Retrieval Models for Audience Selection in Display Advertising

Sarah Tyler
Department of Computer Science
University of California at Santa Cruz
skt@soe.ucsc.edu

Sandeep Pandey, Evgeniy Gabrilovich,
Vanja Josifovski
Yahoo! Research
{spandey, gabr, vanjaj}@yahoo-inc.com

ABSTRACT

Web applications often rely on user profiles of observed user actions, such as queries issued, page views, etc. In *audience selection* for display advertising, the audience that is likely to be responsive to a given ad campaign is identified via such profiles. We formalize the audience selection problem as a ranked retrieval task over an index of known users. We focus on the common case of audience selection where a small *seed set* of users who have previously responded positively to the campaign is used to identify a broader target audience. The actions of the users in the seed set are aggregated to construct a query, the query is then executed against an index of other user profiles to retrieve the highest scoring profiles. validate our approach on a real-world dataset, demonstrating the trade-offs of different user and query models and find that our approach is particularly robust for small campaigns. The proposed user modeling framework is applicable to many other applications requiring user profiles such as content suggestion and personalization.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing

General Terms

Algorithms, Experimentation

Keywords

Retrieval models, audience selection, ad targeting

1. INTRODUCTION

> *"Half the money I spend on advertising is wasted;*
> *the trouble is I don't know which half."*
> – John Wanamaker (attributed).

Display advertising is the placement of graphical ads displayed on web pages alongside the original content. In display advertising, targeting is often performed by categorizing users into standard segments (e.g., auto, health) based on their online activities. However, such pre-defined segments might be too broad and poorly aligned with audience partitioning by the advertiser. This mismatch has led to the emergence of the current trend of *advertiser-centric* characterization of audiences. One popular approach for the advertisers is to characterize their desired audience by providing a small *seed set* of existing customers as examples for identifying additional prospective customers who are *similar* to those in the seed set, and are likely to "respond" to the advertiser's campaigns. This problem formulation is referred to as *audience selection.*

There are multiple ways to identify users similar to the seed set. One possible approach is K-Nearest Neighbors (KNN) modeling, however, KNN is less effective in very high dimensional spaces. Another possibility is to use Collaborative Filtering (CF) techniques, however, CF requires a sufficiently long history of user-item interactions. In our scenario we only have a few observations of users' prior interactions with ads, and many fewer prior purchases or conversions. Furthermore, unlike CF, where user interests are mostly stationary (e.g., a preference trend for action movies), in online advertising users' interests and needs change over time.

We formulate the audience selection problem as a *retrieval* problem. We build rich user profiles using their entire online experience and explore two retrieval approaches inspired by language modeling and vector space models. Given the seed set of positive examples (users) for the advertiser, we construct a *query*, in a way that is reminiscent of relevance feedback in IR. This query is then executed against the index of all other users (potential customers), and used to identify users for targeting the advertiser's campaign. This method allows for very efficient audience selection when searching over a large space of users.

Our setting differs from that of conventional ad-hoc IR in several interesting ways. First, natural language documents indexed by IR systems are usually fairly coherent around a single topic (or several related topics). On the other hand, users' online history are composed from numerous, often unrelated activities. Furthermore, the events we observe (e.g., Web searches, page views, ad clicks) are heterogeneous and contain varying amounts of information, hence reconciling them into a single user profile is an important research task. Additionally, users' interests evolve over time. The change in their interests can be either personal (e.g., planning a va-

CIKM 2011 Glasgow, UK

cation), or can reflect a global trend (e.g., a popular movie). Finally, in contrast to conventional IR, which judges the relevance of retrieved documents, we focus on maximizing the conversion rate, namely, the fraction of (retrieved) users who purchase the product or service being advertised.

2. METHODOLOGY

Given the seed set of users $U = \{s_1, s_2, ...s_3\}$ who have previously converted on a given campaign, our goal is to rank other users by their conversion potential. We approach this problem within the information retrieval paradigm, developing two alternative representations for indexing users. First, we consider *language modeling* (LM), a generative model where documents are generated by a multinomial distribution estimated through maximal likelihood over a document collection. We represent users as sequences of their observed events, hence we can talk about the probability of "generating" a user by estimating the probability of observing a given event sequence. Our second approach is based on the *vector space model* (VSM), which maps documents and queries into a space defined by their features.

2.1 User representation

We represent a user by the set of events $u = \{e_1, e_2 \dots e_n\}$ she has performed over a given history span. Each event is a triplet $e_i =< type_i, int_i, c_i >$, where $type_i$ denotes the event *type*, int_i is the time *interval*, and c_i is the event *content*.

An event *type* describes the nature of the event, such as a page view, ad click, etc. An event occurs at a specific time, and events are grouped in *time intervals* of varying length, ranging from 1 day to 1 month. Time intervals span non-overlapping continuous ranges of user activity, and together cover the whole period of user activity. Different intervals can be assigned different weights, effectively "decaying" the importance of older events. Finally, the event *content* contains the observed information about the event, usually in textual or numeric form, i.e., ids of ads viewed, the text of search queries issued, etc. We employ the bag of words method for modelling content strings, like queries, using both unigrams and bigrams, as well as nodes of a topical taxonomy that represent more general text categories.

An important research question is how to reconcile different events (ie queries the user issued two weeks ago, with ad clicks from the last minute) in a single model. Each of these cues can provide a valuable signal, yet heterogeneous nature and large volume of user activity would make very noisy representation if all combined in a single bin. To this end, we propose to represent users as a two-dimensional *array* of models, where we build a separate model (whether a language model or a vector space model) for each combination of a time interval and event type (see Figure 1). For example, if our time intervals are week-long, we have a separate model for the user's page views each week, as well as an additional models for her search queries issued each week. Each entry of this two-dimensional array contains a single model built over the concatenation of the content of all the events of the same type observed in the given interval. We build a single user model by learning weights for the different cells of the two-dimensional array.

2.2 Language modeling for audience selection

Language modeling (LM) has been successfully applied to representing documents and queries in textual information retrieval. LMs are often favored for their competitive performance, clean formalization and probabilistic interpretation.

User model.

Language models are generative models, where each user is assigned a probability of being generated. This probability is computed by estimating the probability of the sequence of events observed for this user. With a simplifying assumption of independence between events, we can transform this probability into a product of the probabilities of individual events. In this form, the model assumes a multinomial distribution over the space of events: $p(u) = p(e_1, e_2 \dots e_n) = p(e_1)p(e_2|e_1)p(e_3|e_2e_1)\dots p(e_n|e_1e_2\dots e_{n-1}) \sim \prod_{i\in 1\dots n} p(e_i)$.

A common strategy to alleviate the independence assumption in document retrieval models is to define composite features or n-grams. Similarly, here we can define composite events that are based on commonly performed actions, e.g., submitting a query followed by browsing the top search results. While we do not present experiments with this type of features, they can be easily facilitated by our model.

We further develop this model by considering the structure of events, namely, their time interval, type, and content: $p(u) = \prod_{i\in 1\dots n} p(e_i) \sim \prod_{i\in 1\dots n}\{p(int_i)\cdot p(type_i|int_i)\cdot p(c_i|int_i, type_i)\}$.

The probability of each event is estimated using three components: (1) the probability of observing an event in the given time interval ($p(int_i)$), (2) the probability of observing in that interval an event of the given type ($p(type_i|int_i)$), and (3) the probability of observing a specific event content given the interval and the event type ($p(c_i|int_i, type_i)$).

The model is simplified by assuming that the activity of the user proceeds with constant frequency. That is, $p(int_i)$ is proportional to the length of the interval, and $\sum_{int\in T} p(int) = 1$, where T is the entire history span. A further simplification can be made by assuming that the mix of event types does not depend on the interval: $p(type|int) \sim p(type)$, that is, the ratio between the number of page views, queries and other event types stays constant. Although this assumption does not necessarily always hold, our intervals are usually fairly long (several days), hence the assumption is not violated too much in practice.

To compute $p(c_i|int_i, type_i)$, we use standard modeling techniques to generate the content of the events. The probability of the textual content of the event is generated based on the probability of individual words: $p(c_i|int_i, type_i) = \prod_{w\in c_i} p(w|int_i, type_i)$, where w stands for a word from the event content and $\sum_w p(w|int_i, type_i) = 1$. These word probabilities are estimated using the maximum-likelihood estimator over the aggregated content of all events of a given type in a given time interval. We used Laplace smoothing.

Query construction.

The query for audience selection is very different from than in standard retrieval tasks. In the standard document retrieval, the query is essentially a short fragment of text. In the audience selection task the query is created from a set of seed users, and contains more events than an individual user might. To develop an adequate query representation, we build upon the work on language modeling with relevance feedback [16]. There, the query is composed by merging several documents (users in our case). Conceptually, we treat the seed set of users as relevance

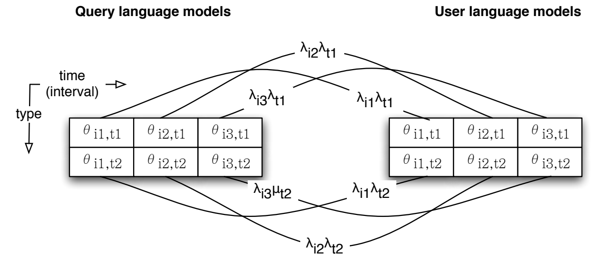

Figure 1: Component models of the user and query.

feedback from the advertiser. The query is thus calculated by finding a language model θ^q that is close to the models of the users in the seed set, but far from the other users (the collection background model) [16]: $p(w|\theta^q) = \exp(\frac{1}{1-\mu}\frac{1}{n}\sum_{i=1}^{n} log\, p(w|\theta^i) - \frac{1}{1-\mu} log\, p(w|C))$, where θ^i are the models for the users in the seed set, C is the background model composed of all other users, and μ is a parameter learned on a validation set. By subtracting the background model we are emphasizing those words/events in the query that discriminate positive users from the rest. Note that this approach is reminiscent of the Rocchio method [12].

Scoring functions.

We use two alternative LM scoring paradigms. First, as in standard language modeling, we score users based on the query likelihood — the probability of the query being generated using the user's language model: $p(q|u) \sim \prod_{w\in q} p(w|\theta^u)$, where θ^u is the language model of the user u. To compute θ^u, we separately analyze the content of events for each combination of interval and event type. More precisely, we manipulate an array of models, $\theta^u_{int,type}$. The user model is then computed as a combination of individual models for different intervals and events types (in a way that is reminiscent of the mixture language models [2, 17]): $p(w|\theta^u) = \sum_{int,type} \lambda_{int} \cdot \lambda_{type} \cdot p(w|\theta^u_{int,type})$. To make the model simpler, we opted to reduce the number of parameters and to learn one parameter for each time interval and event type (λ_{int} and λ_{type}, respectively) from the held-out validation set, instead of learning one parameter for each possible combination (i.e., $\lambda_{int,type}$). Figure 1 illustrates how the query and user models are compared when each is represented as a two-dimensional array of language models.

In the second scoring paradigm we use the model comparison approach. Drawing on the language modeling work for relevance feedback, we use KL-Divergence to compare the query and the user sub-models: $KL(\theta^u, \theta^q) = \sum_{w\in u\cup q} p(w|\theta^q)\cdot log\frac{p(w|\theta^q)}{p(w|\theta^u)}$. Note that log-likelihood is a special case of KL-divergence when the model does not use smoothing, but in our case these two represent different metrics. Here we also compare an array of models, as shown in Figure 1.

2.3 Vector space models for audience selection

Users are represented in our framework as a set of events we observed for them, where each event is a triplet of its time interval, type, and content. In essence, we first construct a meta-document by concatenating the content of all the events of the same type in the same interval. Then, we represent this meta-document as a TFIDF vector of its bag of words. Each user profile is then represented as a weighted combination of such vectors, where the weights are learned on a validation set: $\sum_{int,type} \varphi_{int}\cdot\varphi_{type}\cdot VSM_{int,type}$. Again, to make the model simpler, we reduce the number of parameters and learn one parameter for each time interval and event type (φ_{int} and φ_{type}, respectively). In order to compute $VSM_{int,type}$, we experimented with several TF and IDF formulations (Table 1), and used dot product as the distance metric to compare query and user vectors.

Query construction using the Rocchio algorithm.

We use the Rocchio approach to incorporate pusedo relevance feedback and compose the queries [12] for our vector space model. As a first approximation, we construct the query from two sets of users—the seed set of positive examples (U) who converted on this campaign, and a set of negative examples (V') sampled from the rest of the users (who did not convert on this campaign): $\vec{q} = \rho \cdot \frac{1}{|U|} \cdot \sum_{\vec{u}\in U} \vec{u} - \tau \cdot \frac{1}{|V'|} \cdot \sum_{\vec{u'}\in V' \subset \overline{U}} \vec{u'}$.

Since conversion rates are usually very low, the set of converted users is small, which leads to extreme data sparsity over a rich feature set. To this end, we also experimented with augmenting the seed set of users who converted with an additional set (U^{click}) of *pseudo-positive* users, who clicked on the ad but did not convert (a similar approach was used in [1]). The assumption behind this approach is that the users who clicked on the ad found it relevant at least to some degree, and might still convert later: $\vec{q} = \rho \cdot \frac{1}{|U|} \cdot \sum_{\vec{u}\in U} \vec{u} + \sigma \cdot \frac{1}{|U^{click}|} \cdot \sum_{\vec{u'}\in U^{click}} \vec{u'} - \tau \cdot \frac{1}{|V''|} \cdot \sum_{\vec{u''}\in V'' \subset \overline{U \bigcup U^{click}}} \vec{u''}$, where $V'' \subset \overline{U \bigcup U^{click}}$ is a subsample of negative examples.

In both formulations, values of the free parameters ρ, σ and τ are estimated on held-out validation data.

3. EXPERIMENTAL SETUP

We randomly selected 34 different display ad campaigns, which were registered on the Yahoo Advertiser Network. All these campaigns are performance-based, i.e., advertisers only pay for actual conversions. Of the 34 campaigns, some had only 30-50 conversions per week on average, while others receive many thousands of conversions every week.

User who either viewed, clicked, or converted on the ads from the three week period from 02/04/2010 to 02/24/2010 were identified. Users who converted for campaign c in the first two weeks make up our (positive) seed user set U for c (Section 2.3). The first 2 days of the final week make up the validation set for parameter tuning, and the last 5 days for the test set. In total there were more than 150K validation and 450K test instances across the 34 campaigns.

Each user's profile is constructed from four weeks online activity preceding the user's ad view. Note that while predicting a test instance, say on day t, we access user history up to day $t-1$. Hence, the method is not using any future information.

3.1 Evaluation Metric

One way to evaluate the ranked list of users produced by the different audience selection methods is to use the Receiver Operating Characteristic (ROC) curve. A ROC curve plots true positives versus false positives for different classification thresholds. The Area Under Curve (AUC) for a ROC curve is the probability that the audience selection method assigns a higher score to a random positive example than a random negative example (i.e., probability of concordance).

Term Frequency		Document Frequency		Normalization	
n (natural)	$n(d,t)$	n (no)	1	n (none)	1
r (ratio)	$\frac{n(d,t)}{\sum_{t' \in d} n(t',d)}$	t (idf)	$log(\frac{N}{df_t})$	c (cosine)	$\frac{1}{\sqrt{w_1^2+w_2^2+...+w_M^2}}$
l (log)	$1 + log(n(d,t))$	p (prob idf)	$max\{0, log(\frac{N-df_t}{df_t})\}$		
b (boolean)	$\begin{cases} 1, & \text{if n(d,t)} > 0 \\ 0, & \text{otherwise} \end{cases}$	r (PrTFIDF [7])	$\sqrt{\frac{N}{df_t}}$		

Table 1: TFIDF variants we experimented with.

So, a purely random selection method will have an area under the curve of exactly 0.5. An algorithm that achieves AUC of 0.6 can distinguish a positive user from a negative user with 60% accuracy.

4. RESULTS

We begin by evaluating Vector Space Modeling (VSM) and for Language Modeling (LM), and vary one parameter at a time to study its effect. In most cases, the effect of varying parameter values was similar for VSM and LM, suggesting that both approaches use the information in the data in a similar way; consequently, owing to lack of space we present parameter exploration results only for the VSM model.

While an AUC of 0.8 or more is common in many retrieval tasks, note that the audience selection task is inherently much more difficult than standard textual query-based retrieval. This is due in part because conversions are extremely rare and there are no true negative examples. Users who did not convert right away might still convert later. To put the sparsity into perspective, in the literature researchers have often found it to be difficult to predict clicks where the click-through rates are very small (e.g. 0.01 for certain applications) [3, 8, 11]. Conversions are usually two to three orders of magnitude rarer than clicks. Additionally, there are a multitude of activities in the user profiles that may not be relevant to the conversion.

4.1 Initial Results

For the initial vector space model we set TF to $\frac{n(d,t)}{\sum_{t' \in d} n(t',d)}$ (ratio), IDF to $max\{0, log(\frac{N-df_t}{df_t})\}$ (prob idf), without normalization. In this configuration all events were put into a single time interval and one event type (i.e., only one cell in the two-dimensional array of Figure 1). We study the effect of these parameters later. The query from the seed set was constructed using the Rocchio algorithm with weights $\rho = 1$, $\sigma = 0$, and $\tau = 1$ and dot product was used to score the users. Using this configuration, macro-averaged across all campaigns achieved an AUC of 0.65. Figure 2 shows the performance breakdown of the VSM model over individual campaigns. For some campaigns our approach does remarkably well (and achieves 0.90 AUC), while for some others the model does not perform any better than random guessing. In Section 4.6 we analyze these results and attempt to understand the variation in performance across campaigns.

For the language modeling approach, we use the query construction method as described in Section 2.2. Again, we use a single time interval and merge all the events as if they were of the same type (i.e., effectively consider one event type for this experiment). Overall, we found that the language modeling approach performed on par with the vector space one. In our experiments log-likelihood had a AOC of 0.66, better than both KL-divergence which had a ROC score of 0.63 and VSM score of 0.65.

4.2 The effect of relevance feedback

Recall from Section 2.3 that the query is constructed from the seed set of relevant users and a set of non-relevant users, with ρ and τ representing their weights in the query composition (we discuss the effect of using clicks to create pseudo-positive examples later). Figure 5 illustrates the effect of these two parameters.

Note that when $\tau = 0$, the query is being constructed using only the relevant users. Even though many retrieval systems use only positive feedback [10], in our experiments we found that negative examples can help significantly. For example, the performance goes up from 0.62 to 0.65 when the ratio $\frac{\tau}{\rho}$ is increased from 0 to 1. However, further increasing the value of $\frac{\tau}{\rho}$ causes the non-relevant users to overwhelm the relevant users, which in turn hurts the performance.

When compared on the per campaign basis, using non-relevant users leads to superior performance for most campaigns (over 80%). The increase in performance over the model from Section 2.3 is illustrated in Figure 3. Augmenting the seed set of converted users with users who clicked on ads but did not convert (which we called pseudo-positive examples) resulted in only a negligible improvement in AUC.

4.3 The effect of using time intervals

We represent the user history with a two-dimensional array of time intervals and event types. In section 2.3 we used one bucket of time, yet user histories are four weeks long in our dataset. We divide them into equi-width intervals where each interval is w days long. We vary the value of w from 1 to 28. We learn the importance weights λ_{int} for each time interval using the validation set. When width w is small, it allows us to capture finer patterns in the user history. On the other hand, by making w smaller, we reduce the amount of content in each cell of this two-dimensional array. This leads to data sparsity and worsens the individual cell-based scores computed for a user. For example, while a user and query may match reasonably well when their aggregated representation of a whole week, when compared on a daily basis they may look much less similar.

The best performance comes from $w = 28$, i.e., when the entire user history is put into one interval (Figure 6). This shows that the performance loss due to sparsity when w is decreased, outweighs the benefit of capturing finer user patterns. When $w < 28$, we found that the model learned higher weights for more recent intervals than the older ones. For example, with $w = 10$ we get the weight of 0.76 for the most recent interval, 0.19 for the second interval, and only 0.05 for the oldest one. This shows that recent activities are a better indicator of conversion likelihood, as expected.

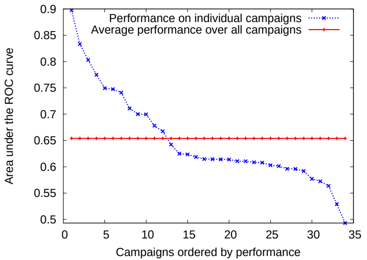

Figure 2: Performance of our VSM approach.

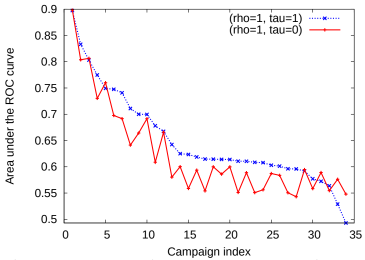

Figure 3: Using vs. not using non-relevant users.

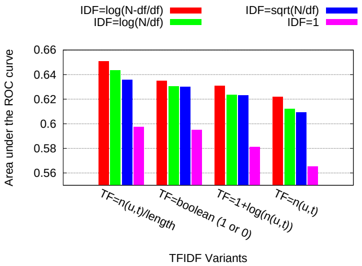

Figure 4: The effect of TFIDF variations.

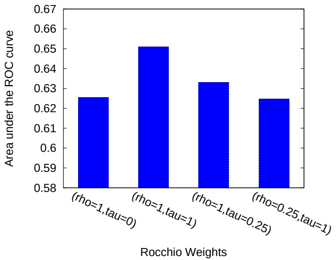

Figure 5: The effect of using non-relevant users.

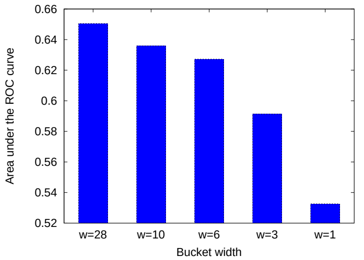

Figure 6: The effect of the time interval width.

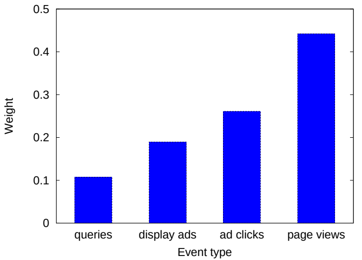

Figure 7: Weights learned for different event types.

4.4 The effect of event types

User history consists of different kinds of events (such as page views, ad clicks and the like). Instead of treating them collectively as done in the previous experiments, each event is divided into four event types: page views, search queries, organic and sponsored result clicks, and ad views. As described in Section 2.3, we construct a separate representation for each event type, score them, and then take a weighted linear combination of these scores (where the weights for the different event types, λ_{type}, are learned using the validation set). By doing so, we achieved a marginal improvement of 0.005 in AUC over the 0.65 AUC achieved by the baseline (Section 4.1). More interestingly, we found that some event types are much more indicative of user interests than others. In Figure 7, we show the weights learned for the different event types using the validation set. Browsed page views seem to help the most in predicting conversions. Even though one might expect page views to be less predictive than other event types such as queries and clicks, in our dataset they have higher density (i.e., much more events of this type) compared to other event types.

4.5 TFIDF Variants

TFIDF weighting of features has been central to the vector space models. The success of the TFIDF weighting scheme is due to its capturing the importance of features both within the document (TF) and in the entire collection (IDF).

In traditional IR, the TF function is based on the number of occurrences of the term within the document, namely, the raw term frequency which we will denote by $n_{u,t}$. In user modeling there are multiple ways of defining $n_{u,t}$. We considered (1) number of days on which feature t appeared in the (time interval, event type) cell for user u (ie #days) and (2) number of times feature t appeared in the (time interval, event type) cell for user u (ie #occurrences). While both forms of $n_{u,t}$ perform fairly similar, the best performance comes from the #days definition of $n_{u,t}$. A likely explanation is that it captures sustained user interest in an activity for a conversion to happen. Additionally, #days is more robust, e.g., some Web pages reload automatically and can dramatically bloat the number of the occurrences, while their adverse effect on #days is significantly limited.

We evaluate four different forms of TF defined in terms of $n_{u,t}$: boolean, log, natural, and ratio and different forms of and IDF functions (see Table 1). The results are shown in Figure 4. First, we note that the rightmost bar corresponds to no TFIDF weighting, i.e., TF $=n_{u,t}$ (natural), IDF $=1$ (no). It achieves an AUC of 0.56 while our best setting achieves AUC $=0.65$. This shows the significance of TFIDF weighting in the vector space model for users.

Among the various TF forms, TF $=\frac{n_{u,t}}{\sum_{t'\in u} n_{u,t'}}$ (ratio) is consistently the best across all forms of IDF. This is remarkable because we normally expect more active users to be more likely to convert, and vice versa. However, this TF form, normalized with respect to the total amount of user activity, shows that it is not necessarily the case. Instead, it is the user activity in a specific topic relative to her overall activity that leads to the best prediction performance.

Among the IDF variants, we found $max\{0, log(\frac{N-df_t}{df_t})\}$ (prob idf) to perform the best, with $log(\frac{N}{df_t})$ and $\sqrt{\frac{N}{df_t}}$ taking the second and third spot.

4.6 Individual Campaigns

To investigate the variation in performance over individual campaigns, we divided the campaigns into 3 sets, *large*, *medium* and *small*, where the *large* group contains the top one-third of the campaigns with the highest number of conversions, while the *small* group contains the bottom one-third. We found that our approach performs well in all three groups, in fact it does better on the medium (0.657) and small (0.673) campaigns compared to the large ones (0.624). Since most methods suffer when they are trained on fewer positive examples, this is a nice advantage of our approach, as it seems it can be applied to these tail/small campaigns, where other methods, especially discriminative ones, struggle to perform. This can also explain why our approach did not benefit much from using pseudo-positive examples based on users who clicked on the ads but did not immediately convert. These pseudo-positive examples typically prove helpful on small campaigns where there are few positive examples. However, our approach already performs well on those campaigns.

In further analysis of large campaigns we found that these campaigns are quite diverse and are thus associated with a variety of goals (ad groups, in computational advertising terminology), and thus different kinds of users convert on them. In other words, there is more heterogeneity among the seed users in these campaigns (compare to the smaller ones), and when these diverse users are combined together during the query construction process, they weaken the signal and hurt the performance of our approach.

5. RELATED WORK

Audience selection/segmentation is an important task in marketing. Since customers have diverse interests and needs, marketing strategies that target individual segments perform better than a single global strategy for the entire population [6]. In this paper we formulated the problem of audience selection in display advertising within the IR paradigm.

Our approach has similarities to user profiling and behavioral targeting based on observed past events. A commonly used approach in behavioral targeting is to infer user interests and use them to predict whether she will be interested in a product. Chen et al. [4] proposed a linear regression model to leverage user behavior for predicting ad clicks. However, learning regression models for conversions is difficult since conversions are typically several order of magnitude fewer than clicks. Another difficulty in modeling user behavior is that user interests are not always fixed, and some interests are transient, influenced by media and pop culture. Shmueli-Scheuer et al. [13] used a decay model to predict clicks to give recent features more emphasis. In their analysis of query interests, Wedig et al. [14] found that users tend to stabilize on a distribution of interests. In contrast, Liu et al. [9] found that user interests change from month to month based on their analysis of news topics and attributed this change to the task of browsing news (compare to the task of issuing queries studied by Wedig et al. [14]).

In their empirical study on understanding the potential of behavioral targeting for online advertising, Yan et al. [15] studied how ad click-through rates relate to the search queries and page views of the users who clicked on these ads. They found that users who clicked on the same ads tend to have more behavioral similarities than users who clicked on different ads. Our work builds on this hypothesis to perform audience selection, where we model user similarity using multiple sources of information, such as page views, clicks on search results, and the like, and not only ad clicks. We also focus on conversions rather than clicks. In particular, we first construct a query based on the seed set of users who converted in the past, and then execute this query against an index of user profiles. This allows us to retrieve more users who are similar to the seed set, and are therefore likely to convert in the future.

6. CONCLUSION

In this paper, we formulated the problem of audience selection as an information retrieval task, retrieving user profiles instead of documents. These profiles are constructed based on users' online actions such as browsing and searching. Unlike documents, which are often coherent units, user activity is composed of multiple events at distinct time points, each potentially having distinct intent. We then defined a formal retrieval model for the audience selection task based on language modeling and vector space modeling.

We found that both vector space and language models performed well for this task. We also found that using TFIDF feature weighting, as well as using a set of non-relevant users for Rocchio-style query expansion significantly improves performance. Although recent activity proved more important than past activity, bucketing based on time intervals degraded performance, as it led to the sparsity of data.

Acknowledgements.

We thank Fernando Diaz for his insightful comments.

7. REFERENCES

[1] A. Bagherjeiran, A. O. Hatch, and A. Ratnaparkhi. Ranking for the conversion funnel. In *SIGIR*, 2010.

[2] J. Bai, J.-Y. Nie, H. Bouchard, and G. Cao. Using query contexts in information retrieval. In *SIGIR*, 2007.

[3] D. Chakrabarti, D. Agarwal, and V. Josifovski. Contextual advertising by combining relevance with click feedback. In *WWW '08*, pages 417–426, 2008.

[4] Y. Chen, D. Pavlov, and J. F. Canny. Large-scale behavioral targeting. In *KDD '09*, 2009.

[5] M. Gonen. Receiver operating characteristic (ROC) curves. *SAS Users Group International (SUGI)*, 31:210–231, 2006.

[6] B. J. P. II. Mass customizing products and services. In *Strategy & Leadership. Vol. 21, no. 4*, 1993.

[7] T. Joachims. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In *ICML'97*, pages 143–151, 1997.

[8] A. Lacerda, M. Cristo, M. A. Goncalves, W. Fan, N. Ziviani, and B. Ribeiro-Neto. Learning to advertise. In *SIGIR '06*, pages 549–556, 2006.

[9] J. Liu, P. Dolan, and E. R. Pedersen. Personalized news recommendation based on click behavior. In *IUI*, 2010.

[10] C. D. Manning, P. Raghavan, and H. Schuetze. *Introduction to Information Retrieval.* Cambridge University Press, 2008.

[11] M. Richardson, E. Dominowska, and R. Ragno. Predicting clicks: estimating the click-through rate for new ads. In *WWW '07*. ACM Press, 2007.

[12] J. J. Rocchio. Relevance feedback in information retrieval. In *The SMART Retrieval System: Experiments in Automatic Document Processing.* 1971.

[13] M. Shmueli-Scheuer, H. Roitman, D. Carmel, Y. Mass, and D. Konopnicki. Extracting user profiles from large scale data. In *MDAC'10*, 2010.

[14] S. Wedig and O. Madani. A large-scale analysis of query logs for assessing personalization opportunities. In *KDD '06*, pages 742–747, 2006.

[15] J. Yan, N. Liu, G. Wang, W. Zhang, Y. Jiang, and Z. Chen. How much can behavioral targeting help online advertising? In *WWW '09*, 2009.

[16] C. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *CIKM '01*, pages 403–410, 2001.

[17] C. Zhai, X. Lu, X. Ling, X. He, A. Velivelli, X. Wang, H. Fang, and A. Shakery. Uiuc/musc at trec 2005 genomics track. In *TREC '05*, 2005.