

# Data Driven Recommendations for Display Advertising

Rushi P. Bhatt, Kevin L. Chang, Vijay K. Narayanan, Rajesh G. Parekh  
Yahoo! Labs, Santa Clara, CA  
{rushi|klchang|vnarayan|rparekh}@yahoo-inc.com

## ABSTRACT

Advertisers running display advertising campaigns often request actionable recommendations for booking the most effective new ad campaigns and improving the performance of ongoing campaigns. Typically, the recommendations desired by advertisers fall into two broad categories: improved performance in terms of metrics like CTR, CPC, and CPA; and increased reach, which is the number of unique users exposed to the campaign. Account managers provide recommendations to advertisers based on their personal intuition of the advertisers' needs. This approach is not scalable and recommendations are often not consistent across account managers and advertisers. We developed a data-driven approach that leverages historical ad campaign information and granular user data coupled with the advertiser's current campaign objectives to make effective recommendations. This paper presents the following key results:

1. A novel application of the PLSI algorithm for effectively identifying neighbors of ad campaigns.
2. Application of large-scale collaborative filtering methods for making recommendations to optimize ad campaigns.
3. Design of a complementary user segments algorithm to significantly increase the reach of ad campaigns, while maintaining or improving performance.

The key advantages of this method of producing recommendations are scale: even small advertisers who do not have dedicated account managers can leverage them, and novelty: mining historical campaign and granular user level interaction data enables discovery of non-obvious recommendations.

## Categories and Subject Descriptors

I.2.6 [Artificial Intelligence]: Learning; I.2.1 [Artificial Intelligence]: Applications and Expert Systems

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ADKDD'10, July 25, 2010, Washington D.C., USA.  
Copyright 2010 ACM 978-1-4503-0221-0 ...\$10.00.

## General Terms

Computational advertising, Recommendations, Display Advertising

## Keywords

computational advertising, actionable recommendations, frequent itemset mining, statistical hypothesis testing, large scale recommendation system

## 1. INTRODUCTION

Online display advertising is a popular method of advertising on the internet. Display ads include graphical or banner ads in the form of images or rich media like flash or mpeg that are shown to users alongside web content. At Yahoo! we have developed tools to automatically provide advertisers with a comprehensive platform to monitor and optimize their display ad campaigns on the Yahoo! ad network. A key feature of these tools is the ability to provide actionable recommendations for booking the most effective new ad campaigns and enhancing ongoing ad campaigns. Account managers often provide recommendations to advertisers based on their domain expertise and personal intuition of the advertisers' needs. This approach is not scalable and the recommendations are often not consistent across account managers and advertisers. Furthermore, the domain expertise remains restricted to individual account managers instead of getting assimilated by a larger set of managers. In this paper we present a systematic data-driven approach for generating actionable recommendations for display ad campaigns. This approach has been deployed in production at Yahoo! and is being used by account managers to make recommendations to advertisers they manage.

Automated recommendation systems have been successfully used on e-commerce web sites like Amazon [12] and Netflix [3, 10] to deliver personalized product recommendations that suit the individual user's tastes. *Associations Rules Mining* [2] and *Collaborative Filtering* methods [8] are extremely popular in recommendation systems. These methods use historical user behavior - such as prior purchases or product ratings - and analyze the relationships between users and interdependencies between products to identify new user-product associations. The prospect of using collaborative filtering to generate recommendations for display advertising campaigns seemed promising wherein the individual ad campaigns are analogous to users and the different display advertising products available for purchase are akin to the products like books or movies. We present a large-scale collaborative filtering based method for generating display advertising product recommendations. The algorithms generate recommendations with respect to a *neighborhood* of a given ad campaign to ensure that the provided

recommendations are relevant to the ad campaign. We use a machine learning approach - *Probabilistic Latent Semantic Indexing* (PLSI) [9] - to automatically learn and generate the neighborhood of each ad campaign. PLSI and its variants have been used very effectively in the information retrieval literature to identify similar documents. We use PLSI to identify similar ad campaigns. Further, the generated recommendations are filtered by key criteria - *performance, novelty, popularity, inventory availability, and price* to ensure that the most relevant recommendations capable of improving ad campaign performance are surfaced to advertisers.

Whereas PLSI recommendations are optimized for improving performance, advertisers also seek recommendations to increase the *reach* of their display ad campaigns while maintaining or improving performance. The reach of an ad campaign is the number of unique users that are exposed to the relevant advertisements. The recommendations generated by collaborative filtering systems are by design based completely on advertisements run in the past. As a result, the recommendations only include display ad products that were historically purchased by the advertisers. Significantly increasing the reach of an ad campaign mandates looking beyond products that were purchased previously and exploring new or formerly unexplored products. A challenge in considering previously unused products is the lack of historic performance data. This paper describes the *complementary user segments* algorithm for effectively exploring these new products and robustly estimating their performance.

Audience segmentation for more effective ad targeting has been extensively studied in the market research literature. The Nielsen Claritas PRIZM provides a broad segmentation of the US population based on consumer behavior, geographic, and demographic data [1]. The explosive growth of the internet in the past decade has opened up tremendous opportunities for marketers to target their ads online. Traditional advertising based on *geographic* and *demographic* segmentation and *contextual relevance* [5] has been effective in online advertising. Further, recent work has shown how targeting approaches based on more granular user data like *behavioral targeting* [7, 16] and *social targeting* [14] can be effectively applied to online display advertising. These methods focus on building effective models for targeting the display ads to right users.

A typical display advertising campaign will include a combination of several of the above segmentation and targeting approaches. For example, a campaign may focus on women between the ages of 25 and 44 who visit the Yahoo! Shine property. This campaign combines a demographic segmentation - women between ages 25 and 44 with contextual segmentation - users visiting Yahoo! Shine. The number of such combinations of user segments is very large. We propose approaches for generating recommendations of user segments that can be targeted by advertisers with specific *performance* and *reach* goals in mind. More importantly, the recommendations are produced automatically at scale so that they are available even to small advertisers who do not have dedicated account managers. To the best of our knowledge, this is the first published work on automated, large-scale recommendations for display advertising.

The key contributions of this paper are:

1. A novel application of the PLSI algorithm for effectively identifying neighbors of ad campaigns.
2. Application of large-scale collaborative filtering methods for making recommendations to optimize ad campaigns.

3. Design of a complementary user segments algorithm to significantly increase the reach of ad campaigns, while maintaining or improving performance.

The remainder of the paper is organized as follows: Section 2 presents an overview of the terminology and the notation used in the paper. Section 3 motivates the need for a data-driven approach to producing recommendations. Section 4 describes the machine-learning approach to identify neighbors of an ad campaign. Section 5 presents the statistical hypothesis-based and large-scale collaborative filtering approaches for making campaign-specific performance recommendations. These algorithms are applied to the PLSI neighborhood of the campaign of interest. Section 6 describes the *complementary user segments* algorithm for making recommendations that increase user reach while maintaining campaign performance. Section 7 concludes with a summary and directions for further research.

## 2. TERMINOLOGY

A display ad campaign is also called an insertion order (IO). We will use the two terms interchangeably. Let  $I$  denote an IO.  $I$  is a collection of lines denoted by  $I_1, I_2, \dots, I_m$ , where  $m$  is the total number of lines in  $I$ . A line completely specifies the positioning and targeting of a single advertisement. Lines are instantiations of line profiles (denoted by  $P$ ), where each line profile specifies *placement attributes* and *user targeting attributes*. Placement attributes specify the *property*, which is the web page on which the ad is displayed, and the *position*, which indicates where on the web page the ad is shown. For example, an ad with “property = autos” and “position=North” indicates that the ad should be shown on the Yahoo! Autos webpages (with content related to cars), in the North position, which is the ad displayed near the top of the page. Other properties include Yahoo! Mail, Finance, and News. The property Network is used to denote the collection of all properties in the network. User targeting attributes include geographic location (e.g., state and country), user interests according to the Yahoo! behavioral targeting (BT) taxonomy (e.g. users interested in Automotive) and user-declared demographic attributes (age and gender). Any single placement or user targeting attribute is a feature-value pair denoted by  $FV$  and is represented as  $(f = v)$  where,  $f$  is the type of the feature and  $v$  is its value. For example,  $FV = (\text{property} = \text{Autos}/\text{Ford Trucks})$  is a feature-value pair where the feature is the placement attribute property and the value is the list of pages related to Ford Trucks on the Yahoo! autos pages. A line profile  $P$  is represented as a conjunction of feature-value pairs. i.e.,  $P = FV_1 \wedge FV_2 \wedge \dots \wedge FV_k$ , where  $k$  is the total number of feature-values specified in  $P$ . In this presentation,  $A$  and  $B$  are also used to denote line profiles. A user segment  $U_P$  is the set of users that satisfy all attributes of a line profile  $P$ .

Table 1 provides example lines from an IO booked by an automotive dealership.

Any feature-value pair ( $FV$ ) left unspecified in a line profile  $P$  implies that all values for the corresponding feature value pair  $FV$  are admissible. For example, the line  $I_1$  in Table 1 does not contain an entry for the feature gender, indicating that the ad can be shown to both men and women.

## 3. PERFORMANCE AND REACH RECOMMENDATIONS: MOTIVATION

The space of attributes defining a line profile is the cross-product of the placement attributes and user targeting at-

line	line profile
$I_1$	(property = Network) $\wedge$ (position = North) $\wedge$ (age = young OR middle age) $\wedge$ (geo = US) $\wedge$ (BT = Automotive)
$I_2$	(property = Mail) $\wedge$ (position = Large Rectangle) $\wedge$ (geo = US) $\wedge$ (BT = Life-stage/Parenting and Children)
$I_3$	(property = Autos) $\wedge$ (position = North) $\wedge$ (geo = US)

Table 1: Example IO for an automotive dealership in Northern California.

tributes. Given that each individual attribute in the cross-product can take on multiple values (a hierarchy of values in the case of geo locations and BT segments), the space of possible line profiles is extremely large (greater than 10 billion). Very few nodes (of the order of a few 10's of thousands) from this large space have actually been explored by advertisers in the form of historically booked lines. Advertisers seek recommendations for line profiles that they should book in order to either improve the performance or increase the reach of their campaigns. Performance recommendations are geared towards optimizing campaign performance (in terms of metrics like CTR, CPC, or CPA). More specifically, performance recommendations are of two types: *1-D recommendations* [4] that comprise of a single feature value pair (e.g., gender = Female) and *full product recommendations* [13] that recommend a line profile (e.g., (property = Mail)  $\wedge$  (position = North)  $\wedge$  (geo = US)  $\wedge$  (BT = Sports/Snow)). Reach recommendations [6] are designed to substantially increase the number of unique users reached by ad campaigns, and are in the form of new line profile recommendations. Some example 1-D recommendations are shown in Table 2, while example full product recommendations are shown in Table 3.

#### 4. PLSI BASED CAMPAIGN NEIGHBORHOOD GENERATION

Collaborative filtering approaches are popular for generating recommendations [15] and can be leveraged to making recommendations of display advertising products to advertisers. The straight-forward approach of generating recommendations based on all advertisers and all the display advertising products they purchased in the past tends to produce recommendations that are overly generic and at times inappropriate for specific advertisers. This is because advertising campaigns have very different objectives and might need to reach very different audiences to be effective. For example, recommendations that might be appropriate for automotive advertisers might not be suitable for consumer packaged goods advertisers. For recommendations to be relevant, they need to be generated with respect to a *neighborhood* of a given ad campaign. The neighborhood of an ad campaign is defined as a set of similar ad campaigns. Identifying the neighborhood of an ad campaign is challenging. Manual specification of the neighborhood of each ad campaign is not scalable. Further, the simple approach of grouping together ad campaigns from all advertisers belonging to the same product vertical (e.g. consumer electronics) would be inappropriate because large advertisers run multiple campaigns spanning several different brands or product lines, each with different objectives. In this paper we proposed using a machine learning approach - *Probabilistic Latent Semantic Indexing* (PLSI) [9] - to automatically learn and generate the neighborhood of each ad campaign. PLSI and its variants have been used very effectively in the information retrieval literature to identify similar documents by clustering documents based on latent topic distributions. We use PLSI to identify similar ad campaigns. A campaign is represented as a collection of the words derived from its

constituent line profiles. For example, the campaign run by an automotive dealership shown in Table 1 is represented as a collection of terms: Network, Mail, Autos, North, Large Rectangle, young, middle age, US, Automotive, Parenting and Children. We treat IOs  $I$  as documents and terms from the lines as words  $w$  and use the PLSI method to represent each IO as a distribution over latent topics  $T$  as follows:

$$P(T|I) = \sum_w P(T|w, I) \cdot P(w) = \sum_w \frac{P(w, I|T) \cdot P(w) \cdot P(T)}{P(w, I)}.$$

Assuming that the campaigns and words are conditionally independent given the topic, this reduces to

$$P(T|I) = \sum_w \frac{P(w|T) \cdot P(C|T) \cdot P(w) \cdot P(T)}{P(w, I)}.$$

For each campaign, PLSI estimates the probabilities of latent topics that may have given rise to the specific campaign lines. See [9] for details on estimating the probabilities of the model. The similarity between two campaigns is then measured using the Jensen-Shannon distance [11], from information theory, between the topic probability distributions of the two campaigns. If  $P$  and  $Q$  are discrete distributions over terms  $T$ , this is:

$$\frac{1}{2} \sum_T P(T) \log \frac{P(T)}{\frac{1}{2}(P(T) + Q(T))} + Q(T) \log \frac{Q(T)}{\frac{1}{2}(P(T) + Q(T))}$$

For the purpose of the experiments presented in this paper the number of topics  $T$  was set to 50. We define the neighborhood of a campaign as the  $K$  campaigns that are most similar to the given campaign.

Each topic is a collection of words in the campaign. In an example from a PLSI model of real campaigns, the words (and corresponding weights) with the 5 highest weights in one cluster are finance (0.25), loan (0.24), mortgage (0.17), US (0.06), and deposit (0.05), which are related to the topic of *finance*. In another cluster, the words with the 5 highest weights are sport (0.43), fantasy (0.10), nfl (0.10), ncaab (0.03) and mlb (0.03) which are common words related to the topic *sports*.

For each IO  $I$ , let  $N^I$  denote the set of neighboring campaigns (IOs) generated by PLSI. We identify the  $K$  closest neighboring campaigns, rank-ordered by their distance from this campaign and denote them by  $N_1^I, \dots, N_K^I$ . For the purpose of our experiments we empirically found  $K = 150$  to give good performance recommendations. Let  $CTR(I)$  denote the CTR of IO  $I$ .  $CTR(I)$  is the ratio of the number of total clicks to the number of total views on all lines of  $I$ . The CTR of an individual line  $I_m$  is denoted by  $CTR(I_m)$ . Analogously, define  $CTR(I, FV)$  as the mean CTR of all lines in IO  $I$  whose line profile contains feature-value  $FV$ ;  $CTR(N^I, FV)$  as the mean CTR of all lines among the set of neighbors of IO  $I$  whose line profile contains feature-value  $FV$ ; and  $CTR(U_P)$  as the mean CTR of the user segment  $U_P$  (the set of users whose characteristics satisfy the attributes of the line profile  $P$ ). The standard deviation of the CTR of lines in IO  $I$  is denoted by  $\sigma(CTR(I, FV))$ . Similarly,  $\sigma(CTR(N^I, FV))$  is the standard deviation of the

Recommendation Type	Advertiser Category	Feature-Values in Campaign	Feature-Values Recommended
1-D (hypothesis testing)	Consumer Electronics	(property = tv), (property = launch.com), (property = sports)	(property = entertainment)
		(BT = Technology/Consumer Electronics/Home Video), (BT = Entertainment/Television), (BT = Entertainment/Movies)	(BT = Entertainment/Music)
	Mortgage	(BT = Finance/Real-estate/Residential Purchase)	(BT = Finance/Loans/Mortgage)
1-D (affinity)	Consumer Electronics	(property = tv), (property = launch.com), (property = sports)	(property = movies), (property = games)
	Mortgage	(property = mail), (property = my), (property = network)	(property = finance), (property = real estate)
		(BT = Finance/Real-estate/Residential Purchase)	(BT = Finance/Loans/Mortgage)

Table 2: Example 1-D performance recommendations

Recommendation Type	Advertiser Category	Line Profile in Campaign	Line Profile Recommended
Full product (hypothesis testing)	Consumer Electronics	(property = games) $\wedge$ (position = Sky) $\wedge$ (country = US)	(property = movies) $\wedge$ (position = Large Rectangle) $\wedge$ (country = US)
	Mortgage	(property = network) $\wedge$ (position = Large Rectangle) $\wedge$ (BT = Finance/Real Estate/Residential Purchase) $\wedge$ (country = US)	(property = real estate) $\wedge$ (position = North) $\wedge$ (country = US)
Full product (affinity)	Consumer Electronics	(property = mail) $\wedge$ (position = North) $\wedge$ (BT = Entertainment/movies)	(property = entertainment) $\wedge$ (position = North) $\wedge$ (country = US)
	Mortgage	(property = network) $\wedge$ (position = Large Rectangle) $\wedge$ (BT = Finance/Real Estate/Residential Purchase) $\wedge$ (country = US)	(property = network) $\wedge$ (position = Large Rectangle) $\wedge$ (BT = Finance/Loans/Mortgage) $\wedge$ (country = US), (property = real estate) $\wedge$ (position = Large Rectangle) $\wedge$ (country = US)

Table 3: Example full product performance recommendations

CTR of all lines among the set of neighbors of IO  $I$  whose line profile contains the feature-value  $FV$ ; and  $\sigma(CTR(I_P))$  is the standard deviation of the CTR of the user segment  $U_P$ .

## 5. PERFORMANCE RECOMMENDATIONS

We present two approaches for generating performance recommendations – a statistical hypothesis testing based strategy and a large-scale collaborative filtering strategy for identifying better performing frequent itemsets [2]. Both approaches generate a candidate list of recommendations. These recommendations are filtered to ensure that the recommendations presented to the advertiser are maximally useful. The filtering criteria implemented are:

1. *performance*: the recommended products must have a certain level of historical performance in terms of click-through rate (CTR) on displayed banners, or the cost to advertiser per received click (CPC), or a specified cost per action (CPA)
2. *novelty*: the recommended products should not have been booked by the advertiser in the past in a different campaign

3. *popularity*: the recommended product should have been adopted by a certain minimum number of advertisers and should have served a certain minimum number of impressions.
4. *availability*: the recommended products should have sufficient amount of inventory available to meet the advertiser’s desired reach goals.
5. *price*: the recommended product should be priced in line with the advertiser’s budgetary constraints.

The above criteria are incorporated as post-filtering rules. The recommendations after applying the filters are rank-ordered in descending order of historical performance and presented to the advertiser.

The following sub-sections describe two types of performance recommendations: (i) *1-D* recommendations that recommend a single placement or user targeting attribute FV pair and (ii) *full product* recommendations that recommend a complete line profile.

### 5.1 1-D recommendations

In what follows, we describe two different recommendation algorithms, based on hypothesis-testing and frequent itemsets. The overall flow in both these approaches is similar, namely, generating a set of candidate recommendations

through a search in the neighborhood followed by applications of heuristically derived filters. We use CTR as the example performance metric in the discussion below; the steps are identical for other performance metrics. Note that the absolute values of the performance metrics are, in general, sensitive to a number of other parameters including the category of the advertiser, how appealing the ad creative is, etc. In such cases, the performance metrics should be suitably normalized to be comparable across different campaigns.

### 5.1.1 Hypothesis-testing based recommendations

Hypothesis-testing based recommendations are generated by identifying better-performing  $FVs$  in the neighborhood of the IO  $I$  of interest. For each unique  $FV$  in the neighboring campaigns, the performance of lines containing the  $FV$  is compared against the performance of lines not containing the same  $FV$ .

**Performance Normalization:** CTR and other performance metrics vary widely across different web-sites, campaigns, and even positions on which the ad is displayed, making it essential to normalize for this intrinsic variation before comparing across different lines and campaigns. We normalize the CTR to the CTR of a given line relative to the overall campaign performance as well as the overall CTR of the  $FV$  under consideration. For a given Feature-Value  $FV$ , IO  $I$ , and line  $I_m$ ,

$$CTR_{norm,FV}(I_m) = \frac{CTR(I_m)}{\omega \cdot CTR(I) + (1 - \omega) \cdot CTR(FV)},$$

where  $CTR(I_m)$  is the CTR of line  $I_m$  (in IO  $I$ ) and  $CTR(FV)$  is the average CTR of all lines that contain  $FV$ . The parameter value  $\omega = 0.5$  is determined empirically.

The intuition behind normalizing by  $CTR(FV)$ , which is the CTR of all lines with  $FV$  in the entire data set, is to ensure that we do not simply recommend  $FVs$  that have high baseline performance, but rather  $FVs$  that perform particularly well in the specific context of the neighborhood.

**CTR Z-score Computation:** The Z-score measures the difference between the mean values of two different sets of observations, normalized by the individual group variances. Essentially, the Z-score is a measure of confidence that the two means differ. We utilize this measure to select only the feature-values that contribute towards improved IO performance with high statistical significance. Each candidate Feature-Value recommendation  $FV$  for IO  $I$  is scored using a weighted sum of Z-scores of the difference in performance of lines with the  $FV$  and lines without the  $FV$  in the neighborhood ( $Z(N^I, FV, I)$ ) as well as within IO  $I$  itself ( $Z(I, FV, I)$ ):

$$Z(FV, I) = \alpha Z(N^I, FV, I) + (1 - \alpha) Z(I, FV, I)$$

where

$$Z(X, FV, I) = \frac{CTR(X, FV) - CTR(X, \bar{FV})}{\sqrt{VAR(X, FV) + VAR(X, \bar{FV})}}$$

where  $\alpha = 0.25$  is an empirically tuned parameter. In the above computation,  $CTR(N^I, FV)$  denotes the weighted mean  $CTR_{norm,FV}(I_m)$  of the lines  $I_m$  in the neighborhood of IO  $I$  that contain the feature value  $FV$ . The weight  $w_{N^I_k}$  is proportional to the reciprocal of the rank  $k$  of the neighbor.  $CTR(N^I, \bar{FV})$  denotes a similarly computed weighted CTR of lines in the neighborhood that do *not* contain  $FV$ .

Variances in CTR of lines with  $FV$ ,  $VAR(N^I, FV)$ , and without,  $VAR(N^I, \bar{FV})$ , are computed analogously.

**Recommendation Output:** The candidate feature-value pairs  $FV$  are filtered based on the criteria described above and the surviving recommendations are presented to the advertiser in descending order of  $Z(FV, I)$ .

### 5.1.2 Frequent itemset based recommendations

While the hypothesis testing based strategy recommends  $FVs$  that are popular and successful in the neighboring IOs, the frequent itemset based approach brings more context into the recommendations by recommending items from sets that contain  $FVs$  that are already used.

Frequent itemset mining usually operates on a set of *transactions* and *items* in each transaction. In its prototypical application, a transaction is a customer purchase of a basket of individual items; frequent itemset mining typically identifies frequently co-occurring items in the purchased baskets of several different transactions. In our application, each neighboring IO  $N^I_k$  of IO  $I$  is considered a transaction; the  $FVs$  in  $N^I_k$  are the items in the basket. Frequently co-occurring  $FV$  pairs in neighboring campaigns are identified through the frequent itemset mining algorithm [2]. Since the frequent itemsets for a campaign  $I$  are computed using only the neighboring IOs, the recommendations tend to be specific and relevant to  $I$ . From each frequent item pair ( $FV_u, FV_v$ ) where  $FV_u$  was used within the campaign  $I$  but  $FV_v$  was not, the feature value  $FV_v$  is identified as the candidate recommendation.

The candidate recommendations are filtered by the the filtering rules described above and the surviving recommendations are ranked ordered in descending order of the historical performance and presented to the advertiser.

## 5.2 Full Product Recommendations

### 5.2.1 Hypothesis testing based recommendations

Statistical hypothesis testing based recommendations for full-products are similar to the 1-D recommendations, in that we search in the neighborhood for better performing lines. Instead of selection through Z-scores of differences in CTR estimates, lines that outperform the mean CTR of their respective IOs by specified thresholds are selected for recommendation. The algorithm for producing full product recommendations is as follows.

- For every line profile  $P$  in  $N^I_k$ , add  $P$  to the recommendation candidate set if lines with this profile in the neighborhood have a higher performance than the average performance of all lines
  1. in this campaign:  $CTR(N^I_k, P) \geq CTR(I) + \gamma\sigma(CTR(I))$
  2. with this profile:  $CTR(N^I_k, P) \geq CTR(P) + \delta\sigma(CTR(P))$
  3.  $\gamma = 0.8$  and  $\delta = 0.8$  are chosen empirically.
- Filter the candidate line profiles using the post-filtering rules described above.

### 5.2.2 Frequent itemset based recommendations

The frequent itemset based full profile recommendation algorithm is identical to its 1-D counterpart, except that each item in the transaction sets (i.e., elements in itemset for each neighboring IO) is a line profile  $P$  instead of individual Feature-Value  $FV$ .

### 5.3 Experimental Results

Evaluating the quality of recommendations is a challenge. The ultimate proof of the concept is in monitoring the *performance* and *reach* of the recommendations that are used to book lines in display advertising campaigns. However, advertisers would first like to see some offline metrics to convince them to use the recommendations in actual campaigns. We used data from historical campaigns run by advertisers over a two year period to generate recommendations. The total number of campaigns used after filtering out test and internal marketing campaigns was of the order of hundreds of thousands. Tables 2 and 3 show several actual recommendations produced by the algorithms. We set the values of the various parameters in the algorithms to provide recommendations that are intuitively appealing based on our domain knowledge of these campaigns. We reviewed these recommendations with several account executives, who provided several campaign insights that helped to better tune the heuristics components of the algorithms. However, this approach is neither scalable nor quantitative for evaluating the quality of the recommendations.

We used the following strategy for offline quantitative evaluation of recommendation quality. The historical campaigns over the two year period were split by time into a train set and a test set. Campaign lines that ran in the first 23 months were part of the training set and the lines that ran in the last one month were part of the test set. The neighborhood of each campaign and the corresponding recommendations were generated using only the training set. The performance of recommendations that actually appeared in the test set (i.e., the recommendations had been independently used by the advertiser in the campaigns during the test period) was evaluated. The following two quality measures were evaluated – (i) *Lift*, which is the ratio of the CTR of the recommendation over the mean CTR for the campaign and the (ii) percentage of *Good Recommendations*, which is the percentage of all recommendations whose performance was better than the mean CTR for the campaign. Table 4 shows the results for the 1-D and full-product recommendations. In this table, *Lines* is the number of lines in the test set that had used a recommended 1-D FV pair, *IOs* is the number of campaigns with at least one matching recommendation, *Improved IOs* denotes the percentage of matching IOs where the recommendation produced a lift above 1, and *Lift* and *Good Recommendations* are the metrics defined above. Note that in Table 4 the number of lines and IOs have been anonymized for confidentiality. The relative proportion of lines and IOs for which recommendations are produced by each of the methods, the lift, and the percentage of good recommendations is preserved.

The above offline evaluation approach may appear to be optimistic since only the recommendations that were independently booked and run by the specific advertiser were used for evaluation. How the other recommendations that were not booked and run by this advertiser would perform is not known for the specific advertiser’s campaign. However, we do know from historical data that these lines had performed well when booked in campaigns that are neighbors of the specific advertiser’s campaign, and hence we can reasonably expect these recommendations to perform well for this advertiser as well.

Table 4 shows that the statistical hypothesis testing based method and the affinity based method produce a significantly large number of non-overlapping recommendations. Thus, taking the union of the sets of recommendations from both these approaches results in a larger total number of recommendations that also perform well.

## 6. COMPLEMENTARY USER SEGMENTS FOR REACH RECOMMENDATIONS

The algorithms presented in Section 5 are designed to improve campaign CTR performance. These fine-grained targeting recommendations can provide superior performance, but the available user inventory for such targeting profiles may be too small to fulfill the advertiser’s demands for the number of users exposed to the campaign. Furthermore, the above algorithms generate recommendations based on what has been successfully tried in the past, but cannot generate novel recommendations. In this section, we describe the Complementary Segments algorithm that addresses both these concerns: it explicitly considers reach by leveraging user-level clickstream data to consider line profiles that have not been run in past campaigns.

Ads are manually categorized into one or more of the interest categories in the BT taxonomy, which we refer to as the ad category. Consider an advertiser who targets a user segment  $U_A$  satisfying a line profile  $A$  with an ad of category  $C$ . The number of users who satisfy this profile is the reach of the profile. The reach may be too small or may even be booked by other advertisers and be completely unavailable for new advertisements. A complementary segment is a set of users  $U_B$  satisfying a line profile  $B$ , such that

1. targeting  $U_A$  and  $U_B$  substantially increases the reach over targeting  $U_A$  alone and
2. the additional unique users garnered by targeting  $U_B$  offer similar performance, (as measured by CTR) to  $U_A$  on ads belonging to category  $C$ .

Figure 1 illustrates the approach using an example. Suppose an advertiser is targeting ads with category  $C = \text{Autos}$  with line profile  $A = (\text{Property} = \text{Sports})$ . The targeted segment  $U_A$  is represented by the blue oval. This segment has an average CTR of  $x\%$  on category  $C = \text{Autos}$ . The segment  $U_B$  comprising of users targeted by the line profile  $B = (\text{Property} = \text{Mail}) \wedge (\text{Gender} = \text{Male}) \wedge (\text{BT} = \text{Autos/Sports cars})$  is represented by the yellow oval. Users in the intersection of segments,  $U_A \cap U_B$ , have a CTR of  $2.65x\%$  on Autos ads on the Sports property. The segment  $U_B \setminus U_A$  has a CTR of  $4.65x\%$  (i.e., 4.6 times higher CTR) on Autos ads on the Mail property. Furthermore,  $|U_B \setminus U_A| > 0.2|U_A|$ .  $U_B$  is thus a complementary segment of  $U_A$ , since it provides an additional reach of 20% over  $A$  and has comparable (in this case, superior) performance to  $A$  on Autos category ads.

**Remark** This analysis does not require the segments  $A$  and  $B$  to have been booked in the past. Typically, several ads of category  $C$  will be shown to a large set of users that can be considered “random”. Thus, for any arbitrary segments  $A$  and  $B$  that are sufficient large, many of these random users that view a category  $C$  ad will happen to be in segments  $A$  or  $B$ . Therefore, complementary segments analysis is indeed capable of producing novel recommendations.

### 6.1 Algorithm

Define  $CTR_C(U)$  to be the CTR of users in the set  $U$  on ads of category  $C$ . Our main algorithm requires five input parameters  $(X, Y, Z, N, R)$  that control the quality and number of recommendations produced by the algorithm.

1. For each ad category  $C$  and user segment  $A$  of interest, identify candidate line profiles  $B$ . The candidates  $B$  are chosen through an exhaustive search through possible segments and pruning those segments that do not contain a preset minimum number of users.

1-D Recommendations					
Recommendation Strategy	Lines	IOs	% Improved IOs	Lift	% Good recommendations
Hypothesis-testing based (a)	$x$	$y$	76%	1.70	69%
Frequent itemset-based (b)	$0.34x$	$0.75y$	90%	2.02	74%
Union of (a) and (b)	$1.34x$	$1.70y$	85%	1.91	72%
Full Product Recommendations					
Hypothesis-testing based (a)	$x$	$y$	88%	5.7	74%
Frequent itemset-based (b)	$0.55x$	$0.66y$	72%	1.94	69%
Union of (a) and (b)	$1.45x$	$1.3y$	80%	4.5	70%

Table 4: Performance evaluation of performance based recommendations

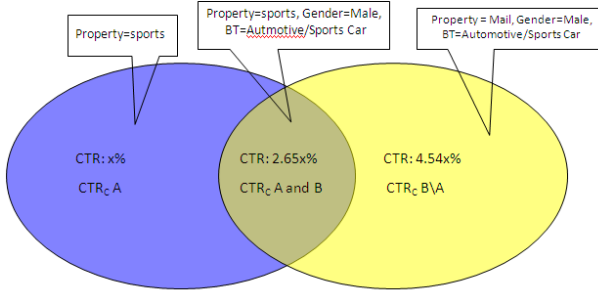


Figure 1: Example complementary segments

- Large incremental reach:** Retain only those  $B$  such that  $|U_B \setminus U_A|$  is greater than a pre-specified threshold  $R$ .
- For each remaining  $B$ , compute metrics,  $CTR_C(U_B \setminus U_A)$  and  $CTR_C(U_A \cap U_B)$ .
- Recommend all  $B$  that satisfy the following criteria as complementary segments of  $A$ .
  - Good performance in incremental user segment**  $CTR_C(U_B \setminus U_A) > X \cdot CTR_C(U_A)$  at significance level  $Z$  (using 1-sided t-test) to ensure that the recommended complementary segment  $B$  will have performance comparable to  $A$ .
  - Good performance in overlapping user segment** If  $U_A$  and  $U_B$  are not mutually exclusive, then we require that  $CTR_C(U_A \cap U_B) > Y \cdot CTR_C(U_A)$ , to ensure users belonging to both segments  $U_A$  and  $U_B$  have higher performance than those in  $U_A$  alone (with  $Y > 1$ ). This heuristic provides intuitively appealing rules.
  - Consistent performance over time** Satisfy the above two filters for at least  $N$  weeks worth of historical data in order to ensure consistency over time.

For some choices of  $A$  and  $B$ ,  $U_A$  and  $U_B$  are mutually exclusive, such as the case where  $A = (\text{gender} = \text{male})$  and  $B = (\text{gender} = \text{female})$ . We do not require rule 4(b) to hold in this case.

We make a few remarks about the rules above. Rule 4(a) is our main performance criterion.  $CTR_C(U_B \setminus U_A)$  is exactly the performance of the increased reach of the advertiser, and  $X$  therefore specifies a factor that controls this performance relative to the originally targeted user segment. Rule 4(b) is a heuristic, and the value of the parameter  $Y$  can be tuned

up to decrease the number of segments. As we see in the experimental results below, this will simultaneously increase the quality of the recommendations. Lastly, Rule 4(c) ensures that the previous rules hold for longer time horizons; increasing the value of  $N$  will increase the consistency of the performance of the recommendations.

### 6.1.1 Correcting for biases in $CTR_C(U_B \setminus U_A)$

As we mentioned before, ads are labeled with one or more categories in the BT taxonomy. In our canonical recommendation scenario, an ad belongs to category  $C$  and was originally targeted to a user segment  $U_A$ . We wish to consider targeting the set of users  $U_B$ . In the straightforward algorithm above, we would compute the CTR of every user in  $U_B \setminus U_A$  on ads of category  $C$ .

However, suppose an advertisement has been categorized into two categories  $C$  and  $D$  and further has been shown to a user in  $U_B$  who also belongs to the BT category  $D$ . This user has a hidden interest in this ad, based on his interest in the category  $D$ , which is independent of the relationship between ad category  $C$  and user segment  $B$ . Therefore counting the user's ad view on this particular ad will make the metric  $CTR_C(U_B \setminus U_A)$  overly optimistic in evaluating this recommendation.

To correct for this bias,  $CTR'_C(U_B \setminus U_A)$  is computed in place of  $CTR_C(U_B \setminus U_A)$ , by excluding users who also belong to the BT category  $D$  when the ad is also categorized into the category  $D \neq C$ .

Example recommendations for ad category Automotive are shown in Table 5.

Original Line Profile ( $A$ )	Recommended Line Profile ( $B$ )
(Property = Mail) $\wedge$ (BT = Automotive)	(Property = Autos) $\wedge$ (Gender = M)
(Property = Frontpage) $\wedge$ (Gender = M)	(Property = AT&T) $\wedge$ (Age = 45-54)
(Property = Network) $\wedge$ (BT = Automotive/Midsize/SUV)	(Gender = F) $\wedge$ (Age = 45-54) $\wedge$ (BT = Sports/Snow)
(Gender = M) $\wedge$ (Age = 21-25)	(Gender = F) $\wedge$ (Age = 35)

Table 5: Some example complementary segment recommendations for ad category Automotive

The complementary segments algorithm is parameterized by the 5 parameters  $(X, Y, Z, N, R)$ . Increasing the value of any one of these parameters increases the quality of the complementary segments, but decreases the number of recommended complementary segments, thereby providing sev-

eral dials for trading off quality and quantity of recommendations. The effect of changing parameter settings is described in Table 6.  $R$  and  $Z$  have been fixed to be  $5 \cdot 10^5$  and 0.975, respectively. These metrics were generated from running the algorithm on four weeks of historical data.

Parameters	lines w/ $\geq 1$ recs.	Total recs.
$X = 0.5, Y = 1.0, N = 3$	59%	1,967,615
$X = 0.5, Y = 1.0, N = 4$	54%	1,049,120
$X = 0.75, Y = 1.0, N = 4$	52%	569,232
$X = 0.75, Y = 1.2, N = 4$	50%	506,186

**Table 6: Sensitivity of complementary segments recommendations to parameter settings**

## 6.2 Experimental Results

In order to evaluate our algorithm, we consider the sensitivity of the recommendations to the values of  $X, Y$  and  $N$  and the quality of the recommendations produced.

The algorithm was run on three weeks of historical data with different settings of  $(X, Y, Z, N, R)$  in order to produce different sets of recommendations. Further, the recommendations were evaluated in the week immediately following the three week window.

Suppose given an ad of category  $C$  initially targeted to segment  $U_A$ , we recommend targeting user segment  $U_B$ . We will call this a *good* recommendation if, in the week’s worth of evaluation data,  $CTR'_C(U_B \setminus U_A) > X \cdot CTR_C(A)$  (i.e. a recommendation is considered good if in the evaluation period its performance satisfies the main performance criterion of our algorithm). We also measure the *lift* of the recommendation over the originally targeted segment, which is  $CTR'_C(U_B \setminus U_A)/CTR_C(U_A)$ .

For these experiments, the confidence parameter is fixed to  $Z = 0.975$  and the minimum reach threshold is set to  $R = 5 \cdot 10^5$  users.

### 6.2.1 Varying the value of $N$

The effect of varying the value of the parameter  $N$  (the number of weeks for which the candidate segments  $A$  and  $B$  must pass all rules) is described in Table 7. The last column shows the case where all 3 weeks of data are aggregated and the first two rules are applied to this aggregate, ignoring the last rule. That is, the various CTR metrics  $CTR_C(U_A), CTR_C(U_B)$ , etc. are computed for for all three weeks in aggregate. For confidentiality, only the relative CTR is presented with  $x$  denoting a reference CTR.

For all values of  $X$ , increasing  $N$  increases the percentage of good recommendations, as well as the average CTR and average lift of the recommendations. The number of recommendations produced decreases substantially as  $N$  increases. Furthermore, we see that the set of recommendations derived from  $N = 3$  improves substantially in all three metrics against the baseline of considering all weeks of training data in aggregate.

### 6.2.2 Varying the value of $Y$

We consider the effect of varying the value of  $Y$ , which controls the quality of the recommendations by setting a minimum performance requirement for  $CTR_C(U_A \cap U_B)$ . For the case where  $Y = 0$ , the rule is effectively ignored and can be considered as the performance of a baseline system. For this table, we only evaluated recommendations such that  $|U_A \cap U_B| \geq 1$ , since the rule only applies to such recommendations. The results presented in Table 8 show that for

all values of  $X$ , increasing  $Y$  generally increases all evaluation metrics while the number of recommendations decreases substantially.

## 7. CONCLUSION

We have implemented a scalable, data-driven system for providing actionable recommendations to advertisers to increase both the performance and reach of their display ad campaigns. The feedback from account managers regarding the quality of the recommendations is very positive. The following areas can be considered for additional research to further improve the quality of the recommendations. Advertisers often have explicit guidelines on line profiles they would prefer to use or avoid. This is based either on marketing intuition or on experience from the advertiser’s own past campaigns. A systematic approach of incorporating advertiser feedback or guidelines and altering the recommendations accordingly is desired. The current approach of generating recommendations based on the campaign neighborhood suffers from the *cold start* problem. The challenge is in generating the neighborhood for a new campaign or a new advertisers. One approach is to use rely on neighborhoods generated from the prior campaigns for a new campaign or to rely on expert judgement to identify advertiser(s) that are most similar (perhaps in the vertical) to the new advertiser. A more principled way of handling new campaigns and new advertisers would help to alleviate the cold start problem. The PLSI algorithm is but one approach for generating neighbors of a campaign. Other methods for neighborhood generation should be studied to analyze the relative strengths and weaknesses of these approaches compared to PLSI.

## 8. ACKNOWLEDGEMENTS

Our sincere thanks to Jignashu Parikh, Narayan Bhamidipati, Shiva Singh for numerous technical discussions and their help with the prototype implementation and the data. Thanks also to Sharon Wan, Harris Yu, Weiguo Liu, and Jim Cheng for their deep insights and their support in deploying our research prototype in production at Yahoo!.

## References

- [1] Nielsen Claritas PRIZM. [http://en-us.nielsen.com/tab/product\\_families/nielsen\\_claritas/prizm](http://en-us.nielsen.com/tab/product_families/nielsen_claritas/prizm).
- [2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In J. B. Bocca, M. Jarke, and C. Zaniolo, editors, *Proceedings of 20th International Conference on Very Large Data Bases, September 12-15, 1994, Santiago de Chile, Chile*, pages 487–499. Morgan Kaufmann, 1994.
- [3] J. Bennet and S. Lanning. The netflix prize: Kdd cup and workshop. 2007.
- [4] R. P. Bhatt, V. K. Narayanan, R. Parekh, and X. S. Wan. Feature value recommendations for advertisement campaign performance improvement. *U.S. Patent Application*, (Docket: 12729/608, Reference: Y05590US00), 2009.
- [5] A. Broder, M. Fontoura, V. Josifovski, and L. Riedel. A semantic approach to contextual advertising. In *Proceedings of the 30th Annual International ACM SIGIR Conference*, pages 559–566, 2007.

parameters	metrics	$N = 1$	$N = 2$	$N = 3$	Aggregate
X=.75,Y=1.25	num. recs.	3034701	1317536	588179	3021865
	fraction good	0.815	0.899	0.962	0.799
	average ctr	$x\%$	1.06 $x\%$	1.13 $x\%$	0.97 $x\%$
	average lift	2.310	2.610	2.811	2.367
X=1,Y=1.25	num. recs.	2153122	871625	365276	2101694
	fraction good	0.776	0.871	0.995	0.838
	average ctr	1.09 $x\%$	1.17 $x\%$	1.27 $x\%$	1.07 $x\%$
	average lift	2.774	3.196	3.496	2.879
X=1.5,Y=1.25	num. recs.	1252700	463150	176556	1193620
	fraction good	0.884	0.948	0.992	0.862
	average ctr	1.27 $x\%$	1.34 $x\%$	1.48 $x\%$	1.25 $x\%$
	average lift	3.704	4.345	4.914	3.858

Table 7: Sensitivity of complementary segments recommendations to varying the parameter  $N$ .

parameters	metrics	$Y = 0$	$Y = 0.5$	$Y = 1$	$Y = 2$
X=.5,N=3	num. recs.	1011961	823717	269127	29755
	fraction good	0.975	0.976	0.980	0.984
	average ctr	$x\%$	$x\%$	1.15 $x\%$	1.46 $x\%$
	average lift	2.08	2.07	2.47	3.16
X=.75,N=3	num. recs.	594774	489531	177573	21845
	fraction good	0.964	0.965	0.972	0.979
	average ctr	1.15 $x\%$	1.15 $x\%$	1.31 $x\%$	1.61 $x\%$
	average lift	2.81	2.77	3.16	3.79
X=1,N=3	num. recs.	386961	317680	121625	16661
	fraction good	0.955	0.956	0.964	0.971
	average ctr	1.31 $x\%$	1.31 $x\%$	1.46 $x\%$	1.69 $x\%$
	average lift	3.61	3.56	3.98	4.44
X=1.25,N=3	num. recs.	275576	225321	87903	12985
	fraction good	0.951	0.952	0.958	0.972
	average ctr	1.38 $x\%$	1.46 $x\%$	1.54 $x\%$	1.85 $x\%$
	average lift	4.45	4.39	4.92	5.16
X=1.5,N=3	num. recs.	208891	170024	66421	10333
	fraction good	0.950	0.950	0.954	0.969
	average ctr	1.54 $x\%$	1.54 $x\%$	1.69 $x\%$	2.00 $x\%$
	average lift	5.33	5.28	6.03	5.95

Table 8: Sensitivity of complementary segments recommendations to varying the parameter  $Y$ . In order to protect proprietary data, we only provide relative CTR numbers.  $x$  denotes a reference value

- [6] K. L. Chang and R. Parekh. Complementary user segment analysis and recommendation in online advertising. *Patent Application*, (ID: 09-5466, Reference: Y05466US00), 2009.
- [7] Y. Chen, D. Pavlov, and J. Canny. Large-scale behavioral targeting. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 209–218. ACM, New York, USA, 2009.
- [8] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave the information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.
- [9] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 50–57, 1999.
- [10] Y. Koren, R. Bell, and C. Volinsky. Factorization techniques for recommender systems. *IEEE Computer*, (August):42–49, 2009.
- [11] J. Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–150, 1991.
- [12] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, (January-February):76–80, 2003.
- [13] J. Parikh, V. K. Narayanan, R. P. Bhatt, X. S. Wan, and R. Parekh. Profile recommendations for advertisement campaign performance improvement. *U.S. Patent Application*, (Docket: 12729/623, Reference: Y05590US01), 2009.
- [14] F. Provost, B. Dalessandro, R. Hook, X. Zhang, and A. Murray. Audience selection for on-line brand advertising: Privacy-friendly social network targeting. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 707–716. ACM, New York, USA, 2009.
- [15] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the Conference on Computer Supported Cooperative Work*, pages 175–186, 1994.
- [16] J. Yan, N. Liu, G. Wang, W. Zhang, Y. Jiang, and Z. Chen. How much can behavioral targeting help online advertising? In *Proceedings of the 18th International World Wide Web Conference (WWW 2009)*, pages 261–270, 2009.