Automated Audience Segmentation Using Reputation Signals

Maria Daltayanni*
University of San Francisco
mariadal@cs.usfca.edu

Ali Dasdan†
KD Consulting
Saratoga, CA, USA
alidasdan@gmail.com

Luca de Alfaro
UC Santa Cruz
luca@ucsc.edu

ABSTRACT

Selecting the right audience for an advertising campaign is one of the most challenging, time-consuming and costly steps in the advertising process. To target the right audience, advertisers usually have two options: a) market research to identify user segments of interest and b) sophisticated machine learning models trained on data from past campaigns. In this paper we study how demand-side platforms (DSPs) can leverage the data they collect (demographic and behavioral) in order to learn reputation signals about end user convertibility and advertisement (ad) quality. In particular, we propose a reputation system which learns interest scores about end users, as an additional signal of ad conversion, and quality scores about ads, as a signal of campaign success. Then our model builds user segments based on a combination of demographic, behavioral and the new reputation signals and recommends transparent targeting rules that are easy for the advertiser to interpret and refine. We perform an experimental evaluation on industry data that showcases the benefits of our approach for both new and existing advertiser campaigns.

CCS CONCEPTS

• **Information systems → Online advertising**; **Reputation systems**;

KEYWORDS

Reputation, Audience Segmentation, Expert Crowds

ACM Reference Format:
Maria Daltayanni, Ali Dasdan, and Luca de Alfaro. 2018. Automated Audience Segmentation Using Reputation Signals. In *KDD '18: The 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August 19–23, 2018, London, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3219819.3219923

*This work was completed while author was at Turn Inc (now Amobee Inc).
†This work was completed while author was at Turn Inc (now Amobee Inc).

KDD '18, August 19–23, 2018, London, United Kingdom

ACM ISBN 978-1-4503-5552-0/18/08…$15.00
https://doi.org/10.1145/3219819.3219923

1 INTRODUCTION

"Whom should I show my ads to?" This is the question that every advertiser has to answer before starting a new advertising campaign [16]. Typically advertisers convey their audience selection requirements with demand side platform (DSP) companies like Amobee or Rocketfuel via Boolean expressions on user *demographics*, e.g., all males in California, or *technographics*, e.g., all users that use Chrome Web Browser. The DSPs are responsible for identifying optimal targeting and bidding on advertising opportunities within the audience space as defined by the broader advertiser selections. While the targeting and bidding processes are machine learning driven, based on market research and history data, the final audience selection phase pertains to translating the learning into user segments to target. Advertisers come with certain initial targeting rules; then they allow the campaigns to run and they refine their audience selection preferences according to the observed results.

Although the use of explicit Boolean constraints ensures control and transparency in the audience selection process for the advertiser, often advertisers have to manually determine the definition of their final targets. Learning from behavioral signals is not directly translatable to user segments, which inevitably becomes a source of inefficient choices. While most advertisers research the market and can adequately describe the audience that are more likely to engage with their campaigns, they usually lack the data or the skills to express their preferences as Boolean expressions over user demographics and behavior and build user profiles [10]. For example, an advertiser may be aware that the users of interest are non-tech-savvy individuals. However, he is probably unaware that a way to describe a superset of such users is by selecting people that use an outdated version of the default browser in their operating system, i.e., Internet Explorer on Windows or Safari on Mac. A suboptimal audience selection negatively affects the success of an advertising campaign in various ways: a selection that is too narrow may dramatically limit the reach of a campaign, while a very broad selection will make the optimal bidding problem intractable. Also, the usage of behavioral data may help identify users likely to respond to a campaign. However, identifying which piece of historical data contributed to the conversion is often information that includes noise and bias. That makes it even harder to describe user profiles of high convertibility.

Several attempts have been made to bridge the gap between the advertiser domain specific knowledge and the language perceived by the advertising systems. In most cases, targeting solutions include the use of proprietary data such as user browsing behavior, search history or emails [3, 4, 11, 13, 19] and as such, they can only be leveraged by companies that have access to this data. Even if access to proprietary data was granted, two issues arise. First, traditional solutions suggested in the above papers, would not be as useful to advertisers that want to retain control over the audience selection with transparent rules, since they usually provide a complex, not interpretable function that determines whether a user belongs to the ad audience or not, in a binary fashion. Second, audience segmentation seems to contain multiple np-hard graph clustering problems as subproblems. Hence, its complexity makes it hard to solve with traditional machine learning methods.

In this work we propose a mechanism that builds target user segments to recommend for a particular ad campaign, based on three input signals; demographic data, behavioral data (these two are typically available to a DSP) as well as reputation, a new type of signals that our model produces to represent relevance quality between users and ads. Based on user demographics and past behavior, our reputation system produces a score for each side (users and ads) in an iterative procedure of learning, thus removing noise and bias of the original signals. Then the three types of signals are used to devise a user segmentation that is described by transparent rules and suggested to the advertiser for his campaign.

The data that are typically available to a DSP are ad impressions, user conversions and user demographic and technographic data. The output of our method is binary Boolean expressions on user data that are transparent and easy to communicate with advertisers. The basic components of our solution are a reputation system that we build upon users and advertisers and decision tree learning on past user conversion data. The proposed reputation system uses link analysis on the bipartite graph between users and advertisers (ads) and it yields: (a) an individual score per user that reflects the advertisers aggregated belief about their targeting quality, that is, whether the advertiser should target them or not, and (b) a score per advertiser reflecting his targeting quality, i.e., his performance in identifying converters. Our reputation system builds upon the HITS [14] algorithm; in our context, we are interested in finding how many advertisers a user was targeted by, along with the advertiser's quality measured by his success in identifying converters in the given content category. The content taxonomy may introduce a level of noise, which we partially address by removing fraud and taxonomy inconsistencies before the model is run.

Given the reputation signals just learned, we derive audience selection rules by using decision trees on reputation and user historical conversion data. During the learning process, we define the target variable to be proportional to advertiser's Return On Investment (ROI). Our main features are a) user demographic data (age, gender, and more), b) technographic data (operating system, browser type, net connection speed, and more), c) behavior data (conversions on past ads) and d) the new computed reputation scores. To illustrate how user segments are extracted from trees regressed on our data, we provide a simple example of demographics in Figure 1. Our method also derives non favorable user segments, thus yielding *negative* recommendation criteria, that is, which users the advertisers are advised *not* to target.

In our experiments, we evaluate our proposed audience selection mechanism in a real-world dataset provided by Amobee. The results show that new advertisers that adopt our recommendations can enjoy an increase of up to 450% in conversion rate compared to advertisers that stick to their own rules. Advertisers with existing campaigns can also use our algorithm to refine their audience selection preferences and identify segments that have up to 10x higher conversion rate than the average of their campaigns. Finally, we show that advertisers can potentially increase the conversion rates of their ads if they adopt our negative selection recommendations.

The remainder of the paper is organized as follows. In Section 2 we present the notation and problem statement of this work. In Section 3 we describe the *Auto-Segmentation* algorithm, our proposed algorithm for audience selection. In Section 4 we present a set of experiments which illustrate the effectiveness of our models over baselines. In Section 5 we describe related work in audience selection and reputation systems. Finally, in Section 6 we conclude this study.

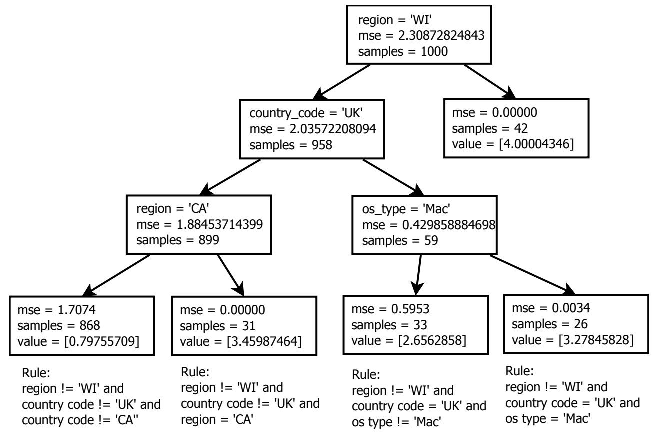

Figure 1: A sample regression tree. Example segment/rule: region ≠ 'WI' & country code = 'UK' & os type = 'Mac', mean(ROI) = 3.27

Table 1: Advertiser Categories

Category	Subcategory 1	Subcategory 2
Telecommunications	Wireless	TV
Autos	Cars	Commerical Trucks

2 NOTATION AND PROBLEM STATEMENT

We consider sets of *line items* L, *advertisers* A and *users* U. Each advertiser $a \in A$ owns one or more line items, denoted as $l_a \in L_a \subseteq L$, that he advertises to users. Note that an advertising campaign usually includes multiple line items, where each line item provides more precise focus on a particular set of audience segments, goals, budgets, etc. Advertisers belong to a specific content *category* $q \in Q$, where Q is a set of categories that describes the content of the advertiser's business domain, such as autos or telecommunications. A sample of categories is shown in Table 1. In fact, advertisers are categorized across a taxonomy, but for the scope of this work we consider the lowest level in the taxonomy as a distinct category. For example, if node "credit cards" has two children, "loans" and "mortgages", we consider the two categories, "credit cards - loans" and "credit cards - mortgages".

Table 2: User features

Field	Domain	Type
age	[19,99]	numeric
gender	{male, female}	nominal
oper.system	{ Windows, ..., Mac }	nominal
country	{Albania,...,Zimbambwe }	nominal
action (conversion)	{True, False}	Boolean
reputation score	[-1,+1]	float

Users *click* or *view* a line item; we define click and view functions: $c(u,l) = \mathbf{1}(\text{user } u \text{ clicked line item } l)$, $v(u,l) = \mathbf{1}(\text{user } u \text{ viewed line item } l)$, where $\mathbf{1}$ is the indicator function. We denote users who clicked line item l as $U_l = \{u \in U : c(u,l) = 1\}$ and users who viewed line item l with $V_l = \{u \in U : v(u,l) = 1\}$. Users perform *actions* after viewing and clicking a line item, which may reflect product purchases, subscriptions, friend invitations, and more. We denote number of actions of user u during time period dt after viewing/clicking ad of line item l, as $\alpha(u,l,dt) \in \mathbb{N}$. Interesting quantities related to actions are the *action rate*, computed as percentage of a user's actions over clicks or impressions, and the *action density*, computed as the percentage of actions within a time period over the actions performed during a wider study period. These are defined and used in Section 3.3 to describe segment feasibility and optimality. Note that although the time parameter is naturally involved in the actions definition, it is omitted from the impression (view) and click definitions, for simplicity.

User *features* F are defined as functions of users, with $f \in F$ and $f : U \to D_f$, where $d_f \in D_f$ is a value in the domain of values D_f for a particular feature f. Based on the feature, $D_f = \mathbb{R}$ if f = income, $D_f = [19, 99]$ if f = age, D_f = {male, female} if f = gender, $D_f = \{True, False\}$ if f = a past purchase or friendhsip action. Examples of features are shown in table 2. *Rule-set* $\mathcal{R}$ is defined as the set of possible feature key-value pairs along with conjunctive and disjunctive expressions of them. Keys are related with values with any of the comparison operators $op = \{<, =, \neq, >, \leq, \geq\}$ where applicable. Note that we use only the $=, \neq$ operators for nominal fields. Given the above, we define rule-sets as formulas $\mathcal{R}$ for which the following hold:

$$\begin{aligned} f \in F, d_f \in D_f, \bowtie\ \in op \to f \bowtie d_f \in \mathcal{R} \\ \phi_1, \phi_2 \in \mathcal{R} \to \neg\phi_1, \phi_1 \vee \phi_2, \phi_1 \wedge \phi_2 \in \mathcal{R} \end{aligned} \tag{1}$$

Also, we consider $\mathcal{P}(U)$, powerset of users U, that is the set of all possible subsets of U. User *segment* $S \in \mathcal{P}(U)$ is defined as the set of users described by rule $r \in \mathcal{R}$: $S = \{u : r(u) = \text{True}\}$. *Reach* of segment S with respect to a line item $l \in L$, is the size of the segment, that is, the number of users targeted for the line item's campaign,

$$\text{reach}(S,l) = |V_l|_{u \in S} = |\{u \in S : v(u,l) = 1\}| \tag{2}$$

Return of a campaign is defined as the gain (campaign's gross profit) from the investment minus the cost of investment, as specified by the budget planned by the advertiser for a line item. *Cost* of a campaign includes targeting cost, data cost and several other fees. Targeting cost refers to the cost for reaching users, mainly through the Demand Side Platform (DSP) such as Amobee, and the publisher. Data cost pertains to the cost of consuming user feature data for targeting; since most of the user attributes are offered by third-party providers, their availability incurs additional fees for the advertiser. For simplicity, we do not provide further details about how these quantities are broken down. In this work we consider Cost-Per-Action (CPA) campaigns, where the advertiser pays for each observed user action (the type of actions are specified by the advertiser). The Return On Investment (ROI) function $g : U_L \times L \to \mathbb{R}$ is defined on the set of line items $l \in L$ clicked by users $u \in U_L$ within a given study period,

$$g(u,l) = \frac{return}{cost} \tag{3}$$

Note that in the above definition ROI is computed in user level, that is, return pertains to user action gain. Also, g is a function of line items, users, and *time*; however in the scope of this paper we assume that the time period of our study is fixed and the candidate audiences will not change during the experiments. Using the simplified expression, we consider the following problem:

Given a history of user activity, and a set of initial features about users, derive rules to describe user segments that optimize return over cost, given that their reach size meets a minimum threshold.

History data of activity involve a set of users (clickers $U_L = \bigcup_{l \in L} U_l$ or viewers V_L) targeted for a set of line items L that are owned by one or more advertisers in A, along with the ROI of each user - line item interaction, $g(u,l), \forall u \in V_L$ (or U_L), $\forall l \in L$. The input space granularity may be that of an entire advertisers category q, in case we are interested in recommendations for a new line item $x \notin L$ owned by advertiser in category q, or that of one or more campaigns of a particular advertiser. The features data involve values $f(u) \bowtie d_f, \forall f \in F, d_f \in D_f, u \in U_L$, of demographic, techno-graphic, and other features of users available from their behavior history. The rules to be derived are described as rules $r \in \mathcal{R}$ that describe feasible segments of users $S_r \subseteq U$. We notice that g is defined to take values in $\mathbb{R}$. However, our ROI maximization is bound by a) the budget that is invested by the advertiser for line item x, and b) by the total return that is earned at bid win. Bid win is affected by the bid prices that hold during the time of study in the specific market. (Feasibility is defined in Section 3.3.1.) With return on investment optimization, we aim at identifying segments of users who have high probability of conversion (action) upon viewing x's ad. At the same time, we prioritize large size segments such that reach is considerable for the advertiser. Finally, we introduce the pass/fail testing of feasibility and negativity about segments, which reflects whether the segments to recommend are meaningful for the advertiser. For example, an airline that flies only in Europe would not be as interested to learn the best user segments in the US.

The optimization aims at satisfying three major priorities for advertisers; high ROI, high user reach, and high confidence. The first two are discussed above. Confidence is regarded with respect to reliability in prediction behind any targeting recommendation. That may evolve from the number of examples used to learn our predictions, from the prediction accuracy itself, as well as from the reputation quality of the examples being used. For example, if a user has converted in the past, was that a random occurrence, or was it a meaningful response that reflects longer term human interest? Since at the moment of recommendation we have not seen the actual return for the new line item x yet, the confidence component is captured by $E[g(u, x)]$, the expectation about the unobserved ROI for x. As we discuss in the algorithm Section, we address confidence by ensuring considerable number of examples in the input, and by introducing a reputation system about users and advertisers to capture reliability in the user interest and user targeting signals.

3 AUTO-SEGMENTATION

In our proposed approach, we devise an algorithm that derives a set of rules that describe a set of optimal segments (in terms of user conversion) for the given input data. The applications of Auto-Segmentation in this paper are: a) campaign refinement and b) cold start. In the first case, the goal is to produce refined user segments based on the advertisers existing segment selections, with respect to user conversion; the input of the algorithm is the set of line items of a particular advertiser along with his targeted user segments, the user features, and the ROI performance in each user-line item relation. In the second case, the goal is to recommend rules that describe optimal segments for a new campaign line item for which only business context is know, that is, information about its type and content (for example, line item of an airlines company that flies in Europe); the input of the algorithm is the new line item, the line items of advertisers in the same business taxonomy of the new line item owner, along with the user segments selected by the relevant advertisers, user features and ROI performance per (user, line item).

Our main algorithm (Algorithm 1), consists of three stages. In the first stage (steps 1 - 2), a reputation system is used to identify consistent and reliable responders to popular line items in the category. We build a bipartite graph from the input elements and we run the weighted HITS algorithm, *wHITS*, as described in Section 3.1. Our focus in this stage is to first look at the user preferences (line items clicked or to which the user converted) along with the return on investment as an implicit indication about the interest strength of the user. Then we intend to compute quality of these preferences using a reputation system that is based on a weighted version of the HITS algorithm [14] that accounts for ROI-based weighting along with popularity of items. Along with line item reputation, the reputation system also computes a bias/reliability score for users, to reflect a measure of quality for their judgements / click decisions.

In the second stage (step 3), we learn candidate user segments, by using weighted decision trees on the input; our input consists of the user features, including demographics, behavioral and reputation signals learned in the first stage of the algorithm. User features are used as variables and value domains are used as possible split points, while the target is set as the ROI at user-line item level, weighted by the crowd computed reputation and reliability scores. Tree decision learns the distribution of the weighted ROI success metric as a function of the user features; then the highest metric score leaves are used as candidate segments for recommendation. The decision tree algorithm *DecTree* is a variation of the CART [12] algorithm, as described in Section 3.2.

In the third stage (steps 4 - 5), we apply segment selection out of the suggested candidate segments from the second stage. Candidate segments are filtered by an optimality function that determines segment performance in terms of the used success metric (such as conversion rate). In addition, negative recommendations are derived in this stage, where segments advised *not* to target are provided as well. *SelectSegments* is described by Algorithm 4 in Section 3.3.

Algorithm 1 Auto-Segmentation

Input: Advertisers A, line items L, users U_L, ROI $g(U_L, L)$, features $f(U) \bowtie d_f$, [new line item $x \notin L$ in cold start case]
Output: Rules $\{r \in \mathcal{R}\}$, segments $\{S_r\}$ [for new line item x]
1: $G = (U, L, E) \leftarrow$ build graph from $A, L, U_L, g(U, L)$
2: $z(U), b(L) \leftarrow$ wHITS(G) ▷ Reputation
3: $\{s_i\} \leftarrow$ DecTree$(U, [f(U) \bowtie d_f$ and $z(U), b(L)], g(U, L))$ ▷ Candidate Rules Extraction
4: $\{s_{best}\} \leftarrow$ SelectSegments$(\{s_i\})$ ▷ Segment Selection
5: **return** s_{best}

3.1 User Reputation

In the first stage, we create a reputation system that estimates the quality of the items preferred by users, on top of which items they prefer, to improve accuracy in representing user interest. A signal about a converter's reliability comes from the user's target activity. Activity is allowed for with respect to, at what frequency does the user respond (popularity) when targeted, and what is the gain obtained for the advertiser (ROI) from the user's response. Since in this work we account for ROI with focus on CPA campaigns, gain refers to user actions. To derive a *conversion reliability* score for the users, we make use of the above signal as follows.

We devise a link analysis algorithm on a graph that connects users with their clicked line items, owned by a set of advertisers. An edge between a user and a line item holds the information that the corresponding advertiser targeted that user and the information of whether the user responded, along with his interest strength. The latter is measured by the total return from the response over to the total cost for the advertiser targeting this user. We use the ROI weight on the edge also to cover the information about the *targeting value* of the advertiser. Then we apply a mutual recursion computation of scores between line items and users as shown in steps 4 - 5 of Algorithm 2. That recursion attributes to the user not only the number of his clicked line items weighed by the ROI that followed from the corresponding actions; it also attributes to the user the quality of the line items that he chose, in terms of their popularity. If a line item was chosen by many users, that indicates that the ad was successful, and also that the targeting was successful, that is the right people were reached for a product of their interest. If, in addition, the total ROI for the related actions of those clickers was high, including the original user, that validates the monetary gain from the line item - user targeting relation.

Formally, we consider bipartite graph $G = (L, U_L, E)$, as shown in Figure 2, where L is the set of line item nodes owned by advertisers A within category q, U_L is the set of user nodes, for users who clicked or viewed the line items in L, and E is the set of edges between L and U, denoting the targeting relation between them; an edge $e = (l, u)$ exists if user u clicked on line item l posted by an advertiser $a \in A$. Each edge is accompanied with a weight that

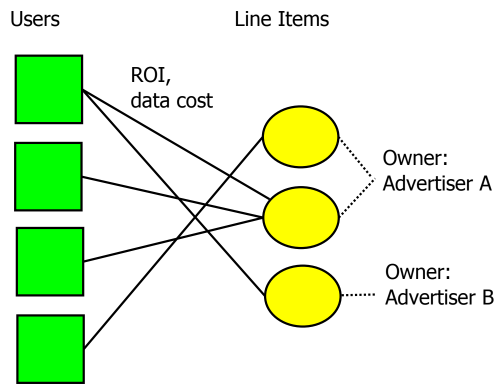

Figure 2: Reputation Bipartite Graph between users and advertisers

characterizes the success of the targeting relation, measured by the ROI of the action following the click, $g(u, l)$. We run *wHITS*, a weighted version of the HITS algorithm [14] where we update the scores of each node by the linked nodes scores weighted by the ROI relation between line item and user nodes. In particular, if a user has clicked on a line item and that click was followed by an action, the degree of success for that action measured for the advertiser is taken into account. In this way, the scores of user nodes reflect not only the user's activity popularity, that is how many line items they clicked on, but also the value of their click as measured by the advertiser. In the long run, a user with high score ends up representing a reliable converter, that is, whose clicks and actions are rather not random events but they imply true interest in the line items clicked. At the same time, a line item with high score represents a popular component of the targeting value of its advertiser. Note that convergence is guaranteed as the quantities in steps 4, 5 are positive [14].

Algorithm 2 wHITS

Input: $G = (L, U_L, E)$
Output: Reputation scores $z(u), \forall u \in U_L$, $b(l), \forall l \in L$
1: Initialize $z(u) = 1, \forall u \in U_L$; $b(l) = 1, \forall l \in L$
2: **repeat**
3:     **for** $u \in U_L, l \in L$ **do**
4:         $z(u) \mathrel{+}= g(u, l) \cdot b(l)$
5:         $b(l) \mathrel{+}= z(u)$
6:     **end for**
7:     Normalize $z(u), \forall u \in UL$ and $b(l), \forall l \in L$
8: **until** convergence

3.2 Rule Extraction

In the second stage of the algorithm, we learn candidate user segments for the given input, by using weighted decision trees. In particular, we build a decision tree, DecisionTree, based on a variation of the CART algorithm [12] using as features the user demographic, technographic and behavioral features along with the computed reputation and reliability scores of users and line items from the previous stage, obtained from Algorithm 2. In traditional CART, for different feature-value pairs we successively split the user space into two partitions, aiming for optimal splitting at each level. Optimal split is considered in terms of node impurity computed by the error on the average ROI of the ads clicked by the users in each partition. Then we iterate until a minimum number of samples is left on each sub-region. Below we show our varied version of the algorithm, with the addition of weights in the splitting process. We consider set of users $u \in U$ as the set of inputs, and user features $f \in F$ as the set of variables with splitting points d_f. We also add reputation scores of line items and reliability scores of users as *new features* whose domain of splitting points is the range of scores computed from Algorithm 2. We also consider the response variable $\gamma(u)$, where $\gamma(u)$ is the average ROI of the ads that user u has clicked on.

$$\gamma(u) = \operatorname*{avg}_{l:c(u,l)=1} [g(u, l)] \tag{4}$$

Then, for each splitting variable f and split point d, we consider regions $R_1(f, d) = \{u : f(u) \leq d\}$ and $R_2(f, d) = \{u : f(u) > d\}$, if f is numeric, or $R_1(f, d) = \{u : f(u) = d\}$ and $R_2(f, d) = \{u : f(u) \neq d\}$, if f is nominal. We select the best $\langle f', d' \rangle$ pair for which:

$$(f', d') = \operatorname*{argmin}_{f,d} \left[\sum_{u \in R_1(f,d)} (\gamma(u) - \bar{\gamma}(R_1(f, d)))^2 + \sum_{u \in R_2(f,d)} (\gamma(u) - \bar{\gamma}(R_2(f, d)))^2 \right] \tag{5}$$

In the above expression, we set $c_1 = \bar{\gamma}(u)$ and $c_2 = \bar{\gamma}(u)$ and we use $\gamma(u)$ which reflects the weights of the line items, $b(l), \forall l \in L$. Note that the possible values considered for the splitting points d by default are decided based on the existing values in the data. The process is repeated until a minimal number of samples is reached on each of the sub-regions.

To give an example of splitting across user features, consider pair $\langle$age, 35$\rangle$, which divides the space of users into users whose age is less than 35 (class k_1 of 19-34 year old users) and users with age equal to or higher than 35 (class k_2 of 35-99 year old users). In an ideal scenario, region R_1 would contain users with total $\gamma = 0$, while R_2 would contain users with total $\gamma = 100$, assuming that the max value or ROI is 100. That would split the users in pure nodes of low and high ROI values, which would easily suggest candidate region R_2 as a segment to consider for recommendation. Such a performance depends on the data though, hence extracting top ROI segments from the tree regressor does not suffice to represent suitable segments. In Section 3.3 we describe our approach for selecting segments to recommend in a personalized way for advertisers. A benefit from using regression trees in the above model is the fact that trees automatically yield segments where high ROI is concentrated across the multiple user selections that the advertisers in category q have made. In particular, the best performance leaf nodes become good segment candidates among which we can select recommendations for the new ad, x. What is more, the structure of trees automatically yields the rule sets to describe the users of the suggested segments, by taking one or more paths across its nodes. For example, a candidate rule set in the example shown in Figure 1 would be region ≠ 'WI' and country code = 'UK' and os type = 6 (Mac). A single path in the tree consists of feature-value pairs related with conjunction (*and*), as in the example, while multiple tree paths may be combined in a disjunction (*or*) relation to broaden the users included for the recommended targeting.

Algorithm 3 DecTree

Input: $U, f(U) \bowtie d_f, g(U, L), z(U), b(L)$
Output: Candidate user segments $\{s_i\}$

```
for all u ∈ U do Compute γ(u) according to Equation 4
end for                                        ▷ Average user ROI
for all (splitting variables f, splitting points d_f) do
    Find (f′, d′) that minimizes Equation 5
end for
R1, R2 ← Divide user space in 2 regions according to (f′, d′)
{s_lft} ← DecisionTree(R1, f(R1) ⋈ d_f, g(R1, L), z(R1), b(L))
{s_rgt} ← DecisionTree(R2, f(R2) ⋈ d_f, g(R2, L), z(R2), b(L))
if |R1| ≤ min_samples then return s_lft
end if
if |R2| ≤ min_samples then return s_rgt
end if
```

3.2.1 Boosting for Feature Selection. In our domain, the set of features used in the input, play an important role in extracting optimal segment rules. Applying feature selection is expected to improve the accuracy of our model prediction and yield better segments. It is well known that trees are powerful in representing the structure of multiple feature data and in supporting complex functions. However, their predictive capability is limited. Boosting trees achieve improved predictive accuracy by training multiple single tree weak predictors and aggregating single predictions for best performance.

We tried to vary the features in the input of DecisionTree in Algorithm 3. In particular, we used boosting trees on our entire dataset to extract feature importance scores. Then we retained only the highest importance score features and provided them as input to DecisionTree. We found that the prediction accuracy of our final tree was improved for certain advertiser categories, as the targeting rules that were extracted, yielded equal or better conversion rate.

3.3 Segment Selection

In this Section we describe our approach for selecting segments out of the pool of candidate segments derived from the decision tree model. Usually advertisers determine their targeting based on two criteria; user response and hard constraints. The performance criterion dominates the literature interest, where the probability of user response is studied. The second criterion pertains to custom hard constraints set by the advertisers to achieve brand advertising or for individual campaign interest goals. For example, a European airline company may always want to target all Europeans even if 90% of their ad responses come only from UK citizens. Although hard constraints are not easy to describe, unless specifically specified by the advertiser, they could be studied, for example, by looking at common targeted users across several line items of an advertiser. In this work, our goal is to recommend segments with optimal expected response performance, leaving hard constraints as secondary priority. First, we describe the criteria that determine *feasibility* and *optimality* of segments, and next we propose a method to recommend both which segments to include in targeting based on these criteria, and which segments to exclude from targeting that the advertiser is probably already using.

3.3.1 Segment Feasibility. With user segment *feasibility*, we aim to capture a confidence level about the recommended users response expectation. We set two main criteria to define feasibility; segment applicability and user activity. First, segment recommendation has to be applicable in the advertiser's domain; for example, recommending a truly high response segment of US users to an airline company that flies only in Europe is not very meaningful, as the expectation is that these users will depart from/land in the US. Hence the European company would only be interested in case it had an alliance with some US-flying airline that extends its network. To address applicability, we prioritize segments which overlap to some extent with the existing targeting selections of the advertiser. Note that our focus is in adjusting the targeted population of an advertiser towards best performance, rather than extending it with similar audiences. Second, the time parameter is important for recommending segments. For example, a user who recently booked an airline ticket, will probably not be interested in buying another one to the same destination within the next months. Hence, usually active users with long inactivity by the study time, are more eligible to get recommended as they are expected to respond relatively soon. We define feasibility as follows:

Definition *feasible(s,l)*: Segment $s \subseteq U$ that is candidate for recommendation at time t to advertiser a who posted line item l and had past targeted user segments $s_a \in S_a \subseteq \mathcal{P}(U)$, is called *feasible* for a, if $s \bigcap s_a \neq \emptyset$, and $\frac{\sum_{u \in s} \alpha(u,l,[t-\sigma,t])}{\sum_{u \in s} \alpha(u,l,[0,t])} \leq \epsilon$, given that $reach(s_a) > \eta$ and $reach(s) > \eta$, for $\eta \in \mathbb{N}$, $\epsilon \in [0,1]$, $\sigma \in [0,t]$ constant parameters.

Feasibility of segment s with respect to line item l is defined as the indicator function with condition $feasible(s,l)$, according to the above definition. Note that $\frac{\sum_{u \in s} \alpha(u,l,[t-\sigma,t])}{\sum_{u \in s} \alpha(u,l,[0,t])}$ represents action density with regards to line item l within time period $[t-\sigma, t]$. Also, η is determined by the traffic observed within the category of the advertiser, and σ represents the level of *recency* under study. The recency threshold is determined by the line items content; for example, recency in airline tickets may refer to months, while recency in autos may refer to years. Also note that feasibility partially covers the advertisers hard constraints, along with user response performance, since the overlap with past defined segments entails some confidence about the advertisers interest in the recommended users. Unfortunately, it is very hard to extract or simulate the hard constraints of the advertisers targeting and use them for extensive feasibility testing, since this data is not provided by the advertiser nor any other party.

3.3.2 Segment Quality. With user segment *quality* we aim to capture the quality of recommendation, with regards to the expected user response. In particular, we regard quality segments as those which include the users that are expected to show the highest response among all the clickers or viewers of the advertiser's line item ads. Formally, we use the following definition:

Definition *quality(s,l)*: Segment $s \subseteq U$ that is candidate for recommendation at time t to advertiser a who posted line item l and had past targeted user segments $s_a \in S_a \subseteq \mathcal{P}(U)$, is a *quality* segment, if $E[cta(s)] > \text{avg}_{\, s_a \in S_a} E[cta(s_a)]$ and $\sum_{u \in s} \alpha(u, l, [0, t]) \geq \zeta$, given that $reach(s_a) > \eta$ and $reach(s) > \eta$, for $\eta \in \mathbb{N}$, $\zeta \in \mathbb{N}$, constant parameters.

Quality of segment s with respect to line item l is defined as the indicator function with condition $quality(s, l)$, according to the above definition. Note that $E[cta(s)]$ reflects the expected action conversion rate (known as Click-Through-Action) of segment s, that is the ratio of actions over clicks by the time of recommendation, t, $cta(s) = \frac{\sum_{u \in s} \alpha(u,l,[0,t])}{\sum_{u \in s} c(u,l,[0,t])}$. Parameter η is determined by the traffic observed within the category of the advertiser. Parameter ζ reflects action *frequency* as an infimum of actions that must be observed from the recommended users by the time of recommendation t. Again, the frequency threshold is determined by the content of the line items in the category; for example, frequency in airline tickets may refer to dozens, while frequency in bath products may refer to hundreds. Quality covers response performance, however hard constraints may not be covered; for example, in case the advertiser includes some users for brand advertising, the conversion rate within those users sub-segment is not expected to be high.

3.3.3 Who not to target. Along with recommending which users to target ("good" segments), it would be of great value to the advertiser to get an insight also about which users to *stop* targeting ("bad" segments). It is often useful to know which user segments perform worse than the average selections within an advertiser's target groups, or within the category's overall target population, so that future campaigns can be adjusted towards higher conversion. In our context, "bad" user segments reflect users who constantly do not respond to ads of that particular advertiser, or to ads of line items in the advertiser's category (for instance, autos line items). Hence, since our recommendation mainly optimizes user response performance, we tackle *negative* recommendations from that perspective. The advertiser may find negative targeting recommendations useful, or they may choose not to remove any targeted users as those may correspond to users that satisfy the advertiser's hard constraints. For example, if New Zealand local population do not respond to airline offers of a company that flies to New Zealand, the company may always want to be targeting that population, with the expectation that when they choose to travel, they will prefer that airline.

In this Section we study quality of segments in category level, that is which segments perform worse among users targeted within a given category, or in advertiser level, that is which segments perform worse among users targeted by an individual advertiser. In the former case, we propose negative recommendation rules 3.1 and 3.2 and in the latter case we propose rule 3.3. In both cases the recommendation is applied on the last stage of our algorithm. Consider set of users S who have clicked on line items of category q in the past, owned by n advertisers $a \in A$. Also consider segments $s_a \subseteq \mathcal{P}(S)$ that the advertisers have targeted in the past. For new advertiser in the category (that is, who has not targeted users in S yet) who wants to find which users not to target for a new line item campaign, choose "bad" segments among m candidate segments $s_i, 1 \leq i \leq m$, based on the following recommendations:

Recommendation 3.1 (Category-wise). *Do not target s_i, if $cr(s_i) < cr(S)$.*

Recommendation 3.2 (Category-wise). *Do not target s_i, if $cr(s_i) < \text{avg}_{\, 1 \leq j \leq n}\, cr(s_{a_j})$.*

For advertiser in the category who has targeted users from S in the past and who wants to find which users not to target for a new line item campaign, choose "bad" segments among m candidate segments $s_i, 1 \leq i \leq m$, based on the following recommendations:

Recommendation 3.3 (Advertiser-wise). *Do not target s_i, if $s_a \cap s_i \neq \emptyset$ and $cr(s_i) < cr(s_a)$.*

Note that most often, advertiser a creates more than one segments, however for simplicity and without loss of generality, we use s_a to denote any segment created by advertiser a. Finally, along with "bad" segments definition, we also consider confidence level for the negative recommendation, based on the ratio of the conversion rates under comparison in each of the above recommendations.

Algorithm 4 SelectSegments

Input: Line item x, candidate segments $\{s_i\}$
Output: Best user segment s_{best} for x

```
for all segments s_i ∈ {s_i} do
    λ(s_i) ← feasible(s_i, x)
    μ(s_i) ← quality(s_i, x)
    ν(s_i) ← if any of the recommendations 3.1, 3.2, 3.3 is True
end for
return s_best = argmax_{s_i} λ(s_i) · μ(s_i) · (1 − ν(s_i))
```

4 EXPERIMENTS

In this Section, our goal is to show that the proposed Auto-Segmentation algorithm solves the advertisers' cold start problem and that its solution for the campaign refinement problem outperforms baseline approaches. First, we show how our segment recommendations solve the cold start problem of new campaigns for which only their business context is known (such as their content category or taxonomy) by deriving better performing segments based on a category-wide input dataset, as opposed to advertisers original selections. Second, to test how auto-segmentation contributes to campaign refinement via automatic rule derivation, we compute conversion rates of three types of segments; baseline segments defined by the original advertiser's selections for a campaign, segments recommended by our basic model computed on the campaign's data, and segments recommended by our reputation-based model on the same data. The basic version of our model implements steps 3 to 5 of algorithm 1, using only basic features about users (demographic, techno-graphic and behavioral), which mainly describe his online past behavior. The reputation system version of our model implements the entire algorithm and includes reputation and reliability scores as new features for rule extraction. These

Table 3: Feature importances across categories

Category	Features (Importances)
airlines	region:MA (0.12), region:FL (0.09), city:Buffalo (0.08)
autos	os type:6 (0.3), os type:4 (0.15), age(0.07)
hotels	age (0.19), country code: US (0.15), region:CA (0.05)
insurance	os type:1 (0.13), browser type:3 (0.08), age (0.07)

scores represent the signal about advertiser's user targeting; their contribution shows the value of integrating advertiser with user oriented signals. Finally, we show how negative recommendations benefit targeting in an offline training-testing experiment. Knowing which users not to target within a candidate segment saves investment, time and data cost resources for the advertiser.

4.1 Dataset and Metrics

In our experiments we use a dataset that corresponds to 15 days of data obtained from Amobee Inc. After removing click fraud using external vendors and internal methods, the dataset includes $660M$ impressions, $170M$ distinct users, $7.5K$ line items, 150 advertiser categories, $1M$ actions. We use 20 organic user features, including demographic, locale, technographic (such as operating system and browser), and behavioral features (such as sites visited) and we use return on investment (ROI) as target. Table 3 shows the feature importances computed as the (normalized) total reduction of the mean squared error node split criterion across the tree nodes that is brought by that feature (Gini importance). A general observation is that of different attribute importances across different advertiser categories. For example, for category female clothing, age appears to be the most important feature, while for category autos, operating system and age appear as the most important features. To measure the performance of segmentation, we use a custom conversion metric for each segment defined as

$$cr(s) = \frac{\sum_{u \in s} \gamma(u) \cdot \mathbf{1}(\gamma(u) > 1)}{\sum_{u \in s} \gamma(u)} \quad (6)$$

where $\mathbf{1}()$ is the indicator function. The above expression captures the percentage of impressions in which the return on investment is positive over all impressions. Positive return on investment occurs when the total return from an advertiser's investment on an ad for a particular line item, is (equal or) greater than the total cost that the advertiser had to pay based on the type of his campaign for advertising the product. The most common types of campaigns are the per action payment basis, and the per click payment basis. In the current study we only account for the former. The intuition behind defining the above custom conversion rate lays on the fact that the actual gain for the advertiser occurs only when the user actually converts to an advertised product, either by signing up or purchasing or providing data such as in survey ads. In the current study our goal is to optimize user engagement. Other types of advertising, such as brand advertising, require other approaches for testing.

4.2 Cold Start

Advertisers manually select an initial audience for a new campaign, based on their first estimations about user interest (plan phase).

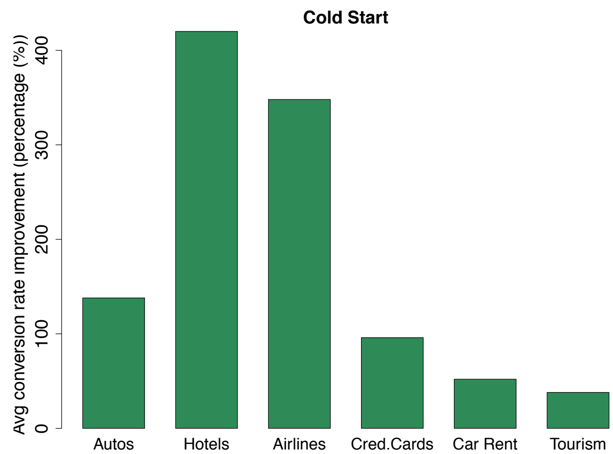

Figure 3: Cold Start

Then they start the campaign (execute phase) and after observing user behavior for a short time, they refine targeting based on recent activity high response groups (analyze phase). They repeat this process in several iterations until their targeting achieves the desired performance in terms of return on investment. Since that costs investment expenses and time, in this experiment we show how Auto-Segmentation provides audience recommendations which outperform the average expected performance of targeting selected in the cold start of a new campaign.

For a given advertiser category, such as "autos", we collect all advertisers campaigns and we run the Auto-Segmentation algorithm on the input of user activity; on the output segments we compute conversion and we compare it against conversion of the original segments targeted across the line items owned by advertisers in the category. In Figure 3 we show the percentage of average conversion rate improvement by the suggested segments versus the average conversion rate of original segments selected by advertisers within a particular period. We illustrate the results for several categories. Note that for the baseline we use the mean conversion rate of existing segments within a 15-day period, assuming that this represents the expected performance of a new campaign targeting in cold start. Both for anonymity but also because early targeting data availability is limited for most categories, we approach early segment performance by using the average performance of the segments. The recommended segments outperform the original segmentation selections for all categories.

4.3 Campaign Refinement

To show the quality of Auto-Segmentation recommended segments, we compare their conversion rates against rates on segments originally selected by advertisers. In particular, we split user data into training and testing sets; then we learn segments based on the training data and we compute conversion of the testing population that corresponds to each segment. Along with each segment's conversion rate, we also compute its reach, that is, the amount of users reached when each segment is targeted. Since advertisers are particularly interested in reach along with conversion, we compute conversion rate performance as reach is being increased, by taking weighted average of $reach \times cr$ across segments, described in the following:

First, we sort the M recommended segments $s_1, s_2, \ldots, s_M$ by decreasing conversion rate, that is, $cr(s_1) > cr(s_2) > \ldots cr(s_M)$. Then to find conversion rate $cr(N)$ of the first N users reached, we take the segment s_n in which users fall. We find the $n-th$ segment s_n by $n = \arg_n\{(\sum_{i=1}^{n} s_i) < N\}$. Then we compute the weighted average:

$$\frac{[\sum_{i=1}^{n-1} cr(s_i) \cdot reach(s_i)] + [cr(s_n) \cdot (N - \sum_{i=1}^{n-1} reach(s_i))]}{N}$$

where N is the number of users reached. We choose a popular advertiser (we keep their information anonymous for privacy), and we run AutoSegmentation on one of his campaign's data, that is $650K$ impressions for a 15 day-long period. About $4K$ impressions have $\gamma(u) > 1$ for users $u \in U$, which means that the return on investment is greater than the total cost for the advertiser. Our algorithm runs with this campaign's data as input, and it produces a set of sub-segments with optimal conversion performance. To prove the value of using advertiser reputation and user reliability scores as features during rule extraction, we show the performance of both our reputation model, that is, including the scores among the features (named as "reputation" in the plot), and the performance of our model using only the basic demographic, techno-graphic and behavioral features (named as "basic-model" in the plot). We compare the model segments against the segments originally formed by the advertiser and we display our results in Figure 4. Figure 4 illustrates the performance of $cr(N)$ as reach N increases (log scale figure shown in Figure 4). Note that in these figures, reach and conversion are computed on the entire data-set of impressions of users, that is all viewers, clickers and action takers are included.

We notice that for the first 1,000 users reached, the conversion rate of the recommended segments ranges between 0.15 and 0.24 for our basic model and between 0.32 and 0.5 for the reputation model, while for the next 24,000 users it ranges between 0.08 and 0.25 for the basic model and between 0.22 and 0.5 for the reputation model. The rates of the next 500,000 users vary between 0.01 and 0.05. On the other hand, the advertiser's original segments do not reach higher conversion rate than 0.025 for the entire population of users targeted in this campaign. These results show that our model improves conversion significantly. Similar results are extracted when this experiment is performed in other campaigns.

4.4 Negative Recommendations Contribution

Besides recommending which users to target it is also useful to denote which users is not advisable to target, since they are not expected to convert. In this experiment we learn "bad" segments, that is segments with low conversion rates, on a training set of line items, and we recommend that advertisers do not target these users in a testing set of line items. Then we compare the conversion performance of the original segments as selected by the advertisers, against conversion performance as it would be if the "bad" users suggested by the negative recommendations were removed. The results in Figure 5 show that the rates are improved by 7% on average across the segments.

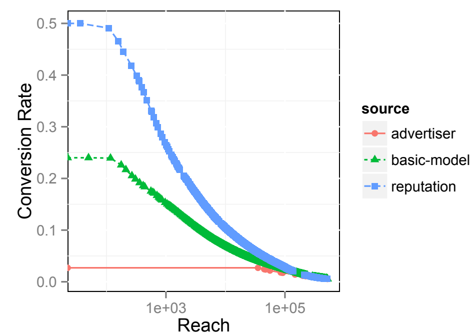

Figure 4: Cumulative conversion rate as reach increases (log scale)

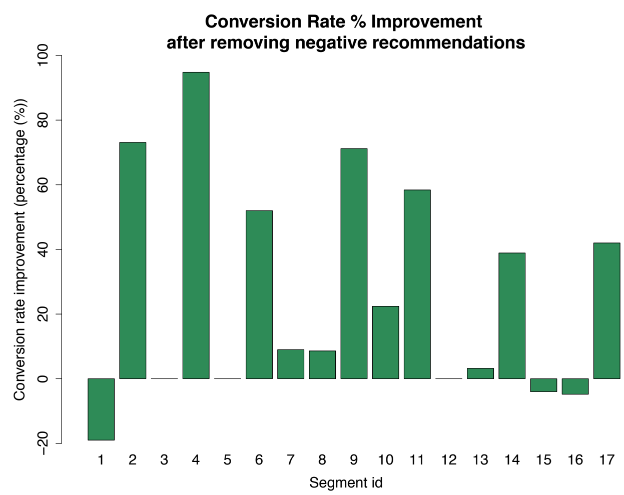

Figure 5: Negative Recommendations Effect: Conversion Rate improvement Percentage after removing negative recommendation users

5 RELATED WORK

The current work mainly overlaps with two areas of related work; audience selection and reputation systems. In audience selection, Pandey et.al. [17] present a description of targeting approaches and how focusing on conversion instead of clicks benefits targeting quality in behavioral targeting. In audience selection related work, the three typical approaches for identifying the best users to target (ad targeting) are property targeting, user-segment targeting and behavioral targeting (BT). In property targeting, users expected to visit a particular page are targeted with the placement of particular ads on that page. In user-segment targeting user with common demographic features are targeted, such as age and gender, such that a meaningful user group is defined (e.g., young adults). In BT, user online behavior is examined, including search queries, email responses, browsing activities. Users likely to convert are then targeted. In the first two cases groups of users are formed, while in

BT users are targeted in individual level. Tyler et.al. [19] solve audience selection as a ranked retrieval problem. Fuxman et. al. [10] present an audience selection method that focuses on modeling user interests to infer targeting. Archak et.al. [4] describe ad factors based aggregation of user information that the advertiser can use to extract deeper insights about the effects of their ads. Bilenko et. al. [5] present a user personalized advertising model that build a user profile under the user's privacy control. Provost et. al. [18] suggest extracting quasi-social networks from browser behavior on user-generated content sites, to find good audiences for brand advertising. Kanagal et.al. [13] propose a focused matrix factorization model to learn user preferences towards specific campaign products, while also exploiting information about related products. Also, Aly et. al. [3] build a web-scale user modeling platform for optimizing display advertising targeting. Finally, Grbovich et. al. identify users to target based on advertisers expectation about user behavior using (manually defined rules [11].

Research on reputation systems is related to our work, as we build a reputation system for advertisers, and we derive reliability scores for users. In reputation systems, several works propose systems that represent the quality of the involved parts and methods to compute reputation scores along with bias. In [8] and [9], Daltayanni et. al. propose WorkerRank, a reputation system to score workers and employers in an online labor marketplace. This work is also based on bipartite relations, similar to the approach in the current study. Auto-segmentation is one case of reputation systems in two-sided marketplaces as described in[7]. In [15], Kokkodis et. al. address data sparseness in building reputation systems in labor marketplaces. In [20], Weng et. al. build reputation scores such that they represent an influence measure for Twitter users. In [6], Chen et. al. discuss how to de-bias reputation in a comments rating environment. Finally, the works of Adler et. al. in [1] and [2] study reputation in the Wikipedia environment and they achieve to measure the quality of contributions.

6 CONCLUSIONS

Audience selection is a hard problem that advertisers do not have good enough data to solve; the available user behavior data is too large and sparse and there are not enough informative signals to use in order to constrain the user space and select the best users suitable for a campaign. In this study, we showed how a DSP that has data from many advertisers and users can help advertisers solve the above problem. We proposed Auto-Segmentation, a novel approach to combine the signals that we take from users and advertisers, and to use them within the context of a reputation system to automate user segmentation. We showed experimentally how to use auto-segmentation for audience selection; first, we showed the contribution in recommending optimal conversion segments to new advertisers, improving the performance for new campaigns that face the cold start problem by $40 - 450\%$. Second, we showed how the recommended segments can replace existing ones, contributing to refining the advertisers' campaigns and achieving better conversion rates. In future research, we would explore using advertiser targeting decisions as signals in bidding optimization.

REFERENCES

[1] B. Thomas Adler and Luca de Alfaro. 2007. A Content-driven Reputation System for the Wikipedia *(WWW '07)*. ACM, New York, NY, USA, 261–270. https://doi.org/10.1145/1242572.1242608

[2] B. Thomas Adler, Luca de Alfaro, Ian Pye, and Vishwanath Raman. 2008. Measuring Author Contributions to the Wikipedia *(WikiSym '08)*. ACM, New York, NY, USA, Article 15, 10 pages. https://doi.org/10.1145/1822258.1822279

[3] Mohamed Aly, Andrew Hatch, Vanja Josifovski, and Vijay K. Narayanan. 2012. Web-scale User Modeling for Targeting *(WWW '12)*. ACM, New York, NY, USA, 3–12. https://doi.org/10.1145/2187980.2187982

[4] Nikolay Archak, Vahab S. Mirrokni, and S. Muthukrishnan. 2010. Mining Advertiser-specific User Behavior Using Adfactors *(WWW '10)*. ACM, New York, NY, USA, 31–40. https://doi.org/10.1145/1772690.1772695

[5] Mikhail Bilenko and Matthew Richardson. 2011. Predictive Client-side Profiles for Personalized Advertising *(KDD '11)*. ACM, New York, NY, USA, 413–421. https://doi.org/10.1145/2020408.2020475

[6] Bee-Chung Chen, Jian Guo, Belle Tseng, and Jie Yang. 2011. User Reputation in a Comment Rating Environment *(KDD '11)*. ACM, New York, NY, USA, 159–167. https://doi.org/10.1145/2020408.2020439

[7] M. Daltayanni. 2015. Reputation Systems in Labor and Advertising Marketplaces. (2015). https://escholarship.org/uc/item/2ch0s50b

[8] Maria Daltayanni, Luca de Alfaro, and Panagiotis Papadimitriou. 2015. WorkerRank: Using Employer Implicit Judgements to Infer Worker Reputation *(WSDM '15)*. ACM, New York, NY, USA, 263–272. https://doi.org/10.1145/2684822.2685286

[9] Maria Daltayanni, Luca de Alfaro, Panagiotis Papadimitriou, and Panayiotis Tsaparas. 2014. On Assigning Implicit Reputation Scores in an Online Labor Marketplace. In *EDBT*. 724–725.

[10] Ariel Fuxman, Anitha Kannan, Jessie Li, and Panayiotis Tsaparas. 2012. Enabling Direct Interest-Aware Audience Selection. In *CIKM*. http://research.microsoft.com/apps/pubs/default.aspx?id=170084

[11] Mihajlo Grbovic and Slobodan Vucetic. 2014. Generating Ad Targeting Rules Using Sparse Principal Component Analysis with Constraints. In *WWW*. 283–284. https://doi.org/10.1145/2567948.2577351

[12] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2001. *The Elements of Statistical Learning*. Springer New York Inc., New York, NY, USA.

[13] Bhargav Kanagal, Amr Ahmed, Sandeep Pandey, Vanja Josifovski, Lluis Garcia Pueyo, and Jeffrey Yuan. 2013. Focused matrix factorization for audience selection in display advertising. In *ICDE*. 386–397.

[14] Jon M. Kleinberg. 1999. Authoritative Sources in a Hyperlinked Environment. *J. ACM* 46, 5 (Sept. 1999), 604–632. https://doi.org/10.1145/324133.324140

[15] Marios Kokkodis, Panagiotis Papadimitriou, and Panagiotis G. Ipeirotis. 2015. Hiring Behavior Models for Online Labor Markets. In *WSDM 2015, Shanghai, China, February 2-6, 2015*. 223–232. https://doi.org/10.1145/2684822.2685299

[16] Sébastien Lahaie, David M Pennock, Amin Saberi, and Rakesh V Vohra. 2007. Sponsored search auctions. *Algorithmic game theory* (2007), 699–716.

[17] Sandeep Pandey, Mohamed Aly, Abraham Bagherjeiran, Andrew Hatch, Peter Ciccolo, Adwait Ratnaparkhi, and Martin Zinkevich. 2011. Learning to Target: What Works for Behavioral Targeting *(CIKM '11)*. ACM, New York, NY, USA, 1805–1814. https://doi.org/10.1145/2063576.2063837

[18] Foster Provost, Brian Dalessandro, Rod Hook, Xiaohan Zhang, and Alan Murray. 2009. Audience Selection for On-line Brand Advertising: Privacy-friendly Social Network Targeting *(KDD '09)*. ACM, New York, NY, USA, 707–716. https://doi.org/10.1145/1557019.1557098

[19] Sarah K. Tyler, Sandeep Pandey, Evgeniy Gabrilovich, and Vanja Josifovski. 2011. Retrieval Models for Audience Selection in Display Advertising *(CIKM '11)*. ACM, New York, NY, USA, 593–598. https://doi.org/10.1145/2063576.2063664

[20] Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. 2010. TwitterRank: Finding Topic-sensitive Influential Twitterers *(WSDM '10)*. ACM, New York, NY, USA, 261–270. https://doi.org/10.1145/1718487.1718520