Ranking of New Sponsored Online Ads Using Semantically Related Historical Ads

Hamed S. Neshat[1] and Mohamed Hefeeda[1,2]

[1]Qatar Computing Research Institute
Doha, Qatar
[2]School of Computing Science
Simon Fraser University
Surrey, BC, Canada

ABSTRACT

Online advertising in search engines is a wide and growing market. In this market, revenue of search engines depends on the number of user clicks received on displayed ads. Thus, in order to increase the revenue, search engines try to select top ads and rank them based on the expected number of clicks they will receive. For ads that were in the system for a period of time, the expected number of clicks could be estimated based on historical data. For new ads, or those ads without enough historical data, search engines need to predict the potential of these ads in attracting user clicks. We purpose a method to estimate the potential of new ads in attracting user's clicks. We use semantic and feature based similarity algorithms to predict the click through rate of new ads using historical similar ads. Our trace-based evaluations show that the proposed method outperforms other approaches in the literature in terms of the accuracy of prediction. In addition, the proposed method is less computationally expensive than previous methods and it can run in real time.

1. INTRODUCTION

Internet advertising is the main source of income for search engines. For example, Google reported $6,475 million revenue from advertisement in 2009 which is 8% more than the previous year [10]. This emphasizes the fact that online advertising is a multi-billion dollar industry with expected high growth rate in the coming years.

Roughly speaking, online advertising works in two steps [12, 11]:

1. Finding relevant ads: Advertisers associate keywords with their ads. When a web user submits a query on a search engine, all ads with keywords related to the search query are put into an auction [11].
2. Selecting top ads for inserting on the result pages: Ads are positioned on the returned result pages based on their ranks. The ad with the highest Ad Rank appears in the first position, and so on down the page. The rank of an ad is given by:

$$AdRank = CPC * QualityScore, \tag{1}$$

where CPC is the cost per click, which is provided by the advertiser and shows how much the advertiser is willing to pay for each click on the ad. The Quality Score depends on various factors including most importantly the click through rate (CTR) of the ad. If an ad is displayed n times and received m clicks, search engines associate m/n as its click through rate [12]. The click through rate is an important metric as it directly impacts the revenue of search engines. We note that the Quality Score usually considers other factors such as the history of the advertiser's accounts.

According to Eq. (1), historical information is needed to compute the quality scores and in turn the ranks of ads. Since search engines continuously receive new ads that have not been displayed before, search engines need a method to estimate the quality scores of these new ads. Accurate estimation of the quality scores of new ads is critical, since it determines which ads (from the old and and new ones) are displayed to users.

In this paper, we propose a new method to predict the quality scores of new ads. The proposed method finds existing ads that are semantically similar to the new ads. It then estimates the quality scores of the new ads based on their corresponding similar ads. The proposed method is unlike previous methods in the literature, e.g., [1, 5, 15], which tend to use general features such as number of words in ad and type of URL of other existing ads. Using a set of general features might not work for different contexts. For example, although good description in a car financial company related ad is important, it is less important for an ad about a new perfume, where users usually look for the brand names in the title or URL [2].

We have implemented our method and compared it against the most recent methods in the literature using large-scale traces collected from a major search engine (Google). Our results show that the proposed method produces more ac-

This work is partially supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada.
Hamed S. Neshat conducted part of this work during his master's degree at Simon Fraser University.

curate predictions than the previous methods. Moreover, unlike other methods which require offline pre-processing to create complex prediction models, our approach requires light weight computation and can run in real time.

2. RELATED WORK

Ashkan et al. [1] estimate the click through rate based on the total number of ads on the page, rank of ads, and the intent underlying the query for which the ad is displayed. Richardson et al. [16] build a prediction model for click through rate based on logistic regression using historical data and existing ads. They find 81 different features for ads and divide them into five categories which are Appearance, Attention Capture, Reputation, Landing Page Quality and Relevance. They use extracted features of new ads as the model inputs to predict click through rate of new ads. Dembczynski et al. [7] propose an approach for predicting click through rates for new ads based on decision rules. They extract features from existing ads and create decision rules which vary the value of predicted click through rate based on existence of those features in the new ads.

Choi et al. [5] do not evaluate ads based on user clicks. Rather, they propose a technique for finding ad quality. They explore different techniques for extracting document summary to select useful regions of landing pages with and without using ad context. By this way, the quality of each ad depends on its landing page.

Regelson and Fain [15] estimate the click through rate for terms not for whole ads. They find the global click through rate for infrequent keywords as well as keywords having high click through rates in specific periods. They use historical data and term clusters to find relationship between historical terms in the system and new terms.

Border et al. [4] work on a semantic approach to contextual advertising. They propose a method to match advertisements to web pages that rely on a semantic match as a major component of the relevance score.

Dave et al. [6] present a model that inherits the click information of rare/new ads from other semantically related ads. The semantic features in their work are derived from the query ad click-through graphs and advertisers account information. However, they do not directly use ad contents for finding semantically related ads.

3. PROPOSED METHOD

This section presents the proposed click through rate prediction method. We start with an overview, followed by more details.

3.1 Overview

The problem we address in this paper is estimating the click through rates of new ads. click through rates are used in estimating the ranks of ads according to Eq. (1). Ranks of ads determine which ads are displayed in web pages returned by search engines and the location of these ads within the page.

To solve the click through rate estimation problem of new ads, we utilize information from existing ads in a novel way. The proposed approach, which is summarized in Figure 1, consists of two main parts: (i) Finding Similar Ads and (ii) Predicting Click Through Rate.

As shown in Figure 1, users submit queries to the search engine. The search engine finds web pages that match the submitted queries. Before returning the web page to users, the search engine may insert one or more ads in this page. The search engine has a dataset of ads which it has already computed their expected click through rates based on historical data. The search engine has also a set of new ads which have not yet accumulated enough history to enable reliable computation of their click through rates. The search engine uses our proposed method to estimate the click through rate for these new ads. In order to find similar ads for a given new ad, two ranked lists of ads are generated. The first list is based on ads' terms semantic and the second list is based on ads' terms features. Ads in these two lists are ranked according to their distances from the new ad. Since the number of terms in an ad is much fewer than in a web document, we cannot use existing algorithms for finding document similarity (Such as Latent Semantic Indexing or Precision and Recall based algorithms [17]). Instead, we use our own vector based ranking (summarized in Section 3.2).

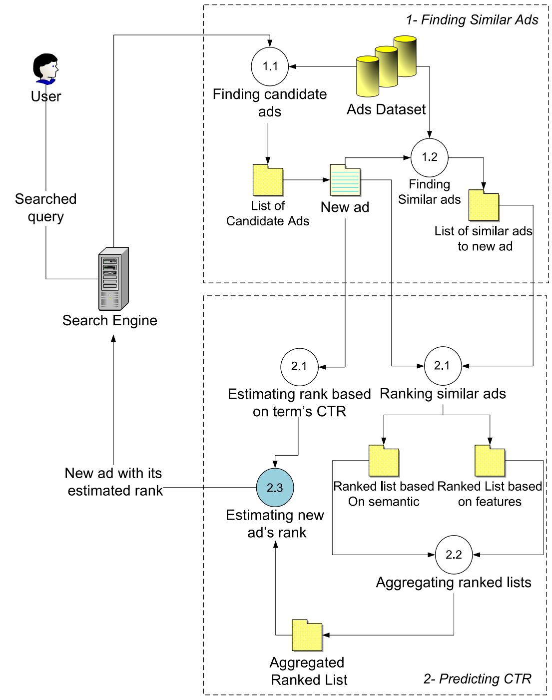

Figure 1: Overview of the proposed method for predicting click through rate of new ads.

In the prediction click through rate step, the two lists are aggregated into one ranked list and we compute weighted average of click through rate from known click through rate of ads: $Score_1$. Meanwhile, the weighted average of new ad's terms click through rate is computed in this step (we name it $Score_2$). Finally, we combine $Score_1$ and $Score_2$ to estimate new ad click through rate, which could be combined with CPC to compute new ad rank.

The focus of the proposed method is on new ads. Once ads accumulate enough historical data, these data are used to estimate the click through rates.

3.2 Finding Similar Ads

We propose a new method to find similar ads based on conceptual and general features of existing terms in ads. Our method has two parts: 1- finding all semantically related historical ads, and 2- ranking found ads based on their similarity with the new ad.

To find semantically related ads, we first retrieve all ads with the same keywords as the new ad. Then we look for those ads that have semantically related keywords in their keywords list. We use WordNet [18] to find semantically related words. WordNet finds related words based on a hierarchical cluster set. In this hierarchical cluster set, words placed in lower clusters are more semantically related together. For example, the outputs of WordNet for "soil" are dirt, land, ground, territory. Our method works with different levels and numbers of clusters in the WordNet. We analyze the impact of clusters on the performance of the proposed method. We rank ads based on the similarity of their terms together. Ranking of similar ads is done as follows:

First, we use our own dataset of a huge collection of ads (information about data collection is available in Section 4.1) and extract all used terms in ads. Then, we cluster all terms used in ads in our dataset. We form two sets of clusters: The first set of clusters is created based on term features, and we use K-means clustering to create them. The second set of clusters is created based on term meaning. We categorize all terms in clusters with hierarchical pattern, from 17 main categories. These categories come form Google keyword tool; other structures can be used in our algorithm as well. We use WordNet to put terms into clusters. Given a term, WordNet is able to find semantically related terms to it. We use our main categories to form 17 initial clusters with one word, and then WordNet puts each term into the most semantically related cluster.

For each ad, we form two vectors related to two cluster sets. Each vector has a number of entries equal to the number of clusters in a set; so we have two pairs of vectors and clusters. For each pair of cluster set and ad vector, if the ad has a term from n-th cluster, we increase its related vector entry by 1. Assume during the searching and selecting ad process, we find a candidate ad which is new. For predicting its click through rate, we look at all other similar ads.

Given a list of ads, in order to measure their similarity with newly entered ad, we examine two distance metrics: X2 and normalized Euclidean distance. normalized Euclidean distance is a reduced version of the Mahalanobis distance [14]. We compare the results of both to select one of them as our final metric in Section 4.3. The X2 distance metric is given by [13]:

$$D_c(X,Y) = \frac{1}{2} * \sum_{n=1}^{V} \frac{(x_n - y_n)^2}{x_n + y_n}, \tag{2}$$

and the normalized Euclidean distance given by:

$$D(X,Y) = \sqrt{\sum_{n=1}^{N} \frac{(x_i - y_i)^2}{\sigma_i^2}}, \tag{3}$$

where σ_i is the standard deviation of the x_i over the sample set. We refer to the normalized Euclidean distance as NED.

Since we have two different vectors for each ad, we will have two ranked lists. In order to have one ordered list, we use a rank aggregation method to combine two generated ranked lists together. Several methods for aggregating ranked lists have been discussed in the literature [8]. Since our ranked lists are partial lists, we use Borda's method [3] which is designed for partial lists. Given ranked lists $t_1, ..., t_k$, for each candidate c in list t_i, Borda's method assigns a score $B_i(c) =$ the number of candidate ads ranked below c in t_i. The total score $B(c)$ is computed as $\sum_{i=1}^{k} B_i(c)$. The candidates are then sorted in decreasing order of the total score.

3.3 Predicting click through rate from Individual Ad Terms

Since there is no guarantee to find enough (or any) historical ads similar to a new given ad, working with similar ads might not work for all new ads. Sometimes, specific words in ads such as brand names, name of a services or goods, or name of a place can attract users. Google keyword tool lets us find the click through rate of a word in a period of time. In order to use effect of specific word's click through rate, we go through terms of ads and look at their click through rates. An ad has 3 parts: Title, Description and URL. We extract all terms from title and description and find their click through rate. But, since the range of terms click through rates is different from ranks in our models, we normalize them based on the distribution of ranks in our data model. Since title and description have different appearance styles, they have different visual effects on user, so we assign weights to words in different parts. We use this formula:

$$\frac{\alpha \sum_{t_t} CTR + \beta \sum_{t_d} CTR}{\alpha + \beta}, \tag{4}$$

where α and β are weights, the first sum goes over title terms and the second sum goes over description terms. In Section 4.3 we find optimal values for weights.

3.4 Click Through Rate Prediction

We estimate the click through rate of a new ad as follows:

$$\frac{w_1\ Score_a + w_2\ Score_t}{w_1 + w_2}, \tag{5}$$

where w_1 and w_2 are weights, $Score_a$ is the click through rate based on aggregation and $Score_t$ is the predicted click through rate from ad's terms click through rates. Information about finding optimal values for weights are presented in Section 4.3.

4. EVALUATION

In this section, we evaluate the proposed method, and compare its performance against the performance of three recent methods in the literature. We start in Section 4.1 by describing how we collected information about ads. Then, we describe how we choose the parameters used in our method. Then, we present our data model for estimating click through rate for ads, which we use in the evaluation. In Section 4.4, we describe our experimental methodology, and we present our results in Section 4.5. Finally in Section 4.6, we analyze the impact of different parameters on the performance of our method.

4.1 Data Collection

Information about ads and their characteristics is not usually public for researchers outside the search engine companies. Thus, we had to construct a dataset ourselves. We started this work with finding common keywords in ads by using the Google Keyword tool. We found about 800,000 common terms and phrases for ads keywords. Then, we searched for each keyword using Google and saved the first 3 pages of search results. In Google, usually ads are displayed in the 3 first pages of search results. Since we do not have top ads for all searched queries, we are not using top ads. Moreover, it is not clear from Google that how an ad can be placed on top of the page, so we skip them and only use side ads. According to Google policies, it is impossible to do a lot of search together with one IP without any gap between searches. For solving this problem, we used 10 different machines and put 10 second gap between each search. By this way, we could retrieve all pages that contain the collected ads keywords in one month. After retrieving all pages (about 2 million pages), we went through them to extract ads. Then, we extracted all terms in ads and used the Google keyword tool to find term features. We found 24 features for each term such as Global Monthly Searches, Estimated Daily Impressions, Estimated Ad Position, Estimated click through rate, Estimated Daily Clicks, Estimated Daily Cost, Estimated Avg. CPC and Term Frequency in dataset's ads.

We collected more than 4,000,000 ads of which 600,000 were unique. These ads had more than 300,000 unique terms. The overall size of our dataset is about 5 GB.

4.2 Data Model for Click Through Rate

We want to predict the click through rate of new ads in comparison with old ones, but do not have access to information about click through rate for each ad. We used ads ranks in the search result page as well as other available features to compute/simulate the click through rate for each ad. Please note that our proposed method is based on comparison between ads. Thus, accessing to actual values doesn't have important impact on accuracy of results. We only need a click through rate factor which is consistent among all retrieved ads. Moreover, ads rank on the page is a true representative for their quality and shows how much an ad is better than other listed ads. However, in order to increase consistency between ad ranks in the data set, we produce a click through rate model with different distribution. By that work, we consider various factors like popularity of searched query and visibility of ad on the page:

$$ClickThroughRate = v * n_r * \frac{1}{rank + 8 * p_n}, \tag{6}$$

where,

- p_n is the page number and $rank$ stands for ads' rank within the search result page.

- n_r is the number of results for searched query. Due to the fact that in many pages, we have less than 8 ads (sometimes we have just one ad on the page), placing in the first spot of the page doesn't always indicate high click through rate. We found that queries which have more search results, attract more sponsored links. If a query can get more results, it means it is more popular and there will be more ads want to appear on the page, so the selected ads probably have more click through rate. n_r shows popularity of searched query by user.

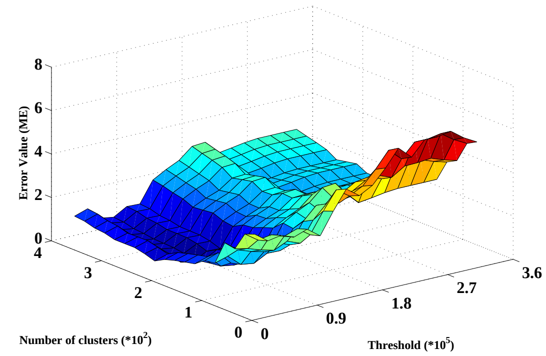

Figure 2: Mean error for different cluster numbers and threshold values with NED.

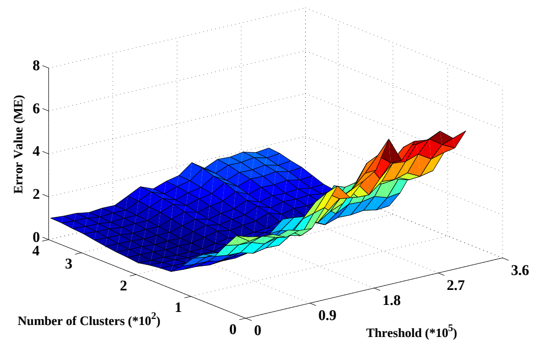

Figure 3: Mean error for different cluster numbers and threshold values with X2.

- v is visibility factor which is equal to eye tracking numbers. As Richardson et al. [16] said, whenever an ad is displayed on a page, it has a probability of being viewed by user. So the chance of an ad to receive a click depends on two factors: the probability that it is viewed and the probability that a user clicks on it. Thus,

$$p(click|ad, pos) = p(click|ad, pos, seen) \times p(seen|ad, pos). \tag{7}$$

A joint eye tracking study conducted by search marketing companies, Enquiro and Did-it, shows that the majority of eye tracking activities during a search happens in a triangle at the top of the search results page [9]. Moreover, they found even if an ad is placed in the best position of the page, it will be viewed by just 50% of users. In this research, they claimed that ads which are placed in rank 1 to 8 in Google can attract users view by these percentages respectively: 50%, 40%, 30%, 20%, 10%, 10%, 10%, 10%. In our data model, we include visibly of ads with popularity.

4.3 Parameters Optimization

We examine our algorithm with different numbers of clusters and thresholds to find best parameters' values for maximum accuracy. Threshold means the number of ads which

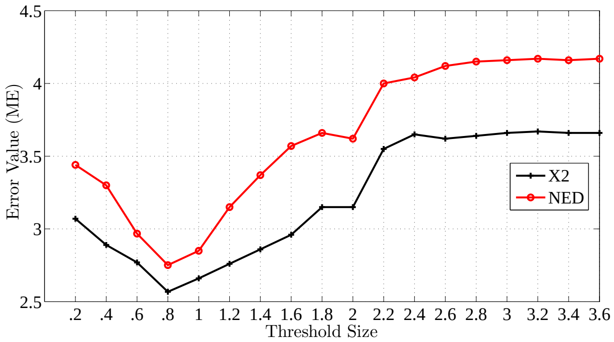

Figure 4: Mean error for different threshold values.

α	β	w_1	w_2
0.71	0.29	0.12	0.78

Table 1: Optimal values for Eq. (2) and (3).

will be used to find their similarity with our new ad. Figure 2 and Figure 3 show the results for different numbers of clusters and thresholds. For both distance metrics, the best results achieved when we have 220 clusters, and the threshold is set to $0.8 * 10^5$.

Figure 4 shows the results for X2 and NED with various numbers of threshold while there are 220 clusters (smaller numbers show better performance). The error value decreases as threshold increases, but when threshold goes further than $0.8 * 10^5$, neither X2 nor NED isn't improved. Greater threshold causes to use keywords which placed in higher clusters in hierarchical meaning cluster. Using keywords in higher clusters results in less relevant ads. Finally, Figure 4 shows that X2 overcomes NED and has less errors, so we use X2 as our similarity measurement metric.

Next, we compute the weights in Equations (2) and (3) that resulted in the best performance. We run regression to figure out effect of different values for the weights, and find optimal values for them in our data model. More specifically, we tune weights with examining values $0, 0.1, \ldots, 1$. We use half of ads from our ad data set as train data, and other half to test computed weights. Tables 1 shows optimal values for Eq. (2) and (3).

4.4 Methodology

We removed 100,000 ads which their click through rate and rank were known. Then we used our approach to re-estimate their click through rate. By comparing our estimated rank and their real rank we can find how accurate our approach is in predicting new ads' click through rate.

In order to compare results, we use three metrics: MSE (Mean Square Error), KLD (Kullback"-"Leibler divergence), and ME (Mean Error). MSE is given by:

$$MSE = \frac{\sum_{i=1}^{i=n} (x_i - t_i)^2}{n}, \tag{8}$$

where x_i is actual value and t_i is estimated value. MSE has been used by Richardson et al. [16], Ashkan et al. [1] and Debmbsczynski [7] as a performance metric.

KLD is defined as follows: Given two probability distributions P and Q, the Kullback-Leibler divergence between P and Q is

$$D_{\mathrm{KL}}(P\|Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}, \tag{9}$$

where P is set of estimated values and Q is set of actual values. Richardson et al. [16] and Ashkan et al. [1] both use KL- divergence between their model's predicted click through rate and the actual click through rate, which get 0 to the perfect model.

ME is given by:

$$E = \frac{|estimated\ CTR - actual\ CTR|}{actual\ CTR + 1}. \tag{10}$$

ME ensures that large errors on small rates are not neglected. In all mentioned metrics, smaller values show better performance.

4.5 Comparison with other methods

We compared the proposed method against the most recent approaches proposed in [1] [5] [15]. these methods are: i) a model based on query intent model [1], ii) a model based on logistic regression using statistics of existing ads (LR model) proposed by Richardson et al. [16], and iii) a model based on decision rules proposed by Debmbsczynski et al. [7]. The results from the work by Regleson and Fain [15] are not listed in the tables, because their results are based on term click through rate prediction not on ad click through rate prediction. In addition, we compare against a simple method as a base line for comparison. This method is denoted by Base Line (Average) in the tables, and it is the average of all ads' click through rate as estimated click through rate for new ad. We mentioned that we did try our best in implementing the previous methods and find their performance based on the available information.

The results for the data model are shown in Table 2 for our click through rate model described in Section 4.2. The results in tables show that the proposed method produces more accurate prediction for click through rate than all previous methods in all considered performance metrics. For example, Table 2 shows that our model results in 27%, 14%, and 47% reduction in MSE compared to the Query Intent Model [1], LR Model [16], and Decision rules Model [7] respectively. Across all results, our model achieved at least 14%, 9%, and 10% improvement in MSE, KLD, and ME respectively. The minimum improvement in a metric is computed as the difference between the results of our method and the best result produced by any other method.

4.6 Analysis of the Proposed Method

We compare performance of our models with different features on all three models. The results for our data model are summarized in Tables 3.

The results in the tables show how much each part can increase the accuracy of click through rate prediction. The improvement numbers are cumulative, which means they show improvement when each part is added to the model.

All tables have base line in the first line. In the base line

Prediction Model	*MSE*	*KLD*	*ME*
Baseline(average)	5.53	5.33	4.06
Our Model	3.26	3.17	2.56
Query Intent Model [1]	4.12	3.98	3.69
LR Model [16]	3.84	3.48	2.87
Decision rules Model [7]	4.01	3.79	3.88

Table 2: Comparison with previous methods in the literature.

	MSE	*KLD*	*ME*	*improvement*
Base Line (Average)	5.53	5.33	4.06	-
+Feature	4.11	3.94	3.32	18.13%
+Semantic	3.75	3.59	2.76	17.02%
+terms CTR	3.26	3.17	2.56	7.31%

Table 3: Impact of different parts of our method on the performance with our data model.

model, we look at all other ads in the system, and compute the average of their click through rate as estimated click through rate for new ads. Feature in tables means selecting similar ads based on similarity of their terms general features. In this step, instead of using all existing ads in the system, we select those ads which have similar features in their terms. More information about the ad selection procedure is discussed in Section 3.2. As it is expected, selecting some similar ads instead of using all ads in the system can at least improve accuracy by 18%. In the next step, we include conceptual similarity in ad selection. By this way, we use those existing ads which are in the same context with new ads (more details are available in Section 3.2). This feature increases accuracy of click through rate prediction by at least 17%. Finally, adding terms click through rate ensures us that we can predict click through rate even for those new ads which don't have enough similar ads. Section 3.3 provides more information about using term click through rate. Using terms click through rate improves prediction accuracy at least 7%.

5. CONCLUSION

We have proposed a novel method to address the problem of estimating the click through rate for new ads. The major difference between our work and other works in this area is that the proposed approach needs only light weight computation which allows us to use more recent historical data. Moreover, our method works with the semantic of ads contents and does not look only at general ads features. These two new features increase the accuracy of click through rate predictions produced by our method compared to previous methods. In particular, our trace-based evaluations show that the proposed method achieves at least 10% and up to 47% improvements in the accuracy compared to the most recent three methods in the literature.

6. REFERENCES

[1] A. Ashkan, C. L. A. Clarke, E. Agichtein, and Q. Guo. Estimating ad clickthrough rate through query intent analysis. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, WI-IAT '09, pages 222–229, 2009.

[2] G. S. Becker and K. M. Murphy. A simple theory of advertising as a good or bad. *The Quarterly Journal of Economics*, 108(4):941–964, 1993.

[3] J. C. Borda. *Memoire sur les elections au scrutin*. Histoire de l'Academie Royale des Sciences, 1781.

[4] A. Broder, M. Fontoura, V. Josifovski, and L. Riedel. A semantic approach to contextual advertising. In *Proceedings of the 30th annual international conference on Research and development in information retrieval*, SIGIR '07, pages 559–566, New York, NY, USA, 2007.

[5] Y. Choi, M. Fontoura, E. Gabrilovich, V. Josifovski, M. Mediano, and B. Pang. Using landing pages for sponsored search ad selection. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 251–260, 2010.

[6] K. S. Dave and V. Varma. Learning the click-through rate for rare/new ads from similar ads. In *Proceeding of the 33rd international conference on Research and development in information retrieval*, SIGIR '10, pages 897–898, 2010.

[7] K. Dembczynski, W. Kotlowski, and D. Weiss. Predicting ads click throughrate with decision rules. In *Proceedings of the 16th international conference on World Wide Web*, WWW '08, 2008.

[8] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th international conference on World Wide Web*, WWW '01, pages 613–622, 2001.

[9] Google Search's Golden Triangle. http://eyetools.com/articles/version-2-google-golden-triangle-eyetracking-search-report.

[10] Google investor relations. http://investor.google.com/financial/2009/tables.html.

[11] Growing your business with adwords. http://static.googleusercontent.com.

[12] How are ads ranked. http://adwords.google.com/.

[13] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1 –8, 2008.

[14] R. D. Maesschalck, D. Jouan-Rimbaud, and D. L. Massart. The mahalanobis distance. *Chemometrics and Intelligent Laboratory Systems*, 50(1):1 – 18, 2000.

[15] M. Regelson and D. Fain. Predicting click-through rate using keyword clusters. In *Proc. InSecond Workshop on Sponsored Search Auctions*, 2006.

[16] M. Richardson, E. Dominowska, and R. Ragno. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 521–530, 2007.

[17] I. Witten, E. Frank, and M. Hall. *Data Mining: Practical Machine Learning Tools and Techniques.* Morgan Kaufmann.

[18] WordNet a large lexical database of english. http://wordnet.princeton.edu/wordnet.