Recommendations to Boost Content Spread in Social Networks

Vineet Chaoji[#1] Sayan Ranu[†2] Rajeev Rastogi[#3] Rushi Bhatt[#4]
[†]Dept. of Computer Science
UCSB, Santa Barbara, USA
[#]Yahoo! Labs
Bangalore, India
[2]sayan@cs.ucsb.edu, {[1]chaojv, [3]rrastogi, [4]rushi}@yahoo-inc.com

ABSTRACT

Content sharing in social networks is a powerful mechanism for discovering content on the Internet. The degree to which content is disseminated within the network depends on the connectivity relationships among network nodes. Existing schemes for recommending connections in social networks are based on the number of common neighbors, similarity of user profiles, etc. However, such similarity-based connections do not consider the amount of content discovered.

In this paper, we propose novel algorithms for recommending connections that boost content propagation in a social network without compromising on the relevance of the recommendations. Unlike existing work on influence propagation, in our environment, we are looking for edges instead of nodes, with a bound on the number of incident edges per node. We show that the content spread function is not submodular, and develop approximation algorithms for computing a near-optimal set of edges. Through experiments on real-world social graphs such as Flickr and Twitter, we show that our approximation algorithms achieve content spreads that are as much as 90 times higher compared to existing heuristics for recommending connections.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*information networks*

General Terms

Algorithms,Experimentation

Keywords

content spread, recommendation, social networks

1. INTRODUCTION

Social networks are increasingly becoming a powerful medium for disseminating and discovering useful content. In popular social networking sites like Google+ and Twitter, users share activity updates with their neighbors or followers. The updates typically include recently uploaded photos, comments on photos and news articles, reviews and ratings that the user has assigned to a movie or restaurant, or simply an article or game on the web that the user has liked. Each neighbor recursively shares received updates within its own neighborhood, and content generated by a user propagates through the network to a wide user population. Thus, social networks enable users to share content at an unprecedented scale, and discover new content of interest to them.

On friendship networks such as Facebook, the content spread is confined since connections are typically made to a close group of friends[1]. On the contrary, *content-centric* networks such as Twitter and Google+ promote content spread by allowing users to connect with people having common interests, who are most likely not their friends. The extent to which a social network spreads content is a key metric that impacts both user engagement and network revenues. The more content spreads, the more novel content users end up discovering, and the more value users derive from being part of the social network. The effective dissemination of generated content also helps users build their "online social reputation". For instance, on microblogging sites such as Twitter, the number of active followers is indicative of a user's online reputation [22]. Building an active following is contingent on the content reaching the right set of interested users on Twitter.

From the social network's perspective, higher content spread helps drive up user engagement which in turn leads to improved user retention and audience growth. Furthermore, as users spend more time accessing diverse content in the form of photos, news articles, games etc., there are increased opportunities for monetizing the content via online ads, sale of virtual goods, subscriptions, and so on. As a result of the above benefits, it is crucial for social networks to maximize the dissemination of interesting content across the entire social graph.

One way to boost content spread in a social network is by increasing the connectivity among users. Social networking sites like Twitter and Google+ already offer "people recommendations" to users to increase connectivity. These people recommender implementations, however, focus primarily on making relevant recommendations without an explicit effort towards increasing content availability. For instance, the "People You May Know" feature employs the *Friend-of-Friend* (FoF) algorithm [1] that recommends users based on the number of common friends with the user receiving the recommendation. Other recommender algorithms, for e.g., in Twitter [2, 12], suggest users whose profiles, interests, or updates have substantial overlap with the receiver of the recommendation. In a typical social network, the number of relevant recommendations that qualify based on criteria

WWW 2012, April 16–20, 2012, Lyon, France.
ACM 978-1-4503-1229-5/12/04.

[1]The recently introduced Subscribe Button now allows content to spread beyond friends on Facebook.

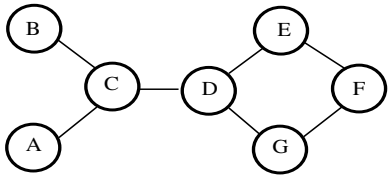


Figure 1: Example illustrating that connecting users with the maximum mutual friends does not maximize content availability.

such as FoF or interest similarity can be significantly large. However, different subsets of these relevant recommendations may have very diverse content spread characteristics. Consequently, simply recommending connections based on the number of mutual friends or similarity between profiles and posted content may not maximize content spread in the social network. This is illustrated in the following example.

EXAMPLE 1. *Consider the simple social network in Figure 1 with users A, ..., G. Suppose that user G generates a piece of content c, and each user (except for E which does not share content) shares content with its neighbors with a probability of* $\frac{1}{2}$. *Then, users D and F each receive c with probability* $\frac{1}{2}$, *E receives c through D or F with probability* $(1-(1-\frac{1}{2}\cdot\frac{1}{2})\cdot(1-\frac{1}{2}\cdot\frac{1}{2}))=\frac{7}{16}$, *C receives with probability* $\frac{1}{4}$, *and A and B receive with probability* $\frac{1}{8}$ *each.*

Let us assume that the set of two-hop neighbors form the set of relevant recommendations for a user. Now, suppose that we are looking to recommend a user to G to connect with. The two relevant candidates for G are C and E. If we use the FoF heuristic, then we would end up recommending user E since it has the maximum number of mutual friends with G. With the additional connection (G, E), it can be shown that content c reaches from G to E with probability $\approx \frac{3}{4}$ *(c reaches E with probability* $\frac{1}{2}$ *along edge (G, E) and with probability* $\frac{1}{4}$ *along the two paths passing through D and F; thus, the probability that c reaches E along one of the paths is* $1-\frac{1}{2}\cdot\frac{3}{4}\cdot\frac{3}{4}\approx\frac{3}{4}$*). Furthermore, there is no change in the probability with which c reaches users C, D and F. Thus, with edge (G, E), the only change in content spread is that c reaches E with* $\frac{3}{4}-\frac{7}{16}=\frac{5}{16}$ *more probability.*

On the other hand, if G connects with C instead, then c reaches C with probability $\frac{5}{8}$ *(c reaches C with probability* $\frac{1}{2}$ *along edge (G, C) and with probability* $\frac{1}{4}$ *along the path through D). Furthermore, c reaches users A and B with probability* $\frac{5}{16}$. *Thus, with edge (G, C), c reaches C with* $\frac{3}{8}$ *more probability, and users B and A each with* $\frac{3}{16}$ *more probability.*

Consequently, even though both E and C are two hops from G, and E has the maximum number of common friends with G, connecting G with C results in content spreading to users with $\frac{3}{4}$ $(=\frac{3}{8}+\frac{3}{16}+\frac{3}{16})$ *higher probability compared to* $\frac{5}{16}$ *as a result of connecting G with E.* □

In this paper, we consider the problem of recommending connections that maximize content spread in a social network while ensuring that the recommendations are relevant. For each user u, our approach first identifies a candidate set N_u of similar users based on the number of common neighbors, proximity in the social graph, similarity of profiles and posted content, etc. It then selects up to k users R_u from each user's candidate set N_u such that if every u connects with users in R_u then the content spread in the network is maximized. The k users in R_u are recommended for connection to user u.

Observe that by recommending a subset of N_u to user u, we ensure that the connections recommended to each user are relevant. In typical social networks, the set N_u can be very large (in the order of hundreds or thousands). By using the content spread objective to select the subset R_u, our recommendation scheme essentially balances the needs of both users (by recommending relevant users) and the social network provider (by boosting content spread in the network). In addition, the constraint k on the maximum number of new connections per user prevents deluging active or highly connected users with an unreasonable number of recommendations. Presenting each user with a bounded number k of relevant connections ensures a good user experience. For the same reason, social networking sites such as Twitter typically limit the number of recommendations to about 10.

Our work differs from prior research on influence maximization in social networks [14]. The objective in [14] is to select the top-k influential nodes in the network that can be targeted to maximize influence spread in the network. The authors show that even though influence maximization is NP-hard, the influence spread function on nodes is *submodular*, and thus a greedy strategy yields influence spreads that are within $(1-\frac{1}{e})$ of the optimum. In contrast, our content spread maximization problem looks to add up to k new connections per node so that content spread is maximized. As edges are added to the social network, its structure itself changes, and so the content spread function on edges is no longer submodular – this precludes simple greedy solutions. Furthermore, in our problem setting, there are complex constraints requiring that the number of new edges incident on any node of the social graph is at most k. Thus, we have millions of local constraints on selected edges as opposed to a single global constraint in [14] on the number of selected nodes. The lack of submodularity coupled with complex local constraints make our content spread maximization problem a lot more challenging.

To summarize, our main contributions are as follows:

• We formally define the content maximization problem that seeks to add up to k connections per user such that the (probabilistic) propagation of content in the social network is maximized. To the best of our knowledge, the people recommendation problem with the explicit goal of maximizing content availability in the social network is new.

• We show that our content maximization problem is NP-hard. Moreover, our content spread function lacks desirable properties like submodularity that allow for efficient approximation algorithms. We propose a restricted variant that is submodular and closely approximates our original content spread function.

• For our restricted content spread function, we devise an approximation algorithm that computes an edge set satisfying constraints and whose content spread is provably close to the optimum.

• We conduct an extensive experimental study with real-life social networks from Twitter, Flickr, etc. In our experiments, the connections recommended by our algorithm achieve content spreads that are as much as 90 times higher compared to the FoF heuristic and 4 times more than simple greedy strategies.

2. THE CONTENT MAXIMIZATION PROBLEM

We model the social network as an undirected graph $G = (V, E)$ where nodes represent users and edges are the connections between them. Furthermore, we denote the pieces of content (e.g., photos, comments, articles) that nodes share with their neighbors over a fixed time period (e.g., a month) by C. Each node i in the graph has the following three parameters: (1) p_i, the probability with which node i shares content *independently* with each of its neighbors, (2) $c_i \subseteq C$, the content generated or discovered by node i, and (3) N_i, the set of nodes in G that is compatible with node i. The parameter p_i can be empirically estimated by observing the fraction of content that a node shares with its neighbors. Also, $N_i = \{j : sim(i,j) > \alpha \wedge j \in V\}$. Here, $sim(i,j)$ is the similarity between nodes i and j computed based on the number of hops between the nodes, the number of common neighbors, node profiles (e.g., preferences, educational background), and posted content. The user-defined parameter α ensures that nodes in N_i are fairly similar to i, and are thus relevant candidates for recommendation to i.

Observe that the parameters c_i and p_i determine the flow of content through the network. We define the *content spread* within the network as: $\sum_c$ *Expected number of nodes with content* c. Our objective in this paper is to compute a set of relevant recommendations X such that the content spread is maximized. Each recommendation in X is a node pair (i,j), and indicates that node i is recommended to j, and vice versa. Now, suppose that $P_X(i,c)$ is the probability of content c reaching node i over the edge set $E \cup X$. Then, the expected number of nodes with content c is given by $\sum_i P_X(i,c)$, and the content spread with new edges X is $f(X) = \sum_c \sum_i P_X(i,c)$. We formally define our content maximization problem below.

Definition 1. Content Maximization Problem: *Given a graph $G = (V, E)$ and a constant k, find an edge set $X \subseteq \{(i,j) : i,j \in V\}$ such that: (1) At most k edges from X are incident on any node in V, (2) For each $(i,j) \in X$, $i \in N_j$ and $j \in N_i$, and (3) $f(X)$ is maximum.* □

The term $P_X(i,c)$ within the content spread expression $f(X)$ depends on the specific content propagation model [10, 9, 14, 4] employed. A popular model for the spread of information or influence through a social network is the *Independent Cascade* (IC) model [14]. In this model, when a node receives or generates a new piece of content c (that it has not seen before) at step t, it shares the content with its neighbors in the subsequent step $t+1$. Thus, each node shares specific content with its neighbors only once.

Our content spread function $f(\cdot)$ under the IC model has two main drawbacks. First, computing the expected number of nodes with specific content c is #P-hard [4], and so accurately estimating the content spread requires running expensive simulations for a large number of times. To overcome the high computation cost, Chen et al. [4] propose an efficient heuristic that restricts *influence propagation* between a pair of nodes to be only along the *maximum probability path* (MPP) between the nodes. This propagation model can also be applied to our setting, thus allowing the content spread function $f(\cdot)$ to be efficiently computed. We will refer to this model as the MPP model in this paper. Interestingly, even though the MPP model is more restrictive compared to the IC model, Chen et al. [4] empirically show that the MPP model closely matches the IC model in terms of the influence spread.

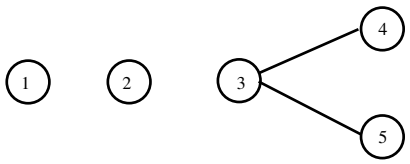


Figure 2: Example illustrating that $f(\cdot)$ is not submodular under the MPP model.

Unfortunately, the set cover problem can be reduced to our content maximization problem resulting in the following theorem.

Theorem 1. *The content maximization problem is NP-hard under both the IC and MPP models.*

Proof. Follows from a reduction of the *Set Cover* problem to the content maximization problem. Details omitted due to space constraints. □

The second drawback is that under both the IC and MPP models, $f(\cdot)$ lacks properties that would allow us to devise good approximation algorithms. One such property is *submodularity* – a function h on subsets of edges is submodular if $h(S \cup \{e\}) - h(S) \geq h(T \cup \{e\}) - h(T)$ for all edges e and all pairs of edge subsets $S \subseteq T$. Kempe et al. [14] and Chen et al. [4] show that *influence spread* is submodular under the IC and MPP models, thus enabling a simple greedy strategy to yield a solution that is within a factor of $(1 - \frac{1}{e})$ of the optimum. However, our content spread function $f(\cdot)$ defined on edges is very different from influence spread which is defined on nodes. Specifically, when computing the content spread $f(X)$, the new edge set X gets added to the underlying graph G and this changes the structure of G. In the following example, we show that $f(\cdot)$ is not submodular under the IC and MPP models.

Example 2. *Consider the social network graph with 5 nodes and 2 edges depicted in Figure 2. Let each node have propagation probability 1 and let only node 1 contain a single piece of content c. Let $S = \emptyset$, $T = \{(2,3)\}$ and $e = (1,2)$. Now $f(S \cup \{e\}) - f(S) = 2 - 1 = 1$ since with edge e, content c from node 1 reaches node 2 with probability 1. On the other hand, $f(T \cup \{e\}) - f(T) = 5 - 1 = 4$ since with edges $(1,2)$ and $(2,3)$, content c from node 1 reaches every node j with probability 1 along the unique path from 1 to j. Thus, since $f(S \cup \{e\}) - f(S) < f(T \cup \{e\}) - f(T)$ for $S \subseteq T$, $f(\cdot)$ is not submodular.* □

In the next section, we propose a restricted content propagation model that closely approximates the *MPP* model but in which content spread $f(\cdot)$ is submodular. This allows us to develop efficient approximation algorithms for content maximization. Although, in our problem setting, we have more complex constraints that require the number of edges incident on each node to be no more than k. The constraints preclude simple greedy approaches, and necessitate more involved approximation algorithms.

3. A SUBMODULAR CONTENT SPREAD FUNCTION

We now present our new content propagation model in which content spread is both submodular and efficient to compute.

3.1 Restricted Maximum Probability Path Model

For a path $\langle i = u_1, u_2, \ldots, u_r = j\rangle$ through nodes $u_1, u_2, \ldots, u_r$, we define the propagation probability of the path as $p_{u_1} \cdot p_{u_2} \cdots p_{u_{r-1}}$. This is essentially the probability that content from node i reaches j along the path. Our new model transmits content along paths that are restrictions of MPPs – we define these paths below.

Definition 2. Restricted Maximum Probability Path (RMPP): *For edge set X, the restricted maximum probability path $RMPP_X(i,j)$ from node i to node j is the path with the maximum probability among all paths from i to j containing at most one edge from X. Ties are broken arbitrarily.* □

In our new RMPP model, content from node i flows to node j only along the path $RMPP_X(i,j)$. These RMPPs between nodes are used to compute the content spread $f(X)$ for an edge set X. Thus, the RMPP model restricts content propagation paths to contain at most one edge from X, and this ensures submodularity of content spreads. In Section 5, we show that considering these restricted propagation paths has little effect on content spread values; this is because a bulk of the probability mass is concentrated along such paths. Note that RMPPs can be efficiently computed using a variant of Dijkstra's shortest path algorithm; omitted here due to space considerations.

To illustrate the submodularity property in the RMPP model, let us revisit the social graph in Figure 2. Assume that all nodes have propagation probabilities of 1 and content c is only at node 1. Let us compute the content spread for edge sets $S = \emptyset$, $T = \{(2,3)\}$ and $e = (1,2)$ in the RMPP model. The content spread is 1 for edge sets S and T since node 1 is completely disconnected from the rest of the graph. For edge set $S \cup \{e\}$, the content spread is 2 because content c reaches node 2 with probability 1 along path $\langle 1,2\rangle$. The content spread for edge set $T \cup \{e\}$ is also 2 because the content c from node 1 can only reach node 2. It cannot reach other nodes since this would require the content to traverse a path containing both edges in $T \cup \{e\}$ which is not allowed. Thus, $f(S \cup \{e\}) - f(S) = f(T \cup \{e\}) - f(T)$ in the RMPP model. This is in contrast to the MPP model under which $f(S \cup \{e\}) - f(S) < f(T \cup \{e\}) - f(T)$ (see Example 2).

In addition to submodularity, the content spread in the RMPP model can also be efficiently computed. We make the following two simplifying assumptions to speed up computation and design effective algorithms: (1) Similar to the MPP model in [4], we use a threshold θ to prune paths with too small propagation probabilities, and (2) We assume that content propagates along each path independent of other paths. Note that the MPP model in [4] does not assume path independence. In Section 5, we show that the path independence assumption minimally impacts the computed content spread values. Thus, the content spreads for the RMPP and MPP models are very close.

Now, for our RMPP model, we can compute the probability $P_X(i,c)$ of content c getting to node i for a new edge set X as follows. Let $V(c)$ denote the nodes containing content c. Further, for $j \in V(c)$, let $q_X(j,i)$ be the probability of the path $RMPP_X(j,i)$ from j to i if it is above threshold θ. On the other hand, if the probability of path $RMPP_X(j,i)$ is less than θ, then $q_X(j,i) = 0$. Since the propagation of content to node i along the individual paths $RMPP_X(j,i)$ are independent, we get that $P_X(i,c) = 1 - \prod_{j \in V(c)}(1 - q_X(j,i))$. Thus, the content spread function in the RMPP model is given by:

$$f(X) = \sum_c \sum_i P_X(i,c) = \sum_c \sum_i (1 - \prod_{j \in V(c)} (1 - q_X(j,i))) \quad (1)$$

Our content maximization problem in the RMPP model is then to find an edge set X in graph G that maximizes the content spread $f(X)$ in Equation (1) above subject to constraints (1) and (2) in Definition 1. This problem can also be shown to be NP-hard using a reduction similar to the one used for content maximization under MPP in Theorem 1 earlier. The following example illustrates content spread computation under the RMPP model.

Example 3. *Consider the social graph in Figure 2. Let the propagation probabilities for all nodes be $\frac{1}{2}$. Furthermore, let node 1 contain content c, and nodes 4 and 5 contain content c'. Finally, let $X = \{(1,2),(2,3)\}$. Now, $P_X(2,c) = p_1 = \frac{1}{2}$ since c can flow from 1 to 2 along path (1, 2). However, $P_X(j,c) = 0$ for $j \geq 3$ since there is no path from 1 to j containing at most one edge from X. Content c' can reach node 3 from 4 and 5 along two paths $\langle 4,3\rangle$ and $\langle 5,3\rangle$, respectively. Thus, $P_X(3,c') = 1-(1-p_4)(1-p_5) = \frac{3}{4}$. Similarly, since content c' can reach 2 along paths $\langle 4,3,2\rangle$ and $\langle 5,3,2\rangle$, we get $P_X(2,c') = 1-(1-p_4 \cdot p_3)(1-p_5 \cdot p_3) = \frac{7}{16}$. However, content c' cannot reach node 1 because paths to 1 from 4 and 5 involve two edges from X. Thus, content spread $f(X) = \sum_i P_X(i,c) + \sum_i P_X(i,c') = (1+\frac{1}{2}) + (2+\frac{3}{4}+\frac{7}{16}) = 4.6875$.*

Observe that in our derivation of $P_X(2,c') = \frac{7}{16}$ above, we assumed that the paths $\langle 4,3,2\rangle$ and $\langle 5,3,2\rangle$ are independent. Without the path independence assumption, we would have obtained $P_X(2,c') = P_X(3,c') \cdot p_3 = \frac{3}{4} \cdot \frac{1}{2} = \frac{3}{8}$. Thus, the $P_X(2,c')$ values with and without path independence are fairly close. □

It is straightforward to see that $f(\cdot)$ is monotonic. The following theorem proves submodularity.

Theorem 2. *The content spread function $f(X)$ under the RMPP model is submodular.*

Proof. See Appendix A. □

4. APPROXIMATION ALGORITHM FOR CONTENT MAXIMIZATION

We are now ready to present our approximation algorithm for the content maximization problem in the RMPP model. Let $Z = \{e_1, e_2, \ldots, e_m\}$ be the set of edges between similar nodes in V corresponding to compatible users. We are looking for a set $X \subseteq Z$ such that at most k edges from X are incident on any node and $f(X)$ as defined in Equation (1) is maximum.

Since $f(\cdot)$ is submodular, one option is to use a simple greedy strategy that (in each step) selects the edge that provides the maximum marginal increase in function value. However, this does not give good approximation bounds because of the constraint k on the number of edges incident on a node. Specifically, an edge that results in the maximum increase in content spread might violate node incidence constraints and thus be ineligible for selection. Consequently, to handle these feasibility constraints, we adopt a different approach that considers a continuous relaxation of our problem, and computes a fractional (approximate) solution for edge membership in set X using the algorithm of [23]. We

then use randomized rounding to convert our fractional solution into an integral solution, and incur a constant factor reduction in the benefit due to rounding.

Continuous relaxation. Let $\bar{y} = (y_1, \ldots, y_m)$ be an m-dimensional vector of variables $y_i \in [0, 1]$. The semantics here are that edge $e_i \in X$ with probability y_i. We define $F(\cdot)$ to be the following continuous extension of $f(\cdot)$. Let $X \subseteq Z$ be a random variable such that $e_i \in X$ with probability y_i. Then,

$$F(\bar{y}) = \mathbf{E}[f(X)] = \sum_X f(X) \prod_{e_i \in X} y_i \prod_{e_i \notin X} (1 - y_i) \quad (2)$$

The continuous relaxation of our content maximization problem then is to find $\bar{y}$ such that $F(\bar{y})$ is maximized with

$$\sum_{j \in e_i} y_i \leq k \text{ for all } j \in V \quad (3)$$

$$y_i \in [0, 1] \quad (4)$$

Note that Equation (3) enforces the constraint that each node j has at most k incident edges in the discrete case. Now, let $F(\bar{y}_{opt})$ be the maximum value of $F(\cdot)$ subject to the constraints, and X_{opt} be the edge set satisfying constraints for which $f(X_{opt})$ is maximum. Also, let $\bar{z}$ be defined as follows: $z_i = 1$ if $e_i \in X_{opt}$, and 0 otherwise. Then, observe that $F(\bar{z}) = f(X_{opt})$, and $\bar{z}$ is feasible. Thus, we have that $F(\bar{y}_{opt}) \geq F(\bar{z}) = f(X_{opt})$.

Algorithm 1: ContinuousGreedy

Input: Graph $G = (V, E)$, candidate edge set Z;
Output: $\bar{y}$ satisfying Equation (3) and $F(\bar{y}) \geq (1 - \frac{1}{e}) \cdot f(X_{opt})$;

```
ȳ = 0; l = 0;
while l < δ do
    Generate r samples X_1, X_2, ..., X_r, where e_i ∈ X_j with
    probability y_i. Set w_i = (Σ_j f(X_j ∪ e_i) − f(X_j)) / r.
    Compute a subset of edges Y such that no node has
    more than k incident edges and Σ_{e_i ∈ Y} w_i is maximum.
    This is an instance of the graph matching problem and
    can be solved using the algorithm of [13] in O(m^3) steps;
    foreach e_i ∈ Y do y_i = y_i + 1/δ;
    l = l + 1;
return ȳ;
```

Continuous greedy algorithm. We use the continuous greedy algorithm of Vondrak [23] (see Algorithm 1) to find a $\bar{y}$ satisfying Equation (3) above such that $F(\bar{y}) \geq (1 - \frac{1}{e}) \cdot F(\bar{y}_{opt}) \geq (1 - \frac{1}{e}) \cdot f(X_{opt})$. Algorithm 1 considers δ intervals of width $1/\delta$, and in each iteration, it increments y_i values of edges e_i in a feasible edge set Y with the maximum sum of *gradients* $\sum_{e_i \in Y} \frac{\partial F}{\partial y_i}$. Each gradient $\frac{\partial F}{\partial y_i}$ can be approximated as $\mathbf{E}[f(X \cup e_i) - f(X)]$ which is estimated by averaging over r samples X_j. The graph matching algorithm of [13] is then used to compute the optimal set Y with at most k edges per node and the maximum sum of gradient estimates. Note that since the y_i values of only edges $e_i \in Y$ are incremented by $1/\delta$ in each iteration, it follows that the final $\bar{y}$ satisfies Equation (3). In fact, [23] proves the following theorem.

Theorem 3. *[23] For $\delta = m^2$ and $r = m^5$, Algorithm 1 returns $\bar{y}$ satisfying Equation (3) and $F(\bar{y}) \geq (1 - \frac{1}{e}) \cdot f(X_{opt})$.* □

Randomized rounding procedure. Once we have computed $\bar{y}$ satisfying $\sum_{j \in e_i} y_i \leq k$ for all $j \in V$ and $F(\bar{y}) \geq (1 - \frac{1}{e}) \cdot f(X_{opt})$, we use randomized rounding [23] to compute the final set X of edges. Essentially, we add element e_i to X with probability y_i. Note that $\mathbf{E}[f(X)] = F(\bar{y})$ and so $\mathbf{E}[f(X)] \geq (1 - \frac{1}{e}) \cdot f(X_{opt})$. However, the result of rounding X may no longer be feasible, that is, the number of edges in X that are incident on a node j may exceed k. So we need to delete edges from X to ensure that it is feasible – we do this by partitioning X into a small number of feasible sets X_i and returning the X_i for which $f(X_i)$ is maximum.

Our partitioning scheme starts with $X_1 = X$ and for each node j with $k' > k$ incident edges in X_1, it deletes (an arbitrary set of) $k' - k$ edges incident on j from X_1 and inserts them into a new (overflow) set X_2. Thus, X_1 now becomes feasible, and the procedure is repeated for $X_2, \ldots, X_s$ until we get an overflow set X_s that is feasible.

Analysis of Approximation Algorithm. We can show the following approximation guarantee for our algorithm.

Theorem 4. *Let $|V| = n$, $\delta = m^2$ and $r = m^5$. Further, let our partitioning scheme generate edge sets $X_1, \ldots, X_s$. Then w.h.p. $\mathbf{E}[\max_i f(X_i)] \geq \frac{1}{3+2\epsilon} \cdot (1 - \frac{1}{e}) \cdot f(X_{opt})$, where $\epsilon = \sqrt{\frac{8}{k} \log(n)}$.*

Proof. See Appendix B. □

Note that Theorem 4 provides worst-case bounds. In practice, our experimental results indicate that our approximation algorithm returns edge sets with good content spreads for much smaller values of parameters δ (set to 2000) and r (set to 30). The time complexity of our approximation algorithm is dominated by the matching procedure in Step 3 of Algorithm 1. The matching algorithm has time complexity $O(m^3)$ and is run δ times; so the overall time complexity of our approximation algorithm is $O(m^3 \cdot \delta)$. To overcome the computation cost, for large m, we can cluster the edges in Z and run our recommendation algorithm on the smaller clusters. We can also achieve further speedup using approximate matching based on greedy heuristics instead of exact matching. In our experiments in Section 6, computing recommendations on a one million node Twitter graph took a little over 11 hours on a stand-alone PC.

5. DISCUSSION

In this section, we present intuitive arguments to show that the three models – MPP with path dependence assumption [4], MPP with path independence assumption, and RMPP – result in similar content spreads for realistic graphs.

5.1 Closeness of Content Spreads in the Dependent and Independent Path MPP Models

To show that the path independence assumption does not affect content spread values significantly, we compute $P_X(i, c)$ for a node i with and without the path independence assumption in the MPP model. For simplicity, let the propagation probability of all nodes be p. Furthermore, let the paths (over the edge set $E \cup X$) that carry content c to node i form a tree of depth $l = \lfloor \log_p \theta \rfloor$ and degree d at each node (thus, there are d^l paths). Recall (from Section 3.1)

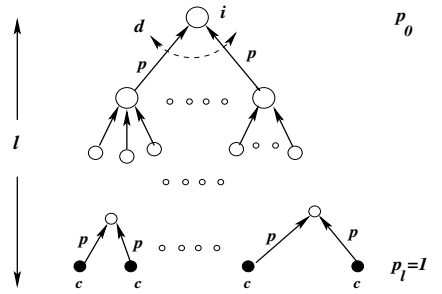


Figure 3: Propagation tree of depth l for carrying c from content nodes to node i.

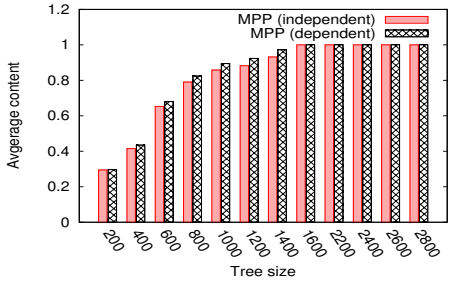


Figure 4: Histogram of average content at a node in the Twitter graph under the dependent and independent path MPP models.

that θ is the threshold to prune paths with low propagation probabilities. This is pictorially depicted in Figure 3. In general, the degree d of a node in the propagation tree is much smaller compared to its degree in the graph G due to the limited number of sources with content c. Thus, since propagation probability is typically small ($\tilde{0}.05$), $p \cdot d \ll 1$.

Under the independent path assumption, $P_X(i,c) = 1 - (1-p^l)^{d^l} \approx p^l \cdot d^l$. Now, let us compute $P_X(i,c)$ considering dependencies between paths as in [4]. Let p_x be the probability of content reaching an intermediate node in the content propagation tree at depth x. Then, the probability of content reaching a node at depth $(x-1)$ can be recursively computed as $p_{x-1} = 1-(1-p_x \cdot p)^d$. Since the probability of content at the leaf nodes $p_l = 1$, we get $p_{l-1} = 1-(1-p)^d \approx p \cdot d$. Similarly, $p_{l-2} = 1-(1-p \cdot d \cdot p)^d \approx p^2 \cdot d^2$. Computing recursively, the probability of content reaching the root (node i) $p_0 = P_X(i,c) \approx p^l \cdot d^l$. Thus, the content spread with and without path independence is approximately the same.

Figure 4 empirically compares the average content at a node in a one million node Twitter graph (described in Section 6.1) under the independent and dependent path assumptions. Content c is randomly assigned to 1% of the nodes in the graph which in turn spreads through paths of the propagation trees. The propagation trees are grouped into bins (x-axis) based on the number of nodes in the tree. The average content at root nodes in a bin is shown on the y-axis. As can be seen, the content spreads with and without the path independence assumption are very close.

The path independence assumption in our setting is essential for submodularity in Theorem 2. In fact, it can be shown that the submodularity property does not hold for the dependent path MPP model in [4].

5.2 Closeness of Content Spreads in the RMPP and Independent Path MPP Models

Another natural question arises – what is lost when the restriction of at most one new edge per path is imposed in the RMPP model? For a simplified yet representative setting, we present informal arguments to show that the content spread along paths containing at most one edge from X (RMPP model) matches the spread along paths containing an arbitrary number of edges from X (MPP model).

Our arguments rely on two basic observations: (1) In social networks, the average degree of nodes is typically much larger than the number of social recommendations k per node, and (2) With each additional hop, the probability of content traversing a path decreases by a factor equal to the propagation probability of the hop. Thus, shorter paths tend to have higher probabilities, and so content is more likely to spread along shorter paths compared to longer paths.

We will illustrate this in the context of a simple scenario where a single node i contains a piece of content c and we trace the spread of c from i along different types of maximum probability paths. Consider the tree T of maximum probability paths over edges in $E \cup X$ originating from i. To keep our analysis simple, let us assume that each node in T has $h+k$ children connected to it by h edges from E and k edges from X. Also, let us assume that each node shares content with its neighbors with probability p. Since we ignore paths with probability less than θ, the depth of tree T is at most $l = \lfloor \log_p \theta \rfloor$. See Figure 5 for an illustration.

Now, it is easy to see that the number of paths in T originating at root i of length r with zero edges from X is h^r. Similarly, the number of paths in T of length r with a single edge from X is $r \cdot h^{r-1} \cdot k$. The reason for the factor r is that the single edge from X can occur in one of r positions, and the factor k is there because each node has k incident edges from X. Also, the total number of paths starting at root i of length r in T is $(h+k)^r$. Suppose that P_0, P_1 and P_∞ are the content spread values from node i along paths with at most zero, one, and unlimited edges from X, respectively. P_∞ is essentially the total probability of all the paths, and so we get that $P_\infty = \sum_{r=1}^{l} (h+k)^r \cdot p^r$. Similarly, P_1 is the probability of paths with at the most one edge from X, and so $P_1 = \sum_{r=1}^{l} h^r \cdot p^r + \sum_{r=1}^{l} r \cdot h^{r-1} \cdot k \cdot p^r$. And finally, $P_0 = \sum_{r=1}^{l} h^r \cdot p^r$.

Typical values of parameters are $h = 100$, $k = 10$, $p = 0.05$, and $l = 3$. For these values, $P_0 = 155$, $P_1 = 198$ and $P_\infty = 202.125$. Thus, the content spread along paths with at most one edge from X is very close to the content spread along paths with no restrictions on the number of edges from X. In fact, if we only consider paths containing at least one edge from X, then the bulk of the probability mass is concentrated in paths with exactly one edge from X. This is because the probability of paths with exactly one edge from X is $P_1 - P_0 = 43$ while the probability of paths with more than one edge from X is $P_\infty - P_1 = 4.125$.

Emperical results comparing content spreads in the RMPP and the MPP models are presented in Section 6.3.

6. SIMULATION EXPERIMENTS

Through simulations on real-life social network data, we show the superior performance of the continuous greedy algorithm compared to other popular approaches – simple greedy, degree-based heuristics, and Friend-of-Friend (FoF) based selection. We also (empirically) substantiate our claim about the closeness of content maximization under the RMPP and MPP models.

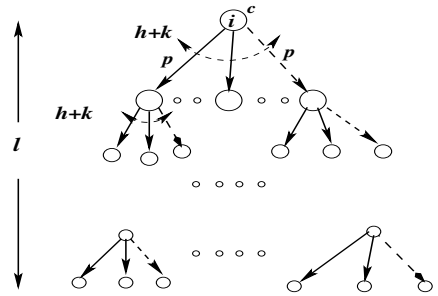


Figure 5: Propagation tree for spread of content c from node i along h (solid) edges from E and k (dashed) edges from X.

Table 1: Statistics for the datasets. (SCC: Strongly Connected Components, WCC: Weakly Connected Components).

	Wikipedia	Flickr	Epinions	Twitter
#Nodes	7.1 K	81 K	76 K	1 M
#Edges	104 K	5.9 M	508 K	3088467
Average Degree	14.6	72.8	6.7	3.1
Maximum Degree	893	4714	1801	216
# WCC	24	1	1	15
# SCC	5.9 K	1	42 K	896749
Largest SCC size	1.3 K	81K	3.2 K	103113
Avg. SCC Size	1.22	81K	1.79	1.12
# groups	NA	195	NA	27

6.1 Experimental Setup

To simulate content dissemination, we need realistic graph datasets, and models for content generation and propagation.

Datasets: We use the Wikipedia, Flickr, Epinions, and Twitter social graphs. Between them, these graphs capture a wide variety of social relations (e.g., trust relations, follower-following relations, and friendship relations). Important statistics of these graphs are summarized in Table 1.

Twitter: This dataset is obtained by crawling the twitter.com site starting from a randomly chosen set of popular personalities on Twitter. A directed edge from node X to Y indicates that X is following Y's tweets. Due to Twitter's limit on the number of web server requests, only a subset of users followed by X could be obtained. Based on the content of their tweets, users were assigned to zero or more groups from a set of 27 predefined groups (e.g., politics and sports).

Flickr: The Flickr (undirected) graph [24] consists of friendship relations between users on the popular image site flickr.com. In addition to the friendship connections, each user belongs to one or more Flickr groups (e.g., wildlife and nature) from a set of 195 groups.

Wikipedia: The Wikipedia(directed) graph [16] is generated using the voting activity in elections for granting administrator rights to Wikipedia users. Each node in the graph represents a Wikipedia user with voting rights. A directed edge from i to j denotes user i's vote for user j.

Epinions: Directed social network captures the who-trusts-whom relation on the consumer reviews site epinions.com.

We consider the similarity $sim(i,j) = 1$ if the pair of users (i,j) shares a common group and $sim(i,j) = 0$ otherwise. The similarity function determines the candidate set N_u of nodes with which a user u can connect. In the absence of groups in a dataset, an edge can be added between any pair of nodes in the network.

Content generation model: In the absence of information related to content at the nodes, we make the following assumptions for simulation purposes – our algorithm is independent of the exact values assumed here. We assume that a single content type c is generated by a *set of seed nodes*, S. Moreover, the rate of content generation is assumed to be uniform across all nodes in S. Unless specified otherwise, $|S|$ is 1% of the node set size. The set S is selected randomly for all datasets other than Twitter. For Twitter, S consists of about 38K users who tweet about *soccer*. To maintain parity, the same set of randomly generated seed nodes is used for all the algorithms.

Propagation model: Since we do not have the data to infer propagation probabilities, we instead consider two propagation probability assignments that are simple, yet illustrative. In the uniform assignment (henceforth *UNI*), all nodes have the same probability p of sharing content. In the weighted (henceforth *WT*) assignment, inspired by the *weighted* cascade model [15], pairwise propagation probabilities are inversely proportional to the degree of the originating node. These assignments have been considered elsewhere for studying the MPP model [4]. We use the setting $p = 0.05$, unless stated otherwise.

6.1.1 Performance Evaluation

To compare the different edge selection methods, we define the *lift* metric as follows:

$$Lift(X) = \frac{f(E \cup X) - f(E)}{f(E)} \times 100, \quad (5)$$

where E is the set of edges in the original graph G and $f(.)$ is the content spread function. X is the set of recommendations computed by the following edge selection methods. Since the initial set S of nodes where content originates is determined randomly, all the *Lift* results reported here are averages over 10 independent runs.

In this paper, we do not focus on evaluating the quality of the recommendations, since the candidate set N_i selected using traditional recommendation methods is itself likely to be highly relevant.

6.2 Edge Selection Algorithms

We compared the following edge selection strategies.

Greedy, where edges with the largest *lift*, given the current set of edges in the graph, are added one at a time. This process continues until no further edges can be added as a result of the maximum edge constraint for each node.

Continuous Greedy (CG), where edges are added to the original graph based on Algorithm 1 followed by randomized rounding. Unless stated otherwise, we use $\delta = 2000$ and $r = 30$ as parameters (see Section 4 for parameter description). Threshold θ for pruning paths is set to 0.01.

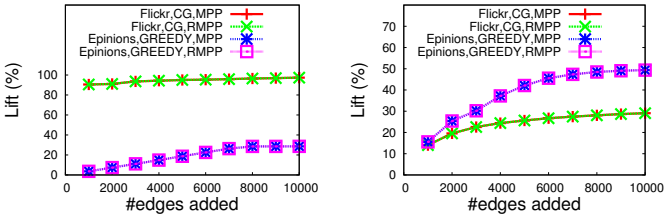
Degree based selection adds edges between high degree node pairs, wherein one of the nodes has content c. This heuristic is intuitively competitive because it exploits the high degree of nodes to maximize content spread.

Friend-of-Friend (FoF) based selection, where node pairs are ranked by the number of common neighbors. Edges are added between unconnected node pairs in this rank order.

In our experimental setup, k is set to 10. In principle, all of the above algorithms terminate when the maximum recommendations limit k is reached for all nodes in the graph. To highlight key insights, we instead terminate the simulations after a certain fixed number of new edges are added.

All simulations were performed on a 64-bit Intel Xeon 2.5GHz processor with 32 GB of main memory.

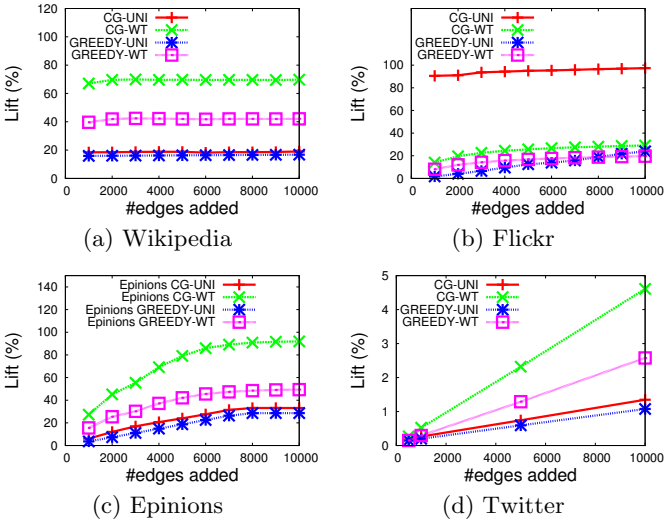
6.3 Simulation Results



(a) UNI propagation (b) WT propagation

Figure 6: Lift for $RMPP$ and MPP models under uniform and weighted propagation.

Comparison between $RMPP$ and independent path MPP. For the Epinions and Flickr datasets, Figures 6(a) and 6(b) show the percentage lift as edges are added to the initial graph under the *UNI* and WT probability assignment models, respectively. In both models, the MPP and $RMPP$ lifts are almost identical. However, the $RMPP$ and MPP propagation models tend to deviate minimally under the WT probability assignment since WT has higher likelihood of longer paths in graphs with low-degree nodes. Since long paths have a higher chance of more than one new edge, we would expect MPP and $RMPP$ to diverge in such situations. Subsequent results consider the $RMPP$ model alone.



(a) Wikipedia (b) Flickr

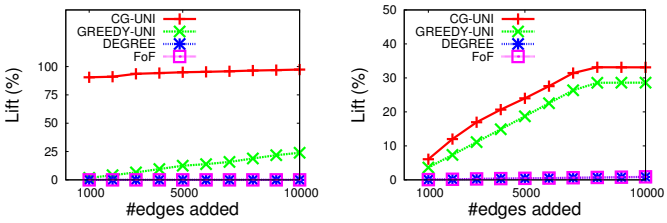
(c) Epinions (d) Twitter

Figure 7: Lift as a function of number of edges added to the initial graph.

Greedy versus Continuous Greedy (CG) maximization. The *Greedy* approach selects edges with the highest immediate lift. On the contrary, *CG* takes an approach that captures the correlation between edges with high marginal gain. The iterative computation in *CG* has the following benefits: (1) Each iteration selects a locally optimal edge set (to maximize sum of gradients), and (2) Each iteration takes into account edge sets selected in previous iterations, thus capturing the interplay between edges.

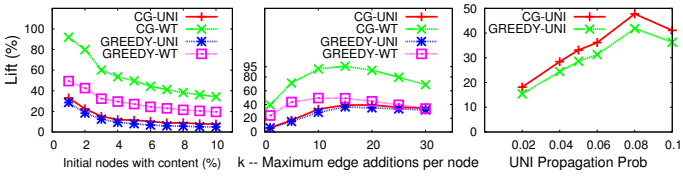
Figure 7 shows the lift in content propagation for 10K edges added to the initial graphs. The combination of continuous greedy with weighted propagation outperforms other algorithms by a factor of 1.75 to 2 for all datasets other than Flickr. For Flickr, *CG-UNI* outperforms other algorithms by as much as a factor of 6. The lower lift for *CG-WT* on the dense Flickr graph can be attributed to the propagation probability, which is inversely proportional to node degree in the WT assignment. In contrast, for low density datasets (e.g. Wikipedia and Epinions) WT propagation achieves maximum marginal gain by connecting low degree nodes.

Comparison with edge recommendation heuristics. We now compare *Greedy* and *CG*, under *UNI* propagation, with two commonly used heuristics – *Degree* and *FoF*. In Figure 8, both *Degree* and *FoF* have insignificant lifts compared to *CG* and *Greedy*, which shows the merit of applying optimization techniques instead of generally accepted heuristics. To highlight the difference, *CG* has a lift 80–95 times that of *FoF* and *Degree*.



(a) Flickr (b) Epinions

Figure 8: Comparison with *Degree* and *FoF* based heuristics.



(a) Varying F (b) Varying k (c) Varying UNI p

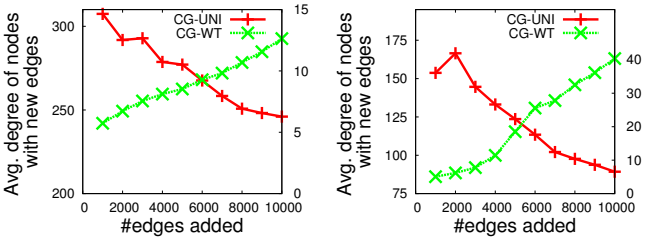
Figure 9: Lift versus varying parameters for the Epinions dataset.

Varying *CG* algorithm parameters. Figure 9(a) shows lift as the fraction of nodes with original content varies. We show results for the Epinions dataset alone; other datasets show similar trends. 10K edges are added for each $F = \frac{|S|}{|V|}$, varying from 1% to 10%. As expected, with increasing F the lift decreases as the fraction of yet-to-be-reached nodes decreases. Again, *CG* optimization yields the best results over the entire range of F considered.

Figure 9(b) shows the impact of varying the number of recommended edges per node. Initially, the lift increases for all the models. Beyond $k = 20$, however, the lift for the *UNI* models stabilizes whereas the lift for WT propagation *decreases*. For the WT propagation probability assignment, as the degree of a node increases with additional edges its propagation probability decreases; beyond a point the overall spread begins to get impacted.

Finally, Figure 9(c) shows the impact of varying the propagation probability p for the *UNI* model. As expected, the lift increases with p. At $p = 0.1$, the network has ample content prior to any edge additions. As a result, we observe the lift beginning to drop.

Comparing UNI and WT. Figure 10 shows how *UNI* and *WT* probability assignments result in different *kinds* of nodes being selected for edge recommendations. The *WT* probability assignment results in new links originating from low degree nodes, since those nodes have the highest *per-edge* propagation probabilities. Thus, low-degree nodes are preferred until such nodes become increasingly rare. On the other hand, *UNI* is biased towards picking high-degree nodes, which have the highest expected number of neighbors receiving content.



(a) Flickr (b) Wikipedia

Figure 10: *UNI* versus *WT* – Average degree of nodes with new edges. Note: Right axis is for *CG-WT* and left is for *CG-UNI*.

Runtime performance of continuous greedy. Computing 5K recommendations for the Twitter, Flickr and Epinions datasets took 40913, 2169 and 1581 seconds, respectively, under the *UNI* propagation model. Clearly, the sparse one million node Twitter graph takes comparatively much longer due to the larger number of missing edges that are potential recommendations. It is important to note that in practice, recommendations are computed offline, thus allowing for the higher computation cost. Observe that we can also speed up computation by parallelizing parts of our algorithms using a Map-Reduce framework. Nevertheless, efficient methods based on greedy heuristics is a direction for further exploration.

7. RELATED WORK

Following categories of research relate to the current work.

Recommendations and link prediction in social networks: As mentioned in Section 1, a large number of interests or profiles (i.e. college attended, current city, etc.) based and FoF based recommendation algorithms have been proposed in the literature [3, 1, 11]. Link prediction algorithms have also been developed for friend recommendations [21, 18]. These algorithms, unlike the proposed work, do not consider content spread as an explicit objective while recommending links. Recent work by Roth et al. [20] defines an Interactions Rank (IR) metric based on email interactions between users. The IR score is used within a Friend Suggest algorithm to recommend connections similar to a seed set of users. Another work, *Twittomender* [12], recommends users to follow on Twitter based on a combination of content and collaborative filtering type features.

Influence propagation and maximization: A large number of social contagion models have been proposed for explaining the diffusion of information and ideas through social connections. The *linear threshold model* [10] and the *independent cascade (IC) model* [9] are the most widely studied probabilistic models of diffusion. Variations of these models – *decreasing cascade model* [15], *triggering model* [14], and *non-progressive models* such as the *Susceptible/Infected/Susceptible* (SIS) model [19] have also been studied. Most of these models are compute intensive to simulate.

The *influence maximization problem*, also known as the *target set selection problem* [7, 14, 15] addresses the maximization of social contagion within the propagation models mentioned above. Recent work [17, 5, 4] has focused on efficient techniques for influence maximization. These techniques identify a set of nodes as opposed to edge recommendations in our setting.

Edge augmentation: Adding edges to graphs has been explored with other objectives – minimizing the network diameter [6] and maximizing algebraic connectivity for robustness [8], to list a few.

8. CONCLUSION

We introduced the problem of recommending connections in a social network with the explicit objective of maximizing content spread in the network. Our content maximization problem is interesting in two ways. First, the problem is NP-hard and non-submodular. Second, we impose per-node constraints on the maximum number of new links as opposed to a global constraint on the number of selected nodes as in the influence maximization problem. We proposed a novel RMPP model that admits submodularity leading to computationally feasible approximation algorithms in the presence of the above constraints. Simulation results on realistic graphs demonstrate the superiority of our approach in comparison with commonly used heuristics.

The proposed content maximization framework has interesting extensions. The model currently assumes that the content generated at each node is independent and non-competing. Adapting the model to overcome these assumptions is a direction worth exploring. Scalability, alternate models for propagation (e.g. SIS diffusion model) and effectiveness on live web-scale networks are aspects that also need further investigation.

9. REFERENCES

[1] Official Facebook Blog:. `http://blog.facebook.com/blog.php?post=15610312130`.

[2] Discovering who to follow. Official Twitter Blog. `http://blog.twitter.com/2010/07/discovering-who-to-follow.html`.

[3] J. Chen, W. Geyer, C. Dugan, M. Muller, and I. Guy. Make new friends, but keep the old: recommending people on social networking sites. In *CHI '09*, pages 201–210, Boston, MA, USA, 2009.

[4] W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *KDD '10*, pages 1029–1038, Washington, DC, USA, 2010.

[5] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In *KDD '09*, pages 199–208, New York, NY, USA, 2009.

[6] E. D. Demaine and M. Zadimoghaddam. Minimizing the diameter of a network using shortcut edges. In *SWAT*, pages 420–431, 2010.

[7] P. Domingos and M. Richardson. Mining the network value of customers. In *KDD '01*, pages 57–66, San Francisco, California, 2001.

[8] A. Ghosh and S. Boyd. Growing well-connected graphs. In *Growing Well-connected Graphs*, pages 6605 – 6611, 2006.

[9] J. Goldenberg, B. Libai, and E. Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters*, pages 211–223, August 2001.

[10] M. Granovetter. Threshold models of collective behavior. *American Journal of Sociology*, (83):1420–1443, 1978.
[11] I. Guy, I. Ronen, and E. Wilcox. Do you know?: recommending people to invite into your social network. In *IUI '09*, pages 77–86, 2009.
[12] J. Hannon, M. Bennett, and B. Smyth. Recommending twitter users to follow using content and collaborative filtering approaches. In *RecSys '10*, pages 199–206, 2010.
[13] B. Huang and T. Jebara. Loopy belief propagation for bipartite maximum weight b-matching. In M. Meila and X. Shen, editors, *AISTATS '07*, volume 2 of JMLR: W&CP, March 2007.
[14] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *KDD '03*, pages 137–146, Washington, D.C., 2003.
[15] D. Kempe, J. M. Kleinberg, and É. Tardos. Influential nodes in a diffusion model for social networks. In *ICALP'05*, pages 1127–1138, 2005.
[16] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Predicting positive and negative links in online social networks. In *WWW '10*, pages 641–650, 2010.
[17] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *KDD '07*, pages 420–429, San Jose, California, USA, 2007.
[18] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *CIKM '03*, pages 556–559, New Orleans, LA, USA, 2003.
[19] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.
[20] M. Roth, A. Ben-David, D. Deutscher, G. Flysher, I. Horn, A. Leichtberg, N. Leiser, Y. Matias, and R. Merom. Suggesting friends using the implicit social graph. In *KDD '10*, pages 233–242, Washington, DC, USA, 2010.
[21] R. Schifanella, A. Barrat, C. Cattuto, B. Markines, and F. Menczer. Folks in folksonomies: social link prediction from shared metadata. In *WSDM '10*, pages 271–280, New York, NY, USA, 2010.
[22] 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 *SOCIALCOM '10*, pages 177–184, Washington, DC, USA, 2010. IEEE Computer Society.
[23] J. Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *STOC*, pages 67–74, 2008.
[24] R. Zafarani and H. Liu. Social computing data repository at ASU, 2009.

APPENDIX

A. PROOF OF THEOREM 2

THEOREM: The content spread function $f(X)$ under the RMPP model is submodular.

We show that $P_X(i,c) = 1 - \prod_{j \in V(c)}(1 - q_X(j,i))$ is submodular. Since the sum of submodular functions is also submodular, we get that $f(X) = \sum_c \sum_i P_X(i,c)$ is submodular.

For edge e and edge subsets $S \subseteq T$, we are looking to show that $P_{S\cup\{e\}}(i,c) - P_S(i,c) \geq P_{T\cup\{e\}}(i,c) - P_T(i,c)$. Without loss of generality, let $V(c) = \{1, 2, \ldots, n\}$. Recollect that $V(c)$ is the set of nodes with content c. Furthermore, for edge set $T \cup \{e\}$, let the RMPPs from only nodes $1, \ldots, r$ to i pass through edge e. We rely on the following three observations for proving that $P_X(i,c)$ is submodular.

1. (a) $q_T(j,i) \geq q_S(j,i)$, (b) $q_{T\cup\{e\}}(j,i) \geq q_{S\cup\{e\}}(j,i)$, and (c) $q_{S\cup\{e\}}(j,i) \geq q_S(j,i)$. Probability of RMPPs cannot decrease with the addition of new edges.

2. $q_{S\cup\{e\}}(j,i) = q_{T\cup\{e\}}(j,i)$, $j \leq r$. With the restriction of at the most one new edge per RMPP, the RMPPs from j to i for edge set $S \cup \{e\}$ also pass through e and are identical to those for $T \cup \{e\}$.

3. $q_{T\cup\{e\}}(j,i) = q_T(j,i)$, $j > r$. RMPPs from j to i for $T \cup \{e\}$ that do not contain edge e must be identical to the ones for T.

Now, $P_{T\cup\{e\}}(i,c) - P_T(i,c)$

$$= (1 - \prod_{j=1}^{n}(1 - q_{T\cup\{e\}}(j,i))) - (1 - \prod_{j=1}^{n}(1 - q_T(j,i)))$$

$$= \prod_{j=1}^{n}(1 - q_T(j,i)) - \prod_{j=1}^{n}(1 - q_{T\cup\{e\}}(j,i)) \text{/* Applying 3 */}$$

$$\leq (\prod_{j=1}^{r}(1 - q_T(j,i)) - \prod_{j=1}^{r}(1 - q_{T\cup\{e\}}(j,i))) \quad (6)$$

$$\cdot \prod_{j=r+1}^{n}(1 - q_{T\cup\{e\}}(j,i)) \quad \text{/* Applying 1(a), 1(b) and 2 */}$$

$$\leq (\prod_{j=1}^{r}(1 - q_S(j,i)) - \prod_{j=1}^{r}(1 - q_{S\cup\{e\}}(j,i))) \quad (7)$$

$$\cdot \prod_{j=r+1}^{n}(1 - q_{S\cup\{e\}}(j,i)) \quad \text{/* Applying 1(c) */}$$

$$\leq \prod_{j=1}^{n}(1 - q_S(j,i)) - \prod_{j=1}^{n}(1 - q_{S\cup\{e\}}(j,i))$$

$$\leq P_{S\cup\{e\}}(i,c) - P_S(i,c)$$

Thus, $f(\cdot)$ is submodular.

B. PROOF OF THEOREM 4

THEOREM: Let $|V| = n$, $\delta = m^2$ and $r = m^5$. Further, let our partitioning scheme generate edge sets $X_1, \ldots, X_s$. Then w.h.p. $\mathbf{E}[\max_i f(X_i)] \geq \frac{1}{3+2\epsilon} \cdot (1 - \frac{1}{e}) \cdot f(X_{opt})$, where $\epsilon = \sqrt{\frac{8}{k}\log(n)}$.

Consider the set X obtained as a result of our randomized rounding procedure. Due to Theorem 3, we have that $\mathbf{E}[f(X)] \geq (1 - \frac{1}{e}) \cdot f(X_{opt})$.

Now, let Y_j be the number of edges in X incident on node j. Also, let $\epsilon = \sqrt{\frac{8}{k}\log(n)}$. Recall that an arbitrary edge e_i is included in X with probability y_i. Further, since $\bar{y}$ is feasible, $\sum_{j \in e_i} y_i \leq k$ for all nodes j. Thus, $\mathbf{E}[Y_j] \leq k$. Applying Chernoff Bounds, we get

$$P(Y_j \geq (1+\epsilon) \cdot \mathbf{E}[Y_j]) < exp(-\frac{\mathbf{E}[Y_j] \cdot \epsilon^2}{4})$$

Since $\mathbf{E}[Y_j] \leq k$, we get

$$P(Y_j \geq (1+\epsilon) \cdot k) < exp(-\frac{k \cdot \epsilon^2}{4})$$

Now define $Y_{max} = \max_j\{Y_j\}$. By the union bound, we get that

$$P(Y_{max} \geq (1+\epsilon) \cdot k) < n \cdot exp(-\frac{k \cdot \epsilon^2}{4})$$

The above probability is extremely small ($\leq \frac{1}{n}$) for $\epsilon = \sqrt{\frac{4}{k}\log(n^2)}$. Thus, we get that w.h.p. $Y_j \leq (1+\epsilon) \cdot k$ for all nodes j.

Next, we show that our partioning scheme divides set X into at most $2\epsilon + 3$ feasible sets X_i. This is because w.h.p., for any edge $(u,v) \in X$, at most $1 + \epsilon$ sets X_i can have k edges incident on each of the vertices u and v (since $Y_j \leq (1+\epsilon) \cdot k$). Thus, w.h.p., in at least one of the $2\epsilon + 3$ sets X_i, both u and v must have fewer than k incident edges, and so $X_i \cup \{(u,v)\}$ is feasible.

Now, since $f(\cdot)$ is submodular, we have that $\sum_i f(X_i) \geq f(\cup_i X_i)$. Furthermore, there are at most $2\epsilon + 3$ feasible sets X_i. Thus, we get that $\max_i f(X_i) \geq \frac{f(X)}{2\epsilon+3}$, and so $\mathbf{E}[\max_i f(X_i)] \geq \frac{1}{3+2\epsilon} \cdot (1 - \frac{1}{e}) \cdot f(X_{opt})$.