (APL) of all node pairs is small. Also, a small-world network conforms the 80-20 rule: an individual in the network is primarily influenced by only a minor portion of his connections.

From the small-world definition, the retweet network is very likely to be a small-world, too. The reasons are twofolds: (1) retweeting is disseminated in a radiation-like manner through relationships, which is clustered in nature. (2) The APL for Twitter is 4.1 according to [14], which is shorter than that of physical human networks (around 6). From 80-20 rule, we may keep only a fraction of edges that reserves the essential clustering structures of the retweet network.

B. Graph Partitioning

While the general graph partitioning problem is NP-complete, a commonly used O(n^2log$n$) heuristic is the Kernighan-Lin algorithm [11]. Our partitioning algorithm adapted the same greedy spirit as in Algorithm 1. It takes as input a graph G and a controlling parameter δ, and gives two disjoint sets of nodes from the original graph. Before starting the greedy iterations, it first add $\delta \times |U|$ dummy nodes with no edges, and produce a random partition A and B. At the end, the size difference between the two partitions is bounded by $\delta \times |U|$.

Let the *internal cost* I_a be the sum of all edge costs between a node $a \in A$ and every other nodes in A, and let the *external cost* E_a be the sum of all edge costs between a and all nodes $b \in B$. Further, let the difference $D[a]$ be $D[a] = E_a - I_a$. If we exchange a node $a \in A$ with another $b \in B$, then the reduction g in inter-partition cost is

$$g = D[a] + D[b] - 2 \times \text{mutualCost}(a, b) \tag{7}$$

where mutualCost(a, b) adds up of all edge costs between (a, b).

In each iteration from line 2 to 17, we first calculate the D for all nodes. Then in the loop from line 6 to 12, it greedily picks

Algorithm 1 Graph Partitioning with approximately equal size

Input: $G(U, E)$: the input graph
Input: δ : the parameter controlling the difference between the sizes of output partitions
Output: A,B:the output partitions (nodes only)
1 (A, B) = randomPartition(U, $\delta \times |U|$)
2 **repeat**
3 $(\bar{A}, \bar{B}) = (A, B)$
4 $D[a]$ = externalCost(a) - internalCost(a), $\forall a \in \bar{A}$
5 $D[b]$ = externalCost(b) - internalCost(b), $\forall b \in \bar{B}$
6 **for** $k = 1$ to $|U|/2$ **do**
7 find $aa[k] = a^* \in \bar{A}$ and $bb[k] = b^* \in \bar{B}$ s.t. $g[k] = D_{a^*} + D_{b^*} - 2*$mutualCost$(a^*, b^*)$ is maximal;
8 move a^* to $\bar{B}$ and b^* to $\bar{A}$
9 remove a^* and b^* from further selections in this loop
10 update $D[a]$ for nodes in $\bar{A} - \{a^*\}, \forall a \in \bar{A}$
11 update $D[b]$ for nodes in $\bar{B} - \{b^*\}, \forall b \in \bar{B}$
12 **end for**
13 find k^* s.t. $g_{max} = \sum_{k=1}^{k^*} g[k]$ is maximal
14 **if** $g_{max} > 0$ **then**
15 Exchange $\{aa[1], \ldots, aa[k]\}$ with $\{bb[1], \ldots, bb[k]\}$ in A and B
16 **end if**
17 **until** $g_{max} \leq 0$
18 remove all dummy nodes in A and B
19 **return** (A, B)

the best node pair to switch according to their cost reduction g. The iteration repeats until the best g is negative.

Note that when any node is switched with a dummy node, it is equivalent to move that node to another set. Such a switch can only reduce the total cost, because it does not block any better switches between real nodes. Finally, because there are only a total of $\delta \times |U|$ dummy nodes, the most unbalanced case happens when all of them reside in one set. We recursively partition the whole networks into subnetworks, while the number of edge-potential terms being removed is minimized.

C. Implicit Graphs

Up to now we have assumed by default the original network structure used by CRF is identical to the Twitter users' following relationships. However, it is possible that the underlying retweeting network forms a different shape. One hypothesis is that retweeting behavior may be more about self-image projection, while following another user is to maintain a certain relationships. If this is true, we should be able to remove more edges from the current network to reduce complexity of the problem while still being able to properly model the retweeting network.

Based on the original explicit network, an intuitive way to build an implicit network could be reserving an edge in the original network only if the users in both ends have a certain co-retweeting history, say, if they ever co-retweeted from the same author. Such an alternative hypothesis is investigated in Section 5.2 and 5.4.

Table I

FEATURE LIST GROUPED BY BOTH FEATURE AND INFLUENCE TYPES. THE LAST COLUMN LISTS LDA-BASED RANKING

Influence	Description	Qty	LDA Rank
Tweet Features			
Content	Topic similarity	3	24,25,30
	URL	2	38,39
	Hashtag	2	35,36
	Mention	1	20
Network	Author's friend/follower	2	8,32
	Author's tweet/retweet	2	9,28
User Features			
Content	Topic similarity	9	2,13,17,19,21 22,23,26,31
	URL	1	33
	Hashtag	1	4
Network	Relationship with author	6	1,3,5,6,11,16
	Relationship with mention	6	7,10,12 14,15,18
Temporal	Self activity	2	29,37
	Friend activity	2	27,34
Relationship Features			
Content	Topic similarity	1	-
Network	Relationship with author	1	-

IV. FEATURE GENERATION

From Section 2, we defined the tweet, user, and relationship features. In this section, we introduce *source of influence* as an orthogonal axis to more intuitively enumerate features, which are the content, the network, and the temporal influences. All features are summarized in the first three columns of Table I.

A. Content Influence

Topic Similarity. On Twitter, a tweet may draw an user's attention because the it is interesting for general public, for the user specifically, or for the user's friends. Considering each tweet as a small document, a topic is modeled as a a probability distribution of tokens. The global interests model is built using all the tweets and retweets in the data set, while an individual interest model is built with all the tweets authored or retweeted by this person. Both of them are represented as term frequency vectors, where the similarity between them is defined as the cosine distance, including:

- Similarity of global interest and the tweet
- Similarity of an user's followers' interest and the tweet
- Similarity of an user's friends' interest and the tweet
- Similarity of an user's own interest and the tweet
- Tri-similarity of two users' interests and the tweet

URL, hashtag, and mention. It is reported in [20] that URL's, hashtags, and mentions can help predict retweets. We include a series of such features: whether the tweet contains a URL; how frequent does the (unshortened) URL domain appear in global and an user's retweets; whether the tweet contains a hash tag; how frequent the hash tag appear in global and an user's retweets; whether the tweet mentions other users and how often they have been mentioned elsewhere. Overall, content influence accounts for 8 tweet features, 11 user features, and 1 relationship feature.

B. Network Influence

Author context. From [20], social credibility is essential for an author to get retweeted. Such credibility of an author is modeled by his (1) number of friends (2) number of followers (3) number of tweets and (4) number of retweeted tweets.

Social relationship Whenever an author, a retweeter, or a mentioned person in the tweet has social connection with a reader, it is more likely for him to retweet the tweet. We measure such a social tie between two users by the number of their (1) mutual friends, (2) mutual followers (3) mutual mentions, and (4) mutual retweets. For each tweet and each user, we calculates these measures between the user, the author, and the mentions in the tweet. We also measure the number of co-retweets from the two users to an author. Overall, network influence accounts for 4 tweet features, 12 user features, and 1 relationship feature.

C. Temporal Influence

The importance of timing in Twitter has been discussed in both [14] and [27]. From [14], half of retweets occur within an hour, and 75% within a day. Such an observation suggests a "window of survival", from 1 hour to 1 day, where a certain tweet gets a higher chance to get retweeted. To model the timing factor at user level, consider the scenario when using the standard Twitter Web interface. Whenever a user checks his timeline, only a certain unread tweets can fit in one page view, where tweets with higher ranks naturally have better chances to get retweeted. To model this effect, we introduce two features for a tweet: the first characterizes a user's response time to it; the other characterizes its rank in the timeline.

Self activity. We model a user's response time as a poisson process similar to [19]: for an user u, we denote his activity level at time t as $h_u(t)$, estimated by the average number of tweets he publishes in a periodical time slot, e.g., every Wednesday. The average waiting time is then estimated by $\frac{1}{h_u(t)}$

Friends' activity. With the estimated response time Δt, the number of accumulated tweets can be written as:

$$\sum_{j \in F_u} \int_{t^w}^{t^w + \Delta t} h_j(t)\ dt \tag{8}$$

where F_u denotes the set of user u's friends. We calculate both activities using periods of a day and a week. Accordingly, temporal influence accounts for 4 user features.

V. EXPERIMENTAL RESULTS

We used Twitter API to obtain the data of 260,700 users and 92,149,804 tweets. Using these data, we build a dataset that consists of 1000 users and 25,704 unique tweets. For each tweet in the dataset, we calculate the features for all 1000 users and their relationships as mentioned in Section 4.

Using the data, we first apply an analysis on node-features. Second, we show that improved results can be obtained by incorporating the network effects compared to the baseline that considers only local characteristics. Third, we show that partitioning the network will bring better results in modeling larger networks. Finally, we conduct a micro-scope analysis to

Table II
SETTINGS FOR MEASURING THE CRF PERFORMANCES

Experiment 1 Settings				
	Constant	CoSim	CoRt	Both
No Edge	*Logistic Regression (LR)*			
Explicit Edge	*E-Cons*	*E-CoSim*	*E-CoRt*	*E-Both*
Implicit Edge	*I-Cons*	*I-CoSim*	*I-CoRt*	*I-Both*

investigate how and why the proposed approach can deliver better results. All experiments are conducted on a linux-based machine with 16 Intel Xeon 2.54GHz cores and 16G Memory.

A. Feature Analysis

To gain more insight on the features, we applied Linear Discriminant Analysis (LDA) with combined data from all users (excluding edges and, thus, relationship features). Such a result does not contribute to feature selection of CRF training. Rather, it is the feature importance ranking when building a single user model that tries to fit all users.

After applying LDA on normalized data, we sort the features according to the absolute values of their coefficients for the first linear discriminant. The ranking is summarized in the last column of Table I, where 3 facts are observed. First, user features are generally better predictors compared to tweet features. In other words, whether a tweet fits an user is a better indicator for a user's retweeting decision compared to whether it is popular in general. The only exceptions are the numbers of the author's tweets and followers, which accounts for the author's credibility. Second, among the user features, the best predictors are all obtained from the user's previous retweet: author in previous retweets, topics of previous retweets, mentions in previous retweets, hashtags in previous retweets. Apart from these, the best predictors are those of network influences from the authors and mentions. In fact, features of network influences are slightly better predictors compared to that of content influences, which is often overlooked in previous works. Third, some features are not as predictive as expected, such as URL and time factors. But this might be due to our specific feature design, which may still need further investigations.

B. Unpartitioned CRF Performance

We test the prediction performance of our unpartitioned CRF using different settings along two axises summarized in Table II. For the vertical axis (network structure), we use the baseline structure that removes all edges where the CRF model shrinks to Logistic Regression (LR). The explicit structure is obtained by Twitter users' following relationships; the implicit structure is obtained by removing explicit edges where the two users in its ends never retweeted the same author. For the horizontal axis (edge-feature), we use four cases consisting of using a constant feature, the co-similarity feature along, the co-retweet feature along, and both. For each of these settings, we run 10-fold cross validation using varying network sizes. Note that in [25], the best result reported has a precision / recall of 28.8% and 37.3%, respectively. Because they did not report the network size being evaluated, we are not able to compare their results with ours.

Table III
SETTINGS FOR CRF WITH GRAPH PARTITIONING

Graph Partitioning Settings		
	Random-Part	Min-Part
No Edge	*N-Rd*	-
Explicit Edge	*E-Rd*	*E-Min*

The prediction results are presented in Figure 2, where several facts are observed. (1) For network structure, the explicit settings performs equally or better than the implicit settings. It implies that the underlying retweet network cannot be represented by simply putting together users that co-retweeted from the same authors in the past. (2) For edge features, the constant feature outperforms the two intuitive edge-features. It may be the case that the co-retweeting behavior is more related to the users rather than the tweet, or there is just some better but unintuitive features worthy looking for. (3) Under *E-Cons*, our CRF model outperforms the baseline LR and all other settings in all cases. Compared to LR, *E-Cons* brings $2.3\% \sim 9.0\%$ more precision. For recall, the improvement is not as much at up to 3.3%.

Finally, as the network size grows, both precision and recall drop. For a 32-user network, the prediction achieves the precision and recall of 91.9% and 80.9%, respectively. For a 200-user network, however, they quickly drop to 50.3% and 20.8% (not plotted for visual clarity). We also tries to run unpartitioned experiments for networks of more than 200 users. However, the training time grows quadratically and become much slower. Accordingly, we conclude that unpartitioned CRF model is not applicable for larger networks.

C. Partitioned CRF Performance

To improve the poor prediction performance for larger networks, we conduct two experiments with network partitioning. In the first, we fix the network size while varying the subnetwork size; in the second, the contrary. Both experiments are conducted under three settings as in Table III. The *N-Rd* setting considers only local features by eliminating all edges and partitions the network randomly. The *E-Rd* setting considers all explicit edges and partition the network randomly. The *E-Min* setting partition the network according to Algorithm 1.

Under all settings, each partitioned subnetworks are trained using 90% of the tweets that are at least retweeted once by a user within the subnetwork, matched with equal numbers of pure-negative tweets. For testing, we collect all the reserved 10% retweeted tweets from all subnetworks, with randomly selected pure-negative tweets from the original network. These tweets are then tested in all subnetworks. Further, the time for training and testing are measured with all subnetworks running in parallel.

1) Varying the subnetwork size: The prediction and runtime results with fixed total network size (200 users) and varying subnetwork sizes (12, 25, 50, and 100 users) are presented in Figure 3 and Table IV. Note that the result for unpartitioned CRF for 200 users using $E\text{-}Cons$ in the previous experiment is also plotted and illustrated as "NoPart".

There are several observation from the figure. First, all partitioned settings outperform the unpartitioned "NoPart" baseline.

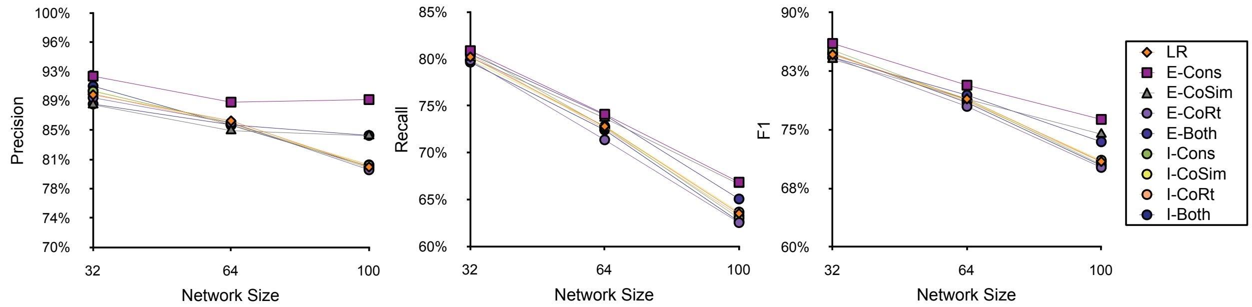

Figure 2. Precision, recall, and F1 score of CRF performance measurements under different settings.

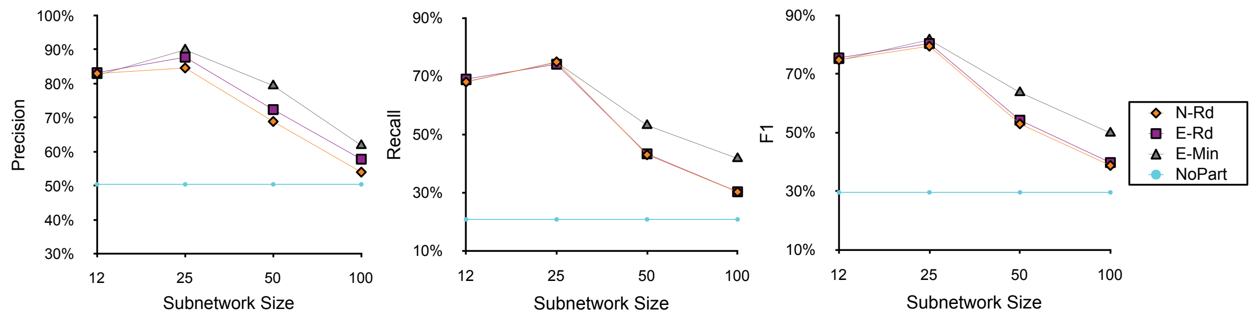

Figure 3. Precision, recall, and F1 score of CRF performance measurements with graph partitioning using common network (200 users) but different subnetwork sizes (12, 25, 64, and 100 users).

Table IV
RUNTIME FOR CRF TRAINING AND MAP WITH GRAPH PARTITIONING USING COMMON NETWORK SIZE (200 USERS) AND VARYING SUBNETWORK SIZES (12, 25, 64, AND 100 USERS)

Training Time (sec.)				
$\lvert U_{sub} \rvert$	12	25	64	100
N-Rd	17.25	61.65	50.38	98.11
E-Rd	18.02	84.08	65.80	208.27
E-Min	21.93	151.51	347.18	257.26
MAP Time (sec.)				
N-Rd	0.04	0.26	0.71	1.57
E-Rd	0.62	0.78	5.13	42.15
E-Min	0.63	6.84	22.29	65.04

Table V
RUNTIME FOR CRF TRAINING AND MAP WITH GRAPH PARTITIONING USING COMMON SUBNETWORK SIZE (25 USERS) AND VARYING TOTAL NETWORK SIZE (200, 500, AND 1000 USERS)

Training Time (sec.)			
$\lvert U \rvert$	200	500	1000
N-Rd	61.65	118.69	148.66
E-Rd	84.08	253.49	378.37
E-Min	151.51	322.74	439.52
MAP Time (sec.)			
N-Rd	0.26	0.53	0.63
E-Rd	0.78	12.21	20.77
E-Min	6.84	14.27	24.22

It clearly shows that partitioning the network into subgraphs can give better prediction performances. Second, *E-Min* outperforms other two settings in all cases by $3.2\% \sim 10.5\%$. It shows that edges do play an important role since *E-Min* reserves as much edges as possible when splitting the network compared to the *N-Rd* and *E-Rd*. Such a difference is also reflected in the training and MAP time in Table IV: *E-Min* takes more time compared to that of the *N-Rd* and *E-Rd* in all cases.

Finally, across different subnetwork sizes, we found the prediction performances peaks at 25-users: it goes down for both larger and smaller subnetworks. For larger subnetworks, it could still due to the dilution problem. For smaller subnetwork of 12-users, however, the predictive benefit brought by incorporating network effects diminishes as the subnetwork is too small to provide enough information. That also implies that 25 may be close to the natural subnetwork size for characterizing a Retweet Network. Since a Twitter user has an average of following relationships at hundreds, this observations suggests that only a relatively small portion of relationships may actually matter when it comes to retweeting behavior, which conforms to the small-world property.

2) Varying the total network size: The prediction and runtime results with fixed subnetwork size (25 users) are varying total network sizes (200, 500, and 1000 users) are presented in Figure 4 and Table V, where two facts are observed. First, between settings, *E-Min* still outperforms both *N-Rd* and *E-Rd* in all cases, by $5.4\% \sim 10.3\%$ for precision, and only in minor for recall, which is very similar to the previous experiment. Second, both the training and MAP times scale sublinearly as the network size goes up. Running in parallel, the training and MAP for the 1000-user network takes less then 8 and 0.5 minutes in average, respectively. This is much more efficient compared to the unpartitioned 200-user case, where the training and MAP takes an average of 33 minutes and 10 minutes, respectively. Finally, the precision and recall remains at 84.6% / 54.4% for 1000 users in this partitioned case, while in the unpartitioned case the results drop to 50.3% / 20.8% for 200-users. Therefore,

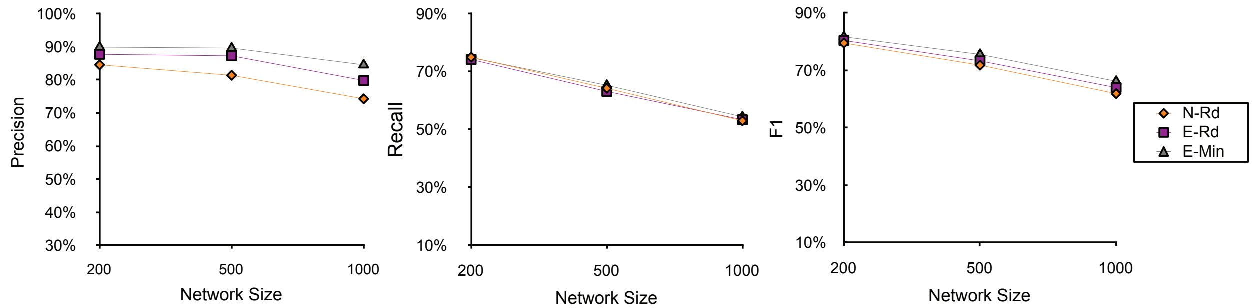

Figure 4. Precision, recall, and F1 score of CRF performance measurements with graph partitioning using common subnetwork (25 users) but different total network sizes (200, 500, and 1000 users).

we conclude that using CRF with graph partitioning is both more effective and efficient for modeling large retweet networks.

D. Illustrative Example

From our last experiment of 1000-user network, we select one tweet to more closely investigate how our model works. This tweet is an e-dictionary advertisement (Twitter ID:145838895-36225280) that was retweeted 43 times globally. Five users within our 1000-user network (Node 530, 816, 845, 855, and 783, respectively) retweeted this tweet. The first four retweeters belongs to the same subgraph according to our partition, and has no following relationships with the last. There also seem to be a difference in the profile of these two groups of retweeters: the first four seem to be prolific internet writers and trend observers, while the last is a software programmer.

Within the 1000 user network, our model predicts exactly these 5 retweeters, where the retweeters and their respective subnetworks is visualized in Figure 5. The coloring of nodes and edges are based on the node- and edge-potentials calculated during MAP, which are defined as:

$$\begin{aligned}\delta_u &= \Phi^u(Y_u = 1) - \Phi^u(Y_u = 0)\\ \delta_{u,v} &= \Phi^{u,v}(Y_u = 1, Y_v = 1) + \Phi^{u,v}(Y_u = 0, Y_v = 0)\\ &- \Phi^{u,v}(Y_u = 1, Y_v = 0) - \Phi^{u,v}(Y_u = 0, Y_v = 1).\end{aligned}$$

A larger δ_u (dark node) correspond to larger predicted retweet probability based the node's local features, whereas a larger $\delta_{u,v}$ (dark edge) correspond to larger predicted co-retweet probability of two nodes to made the same retweet decision. For visual clarity, edges with minor $\delta_{u,v}$'s are eliminated from the figure.

Figure 5 illustrates several interesting perspectives. First, observe how strong local retweetability (dark nodes) does not necessarily lead to actual retweets, i.e., Node 796 in the upper subnetwork and Node 660 and 727 in the lower subnetwork. Based on only their local potentials, they should have been predicted as retweeters, as with any predictions made by a model considering only local features.

Considering network effects, however, we can see these dark nodes also has strong relations (dark edges) with the white nodes (Node 762, 854, 801, 800, etc...) with weakest local potentials. These edges compensate the local effects for the nodes at their both sides. The false tendency to predicting Node 796, 660, and 727 as retweet nodes are therefore corrected. Note that some of

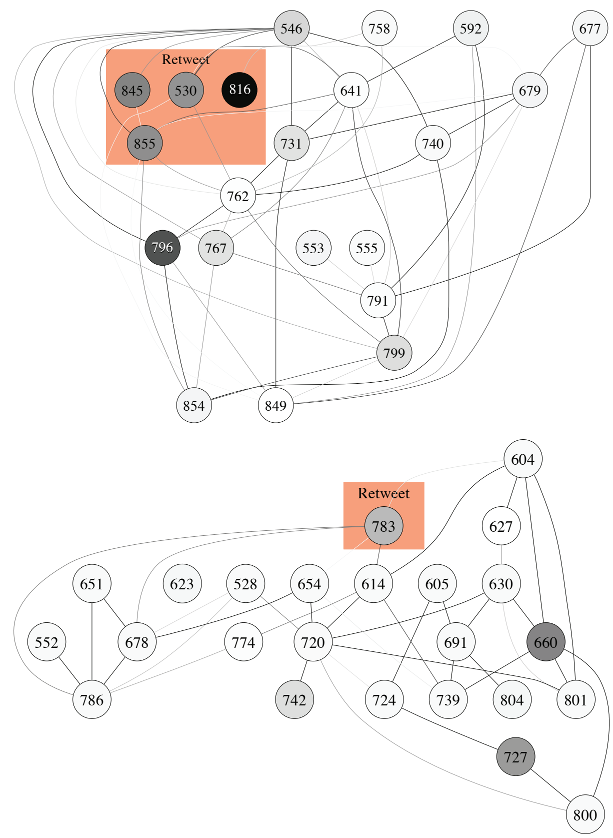

Figure 5. Close investigation of the retweet prediction of a specific tweet

the retweet nodes surrounded by the rectangle also have edges to white nodes (e.g., Node 530, 855, and 783). Their edges, however, are lighter compared to the edges of Node 796, 660, and 727, and are not strong enough to dominate these retweeting nodes' local potentials. Since all edges are characterized using the training data, we can see how a graphical model incorporate network factors and improve overall prediction performances.

Another interesting fact is that the weak edges eliminated from the figures actually account for about 70% of total edges. Although it is just the case for this particular tweet, it may be a signal that more edges from the explicit network can be removed before training to better approximate the intrinsic retweet network. This again relates to the small-world effects, and will make possible better partitioning, shorter runtime, and probably even more accurate prediction results. While this is somewhat out of scope, this work tries to explore in this direction.

VI. Conclusions

We propose using conditional random fields (CRFs) to model and predict the retweet patterns with three types of user-tweet features, i.e., content influence, network influence and temporal decay factor. To improve retweet prediction effectiveness and efficiency, we also investigate partitioning the social graphs and construct appropriate network relations for better CRF modeling. The performance of the proposed algorithms are evaluated by analyzing retweet decisions on 1000 sample users who have complete connection information in a 260K-user Twitter collection. The experimental results show that CRFs can outperform the baseline logistic regression models by a noticeable margin. Our feature analysis suggests that user features, particularly the user-retweet history based features, is the most predictive indicator for retweet modeling. In addition, we show partitioning original social networks into compact subnetworks can significantly reduce the prediction time and improve the detection accuracy. Finally, our insights on illustrative examples suggest that retweeting is jointly impacted by user retweet preference and personal relationship strength.

References

[1] Haiti earthquake: Twitter updates from the disaster zone, 2010.

[2] L. A. Adamic. The Small World Web. *World Wide Web Internet And Web Information Systems*, 2010.

[3] P. Awasthi, A. Gagrani, and B. Ravindran. Image modeling using tree structured conditional random fields. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*, pages 2060–2065, 2007.

[4] R. H. Byrd, J. Nocedal, and R. B. Schnabel. Representations of quasi-Newton matrices and their use in limited memory methods. In *Math Program*, pages 129–156, 1994.

[5] A. Click. Iran's Protests : Why Twitter Is the Medium of the Movement. *Time*, pages 4–7, 2009.

[6] D. Gaffney. # iranElection : Quantifying Online Activism. In *Analysis*, 2010.

[7] W. Galuba and K. Aberer. Outtweeting the Twitterers - Predicting Information Cascades in Microblogs. In *Conference on Online social networks (WOSN)*, 2010.

[8] J. Golbeck, J. M. Grimes, and A. Rogers. Twitter Use by the U . S . Congress. *Journal of the American Society for Information Science*, 61(8):1612–1621, 2010.

[9] S. Golder. Tweet , Tweet , Retweet : Conversational Aspects of Retweeting on Twitter. In *Sciences-New York*, pages 1–10, 2010.

[10] G. Grosseck and C. Holotescu. CAN WE USE TWITTER FOR EDUCATIONAL ACTIVITIES ? In *4th Scientific Conference eLSE "elearning and Software for Education"*, 2008.

[11] B. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal.*, 1970.

[12] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28:1568–1583, October 2006.

[13] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE transactions on pattern analysis and machine intelligence*, 26(2):147–59, Feb. 2004.

[14] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter , a Social Network or a News Media ? Categories and Subject Descriptors. In *International conference on World wide web (WWW)*, 2010.

[15] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*, pages 282–289. Citeseer, 2001.

[16] J. Letierce, A. Passant, S. Decker, and J. G. Breslin. Understanding how Twitter is used to spread scientific messages. In *Web Science Conference*, 2010.

[17] D. A. Shamma and E. F. Churchill. Tweet the Debates Understanding Community Annotation of Uncollected Sources. In *SIGMM workshop on Social media*, pages 3–10, 2009.

[18] S. E. Shimony. Finding maps for belief networks is np-hard. *Artif. Intell.*, 68:399–410, August 1994.

[19] K. C. Sia, C. J, K. Hino, Y. Chi, S. Zhu, and B. L. Tseng. *Monitoring RSS Feeds Based on User Browsing Pattern*, pages 161–168. 2007.

[20] B. Suh, L. Hong, P. Pirolli, and E. H. Chi. Want to be Retweeted ? Large Scale Analytics on Factors Impacting Retweet in Twitter Network. In *IEEE International Conference on Social Computing (SocialCom)*, 2010.

[21] C. Sutton and A. McCallum. An Introduction to Conditional Random Fields for Relational Learning. *Introduction to statistical relational learning*, (x):93, 2007.

[22] S. Vieweg, A. L. Hughes, K. Starbird, and L. Palen. Microblogging During Two Natural Hazards Events : What Twitter May Contribute to Situational Awareness. In *International conference on Human factors in computing systems (CHI)*, pages 1079–1088, 2010.

[23] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. MAP estimation via agreement on trees : Message-passing and linear programming. *IEEE Transactions on Information Theory*, 51:3697–3717, Nov 2005.

[24] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–2, June 1998.

[25] Z. Yang, J. Guo, K. Cai, J. Tang, J. Li, L. Zhang, and Z. Su. Understanding Retweeting Behaviors in Social Networks. In *International Conference on Information and Knowledge Management*, 2010.

[26] J. Yedidia, W. Freeman, and Y. Weiss. Constructing free energy approximations and generalized belief propagation algorithms. *Technical Report TR2004-040*, 2004.

[27] D. Zarrella. The Science of ReTweets, 2009.