Stochastic Dynamic Programming Model for Revenue Optimization in Social Networks

Patrick Hosein
Department of Computer Science
The University of the West Indies
St. Augustine, Trinidad
Email: patrick.hosein@sta.uwi.edu

Trisha Lawrence
Department of Mathematics
University of Saskatchewan
Saskatoon, Canada
Email: trisha.lawrence@usask.ca

***Abstract*—The rapid increase in the global availability and use of the Internet has made it the most effective platform for the distribution of product information and advertising when compared to the traditional media such as print, television and radio. In addition, in the case of Online Social Networks (OSN), one can take advantage of the influences that one's actions has on other users and, in particular, one's friends. Even if a user does not purchase a product, their online mention of the product can have positive or negative influences. This social behavior can be used to improve advertising strategies. For example, if a user is given an advertisement impression (a linked icon) and clicks on it then it is more likely that their friends (if told of the action) will also click if they are subsequently given the impression. In this paper we provide a Stochastic Dynamic Programming formulation of this problem together with its solution. Because of the computational complexity of the solution we provide a simple heuristic that we show to be computationally much faster.**

I. Introduction

The use of the Internet and its many applications continues to have a rapid global growth. This rate of penetration has increased dramatically due to the availability and falling costs of mobile devices, especially in third world countries. This platform is now recognized to be more efficient for advertising products [1] when compared to traditional media because of (a) ease of deployment (b) worldwide reach and (c) ability to target one's audience based on readily available information. Many Internet companies derive the majority of their revenue through advertisements (e.g., over 90% in the case of Facebook [2]). Hence the optimal timing, placement and targeting of advertisements, especially on Online Social Networks, has become an important research topic.

In this paper we focus on the placement of advertisement impressions in an OSN. Customers who wish to advertise pay for a number of these impressions and the OSN operator must then place these impressions on pages that are displayed to its users. The objective is to place these impressions so as to maximize the overall value to the customer. For example, this value can be the resulting purchases of the customer's products by those OSN users that were given impressions. In our model our objective is to place impressions so as to maximize the total expected number of clicks. In general, the number of purchases is proportional to the number of clicks and hence our objective will, in general, achieve the customer's objective. Impressions are made in stages and, in each stage, a user is provided with the total number of her friends who clicked on the impressions in prior stages. This information influences the probability that the user clicks on the impression and hence can be used to improve the total number of clicks achieved.

Next we provide related work and show that our model is quite different to what has been studied in the past and hence there is no past work with which to compare. We then provide, in Section III, a mathematical formulation of the model and explain why it is realistic and captures the important features of the problem being addressed. We then provide a simple illustrative example to demonstrate the model and the computational complexity of the solution. Finally in Section V we provide a heuristic that can be used to achieve acceptable performance at significantly reduced computational complexity. Analytic bounds on the performance of this heuristic as well as simulation results will be included in future work.

A. Related Work

Influence maximization in social networks was first formulated as an optimization problem by Kempe, Kleinberg and Tardos [3]. In their formulation users influence the buying choices of each other and the objective is to choose a subset of users who, if chosen, will result in the maximization of the outcomes of these influences. In [4] the authors focus on the influence model and, in particular, introduces a measure of tie strength. The paper [5] proposes a method (which they call AdHeat) in which hint words of influential users are distributed to other users and aggregated hints are used to determine the overall influence factor. Bhagat et. al. [6] argue that influences alone are insufficient and that one also needs to take into account adoption rate. Note that in their model they include negative influences which we also do in ours. In the paper [7], the authors use a voter model instead of the influence models in prior papers and provide simple, efficient solutions. In the paper by Saez et. al. [8] the authors investigate the relative values of users in an OSN while in [9] a credit distribution system is proposed for modeling the spread of influence throughout the OSN. Finally [10] takes a different approach which they call an influence and exploit strategy where the product is initially made available for free and the feedback from this is then used to market the product.

One difficulty in these influence models is the determination of the influence probabilities between parties. Lei et. al. [11] propose an approach whereby user feedback is used to update influence information while the influence campaign is being executed. Several heuristics have also been developed for the influence maximization model such as Prefix Excluding Maximum Influence Path [12] and Influence Ranking and Influence Estimation [13]. These are fast heuristics but they do not provide any bounds on performance. Borgs et al. [14] does provide and algorithm with accuracy guarantees and this was further enhanced by Tang [15]. Work has also been performed in topic propagation [16] and on influence propagation [17] as well as in influence probability estimation [18] and on community-based algorithms [19]. However these papers are mostly based on the Influence Maximization model which is somewhat different to our stage-based dynamic programming model.

B. Contributions

The model we propose is unlike those of previous authors and hence there are no other solutions with which we can compare. The closest work to ours is that of [20] so we will compare our model with theirs. In their paper a first user is chosen and given an impression. Two outcomes (the user clicked or did not click) must then be evaluated. In each case the click probabilities are updated and another user is chosen. This process is repeated until all impressions are made. This, though interesting, may not be suitable in practice because (a) the time between giving an impression and deciding to click is variable and (b) if hundreds of impressions must be made then the process is lengthy. Our approach uses stages and in each stage a number of impressions are allocated such that over all stages all impressions are assigned. Within a stage probabilities are updated based on all impression allocations made in that stage before solving the subsequent stage. Therefore one can consider the model in [20] as the special case of our model in which a single impression is allocated per stage. The other extreme case of our model (i.e., all impressions assigned in a single stage) has a simple solution in which the users with the top click probabilities are chosen. Note that the optimal solution increases with the number of stages (i.e., more information is available for each decision made) but we believe that a small number of stages will be sufficient to achieve gains similar to what is obtainable with one impression per stage. We are in the process of evaluating heuristics as well as bounding the optimal solution.

II. Problem Description

We model the OSN as a graph $G(V, E)$ where each vertex $v \in V$ represents a member of the OSN and each edge $e \in E$ represents friendship (assumed to be reciprocal) between the two users at the ends of e. We assume that an impression is provided at most once to a user and the user either clicks on this impression or does not click. We also assume that the probability that a user clicks on an impression is influenced by the set of friends of that user who were given impressions and clicked or did not click. In other words, if several friends of a user received impressions and many clicked then we assume that the user is more likely to click. If, on the other hand, many received impressions but many did not click then it is less likely that the user will click. This influence relationship has been well studied in the literature (e.g., see [20]) and is typically modeled by some probability function.

The model consists of multiple stages with each stage separated in time (e.g., one hour). Within each stage we assume that a certain number of impressions can be assigned (this must also be optimized). We must determine to which users these impressions should be allocated. The outcomes of a stage (i.e., which of those given impressions clicked) are observed before allocations in the subsequent stage are made. These outcomes affect the click probabilities of each user and hence affect the optimal allocation to be made in the subsequent stage.

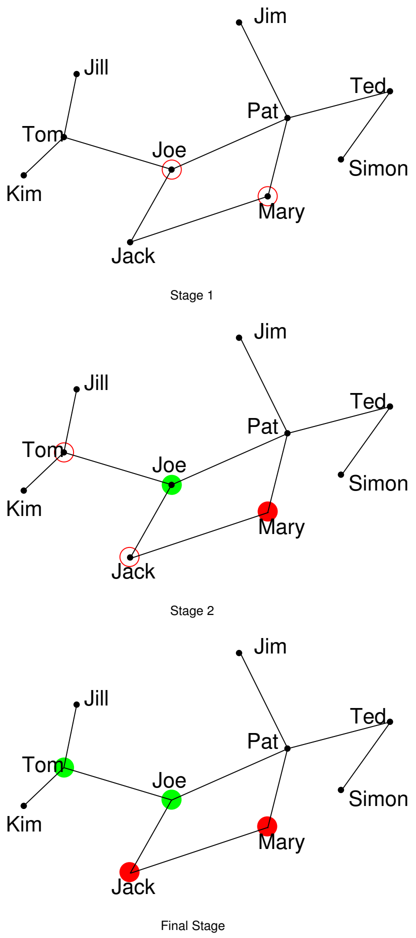
We assume that the total number of impressions to be assigned over all stages is given. This is determined by the amount of money the client is willing to spend. Typically a client (one who wants to advertise) pays per impression made. The optimization problem therefore consists of (a) determining the optimal number of impressions to be allocated in each stage and (b) given the number of impressions in a stage determining the optimal allocation. Note that each of these will depend on the outcomes of prior allocations and outcomes so the problem needs to be resolved at each stage. In other words, at each stage we first determine how many of the remaining impressions should be assigned in the stage and then we find the optimal allocation of those impressions to users.

In order to better illustrate the problem we provide a simple two stage example which is illustrated in Figure 1. Here we have a friendship graph with users Joe, Jack etc. Tom has three friends, Jill, Joe and Kim. Note that it is possible to have cycles in this graph as we see in the cycle containing Joe, Pat, Mary and Jack. Given this graph, suppose that impressions (indicated by red circles) were given to Joe and Mary in stage 1 (the top graph). This initial decision can, for example, be based on whomever is most likely to click given an impression. Once these impressions are given, some time period (the inter-stage period) is waited before additional allocations are made.

The second graph contains the second stage of the problem and the outcomes of the first stage. Suppose that Joe has clicked his impression (represented by the green solid circle) but Mary did not click her impression (represented by the red circle). Because Joe clicked then this increases the probability that Joe's friends will click if given an impression. In the second stage suppose that Tom and Jack are given impressions given their updated click probabilities. Once these impressions are allocated we again wait for the outcomes before solving the subsequent stage. Note that Tom, Joe, Mary and Jack can no longer be given impressions because an impression can only be allocated at most once to each user.

The outcomes of the second stage are provided in the last graph. Here we find that Tom clicked but Jack did not click (e.g., because of the negative influence of Mary). So we have a total of four impressions over the two stages and two of those resulted in clicks. The ratio of clicks to impressions is called the click-through rate which in this case is 50%. The optimization problem is to determine the number of impressions to use per stage and the allocation of those impressions, to users, in each stage so as to maximize the click-through rate.

Note that, even if a user clicks they may not purchase the product but typically the purchase rate grows linearly with the click-through rate and so maximizing the latter leads to maximum revenue. Also note that the time between stages is



Stage 1

Stage 2

Final Stage

Fig. 1. A Three Stage Problem

an important factor. If this time is too short then some clicks may actually occur after the subsequent stage and these would not have been taken into account. We assume that this time is sufficiently long so that all users who were given an impression have seen it and have decided to click or not click.

III. Formulation of the Optimization Problem

We provide a stochastic dynamic programming formulation of the problem. Let K denote the number of stages and let the index k denote the number of stages to go so that in the first stage $k = K-1$ and in the last stage $k = 0$. We formulate the optimization problem for index k in terms of the optimal solution of the problem with index $k-1$. We then provide the solution for the problem in the final stage. Therefore given the solution for the last stage we can obtain the solution for the second to last stage problem and hence, by induction, for any stage of the problem. In the following we use the variable x to keep track of who was given impressions, the variable c to keep track of who has clicked on impressions, the variable u is the decision variable (the present impression assignments) and p is used to represent the probability of clicking. We assume that users are indexed from 1 to $|V| \equiv N$ and, for user i in stage k, we use the following notation:

$$x_k[i] = \begin{cases} 1 & \text{if } i \text{ was previously given an impression} \\ 0 & \text{otherwise} \end{cases}$$

$$c_k[i] = \begin{cases} 1 & \text{if } i \text{ clicked a given past impression} \\ 0 & \text{otherwise} \end{cases}$$

$$u_k[i] = \begin{cases} 1 & \text{if } i \text{ is given an impression in this stage} \\ 0 & \text{otherwise} \end{cases}$$

$$p_k[i] = \begin{cases} \text{prob}(c_{k-1}[i] = 1 | u_k[i] = 1, \vec{x}_k, \vec{c}_k) & \text{if } x_k[i] = 0 \\ 0 & \text{otherwise} \end{cases}$$

Note that if $c_k[i] = 1$ then $x_k[i] = 1$ since a user can only click after given an impression. Furthermore $u_k[i] + x_k[i] \leq 1$ since an impression cannot be given to a user more than once. The vectors $\vec{c}_k$ and $\vec{x}_k$ contain the state of the system (i.e., the past clicks and impressions) while the vector $\vec{u}_k$ represents the control (what actions need to be taken in the present stage). The probabilities $\vec{p}_k$ depend on the state in that it depends on how many of i's friends were given impressions ($\vec{x}_k$) and how many of those clicked ($\vec{c}_k$). We will assume that $p_{K-1}[i] = \hat{p}$ for all users. The probabilities are then updated at each stage based on the state.

Let m_k denote the number of impressions allocated in stage k and assume that the total number of impressions available is M. Typically the number of impressions is less than the number of users so we assume $M < N$. Note that allocating an impression can never decrease the number of clicks and hence if we are maximizing the number of clicks we can assume that all impressions are used so that

$$\sum_{k=0}^{K-1} m_k = M. \tag{1}$$

For a given impression allocation $\vec{u}_k$ in stage k, the expected total number of clicks is given by the sum over all possible outcomes, the probability of the outcome times the optimal expected number of clicks for the following stage problem given the outcome. Let $\mathcal{V} \in \{0,1\}^N$ represent the set of possible outcome vectors given some allocation $\vec{u}_k$. If $\vec{v} \in \mathcal{V}$ then $v[i] = 0$ if $u_k[i] = 0$ (since a click cannot occur without an impression) and $v[i] = 0$ or 1 if $u_k[i] = 1$. Therefore $|\mathcal{V}| = 2^{m_k}$ since m_k impressions are provided in this stage. The probability that $\vec{v}$ occurs is given by

$$Pr(\vec{v}) = \prod_{i=1}^{N} u_k[i]\{p_k[i]v[i] + (1-p_k[i])(1-v[i])\} + 1 - u_k[i] \tag{2}$$

If $u_k[i] = 0$ (no impression given to the user) then the corresponding term in the product is 1 while if $u_k[i] = 1$ then the corresponding term in the product is $p_k[i]$ if $v[i] = 1$ (corresponding to a click outcome) and it is $(1 - p_k[i])$ if $v[i] = 0$ corresponding to a non-click outcome.

For an allocation $\vec{u}$ and outcome $\vec{v}$ the state is updated as

$$\vec{x}_{k-1} = \vec{x}_k + \vec{u}_k$$

since new impressions have been allocated and

$$\vec{c}_{k-1} = \vec{c}_k + \vec{v}$$

to account for those users who have clicked. Therefore if $J^*_{k-1}(\vec{x}_{k-1}, \vec{c}_{k-1}, \vec{p}_{k-1})$ is used to denote the optimal expected total number of clicks for the subsequent stage then the optimization problem for stage k can be written as

$$J_k^* = \max_{\vec{u}\in\{0,1\}^N} \sum_{\vec{v}\in\mathcal{V}|\vec{u}} Pr(\vec{v}) J_{k-1}^*(\vec{x}^k + \vec{u}, \vec{c}_k + \vec{v}, \vec{p}_{k-1}) \quad (3)$$

$$\text{subject to: } \sum_{i=1}^N u[i] = m_k \text{ and } \vec{u} + \vec{x}_k \leq 1.$$

Therefore the problem with k stages to go can be stated in terms of the one with $k-1$ stages to go. Let us now consider the final stage. In this case, given an allocation of impressions, the expected number of new clicks can be obtained by summing, over all users given impressions, the probability that the user clicks. This can be done because these probabilities are independent. The set of previous clicks are given in $\vec{c}_0$. Therefore in this case the optimization problem can be simplified as follows:

$$J_0^* = |\vec{c}_0| + \max_{\vec{u}\in\{0,1\}^N} \sum_{i=1}^N p_0[i]u[i] \quad (4)$$

$$\text{subject to: } \sum_{i=1}^N u[i] = m_0 \text{ and } \vec{u} + \vec{x}_0 \leq 1.$$

Therefore we are trying to find m_0 users that have not yet been given an impression such that the sum of their probabilities is maximum. The optimal solution is simply the m_0 users with the largest click probabilities who have not yet been given an impression. Denote this optimal allocation by $\vec{u}^*$ then we have

$$J_0^* = |\vec{c}_0| + \sum_{i=1}^N p_0[i]u^*[i] \quad (5)$$

Hence we can explicitly solve the last stage and by induction solve for any number of stages.

In the above we assumed that the number of impressions to be allocated in a stage is given. However this too must be optimized. In the last stage the total number of impressions remaining are assigned and hence this quantity is known. Now consider the two stage case. We can find the optimal number of impressions to use in the first stage by solving for each of the M possible cases (i.e. 0, 1, 2, ..., $M-1$ impressions in stage 1) with the remaining used in stage 2. The case that provides the maximum expected total number of clicks is the optimal choice. Again by induction one can similarly solve for any number of stages. However, note the exponential growth in possibilities that must be evaluated and the fact that each evaluation is computationally intensive. Therefore for large networks we will investigate heuristics that are tractable while providing near optimal performance.

IV. An Illustrative Example

In order to better understand the proposed model and the underlying optimization problem we will provide a simple illustrative example. Such examples can help provide insight that can be used to develop useful heuristics. We consider an example with 6 users (A, B, ..., F) with A and D each having two friends while all others have three (see Figure 2).

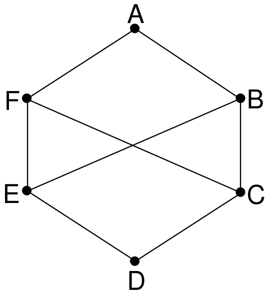


Fig. 2. Friendship graph for Illustrative Example

We assume that a total of $M = 4$ impressions are to be allocated over two stages. We will solve for each possible combination of impressions per stage, $m_1 = 1, 2, 3, 4$. Note that the cases $m_1 = 0$ and $m_1 = 4$ are identical because both cases are simply single stage problems. We assume that the click probabilities in the first stage is 0.25 for all users. Let us consider any of the users and denote their number of friends by f. Also let y denote the number of their friends that were given an impression and clicked in the first stage and let n denote the number of friends that were given an impression and did not click in the first stage. We define the click probability in the second stage as

$$p_2 = \max[0, \min[1, 0.25 + \alpha\frac{y}{f} - \beta\frac{n}{f}]] \quad (6)$$

where the factors α and β are determined by the degree of positive and negative influence of a friend's decision. In other words the probability of clicking grows linearly with the fraction of friends who have clicked and falls linearly with the fraction of friends who decided not to click. In practice one needs to estimate these probabilities from past data as is done in [11]. Note that if $\alpha = \beta = 0$ then having multiple stages does not help because the decisions of one's friends become irrelevant. In practice one would set $\beta = 0$ since it does not make sense providing the user with negative information (i.e., their friends who were given impressions but did not click and hence showed no interest in the product) since the idea is to encourage users to click and hopefully purchase. However to show the flexibility of our model we will use $\alpha = \beta = 0.25$. Note that the potential gains will therefore be less than if we had used $\beta = 0$. Therefore if all of my friends were given impressions and clicked then my probability of clicking increases to 0.5. If, on the other hand, all of my friends were given impressions and did not click then I too will not click because my click probability becomes 0. Note that the case $m_1 = 4$ is a single stage problem and the 4 impressions can be given to any 4 users and so the expected number of clicks is $0.25 \times 4 = 1$.

Let us now consider the case $m_1 = 1$. In this case we can see that the expected value will be the same for the cases A and D and similarly for the cases B, C, E and F because of symmetry. Suppose an impression is provided to A. If A clicks then the probability vector in stage 2 becomes $[0, 1/3, 1/4, 1/4, 1/4, 1/3]$. In the second stage 3 impressions are available and so we can assign them to B, C and F for an expected click total of 11/12 + 1 (for A). If A did not click then the probability for the second stage is $[0, 1/6, 1/4, 1/4, 1/4, 1/6]$ and we can assign the impressions

TABLE I. OPTIMAL VALUES FOR $m_1 = 2$

Pair	AB	AC	AD	BC	BE	BF
Value	97/96	187/192	47/48	1	1	179/192

TABLE II. OPTIMAL VALUES FOR $m_1 = 3$

Triplet	ABC	ABD	ABE	ABF	ACE	BCE	BEF
Value	1.014	0.995	1.014	1.010	0.906	0.992	0.992

to C, D and E for a expected click total of 3/4. The expected click total is therefore given by $0.25(23/12) + 0.75(3/4) = 25/24$. Suppose that the impression is instead given to B then if B clicks the probability vector for the second stage becomes $[3/8, 0, 1/3, 1/4, 1/3, 1/4]$ and so we can choose A, C, E in stage two for a value of 25/24 + 1. If B does not click then the probability vector becomes $[1/8, 0, 1/6, 1/4, 1/6, 1/4]$ and we obtain an optimal second stage total of 2/3. Hence in this case we get $0.25(49/24) + 0.75(2/3) = 97/96$. Therefore the value is larger if A (or D) is given the impression in stage 1 and the optimal value is 25/24.

Next we consider the case $m_1 = 2$. Again because of symmetry we only need to allocate impressions to the following pairs AB, AC, AD, BC, BE, BF. If impressions are allocated to A and B then the possible outcomes are (11, 10, 01, 00) where 1 corresponds to a click. For 11 the resulting probability vector is $[0, 0, 1/3, 1/4, 1/3, 1/3]$ and so the second stage value is 2/3 (two impressions). For 10 we get $[0, 0, 1/6, 1/4, 1/6, 1/3]$ resulting in a second stage value of 7/12. For 01 we get a probability vector of $[0, 0, 1/3, 1/4, 1/3, 1/6]$ for a value of 2/3. For 00 we get $[0, 0, 1/6, 1/4, 1/6, 1/6]$ for a value of 5/12. Hence the expected value is given by

$$\frac{1}{16}(2/3+2)+\frac{3}{16}(7/12+1)+\frac{3}{16}(2/3+1)+\frac{9}{16}(5/12) = \frac{97}{96}$$

Similar calculations can be performed for the other cases. The resulting optimal values are provided in Table I. Therefore the optimal expected click total is achieved when impressions are given to A and B with a value of 97/96.

Finally consider the case $m_1 = 3$. In this case there are 20 different combinations for providing impressions and for each of these we must evaluate the last stage for eight different possible outcomes. We can eliminate some of these options because of symmetry and so we only evaluate the optimal solutions for the triplets provided in Table II. We find that the optimal value of 1.014 is obtained for the solution in which impressions are given to A, B and C (or any similar combinations).

In Table III we provide the optimal value for the different split of impressions across stages. The net result is that it is optimal to allocate 1 impression in stage 1 and 3 in stage 2. The impression in stage 1 should be given to either A or D. Once this allocation is made we then await the outcome to determine the optimal allocation for the last stage. Note that, although we included the negative effects of non-clicking, there is still a 4% increase for the two stage case compared to the single stage case.

TABLE III. OPTIMAL VALUE VERSUS m_1

m_1	0	1	2	3	4
Optimal Value	1	1.042	1.010	1.014	1

V. A SIMPLE HEURISTIC

As one can see from the illustrative example in the previous section. The computational complexity for the solution of this problem grows rapidly with the number of users, the number of stages, the number of impressions and the number of friendship pairs. In practice there can be millions of users and hundreds of impressions and so it is not feasible to optimally solve the problem for real-world scenarios. In this section we consider a simple heuristic that can provide acceptable performance but which is computationally much faster and hence practical.

Suppose that at some stage we have n users who have not yet been given an impression and m impressions to be assigned. We first search over all users and find the user u_1 whom, if given an impression provides the largest expected number of total clicks (with no other impressions allocated). We allocate an impression to this user and we then find another user u_2 which, if given an impression together with u_1, provides the largest expected number of total clicks. We repeat this process until we obtain user u_m. We then assign the m impressions to these m users. This process is repeated at each stage. Note that it is optimal for the final stage.

For example, consider the illustrative example from the previous section with $m_1 = 3$. The first user u_1 would be A (or D) since this single user provides the largest expected number of total clicks. We next find another user which together with A will give the maximum expected number of total clicks. For this example this will be user B (see Table I). Finally we repeat to find the third user which will be user C (see Table II). Similarly for the cases $m_1 = 1$ and $m_1 = 2$. In this particular example the heuristic solution happens to be the optimal solution but this may not be the case in general. In future work we plan to investigate how far from optimal the heuristic solution lies.

Next we compute the computational complexities of the optimal solution and this sub-optimal heuristic. We will find the number of single stage problems that needs to be solved in each case. For the optimal case we must evaluate $\binom{n}{m}$ possible sets of impression allocations. For each of these we must then evaluate all 2^m possible outcomes and solve the single stage problem for each. Therefore the number of single stage problems that must be solved is given by

$$C_{opt} = \binom{n}{m} 2^m. \tag{7}$$

For the heuristic case, we first must solve 2 single stage problems (i.e. whether user clicks or did not click) for each of the n users. We then must search over $n - 1$ users and each time evaluate 2^2 possible combinations (for the two users). This is repeated until m users are chosen. Hence in this case the number of single stage problems that must be solved is given by

$$C_{sub} = \sum_{k=1}^{m} (n-k+1)2^k = 2^{m+1}(2+n-m) - 2n - 4. \quad (8)$$

Assuming that m is some small fraction of n then for large n we find

$$\frac{C_{opt}}{C_{sub}} = \mathcal{O}\left(\frac{1}{n}\binom{n}{m}\right), \quad (9)$$

which grows very rapidly with n. Hence the proposed heuristic can provide significant computational savings.

VI. Conclusions and Future Work

We addressed the problem of optimizing the revenue derived from advertising in an online social network. Our formulation is different to past work in that it is based on a stochastic dynamic programming model. We also showed that, because of the computation complexity of obtaining the optimal solution, heuristics are needed.

The focus of this paper was on providing the optimization model and demonstrating, through a simple example, the potential gains one can achieve by using this multi-stage approach with feedback as opposed to a single stage optimization problem. We next plan to develop and implement a simulation platform of a real-world OSN and demonstrate the gains that can be achieved by using simple, practical heuristics, as the one proposed in this paper.

References

[1] (2013) Iabinternet advertising revenue report. [Online]. Available: http://www.iab.net/media/file/ IABInternetAdvertisingRevenueReport-tHY2013FINALdoc.pdf

[2] (2014) Facebook doubleclick for publishers (dfp) optimization website. [Online]. Available: https://www.facebook.com/business/a/online-sales/ad-optimization-measurement

[3] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '03. New York, NY, USA: ACM, 2003, pp. 137–146.

[4] E. Bakshy, D. Eckles, R. Yan, and I. Rosenn, "Social influence in social advertising: Evidence from field experiments," in *Proceedings of the 13th ACM Conference on Electronic Commerce*. ACM, 2012, pp. 146–161.

[5] H. Bao and E. Y. Chang, "Adheat: An influence-based diffusion model for propagating hints to match ads," in *Proceedings of the 19th International Conference on World Wide Web*, ser. WWW '10. New York, NY, USA: ACM, 2010, pp. 71–80. [Online]. Available: http://doi.acm.org/10.1145/1772690.1772699

[6] S. Bhagat, A. Goyal, and L. V. Lakshmanan, "Maximizing product adoption in social networks," in *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, ser. WSDM '12. New York, NY, USA: ACM, 2012, pp. 603–612. [Online]. Available: http://doi.acm.org/10.1145/2124295.2124368

[7] E. Even-Dar and A. Shapira, "A note on maximizing the spread of influence in social networks," in *Proceedings of the 3rd International Conference on Internet and Network Economics*, ser. WINE'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 281–286. [Online]. Available: http://dl.acm.org/citation.cfm?id=1781894.1781928

[8] D. Saez-Trumper, Y. Liu, R. Baeza-Yates, B. Krishnamurthy, and A. Mislove, "Beyond cpm and cpc: Determining the value of users on osns," in *Proceedings of the Second ACM Conference on Online Social Networks*, ser. COSN '14. New York, NY, USA: ACM, 2014, pp. 161–168. [Online]. Available: http://doi.acm.org/10.1145/2660460.2660477

[9] A. Goyal, F. Bonchi, and L. V. S. Lakshmanan, "A data-based approach to social influence maximization," *Proc. VLDB Endow.*, vol. 5, no. 1, pp. 73–84, Sep. 2011. [Online]. Available: http://dx.doi.org/10.14778/2047485.2047492

[10] J. Hartline, V. Mirrokni, and M. Sundararajan, "Optimal marketing strategies over social networks," in *Proceedings of the 17th International Conference on World Wide Web*, ser. WWW '08. New York, NY, USA: ACM, 2008, pp. 189–198. [Online]. Available: http://doi.acm.org/10.1145/1367497.1367524

[11] S. Lei, S. Maniu, L. Mo, R. Cheng, and P. Senellart, "Online influence maximization," in *Proc. KDD*, Sydney, Australia, Aug. 2015.

[12] W. Chen, C. Wang, and Y. Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '10. New York, NY, USA: ACM, 2010, pp. 1029–1038. [Online]. Available: http://doi.acm.org/10.1145/1835804.1835934

[13] K. Jung, W. Heo, and W. Chen, "Irie: Scalable and robust influence maximization in social networks," in *Data Mining (ICDM), 2012 IEEE 12th International Conference on*. IEEE, 2012, pp. 918–923.

[14] C. Borgs, M. Brautbar, J. T. Chayes, and B. Lucier, "Influence maximization in social networks: Towards an optimal algorithmic solution," *CoRR*, vol. abs/1212.0884, 2012. [Online]. Available: http://arxiv.org/abs/1212.0884

[15] Y. Tang, X. Xiao, and Y. Shi, "Influence maximization: Near-optimal time complexity meets practical efficiency," in *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '14. New York, NY, USA: ACM, 2014, pp. 75–86. [Online]. Available: http://doi.acm.org/10.1145/2588555.2593670

[16] C. Aslay, N. Barbieri, F. Bonchi, and R. A. Baeza-Yates, "Online topic-aware influence maximization queries." in *EDBT*, 2014, pp. 295–306.

[17] A. Goyal, W. Lu, and L. V. Lakshmanan, "Celf++: Optimizing the greedy algorithm for influence maximization in social networks," in *Proceedings of the 20th International Conference Companion on World Wide Web*, ser. WWW '11. New York, NY, USA: ACM, 2011, pp. 47–48. [Online]. Available: http://doi.acm.org/10.1145/1963192.1963217

[18] A. Goyal, F. Bonchi, and L. V. Lakshmanan, "Learning influence probabilities in social networks," in *Proceedings of the third ACM international conference on Web search and data mining*. ACM, 2010, pp. 241–250.

[19] Y. Wang, G. Cong, G. Song, and K. Xie, "Community-based greedy algorithm for mining top-k influential nodes in mobile social networks," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2010, pp. 1039–1048.

[20] Z. Abbassi, A. Bhaskara, and V. Misra, "Optimizing display advertising in online social networks," in *Proceedings of the 24th International Conference on World Wide Web*, ser. WWW '15. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2015, pp. 1–11. [Online]. Available: http://dx.doi.org/10.1145/2736277.2741648