AdHeat: An Influence-based Diffusion Model for Propagating Hints to Match Ads

Hongji Bao
Google Research
Beijing 100084, China
hongjibao@google.com

Edward Y. Chang
Google Research
Beijing 100084, China
edchang@google.com

ABSTRACT

In this paper, we present AdHeat, a social ad model considering *user influence* in addition to *relevance* for matching ads. Traditionally, ad placement employs the *relevance* model. Such a model matches ads with Web page content, user interests, or both. We have observed, however, on social networks that the relevance model suffers from two shortcomings. First, influential users (users who contribute opinions) seldom click ads that are highly relevant to their expertise. Second, because influential users' contents and activities are attractive to other users, *hint words* summarizing their expertise and activities may be widely preferred. Therefore, we propose AdHeat, which diffuses hint words of influential users to others and then matches ads for each user with aggregated hints. We performed experiments on a large online Q&A community with half a million users. The experimental results show that AdHeat outperforms the relevance model on CTR (click through rate) by significant margins.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data Mining*; I.2.6 [**Artificial Intelligence**]: Learning—*Knowledge Acquisition*; J.4 [**Social and Behavioral Sciences**]: Economics

General Terms

Algorithms, Economics, Experimentation, Theory

Keywords

Online Advertising, Contextual Advertising, Behavior Targeting, Social Network Analysis, Influence Propagation, Heat Diffusion

1. INTRODUCTION

Online social communities flourish worldwide nowadays. According to the report of alexa[1], Facebook[2], YouTube[3] and Blogger[4] have stepped into the world's top-10 sites in terms of number of visits a day. An increasing number of people meet in online communities to find like-minded people, to debate topical issues, to play games, to give or ask for information, to find support, or to shop. These social community or social networking sites, however, are yet to monetize effectively. At present, most sites generate revenue via *content-based* ad placement, which analyzes the content of a Web page and embeds relevant ads into it. The other popular approach is *user-targeting* ad placement, which matches ads by users' interests. Both *content-based* and *user-targeting* approaches suffer from data sparsity challenges (details are presented in Section 2).

In this work, we have been developing and experimenting with an *influence-based propagation* model, which we call AdHeat. AdHeat identifies influential users on social networks by analyzing user contributions including, for example, content generation (on forum-like applications), problem solving (at Q&A sites), and other services. AdHeat represents a user's interests by some hint words, which are calculated based on her profile and recently generated contents. A hint word represents a user's interests, for example, a musician's hint may be *rock music*. A community site can perform influence analysis periodically to include the most recent user activities. After the hint words have been generated for influential users, AdHeat propagates these words throughout the networks like heat diffuses through a forest of trees. AdHeat ensures that hints are properly and rapidly spread, and then for each user, it aggregates all incoming hints for matching ads.

During our experiments with AdHeat, we made two interesting observations, which explain the reasons why AdHeat works effectively. First, we observed that an influential user seldom clicks on an ad that is highly relevant to his expertise. This behavior is understandable because an expert may have known the company or product behind the ad or her self-confidence on the subject matter renders relevant ads useless. In fact, the CTR (click through rate) of an influential user on a random ad is higher than on a relevant ad. This further demonstrates the factor of familiarity diminishing curiosity. The second observation is that information about non-influential users is often too sparse to perform effective content or user-based ad targeting. The propagation scheme of AdHeat remedies this shortcoming. Influential users have a strong ability to attract other users' attention and affect their thoughts and actions: their notes are often followed by many users, their questions attract many users to join the discussion, their answers are widely accepted, etc. Users like influential users' posts. Thus, AdHeat propagates influen-

[1] http://www.alexa.com/topsites/global
[2] www.facebook.com
[3] www.youtube.com
[4] www.blogger.com

WWW 2010, April 26–30, 2010, Raleigh, North Carolina, USA.
ACM 978-1-60558-799-8/10/04.

tial users' hint words to the others via the edges of the social graph to achieve two effects. First, hint-word propagation remedies the information sparsity problem. Second, using hint words "suggested" by influential users for matching ads produces ads that are more receptive by less influential users who have been following opinions of influential users.

We performed three experiments on Google Confucius, an online Q&A service available in China, Russia, Thailand, and 17 Arab-speaking countries. The results show that AdHeat to be more effective over content-relevance and user-targeting ad models, outperforming them in CTR by significant margins.

In summary, the contributions of this paper are as follows:

- We propose an influence-based advertising model, which diffuses "hints" from influential users to the others for matching ads. We show that this influence-based model is both necessary—addressing data sparsity problem, and desirable—improving ad relevance.
- We propose AdHeat algorithm, which generates users' "hints" and identifies influential users by performing social graph analysis, using PLDA and HITS algorithms.
- We propose using a diffusion model to propagate "hints" to improve information density for each user to match ads.
- Our experimental results show AdHeat to be effective, and the results also reveal useful insights into understanding influential users' reactions to ads.

The reminder of the paper is organized as follows. Section 2 presents related work. Section 3 depicts the three main steps of AdHeat. Section 4 presents experimental results. We offer concluding remarks in Section 5.

2. RELATED WORK

We present related work in social ad placement in three parts: *content relevance analysis*, *user relevance analysis*, and *influence analysis and propagation*.

2.1 Content Relevance Analysis

Relevance analysis finds the most relevant ads for a Web page or a user. The idea is that the higher the relevance, the more likely an ad is to be clicked [24, 8]. Content relevance is the pioneer online advertisement model for Web pages. This model takes three steps to match ads with a Web page. First, it analyzes the contents of the target page and generates keywords to represent that page. It then uses the keywords to query relevant ads. Finally, the most relevant ads are embedded into the target Web page and displayed to users. Google AdSense is a representative product that is based on content relevance.

Several methods have been proposed based on relevance for matching ads with Web pages. The work of B. Ribeiro-Neto et al. [21] represents both ads and Web pages by vectors, and matches them according to several vector similarity measurements. A. Lacerda et al. [17] propose matching ads with a function generated by learning the impact of individual features using genetic programming. The experimental results show that the matching function outperforms the best method in [21] in finding relevant ads. Another approach treats contextual advertising as an aspect of sponsored search, which extracts phrases from search queries and matches them to bid phrases of the ads [28]. Such a method leads to many irrelevant ads because of the vagaries of phrase extraction and the lack of contextual information. Andrei Broder and Marcus Fontoura [3] propose a method for contextual ad matching based on a combination of semantic and syntactic features, and the method obtains better performance than considering these features individually. The work of Deepayan Chakrabarti et al. [6] improves relevance significantly by incorporating click feedback.

2.2 User Relevance Analysis

A logical improvement for social ad targeting is to personalize ad placement by considering users' idiosyncracies in addition to performing content relevance analysis on Web pages. One source of user information is a user's profile with their age, gender, education, income, and interests. Facebook allows advertisers to choose their targeted audience by demographics. Facebook also monetizes friend graphs by recommending ads clicked by friends.

Unfortunately, user profiles may not always be accurate, complete, or up-to-date. User activities can be a better source for understanding users in an implicit way. For instance, a user who likes to listen to 50s music may be older. User generated content such as blogger/twitter posts and uploaded photos/videos are other sources for understanding users. Relevance analysis on user activities or behavior can obtain users' needs in near-real-time. Some experiments provide evidence that user behavioral targeting helps improve the effectiveness of online advertisement [26]. The work of Foster Provost et al. [20] provides a method to find good audiences for brand advertising by extracting quasi-social networks from browser behavior on user-generated content sites. Collaborative filtering is a widely used technique for user ad targeting. A survey of E. Y. Chang [7] presents representative collaborative filtering algorithms including frequent itemset mining (FIM), SVD, Latent Dirichlet Allocation (LDA). Chen et al. [9] compare the effectiveness of FIM and LDA on user-centric information recommendations.

A well-known shortcoming of user ad targeting is data sparsity. Aside from the fact that most profiles may not be complete or accurate (especially in emerging countries, most users do not provide true identities or real profiles), users seldom generate content. Our proposed AdHeat considers *influence* in addition to *relevance* to remedy this data sparsity problem. AdHeat finds hints from influential users who generate ample activities and then propagates these hints through network edges (representing user relationship, either explicit or implicit) to all users through a heat propagation model.

2.3 Influence Analysis and Propagation

The basic influence model considers only users' declared friendships. Such a model makes ad-placement decision partially based on friends' characteristics. This model may work well with sites such as Facebook and LinkIn, where relationship between people is explicitly expressed. For social sites where user relationship is not explicitly established, such as forum and Q&A sites, the influence model must find implicit relationship through mining user interactions. We believe such implicit-relationship mining to be more useful than explicit expression for two reasons. First, not all users articulate all their relationships or keep such information up-to-date. Second, an explicit link cannot tell how often two people interact or the nature of their interactions. Influence should also consider transitivity, i.e., if user a is influential to user b, and user b to c, then user a can influence user c indirectly. AdHeat considers implicit relationships in addition to explicit ones, and transitive relationships in addition to mutual ones.

On propagating interests or hints on social networks, different mechanisms have been studied mainly in the light of information diffusion and virtual marketing. Various real networks have been analyzed to find the characteristics of information propagation. For example, Daniel Gruhl et al. [11] study the dynamics of information propagation in environments of low-overhead personal publishing, and proposes an algorithm to induce the underlying propagation network from a sequence of posts. The work of Eytan Adar et al. [1] takes advantage of historical, repeating patterns of infection to track the information flow on blogs. The work of Jurij Leskovec et al. [19] studies the propagation of recommendations in person-to-person recommendation network. Both Kristina Lerman et al. [18] and M. Cha et al. [5] analyze a Flickr data set to research photo propagation through the social networks. Gueorgi Kossinets and Jon Kleinberg [16] introduce the structure of information pathways in social communication network, and define the network backbone as the subgraph consisting of the edges on which information has to the potential to flow quickest. To the best of our knowledge, AdHeat is the first ad model that uses influence to diffuse hints for matching ads.

3. ALGORITHMS

The AdHeat consists of three major steps:

1. *Hint word generation.*
2. *Influential user ranking.*
3. *Influence Propagation.*

3.1 Hint Word Generation

Hint word generation characterizes each user with a list of words. The input to *hint word generation* is a user's activities in communities. Given the input, AdHeat employs Latent Dirichlet Allocation (LDA) [2] to perform this hint-word generation task.

LDA was first proposed by Blei, Ng and Jordan to model a document as a bag of words. In communities, each user is involved in some activities. Since most activities are either posting or viewing contents, AdHeat characterizes a user by aggregating the contents he generated or viewed. After converting the contents to words, a user is also represented by a bag of words. If you suppose each user is a mixture of K latent characteristics, taking words-denoted users as input, LDA can infer each user's probability distribution on K characteristics, where each characteristic is a multinomial distribution ϕ_k over a V-word vocabulary. To process a large volume of data, we use a parallel implementation of LDA (PLDA) to generate hint words. We have previously developed PLDA and made it open source. Algorithms 1 shows how PLDA is used to generate users' hint words. For the details of PLDA, please consult reference [25].

An important consideration in generating hint words is freshness of information. For activities that took place a long time ago, we may want to discount their importance. For activities that just took place, e.g., discussion of a new mobile device, they should be weighted higher. To consider freshness, we represent the user by most recent t days of words she generated or viewed, and use them as the inputs of PLDA. The output Θ includes each user's probability distribution on K characteristics, and Φ contains each characteristic's probability distribution on V words. Both input and output are depicted on the top of Algorithms 1

```
Algorithm 1: Hint-Word-Generation.
Input:
$U$: User set, with each user denoted by words.
Output:
$\Theta = \{\vec{\theta}_m\}_{m=1}^{M}$: a $M \times K$ matrix.
$\Phi = \{\vec{\phi}_k\}_{k=1}^{K}$: a $K \times V$ matrix.
Parameters and variables:
$M$: number of users.
$K$: number of user characteristics.
$V$: vocabulary size.
$\vec{\alpha}, \vec{\beta}$: Dirichlet parameters.
$\vec{\theta}_m$: characteristic distribution for user $m$.
$\vec{\phi}_k$: word distribution for characteristic $k$.
$N_m$: the length of user $m$'s activities represented her
generated words, here modelled with a Poisson
distribution with constant parameter $\xi$
$z_{m,n}$: mixture indicator that chooses the characteristic
for the $n^{th}$ word of user $m$.
$w_{m,n}$: term indicator for the $n^{th}$ word of user $m$.
begin
    forall $k \in [1, K]$ do
        Sample mixture components $\vec{\phi}_k \sim Dir(\vec{\beta})$;
    end
    forall user $m \in [1, M]$ do
        Sample mixture proportion $\vec{\phi}_k \sim Dir(\vec{\alpha})$;
        Sample user activity length $N_m \sim Poiss(\xi)$;
        forall word $n \in [1, N_m]$ of user $m$ do
            Sample characteristic index
            $z_{m,n} \sim Mult(\vec{\theta}_m)$;
            Sample term for word $w_{m,n} \sim Mult(\vec{\phi}_{z_{m,n}})$;
        end
    end
end
```

Based on the output of PLDA, we generate hint words for each user by the following method: For the m^{th} user, we choose the top i characteristics by weights in $\vec{\theta}_m$, and for each selected characteristic k, we choose top j words by weights in $\vec{\phi}_k$ and add them to the m^{th} user's hint word list. A hint word's weight is its weight in $\vec{\phi}_k$ multiplied by its characteristic's weight in $\vec{\theta}_m$. Finally, each user have $i * j$ weighted hint words. They are used to match relevant advertisements to target users.

The method above generates hint words for individual users and does not consider interactions between users. AdHeat separates individual analysis and propagation into two orthogonal steps, and the computation of influential scores, discussed next, determines the scopes and directions of hint-word propagations.

3.2 Influential User Ranking

Our goal in this step is to rank users based on their influence on social networks. A person becomes influential due to two factors: level of activity and authority. Active users interact with many users frequently. Active users are like broadcasters who can propagate news (and gossip) quickly to many users. Authoritative users may or may not always engage with other users. They, however, are the authors of high-quality content that draws attention. Authoritative users are highly respected and their opinions followed by many users. From the perspective of information dissemination, both active and authoritative users are influential. Therefore, if one would like to propagate ads, one should place ads by closely consulting with influential users.

To compute influence user rank, we employ HITS (Hypertext Induced Topic Selection) [15]. Such link-based methods have been proven to be successful in social network analysis tasks [22]. Gyongyi et al. applied the HITS algorithm to the question and answer graph of Yahoo! Answers and observed a positive correlation between user quality and hub/authority scores [12]. Various other researchers have also reported the effectiveness of HITS in expert identification on Q&A sites [13, 29] and email exchange networks [4, 10]. Our methods differ in the constructing social network graph. We assign edge weights between users to enhance HITS algorithms in ranking the users by their influence. In the remainder of this section, we depict how we assign weights on edges.

3.2.1 Constructing Social Graph

The social network is a group of users connected by relationships. Let $G(U, E)$ denote the social network graph, where $U=\{u_1, u_2, ..., u_n\}$, $E=\{(u_i, u_j)| \exists$ directed edge from u_i to $u_j\}$, U is the user set, E is the edge set, and the weight of (u_i, u_j) quantifies u_i's dependence to u_j. Generally, the weight of (u_i, u_j) isn't equal to that of (u_j, u_i). That's because the relationship between two users are asymmetric. For example, u_j is a pop star and u_i is his fans. While u_i keeps following u_j's activities on the social communities, u_j may never be concerned u_i at all.

The simplest way to build edges is according to friendship ties, that is, if two users are friends, there is an edge between them, and its weight is one. Friendship-generated edges have no direction, and they are all equal because all such edge have a weight of one. Therefore, this method alone is clearly ineffective to quantify influence. Instead, we assess the relationship between two users by the frequency and quality of their interactions. Frequency is simple to account for. To assess quality, we consider several factors in an application-dependent way. For a Q&A application, an answerer is considered to be influential if the answers provided are timely, highly relevant to questions, and useful. For a forum/BBS application, a user is influential if her posts enjoy high page views. Since our experiments are conducted on a Q&A site, we consider the quality factors for Q&A application. Our previous work, which is proposed by Si et al. in [23], concludes that several factors listed in table 1 may affect interaction quality. Among the the factors, Si et al. identify that *#cov* (coverage), *#npbaul* (voted as a good post), *#rword* (relevance), *#origt* (originality), and *#focus* (focus) weight the highest through empirical studies. Thus, we use these factors to quantify user interaction quality. After getting the factor weight of each interaction, we combine the weights of user interactions which have the same source and target. The edge weight from u_i to u_j is determined by two factors: the number of interactions from u_i to u_j and each interaction's weight. We normalize all edge-weights to [0,1] by *MAX-MIN* method. The generated social network graph G is the input to HITS algorithm in section 3.2.2 and the input to influence propagation algorithm in section 3.3.

Factor	Description
#nw	Number of words.
#nuw	Number of unique words.
#puw	Percentage of unique words.
#prompt	Promptness, time from when the question was asked to when the answer was provided.
#cov	Coverage, computed as the sum of the word's IDFs.
#nau	Total number of answers provided by the answerer before.
#npbau	Percentage of best answers in all closed threads participated in by the answerer.
#naul	Number of answers posted by the answerer to questions on the same topic.
#npbaul	Percentage of best answers posted by the answerer in closed threads within the same topic.
#rlda	Relevance, computed as the cosine similarity of LDA latent topic distribution between the question and the answer.
#rword	Relevance, computed as the cosine similarity of TF*IDF weighted word vectors between the question and the answer.
#origt	Originality, compared to earlier answers for the same question.
#origu	Originality, compared to earlier answers from the same user.
#focus	Focus, how focused the answerer's domain of knowledge or interests is, based on her past answers.

Table 1: Features extracted from each answer in a community Q&A system.

3.2.2 HITS

Taking the social network graph G as input, the HITS algorithm computes two scores for each user:

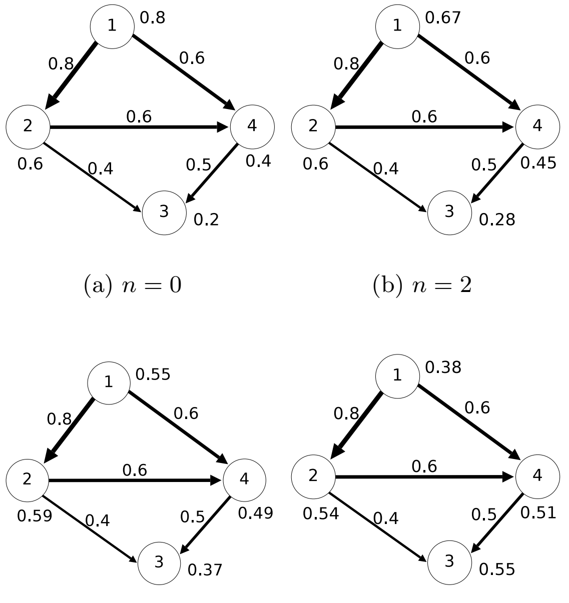
- A **hub** score, which indicates user's contribution to propagate information.
- An **authority** score, which indicate user's contribution to provide attractive contents.

Let the adjacency matrix W denote the social network graph G, where w_{ij} denotes the edge weight from user i to j. We initialize the hub scores $\vec{h}$ and the authority scores $\vec{a}$ by random values at the start. Then we update the two score vectors iteratively by the following equations:

$$\mathbf{h}^{(n+1)} = \lambda \mathbf{1} + (1 - \lambda) W_{col} \mathbf{a}^n \tag{1}$$

$$\mathbf{a}^{(n+1)} = \lambda \mathbf{1} + (1 - \lambda) W_{row}^T \mathbf{h}^n, \tag{2}$$

where $\mathbf{1}$ is the vector of all ones, W_{row} is the same as W with its rows normalized to sum to one, W_{col} is W with its columns normalized to sum to one, and λ is a reset probability to guarantee the convergence of the algorithm. HITS



(a) $n = 0$ (b) $n = 2$

(c) $n = 4$ (d) $n = 8$

Figure 1: Heat Distribution Changes.

terminates when satisfying the following conditions:

$$||\mathbf{h}^{(n+1)} - \mathbf{h}^n|| \le \epsilon \quad (3)$$

$$||\mathbf{a}^{(n+1)} - \mathbf{a}^n|| \le \epsilon, \quad (4)$$

where ϵ is a predefined convergence threshold. We compute influence scores by

$$\mathbf{I} = \alpha\mathbf{h} + (1 - \alpha)\mathbf{a}, \quad (5)$$

where $\alpha \in [0, 1]$. $\mathbf{I}$ is one of the inputs for influence propagation algorithm in section 3.3.

3.3 Influence Propagation

Given a list of users ranked by their influence scores, AdHeat propagates their hint words throughout the networks. We propose using heat diffusion to perform propagation. Heat diffusion describes the procedure by which heat diffuses from one or more areas to others in a manifold. When we treat a social graph as a manifold with the users' influence as the heat sources, influence propagation fits the heat diffusion model. We use an example to illustrate how such diffusion works for propagating hint words, followed by a formal specification of the algorithm.

3.3.1 Illustrative Example

Figure 1(a) and the first row of Table 2 show the initial state of an example social graph. The diffusion manifold in the figure consists of four users denoted as #1, #2, #3, and #4. The influence scores of the users are printed next to the user nodes, and they are 0.8, 0.6, 0.2, and 0.4 for users #1, #2, #3, and #4, respectively. The edges of the graph are the diffusion rates, initialized by the edge weights of the social network graph. Table 2 shows that PLDA assigns each user two hint words with relevance scores. For instance, words 'a' and 'b' are most relevant to user #1 with a normalized weight of 0.6 and 0.4, respectively.

Table 2: Propagating Hint Words.

n	u#	Hint Words
0	#1	(a, 0.6) (b, 0.4)
	#2	(c, 0.8) (b, 0.2)
	#3	(e, 0.5) (f, 0.5)
	#4	(d, 0.9) (b, 0.1)
1	#1	(a, 0.6) (b, 0.4)
	#2	(c, 0.69) (b, 0.23) (a, 0.08)
	#3	(e, 0.4) (f, 0.4) (c, 0.1) (d, 0.07) (b, 0.03)
	#4	(d, 0.66) (b, 0.16) (a, 0.11) (c, 0.07)
2	#1	(a, 0.6) (b, 0.4)
	#2	(c, 0.63) (b 0.24) (a, 0.13)
	#3	(e, 0.32) (f, 0.32) (c, 0.17) (d, 0.11) (b, 0.07) (a, 0.02)
	#4	(d, 0.51) (b, 0.2) (a, 0.17) (c, 0.11)
4	#1	(a, 0.6) (b, 0.4)
	#2	(c, 0.59) (b, 0.25) (a, 0.16)
	#3	(c, 0.24) (e, 0.22) (f, 0.22) (d, 0.13) (b, 0.12) (a, 0.06)
	#4	(d, 0.36) (b, 0.24) (a, 0.24) (c, 0.16)
8	#1	(a, 0.6) (b, 0.4)
	#2	(c, 0.59) (b, 0.25) (a, 0.16)
	#3	(c, 0.3) (e, 0.16) (f, 0.16) (b, 0.15) (a, 0.13) (d, 0.1)
	#4	(d, 0.29) (b, 0.25) (a, 0.24) (c, 0.22)

1. **End of iteration** #2. Figure 1(b) shows the influence scores after two iterations of diffusion. The influence score of user #1 has dropped from 0.8 to 0.67, whereas that of user #3 and #4 have increased. At the same time, the hint words have been propagated. The third row of Table 2 shows that hint word 'a' has reached third on the word list of user #4. This is because compared to user #4, user #1 is twice more influential, and his top word has infected the list of user #4.

2. **End of iteration** #8. Figure 1(d) shows influence scores after the diffusion algorithm converges, at which the heat of the manifold is in a balanced state. The final row of Table 2 shows hint words on all nodes. The top hint word of user #3 have been replaced by the words of users #2. User #4 adopts user #1's top words because user #1 is four times more influential. At the same time, user #4 keeps word 'd' on her list because the initial relevance of that word is very high on her list.

The example shows that hint-word propagation properly balances local relevance and global influence. At the end of the propagation, AdHeat uses the list of hint words of each user to query against Google AdSense for the most relevant ads for the user.

3.3.2 Heat Diffusion Model

Heat diffusion models have been studied extensively in epidemiology to predict the spread of infections diseases, in medicine to track the progression of cancer cells, and in mobile ad hoc networks to choose information dissemination strategies [14]. The heat equation (6) is the basis for most diffusion models, including DiffusionRank [27], where ∇^2 is the Laplace operator over the spatial dimensions and Γ is the diffusion coefficient.

$$\frac{\partial \phi}{\phi t} = \Gamma \nabla^2 \phi(\mathbf{x}, t) \quad (6)$$

In adapting the diffusion model to analyze the influence network, we start with a more general form of the heat equation, shown below

$$\frac{\partial \phi}{\phi t} = \nabla \cdot (\Gamma(\phi, \mathbf{x}) \nabla \phi(\mathbf{x}, t)), \tag{7}$$

where $\phi(\mathbf{x}, t)$ is the heat distribution at node $\mathbf{x}$ at time t, and $\Gamma(\phi, \mathbf{x})$ is the diffusion coefficient for heat distribution ϕ at node $\mathbf{x}$; the ∇ symbol represents the vector differential operator acting on the space coordinates. The general diffusion equation above is non-linear and difficult to solve analytically. Given our problem is in a discrete space, we can perform the diffusion of heat on a node-by-node basis, where we calculate the amount of diffused and received heat for each iteration of our algorithm.

On a directed graph, received heat (RH) for node i should be proportional to (1) the time period Δt, (2) the diffusion rate γ_{ij} of a neighboring node i, and (3) the heat at node v_j. Similar guidelines can be made for diffusing heat. The extended model sets different γ_{ij} values for different edges, which reflects the relative importance of connections between various nodes v_j linked to a given node v_i. The use of varying γ_{ij} values allows us to treat outgoing edges differently. Following the above guidelines, we propose these equations for the heat received and diffused for node v_i:

$$RH(v_i) = \sum_{j:(v_j, v_i) \in E} \gamma_{ji} \mathbf{f}_j(t) \Delta t / d_j \tag{8}$$

$$\begin{aligned} DH(v_i) &= \sum_{j:(v_i, v_j) \in E} \gamma_{ij} \mathbf{f}_i(t) \Delta t / d_i \\ &= \frac{\sum_{j:(v_i, v_j) \in E} \gamma_{ij}}{d_i} \mathbf{f}_i(t) \Delta t \\ &= \bar{\gamma}_i \mathbf{f}_i(t) \Delta t \end{aligned} \tag{9}$$

where $RH(v_i)$ and $DH(v_i)$ are heat received and diffused for node v_i respectively; $\mathbf{f}_i(t)$ is the heat of the node i at time t; γ_{ij} is the diffusion coefficient between nodes i and j. The larger the value of γ_{ij}, the faster the heat diffuses from one node to another. The diffused heat $DH(v_i)$ can be simplified by using $\bar{r_i}$ as the average γ of the outgoing edges for node v_i. Combining the diffused and received heat equations, the net change in the heat of a node i is given by the following equation:

$$\mathbf{f}_i(t + \Delta t) - \mathbf{f}_i(t) = (\sum_{j:(v_j, v_i) \in E} \gamma_{ji} \mathbf{f}_j(t)/d_j - \bar{r_i} \mathbf{f}_i(t)) \Delta t \tag{10}$$

We can represent the full heat solution $\mathbf{f}(t)$ by using diffusion rate matrix Γ and transition matrix A. Taken to the limit, when $\Delta \leftarrow 0$:

$$\frac{d\mathbf{f}(t)}{dt} = \Gamma \circ A \mathbf{f}(t) \tag{11}$$

$$\Gamma_{ij} = \begin{cases} \bar{\gamma}_i & \text{if } i = j \\ \gamma_{ji} & \text{if } (v_j, v_i) \in E \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

$$A_{ij} = \begin{cases} -1 & \text{if } i = j \\ 1/d_j & \text{if } (v_j, v_i) \in E \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

where the $\circ$ operator represents the Hadamard product of two matrices. Solving equation (11), we obtain the following:

$$\mathbf{f}(t) = e^{\Gamma \circ A t} \mathbf{f}(0), \tag{14}$$

especially we have

$$\mathbf{f}(1) = e^{\Gamma \circ A} \mathbf{f}(0), \tag{15}$$

where $\mathbf{f}(1)$ denote the final heat distribution on nodes. Computing $e^{\Gamma \circ A}$ is time-consuming, so we use its discrete approximation:

$$\mathbf{f}(1) = (I + \frac{\Gamma \circ A}{N})^N \mathbf{f}(\mathbf{0}). \tag{16}$$

3.3.3 Influence Propagation by Heat Diffusion

```
Algorithm 2: Influence Propagation
  Input:
  $W$: the adjacency matrix that denotes the social
  network graph $G(U, E)$.
  $L^0$: all users' weighted hint word lists, initially.
  $\mathbf{I}$: all users' influence scores.
  Output:
  $L^N$: users' weighted hint word lists after influence
  propagation.
  Parameters and variables:
  $M$: number of users.
  $\Gamma$: heat diffusion rate matrix.
  $A$: transition matrix.
  $\Delta L^n$: the propagated hint word lists after the $n^{th}$
  iteration.
  begin
      Compute $\Gamma$ on graph $G(U, E)$ by equation (12);
      Compute $A$ on graph $G(U, E)$ by equation (13);
      $S \leftarrow \Gamma \circ A$;
      Set iteration number $N$;
      $\mathbf{h}^0 \leftarrow \mathbf{I}$;
      for $n \leftarrow 1$ to $N$ do
          forall user $i \in [1, M]$ do
              forall user $j$ that $W_{ij} > 0$ and
              $\mathbf{h}_i^{(n-1)} > \mathbf{h}_j^{(n-1)}$ do
                  forall hint-weight pair $(h, w)$ in $L_i^{(n-1)}$
                  do
                      $w' \leftarrow (\mathbf{h}_i^{(n-1)} - \mathbf{h}_j^{(n-1)}) \cdot W_{ij} \cdot w$;
                      Append hint-weight pair $(h, w')$ to
                      $\Delta L_j^n$;
                  end
              end
          end
          forall user $i \in [1, M]$ do
              $L_i^n \leftarrow$ Combine the wights of the same hint
              word in $\Delta L_i^n$ and $L_i^{(n-1)}$;
              forall hint word $h$ in $L_i^n$ do
                  Normalize $h$'s weight to [0, 1] by dividing
                  the sum of all weights in $L_i^n$;
              end
          end
          $\mathbf{h}^n \leftarrow (I + \frac{S}{N})\mathbf{h}^{(n-1)}$;
      end
      Return $L^N$;
  end
```

This section presents the *influence propagation* algorithm formally. The input to the algorithm includes:

- The social graph $G(U, E)$ denoted by the adjacency matrix,
- Weighted hint-word lists of individuals, and
- Users' influence scores.

The first step is to build the heat diffusion manifold. Generally, it can be a social graph or a reversed social graph. For a community where repliers are more influential (e.g., a Q&A community), we take the social graph as the diffusion manifold. For a community where source of opinions weight more (e.g., a BBS or forum), we take the reversed social graph as the diffusion manifold. Since we conduct experiments on Q&A community, our algorithm directly uses the social graph as the diffusion manifold in this paper. The initial heat of users are their influence scores. The algorithm uses the discrete heat diffusion model to conduct influence propagation. Algorithm 2 presents the pseudo code of influence propagation.

A couple of steps deserve further discussion. First, the number of diffusion iterations, N, is affected by the structure of the diffusion manifold and all diffusion rates. Finding N is equal to solving such a problem: for a given threshold ϵ, find N such that $||((I + \frac{\Gamma \circ A}{N})^N - e^{\Gamma \circ A})\mathbf{f}(0)|| < \epsilon$ for any $\mathbf{f}(0)$ whose sum is one. N can be empirically determined given data, or one can use this inequality as the termination criteria.

The other important step is re-weighting words after each diffusion iteration. Here, many heuristics can be employed. The best method would to consult CTR of propagated words as feedback to determine the strength of a propagated word. For example, if an ad produced for hint words 'a' and 'b' has enjoyed a high CTR, AdHeat should subsequently weight both words with greater strengths so that they can be propagated further. On the contrary, if their CTR is low, their strengths may be decreased. Lacking such feedback information, AdHeat initially uses two combined factors to determine the strength of a propagated hint word. The first factor is the word weight at the source of propagation; the second factor is the heat difference between the source and destination; and the third factor is the diffusion rate from the source to destination. When the heat diffusion terminates, we obtain the users' weighted hint word lists. Each user's hint words will now be different from their initial list of words.

4. EXPERIMENTS

We perform experiments on Google Confucius, which is a community-based Q&A product launched in twenty countries. Users' main activities on Confucius are asking questions and providing answers. Each user can optionally create a profile. Therefore, Confucius is like a social community in which users help each other in solving problems.

We sampled a subset of 5,000 users in our experiments. It is important to mention that though ad placement experiments were conducted on a subset of users, we employed AdHeat to analyze the entire social graph of half a million registered users (vast majority of the users are read-only ones and they do not have a registered account) to generate hint words, to compute influence scores, and to propagate hints.

Table 3: Post Information.

Post ID	Unique ID for the post.
Source User	The user who created the post.
Target User	The user who the source user followed.
Content	The content of the post.
Time Stamp	The time when the post is created.

We conducted three experiments. The first experiment compared our influence model without propagation with the content-based ad placement model. The second experiment evaluated the influence model with and without propagation. The goal is to see whether propagation can help and by how much. In the third experiment, we studied the behaviors of influential users to gain more insights in designing effective placement methods.

4.1 Data Set and Evaluation Metrics

We describe our data set in three parts: (1) the data for conducting AdHeat, (2) the user set for displaying ads, and (3) the ad set.

The first part of data are organized by thread. A thread contains a question post and several answers to that question. Each post is organized by the format described in Table 3. Since the interests of a user can be time-variant, we collected a month of user activities and interactions to conduct experiments. For our experiments, we only run AdHeat once on the collected data, and then we tallied CTR to perform evaluation and comparison. We expected CTR to decrease over time because users would unlikely to click on the same ads several times. Nevertheless, this static evaluation framework provided us sufficient information for conduct comparison and analysis.

We chose 5,000 out of a pool of active Confucius users for targeting ads. The selected users met the condition that they signed in Confucius at least 30 times a month. We chose active users for showing ads because we would have some assurance that ads would be seen, and we could receive feedback during the experiment period. During the experiment period, our system computed a set of targeted ads for each user, and randomly selected five ads for the user once a time. This placement method provided some changes in ads between different sign-on sessions of a user.

Our ads come from Google AdSense, which has millions of ads provided by all kinds of advertisers. The ads cover nearly all areas of interests such as sports, music, drug and fashion. We query relevant ads from AdSense by providing hint words.

We use CTR as the evaluation metric. For a group of experimental users, its CTR is computed by

$$CTR_{[d_s,d_e]} = \frac{Clicks_{[d_s,d_e]}}{Impressions_{[d_s,d_e]}}, \tag{17}$$

where d_s and d_e denote the start day and the end day respectively, $Impressions_{[d_s,d_e]}$ denotes the number of times that ads are shown to the group users during the period from d_s to d_e, and $Clicks_{[d_s,d_e]}$ denotes the number of times that users in the group click the showing ads. Especially when $d_s = d_e$, we get the CTR of one day. To smooth one day's CTR, we use seven-day average CTR to report some results (in Table 4 and 5). A seven-day window guarantees to cover Saturday and Sunday, when traffic is light and user behavior

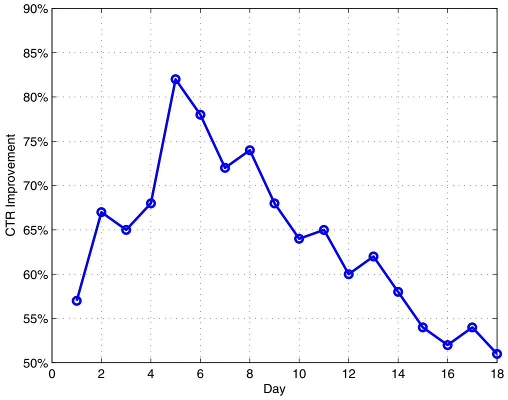


Figure 2: The CTR improvement of influence model over content model in 18 days.

may be different. To simplify, we denote seven-day average CTR as

$$CTR_d = \frac{\sum_{i=0}^{6} CTR_{d+i}}{7}. \tag{18}$$

Due to business confidentiality, we report only relative performance but not the absolute numbers of CTR when showing experimental results.

4.2 Influence Model without Propagation

We randomly divided users who were shown ads into two equal-size groups. Let G_1, G_2 denote the two groups.

- For users in G_1, we used Google AdSense for targeting ads. AdSense analyzes the contents of web pages for matching relevant ads. In this way, the shown ads to a user only relate to the content of the Web page that he is viewing, but have nothing to do with his interests.

- For users in G_2, we took the influence model without propagation for advertising. We implemented such a model by adding hint words to AdSense before it selected ads. In this way, AdSense matches ads for each user based on two parts of words: the first part are the keywords of Web page content that the user is viewing, and the second part are the user's individual hint words. We employed the hint word generation method of the AdHeat to generate 20 hint words for each user. We hadn't performed influence propagation to generate mixed hint words for each user yet in this experiment.

	1	2	3	4	5	6
$\overline{CTR}\uparrow$	0.87	0.89	0.75	0.72	0.90	0.64
±	0.51	0.58	0.46	0.43	0.62	0.29
	7	8	9	10	11	12
$\overline{CTR}\uparrow$	0.58	0.60	0.61	0.52	0.51	0.52
±	0.19	0.23	0.25	0.27	0.26	0.28

Table 4: Mean and standard error of CTR improvement of influence model without propagation over content model.

We performed the first experiment on Confucius for 18 days, and recorded the *Impressions* and *Clicks* of the two models of each day. Let CTR^{inf} and CTR^{con} denote the CTR of influence-base model and that of the traditional content-based model, respectively. Figure 2 reports the accumulated CTR improvement of the influence-based model over the content-based one. The accumulated CTR improvement in the i^{th} day is computed by

$$\frac{CTR^{inf}_{[1,i]} - CTR^{con}_{[1,i]}}{CTR^{con}_{[1,i]}}. \tag{19}$$

During the 18-day period, the peak CTR improvement reaches 82% on the fifth day, then the improvement decreases to 51% on the final day. The average improvement is 66.9%. As predicted, CTR improvement began to decrease after five days. This is partly because we generated only a few ads for each user and a user is unlikely to click on the same ads, and partly because while we kept hint-words static for 18 days, the users' interests might have shifted. The result indicates that the influence-based model of AdHeat to be effective; at the same time, it suggests that hint words should be computed periodically to keep pace with interest drifts.

Table 4 reports seven-day average CTR improvement of the influence-based over content-based model. We used a seven-day window to compute average CTR to mask weekend behavior. The symbol $\overline{CTR}\uparrow$ denotes CTR improvement. We report both the mean improvement and standard error for twelve seven-day windows.

4.3 Influence Model with Propagation

This experiment evaluated the influence model with propagation comparing with that without propagation. The goal was to evaluate whether hint word propagation is useful. We applied AdHeat to the activities of the most recent month on Confucius to compute users' hint words and then propagate those words throughout the network. We again divided users into two groups, G_1 and G_2, for showing ads. In the previous experiment, we targeted ads for G_1 by content model and G_2 by influence model. In this experiment, we targeted G_1 using the influence model with propagation (a more effective model) and G_2 using the influence model without propagation.

	1	2	3	4	5	6
$\overline{CTR}\uparrow$	0.94	0.72	0.73	0.66	0.58	0.79
±	0.48	0.26	0.24	0.23	0.25	0.32
	7	8	9	10	11	12
$\overline{CTR}\uparrow$	0.64	0.73	0.69	0.55	0.29	0.57
±	0.29	0.31	0.35	0.41	0.35	0.52

Table 5: Mean and standard error of CTR improvement of influence model with propagation over the influence model without propagation.

We also ran this experiment for 18 days. Figure 3 shows the accumulated CTR improvement of the propagation effect. The peak improvement reaches an impressive 102% on day three. Propagation helps! The improvement then started to steadily decrease after that to 41% on the final day. The average improvement is 66.4%. Since both schemes are influence-based, and both served a few ads in the span of 18 days, we can eliminate the effect of user boredom from

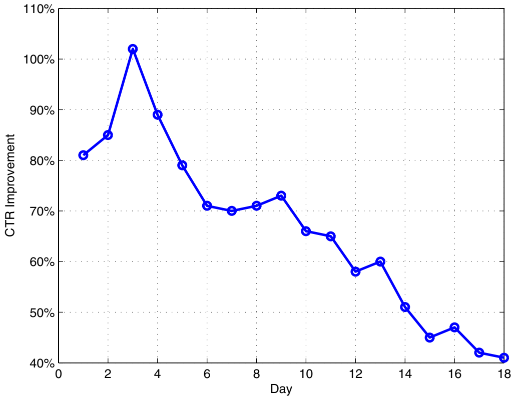


Figure 3: The CTR improvement of influence model with propagation over that without propagation in 18 days.

this experiment. The only telling factor of the decreases in improvement should more likely related to timeliness of the propagated words.

The same to the first experiment, Table 5 reports the mean and standard error of seven-day average CTR improvement of influence model with propagation over without propagation.

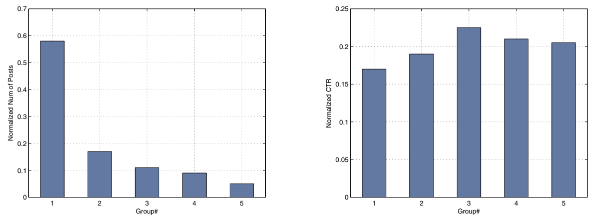
4.4 Correlation between Influence and Performance

We have seen that hint-word-propagation helps improve CTR. We next report our findings on the behaviors of influential users versus non-influential ones.

This experiment evaluated the users' performance on Confucius based on (1) their content contribution and (2) their revenue contribution respectively. We quantified their content contribution by their number of posts during the experimental period, whereas we measured their revenue contribution based on their CTR data.

We ran the experiments as follows: First, we ran AdHeat on recent generated contents (one month) of Confucius to compute all related users' influence scores and individual hint words. Second, we reordered the users to be shown ads by sorting them in decreased influence-score order, and then divided them into five groups of the same size. G_1, G_2, G_3, G_4, G_5 denote the five groups: G_1 includes the most influential users, then G_2, G_3, G_4, and G_5 in decreasing influence. Third, we employed user influence model without propagation to target them ads and tracked their CTRs. We also tracked the number of posts of each group in order to quantify their content contribution.

Figure 4(a) shows the content contribution of each group. The group's performance in term of content contribution was directly proportional to their influence. The more influential the group was, the more posts they created on Confucius. By contrast, Figure 4(b) shows their contribution in term of revenue. We can see that the most influential group has lower CTR than that of the least influential group. The most influential users contributed abundant opinions (shown in figure 4(a)), so the ads shown to them based on their individual hint words should be highly relevant. However, their lowest average CTR indicates that relevance could be counter-productive for them, because they do not need advice on their well-known subjects. For non-influential users, they are also less likely to click on ads generated based on content, since they did not generate enough content for content-based ad matching to be effective.



(a) Content

(b) CTR

Figure 4: Users' Performance Evaluation based on Content and CTR contribution.

The result of this experiment, together with those reported by the previous experiments indicate two productive avenue for matching ads on social networks. First, non-influential users do not contribute enough content so that the traditional content-based ad model may be ineffective due to information sparsity. Second, the propagation of hint words appears to be helpful to remedy the information sparsity problem. Furthermore, correlating ads with information that one follows makes AdHeat effective.

5. CONCLUSIONS

In this paper, we presented an influence-based social ad model. In our proposed AdHeat algorithm, we consider *user influence* in addition to *relevance* for matching ads. AdHeat first identifies the most relevant words or hints for each user based on their individual activities. It then employs HITS to compute for each user an influence score based on user interactions. The hint words are propagated from influential users to the others like heat diffuses throughout a manifold. Finally, AdHeat uses the hint words aggregated for each user to generate for her the most relevant ads. The influence-based model not only alleviates the information sparsity problem of non-influential or inactive users, but also improves ads relevance. We performed experiments on a large online Q&A community with half a million users. The experimental results show that AdHeat outperforms the relevance model on CTR (click through rate) by significant margins.

6. ACKNOWLEDGMENTS

The authors would like to thank Xiance Si, Brian Wong and Alice Li for their previous work, and Jun Yao for his help in conducting experiments on Google Confucius.

7. REFERENCES

[1] Eytan Adar and Lada A. Adamic. Tracking information epidemics in blogspace. In *WI '05: Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 207–214, Washington, DC, USA, 2005. IEEE Computer Society.

[2] David M. Blei, Andrew Y. Ng, Michael I. Jordan, and John Lafferty. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3, 2003.

[3] Andrei Broder, Marcus Fontoura, Vanja Josifovski, and Lance Riedel. A semantic approach to contextual advertising. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 559–566, New York, NY, USA, 2007. ACM.

[4] C.S. Campbell, P.P. Maglio, A. Cozzi, and B. Dom. Expertise identification using email communications. In *Proceedings of the twelfth international conference on Information and knowledge management*, pages 528–531. ACM New York, NY, USA, 2003.

[5] M. Cha, A. Mislove, and K.P. Gummadi. A measurement-driven analysis of information propagation in the flickr social network. In *Proceedings of the 18th international conference on World wide web*, pages 721–730. ACM New York, NY, USA, 2009.

[6] Deepayan Chakrabarti, Deepak Agarwal, and Vanja Josifovski. Contextual advertising by combining relevance with click feedback. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 417–426, New York, NY, USA, 2008. ACM.

[7] E.Y. Chang. Parallel Algorithms for Collaborative Filtering. In *Algorithmic Aspects in Information and Management: 5th International Conference, Aaim 2009, San Francisco, CA, USA, June 15-17, 2009, Proceedings*, page 2. Springer-Verlag New York Inc, 2009.

[8] P. Chatterjee, D.L. Hoffman, and T.P. Novak. Modeling the clickstream: Implications for web-based advertising efforts. *Marketing Science*, 22(4):520–541, 2003.

[9] Wen Y. Chen, Jon C. Chu, Junyi Luan, Hongjie Bai, Yi Wang, and Edward Y. Chang. Collaborative filtering for orkut communities: discovery of user latent behavior. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 681–690, New York, NY, USA, 2009. ACM.

[10] B. Dom, I. Eiron, A. Cozzi, and Y. Zhang. Graph-based ranking algorithms for e-mail expertise analysis. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pages 42–48. ACM New York, NY, USA, 2003.

[11] Daniel Gruhl, R. Guha, David L. Nowell, and Andrew Tomkins. Information diffusion through blogspace. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 491–501, New York, NY, USA, 2004. ACM.

[12] Z. Gyongyi, G. Koutrika, J. Pedersen, and H. Garcia-Molina. Questioning Yahoo! Answers. In *First Workshop on Question Answering on the Web, held at WWW*, 2008.

[13] P. Jurczyk and E. Agichtein. Discovering authorities in question answer communities by using link analysis.

[14] A. Khelil, C. Becker, J. Tian, and K. Rothermel. An epidemic model for information diffusion in manets, 2002.

[15] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[16] Gueorgi Kossinets, Jon Kleinberg, and Duncan Watts. The structure of information pathways in a social communication network, Jun 2008.

[17] A. Lacerda, M. Cristo, M.A. Gonçalves, W. Fan, N. Ziviani, and B. Ribeiro-Neto. Learning to advertise. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, page 556. ACM, 2006.

[18] Kristina Lerman and Laurie Jones. Social browsing on flickr. Dec 2006.

[19] Jurij Leskovec, Lada A. Adamic, and Bernardo A. Huberman. The dynamics of viral marketing, Sep 2005.

[20] Foster Provost, Brian Dalessandro, Rod Hook, Xiaohan Zhang, and Alan Murray. Audience selection for on-line brand advertising: privacy-friendly social network targeting. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 707–716, New York, NY, USA, 2009. ACM.

[21] B. Ribeiro-Neto, M. Cristo, P.B. Golgher, and E. Silva de Moura. Impedance coupling in content-targeted advertising. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, page 503. ACM, 2005.

[22] John P. Scott. *Social Network Analysis: A Handbook*. SAGE Publications, January 2000.

[23] Xiance Si, Zoltán Gyöngyi, Edward Y. Chang, and Maosong Sun. Userrank: Improving user generated content search by user reputation. Technical report, Google Research, 2009. `http://infolab.stanford.edu/~echang/UserRank.pdf`.

[24] C. Wang, P. Zhang, R. Choi, and M.D. Eredita. Understanding consumers attitude toward advertising. In *Eighth Americas Conference on Information Systems*, pages 1143–1148. Citeseer, 2002.

[25] Y. Wang, H. Bai, M. Stanton, W.Y. Chen, and E.Y. Chang. PLDA: Parallel Latent Dirichlet Allocation for Large-scale Applications.

[26] J. Yan, N. Liu, G. Wang, W. Zhang, Y. Jiang, and Z. Chen. How much can behavioral targeting help online advertising? In *Proceedings of the 18th international conference on World wide web*, pages 261–270. ACM New York, NY, USA, 2009.

[27] Haixuan Yang, Irwin King, and Michael R. Lyu. Diffusionrank: a possible penicillin for web spamming. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 431–438, New York, NY, USA, 2007. ACM.

[28] W. Yih, J. Goodman, and V.R. Carvalho. Finding advertising keywords on web pages. In *Proceedings of the 15th international conference on World Wide Web*, pages 213–222. ACM New York, NY, USA, 2006.

[29] J. Zhang, M.S. Ackerman, and L. Adamic. Expertise networks in online communities: structure and algorithms. In *Proceedings of the 16th international conference on World Wide Web*, page 230. ACM, 2007.