Large-Scale Behavioral Targeting with a Social Twist

Kun Liu
Yahoo! Labs
4301 Great America Pkwy
Santa Clara, CA 95054
kun@yahoo-inc.com

Lei Tang
Yahoo! Labs
4301 Great America Pkwy
Santa Clara, CA 95054
ltang@yahoo-inc.com

ABSTRACT

Behavioral targeting (BT) is a widely used technique for online advertising. It leverages information collected on an individual's web-browsing behavior, such as page views, search queries and ad clicks, to select the ads most relevant to user to display. With the proliferation of social networks, it is possible to relate the behavior of individuals and their social connections. Although the similarity among connected individuals are well established (*i.e.*, homophily), it is still not clear whether and how we can leverage the activities of one's friends for behavioral targeting; whether forecasts derived from such social information are more accurate than standard behavioral targeting models. In this paper, we strive to answer these questions by evaluating the predictive power of social data across 60 consumer domains on a large online network of over 180 million users in a period of two and a half months. To our best knowledge, this is the most comprehensive study of social data in the context of behavioral targeting on such an unprecedented scale. Our analysis offers interesting insights into the value of social data for developing the next generation of targeting services.

Categories and Subject Descriptors

H.2.8 [**DATABASE MANAGEMENT**]: Database applications—*Data mining*; J.4 [**SOCIAL AND BEHAVIORAL SCIENCES**]: Economics

General Terms

Algorithms, Experimentation

Keywords

advertising, social targeting, behavioral targeting, social-network analysis, large-scale data mining

1. INTRODUCTION

Behavioral targeting (BT) [4, 5, 25] is an online marketing service that infers the specific interests of consumers based on their online activities. By understanding factors such as the frequency of content consumed, the recency of user engagement, and interactions on the site, BT can aggregate large, yet granular audience to whom advertisers can deliver the most relevant messages. Industry research shows that behaviorally targeted ad spending will reach $4.4 billion by the end of 2012, nearly 25% of US display ad spending [10]. Almost all major online publishers such as Yahoo!, Microsoft and Google have enthusiastically embraced this business model.

Today, the advertising inventory of BT often comes in the form of some kind of demand-driven taxonomy, *e.g.*, *Finance/Loans* and *Life Stages/Parenting and Children*. For each category of interest, BT system builds a model that can derive a response score for each individual from his past online activities (*e.g.*, page views, search queries). The score indicates the likelihood that this user will respond to an ad in that category. The response can be ad clicks or conversions (*e.g.*, product purchases and account sign-ups).

Should the user appear online during a targeting time window, the ad-serving system will qualify this user (to show ads in that category to the user) if her score is above a certain threshold. The threshold is predetermined by domain experts in such a way that both a desired level of *response* (measured by the cumulative click-through-rate) and *reach* (measured by the volume of targeted ad impressions served or the number of qualified users) can be achieved. The revenue generated by BT is a function of both *response* and *reach*.

With the proliferation of social media and social-networking sites, it is now possible to relate the behavior of individuals and their social contacts. In fact, a few startup companies have begun to target consumers based on who they are connected to – generating a lot of buzz around a new advertising model called *social targeting*. If information from social networks can drive more accurate and effective advertising, it is desirable to devote more effort to developing new targeting technologies that combine both behavioral data and social signals. However, before we jump on this bandwagon of *social targeting*, it is important to answer the following fundamental questions:

- *Whether and how can we leverage one's friend activities for behavioral targeting?*
- *Whether forecasts derived from such social information are more accurate than standard behavioral targeting models?*

In this paper, we strive to answer the above questions by

CIKM'11, October 24–28, 2011, Glasgow, Scotland, UK.

evaluating the predictive power of social data on a large online network of over 180 million users in a period of two and a half months. We develop a wide array of supervised and unsupervised machine-learning approaches to incorporate social signals to standard BT models. We conduct extensive experiments to assess the effectiveness of these methods on users with different levels of online activities, and across over 60 consumer domains including *Technology*, *Retail*, *Entertainment*, *Finance*, *Travel*, *Life Stages*, *Automotive*, etc.

As the behavioral and social data are intrinsically in large scale (*e.g.*, tens of terabytes of data and hundreds of billions events in two months), we parallel all the machine-learning algorithms using Hadoop MapReduce framework. Specifically, we have designed and implemented a highly scalable end-to-end solution to conduct large-scale data analysis using Hadoop. Our solution handles the generation of behavioral and social features, model training, scoring, network propagation, and model evaluation in a very efficient and scalable fashion.

To the best of our knowledge, this is the most comprehensive large-scale social-network data analysis in the context of BT. Our study based on real-world applications offers interesting insights into the value of social data for developing the next generation of targeting products. Our findings offer a solid and quantitative guideline for both publishers and advertisers in decision making about social targeting versus behavioral targeting. Our algorithm implementations also serve as the building blocks for future researches in this domain.

Organization of the material: Section 2 reviews related work. Section 3 describes our behavioral and social data. Section 4 introduces BT baseline model and evaluation metrics. In Section 5 we develop a wide array of supervised and unsupervised approaches to evaluate the value of social data for BT. In Section 6 we briefly discuss how these different approaches are implemented on Hadoop using a unified framework. We conclude the paper in Section 7.

2. RELATED WORK

Friends are similar along a variety of dimensions is a long-observed empirical regularity, which sociologists called the *homophily* [16]. The study of this pattern is a recurring theme with increasing interests owing to the boom of online social networking services. Researchers from Microsoft [20] found that people who chat with each other using instant messaging are more likely to share similar personal characteristics (*e.g.*, age, location) and interests (*e.g.*, search topics). Engineers from Facebook [12] developed techniques to infer users' undeclared profiles (*e.g.*, age, gender, profession) from their friends so that advertisers can precisely target more consumers. Scientists from academia also developed models to evaluate the quality of algorithms that derive one's interests from their social contacts [17, 24]. However, most existing work is limited to the inference of "static" profiles such as age, gender, education. Understanding the role of homophily with respect to one's online behavior, and particularly in the context of behavioral targeting has been largely ignored.

The most relevant work to ours is by Bagherjeiran and Parekh [1]. The authors observed that online friends tend to see and click on similar display ads. They developed an ensemble classifier to combine both behavioral and social features to boost the probability that a user will click on an ad. We also evaluated this approach in our experiment (to be described later). We observed that the computational cost of this approach is prohibitively high, which makes it not very much practical in large-scale production systems. Further, our work differs from [1] in that we systematically studied a wide array of supervised and unsupervised data mining strategies to incorporate social data into traditional behavioral targeting. To our best knowledge, our work is the most comprehensive study of the value of social data in the context of behavioral targeting at an unparalleled scale.

Another seemingly relevant work is by Provost *et al.* [18]. The authors proposed to construct a quasi-social network that connects people who visit the same user-generated micro-content sites. Given a set of valued seed customers from advertisers, Provost *et al.* identified more users on the quasi-network who are in close proximity to the seed users for brand advertising. The "proximity" between two users is based on the similarity of the contents they have viewed. The idea, though called "social targeting", is closely related to traditional behavioral targeting because the network is derived from users' browsing activity (*i.e.*, page views).

There is another large body of work in the literature on network *influence modeling*, which is one of the fundamentals to viral marketing [13]. Although Watts [23] challenged the existing influence hypothesis and claimed that the so-called *influencers* in social networks were just accidental, viral marketing recently received increasing attention. Hill *et al.* [11] found that consumers who were linked to prior adopters adopt a telecommunication service at a rate 3-5 times higher than the control group. Bhatt *et al.* [3] also investigated how we might use network information to predict product adoptions. They observed strong signals of peer pressure, but very little evidences of influence from highly connected users. Furthermore, they found that the propagation of the adoption remains mostly local to the initial adopters and their close friends, echoing the discovery made by Bakshy *et al.* [2] that most information cascade in social networks are very shallow. We note that influence analysis is primarily interested in how information spreads over the network, whereas our focus is to understand the value of social data, and how we can leverage social data for behavioral targeting.

3. BEHAVIORAL AND SOCIAL DATA

The analysis and experiments in this paper are conducted on vast amounts of behavioral and social-network data from a large IT company in a period of two and a half month. In this section, we describe the data and its properties.

3.1 Data

Behavioral data: Behavioral data serves as the backbone of our study. It contains individuals' web-browsing behavior such as the pages they have visited or the searches they have made, all aggregated at the BT category level, *e.g.*, *10 page views in category "Retail" at time t*. For evaluation, we split the data into training and test sets. Figure 3 illustrates the generation of both training and test data. The **training data** is collected from a 10-week period of time (2010/08/23–2010/10/31), where the last 4 weeks (2010/10/4–2010/10/31) are used to generate the targets (*i.e.*, clicking on an ad or not). For each user u on each day t_{n+1} in the 4-week target window, we set the *target* to 1 if u clicked on an ad in the BT category being mod-

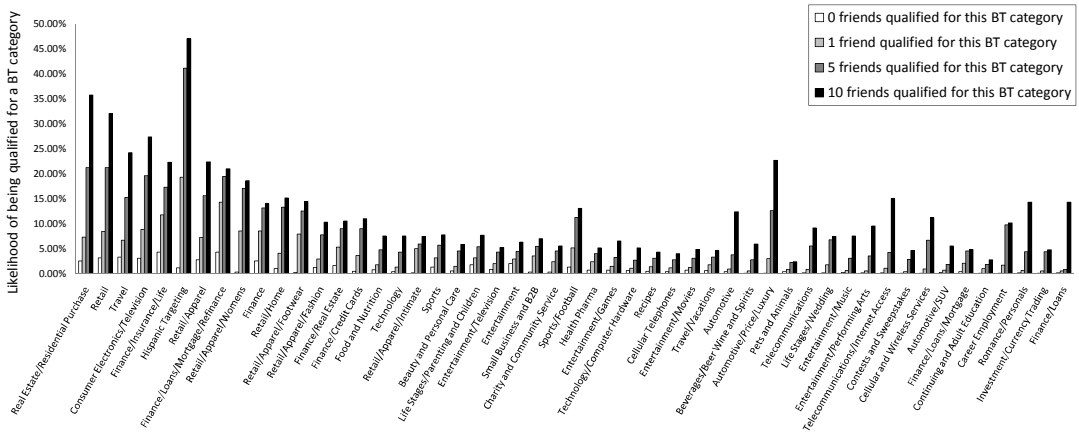

Figure 1: Likelihood of being qualified for a BT category as a function of having social contacts who are also qualified for the same category.

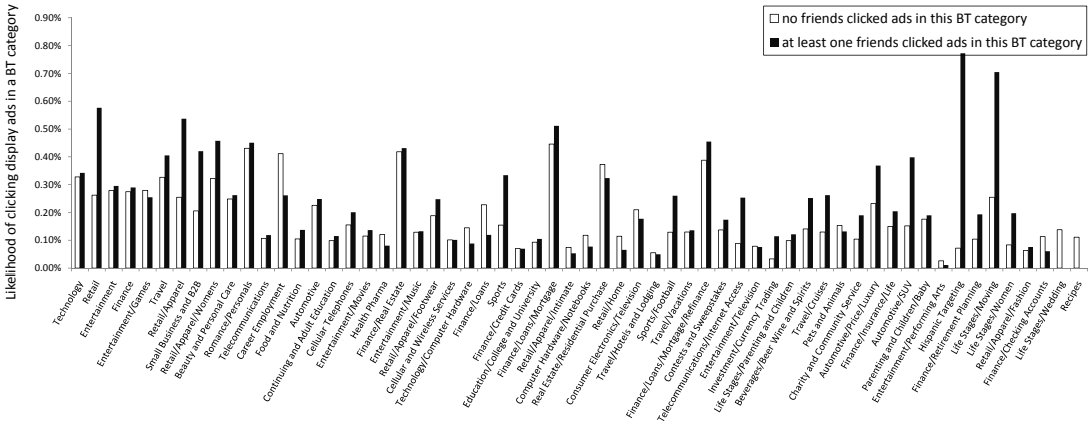

Figure 2: Likelihood of clicking on display advertisements in a BT category as a function of having social contacts who have done the same before.

eled, or 0 if u saw the ad but did not click on it. Next, we create the corresponding *behavioral features* from this user's activities in the preceding 6 weeks (details in Section 4.1). The **test data** is generated from a 7-week period of time (09/20/2010–2010/11/07), where the last 1 week (2010/11/01–2010/11/07) is used to form the targets. This process produces ***13 billion*** training and test examples and approximately ***7 terabyte*** data.

Social data: Our social graph is constructed from users in an Instant Messaging (IM) network operated by a large IT company. We remove singleton users and establish an edge between all pairs of users who mutually authenticate each other as buddies. The resulting network has over ***390 million nodes*** and ***5 billion edges***. Intersecting the behavioral data (training and test, respectively) with this communication network results in approximately ***180 million*** users, for whom we have a record of both their own behaviors as well any behaviors of their friends.

Remarks: We conduct our research in a privacy-friendly fashion. Specifically, we do not use any demographic or geographic information. The behavioral data is aggregated at category level, *e.g.*, 10 *page views in category "Travel/Cruise" at time t*. We do not use any granular user activities.

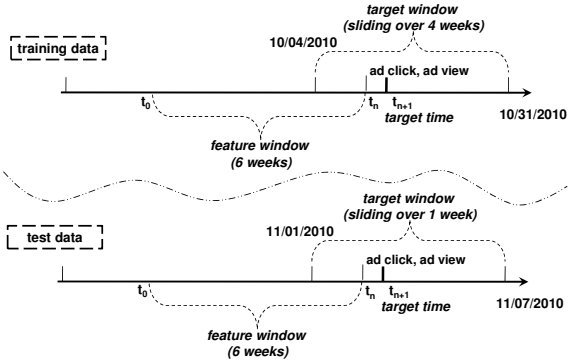

Figure 3: Generating training and test behavioral data.

3.2 Homophily

It is a long-observed empirical regularity that friends are similar on a variety of aspects – a pattern sociologists called *homophily*. As McPherson *et al.* write in their seminal review [16], "homophily limits people's social worlds in a way that has powerful implications for the information they re-

ceive, the attitude they form, and the interactions they experience." In other words, where there is *homophily*, one can in principle predict an individual's behavior based on the information from his or her social contacts. Thus, to assess the value of social data for behavioral targeting, we first attempt to answer the following question: *Can we observe the presence of homophily in our social data, and in particular, along certain dimensions related to behavioral targeting?* We answer this question by studying *BT qualifications* and *ad clicks*.

BT qualifications: Recall that a user is qualified for a BT category if her BT score derived from the corresponding model is above the serving threshold. The plot in Figure 1 shows that it is possible to infer one's BT qualifications from that of her friends. For almost all the 60 major BT categories we studied, users with more friends who are qualified for a certain category are more likely to be qualified for the same category. For example, among all consumers with 5 friends qualified for *Retail*, 21% are also qualified for *Retail*, 6 times higher than consumers with no friends qualified for *Retail*.

Ad clicks: Since a majority of online publishers adopt the pay-per-click model for their BT products, *i.e.*, advertisers pay publishers when their ads are clicked, we also study the homophily of *ad clicks* using our data. Specifically, we compute the likelihood that a user will click on the display ads in a BT category as a function of having friends who have done the same within the last few days. The results are illustrated in Figure 2. Similar to the trend we observed for *BT qualifications*, social data are in general informative for predicting ad clicks as well, though the effect varies considerably across different categories. For 42 of the 60 categories shown in the plot, users whose friends clicked on the ads before have markedly higher rates of clicking themselves, with increases ranging from 0.3% to over 977.0%.

Remarks: Goel and Goldstein [9] also investigated social homophily using data from off-line sales, sign-ups for an online service, and clicks on ten online banner ads. Our findings echo the observations made by them.

4. BEHAVIORAL TARGETING (BT)

In this section, we briefly introduce BT baseline model and evaluation metrics.

4.1 BT Baseline Model

The baseline model [5] takes users' browsing habits as input, and builds a classifier to predict the likelihood that a user is going to click on an ad in a certain BT category. The actual data-mining algorithm to learn the classifier is often not crucial. In fact, it is generally intractable to use algorithms of time complexity higher than linear in solving large-scale machine learning problems of industrial relevance [4]. Our previous experience shows that linear classifiers such as logistic regression, linear regression and support vector machines do not differ significantly in terms of prediction performance. In this paper, we use a customized version of LIBLINEAR [7] to train all models on Hadoop MapReduce platform.

On the other hand, how to construct features for training and scoring has a huge impact on large-scale production systems. As online users constantly change their behaviors by browsing different web pages and searching different subjects, it is generally impractical to continuously create new features and score hundreds of millions of users from scratch – after all, online systems often need to make ad serving decisions in near real time (in the order of millisecond). Next we introduce a simple linear-time method that can incrementally update behavioral features, allowing linear classifiers to incrementally update scores as well.

For each type of user activities $a \in \{$ *page view*, *search query*, *ad click*, ...$\}$ in the BT category being modeled, the baseline model computes two types of input features:

- ***intensity*** I_{a,t_n}: the cumulative count of activity a that the user has performed in the feature time window $[t_0, t_n]$.

$$I_{a,t_n} = \sum_{t=t_0}^{t_n} \alpha^{t_n - t} A_{a,t},$$

  where $A_{a,t}$ denotes number of times the user has activity a at time t, and $\alpha \in [0, 1]$ is a decay factor used to diminish the importance of old events.

- ***recency*** R_{a,t_n}: the time elapsed since the user has performed activity a most recently.

$$R_{a,t_n} = \begin{cases} t_n - t_0, \text{ if } \nexists A_{a,t} > 0, t_0 \le t \le t_n; \\ t_n - \max\{t | A_{a,t} > 0, t_0 \le t \le t_n\}, \text{ otherwise.} \end{cases}$$

As the feature window moves from $[t_0, t_n]$ to $[t_0, t_{n+\delta}]$, it is easy to update features without having to re-process all the prior events:

$$I_{a,t_{n+\delta}} = \alpha^{t_{n+\delta} - t_n} I_{a,t_n} + \sum_{t=t_{n+1}}^{t_{n+\delta}} \alpha^{t_{n+\delta} - t} A_{a,t};$$

$$R_{a,t_{n+\delta}} = \begin{cases} t_{n+\delta} - t, \text{ if } \exists A_{a,t} > 0, t_{n+1} \le t \le t_{n+\delta}; \\ t_{n+\delta} - t_n + R_{a,t_n}, \text{ otherwise.} \end{cases}$$

The target of the model is a binary variable indicating a click on an ad in the category being modeled ($y = 1$), or not ($y = 0$). Although it is possible to use conversions such as product purchases as the target, in this paper we mainly focus on pay-per-click model, where advertisers pay the hosting service when the ad is clicked.

4.2 Evaluation Metrics

We build a BT model for each category c and measure its performance by two metrics: 1) the cumulative CTR of a collection of targeted users whose scores are above a certain serving threshold (or at a certain reach level); and 2) the area under the ROC curve [8].

The cumulative CTR at a certain serving threshold is denoted by $\text{CTR}_c^{\text{reach}}$ and illustrated in Figure 4. It is calculated as the total ad clicks received from users whose scores are above the threshold, divided by total ad impressions served to these users. To eliminate the potential variance across different models, we normalize $\text{CTR}_c^{\text{reach}}$ by the corresponding population CTR where the threshold is set to the minimum (rightmost value on x-axis). We denote this normalized metric by $\text{CTR\_Lift}_c^{\text{reach}}$:

$$\text{CTR\_Lift}_c^{\text{reach}} = \frac{\text{CTR}_c^{\text{reach}}}{\text{CTR}_c^{\text{population}}} - 1. \quad (1)$$

CTR lift provides a sneak peek of the model performance at certain reach levels. To have a global picture, we use the area under the ROC curve, denoted by AUC_c, to examine a

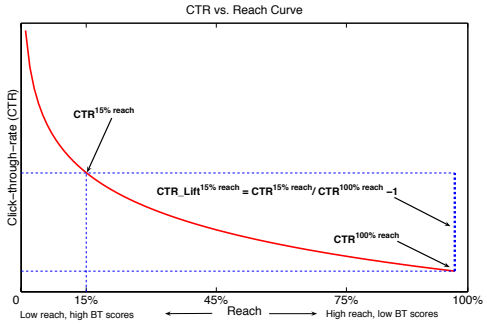

Figure 4: CTR vs. Reach curve. BT scores (sorted in descending order) and reach are along the same x-axis. The serving threshold is set to a value on x-axis where 15% users are qualified. The higher the serving threshold, the less the number of qualified users, and therefore the lower the reach.

model's discrimination power over the entire score distribution. The higher the AUC value, the better the model.

Since we build models for 60 major BT categories in one batch, and ads in different categories receive distinctive serving demands, we report average AUC_c and $\text{CTR\_Lift}_c^{\text{reach}}$, weighted by ad impressions served in each category.

$$\overline{\text{CTR\_Lift}}^{\text{reach}} = \frac{\sum_c \text{CTR\_Lift}_c^{\text{reach}} \cdot v_c}{\sum_c v_c}, \quad (2)$$

$$\overline{\text{AUC}} = \frac{\sum_c \text{AUC}_c \cdot v_c}{\sum_c v_c}, \quad (3)$$

where v_c is the ad impressions in category c. Weighting by ad impressions allows us to pay more attention to revenue-bearing categories which usually have a large amount of contracted impressions to deliver.

5. LEVERAGING SOCIAL DATA FOR BT

In this section, we develop various supervised and unsupervised methods to incorporate social signals into traditional BT. We evaluate the efficacy of these methods through extensive experiments on large-scale real-production data across 60 major consumer domains. Our results offer very interesting insights into the value of social data, allowing us to answer the questions raised before: *How can we leverage one's friends activities for behavioral targeting? Are forecasts derived from such social features more accurate than standard behavioral targeting models?*

5.1 Supervised Approach

5.1.1 BT with Social Features

Our preliminary study in Section 3.2 shows that connected users share similar behavioral patterns such as BT qualifications and ad clicks. Motivated by this finding, we propose to train BT models with additional social features extracted from the network. Specifically, we develop two types of social features: *neighborhood features* and *community features*.

Neighborhood features: These features provide simple statistics of one's social circle. The first set of neighborhood features includes: 1) the number of friends; 2) the number and percentage of *active* friends, where *active* means that the user has certain online activities (*e.g.* browsing pages, clicking ads) in the feature time window; and 3) the number and percentage of ad clickers in one's neighborhood (recall that a user is likely to click the ad if her friends also clicked on the ad). We denote these features by **Neighbors1**.

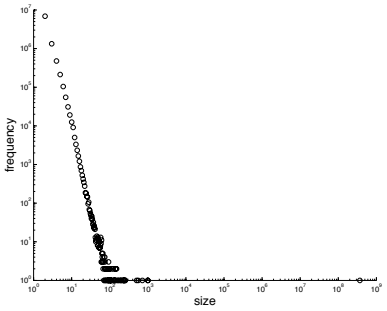

Figure 5: Distribution of the sizes of the connected components in our social network.

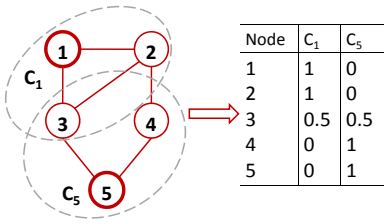

Figure 6: Example of creating community features. Node 1 and 5 are ad clickers, each forming a separate community C_1 and C_5.

Note that ad clicks are extremely rare events – the population CTRs on Automotive and Travel display ads are only about 0.15% and 0.08%, respectively according to a DoubleClick report in 2009 [6]. Thus, the volume of users that can benefit from friends' ad-clicking behavior are quite small. On the other hand, views of web pages and ads represent the most dominant patterns of online activities. Hence, we construct the second set of neighborhood features on top of **Neighbors1** by introducing the number and percentage of friends with page views and ad views in the same category being modeled. We denote these features by **Neighbors2** and **Neighbors1** $\subseteq$ **Neighbors2**. The absolute values of these features may vary drastically in practice (approximately follow a power-law distribution), we apply a logarithmic transformation to scale all quantities to a reasonable range.

Community features: The second approach is to extract latent traits based on network structure. One typical example is the *online community* where members inside the group have more inter-connections than with others outside. Recently, Tang and Liu [21, 22] utilized community membership as features to solve a network-based classification problem. The algorithm first identifies communities from the network, and then treats community memberships as latent features for classical supervised learning. They showed that this approach outperforms other collective-classification methods [19], especially in noisy social-media networks.

However, finding communities in a large-scale social network with 390 million nodes and 5 billion edges is not a trivial task. It is necessary to first understand the over-

	Method	$\Delta_{\overline{\text{AUC}}}$	$\Delta_{\overline{\text{CTR\_Lift}}^{\text{reach}}}$					
			5% reach	10% reach	20% reach	30% reach	40% reach	50% reach
1	**BT baseline**	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
2	**Random targeting**	−24.25%	−100.00%	−100.00%	−100.00%	−100.00%	−100.00%	−100.00%
3	**Neighbors1**	−16.58%	−89.19%	−85.65%	−80.28%	−75.69%	−70.78%	−64.51%
4	**Neighbors2**	−13.34%	−87.19%	−82.27%	−72.90%	−63.85%	−54.18%	−44.81%
5	**Community**	−15.84%	−86.13%	−84.87%	−81.19%	−75.38%	−67.30%	−56.42%
6	**BT + Neighbors1**	+0.97%	−0.14%	+0.22%	+0.84%	+1.20%	+1.73%	+3.63%
7	**BT + Neighbors2**	+0.86%	+0.13%	+0.22%	+0.79%	+1.21%	+1.38%	+2.53%
8	**BT + Community**	+0.08%	−0.08%	−0.08%	+0.07%	+0.43%	+0.62%	+0.63%
9	**Ensemble**	+0.00%	−1.02%	−1.56%	−3.51%	−5.48%	−3.55%	−0.77%

Table 1: Performance improvement over BT baseline model, measured by the lift of view-weighted average AUC and CTR_Lift across all 60 BT categories on all users.

all network structure before delving into any specific algorithms. Figure 5 illustrates the distribution of the connected components in our network. There are more than 9 million connected components, among which the largest component alone covers 94.21% of nodes – a very typical power-law distribution. Standard community-detection algorithms such as matrix factorization and statistical inference, if not well tuned, are very likely to end up finding these connected components instead of real "latent features".

To tackle this challenge, we develop a simple notion of community as *a user and her 1-hop neighborhood*. Recall again that users with clickers in the neighborhood are more likely to click on ads, we only keep communities that are centered at an ad clicker. We treat each such community as a feature; if a user belongs to that community, the corresponding feature value is 1, and 0 otherwise. We further normalize each user's membership features so that they sum up to 1. We denote this type of feature by **Community**. Figure 6 illustrates an example of creating such features.

5.1.2 Ensemble BT with a Social Model

Another way of combining social and behavioral data is to build an ensemble classifier, which merges the outputs of behavioral model with social model to improve predictions. Mathematically, the output of an ensemble model S_e is calculated as

$$S_{\text{ensemble}} = \alpha \cdot S_{\text{behavioral}} + (1-\alpha) \cdot S_{\text{social}}, \quad (4)$$

where $\alpha \in [0,1]$ is a weighting parameter.

A constant weight, say $\alpha = 0.5$, often leads to a poor performance based on our experience. In practice, α is learned through a third classifier that takes both $S_{\text{behavioral}}$ and S_{social} as inputs and the original targets as outputs. Bagherjeiran and Parekh [1] discussed this approach for online advertising in ICDM 2008 workshop. However, we would like point out that the computational cost of this approach is prohibitively high because it needs to train three models: behavioral model, social model, and the ensemble classifier, and score each user multiple times. Thus, this method does not scale very well to large production systems.

5.1.3 Experiments – All Users

We build a bunch of new BT models using the aforementioned social features and ensemble approach. We evaluate these models with respect to the baseline in terms of both $\overline{\text{AUC}}$ and $\overline{\text{CTR\_Lift}}^{\text{reach}}$ (see Section 4.2 for definitions). For illustration purpose, we report the relative improvement of these metrics as follows:

$$\Delta_{\overline{\text{AUC}}} = \left(\frac{\overline{\text{AUC}}_{\text{new}}}{\overline{\text{AUC}}_{\text{baseline}}} - 1 \right) \times 100\%; \quad (5)$$

$$\Delta_{\overline{\text{CTR\_Lift}}^{\text{reach}}} = \left(\frac{\overline{\text{CTR\_Lift}}^{\text{reach}}_{\text{new}}}{\overline{\text{CTR\_Lift}}^{\text{reach}}_{\text{baseline}}} - 1 \right) \times 100\%. \quad (6)$$

The *reach* takes a value of $5\%, 10\%, \ldots, 50\%$, meaning that ad serving thresholds are chosen for each model that top $5\%, 10\%, \ldots$ users are qualified for ad serving.

Results I: The experimental results are summarized in Table 1. A positive value in the cell means that the new model is performing better than the baseline, whereas a negative value indicates that the new model is worse. Line 1 is the baseline model compared with itself, therefore all entries are 0s. Line 2 is a random targeting model that randomly assigns scores to users, so its $\overline{\text{AUC}}_{\text{random}} = 0.5$ and $\overline{\text{CTR\_Lift}}^{\text{reach}}_{\text{random}} = 0\%$ at all reach levels. Line 3–5 are models built from social features alone. From Line 1–5 we have two interesting observations:

- Social features do carry certain informative signals because the models built from social features alone are still better than random targeting.
- BT baseline model substantially outperforms all other models trained from social features alone in terms of both AUC and CTR. In other words, individuals' own behavioral information is much more useful than their friends' in the context of BT.

Line 6–8 are models built from hybrid features that combine both behavioral and social data. We can observe that these models have higher AUC and CTR than baseline, though the improvement seems only marginal (we will elaborate more on this shortly). It is also worth noting that ensemble classifier surprisingly underperforms the BT baseline, possibly due to overfiting in the training phase. Considering its computational cost as well, we exclude it from the subsequent analysis.

5.1.4 Experiments – Type-0, Type-1, Type-2 Users

The experimental results in Table 1 show that models built from the combination of social and individual behavioral features outperform the baseline. But the actual gains

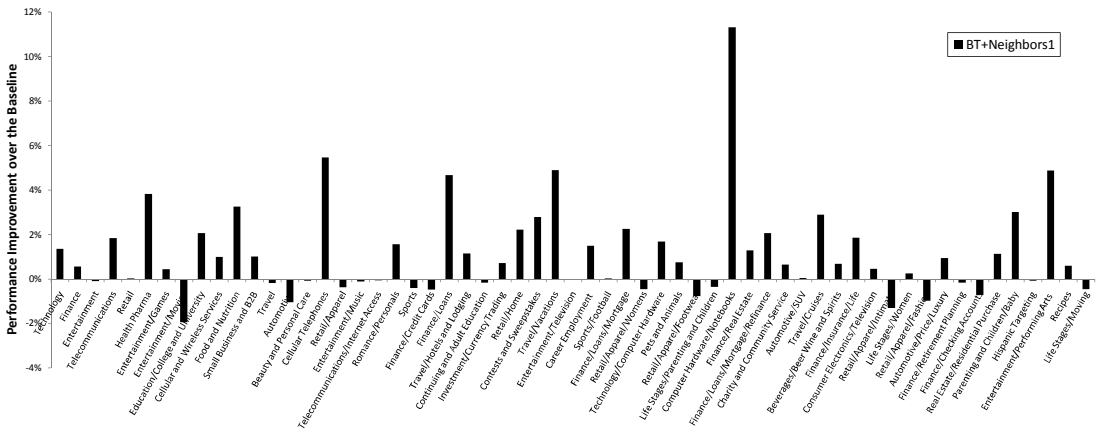

Figure 7: Category-level performance improvement, measured by the relative lift of AUC, *i.e.*, $(\mathbf{AUC}_c/\mathbf{AUC}_{\mathbf{baseline}} - 1) \times 100\%$. Categories are sorted in descending order of ad impressions served in that category. The method is BT + Neighbors1.

on Line 6–9 seem rather marginal (less than 1% of $\Delta_{\overline{\text{AUC}}}$ and $\Delta_{\overline{\text{CTR\_Lift}}^{20\% \text{ reach}}}$). Since the reported metrics are averaged over 60 BT categories, we conducted further analysis to examine the performance improvement of each individual category. In particular, we chose **BT+Neighbors1** method to study as it is one of the best performing methods. The category-level results are shown in Figure 7. We can see that the improvement is highly skewed, some category has as large as 11% boost and some others only less than 1%. So our next questions are: *why could this happen? When does social features help most?*

To answer these questions, we further partition our users into the following three groups:

- Type-0 users: for these users we do not have their behavioral information in the feature time window. It is possible that they are newly registered users; or they do not visit us recently; or they have online activities, but those activities are in a different category not being considered by the model.
- Type-1 users: for these users we only know what ads they have seen in the feature time window. Usually these are mail or messenger only users who do not browse other pages on our site, and their mail pages cannot be captured due to privacy policies. Note that what ads a user can see is often determined by a variety of targeting services including demographic and geographic targeting, search re-targeting, and other behavioral targeting products. Therefore, ad views are still indicative of users' characteristics and interests to certain extent.
- Type-2 users: these are the active users who have lots of online activities which can be utilized for modeling and scoring.

It is clear that the magnitude of behavioral signals vary significantly among these three types of users: Type-0 is the weakest, followed by Type-1, and Type-2 is the strongest.

In the following experiments, we choose the best performing models **BT + Neighbors1**, **BT + Neighbors2** and **BT + Community**, and evaluate their performance against each type of users, in an attempt to answer the questions we raised in this section.

Results II: The experimental results are illustrated in Figure 8. We can observe that

- Type-0 users benefit most from social features. New models improve by +4% with respect to this type of users, much higher than the improvement obtained from all users which is less than 1%. Similarly, for Type-1 users, we also observe a 2% lift. This result implies that social data is most useful for users without much behavioral information.
- On the other hand, the performance of new models decays when evaluated on Type-2 users. It indicates that when users have sufficient behavioral data to characterize themselves, social data can introduce redundancy, if not noise, to the model.

So how does these findings help us answer the questions raised at the beginning of this section? We further investigated categories with large performance improvement and found that they often have more Type-0 and Type-1 users. This is the case when social features are playing important roles. In addition, the findings also help us understand the pattern we observed in Line 6-8 of Table 1, that is, the performance improvement increases as the reach level increases. We found that when the reach is small, only those users with high BT scores are qualified, and most of them are among Type-2 users. Hence, we do not see much gain of performance. As the reach increases, more and more Type-0 and Type-1 users are qualified and targeted with ads, consequently, the value of social features starts to manifest.

5.2 Unsupervised Approach

In this section, we employ network-propagation methods [15, 26, 14, 27] to infer users' BT scores directly from their friends'. The motivation behind this approach is again the *homophily* property we observed in Section 3.2 – users in close proximity share similar behavioral patterns. One advantage of this approach is that we may only need to score a small set of active users and use them as the seeds of the

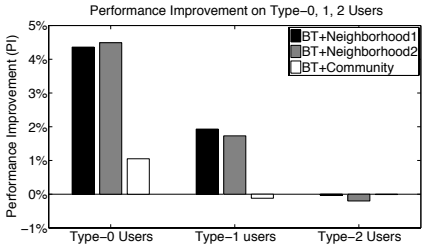

Figure 8: Performance improvement with respect to different types of users, measured by the lift of view-weighted average AUC across all BT 60 categories.

propagation, thus reducing the storage and computational cost of creating a large set of BT features for classification models. We test this idea using three network-propagation schemes as follows.

5.2.1 Methods

SCHEME 1. *Let $s^{(t)}(u)$ denote the score of user u after the t-th iteration, $N(u)$ be the set of users who are friends with u, $d(v)$ be the out-degree of user v (the number of friends of v), $|G|$ be the total number of users in the social network, and $0 < \alpha < 1$ a dumping factor. The first propagation is defined recursively as follows:*

$$s^{(t)}(u) = (1-\alpha) \sum_{v \in N(u)} \frac{s^{(t-1)}(v)}{d(v)} + \alpha \frac{1}{|G|}. \quad (7)$$

SCHEME 1 is essentially the PageRank algorithm [15]. PageRank is used to measure the quality of a web page based on the structure of the hyperlink graph. A page that receives "endorsements" from many other good quality pages in the form of hyperlinks tends to be of good quality too. The stationary state of this propagation depends on graph structure, but not on initial scores.

SCHEME 2. *Let $s^{(0)}(u)$ denote the initial BT score of user u according to the standard BT models, and $0 < \alpha < 1$ a weighting parameter. The second propagation is defined recursively as follows:*

$$s^{(t)}(u) = (1-\alpha) \sum_{v \in N(u)} \frac{s^{(t-1)}(v)}{d(v)} + \alpha s^{(0)}(u). \quad (8)$$

SCHEME 2 has its roots in semi-supervised learning [27]. The basic assumption there is *consistency*: data points close to each other, or on the same cluster or manifold are likely to have the same class labels. The propagation allows every data point to iteratively spread its label information (BT scores in our scenario) to its neighbors until a global state is achieved. During each iteration, each point receives the information from its neighbors, and also retains its initial information. The stationary state of this propagation depends on graph structure as well as the initial scores.

SCHEME 3. *The third propagation is defined recursively as follows:*

$$s^{(t)}(u) = (1-\alpha) \sum_{v \in N(u)} \frac{s^{(t-1)}(v)}{d(v)} + \alpha \frac{1}{|G|} + s^{(t-1)}(u). \quad (9)$$

SCHEME 3 is a variation of PageRank with the exception that a user's score computed in the previous iteration is carried over to the next iteration in computing her new score – a sort of *self-reinforcement*. This propagation is used in [26] to search experts on an author-citation network.

5.2.2 Experiments

Our study of supervised models in Section 5.1 have shown that 1) social features appreciably improve the prediction accuracy for users without much behavioral information (Type-0 and Type-1 users); indicating that these types of users benefit most from their active neighbors; and 2) social features are not quite informative for users with lots of activities (Type-2 user); meaning that for this type of users, baseline model is sufficiently trustworthy. Hence, to evaluate the effectiveness of network-propagation approaches, we hide BT scores (computed from BT baseline model) of Type-0 and Type-1 users (their scores are set to zeros), and initiate propagation from Type-2 users. After each round of propagation, we compute $\Delta_{\overline{\text{AUC}}}$ on Type-0 and Type-1 users respectively to evaluate the performance.

Results III: The experimental results are summarized in Figure 9. Unfortunately, we find that propagation in general does NOT increase prediction accuracy of baseline models on either Type-0 or Type-1 users. Since ad click is rare, it often requires vast amounts of data to train a classification model in order to optimize CTRs. Consequently, unsupervised approaches such as network propagation may not be able to capture this weak signal.

Nevertheless, we still have some interesting observations to point out. First, Type-0 users benefit from the first round of propagation with $\Delta_{\overline{\text{AUC}}}$ of 0.55% while Type-1 users do not; indicating again that when users do not have any behavior data, social information can provide valuable signals. Second, $\Delta_{\overline{\text{AUC}}}$ decays as propagation continues (hold true when the number of iterations $\gg 3$); showing that information from remote friends are noisy and not much useful. Third, the decay of $\Delta_{\overline{\text{AUC}}}$ from propagation SCHEME 3 is slower than other two approaches; implying that the *self-reinforcement* strategy adopted by SCHEME 3 may protect users' scores from being significantly skewed by their neighbors.

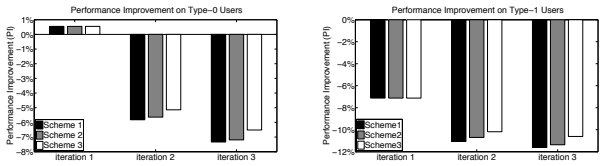

Figure 9: Performance of three network-propagation schemes, measured by the lift of view-weighted average AUC across all 60 BT categories with respect to Type-0 users (Left) and Type-1 users (Right), respectively.

Remarks: It is also possible to initiate propagation from past clickers who often have high BT scores. However, since CTRs are low and the degrees of most nodes are small, relatively few users are connected to any clickers at all. Thus, even though neighbors of clickers have relatively high CTRs, including them in an even moderately-sized set of individuals results in negligible improvement.

6. IMPLEMENTATIONS ON HADOOP

Another contribution we have made is that we designed and implemented a highly scalable end-to-end solution to conduct large-scale data analysis using Hadoop MapReduce framework. Our solution handles the generation of behavioral and social features, model training, scoring, network propagation, and model evaluation in a very efficient fashion. Due to space constraints, we only present a sketch of two major components: *social-feature generation* and *network propagation*. Although they serve different purposes with different specs, we can implement them on Hadoop using a unified framework as follows:

- **Preparation:** join each individual's personal information with her friend list. The output will be user id, the list of her friends' ids, and her personal information. This step is done only once at the very beginning.
- **Map Task:** for each user, emit her personal information to each of her friends. The personal information could be behavioral data (for neighborhood features), clicker data (for community features), or BT scores (for network propagation).
- **Reduce Task:** for each user, aggregate the information received from all her friends to produce social features or prediction scores.

The pseudo-code for *social-feature generation* and *network propagation* are presented in Algorithms 1 and 2.

Algorithm 1: Social-feature Generation

```
1  Mapper(key:uid, value:< adj, f_b >)
   /* adj : graph adjacency list associated with node uid;
      f_b : behavioral features associated with node uid;                    */
2  begin
3      emit(uid, f_b)          /* pass along its own BT info for itself */
4      foreach neighbor ∈ adj do        /* pass info to neighbors */
5          if (FeatureType == Neighborhood) then
6              emit(neighbor, f_b)
7          else if (FeatureType == Community) then
8              if isAdClicker(uid) then  /* pass clicker info only */
9                  emit(neighbor, f_b)
10 Reducer(key:uid, value:[v_1, v_2, ...])
11 begin
12     Q_f ← φ                          /* a queue to store friends BT info */
13     foreach v ∈ [v_1, v_2, ...] do
14         if fromNeighbor(v) then enqueue v into Q_f ;
15         else if fromSelf(v) then f_b ← v
16     if (FeatureType == Neighborhood) then
17         f_s ← CreateNeighborhoodFeature(Q_f, v)
18     else if (FeatureType == Community) then
19         f_s ← CreateCommunityFeature(Q_f, v)
       /* the outputs serve as inputs to model training and scoring      */
20     emit(uid, <f_b, f_s>)
```

In Algorithm 1, a user passes her behavioral information to other users she is connected to (Line 4–9). Note that if we want to create **Community** features, only ad clickers will pass information to neighbors (Line 7–9). In the reducer, each user aggregates all information received from friends (Line 12–15) and invokes CreateNeighborhoodFeature and CreateCommunityFeature functions to construct **Neighborhood** and **Community** features, respectively (Line 16–19). The outputs of the reducer serve as inputs to the model training and the scoring components.

Algorithm 2: Network Propagation

```
1  Mapper(key:uid, value:< adj, s_0, s_t >)
   /* adj : graph adjacency list associated with node uid;
      s_0 : initial BT score;
      s_t : current BT score                                                 */
2  begin
3      emit(uid, < adj, s_0, s_t >)      /* pass along graph structure */
4      s ← (1 − α) · s_t/|adj|            /* α is the dumping factor */
5      foreach neighbor ∈ adj do
6          emit(neighbor, s)             /* pass score to neighbors */
7  Reducer(key:uid, value:[v_1, v_2, ...])
8  begin
9      foreach v ∈ [v_1, v_2, ...] do
10         if isGraph(v) then /* input value contains graph info? */
11             adj ← v.adj; s_0 ← v.s_0; s_t ← v.s_t   /* get graph */
12         else
13             s ← s + v                  /* sum incoming scores */
14     if (Propagation == scheme 1) then s_{t+1} ← s
15     else if (Propagation == scheme 2) then s_{t+1} ← s + α · s_0
16     else if (Propagation == scheme 3) then s_{t+1} ← s + s_t
       /* the outputs serve as input to the next MapReduce iteration */
17     emit(uid, < adj, s_0, s_{t+1} >)
```

In Algorithm 2, the mapper computes for each user the scores need to be distributed to her neighbors (Line 4–6). In the reducer, each user sums up all score contributions from her neighbors and computes updated BT score (Line 12–16). Since it is impossible to maintain a global graph structure in memory, we need to pass along the graph from one iteration to the next. This is accomplished by emitting the adjacency list of each user keyed by the user id (Line 3), and this structure is written back out to disk in the reducer (Line 17). The outputs of the reducer have the same data structure as the inputs to the mapper, which can be used for the next round of MapReduce iteration.

7. CONCLUSIONS

In this paper, We developed a wide-array of supervised and unsupervised methods to leverage social data for BT. We conducted extensive experiments to assess the effectiveness of these methods on a large network of 180 million users, and across 60 consumer domains. To our best knowledge, this is the most comprehensive study of the value of social data for advertising. To conclude, we summarize our major findings here.

1) Social data alone do carry informative signals that can be utilized to compliment standard BT models. However, its value for targeting must be stated carefully as it is not always a *silver bullet*. In our study, categories with a strong homophily effect are more likely to benefit from social data, but the degree of improvement depends on the amount of behavioral information the targeted users have, and how strong the baseline is.

2) Among all the methods we have investigated, appending social features directly to standard BT features seems to be the most effective and scalable way to go.

Finally, we want to point out that this study explores only one aspect of utilizing social-network data for advertising. We mainly treat it as an additional information source, and try to understand its value in the context of BT. However, social networks can be employed in other forms such as improving user engagement with products, and identifying influencers to promote word-of-mouth marketing. All these problems merit further systematic and quantitative study.

8. REFERENCES

[1] A. Bagherjeiran and R. Parekh. Combining behavioral and social network data for online advertising. In *Proceedings of IEEE International Workshop on Data Mining for Design and Marketing (DMDM'08)*, pages 837–846, 2008.

[2] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts. Everyone's an influencer: quantifying influence on twitter. In *Proceedings of the fourth ACM International Conference on Web Search and Data Mining (WSDM'11)*, pages 65–74, 2011.

[3] R. Bhatt, V. Chaoji, and R. Parekh. Predicting product adoption in large-scale social networks. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM'10)*, pages 1039–1048, 2010.

[4] Y. Chen, D. Pavlov, and J. F. Canny. Large-scale behavioral targeting. In *Proceedings of the 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'09)*, pages 209–218, 2009.

[5] C. Y. Chung, J. M. Koran, L.-J. Lin, and H. Yin. Model for generating user profiles in a behavioral targeting system. U.S. Patent 7809740, Issue date: October 5, 2010.

[6] DoubleClick. 2009 year-in-review benchmarks: A doubleclick report. http://www.google.com/doubleclick/pdfs/DoubleClick-07-2010-DoubleClick-Benchmarks-Report-2009-Year-in-Review-US.pdf, 2009.

[7] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[8] T. Fawcett. An introduction to ROC analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[9] S. Goel and D. G. Goldstein. Birds of a feather shop together. `http://messymatters.com/2010/09/01/birdshop/`, September 2010.

[10] D. Hallerman. Behavioral targeting: Marketing trends. `http://www.emarketer.com/Reports/All/Emarketer_2000487.aspx`, June 2008.

[11] S. Hill, F. Provost, and C. Volinsky. Network-based marketing: Identifying likely adopters via consumer networks. *Statistical Science*, 21(2):256–276, 2006.

[12] T. Kendall and D. Zhou. Leveraging information in a social network for inferential targeting of advertisements, April 2009. US Patent App. 12/419,958.

[13] J. Leskovec, L. Adamic, and B. Huberman. The dynamics of viral marketing. *ACM Transactions on the Web*, 1(1):5, 2007.

[14] S. A. Macskassy and F. Provost. Classification in networked data: A toolkit and a univariate case study. *Journal of Machine Learning Research*, 8:935–983, 2007.

[15] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*, chapter 21, page 464. Cambridge University Press, 2008.

[16] M. McPherson, L. Smith-Lovin, and J. M. Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27:415–444, 2001.

[17] A. Mislove, B. Viswanath, K. P. Gummadi, and P. Druschel. You are who you know: inferring user profiles in online social networks. In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining (WSDM'10)*, pages 251–260, 2010.

[18] F. Provost, B. Dalessandro, R. Hook, X. Zhang, and A. Murray. Audience selection for on-line brand advertising: privacy-friendly social network targeting. In *Proceedings of the 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'09)*, pages 707–716, 2009.

[19] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93, 2008.

[20] P. Singla and M. Richardson. Yes, there is a correlation – from social networks to personal behavior on the web. In *Proceeding of the 17th International Conference on World Wide Web (WWW'08)*, pages 655–664, 2008.

[21] L. Tang and H. Liu. Relational learning via latent social dimensions. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'09)*, pages 817–826, 2009.

[22] L. Tang and H. Liu. Leveraging social media networks for classification. *Data Mining and Knowledge Discovery*, 23(3):447–478, 2011.

[23] D. Watts. Challenging the influentials hypothesis. *WOMMA Measuring Word of Mouth*, 3:201–211, 2007.

[24] Z. Wen and C.-Y. Lin. On the quality of inferring interests from social neighbors. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'10)*, pages 373–382, 2010.

[25] 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 (WWW'09)*, pages 261–270, 2009.

[26] J. Zhang, J. Tang, and J. Li. Expert finding in a social network. In *Advances in Databases: Concepts, Systems and Applications*, volume 4443 of *Lecture Notes in Computer Science*, pages 1066–1069. Springer Berlin / Heidelberg, 2007.

[27] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems*, volume 16, pages 321–328. MIT Press, 2004.