2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining

HyperCubeMap: Optimal Social Network Ad Allocation Using Hyperbolic Embedding

Hui Miao*, Peixin Gao†, MohammadTaghi Hajiaghayi*, John S. Baras†
*Dept. of Computer Science, †Inst. for Systems Research and Dept. of Electrical & Computer Engineering
University of Maryland, College Park, MD 20742
*{hui, hajiagha}@cs.umd.edu, †{gaopei, baras}@umd.edu

***Abstract*—Advertising activity on SNS has grown rapidly and is now a billion dollar business. In the SNS advertising model, the SNS serves as the advertising agent, and takes the advantage of network diffusion to attract advertisers and charges for the cascading impressions. The optimal ad allocation task is to choose the ad allocation plan that maximizes the revenue. Each user has different diffusion ability, limited daily impressions and the advertisers have various bidding prices and budget concerns. A feasible plan that obeys the constraints is difficult to find. The solution of this problem lies in the space of $\mathbb{N}_0^{|Ads| \times |User|}$, which makes direct optimization unattractive. In this paper, we study SNS advertising business models, formulate the SNS ad allocation problem and show their connections with hyperbolic embedding. We develop a new embedding algorithm HYPERCUBEMAP that allows for dimension reduction. Our proposed method reduces the dimensionality of the original problem significantly, runs two to four orders of magnitude faster, and reaches 95% of the optimum.**

I. INTRODUCTION

Social network sites (SNS) such as Facebook, Google+ and Twitter have attracted hundreds of millions of daily users since their appearance. In modern SNS, users expose many personal behaviors and connect to each other based on real world relationships, which makes SNS ideal for targeted advertising [1]. SNS advertising has grown rapidly in the past years, for example, Facebook has more than 1 million advertisers and 100 billion hits per day [2], [3]. As shown in Fig. 1, to perform a marketing campaign in an SNS such as Facebook, an advertiser first finds an agent (typically the SNS site), chooses the target audience by specifying desirable user profiles and provides its advertisements (ads) with a bidding price and a budget. Then the agent allocates the ads to the set of users whose profiles match its targeting request. For each impression (page view) of a user, the agent chooses one or several ads whose target audience include the user. Now the user can see and engage with the ad, e.g. 'like' in Facebook , and then her friends may see the ad and further engage. For advertising campaigns, instead of keywords, advertisers bid for a target group of users' actions, which can be mille impressions (cost per thousand impressions), engagements (e.g. click, retweet), or actions (e.g. app installation, product purchase). The agent runs large auctions using the bids and charge advertisers by the user actions. There are associated billing policies, such as pay-per-mille, pay-per-click, pay-per-action, pay-per-engagement [4]. The pay-per-mille is the default and most popular policy in Facebook, and we assume this policy throughout this paper.

Research partially supported by Maryland Procurement Office contract H9823014-C-0127, by US Air Force Office of Scientific Research MURI grant FA9550-10-1-0573, by National Science Foundation grant CNS1035655, and by National Institute of Standards and Technology grant 70NANB11H148.

The SNS ad allocation problem of maximizing the agent's revenue by allocating ads to user impressions while respecting the advertisers' requests (targeting, bidding and budget constraint), is a central problem for advertising agents. The concept of *paid social influence* distinguishes the problem from standard ad allocation (AdWords [5]) and influence maximization problem [6]. Comparing with AdWords in which advertisers bid on search query keywords, in the SNS, they bid on users. Moreover, as the substantial role of information diffusion in SNS [7], the users allocated to a particular ad are allowed to engage with the ad and diffuse it to their neighbors, while advertisers pay for all the impressions. On the other hand, the problem differs from influence maximization problem where one only pays the best initial user set of size N to maximize the total users she can reach by cascading.

A. SNS Ad Allocation Problem

To formulate this problem, let A denote the set of advertisers, and U be the set of users. Each user $u \in U$ has a daily impression I_u and a social influence function $P(u)$. Each advertiser $A_i \in A$ has a target user group $t_i \subseteq U$ defined by user attributes, a budget b_i and bidding price p_i. Without losing generality, we assume each advertiser has only one ad, and on a user's one impression, only one ad is displayed in the sponsor pane. Note that in the news feed, her friends' ad engagement is treated as common friends' updates and displayed anyway.

Example 1: In Fig. 1, the impression of *Alice*, $I_{u=\text{'Alice'}} = 4$, shows how many times she views a refreshed Facebook page per day, while her social influence, $P(\text{'Alice'})$, means how many users will see it eventually when she likes an ad. On the other hand, *ad1*'s target user group t_{ad1} are all graduate students in *US*, her bid is $p_{\text{ad1}} = \$5$, and budget is $b_{\text{ad1}} = \$200$.

The solution of the allocation problem decides for each ad what is the initial set of users to be displayed by con-

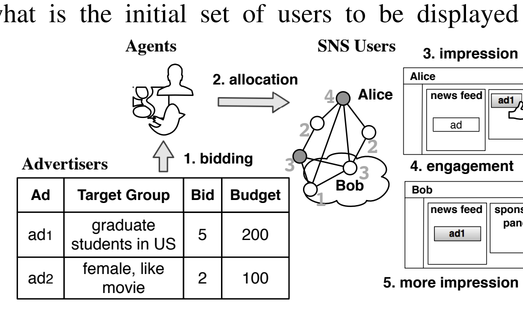


Fig. 1. SNS Ad Campaign and Allocation

ASONAM '15, August 25-28, 2015, Paris, France

DOI: http://dx.doi.org/10.1145/2808797.2809397

sidering their influence capability. Let the decision variable be $X \in \mathbb{N}_0^{|A|\times|U|}$. For each u and A_i, one dimension of the decision variable $x_{u,i}$ represents how many impressions of u to be assigned to A_i. The optimization problem is to find the allocation that maximizes the agent's total revenue. Consider users' influence, it can be formulated as an integer program:

$$\begin{aligned} \max_X \quad & \sum_{A_i \in A} p_i \sum_{u \in t_i} x_{u,i}(1+P(u)) && \text{(revenue)} \\ \text{subject to} \quad & p_i \sum_{u\in t_i} x_{u,i}(1+P(u)) \le b_i \quad \forall A_i \in A && \text{(budget)} \\ & \sum_{A_i \in A} x_{u,i} \le I_u \quad \forall u \in U && \text{(impression)} \end{aligned} \tag{1}$$

To define $P(u)$, as the engaged ads are shown in her friends' news feed, we assume her friend can always see the engagement if it happens when visiting her news feed. The reasons are two folds, a) recent Facebook study [8] shows a user's newsfeed post can be read by 35% of their friends, and 61% of them over a month, b) the news feed is ordered by proprietary ranking algorithm [9], which may treat ads and posts differently. We also assume 1-hop influence, as w is often small (0.3%) in real SNS and network cascading is known to be shallow in general [10], [11]. Accordingly, the social influence function $P(u)$ of user u can be defined as:

$$P(u) = \sum_{\nu \in F_u} w \min\{I_u, I_\nu\} \tag{2}$$

with w the expected engagement probability (click-through rate). The $\min\{I_u, I_\nu\}$ in Eq. 2 means regardless the engagement probability, the user u's engagement can be seen by her friend $\nu \in F_u$, $\min\{I_u, I_\nu\}$ times. If $I_u > I_\nu$, it is bounded by the daily impression of her friend ν. If $I_u \le I_\nu$, the user u at most engages I_u times. Given more SNS constraints, $P(u)$ can be adjusted, and we will discuss more in Sec. V.

B. *Our Techniques and Contributions*

The decision variable $X \in \mathbb{N}_0^{|A|\times|U|}$ lies in high dimension as much as 10^{16} when considering 1 million advertisers and billion users daily in Facebook. This makes the direct optimization impractical. To make it more tractable, we propose an approximation scheme, HyperCubeMap. The key idea of our method exploits the target group concept in the problem by using hyperbolic embedding. Notice that in an advertiser A_i's target group t_i, all users are considered the same by the ad, the only difference are their influence capabilities. If we can approximate the user impression allocation for A_i and revenue calculation with influence on the target group level rather than the user level, we will be able to eliminate several orders of magnitude of dimensions for the problem. For 10^9 users and $10^{3\sim6}$ categories in a real world SNS such as Facebook [12], we can reduce the dimension around $10^{3\sim6}$.

Hyperbolic embedding is a geometric mapping from a complex network $G(V, E)$ to a set of points and segments in a hyperbolic space $\mathbb{D}$. The hyperbolic space is continuous and hyperbolic embedding maps arbitrary size complex networks into a bounded area where each node is assigned a coordinate. If SNS is embedded properly, we could use regions in the hyperbolic space to express a set of users allocated to an ad; then we could approximate the revenue from A_i as an integral of the user's influence function over a certain region $R_i \subset \mathbb{D}$:

$$p_i \sum_{u\in t_i} x_{u,i}(1+P(u)) \cong p_i \int_r \int_\theta I(R_i(r,\theta)) d\theta dr \tag{3}$$

As we can use very few variables to describe a set of users assigned to A_i with regular shapes such as fan and ring, we can reduce dimensions of the original problem significantly.

Example 2: In Fig.2(a), two shapes on the base area represent two target groups w.r.t. Fig.1. Each user is mapped to a coordinate in her group, e.g. *Alice* is assigned to the fan. The influence function defines the top surface. Note the impressions function defines a different surface that is not shown.

Our contributions in the paper include the following: **a)** We study the SNS advertising pay-per-mille impression model used widely in modern SNS and formulate the the SNS ad allocation problem. **b)** We show the connection between hyperbolic embedding and the SNS ad allocation problem, and propose a novel and efficient embedding method, HyperCubeMap, to group users with different profiles onto a Poincaré disc. **c)** We approximate the problem as a hyperbolic space region allocation problem and propose a novel optimization framework to utilize HyperCubeMap embedding, which reduce the dimensionality of the original problem significantly, and show 2 to 4 orders of magnitude runtime improvement while reaching 90% of the optimum of baseline IP problem.

II. Hyperbolic Embedding for SNS Ad Allocation

A. *Preliminary: Hyperbolic Space and Complex Networks*

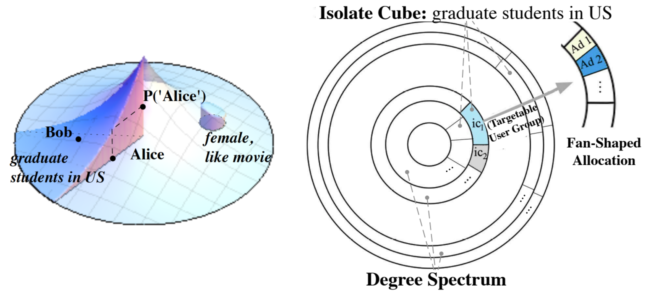
Hyperbolic space is a non-Euclidean geometrically smooth space that generalizes the idea of Riemannian manifolds with negative curvature. In our formulation, we use Poincaré disc model, $\mathbb{D} = \{z \in \mathbb{C} \mid |z| < 1\}$. The connection between complex network and hyperbolic space lies in the Gromov's δ-hyperbolicity of a metric space [13]. By assuming a hyperbolic geometry underlies complex networks, Krioukov et al. [14]–[16] study the connection between topology of hyperbolic geometry and the characteristics of complex networks. They show that power law degree distributions and strong clustering in complex networks can be viewed as reflection of the negative curvature property of the underlying hyperbolic geometry. They design a mapping between hyperbolic space and complex networks, which can accommodate arbitrary size network and successfully captures important features in complex networks, i.e., small world effect, scale-free and community structure.

B. HyperCubeMap*: Hyperbolic Embedding Method*

As advertisers bid on heterogeneous user groups and users have different impressions and influence capability. To apply the embeded SNS for dimension reduction via integration, the hyperbolic embedding should have the following properties:

1) Both node density and degree distribution should be well-defined along angular and radial axises to support integrals.
2) The social influence function defined on the user's coordinate (r, θ) should be continuous on the Poincaré disc $\mathbb{D}$, otherwise the column with the related surface is not integrable.
3) The embedding method should embed users within the same targeting group onto connected regions, otherwise a user group have to be described by a collection of discrete points and the dimension reduction would not be achieved.

To the best of our knowledge, the existing embedding methods [14]–[17] do not obey all of these prerequisites. In [15], Papadopoulos et al. propose an embedding scheme



(a) Poincaré Disc Example (b) Isolate Cubes and Degree Spectrum

Fig. 2. Hyperbolic Embedding

satisfying the first prerequisite, the node density and the expected value of degree along the radial coordinate is well-defined. However, their model does not help SNS ad allocation as the target groups, impression, and influence are not taken into considerations. By extending [15], we propose HYPERCUBEMAP embedding algorithm. It first ensures the node density and degree are well-defined along radical axis:

$$\begin{aligned} \rho(r) &= ae^{r} && \text{(node desnity)} \\ \omega(r) &= ce^{-r/2} && \text{(degree distribution)} \end{aligned} \tag{4}$$

a and c are constants derived from embedding; Then it organizes the Poincaré disc into dimension reduction feasible cubes (Sec. II-B1) and calculates the minimal set of groups to boost dimension reduction effectiveness (Sec. II-B2). In Sec. II-B3, we give the whole embedding algorithm which satisfies all three prerequisites. To improve precision and ease tuning, we propose uniform node density embedding (Sec. II-B4).

1) Isolate Cubes and Degree Spectrum: To let the advertisers specify the target user group, the agent often provides a set of categorical filters, each of which has fixed number of options. In Facebook's case, there are seven major user attributes for filtering, e.g. location, gender, age, language and interests. The target user group of a campaign is the selection of given options. The size of option profiles is not very large, e.g. Facebook has common option profiles bounded by 10^6 and discourages too fine-grained filters by giving warnings [12].

To capture these aspects, we propose the concept of **isolate cube** to express user groupings, and **degree spectrum** to divide Poincaré disc into finer and more regular shapes which ease the calculation and improve the precision, as shown in Fig. 2(b).

Definition 1: Isolate Cube: An isolate cube is a set of unit targetable user groups having the same set of campaigns.

Users in the same isolate cube are shared by the same set of campaigns. Any two users in an isolate cube are interchangeable in an allocation solution. As the isolate cubes are related to dimension reduction performance, the fewer isolate cubes we have, the better performance we can benefit from the embedding. In the worst case, considering the ad platform that defines F categorical filters, and each $f \in F$ has v_f distinct options, there are at most $\prod_{f\in F} v_f$ isolate cubes.

As one can envision, the population in each user group may vary a lot, not to mention the degree distributions in each of them. When embedding, it means different isolate cube can results in very different shapes in a Poincaré disc. To make the embedding useful and ensure accuracy, we introduce the concept of *degree spectrum* to regularize the embedding shape.

Definition 2: Degree Spectrum: A degree spectrum, Λ, is a series of annuli centered at $(0, 0)$ on 2-D Poincaré disc. Each annulus $\lambda \in \Lambda$ with radius (r_s, r_e), represents all the users with degree in the range of $(\omega(r_s), \omega(r_e)]$.

As shown in Fig. 2(b), the annuli are the degree spectrum. Within each annulus, isolate cubes are allocated in fans whose area is proportional to the number of users in an isolate cube. Each advertiser A_i targets at a set of isolate cubes, IC_i, each of which has locations in some or all annuli in the spectrum Λ, thus the allocation is represented by at most $|IC_i| \cdot |\Lambda|$ dimensions for A_i comparing with $|\{u|u \in t_i\}|$ in Eq. 1. Note $|\Lambda|$ is a tuning parameter of our method, which can be tuned by fixing the degree range d. In the extreme case, $d = 1$, each annulus only contains the users with the same degree.

2) Optimal Isolate Cubes: As the size of isolate cubes is important for the dimension reduction performance, we show how to get the minimal set of isolate cubes. Assume the ad platform designs a set of filters F, where each $f \in F$ has a set of possible values, v. Each advertiser A_i selects targeting values (f, v_i) for each filter, denoted by $O_i = \{(f, v_i)|f \in F\}$, which defines a set of target users $T_i = \{u|(f, v_i) \in O_i, u[f] \in v_i\}$. Given all advertisers A and their targeting profiles O, we can cluster them together and derive the *optimal isolate cubes* (*opt_ic*), which is the smallest set of isolate cubes and gives the best dimension reduction performance. We propose a hashing based approach to derive the *opt_ic* in $\mathcal{O}(O)$ time. By scanning O, for each filter value (f, v_i), we build a signature based on the set of advertisers bid it. Then we scann all (f, v_i) and combine those share the same signature to get the *opt_ic*.

3) Embedding Algorithm: In Alg. 1, we give the hyperbolic embedding algorithm HYPERCUBEMAP. Given a SNS $G(U, E)$, advertisers A, targeting profile O and a spectrum design Λ, it places each user $u \in U$ to (r_u, θ_u). It first generates the optimal isolate cubes *opt_ic*, and then for each spectrum annulus $\lambda(r_s, r_e)$, it assigns each $ic \in opt\_ic$ an angular coordinate range (θ_s, θ_e). To ensure the uniform node density along angular axis, the range is proportional to the ic's target user size portion in this spectrum annulus. To ensure the same targetable user groups are allocated together, we assign the angular coordinate of each user in its associated isolate cube ic. β is a mitigating factor determined by the power law exponent γ: $\beta = \frac{1}{\gamma}$; γ can be found for a given network. The algorithm complexity is linear given users sorted by degree.

Algorithm 1 HYPERCUBEMAP

Let *opt_ic* be the Optimal Isolate Cube
Let each annulus $\lambda(r_s, r_e) \in \Lambda$ and its user size be N_a
$\theta_s = 0$
for each $\lambda(r_s, r_e) \in \Lambda$ **do**
  for each $ic \in opt\_ic$ **do**
    Let isolate cube ic's user size be $ic.n_a$
    $\theta_e = \theta_s + 2\pi \cdot ic.n_a/N_a$
    Let ic's angular range $ic_{ang}[\lambda] = (\theta_s, \theta_e)$
    $\theta_s = \theta_e$
  end for
end for
Sort U by degree in descending order $d_1 > d_2 > \cdots > d_n$ and break ties arbitrarily. Let u's degree be d_u
Let $r_1 = 0$, and θ_1 is chosen randomly in $[0, 2\pi]$
for u from 1 to $n-1$ **do**
  Let $r_u = 2\beta \log u + 2(1-\beta)\log n$
  Find spectrum $\lambda'(r_s, r_e)$, satisfying $r_u \in \lambda'(r_s, r_e)$
  Find isolate cube ic satisfying $u[f] \in v_{ic}, \forall (f, v_{ic}) \in ic$
  Let u's angular coordinate θ_u be chosen randomly from $ic_{ang}[\lambda']$
end for

HyperCubeMap produces an embedding that satisfies our prerequisites. On the Poincaré disc, the degree distribution and node density are well defined along angular and radial axises, node density $\rho(r) = ae^r$, and correspondingly expected degree $\omega(r) = ce^{-r/2}$, $r \in [0, R)$. The same targeting group users are embedded into connected regions. With a continuous social influence function, we can reformulate the problem to a region allocation problem with much fewer dimensions.

4) Uniform Node Density Embedding: From Alg. 1 we can notice that the inner area of the Poincaré disc is very sparse due to the exponential node density along the radius, which effects the optimality of approximation and makes it difficult for parameter tuning and allocation scheme design. Thus, corresponding to our application scenario, we propose to adjust the node density and make it uniform along the radius by moving nodes inside as an alternative embedding method. For a point at (r, θ) by Alg. 1, its new coordinate is (r', θ), then the uniform density function $\rho'(r')$ can be derived as follows:

$$\rho'(r') = \frac{\int_0^R \int_0^{2\pi} \rho(\tau) d\theta d\tau}{\pi R^2} = \frac{2a(e^R - 1)}{R^2} \qquad (constant) \tag{5}$$

the mapping $\psi(r)$ from r to r' is:

$$r' = \psi(r) = \sqrt{R^2 \cdot \frac{(e^r - 1)}{(e^R - 1)}} = R\sqrt{\frac{e^r - 1}{e^R - 1}} \tag{6}$$

Thus the expected degree at new coordinate (r', θ) is:

$$\omega'(r') = \omega(r)|_{r=\psi^{-1}(r')} = \frac{cR}{\sqrt{r'^2 e^R - r'^2 + R^2}} \tag{7}$$

III. Region Allocation in Hyperbolic Space

A. General Optimal Region Allocation

Using HyperCubeMap, we can reformulate the ad allocation problem as region allocation problem on the 2-D Poincaré disc $\mathbb{D}$. On $\mathbb{D}$, each user $u \in U$ with impression $I_u \in I$ is placed at (r_u, θ_u) in polar coordinates with expected degree $d_u = \omega(r_u) = ce^{-r_u/2}$ by HyperCubeMap. Given a set of advertisers A, each $A_i \in A$ has a budget b_i and bidding price p_i on a set of bidding isolate cubes (target users) denoting as $T_i = \{ic_1, ic_2, ..., ic_n\}$. $T = \cup T_i$ are the optimal isolate cubes (opt_ic). Given the degree spectrum Λ, the allocation profile for A_i is defined as $S_i = \{S_i^{\lambda,c} | \lambda \in \Lambda, c \in T\}$, where λ denotes the annulus, c specifies the isolate cube. S_i is a set of fans $\{(\theta_{i,s}^{\lambda,c}, \theta_{i,e}^{\lambda,c})\}$, each of which describes how to allocate users in an isolate cube c on a degree spectrum annulus λ. The optimal region allocation problem is to derive an allocation profile S for A to maximize the revenue of the agent while respecting the budget and impression constraints:

$$\begin{aligned} &\max_S \sum_{A_i \in A} p_i f_i(S, I) \\ &\text{subject to } S_i \subset T_i && \forall A_i \in A \\ &\quad 0 \le p_i f_i(S, I) \le b_i && \forall A_i \in A \\ &\quad \phi_u(S, I) = \sum_{S_i \in S} \phi_u(S_i, I) \le I_u && \forall u \in U \\ &\quad \theta_e^{\lambda,c} \ge \theta_{i,e}^{\lambda,c} \ge \theta_{i,s}^{\lambda,c} \ge \theta_s^{\lambda,c} && \forall c \in T, \lambda \in \Lambda \end{aligned} \tag{8}$$

where $f_i(S, I)$ is A_i's actual sum of impressions considering social influence. $\phi_u(S_i, I)$ is u's impressions assigned to A_i.

As we can see, comparing with the original optimization problem, now a set of users is assembled as a fan shape on the Poincaré circle, which reduces the dimensions significantly. On the other hand, angular coordinates are continuous values instead of discrete values as before. If we can give closed forms for each ad A_i's assigned impressions $f_i(S, I)$ and each user u's allocated impression $\phi_u(S, I)$, then we can solve the problem directly. In order to do so, we need to specify how to incorporate with social influence, and address two major challenges: a) the impression distribution may not be well-defined and uncorrelated with degree, b) the overlapping fans. The first issue prevents us to apply integral, while the second one makes the optimization problem much more complicated.

1) Incorporating Social Influence: The actual impressions resulted from user u is different from I_u due to her social influence in the network. All exposed qualified impressions have a cost to the advertiser, thus actual profit of the agent is:

$$p_i \cdot I_u \cdot (1 + P(u)) \tag{9}$$

As discussed in Sec. I, u's influence function $P(u)$ can be defined using its 1-hop degree. After applying hyperbolic embedding, the influence function of user u is:

$$P(u) = P(r_u, \theta_u) = w \cdot d_u = w \cdot \omega(r_u, \theta_u) = w \cdot ce^{-r_u/2} \tag{10}$$

where w is a constant representing the engagement rate. Under uniform node density, the influence function $P'(u)$ is:

$$P'(u) = P(r'_u, \theta'_u) = w \cdot \frac{cR}{\sqrt{r'^2 e^R - r'^2 + R^2}} \tag{11}$$

which are both continuous functions, and can be used in integral to express $f_i(S, I)$ over the Poincaré disc.

B. Unit Impression Decomposition & Fan-shaped Allocation

The unknown user impression distribution in the hyperbolic embedding of the SNS significantly affects our formulation. Complex region intersection may not have an analytical expression or convexity. Also the unknown impression distribution forces us to discretize f_i and inevitably increase the complexity. To address these issues without introducing strong assumptions (e.g. disallow overlapping, enforce well-defined impression distributions), we extend the *Unit Impression Decomposition* optimization framework [18]. The main idea is to decompose the SNS into a series subgraphs where $u \in U$ has an impression $I_u = 1$, so there cannot be any intersections (i.e. one impression cannot be shared by advertisers). A sub step optimization can be conducted in each subgraph by adding a non-overlap constraint. Moreover, $f_i(S, I)$ can be formulated as $f_i(S_i)$, as the volume assigned to A_i is independent.

With the *Unit Impression Decomposition*, we can solve the original problem using a multi-stage optimization process. It finishes when all impressions are allocated or all budgets are used. In the mth stage, given the unit impression graph $G^{(m)}$, we apply HyperCubeMap to embed $G^{(m)}$ in the hyperbolic space. For each advertiser $A_i \in A^{(m)}$ whose budget $b_i^m > 0$, the sub-step of the optimization problem is given in Eq. 12.

$$\begin{aligned} &\max_{S^{(m)}} \sum_{A_i \in A} p_i f_i(S_i^{(m)}) \\ &\text{subject to } \quad S_i^{(m)} \subset T_i^{(m)} && \forall A_i \in A^{(m)} \\ &\quad 0 \le p_i f_i(S_i^{(m)}) \le b_i^{(m)} && \forall A_i \in A^{(m)} \\ &\quad S_i^{(m)} \cap S_j^{(m)} = \emptyset && \forall A_i, A_j \in A^{(m)} \wedge i \ne j \\ &\quad \bigcup_{A_i}^{A^{(m)}} S_i^{\lambda,c(m)} \subset S^{\lambda,c(m)} && \forall c \in T^{(m)}, \lambda \in \Lambda^{(m)} \end{aligned} \tag{12}$$

We then solve the non-overlapping problem stated in Eq. 12, and record its optimal solution $S^{(m)*}$ and optimal value $\sum_{A_i \in A} f_i(S_i^{(m)*})$, the budget vector is updated as

$b_i^{(m+1)} = b_i^{(m)} - p_i \cdot f_i(S_i^{(m)*})$, and the $m+1^{\text{th}}$ unit impression graph is generated with residual impressions and removing users with no impression left and their edges. It ends when all advertisers' budgets are used, or all impressions are exploited.

With exponential node density and degree distribution, the allocation $f_i(S_i^{\lambda,c})$ can be calculated as:

$$\begin{aligned} f_i(S_i^{\lambda,c}) &= f_i(\theta_{i,s}^{\lambda,c}, \theta_{i,e}^{\lambda,c}) = \int_{r_s^\lambda}^{r_e^\lambda} \int_{\theta_{i,s}^{\lambda,c}}^{\theta_{i,e}^{\lambda,c}} \rho(\tau)(1 + P(\tau,\theta)) d\theta d\tau \\ &= a \int_{r_s^\lambda}^{r_e^\lambda} e^{\tau}(1 + wce^{-\frac{\tau}{2}}) \int_{\theta_{i,s}^{\lambda,c}}^{\theta_{i,e}^{\lambda,c}} d\theta d\tau = \Delta_\lambda \theta_i^{\lambda,c} \end{aligned} \tag{13}$$

where $\Delta_\lambda = a(2wce^{\frac{r_e^\lambda}{2}} - 2wce^{\frac{r_s^\lambda}{2}} + e^{r_e^\lambda} - e^{r_s^\lambda})$ is a constant related to the annulus λ and $\theta_i^{\lambda,c} = \theta_{i,e}^{\lambda,c} - \theta_{i,s}^{\lambda,c}$ is the angle range of the region $S_i^{\lambda,c}$. Now the function $f_i(S_i^{\lambda,c})$ is actually a linear function of $\theta_i^{\lambda,c}$, irrelevant to its start and end angles.

If we apply the uniform node density transform, the volume f_i can be calculated with a different boundary $(r_s'^\lambda, r_e'^\lambda)$, then:

$$f_i(S_i^{\lambda,c}) = \int_{r_s'^\lambda}^{r_e'^\lambda} \int_{\theta_{i,s}^{\lambda,c}}^{\theta_{i,e}^{\lambda,c}} \rho'(\tau')(1 + P'(u)) d\theta d\tau' = \Delta'_\lambda \theta_i^{\lambda,c} \tag{14}$$

where $\Delta'_\lambda = a(2wce^{\frac{\psi^{-1}(r_e'^\lambda)}{2}} - 2wce^{\frac{\psi^{-1}(r_s'^\lambda)}{2}} + e^{\psi^{-1}(r_e'^\lambda)} - e^{\psi^{-1}(r_s'^\lambda)})$, still a constant related to λ and Λ, and f_i is linear.

Combining the unit impression decomposition and fan-shaped allocation strategy , we can elaborate the optimal region allocation problem in Eq. 12 as a linear program:

$$\begin{aligned} \max_{\Theta^{(m)}} \quad & \sum_{A_i \in A^{(m)}} p_i \sum_{\lambda \in \Lambda^{(m)}} \Delta_\lambda \sum_{c \in T_i^{(m)}} \theta_i^{\lambda,c(m)} \\ \text{subject to} \quad & p_i \sum_{\lambda \in \Lambda^{(m)}} \Delta_\lambda \sum_{c \in T_i^{(m)}} \theta_i^{\lambda,c(m)} \le b_i^{(m)} \qquad \forall A_i \in A^{(m)} \\ & \sum_{A_i \in A^{(m)}} \theta_i^{\lambda,c(m)} \le \theta_e^{\lambda,c(m)} - \theta_s^{\lambda,c(m)} \quad \forall c \in T_i^{(m)}, \lambda \in \Lambda^{(m)} \end{aligned} \tag{15}$$

where the decision variable $\Theta \in \mathbb{R}_{\ge 0}{}^{|A| \times |\Lambda| \times |T|}$. Δ_λ is the constant in Eq. 13. The uniform node density setting can be derived accordingly by replacing Δ_λ to Δ'_λ (shown in Eq. 14).

If the optimization stops after n stages, then the allocation of ad A_i is the aggregation of optimal solutions: $\cup_{k=1}^n S_i^{(k)*}$. Note that while in one iteration there is no overlap, the final aggregated regions do have overlaps, as each iteration is based on a different Poincaré disc. Comparing with Eq. 1, the dimensions of unknown Θ in our formulation in the worst case is $|A| \times |\Lambda| \times |T|$, which is the number of campaigns multiplied by the degree spectrum and the optimal isolate cubes, while the original problem has $|A| \times |U|$. The improvement is significant as $|A|$ is around one million [19], but $|U|$ is in billions.

IV. EVALUATION

We conduct experiments on HYPERCUBEMAP based region allocation formulation and the baseline IP formulation SNSIP on synthetic data using IBM CPLEX optimizer (v12.6). We implement the hyperbolic embedding algorithm mentioned in Sec. II and the unit graph impression optimization routine in Sec. III-B. We refer HEMBEXP to the linear program of exponential node density distribution in Eq. 15, and HEMBUNI to the one using uniform node density distribution in Eq. 14. As hyperbolic embedding is essentially an approximation algorithm through dimension reduction, our experiments aim at showing the advantages of hyperbolic embedding over the original IP formulation in terms of runtime, scalability and optimality. We also show tuning degree spectrum parameter d to trade-off between runtime and optimality. All experiments run on a linux server with two 2.66 GHz 6-core Xeon X5650 CPUs and 128G memory. The CPLEX is configured to utilize all 24 threads; for the IP, we fix the *MIPSearch* parameter to branch and cut. The time metric are in seconds via CPLEX timer representing actual CPU time used in the optimization.

Dataset Description: We construct our dataset using distributions observed from public available real world advertising datasets. On the advertiser side, we look at keyword bidding and budget distributions from the Yahoo! Webscope dataset A1 [20] and open advertising dataset collected from Google AdWords used in [21]. We find that campaign bidding prices fit well with lognormal distribution, and the advertiser budget follow Pareto distribution approximately. On SNS activity side, we use SNAP 2.2 [22] to generate power law networks by setting $\alpha = 2.2$. The real impression distribution of well-known SNS is not available to the public to the best of our knowledge, we assign daily impression to each node using a Poisson distribution. To cluster users with different profiles into targetable user groups with different sizes, we use $|GR| = 0.0005$ to represent the group/user ratio, and use a Dirichlet prior to generate a multinomial distribution over group size. We then embed the generated networks with default spectrum width $d = 10$. Finally, for bidding prices, we use $|AR| = 0.001$ as the advertiser/user ratio and use bipartite preferential attachment with two Zipfian distributions to represent the popularity. We vary the number of users from 10K to 100M, derive the optimal isolate cubes and summarize it in Table I. All data and codes are available online[1].

Experimental Results: We first show the runtime performance by varying network size in Fig. 3(a). In general, hyperbolic embedding methods HEMBEXP and HEMBUNI finish the optimization process two to four orders of magnitude faster than the baseline SNSIP. Besides runtime, hyperbolic methods require much less memory than the IP model. Network 50M and 100M cannot run under SNSIP as out of memory, while HEMBEXP and HEMBUNI only use 2G memory for network 100M due to the dimension reduction.

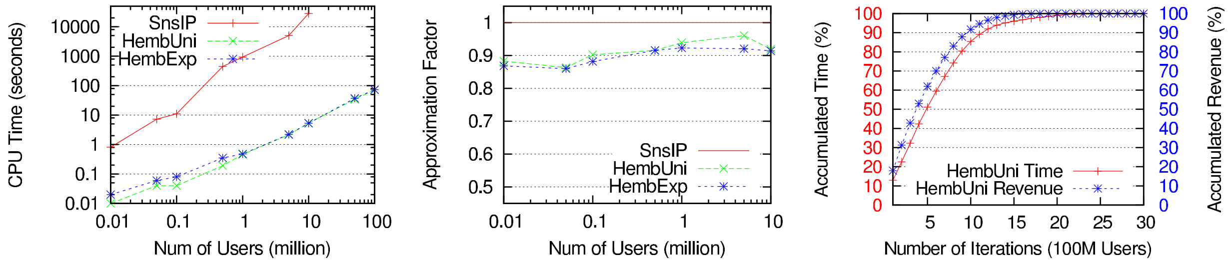
Next in Fig. 3(b), we show the optimality result using approximation factor P, for instance, in HEMBEXP case:

$$\mathrm{P}_{\text{HEMBEXP}} = \frac{\sum_{i=1}^{\max\_iter} OPT_{\text{HEMBEXP}}}{OPT_{\text{SNSIP}}} \tag{16}$$

As IP cannot run on 50M and 100M network, we omit those SNSIP data points. The solution of HEMBEXP and HEMBUNI reach about 90% of the original IP solution on average, and when network size increases, hyperbolic embedding methods have better solutions. In our experiments, the minimum value of P is 85.97% while the maximum is 96.07%. Also HEMBUNI always performs better than HEMBEXP with little cost. The exponential node density distribution makes the embedding coefficient less accurate in the center regions, where the users have high influence. If the engagement rate w becomes larger, the difference will become larger as well.

In Fig. 3(c), we show the accumulated revenue and time in the unit decomposition optimization process in the HEMBUNI

[1] http://www.cs.umd.edu/~hui/code/hypercubemap



(a) Run Time by Varying Problem Size (b) Approximation Factor (c) Accumulated Time and Revenue

Fig. 3. Evaluation Results of SNS Ad Allocation Formulations, SnsIP, HembExp and HembUni

$\|NU\|$	edge	$\|A\|$	ic	opt_{ic}	$\sum_u I_u$
5M	13.1M	5000	2500	1462	50M
10M	26.8M	10K	5000	2722	100M
50M	137.0M	50K	25K	11308	500M
100M	276.2M	100K	50K	21063	1B

TABLE I. Summary of Dataset

experiment on 100M network; HembExp has very similar performance. The left y axis in red is accumulated time percentage, and the right y axis in blue is accumulated objective value. Our optimization process spends most time on the early iterations which also contribute similar percentage in revenue.

Next we show the parameter tuning of our hyperbolic approach in Fig. 4. The degree spectrum width d affects dimension reduction directly and is independent with the SNS itself. We vary d in $\{1, 5, 10, 50, 100, 500, 1000\}$ to see its impact w.r.t. runtime speedup and the approximation factor P. In the extreme case, $d = 1$, each annulus only contains users with the same degree.Increasing d reduces more dimensions, thus the speedup (left y axis) increases, while the approximation becomes less accurate and the approximation factor decreases. HembExp and HembUni have similar benefit, and it is easier to tune d in HembUni as expected. For the speedup and precision tradeoff, we suggest to set d around 10.

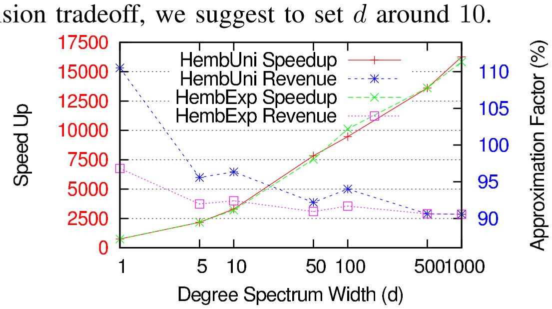


Fig. 4. Effect of tuning Degree Spectrum width d

V. Conclusions & Discussion

In this paper, we develop a novel formulation and dimension reduction method of the SNS ad allocation problem. We introduce a new hyperbolic embedding algorithm, HyperCubeMap, which fulfills the requirements of SNS Ad allocation. We also propose an optimization framework to handle the challenges such as uncorrelated impression distribution and region overlapping issues in the embedding. With our framework, the original integer program can be approximated by a series of linear programs, which successfully reduces the dimensionality and complexity of the optimization and enables application in real world SNS with billions of users.

Discussion: *Social Influence Function*: To embed other influence model is an open problem. For example, recent work [23] shows popular photo cascading may not be shallow. In general, HyperCubeMap will work well with minor modifications, if the approximate influence function is continuous in degree. *Allocation Strategy*: In fan-shaped allocation, as the influence surface is uniform, HyperCubeMap assigns similar influence demographic users to competing advertisers. Different shape design can represent different influence constraints. In [18], we explore fairness meaning and complexity of different shapes in a simplified setting where ads have no target group preference.

References

[1] E. Bakshy, D. Eckles, R. Yan, and I. Rosenn, "Social influence in social advertising: evidence from field experiments," in *ACM EC*, 2012.
[2] Facebook, "One million thank yous," http://newsroom.fb.com/News/639/One-Million-Thank-Yous, [Online; accessed 22-Feb-2013].
[3] R. Johnson, "Scaling Facebook to 500 Million Users and Beyond," http://goo.gl/LKexF5, [Online; accessed 16-Dec-2013].
[4] Facebook, "Campaign cost & budgeting," https://www.facebook.com/help/www/318171828273417, [Online; accessed 16-Feb-2014].
[5] A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani, "Adwords and generalized online matching," *JACM*, vol. 54, no. 5, p. 22, 2007.
[6] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *ACM KDD*, 2003.
[7] E. Bakshy, I. Rosenn, C. Marlow, and L. Adamic, "The role of social networks in information diffusion," in *WWW*, 2012.
[8] M. S. Bernstein, E. Bakshy, M. Burke, and B. Karrer, "Quantifying the invisible audience in social networks," in *ACM CHI*, 2013.
[9] Facebook, "News feed fyi: A window into news feed," http://goo.gl/oTAygF, [Online; accessed 16-Oct-2014].
[10] J. Leskovec, L. A. Adamic, and B. A. Huberman, "The dynamics of viral marketing," *ACM Transactions on the Web*, vol. 1, no. 1, 2007.
[11] S. Goel, D. J. Watts, and D. G. Goldstein, "The structure of online diffusion networks," in *ACM EC*, 2012.
[12] Facebook, "Advertise on facebook," https://www.facebook.com/ads/create/, [Online; accessed 02-Mar-2014].
[13] M. Gromov, *Hyperbolic groups*. Springer, 1987.
[14] D. Krioukov, F. Papadopoulos, M. Kitsak, A. Vahdat, and M. Boguñá, "Hyperbolic geometry of complex networks," *Physical Review E*, vol. 82, no. 3, p. 036106, 2010.
[15] F. Papadopoulos, C. Psomas, and D. Krioukov, "Network mapping by replaying hyperbolic growth," *IEEE/ACM Transactions on Networking*, vol. PP, no. 99, pp. 1–1, 2014.
[16] M. Boguñá, F. Papadopoulos, and D. Krioukov, "Sustaining the internet with hyperbolic mapping," *Nature Communications*, vol. 1, p. 62, 2010.
[17] R. Kleinberg, "Geographic routing using hyperbolic space," in *IEEE INFOCOM*, 2007.
[18] P. Gao, H. Miao, and J. S. Baras, "Social network ad allocation via hyperbolic embedding," *IEEE CDC*, 2014.
[19] R. Hof, "You Know What's Cool? 1 Million Advertisers On Facebook," http://goo.gl/LM0Oqv, [Online; accessed 20-Oct-2014].
[20] Yahoo!, "Webscope A1: Yahoo! Search Marketing Advertiser Bidding Data," http://webscope.sandbox.yahoo.com/.
[21] S. Yuan and J. Wang, "Sequential selection of correlated ads by pomdps," in *ACM CIKM 2012*.
[22] J. Leskovec, "Stanford network analysis package."
[23] P. A. Dow, L. A. Adamic, and A. Friggeri, "The anatomy of large facebook cascades." in *ICWSM*, 2013.