Learning Relevance from Heterogeneous Social Network and Its Application in Online Targeting

Chi Wang[*]
UIUC
chiwang1@illinois.edu

Rajat Raina
Facebook
rajatr@fb.com

David Fong[†]
Stanford University
clfong@stanford.edu

Ding Zhou
Facebook
dzhou@fb.com

Jiawei Han
UIUC
hanj@illinois.edu

Greg Badros
Facebook
badros@fb.com

ABSTRACT

The rise of social networking services in recent years presents new research challenges for matching users with interesting content. While the content-rich nature of these social networks offers many cues on "interests" of a user such as text in user-generated content, the links in the network, and user demographic information, there is a lack of successful methods for combining such heterogeneous data to model interest and relevance. This paper proposes a new method for modeling user interest from heterogeneous data sources with distinct but unknown importance. The model leverages links in the social graph by integrating the conceptual representation of a user's linked objects. The proposed method seeks a scalable relevance model of user interest, that can be discriminatively optimized for various relevance-centric problems, such as Internet advertisement selection, recommendation, and web search personalization.

We apply our algorithm to the task of selecting relevant ads for users on Facebook's social network. We demonstrate that our algorithm can be scaled to work with historical data for all users, and learns interesting associations between concept classes automatically. We also show that using the learnt user model to predict the relevance of an ad is the single most important signal in our ranking system for new ads (with no historical clickthrough data), and overall leads to an improvement in the accuracy of the clickthrough rate prediction, a key problem in online advertising.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data Mining*

General Terms

Algorithms, Experimentation, Economics

Keywords

Online Advertising, Heterogeneous Social Networks, Cognitive Relevance, Clickthrough Prediction

1. INTRODUCTION

Social networking services such as Facebook, Youtube, Twitter and MySpace have attracted hundreds of millions of users and become part of the Internet mainstream and an important part of our daily lives. Compared to traditional interest-based online communities, social networking services offer an additional network layer on top of user interest, as they center around people and their connections [6]. Social network users do not usually state their interest explicitly, but instead make connections, including direct connections with other users and groups, and implicit connections formed by interactions—such as commenting or clicking behavior—with user-generated content and other entities in the network, as shown in Figure 1. These connections provide valuable information about a user's interest, but are typically of varying importance and quality. The ability to infer a coherent model of users' interests from the heterogeneous parts of the network has the potential to provide personalized recommendation and more relevant advertisement targeting [23, 8].

Although the ideas developed in this paper can be applied to recommendation and other content matching tasks, we focus on the specific task of serving relevant advertisements to users in a social network. The foundation of pay-per-click advertising systems is a system for predicting the clickthrough rate (CTR) of an ad for a given user or query. Typically, such predictions are based on a machine learning model that uses various hand-crafted features, and is trained on historical click data. Features employed in current social networking sites mainly inherit the keyword-targeting prototype that is successful in search engine advertising, and demography-targeted advertising that is prevalent in traditional brand advertising. However, the characteristics of social networking sites are not fully exploited by these methods. On one hand, the large amounts of data about user activity and social context are unique in social network services and offer the opportunity for better inference of personalized interests than traditional methods. For example,

[*] This work was done while the author was at Facebook.
[†] This work was done while the author was at Facebook.

SIGIR'11, Jul 24–28, 2011, Beijing, China.
Copyright 2011 ACM 78-1-4503-0757-4/11/07 ...$10.00.

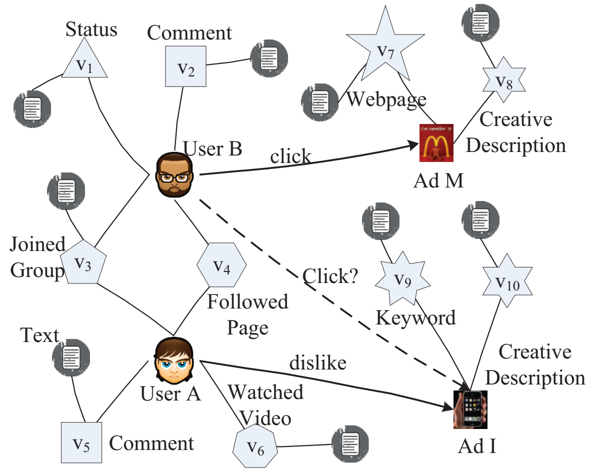


Figure 1: An example of user-ad-source heterogeneous network. Users are linked to different types of objects. These objects typically have associated text information. Ads are also linked to different type of objects with supporting text. We know some ads are clicked and some are shown but not clicked, for a small fraction of user-ad pairs, and we need to predict clicks for other user-ad pairs.

we can use user-generated content as contextual signals for content-based advertising. On the other hand, it is nontrivial to utilize these data. First, not all the contents have strong contextual relevance. For example, when users check friends' news or talk with friends, their intent is usually not related to business. Thus we cannot simply concatenate all the text from linked objects to augment a user's profile. Second, if we use keyword or concept matching to measure the relevance independently of the targeting problem, simply counting the exact matching concepts or even related concepts in the taxonomy cannot capture the latent signal—for example, an ad about Photographers[1] is better targeted to users interested in Weddings, rather than users interested in Photography.

In this paper, we formulate and tackle the problem of relevance learning for online targeting in heterogeneous social networks. To develop a better relevance model in the context of a heterogeneous social network, we propose a framework for CTR prediction based on a variant vector space model. First, for dimensionality reduction, we use semantically meaningful concepts to abstract the interest of users and the topic of ads, such as ”Cartoon Movie” and ”Music Bands and Artists", and thus both user and ad can be represented by a concept vector where each component is the weight for corresponding topic. Second, rather than a simplified view of a friendship social graph, we use the multi-typed networks to characterize user interaction with pages, groups, and other entities. These nodes have text content from which we can extract concepts, and serve as concept sources to their linked objects. Then user interests and other target content can be summarized from the concept classes of linked source nodes. Finally, to quantitatively distinguish the level of importance of different types of nodes in the network, and to systematically capture the intuition that a user concept may match multiple ad concepts with different association weights, we develop a generic user model to learn the weight for heterogeneous linked sources and the association between every pair of user concept and target concept jointly. The model can be applied on top of any concept extraction method.

We demonstrate a series of offline and online experiments that validate our hypothesis: 1) our model is capable of finding hidden associations between user concepts and ad concepts, such as Photographers with Weddings; 2) finding concept class associations and source importance weighting mutually enhance each other in terms of resolving the mismatching between user concept space and ad concept space; and 3) demography and keywords matching help with targeting, but not as precisely as our jointly learnt user model.

2. RELATED WORK

2.1 Online Advertising and User Modeling

Online advertisement targeting and ranking have been widely studied in the context of search-based and content-based targeting, *e.g.*, Google AdWords and AdSense. Search-based targeting, or sponsored search can be thought of as a document retrieval problem, where the ads are the "documents" to be retrieved given a search query. Content-based targeting, a.k.a., contextual advertising, is a sister technology to sponsored search, where the ad retrieval is based on a webpage. Studies on such models show that the vocabulary mismatch problem is critical [9, 20, 7]. Motivated by that, we represent user interests and ads with vectors in a space of concept classes—which is significantly lower dimensional than the space of possible keywords—and learn a weighted matching function based on click history data. There are only a few recent published studies on user-centric targeting in the context of social network sites, perhaps due to the lack of data from widely-used social network sites. Bao *et al.* [4] proposed a influence-based diffusion model for targeting on implicit-relationship Q&A websites with little user-generated content. In comparison, we study a social network site with abundant user-generated content.

In terms of modeling user interests for search and recommendation systems, most works concentrate on one type of user generated data, while on a heterogeneous social network various sources can be used to analyze a user's behavior. Examples include user interaction history during web search [3], user generated tags [21], and browsing behavior in quasi-social networks [19]. A recent work by Wen *et al.* [24] studied users in social networks where there are multiple types of data. However, they assign each source an empirical weight when combining them and study the interest independently of the targeting task, while our model learns the weights from history data and can optimize the relevance measure towards different tasks.

Recent researchers also start to leverage social cues to enhance user interest modeling. Most of them augment the interest of one user from other users. Piwowarski and Zaragoza [18], White *et al.* [25] combine the interests of other users that visit the same page. Likewise, in collaborative filtering tasks it is assumed that those who had similar opinions on a set of items tend to agree again on other items [13, 14]. Konstas *et al.* [16] shows that incorporation of friendship and social tagging can improve the performance of a music recommendation system. Nevertheless their random-

[1] We use capitalized phrases to represent a concept class.

walk-based approach suffer from the scalability issue in large social networks. Distinct with these works, we do not explicitly reply on the friendship link to propagate interests, as previous studies already suggest that explicit relationships are not good indicators of the nature of user interactions[4] and the quality of inferring user interests from their friends varies [24]. Instead, we use links to other entity nodes to propagate interests. These nodes include the interaction with friends such as wallposts, as well as similar behaviors such as liking a page or joining a group. In this way we will not propagate the interests between one and his friends who never have interaction, while the users who share more similar linked objects will have more interest augmentation to each other through the training process.

2.2 Concept Extraction

User interests and ad contents can be extracted by unsupervised or supervised methods. Compared to the classification-based methods, unsupervised topic models such as Latent Dirichlet Allocation [5] are not restricted to a predefined set of labels, but are harder to use and modify when working with thousands of possible topics. With the development of knowledge representation language [11], domain specific ontologies can be constructed and used to label web content according to a concept taxonomy. Supervised learning based on a reference ontology can achieve a good semantic representation of user interests [22, 10], and has the advantages that it is scalable and the results corresponding to a predefined ontology are easy to interpret.

3. PROBLEM STATEMENT

We consider the task of predicting the clickthrough rate for a user on a target object recommended to the user. In our discussions, we assume the target object to be an ad.

Formally, our input is a heterogeneous network $G = (V, E)$. The set of vertices V can be partitioned into user nodes V^u, ad nodes V^a and their various linked source nodes: $V = V^u \cup V^a \cup \Lambda^u \cup \Lambda^a$, where $\Lambda^u = \cup_{i=1}^{n_u} \Lambda_i^u$ and $\Lambda^a = \cup_{j=1}^{n_a} \Lambda_j^a$ are the respective sources. Each user node $v_x \in V^u$ can be linked to one or more source nodes in $\Lambda_i^u, 1 \leq i \leq n_u$; each ad node $v_y \in V^a$ can be linked to one or more source node in $\Lambda_j^a, 1 \leq j \leq n_a$; and each source node $v_z \in \Lambda^u \cup \Lambda^a$ has some text associated with it. In principle we can define source types based on both object type and link type. In this work, we regard each type of object as a type of source, and assume there are two types of links, positive and negative, *e.g.*, "like" and "dislike". The linked sources accordingly have positive or negative contributions to the concepts of the linked user or ad node. For some user and ad node pairs, we know the existence of past impressions and whether the user clicked the ad. For any given pair of user node and ad node, our task is to predict the probability that the user will click on the ad.

Our output will serve as a feature in a machine learning model, to be trained with many different features. To enable an ad ranking system to examine the relevance between any user and any ad inside a large-scale social network in real time, we need to represent user interest and ad content in a concise yet semantically meaningful way allowing efficient measure of the relevance. We use clustered concept classes that are more precise cognitive unit than terms to summarize a user's interest. These concepts also serve as a common representation for integrating information across different sources.

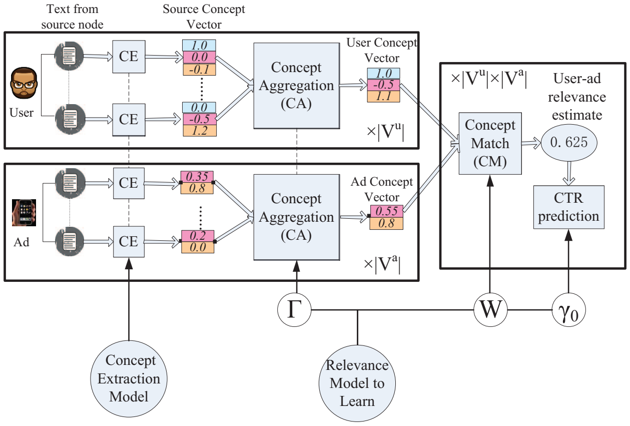


Figure 2: The pipeline of our relevance feature generation for CTR prediction and the model to learn

Figure 2 outlines the pipeline for our relevance prediction. We first give some definitions.

- *Concept Space.* A concept space is a d-dimensional real number space $\mathbb{R}^d$ that encodes some cognitive classification system such as ontology. Each dimension represents a concept class in the system. For example, in a human-edited Web directory [2], each category in the directory can be used as a concept class.
- *Concept Vector.* A concept vector is a vector in the concept space defined above that represents the cognitive property of an object. The component in each dimension indicates how strong the object is associated with the corresponding concept. For example, (Movie:0.5, Soccer:0.3) is a short representation of a sparse concept vector where all the weights are zero except for two concepts. For generality we allow user concept vector and ad concept vector to be in different concept spaces.
- *Concept Extraction(CE).* Given a source node's associated text, output its concept vector $\mathbf{c}$ in certain concept space.
- *Concept Aggregation(CA).* Given a user or an ad v_x, and the set of linked source nodes S_x—for each node $v_i \in S_x$, the corresponding concept vectors are already extracted as $\mathbf{c}_i$—output a concept vector $\mathbf{c}$ for this user or ad.
- *Concept Matching(CM).* Given a pair of user concept vector $\mathbf{u}$ and ad concept vector $\mathbf{a}$, produce a real value feature f to predict the likelihood of the user clicking the ad.

The pipeline of CE→ CA → CM allows user concept vector and ad concept vector be extracted and aggregated offline, and stored in high speed medium. The online feature computation can be efficient with fast access to the concept vector and fast vector matching algorithm. The three modules are tuned in the following manner. First, we choose a supervised concept extraction model. Then we learn the parameters in the rest two models and test them on click data. In this paper we focus on how to learn a good relevance prediction model that manipulates the concept aggregation and matching module for better CTR prediction.

Table 1: Examples of the text concept extraction model used for the experiments. There are tens of thousands of possible concepts, and so the output concept vector is very sparse (tens of nonzero dimensions typically after thresholding).

Input text	Sample concepts with weights
Get low cost car insurance.	Business/Financial_Services/Insurance/Automotive,0.36; Home/Personal_Finance/Insurance,0.30; ...
I am cooking for mom today.	Home/Cooking,0.20; Home/Family/Parenting/Mothers,0.16; ...

4. LEARNING RELEVANCE

We first briefly describe our concept extraction method. Then we discuss several baseline methods for relevance prediction, followed by our solution.

4.1 A Handcrafted Relevance Estimate

Learning Concepts. The task of concept extraction is to transform given text into the most salient topics that occur in the text. A variety of text modeling approaches are possible here, including categorization approaches that classify text into a pre-defined ontology of topics, and topic modeling approaches that automatically find semantically meaningful clusters [5]. While our methods are applicable to any finite vector representation of users and ads, for concreteness, we present results using a categorization approach. Using labeled webpage data from a reference ontology, we trained a large-vocabulary, bag-of-words based linear text classifier that accepts an arbitrary text input, and attempts to predict the strength of association with each label in the ontology. We use each label as a possible concept class, and use the predicted strength as the weight on that concept, thus transforming any input text into a concept vector. This model is not the focus of the paper, and we simply illustrate the method by examples in Table 1.

A baseline for Relevance Prediction. Based on the output of the concept extraction model above, we have a simple method to aggregate the concept vectors and measure the relevance between users and ads. For each user and each ad, we can use unweighted sum of the concept vectors extracted from all linked sources. Then we compute the cosine similarity of two concept vectors as the baseline feature for prediction.

There are several problems with this baseline feature. First of all, the different type of sources should have different prediction power of users' interests in ads. For example, greetings between friends may not be as useful as shared links to a movie or clicks on a previous ad. It is natural to assume that weighted sum of the concept vector from different sources is a better strategy for relevance inference. However, empirical assignment of source weights is not straightforward. It is not obvious whether the group a user belongs to should have higher weight than the posts on the wall. Not everyone publishes status messages, but most have profiles. Different sources have different level of quality, coverage and relevance to business. To find the best weighting we need to learn from the data.

Second, the cosine similarity metric needs to be improved due to the following reasons: 1) different concept may have different power for distinguishing the interest, *e.g.*, top level node Society *VS.* leaf node Society/Relationships/Dating; 2) the intuition that one user concept can match multiple ad concepts (*e.g.*, the video game FIFA Series can match the computer game or the sport soccer) and vice versa; and 3) user concept space and ad concept space are not necessarily identical. So a better measure should allow one user concept to match different ad concepts and vice versa, and does not require they are in the same concept space. This measure is even harder to handcraft, but can be learnt from data.

Now we have two more improved methods, one to learn source weights and the other to learn a better measure. A natural question is can we combine the two to obtain an even better model, and whether they interact with each other. Our intuition is that better assessment of source importance should help us find a better measure; and the better matching function should also facilitate the noise reduction and quality of integration from different sources. Based on that intuition we develop a joint model to integrate the learning of these two parts, and it incorporates all the above means as special cases.

4.2 Jointly Learning Source Weights and Concept Association

Based on above analysis, we add unknown parameters to the aggregation and matching model. Our relevance model consists of both aggregation and matching, and we will learn the parameters for them together from training data. For aggregation, we combine the user concept vector from each source by weighted summation,

$$\mathbf{u}_x = \sum_{i=1}^{n_u} \sum_{v_j \in \Lambda_i^u \cap S_x} \gamma_i^u \mathbf{c}_j = \sum_{i=1}^{n_u} \gamma_i^u \mathbf{s}_i^x = \mathbf{U}_x \mathbf{\Gamma}_u \qquad (1)$$

where $\mathbf{c}_j$ is the concept vector from the node v_j, S_x is the set of source nodes linked with user v_x, and γ_i^u is the weight for the i-th type of user source. For simplification we let $\mathbf{s}_i^x = \sum_{v_j \in \Lambda_i^u \cap S_x} \mathbf{c}_j \in \mathbb{R}^{d_u}$ be the sum of concept vectors from the i-th type source. Then we have the matrix representation $\mathbf{U}_x \mathbf{\Gamma}_u$, where $\mathbf{\Gamma}_u = (\gamma_1^u, \ldots, \gamma_{n_u}^u)^T$ and $\mathbf{U}_x = (\mathbf{s}_1^x, \ldots, \mathbf{s}_{n_u}^x)$. Likewise the ad concept vector will be

$$\mathbf{a}_y = \sum_{i=1}^{n_a} \sum_{v_j \in \Lambda_i^a \cap S_y} \gamma_i^a \mathbf{c}_j = \sum_{i=1}^{n_a} \gamma_i^a \mathbf{t}_i^y = \mathbf{A}_y \mathbf{\Gamma}_a \qquad (2)$$

where $\mathbf{t}_i^y = \sum_{v_j \in \Lambda_i^a \cap S_y} \mathbf{c}_j \in \mathbb{R}^{d_a}$, $\mathbf{\Gamma}_a = (\gamma_1^a, \ldots, \gamma_{n_a}^a)^T$, and $\mathbf{A}_y = (\mathbf{t}_1^y, \ldots, \mathbf{t}_{n_a}^y)$. And for concept vector matching we allow any user concept in user concept space $\mathbb{R}^{d_u}$ to match any ad concept in ad concept space $\mathbb{R}^{d_a}$, with a pairwise weight $w_{i,j}$. Formally, we have the definition for the normalized bilinear form similarity sim as follows:

$$\begin{aligned} sim(\mathbf{u}_x, \mathbf{a}_y) &= \frac{1}{\|\mathbf{u}_x\| \, \|\mathbf{a}_y\|} \sum_{i=1}^{d_u} \sum_{j=1}^{d_a} w_{i,j} u_{x,i} a_{y,j} \\ &= \frac{1}{\|\mathbf{u}_x\| \, \|\mathbf{a}_y\|} \mathbf{u}_x^T \mathbf{W} \mathbf{a}_y \qquad (3) \end{aligned}$$

where $u_{x,i}$ is the i-th component of the vector $\mathbf{u}_x$ and $a_{y,j}$ is the j-th component of the vector $\mathbf{a}_y$. Here $\|.\|$ is the L2 normal number of the vector, and $\mathbf{W} = [w_{i,j}]_{d_u \times d_a}$ is the transformation matrix as it transforms a user concept vector $\mathbf{u}_x$ into ad concept space $\mathbf{u}_x \mathbf{W} \in \mathbb{R}^{d_a}$. $\mathbf{W}$ virtually defines a bilinear form from $\mathbb{R}^{d_u} \times \mathbb{R}^{d_a} \longrightarrow \mathbb{R}$.

Substituting Eq. (1) and Eq. (2) into Eq. (3), we have a new relevance feature by using weighted sources and normalized bilinear form similarity,

$$f(x,y) \quad = \quad \frac{1}{\|\mathbf{U}_x\mathbf{\Gamma}_u\| \, \|\mathbf{A}_y\mathbf{\Gamma}_a\|}\mathbf{\Gamma}_u^T\mathbf{U}_x^T\mathbf{W}\mathbf{A}_y\mathbf{\Gamma}_a \tag{4}$$

We model the probability that a user x will click on an ad y as a generalized linear model in $f(x,y)$. Specifically, using the sigmoid function $g(x) = \frac{1}{1+\exp(-x)}$, we assume:

$$p(\text{user } x \text{ clicks ad } y) \quad = \quad g(f(x,y) + \gamma_0) \tag{5}$$

We can now collect historical data (x_i, y_i, z_i) on user x_i being shown ad y_i, with $z_i = 1$ when this resulted in a click (and $z_i = 0$ otherwise). We learn the unknown parameters in the model by maximizing the discriminative likelihood of this data:

$$L = \sum_{i=1}^{N} \log g((2z_i - 1)(f(x_i, y_i) + \gamma_0)) \tag{6}$$

where N is the number of training examples.

We have three sets of parameters to learn: the user source weights, the ad source weights, and the transformation matrix. Since our concept extraction model can produce tens of thousands distinct concepts, the above learning problem can have hundreds of millions of free parameters. Using a stochastic gradient descent algorithm, we can scan the training data once and iteratively update the three sets of parameters one set after another, as shown in Algorithm 1. Given a training example (x, y, z) from training data, the error in our prediction is given by $z - p(\text{user } x \text{ clicks ad } y)$, or simply $z - p$. We can derive the following gradient descent update rules:

$$\Delta\gamma_0 \quad = \quad r_0(z-p) \tag{7}$$

$$\Delta\mathbf{\Gamma}_u \quad = \quad \frac{r_1(z-p)}{\|\mathbf{u}_x\| \, \|\mathbf{a}_y\|}\mathbf{U}_x^T(I - \frac{\mathbf{u}_x\mathbf{u}_x^T}{\|\mathbf{u}_x\|^2})\mathbf{W}\mathbf{a}_y \tag{8}$$

$$\Delta\mathbf{\Gamma}_a \quad = \quad \frac{r_2(z-p)}{\|\mathbf{u}_x\| \, \|\mathbf{a}_y\|}\mathbf{A}_y^T(I - \frac{\mathbf{a}_y\mathbf{a}_y^T}{\|\mathbf{a}_y\|^2})\mathbf{W}^T\mathbf{u}_x \tag{9}$$

$$\Delta\mathbf{W} \quad = \quad \frac{r_3(z-p)}{\|\mathbf{u}_x\| \, \|\mathbf{a}_y\|}\mathbf{u}_x\mathbf{a}_y^T \tag{10}$$

where r_k is the learning rate for k-th set of parameters. Intuitive choices are $r_0 < r_1 = r_2 < r_3$ because the number of parameters in 3rd set is significantly larger than the first two sets.

The optimization problem is convex if we have only one set of parameters $\mathbf{W}$; in that case it is identical to logistic regression. However, when $\mathbf{\Gamma}_u$ and $\mathbf{\Gamma}_a$ are introduced the objective function is not convex. The gradient descent algorithm converges to a local maximum. The advantage of our stochastic gradient descent algorithm is that a single update is very efficient. We do not need to update all the parameters at each iteration, but just a small fraction of related ones. Assuming the user concept vector and ad concept vector contain l_u and l_a concepts with nonzero weights, only $(l_u l_a + n_u + n_a)$ parameters need be updated. Eq. (8) and (9) dominate the computation time. By storing $\mathbf{W}$ in a big hash table, the $\mathbf{u}_x^T\mathbf{W}\mathbf{a}_y$ can be computed in $O(l_u l_a)$ time. The complexity for training on one sample is $O((n_u + n_a)l_u l_a)$.

The algorithm applies to any general concept vector space. It can learn features of user-ad matching even when the user vector and ad vector are in different spaces. For example, to learn the relevance between user demography and ad concepts, we can let each component of the user vector represent a demography group, and learn the transformation matrix $\mathbf{W}$. Without source weight to learn, the objective function is convex and the algorithm converges to global optimum.

Algorithm 1: Learning without regularization.

```
Input: number of sources $n_u$, $n_a$, user concept space
       $\mathbb{R}^{d_u}$, ad concept space $\mathbb{R}^{d_a}$, learning rates
       $r_0$,$r_1$,$r_2$,$r_3$,overall CTR $c_0$.
Data: training set $T = \{(z_{x,y}, \mathbf{U}_x, \mathbf{A}_y)\}$
Output: $\gamma_0$
Result: $\mathbf{W}, \mathbf{\Gamma}_u, \mathbf{\Gamma}_a$
Initialize all $w_{i,j} \leftarrow 0, 1 \leq i \leq d_u, 1 \leq j \leq d_a$;
Initialize all $\gamma_i^u, \gamma_j^a \leftarrow 1, 1 \leq i \leq n_u, 1 \leq j \leq n_a$;
Initialize $\gamma_0 \leftarrow c_0$;
repeat
    foreach training example $(z_{x,y}, \mathbf{U}_x, \mathbf{A}_y)$ do
1.1     $z \leftarrow z_{x,y}$;
        Combine user sources $\mathbf{u}_x \leftarrow \mathbf{U}_x\mathbf{\Gamma}_u$;
        Combine ad sources $\mathbf{a}_y \leftarrow \mathbf{A}_y\mathbf{\Gamma}_a$;
        Compute prediction $p$ according to Eq. (5);
1.2     Update $\gamma_0$, $\mathbf{\Gamma}_u$, $\mathbf{\Gamma}_a$, $\mathbf{W}$ according to Eq. (7)-(10);
    end
until convergence;
```

We further suggest regularizing the transformation matrix $\mathbf{W}$ for two reasons. First, such regularization has been shown to reduce overfitting and improve generalization to future data, especially in problems with very large numbers of free parameters [17]. Second, the online computation for *sim* is more expensive than the dot product. It takes $O(l_u l_a)$ time even if we assume the lookup of an element in $\mathbf{W}$ takes $O(1)$ time, while the dot product only requires $O(l_u + l_a)$ time if we use the sorted sparse vector as data structure. To optimize it we can precompute the vector $\mathbf{u}_x^T\mathbf{W}$ and $\|\mathbf{u}_x\|$ for every user, and just do dot product for online computation. Using the L1 regularizer provides a principled way to reduce the number of nonzeros in $\mathbf{W}$.

With L1 regularization term, we now minimize the following objective function:

$$L' = -L + C\sum_{i,j} |w_{i,j}| = -L + C\,\|\mathbf{W}\|_1 \tag{11}$$

where large values of $C > 0$ produce fewer number of nonzeros in $\mathbf{W}$, at the potential cost of modeling power.

As before, we can use gradient descent algorithm, with an extra step to clip values at zero. The update rule becomes

$$\Delta\mathbf{W} = \frac{r_3(z-p)}{\|\mathbf{u}_x\| \, \|\mathbf{a}_y\|}\mathbf{u}_x\mathbf{a}_y^T - r_3C * sgn(\mathbf{W}) \tag{12}$$

Algorithm 2 shows the new online learning algorithm. We partition the training data to m chunks, and do the regularization after every $k = N/m$ examples. The complexity depends on the number of nonzero terms in $\mathbf{W}$, say n_w. The running time for training on N samples with m chunks is $O(N(n_u + n_a)l_u l_a + mn_w)$. In our data l_u, l_a, n_u and n_a are constants in the same order of magnitude as 10. n_w remains almost constant, although in the order of magnitude 10^6. So the training time grows linearly with the sample size with these constraints.

Algorithm 2: Learning with an L1 regularizer

```
Input: # of sources $n_u$, $n_a$, user and ad concept space
       $\mathbb{R}^{d_u}$, $\mathbb{R}^{d_a}$, learning rates $r_0$,$r_1$,$r_2$,$r_3$, overall
       CTR $c_0$, threshold $t$, regularization step $m$.
Data: training set $T = \{(z_{x,y}, \mathbf{U}_x, \mathbf{A}_y)\}$
Output: $\gamma_0$
Result: $\mathbf{W}, \mathbf{\Gamma}_u, \mathbf{\Gamma}_a$
Initialize all $w_{i,j} \leftarrow 0, 1 \leq i \leq d_u, 1 \leq j \leq d_a$;
Initialize all $\gamma_i^u, \gamma_j^a \leftarrow 1, 1 \leq i \leq n_u, 1 \leq j \leq n_a$;
Initialize $\gamma_0 \leftarrow c_0$;
$k \leftarrow \lfloor \frac{|T|}{m} \rfloor$;
Separate training set into $k$ chunks of $m$ examples;
repeat
    for $i = 1; i <= k; i++$ do
        foreach $(z_{x,y}, \mathbf{U}_x, \mathbf{A}_y) \in$ i-th chunk do
            Execute 1.1 to 1.2 in Algorithm 1;
        end
    end
    foreach $w_{i,j} \in \mathbf{W}$ do
        $w_{i,j} \leftarrow w_{i,j} - sgn(w_{i,j}) * t$;
        if $|w_{i,j}| < t$ then //filter small weight
            $w_{i,j} \leftarrow 0$;
        end
    end
until convergence;
```

5. EXPERIMENTS

We conduct experiments on Facebook's large-scale online advertising system. First, we do preliminary analyses on the baseline relevance feature, and show its potential and problems in clickthrough prediction. Then we use our model to learn variant relevance features, compare them and verify our assumptions with a series of performance studies and decompositional analyses. Finally, we test the performance of our feature when working together with other features and show its improvement to offline clickthrough prediction. Due to business confidentiality, we report only relevant performance when showing experimental results.

5.1 Experiment Setup

Data Sets. We test our approaches on a sample of ads clickthrough data recorded on `facebook.com`. We sample historical click data anonymously from a small fraction of the 500 million monthly active users, and use the ad impressions seen by these users for training. Users are connected with about 50 different kinds of objects that are treated as the concept sources; ads have 4 concept sources in our setup. We use only non-private sources from users for concept extraction—*e.g.*, chat logs or user messages are never looked at. On average a user is connected to 80 community pages, groups and events, and create 90 pieces of content each month [1]. For offline analysis, we use anonymized click data from the user sample, gathered over one week. Since the positive examples (clicks) have low rates and can lead to biased estimates [15], we reduce the imbalance with a downsampling strategy similar to previous work [7].

Concept Extraction Model. For our experiments, we use the publicly available topic hierarchy in DMOZ Open Directory Project [2] as labeled concept classes, and use the documents linked to each class to train a concept classifier. The classifier is named ODP. The second concept extraction model is a pseudo concept model that uses demography group instead of classified concepts for user concept space (while using the ODP concepts for the ads). We use it to show the generality of our learning algorithm. ODP has tens of thousands concept classes in total, and DEMO contains hundreds of demography groups, classified by country, age, gender, education, political stand and relationship status. For ODP, the concept vectors are high-dimensional though sparse, leading to millions of learning parameters to be learnt in the transformation matrix $\mathbf{W}$.

Relevance Model. We have presented algorithms for two sorts of learnt parameters: the weights $\mathbf{\Gamma}_a, \mathbf{\Gamma}_u$ on each heterogeneous source, and the transformation matrix $\mathbf{W}$. As a demonstration, we compare with our relevance feature against various baselines:

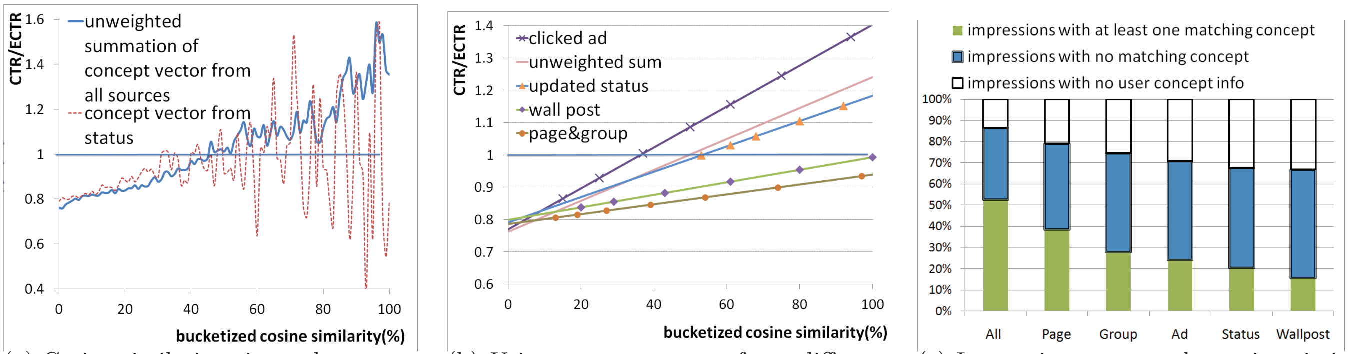
- f: Our relevance feature using weighted sources + learnt pseudo bilinear form similarity.
- *cos*: Baseline model with unweighted sources + a plain cosine similarity feature, or in other words, a uniform diagonal transformation matrix.
- *wcos*: Weighted sources + cosine similarity .
- $f_{unweighted}$: Unweighted sources + learnt similarity.
- f_{sparse}: Weighted sources + transformation matrix learnt with sparse coding.

For each baseline, the parameters (source weights or transformation matrix) were relearnt by adding the relevant constraint to the optimization problem.

5.2 Preliminary study of Baseline Features

We first explore the value provided by the baseline feature *cos* in CTR prediction. Using labeled ads impression data, we compute the baseline feature and bucketize the impressions according to feature values. The i-th bucket ($0 \leq i < 100$) contains all the impressions with feature value $cos \in [\frac{i}{100}, \frac{i+1}{100})$. For all the impressions in each bucket, we plot the average CTR divided by average estimated CTR(ECTR), which is given by an existent ranking system without cognitive relevance feature. A perfect system should always have CTR/ECTR=1. The farther CTR/ECTR is from 1, the more the estimate should be adjusted. We see from Figure 3(a) that if we use unweighted sources and exact concept match, the feature value has a positive correlation with CTR/ECTR. It suggests that tuning up and down the CTR estimation for high and low relevance feature respectively can improve the accuracy.

We repeat this for the feature *cos* obtained by using user concept vectors from each single source. Figure 3(b) shows the regression trend of CTR/ECTR with cosine similarity. The larger slope indicates a stronger prediction power for that source. The prediction power varies a lot from source to source, and unweighted combination of all sources is even worse than using a certain source in this rough estimate. Figure 3(c) shows the coverage of the cosine similarity for all user-ad impressions during a week. As we can see from Figure 3(c), for each source no more than 40% of the total impressions have at least one matching concept; even we use all sources, only half of total impressions are "covered" in this sense. The low coverage of nonzero similarity values provides another motivation to improve the similarity metric beyond the analysis in Section 4.1, while the observation that every



(a) Cosine similarity pivots the correction to the estimated CTR; unweighted summation causes noisy pivoting effect

(b) Using concept vector from different sources to do relevance matching has different pivoting power

(c) Impression coverage by cosine similarity values, using concept vectors from different sources

Figure 3: Potential CTR prediction improvement by cosine similarity feature. Sources vary in the power of relevance matching and the coverage of the ad impressions where they can suggest an exact concept match.

Table 2: logloss(*)/logloss(cos) for different relevance feature, where logloss is the deviation w.r.t. randomness. Higher values indicate that the model does a better job in predicting clickthroughs from input concept data.

Iteration	logloss deviation wrt randomness (ratio to cos)			
	$wcos$	$f_{unweighted}$	f	f_{sparse}
100K	2.6	4.4	**5.6**	**5.6**
200K	2.8	5.7	**6.9**	**6.9**
500K	2.7	6.7	**8.1**	8.0
1M	2.9	7.7	**9.5**	9.4
2M	2.9	8.8	**10.9**	10.8
test	3.8	11.2	**14.3**	13.7

source has unequal power and coverage suggests we should assign different weights to different sources.

5.3 Analysis of Our Relevance Model

This section presents the results of our learning algorithm for source weights and transformation matrix. First we compare the error reduction of by variant relevance models on the same test set. We then analyze the learnt model parameters to interpret the benefits of our method, and present interesting discoveries.

5.3.1 Learning Results

We applied the algorithms described in Section 4.2 to clickthrough data collected anonymously over 4 days from 1 million sampled users, and test on the next 3 days. In our implementation, we used a fixed learning rate, and chose other algorithm parameters using validation error.

Table 2 shows the percentage logloss reduction obtained by our algorithm (left) and various baselines, where the logloss reduction is measured compared to random guessing with the empirical click rate c_0, and displayed as a ratio of the logloss reduction achieved by cos. The baseline feature *cos* is only slightly better than random guessing, due to its low coverage and the lack of ability to match distinct but related concept classes. The cosine similarity with weighted sources ($wcos$) has as 4 times large logloss improvement as unweighted sources, but has the same coverage

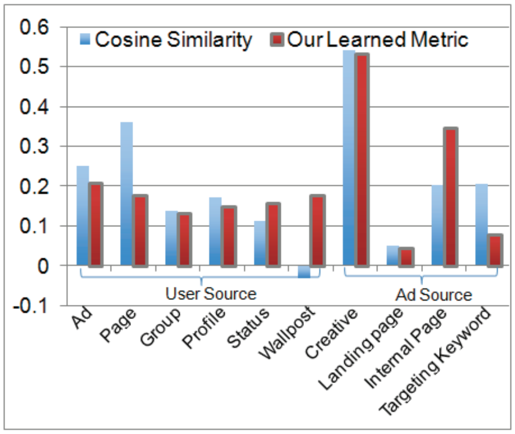


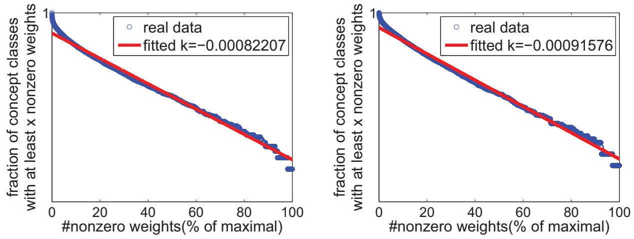
Figure 4: Weights of concept source for two settings: cosine similarity and normalized bilinear form similarity.

problem. The normalized bilinear form similarity can solve that problem and the logloss reduction is 10-fold larger than *cos*. With our jointly learnt model, the logloss reduction can be further improved, gaining a 14-fold of *cos*. Finally, the learning results with sparse coding demonstrate that we can reduce the number of nonzero parameters we need to store by 73-80%, thus reducing the overheads of feature computation, with only a small effect (within 5%, relatively) on the observed logloss performance. That speeds up the online feature computation by 53 times.

5.3.2 Source Weights

The learned weights of concept sources for cosine similarity and the learnt bilinear form similarity are shown in Figure 4. Some sources are entitled nearly zero weight due to extremely low coverage, and we omit it and only show the major sources, 6 for users (the ads they clicked before, the pages they are linked to, the groups they join, their profile interests, status updates and wallposts) and 4 for ads (descriptions created by the advertiser, linked pages outside Facebook, Facebook pages and keywords for targeting).

It is not surprising that "clicked ads" rank high as a user source, because it is the target we want to predict. We also observe that page concepts get higher source weight than



(a) #nonzero weights/row (b) #nonzero weights/col

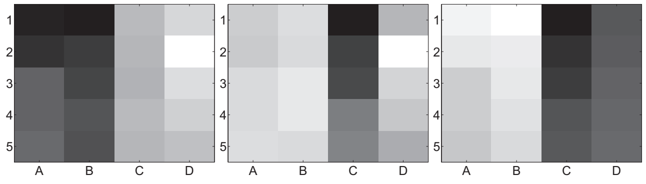
Figure 5: Nonzero elements distribution in the transformation matrix, y axis in log scale

group concepts, which might imply that the groups are not necessarily formed by interests, while pages better reflect the collective interests of their users. For ad concepts, the dominant source is the ad creative text, the visible part to the user. However, the other sources add extra information, beyond what is achievable just using the creative text. For ads that link to a known onsite object (e.g., ads about a music artist), the concepts for that object strongly indicate good ad concepts. Further, important information can be gleaned by looking at the landing page for the ad, as well as the targeting criteria provided by the advertisers. Taken together, these results validate our hypothesis that important semantic information can be obtained by looking at many heterogeneous sources of information, even though the sources have varying levels of noise.

An interesting observation is that if we change the concept matching function, the importance of sources varies qualitatively. For example, if we only allow exact concept matching with cosine similarity, some textual content such as wallposts are noisy and are assigned low source weights, indicating that exact concept match is not appropriate for them. However, the improved similarity metric can handle this problem by changing the association weight between the implicit interests and related classes of ads. When we allow matching across concept classes with our transformation matrix, we are able to use all of these sources successfully.

5.3.3 Transformation Matrix

To better understand the output of the learning algorithm, we now examine the learned transformation matrix $\mathbf{W}$. The i-th row of this matrix corresponds to the i-th user concept, and the j-th column to the j-th ad concept. The weight on (i, j) is used to match the i-th user concept to the j-th ad concept. For ODP, the learned matrix is sparse and contains 3% (which is already millions) nonzero parameter values, with extremely skewed density in various rows and columns of the matrix. The number of nonzero weights in each row and each column both satisfy power-law distributions (Figure 5). On average, each row and column contains several hundred nonzero values—which means that a user or ad concept can potentially match several hundred concepts.



(a) $f_{unweighted}$ (b) f (c) f-$f_{unweighted}$

ac	**A** - Trading Card Games	**B** - Rabbits Personal Pages	**C** - Shrek Series	**D** - Entertainment Weblogs
uc**1**	Music	Music	Arts	Music/ Styles
uc**2**	Christianity	Society	Animation Movies	Society
uc**3**	Movies	Video Games	Humor	Humor
uc**4**	Music/ Styles	Online Communities	Cartoons	Movies
uc**5**	Parenting Weblogs	Music/ Styles	Movies	Animation Movies

Figure 6: The different transformation matrices for unweighted sources and weighted sources. 4 groups of user-ad concept association are shown by gray scale, black=1, white=-1. In each group 5 associate user concepts for a fixed ad concept are chosen.

Figure 6 visualizes three 5-by-4 matrices to show the typical difference between the transformation matrix learned with unweighted sources and weighted sources for ODP. The 4 columns divide the association into 4 groups. In each group we fix the ad concept class, and select 5 user concept classes whose association weight with that ad concept disagree in the two cases. Along each column, say column A, we have 5 cells now corresponding to 5 user-ad concept pair from (uc_{A1}, ac_A) to (uc_{A5}, ac_A). Each cell in the grid is dyed according to the association weight. Dark color means positive association and light means negative. See the table below the figures for a map. In general, the association weight given by f looks more meaningful because it amplifies the concepts from cleaner source and shrinks the concepts from noisier source. For example, the interests in music have nothing much to do with ads of trading card games or personal pages with rabbits ($A1, B1, A4, B5$); the interests in animation movies and humors apparently signal a tendency to click ads about Shrek Series ($C2, C3$). This suggests that transformation matrix can be better learned if the importance of concept sources are better distinguished.

From the analyses from above two sections, we conclude is that source weighting and non-exact matching mutually enhance each other and an integrate combination of them lead to higher quality of prediction. Thus our algorithm for learning them jointly, instead of separately picking the source weights and the matrix $\mathbf{W}$, is expected to be crucial.

5.3.4 Case Study

The values in the learned transformation matrix capture various statistical associations between the user and ad concept classes. Table 3 shows some extreme cases from which we can interpret the user-ad relevance captured by the matrix. Eight categories of such cases are studied. We present four of them due to space limitation. For each category, we give examples from different concept extraction models. The first category we show is about a single ad concept class which is related to many user concept classes. Turn-based strategic video games are associate with more than 70% user interests, and Christianity denominations relate to almost every demography group. Ads with such concepts have diversely targeted users. The other 3 categories we show in Table 3 are about concept pairs which have very strong or

Table 3: Interpreting the transformation matrix

Characteristics	*CE*	*Example:#related concepts(%)*
Ad concepts associated with most user concepts	ODP	Turn Based Video Games: 71
	DEMO	Christianity Denominations: 97
Characteristics	*CE*	*Example: association weight*
Strong positive associations	ODP	Browser Based Casinos →Facebook Platform:0.59
	DEMO	Female→Dance:0.74, Male→Guitar:0.72
Strong negative associations	ODP	Birthdays→ Online Games:-0.72, Arts→ Society:-0.52
	DEMO	Female→Sexuality Politics:-0.76, US → Mah Jongg:-0.46
Identical but weak associations	ODP	US Colleges:-0.20, Poker Personal Pages:-0.18,

Table 4: Relative variance reduction achieved when a feature is added to test feature set. High values for important features.

Test feature set	$wcos$	f	f_{DEMO}
F_0 = user-ad relevance feature	0.12	**0.34**	0.008
$F_1 = F_0$ + user click bias	0.03	**0.06**	0.002

very weak associations. Strong associations could be either positive or negative, *e.g.*, females favor dance while males like guitar; US people do not like Mahjongg, which is a popular gambling game in Asia. Some user concepts have weak association weight with the identical ad concept. That actually reflects a mismatch when we use cosine similarity—though the concept of colleges in US can be found in many user pages and profiles, it does not imply that these users will like ads about colleges in US. On the contrary, such ads will attract people who have not been in US colleges.

There are also several hard-to-interpret results, such as the negative association between birthdays and online games. Such results may either reflect unexpected input statistics, or might be influenced by the imperfectness of concept extraction, or bias in the training set (as the training samples are generated by an optimized ranking system).

5.4 Cognitive Relevance for Targeting

To compare our relevance feature with previously studied or commonly used features, such as keyword-based, clustering-based and friendship-based relevance features, we train new CTR prediction models using two different configurations of features for test purpose. In one configuration (called F_0), we include all existing features in an ad ranking system that capture inferred user-ad relevance (and are thus meaningful even for new ads). This checks the additional relevance value brought by our new feature. In the second configuration (called F_1), we additionally include features that capture the bias of different users to click on ads and the past clicking behavior of users. This helps check that the observed gains from our relevance features cannot be explained solely by capturing a user's past clicks with hand-crafted features. In each case, we experiment with training a model with and without our new relevance feature. We evaluate the importance of each feature by variance reduction achieved [12], and the improvement of logloss reduction when each of them is added to the test feature sets.

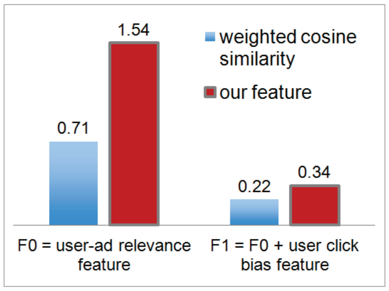


Figure 7: Improvement of logloss reduction (%) by cognitive relevance over test feature sets F_0 and F_1

Table 4 shows the approximate importance for each feature given by training and testing a CTR prediction model over one week of click data (60% for training and 40% for test) for a fraction of users. Among all the inferred relevance features between a user and an ad, the feature generated by our learned model ranks highest and is twice as important as the second highest. When we include user-click-bias features in our comparison, we find our feature still outperforms all individual non-historical features and most historical features as well (not shown in the table). Weighted cosine similarity has importance 1/3 and 1/2 of f in the two cases and also ranks high, but demography relevance is of one order lower importance weight. As a consequence, our feature brings a significant improvement in logloss reduction, while demography feature has improvement lower than 0.1% (and thus not plotted in Figure 7). The relevance feature is especially useful for the challenge of ranking newly created ads, for which historical click data is not available—the feature thus forms an apriori idea of the quality of an ad for each user. When an ad has been shown for an extended period of time, it is possible to collect actual clickthrough information for that ad from a large sample of users, and it is likely that the apriori relevance feature will be less significant.

5.4.1 Online experiment results

As a final evaluation, we test the performance improvement achievable by adding the new feature to the ad ranking system that uses the existing optimized features. We randomly select about 5M users and partition them into two equal large groups. The first group G_0 are targeted by the ranking system trained with all previous features, but without our new relevance features (F_2); the other group G_1 are targeted by a retrained ranking system with the exact concept match feature ($wcos$) and our final relevance feature (f) added. As expected, our relevance features help improve the underlying ads metrics in these online experiments. For example, for the impressions with our relevance feature f larger than a relevance threshold, we observe that the CTR is improved by at least 30% across the board.

For post-experiment analysis, we placed the new feature values into 100 percentile buckets, such that each bucket contains an equal number of impressions. We compare the CTR improvement of G_1 over G_0 by showing observed CTR in G_1 for different values of the feature bucket as a multiplier of the overall CTR of G_0 . For a strongly predictive feature, we expect the feature value to be very indicative of the actually observed CTR in the system. From Figure 8 we find that the two features we added have comparable power when measuring relevance for those impressions with

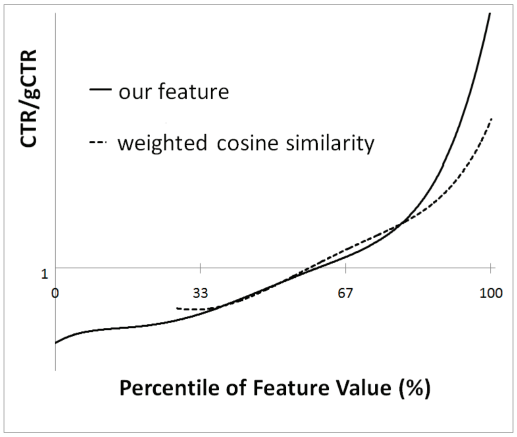


Figure 8: The CTR boost observed (as a ratio of average CTR in the control group G_0) when we add our new relevance feature f and the baseline feature $wcos$ to test feature set. The plot shows that high values of our relevance feature are indicative of possible clicks, and low values are indicative of poor relevance or low clicks. The absolute values of the y-axis labels are hidden for confidentiality.

"moderate" relevance (from 40% to 70%), but our feature is much stronger in finding highly relevant user-ad pairs and boosting CTR for them. Unlike the exact match similarity, which is zero for a large percentage of impressions (more than 30% as seen from Figure 3(c) and 8, when no exact concept matches are found), our feature is also good at ranking them and finding ad impressions that are likely to have low-relevance, and get lower clicks than expected.

6. CONCLUSION AND FUTURE WORK

In this study we aim for relevance based targeting in large scale social network sites. We infer user interests and ad concepts from heterogeneous sources and links, and develop user-ad relevance feature based on weighted matching between any pair of concept classes. We learn both the link weights and matching matrix from click data, and find they mutually enhance each other. Experimental results reveal insights into user preferences in online social networks and suggest valuable potential for more relevant targeting.

We have made some simplifying assumptions in this study that can be relaxed in future work. For example, we do not model temporal aspects of user-generated content, and do not personalize weights on various sources. The method can be applied on top of richer concept extraction models, which should help provide a clearer signal of concept associations. Finally, for more complex models, parallel learning algorithms and better data infrastructure can be studied to facilitate the offline learning and online ranking.

7. ACKNOWLEDGEMENTS

Chi Wang was supported in part by the U.S. National Science Foundation under grant IIS-09-05215, and by the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053 (NS-CTA). The authors thank Andrei Alexandrescu, Wenjie Fu, Hong Ge, John Hegeman, Nuwan Senaratna, Tao Xu and members of the Facebook ads engineering team for their insights and their work on building parts of the system described in this paper, and Zhenhui Li and Quanquan Gu for valuable discussions.

8. REFERENCES

[1] *Facebook Statistics*, `http://www.facebook.com/press/info.php?statistics`.

[2] *The Open Directory Project (ODP)*, `http://www.dmoz.org`.

[3] E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *SIGIR '06*.

[4] H. Bao and E. Y. Chang. Adheat: an influence-based diffusion model for propagating hints to match ads. In *WWW '10*.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[6] D. Boyd and N. B. Ellison. Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication*, 13(1), 2007.

[7] D. Chakrabarti, D. Agarwal, and V. Josifovski. Contextual advertising by combining relevance with click feedback. In *WWW '08*.

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

[9] M. Ciaramita, V. Murdock, and V. Plachouras. Online learning from click data for sponsored search. In *WWW '08*.

[10] M. Daoud, L. Tamine, M. Boughanem, and B. Chebaro. Learning implicit user interests using ontology and search history for personalization. In *WISE'07: Proceedings of the 2007 international conference on Web information systems engineering*.

[11] D. Fensel, F. van Harmelen, I. Horrocks, D. L. McGuinness, and P. F. Patel-Schneider. Oil: An ontology infrastructure for the semantic web. *IEEE Intelligent Systems*, 16, 2001.

[12] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29, 2001.

[13] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35, 1992.

[14] Q. Gu, J. Zhou, and C. H. Q. Ding. Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs. In *SDM*, 2010.

[15] G. King and L. Zeng. Logistic regression in rare events data. *Political Analysis*, 9:137–163, 2001.

[16] I. Konstas, V. Stathopoulos, and J. M. Jose. On social networks and collaborative recommendation. In *SIGIR '09*.

[17] A. Y. Ng. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*.

[18] B. Piwowarski and H. Zaragoza. Predictive user click models based on click-through history. In *CIKM '07*.

[19] F. Provost, B. Dalessandro, R. Hook, X. Zhang, and A. Murray. Audience selection for on-line brand advertising: privacy-friendly social network targeting. In *KDD '09*.

[20] B. Ribeiro-Neto, M. Cristo, P. B. Golgher, and E. Silva de Moura. Impedance coupling in content-targeted advertising. In *SIGIR '05*.

[21] A. Shepitsen, J. Gemmell, B. Mobasher, and R. Burke. Personalized recommendation in social tagging systems using hierarchical clustering. In *RecSys '08*.

[22] D. Vallet, M. Fernández, P. Castells, P. Mylonas, and Y. Avrithis. Personalized information retrieval in context. In *3rd International Workshop on Modeling and Retrieval of Context*, 2006.

[23] 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, 2002.

[24] Z. Wen and C.-Y. Lin. On the quality of inferring interests from social neighbors. In *KDD '10*.

[25] R. W. White, P. Bailey, and L. Chen. Predicting user interests from contextual information. In *SIGIR '09*.