Computational Advertising: Leveraging User Interaction & Contextual Factors for Improved Ad Retrieval & Ranking

Kushal Dave
Language Technologies Research Centre
International Institute of Information Technology
Hyderabad, India
kushal.dave@research.iiit.ac.in

Supervised By: Vasudeva Varma
Language Technologies Research Centre
International Institute of Information Technology
Hyderabad, India
vv@iiit.ac.in

ABSTRACT

Computational advertising, popularly known as Online advertising or Web advertising, refers to finding the most relevant ads matching a particular context on the web. It is a scientific sub-discipline at the intersection of information retrieval, statistical modeling, machine learning, optimization, large scale search and text analysis. The core problem attacked in computational advertising (*CA*) is of the match making between the ads and the context. Based on the context, *CA* can be broadly compartmentalized into following three areas: Sponsored search, Contextual advertising and Social advertising. Sponsored search refers to the placement of ads on search results page. Contextual advertising deals with matching advertisements to the third party web pages. We refer the placements of ads on a social networking page, leveraging user's social contacts as social advertising.

My research work aims at leveraging various user interactions, ad and advertiser related information and contextual information for these three areas of advertising. The research work focuses on the identification of various factors that contribute in retrieving and ranking the most relevant set of ads that match best with the context. Specifically, information associated with the user, publisher and advertiser is leveraged for this purpose.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning—*Induction*; H.1.1 [**Information Systems**]: Models and Principles

General Terms

Algorithm, Experimentation, Performance, Economics

Keywords

Advertising, Sponsored Search, Direct Marketing, Soc-ial Networks, Viral Marketing, Contextual Advertising

1. INTRODUCTION & MOTIVATION

Organizations like Ya-hoo!, Facebook, Google and Microsoft devote substantial share of their research in optimizing and improving the ad retrieval and ranking process. In addition, the recent surge of interest in the research communities (both in industry and academia) is a testimonial for the huge promise this discipline has on offer. All these factors demand further exploration of the domain that can result in better experience for users and increased revenue for the advertisers.

Computational advertising, a term recently coined in, is about using various computational methodologies to do contextually targeted advertising. The central problem addressed in *CA* is: first retrieving a set of ads that best matches the context and then ranking these ads. Based on the context, *CA* can be divided into following three areas: 1) Sponsored search 2) Contextual advertising 3) Social advertising. *Sponsored search (SS)* refers to the placement of ads on search results page. Here the context is the query issued by the user and the problem is to retrieve top relevant ads that semantically match the query. 2) *Contextual advertising (ConAd)* deals with the placement of ads on third-party web pages. It is similar to *SS*, with the ads being matched to the complete web page text as opposed to a query. 3) *Social advertising (SA)* is the newest of all the siblings and it refers to placement of ads on the personalized home pages of the users of the social network. It involves using viral marketing strategies to serve ads to users who can in turn convince their friends to visit the ads. Placing contextually related ads has a two-fold advantage. First, user's immediate interest in the topic can be exploited, this also increases the chance of users exploring the ads. Second, it leads to a better user experience. On the other hand, randomly placing ads may lead to user dissatisfaction.

This process of retrieving most relevant and contextually matching ads require borrowing methodologies from information retrieval, machine learning, statistical modeling and microeconomics. Specifically, one needs information retrieval techniques to efficiently retrieve ads in real time, semantic matching of ads with the text etc. Machine learning techniques are used for tasks such as learn the ranking of ads and prediction of parameters. Statistical modeling is used to model user, ad behavior and for recommender systems, while microeconomics takes care of tasks such as ad budgeting, bid economics.

My research work focuses on exploring various user-level, ad-level and context level factors that can retrieve the most relevant and best matched ads for a context. The first step is identifying such factors for all three areas - *SS*, *ConAd* and *SA*. Next step is to empirically analyze the contribution of all these factors towards better ad retrieval and ranking.

The work presented here is organized as follows: In each of the Sections 2, 3 and 4, I first explain the related work in

Copyright is held by the International World Wide Web Conference Committee (IW3C2). Distribution of these papers is limited to classroom use, and personal use by others.
WWW 2011, March 28–April 1, 2011, Hyderabad, India.
ACM 978-1-4503-0637-9/11/03.

that area, followed by the research work conducted till now and then the results and its analysis. I conclude in Section 5 by presenting the future directions of the work.

2. SPONSORED SEARCH

Sponsored search refers to the placement of a ranked list of ads on the search page contextually matching the user query. Usually, this is done in a two-step fashion. First, an initial set of k ads contextually matching the query are fetched from the database. Next, these k-ads are ranked according to some relevance measure. One widely used measure for assessing ads relevance is its click-through rate (CTR). CTR can be calculated from historical click-logs. For new ads, we do not have any historical information. In order to rank these ads in conjunction with the established ads it is very important to accurately predict the CTR for new ads.

2.1 Related Work

Fain and Regelson [6] predict the click-through rate for rare terms by using hierarchical clustering of bid terms. The CTR of a term is predicted based on the CTR of the terms in the same cluster and the parent clusters. Richardson et al. [7] predict the CTR values based on the information derived from the ad text. Other noticeable attempts at predicting these CTR values include Shaparenko et al.[8], who investigate query-ad word pair indicator features and find that they perform better than traditional vector-space and language modeling features.

2.2 Work Till Now

Most of the earlier work predicted the CTR as a function of the ad text only. An ad's CTR can vary across different queries. To account for these differences, it is also important to consider the query associated with the ad in predicting the CTR value. Thus, CTR of an ad is a function of both the ad and the query. We make use of the ad-query click graph and the advertisers information to find ads similar to the new ad and show that CTR for new ads can be learned with a good accuracy from similar ads fetched from these sources. A detailed explanation of this work is available in my paper [1].

We performed our experiments on 12 day's click logs from Yahoo! search engine's US market. The dataset had about 1.5M unique query-ad pairs from the click through logs with 1,97,080 unique queries and 9,43,431 unique ads. From the data, we have a number of query-ad pairs, with the observed CTR for each pair. For each ad, our goal is to predict its CTR as if we did not know about it. We use Gradient boosted decision trees (GBDT) as a regression model for the prediction of CTR. We use the following three set of features.

Features from Query-ad click graph: These features are based on the semantic relation of a query and an ad with other similar queries and ads. The idea here is to learn the CTR of a query-ad pair from semantically similar queries and ads. Edges are weighted using click frequency-inverse query frequency (CF-IQF) model: $cfiqf(q_i, a_j) = c_{ij} * iqf(a_j)$. The transition probability from a query to an ad, $P(a_j|q_i) = cfiqf(q_i, a_j)/cfiqf(q_i)$. A query q is represented as a vector of transition probability from q to all the ads in the graph. Each query is represented as $q_i = (P(a_1|q_i), P(a_2|q_i), , P(a_n|q_i))$. The similarity between two queries q_i and q_j is the cosine similarity between the two query vectors. Using query similarity, CTR is estimated as:

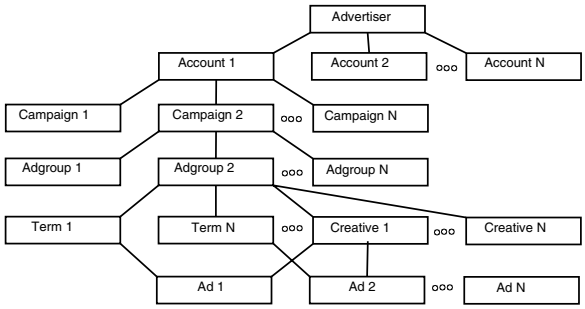

Figure 1: Ad hierarchy managed by ad engines

$$QCTR(q_i, a_k) = \frac{\sum_k CTR(q_j) * Sim(q_i, q_j)}{\sum_k Sim(q_i, q_j)}$$

The similarity between ads is also calculated in a similar fashion, with each ad being represented by the transition probability from an ad to a query $P(q_k|a_i)$ and similarity between two ads is referred as $Sim(a_i, a_j)$. The query and ad similarity features are called *Sim-Q* & *Sim-A*.

Features from Ad Hierarchy: Advertisements on an ad engine are typically maintained in some kind of a hierarchy. One such hierarchy is shown in Figure 1. Each level in the hierarchy does a granular classification of ads. Finally, at the leaf a term and creative make an ad.

We aggregate ads at each level viz. Term, Creative, Adgroup, Campaign and Account, compute the average within each group and use them as features in our model. We refer to these features as *AdH* features. Detailed description of all the features is also available in the report [3].

Features from Query-ad lexical match: In an attempt to capture how relevant an ad is to the query, we compute the lexical overlap between the query and the ad text. We compute various text matching features such as cosine similarity, word overlap, character overlap, and string edit distance for each combination of unigrams and bi-grams. We refer to this category of features as *QADL*.

As shown in Table 1, Sim-Q & Sim-A give good improvements and when combined (*Sim-QA*) gives an improvement of 21.11%. In the AdH category, Campaign (Camp) gave the best result and when Sim-QA was clubbed with Camp the improvement over baseline reached 26.67%. Finally, lexical feature did not yield much improvement alone, but (*Sim-QA+Camp+QADL*) gives the best performance with a good 28.61% improvement over the baseline. Baseline is the average CTR in the training set. All these improvements are statistically significant at 99% significance level.

To summarize, we have proposed an approach to predict the CTR for new/rare ads based on the similarity with other ads/queries. The similarity of ads is derived from sources like query-ad click graph and advertisement hierarchy maintained by the ad engine. The model gives a good prediction on the CTR of new ads. In addition, I am also working on using LDA topic modeling for retrieving relevant set of ads for a query. The idea is to represent both queries and ads as topical vectors and find ads similar to a query topic vector.

Feature	RMSE (∗ e3)	KL Diver-gence (∗ e1)	% Improvement	Feature	RMSE (∗ e3)	KL Diver-gence (∗ e1)	% Improvement
Baseline	7.20	1.72	-	**Campaign**	**5.67**	**1.32**	**21.25%**
Sim-Q	5.86	1.42	18.61%	Account	5.94	1.39	17.50 %
Sim-A	6.31	1.53	12.36%	AdH	6.20	1.46	13.9%
Sim-QA	**5.68**	**1.38**	**21.11%**	**SimQA+Camp**	**5.28**	**1.24**	**26.67%**
Term	6.24	1.45	13.34%	QADL	6.50	1.56	9.72%
Creative	6.51	1.50	9.6%	**SimQA+Camp**			
Adgroup	5.87	1.35	18.48%	**+QADL**	**5.14**	**1.21**	**28.61%**

Table 1: Improvement for various features (p-value ≤ 0.01)

3. CONTEXTUAL ADVERTISING

Contextual Advertising (*ConAd*) refers to the placement of ads that are contextually related to the web page content. It is similar to *SS*, with the web page content equivalent to a query in *SS*. In order to find ads contextually related to a web page, ads are matched against the web page text. The web page content usually contains noise, hence typically advertisement keywords are extracted from the web page text and then the ads are matched against the extracted keywords. Thus, keyword extraction is one of the fundamental task for most of the contextual advertising systems. Yih et al. [9] used a variety of features such as TF-IDF, query logs, occurrence in title of the web page and other commonly used features for keyword extraction from web pages. They found that presence of a candidate in query log is a useful feature.

3.1 Work Till Now

Most of the keyword extraction systems first tokenize the text into all possible words or phrases (N-Grams). This process of tokenization is called *chunking* and the token word/phrase is called *candidate*. A classifier then classifies each candidate, based on some features, into a keyword or a non-keyword.

We propose a new chunking approach that employs part-of-speech (POS) knowledge for selecting candidates from the text. The POS chunking approach takes advantage of the fact that most of the advertising keywords are proper phrases. Hence, only these potential advertising keywords can be considered as candidates. These proper phrases usually follow certain POS patterns. These POS patterns can be learned from the manually annotated data. One way to learn from the data is to pick the most frequent patterns from the data. In our case, some of the frequent patterns found were $(Noun)^+$, $(Adjective)^+(Noun)^+$, $[CD](Noun)^+$, $(Noun)^+[CD]$ etc. where + indicates one or more occurrence of that POS tag. CD is the cardinal tag. For our POS approach, the average reduction in the number of candidates is found to be 86% in comparison with N-Gram approach. (N-Gram approach considers all possible N-Grams). For a detailed explanation please refer our paper [2].

We pose the problem of extraction of keywords as of classification of candidates into keyword/non-keyword. we use naïve Bayes classifier which uses following three category of features. (1)***Linguistic Features (LING)***: This set of features represent the linguistic context of a candidate by considering POS tag of the candidate and its surrounding words. Specifically, we use POS tag of the candidate, POS tag of two words before and after the candidate as features. (2)***IR based Features (IR)***: We use term frequency (TF) and $log(TF)$ of the candidate as a feature. (3)***Other Features (OF)***: This category contains binary features like occurrence in query log, occurrence in title, link etc. For query log feature, we use AOL and Yahoo! webscope query log. AOL query log consisted of $\approx 20M$ web queries, while Yahoo! log contains the 1000 most frequent queries issued to the Yahoo! search engine.

The dataset we used for our experiments comprised web pages in English from various categories such as blog, product review pages, forums and news articles. We concentrated on web pages where contextual advertising seemed desirable, that is, we only took pages which had advertisements placed by some ad-network. In all, we had 810 web pages in which ad keywords were manually identified. It had 379 blogs, 129 news pages, 100 forum pages, 180 product review pages and 20 others. Based on the chunking method & the features used, we designed 4 systems as shown in Table 2. For each system, it shows the chunking method and the set of features used. Table 3 shows performance of these NGram and POS systems with (+ling) and without linguistic (-ling) features. The precision values are averaged over five folds. Both the proposed version of POS systems (+ling & -ling) outperform their counterpart NGram systems. Considering the configurations described in Table 2, the performance improvements can be attributed to the proposed chunking method and the linguistic features. Results obtained are found significant at 99% significance level.

System	Chunking	Features
POS+ling	POS	LING, IR, OF
POS-ling	POS	IR, OF
NGram+ling	NGram	LING, IR, OF
NGram-ling	NGram	IR, OF

Table 2: Description of the systems

System	P@1	P@3	P@5
POS+ling	**0.47**	**0.48**	**0.46**
NGram+ling	0.38	0.40	0.40
POS-ling	0.45	0.47	0.45
NGram-ling	0.35	0.38	0.38

Table 3: Performance of N-Gram, POS systems

4. SOCIAL ADVERTISING

Social advertising (*SA*) goes one step further beyond *SS* and *ConAd*. In SA, the rich neighborhood information can be leveraged to achieve better *reach* for the ads. The idea is to target certain special individuals in the network who can influence others to perform a similar action (say, buying the advertised product). This is commonly known as *viral marketing*. Targeting these influentials usually leads to a vast spread of the information across the network. Once these influencing set of individuals I are targeted, they become *contagious* and start influencing their neighbors. If the

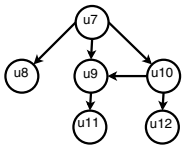

Figure 2: Sample action graph for an action a

neighbor's are influenced, they become contagious and can in turn propagate the information to their neighbors and so on. This usually leads to a chain of propagation of the action across all the neighbors of the users u in I. Hence to get a better reach for the information/ad it is important to identify such influencing individuals in a network.

4.1 Related Work

The work most relevant to our proposed method is by Richardson and Domingos [4] & Kempe et al. [5]. Richardson et al. [4] were the first to tackle the problem of identifying k most influential nodes in a network, such that targeting them yields maximum reach. They show that user's network value and her intrinsic value can be combined to make optimized marketing decisions. Kempe et al. [5] show that the problem of identifying the top influentials from a network is NP-complete. They propose a greedy hill climbing strategy for picking top-k influentials which outperform certain network level heuristics like degree-centrality.

4.2 Work Till Now

We hypothesize that a user's influencing capability also depends on the information/ad (here onwards, I refer to information/ad as *action*) to be spread across the network and the user's social neighborhood's influencing capabilities. The problem of identification of influencers should also take the particular action into account while identifying the influencers. To confirm this hypothesis, we first empirically analyze how factors like user's and their neighborhood's influencing ability and action popularity affect a cascade by a user for an action. Next, we build a regression model that predicts the average cascade triggered for an action by a user. This work is currently in progress.

An action is said to be propagated from user $u1$ to $u2$, if $u2$ performs the same action as $u1$, within a certain timeframe after $u1$ performed that action. Based on this definition, an action graph is built from the social graph. In the action graph, a directed edge from $u1$ to $u2$, indicates that an action was propagated was $u1$ to $u2$. Figure 2 depicts a sample action graph.

We first quantify the average number of cascade a user u triggers for an action a as its reach. Let $reach^a(u)$ be the reach of a user u for an action a. It can be recursively defined as follows:

$$reach^a(u) = \begin{cases} \sum_{u_i \in \vec{P}^a(u)} 1 + \frac{1}{2} * \sum_{u_i \in \vec{P}^a(u)} reach^a(u_i) & \text{if } \vec{P}^a(u) \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

$\vec{P}^a(u)$ is a set of all the immediate neighbors u_i of u in social graph such that there was an action propagation from u to u_i for action a. A user gets a credit of 1 for an action propagation to its immediate neighbor. The significance of $(\frac{1}{2})$ times the reach of the descendants of a user u in the action graph can be understood as assigning decaying credit as we move farther from the node u in the propagation chain.

Active Factors	Avg. log(reach) for p_u^{inf}	Avg. log(reach) for p_u^{prone}
Only u	2.00	2.69
u+h1	2.74	3.27
u+h1+h2	3.23	3.31
u+h1+h2+h3	3.47	3.78

Table 4: Reach increases as neighborhood influence and prone probabilities cross the threshold

For example, in Fig. 2 the overall reach of user $u_7 = (1 + 1+0.5(1)+1+0.5(1)) = 4$. Identifying the small set of users who can elicit greater reach for an action can be formulated as the problem of predicting the reach for each user and a particular action. The problem to be tackled is: "Given a social graph and past action events, accurately predict the reach of each user for a particular action ($reach^a(u)$)".

To apply our framework, we use Flickr social network data for the experiments. The social graph contains O(1M) users and O(100M) edges. We consider an action event as a user 'joining a group'. First, we analyze the co-relation of $reach^a(u)$) with various user-level, action-level and user's neighborhood-level factors. In the user and action level factors, we consider the user's influencing probability (P_u^{inf}) and the actions influencing probability (P_a^{inf}). These probabilities represent the factor's capability of influencing. While in the neighborhood level factors, we consider the average of the influencing capabilities of all the neighbors at hop k from the user u ($P_{u:hopk}^{inf}$). In addition, we also consider, how prone is a user and user's neighborhood at hop k to getting influenced for any action (P_u^{prone}, $P_{u:hopk}^{prone}$) . We first analyzed the impact of these factors on $reach^a(u)$ independently and found that as the user, action and the neighborhood become influencing the $reach^a(u)$ value increases.

We hypothesize that, if the influence probability is more than a certain threshold, we deem that factor as active and say that it is contagious. For example, if P_u^{inf} is less than 0.5, we consider the user to be inactive. On the other hand, if P_u^{inf} exceeds 0.5, the user is considered to be contagious (active). For all the user, action and neighborhood influence probabilities (P_u^{inf}, P_a^{inf}, $P_{u:hopk}^{inf}$), the threshold is set to 0.5. Similarly, if the prone probability is less than a certain threshold, we say that the factor is not susceptible to peer influence (Inactive). In our case, if the user (P_u^{prone}) or the neighborhood prone probability ($P_{u:hopk}^{prone}$) is less than 0.03, we say that it is inactive, otherwise we consider the user/neighborhood to be active.

Table 4 confirms this hypothesis, where row 1 corresponds to only the user being active (both in terms of p_u^{inf} & p_u^{prone}). Row 2 corresponds to the event that only the user and hop1 neighbors are active (while hop2 and hop3 neighbors are Inactive). As shown, the neighborhood becomes active as the reach for user u increases (column 1). Prone probabilities (column 2) at each hop show a similar trend, the reach increases as the neighborhood at each hop becomes susceptible to peer influence. We also check if the action level factors combined with the user and social neighborhood factors have any impact on the reach value. As before, we fix on a threshold (0.5) and if p_a^{inf} is greater than the threshold, we say that the action is contagious (active). Table 5 analyses the impact on reach as the user, action and the neighbors become active. Row 1 gives the average reach value when only the user is active ($p_u^{inf} >= 0.5$ and $p_a^{inf} < 0.5$). In row 2,

only the action is active, while in row 3 both user and action are active and so on. This shows that when all the factors are active in conjunction, it increases the $reach^a(u)$ value further.

Active Factors	Avg. log(reach) for p_u^{inf}
Only u	2.21
Only a	3.60
u + a	4.04
u + a + all hops	4.24

Table 5: Reach increases as user, action and hop become active in conjunction

Based on the above analysis, we build a regression model to predict $reach^a(u)$ for action a and reach u. Apart from the influence probabilities, we consider various user and action specific features like number of friends of u, number of people who have done action a in the past, number of friends in the user's neighborhood etc. The train and test set have non-overlapping user-action pairs, that is, a user-action pair either appears in a train set or a test set. We use GBDT as a regression model for predicting the reach values. We use two baselines: *Baseline1* is the average reach of the user u across all the actions. *Baseline2* is the average reach of the action a for all the users. Table 6 shows the performance of both the baselines and the machine learned model. Improvement 1 & 2 show improvements over baseline 1 & 2 respectively. All the results presented are statistically significant at 99% significance level.

System	MSE	KL Divergence (x 1e-1)	Improvement1	Improvement2
Baseline1	5.20	2.54	-	-
Baseline2	3.11	1.53	40.19 %	-
Model	**2.67**	**1.27**	**48.65%**	**14.15%**

Table 6: Performance of baselines, learned model

As shown, the prediction model gives a very good improvement of 48.65% over baseline 1 and an improvement of 14.15% over baseline 2. Besides, the model does better than baseline 2 by 8.46% in terms of improvements over baseline 1. Also, baseline 2 performs substantially better as compared to baseline 1. On further investigation of the data, it was found that the average coefficient of variation for the actions was 0.29, while for users the average was 0.47. Hence lesser variance in the reach values amongst the action results in baseline 2 performing better than baseline 1.

To summarize, we analyzed various factors contributing to the cascades triggered by a user. The analysis yields several interesting insights - There is a direct association between the reach and various user, action and neighborhood factors. The analysis confirms that more contagious these factors are bigger is the reach for that user and the action. We empirically showed that the action, user and the neighborhood features combined together give a good prediction of the average reach of a user in the graph.

5. FUTURE WORK

Most of the work towards ranking of ads has concentrated only on the query and the ad. It will be interesting to see if any user specific information (such as - topics on which the user has queried in the past) can play a role in the ranking of ads. Also, some users click more frequently on the ads compared to others. Hence, estimating CTR of an ad can be viewed as a user-query-ad triadic function.

Recommending a set of ads to a user can be formulated as a problem of *collaborative filtering (CF)*. The major hurdle with such a collaborative filtering application is that information for very few user-ad click events is available, as most of the user's would have clicked only on a handful of ads. In such applications, usually around 1% of all the possible click information is available. It is a challenging task to estimate click events for the missing 99% of entries based on the sparse information available. Techniques such as matrix factorization or certain log-linear models can come to the rescue. Also, combining this CF score with a ranking score derived from various features pertaining to the query, ads and users identified earlier is a challenging task.

One of the primary problems faced in sponsored search and contextual advertising is of the vocabulary mismatch between query/web page text and ads. There are certain semantic approaches proposed earlier to bridge this gap. It will be interesting to assess query/web page and ad categorization for retrieving topically relevant ads. The idea is to match the categories/topics of query/web page with ads to retrieve categorically/topically similar ads.

For social advertising, the process of user triggering cascades is a good contender to be modeled by some appropriate graphical models such as hidden markov model (HMM) or condition random fields (CRF). These graphical models can be leveraged to make future cascade predictions. Exploring these graphical model is another line of research.

For experiments described in Section 4, an action was simulated by considering 'joining a group' as a measure of cascade. It will be worthwhile to see if the findings are similar for an advertisement related action, such as adopting the advertised product or clicking on an ad.

Based on the analysis of various factors for SS, ConAd, SA, it will be good to come up with a general framework for these advertising setups. It will also be interesting to see if certain factors are prevalent in all the three types of advertising.

6. REFERENCES

[1] K. S. Dave and V. Varma. Learning the click-through rate for rare/new ads from similar ads. SIGIR '10, Geneva, Switzerland, 2010.

[2] K. S. Dave and V. Varma. Pattern based keyword extraction for contextual advertising. In *CIKM*, Toronto, Canada, 2010.

[3] K. S. Dave and V. Varma. Predicting the click-through rate for rare/new ads. *Technical report IIIT/TR/2010/15*, 2010.

[4] P. Domingos and M. Richardson. Mining the network value of customers. In *KDD '01*, pages 57–66, San Francisco, California, 2001. ACM.

[5] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *KDD '03*, Washington, D.C., 2003. ACM.

[6] M. Regelson and D. C. Fain. Predicting click-through rate using keyword clusters. In *Electronic Commerce (EC)*. ACM, 2006.

[7] M. Richardson, E. Dominowska, and R. Ragno. Predicting clicks: estimating the click-through rate for new ads. WWW '07, Banff, Alberta, Canada, 2007.

[8] B. Shaparenko, O. Çetin, and R. Iyer. Data-driven text features for sponsored search click prediction. ADKDD '09, Paris, France, 2009.

[9] W.-t. Yih, J. Goodman, and V. R. Carvalho. Finding advertising keywords on web pages. WWW '06, Edinburgh, Scotland, 2006.