<table>
<thead>
<tr>
<th><strong>Title</strong></th>
<th>The probabilistic maximum coverage problem in social networks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Author(s)</strong></td>
<td>Fan, X; Li, VOK</td>
</tr>
<tr>
<td><strong>Citation</strong></td>
<td>Globecom - IEEE Global Telecommunications Conference, 2011</td>
</tr>
<tr>
<td><strong>Issued Date</strong></td>
<td>2011</td>
</tr>
<tr>
<td><strong>URL</strong></td>
<td><a href="http://hdl.handle.net/10722/158776">http://hdl.handle.net/10722/158776</a></td>
</tr>
<tr>
<td><strong>Rights</strong></td>
<td>©2011 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.; This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.</td>
</tr>
</tbody>
</table>
The Probabilistic Maximum Coverage Problem in Social Networks

Xiaoguang Fan*, Victor O.K. Li†
The University of Hong Kong, Pokfulam, Hong Kong, China
*xgfan@eee.hku.hk †vli@eee.hku.hk

Abstract—In this paper we consider the problem of maximizing information propagation in social networks. To solve it, we introduce a probabilistic maximum coverage problem, and further propose a cluster-based heuristic and a neighborhood-removal heuristic for two basic diffusion models, namely, the Linear Threshold Model and the Independent Cascade Model, respectively. Our proposed strategies are compared with the pure greedy algorithm and centrality-based schemes via experiments on large collaboration networks. We find that our proposed algorithms perform better than centrality-based schemes and achieve approximately the same performance as the greedy algorithm. Moreover, the computational load is significantly reduced compared with the greedy heuristic.

I. INTRODUCTION

Understanding how to maximize the spreading of influence is important in marketing, online social advertising and new product promoting. Online social network sites like Facebook and LinkedIn offer huge resources and free space for information dissemination, spawning tremendous opportunities for electronic commerce. The influence maximization problem is defined as follows: Given that K nodes are allowed to be activated initially, how do we select them in order to gain the maximum influence?


In this paper, to solve the problem, we introduce a probabilistic maximum coverage problem and further propose a cluster-based heuristic and a neighborhood-removal heuristic for the Linear Threshold Model and the Independent Cascade Model, respectively. We compare our heuristics with the greedy algorithm through simulation on large collaboration networks, and find that our algorithms work as well as the greedy heuristic at much lower computational cost.

We proceed in this paper as follows. Problem formulation is described in Section II, probabilistic coverage maximization in Section III, cluster-based heuristic in Section IV, neighborhood-removal heuristic in Section V, experimental results in Section VI and we conclude in Section VII.

II. PROBLEM FORMULATION

In this section, we describe the influence maximization problem and two operational diffusion models in detail.

A. Problem description

Considering a connected social network with N nodes as an directed graph G(V,E) where V is the set of vertices and E is the set of edges, the neighbor set of node j could be defined as N_j = {i ∈ V : (i,j) ∈ E or (j,i) ∈ E}. Each node has two states: active or inactive, and the probability for one switching from inactive to active increases as more of its neighbors become active. Once a node becomes active, it cannot return to be inactive. At the very beginning, all of the nodes are inactive, and we try to assign K nodes to be active. In subsequent discrete steps, nodes are activated by their active neighbors and in turn activate others. The process ends when no more activations are possible. The influence maximization problem is as follow: If K nodes are permitted to be initially active, determine this K-node set to maximize the expected number of active nodes at the end of the process.

B. Diffusion models

In the literature two basic diffusion models are utilized to determine how a node is influenced by its neighbors, namely, the Linear Threshold Model (LTM) [9] and the Independent Cascade Model (ICM) [10]. In the former, a node i has a weight b_{i,j} to influence node j and \sum_{j ∈ N_i} b_{i,j} ≤ 1 (if (j,i) \notin E, b_{i,j} = 0). Node j is pre-assigned a threshold \theta_j which is uniformly distributed in [0,1]. At any single step, node j is successfully activated if the sum of weights from its active neighbors exceeds \theta_j. In the latter model, if node i becomes active at step t, it has a probability p_{i,j} to successfully activate each inactive neighbor j in step t + 1. Note that the order of activation is arbitrary for multiple active neighbors and the probability is independent of the historical activations. Furthermore, whether or not i succeeds, it does not have any chance to activate j again.

III. PROBABILISTIC MAXIMUM COVERAGE PROBLEM

Given a universal set of elements U, an integer K, and a collection of subsets of U, the goal of the maximum coverage problem is to select K subsets \{S_1, S_2, ..., S_K\} so that the
number of covered elements \( \bigcup_{i=1}^{K} S_i \) is maximized. The goal of the influence maximization problem is to select the set of nodes that could influence the majority of the nodes in the network, which in a sense, is similar to the maximum coverage problem. However, due to the uncertain activation process, “cover” in influence propagation is a probabilistic concept.

Thus we formulate the influence maximization problem as a probabilistic maximum coverage problem. Firstly, we define a collection of subsets \( R = \{ S_1, S_2, \ldots, S_N \} \) for the social network \( G \) with \( N \) nodes and each subset \( S_i \) is pre-allocated with node \( i \) deterministically. In the activation process, a node \( j \) successfully activated by node \( i \) is added to \( S_i \). Due to the uncertain activation, the subordinate relationship between node \( j \) and \( S_i \) is probabilistic. Here we use \( P(j \in S_i) \) to define the probability of node \( i \) successfully activating node \( j \). Note that \( P(i \in S_i) = 1 \). Thus the problem could be illustrated as follows: Given a universal set \( U \) with \( N \) elements \( \{ 1, 2, \ldots, N \} \), a collection of subsets \( R = \{ S_1, S_2, \ldots, S_N \} \) with subordinate relationship probability \( \{ P(j \in S_i) : i, j = 1, \ldots, N \} \), and an integer \( K \), we aim to find \( R' \subseteq R \) with \( K \) subsets to maximize the expected number of covered elements \( E(\bigcup_{i=1}^{K} S_i) \), and finally each element \( i \) corresponding to the set \( S_i \subseteq R' \) forms the final solution.

The traditional deterministic maximum coverage problem \cite{11} is NP-hard and there are some good approximation algorithms for the solution \cite{12}. However, due to the complexity of calculating the subordinate relationship probability, we could not apply these algorithms directly to our probabilistic version. Therefore we propose two heuristic methods based on the idea of maximum coverage.

IV. CLUSTER-BASED HEURISTIC

First we describe our solution approach in the Linear Threshold Model. We decompose the problem into two subproblems: cluster identification and cluster head selection.

A. Cluster identification

Clustering is a tool to measure structural equivalence \cite{13} in social networks. It aims to identify the nodes which are sufficiently similar and group them into sets. In terms of network topology, clusters are recognized as separated groups of aggregated nodes with high density for some specific measurements (social distance, degree and so on). Nodes are “close” to one another in a cluster due to their strong relationship and great similarity. Thus a cluster should be a good candidate to be the selected subset mentioned in Section III.

Considering the influence diffusion in the Linear Threshold Model, a weight \( b_{i,j} \) is allocated on edge \((i,j)\) to represent the tie influential power from node \( i \) to node \( j \). Since the weight assignment takes a significant role in determining the network status, we try to do the clustering identification based on weighted links in a divisive way.

We define a cut-off threshold \( \omega \) such that nodes connected with an edge with a higher weight could be regarded as connected and subsumed in the same cluster. The original network \( G \) is reconstructed so that links with weight lower than \( \omega \) are deleted. Then we utilize Depth-First Search (DFS) \cite{14} to identify connected components. Thus the graph is divided into several non-overlapping pieces which could be regarded as the clusters we need. Figure 1 gives a simple example for clustering. We can see that the network on the left is divided into three clusters on the right by removing those edges with weight below the weight threshold \( \omega = 0.4 \). Next we order the clusters by decreasing size and the first \( K \) clusters are selected so as to maximize coverage.

B. Cluster head selection

Next we need to find the target node in each chosen cluster. This turns out to be a cluster head selection problem. Aiming to maximize the number of activated nodes, a cluster head should be the most influential node in the cluster. Drawing the notion of centrality from social network analysis, we define a weighted degree metric \( \gamma_i \) which is the sum of weights from the node \( i \) to all of its neighbors. The node with highest \( \gamma_i \) in the cluster is chosen as the head. In Figure 1 nodes \( A, B \) and \( C \) should be selected as the cluster heads in their corresponding clusters due to their highest metric values (\( \gamma_A = 1.5, \gamma_B = 1 \) and \( \gamma_C = 2.5 \)).

Finally, a formal statement of cluster-based heuristic is given in Algorithm 1.

V. NEIGHBORHOOD-REMOVAL HEURISTIC

Next we introduce our heuristic method for the Independent Cascade Model, which includes two essential components: up-to-k-hop degree metric and neighbor removal mechanism.

A. Up-to-k-hop degree metric

Traditional centrality-based schemes solely focus on the centrality value of each target node since centrality is widely used in sociology to evaluate the social importance of individual nodes. However, one’s influence is not only embodied by the number of friends but also reflected by what kinds of friends he has. In other words, the neighbors’ influence should also be considered when measuring one’s influential power \cite{15}. Here we define an up-to-k-hop degree metric \( \mu^k_i \) to evaluate the influence of node \( i \). \( \mu^k_i \) is the sum of degrees of each node whose social distance \(^1\) is at most \( k \) from node

\(^1\)The social distance is defined as the number of edges on the shortest path between two nodes.
Algorithm 1: Cluster-based heuristic

Let \( e_1, \ldots, e_N \) be nodes and \( C_1, \ldots, C_M \) be clusters.

Input:

Network \( G(V,E) \) with weight \( b_{i,j} \) for \( (i,j) \subseteq E \).
A given integer \( K \) and a threshold \( \omega \).

Output:

The final target set \( S_I \);

1: Start with \( S_I = \emptyset \).
2: if \( (i,j) \subseteq E \) and \( b_{i,j} < \omega \) then
3: drop \( (i,j) \) from \( E \).
4: end if
5: Identify \( M \) clusters \( (M \geq K) \) from \( G \) by DFS.
6: Choose the first \( K \) largest clusters \( \{ C_1, \ldots, C_K \} \).
7: for \( i = 1 \) to \( K \) do
8: find node \( e \) with the largest value of \( \gamma \) in \( C_i \).
9: \( S_I \leftarrow S_I \cup \{ e \} \)
10: end for

Figure 2. Up-to-k-hop degree metric in the Independent Cascade Model.

Thus the original degree of node \( i \) can be expressed as an up-to-0-hop degree metric \( \mu_0^i \). Figure 2 gives an example to show how the up-to-1-hop degree metric affects the node selection to maximize influence. The value near each node \( i \) is the comparison between up-to-0-hop and up-to-1-hop degree metrics with the format \( \mu_0^i / \mu_1^i \). In the viewpoint of traditional degree centrality (up-to-0-hop), the most influential node should be node \( A \) (\( \mu_A^0 = 5 \)). However, the result changes into node \( B \) for up-to-1-hop degree metric (\( \mu_B^1 = 14 \)). In terms of network topology, node \( B \) is indeed more “central” than node \( A \) since it connects most of the nodes with high degree and acts as a bridge associating two subnetworks \( G_1 \) and \( G_2 \) (shown by dotted lines). Thus node \( B \) should undoubtedly be the first choice and it will trigger a greater cascade of activations than node \( A \).

B. Neighbor removal mechanism

After providing a more accurate metric to target influential nodes, we further deal with the problem of overlapping neighborhoods. Although each target node is judged to be influential by some specific metric (up-to-k-hop degree), their activation sets may overlap with each other. A particular situation is described as follows: Suppose an influence diffusion process starts with \( K \) initially activated nodes, the first \( M \) \((M \ll K)\) selected nodes with high metric values may trigger a large set of activations in the network and the additional chosen nodes are very likely to be subsumed in this set. Thus the rest of the \( K - M \) nodes could hardly contribute to the final performance. In order to avoid overlapping neighbors, we introduce a neighbor removal mechanism. Firstly we give a definition of domination as follows:

Definition 1: Given a set of nodes \( S = \{ e_1, e_2, \ldots, e_N \} \) and a node \( e_i \notin S \), \( e_i \) is dominated by \( S \) with confidence level \( \eta \) if the discriminant \( 1 - (1 - p)^m > \eta \), where \( m \) is the number of activated nodes in \( S \) connecting node \( e_i \), \( p \) is the activation probability in the Independent Cascade Model, and \( \eta \) is the domination threshold.

From Definition 1 we can see that \( 1 - (1 - p)^m \) is the probability that node \( i \) is successfully activated by at least one of its neighbors.

Then our removal process is illustrated as follows: Given a sequence of nodes sorted by decreasing up-to-k-hop degree metric, we need to select \( K \) nodes. The head of the sorted list is selected at first. In the following steps we check the rest of the nodes in order and drop the node which is dominated by the set of nodes already selected. The process ends when \( K \) qualified nodes are found. An example is shown in Figure 2. Suppose we want to select 4 nodes. We assume that \( p = 6\% \), \( \eta = 0.1 \) and up-to-1-hop degree metric is utilized to sort nodes. Obviously node \( B \) is chosen at first (\( \mu_B^1 = 14 \)). Node \( A \) and \( C \) have the second and the third largest metric values (\( \mu_A^1 = 13 \), \( \mu_C^1 = 11 \)), respectively, and are both qualified to be chosen since \( 1 - (1 - p)^1 = 0.06 < 0.1 \). The next choice should be node \( D \), but it is dominated by the initially chosen set \( \{ B, A, C \} \) since \( 1 - (1 - p)^2 = 0.1164 > 0.1 \). Thus node \( D \) is dropped and node \( E \) is selected.

Finally, a formal statement of the neighborhood-removal heuristic is given in Algorithm 2.

VI. EXPERIMENTAL RESULTS

In this section we evaluate our cluster-based and neighborhood-removal heuristic with two typical strategies in the Linear Threshold Model and the Independent Cascade Model, respectively, via real academic collaboration networks [16].

A. Dataset and influence model

The dataset we utilized is Arxiv co-authorship network in General Relativity and Quantum Cosmology category from the e-print arXiv [16]. An edge between authors \( i \) and \( j \) is included in the graph if they co-authored a paper, and multiple paper co-authorships will not cause more edges. We use the largest connected component of the dataset to construct a graph with 4158 nodes and 26850 edges.

In addition, we define the reciprocal of degree as the weight of each node in the Linear Threshold Model. If node \( i \) with degree \( d_i \) connects node \( j \) with degree \( d_j \), the edge \((i,j)\) has weight \( \frac{1}{d_i} \) and the edge \((j,i)\) has weight \( \frac{1}{d_j} \). The weight assignment example is shown in Figure 1. For the
Algorithm 2: Neighborhood-removal heuristic

Input:
- The node set $S = \{e_1, ..., e_N\}$ sorted by decreasing up-to-k-hop degree metric.
- Network $G(V,E)$ with activation probability $p$.
- A given integer $K$ and a domination threshold $\eta$;

Output:
- The final target set $S_t$;

1: Start with $S_t = \emptyset$
2: $S_t \leftarrow S_t \cup \{e_1\}$
3: $i = 1$
4: while $|S_t| < K$ do
5:   if $e_i$ is dominated by $S_t$ then
6:     $i \leftarrow i + 1$
7:     continue
8:   else
9:     $S_t \leftarrow S_t \cup \{e_i\}$
10: end if
11: end while

Independent Cascade Model, we discuss the situation with activation probability $p = 5\%$ and $p = 20\%$, respectively.

B. Simulation setting

We compare our two methods with the pure greedy algorithm and the centrality-based schemes.

- **Cluster-based heuristic** is proposed by us in the Linear Threshold Model. We set the cut-off threshold $\omega = 0.4$.
- **Neighborhood-removal heuristic** is also proposed by us in the Independent Cascade Model. Here we utilize the up-to-1-hop degree metric to measure the node’s influential power and set the domination threshold $\eta = 0.1$.
- **Pure greedy algorithm** [3] utilizes a hill-climbing heuristic and chooses each node with maximal marginal gain.
- **Centrality-based schemes** such as [3] select nodes in the order of decreasing centrality values. Here we choose degree centrality and betweenness centrality to be the metrics of centrality. Thus we have degree-based and betweenness-based schemes.

The performance metric is the number of activated nodes at the end of the process. We take a host of runs for each initial target set and calculate the average.

C. Result

Figure 3 shows the performance comparison of cluster-based heuristic (Cluster), greedy algorithm (Greedy), degree-based (Degree) and betweenness-based schemes (Betweenness) in the Linear Threshold Model. The x-axis represents the number of initially active nodes and the y-axis is the final active set size. We find that our cluster-based heuristic gives the best performance of the four algorithms compared except

\[^2\text{Betweenness of a node is the proportion of shortest paths between all possible pairs of nodes that pass through this node [13]}\]
when the target size is less than 14, or when it is between 21 and 22, when the greedy heuristic is better.

Another interesting observation is that when the target size is small, the performance of the cluster-based heuristic is approximately the same as the degree-based and betweenness-based schemes, and even worse at some individual points. This may be explained as follows. As mentioned in Section III, influence maximization could be formulated as a probabilistic maximum coverage problem. We hope to select those influential nodes which could trigger a large fraction of activations individually while avoiding overlapping activations as much as possible. This is a kind of combination between intensification and diversification. The cluster-based heuristic strikes a good balance between them and gets a satisfactory performance since it utilizes cluster identification to realize the diversification, and cluster head selection to achieve the intensification. However, when the initial target size is small, the effect of intensification may be bigger since nearby nodes (for example, nodes in the same cluster) may cooperate to increase the successful activation probability and lead to more activations. By contrast, when the initially chosen set is big enough, the problem of overlapping neighborhoods becomes more important and then the benefit of cluster-based heuristic could be seen.

Figure 4 evaluates the performance of neighborhood-removal heuristic (Removal), greedy algorithm (Greedy), degree-based (Degree) and betweenness-based schemes (Betweenness) in the Independent Cascade Model with probability \( p = 5\% \). We observe that our neighborhood-removal heuristic is always better than centrality-based schemes with the target size from 1 to 30, and is also better than the greedy algorithm when the target size is bigger than 14.

Figure 5 shows the performance of the four algorithms compared in the Independent Cascade Model with probability \( p = 20\% \). One observation is that when the activation probability is big enough (20%), the first chosen node would be responsible for the majority of activations in all of the algorithms. The subsequently selected nodes make nearly no progress for the centrality-based schemes due to the problem of overlapping neighborhoods mentioned in Section V-B. However, our Neighborhood-removal heuristic keeps enhancing the active set size due to its unique removal mechanism to avoid overlapping. Moreover, it preforms approximately the same as the greedy algorithm.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Greedy</th>
<th>Cluster</th>
<th>Removal</th>
<th>Degree</th>
<th>Betweenness</th>
</tr>
</thead>
<tbody>
<tr>
<td>LTM</td>
<td>129</td>
<td>0.423</td>
<td>-</td>
<td>0.5554</td>
<td>0.9138</td>
</tr>
<tr>
<td>ICM (5%)</td>
<td>33.75</td>
<td>-</td>
<td>0.1119</td>
<td>0.0528</td>
<td>0.0552</td>
</tr>
<tr>
<td>ICM (20%)</td>
<td>248.8</td>
<td>-</td>
<td>0.3212</td>
<td>0.1208</td>
<td>0.1252</td>
</tr>
</tbody>
</table>

Table I

| COMPUTATION TIME FOR THE ALGORITHMS (HOURS) |

Next we discuss the program running time. Table I shows the average computation time for each algorithm, using a dual-core 3.0GHz PC. We can see that the running time for the cluster-based strategy is the smallest in the Linear Threshold Model, and is only 0.33% of the greedy algorithm. Meanwhile the computation time of the neighborhood-removal heuristic is only 0.33% and 0.13% of the greedy algorithm in the Independent Cascade Model with \( p = 5\% \) and \( p = 20\% \), respectively. In fact, for larger datasets, the greedy algorithm will be computationally infeasible. Thus our proposed strategies are probably the best choices to achieve high performance with limited cost.

To summarize, our cluster-based and neighborhood-removal heuristics obtain better performance than the centrality-based schemes and achieve approximately the same performance as the greedy algorithm. In addition, our schemes have greatly reduced cost compared to the greedy heuristic.

VII. CONCLUSION

In this paper, we study the influence maximization problem. The problem is formulated as a probabilistic maximum coverage problem, and we propose two efficient heuristics, namely, the cluster-based and the neighborhood-removal heuristics, to solve it.

ACKNOWLEDGEMENT

This research is supported in part by the University of Hong Kong Strategic Research Theme of Information Technology.

REFERENCES