

Effective Audience Extension in Online Advertising

Jianqiang Shen *
Turn Inc.
901 Marshall Street
Redwood City, CA, USA
jshen@turn.com

Sahin Cem Geyik *
Turn Inc.
901 Marshall Street
Redwood City, CA, USA
sgeyik@turn.com

Ali Dasdan
Turn Inc.
901 Marshall Street
Redwood City, CA, USA
adasdan@turn.com

ABSTRACT

In digital advertising, advertisers want to reach the right audience over media channels such as display, mobile, video, or social at the appropriate cost. The right audience for an advertiser consists of existing customers as well as valuable prospects, those that can potentially be turned into future customers. Identifying valuable prospects is called the *audience extension* problem because advertisers find new customers by extending the desirable criteria for their starting point, which is their existing audience or customers. The complexity of the audience extension problem stems from the difficulty of defining desirable criteria objectively, the number of desirable criteria (such as similarity, diversity, performance) to simultaneously satisfy, and the expected runtime (a few minutes) to find a solution over billions of cookie-based users. In this paper, we formally define the audience extension problem, propose an algorithm that extends a given audience set efficiently under multiple desirable criteria, and experimentally validate its performance. Instead of iterating over individual users, the algorithm takes in Boolean rules that define the seed audience and returns a new set of Boolean rules that corresponds to the extended audience that satisfy the multiple criteria.

Categories and Subject Descriptors

H.3.5 [Information Storage and Retrieval]: Online Information Services; I.2.1 [Artificial Intelligence]: Applications and Expert Systems

General Terms

Algorithms, Application

Keywords

Online advertising; Targeting; Audience Extension

* The authors contributed to this work equally.

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. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
KDD'15, August 10-13, 2015, Sydney, NSW, Australia.
© 2015 ACM. ISBN 978-1-4503-3664-2/15/08 ...\$15.00.
DOI: <http://dx.doi.org/10.1145/2783258.2788603>.

1. INTRODUCTION

Advertising budgets are increasingly moving towards programmatic or digital advertising. In digital advertising, the ecosystem is roughly divided into four main entities: Demand side platforms (DSPs), ad exchanges, supply side platforms (SSPs), and data management platforms (DMPs). Advertisers are on the demand side, i.e. they want to reach the right audience at the right place, cost, and time. Advertisers use DMPs to understand and define their audiences and DSPs to define and execute advertising campaigns to reach those audiences.

Access to an audience occurs through a number of ad formats (such as banner ads, video ads, etc.) on a number of devices (such as desktops, TVs, mobile phones, etc.). The audience is of course on the supply side, paying their attention to publishers through browsing web pages, playing games on mobile phones, shopping on e-commerce sites, etc. SSPs help publishers to monetize what keeps bringing audience to publishers.

Ad exchanges are the bridges between demand and supply sides. Given a user in an online context, ad exchanges perform a match between advertisers (i.e. bids and ads) and publishers (i.e. ad-space) in real-time. It is usually the case that the highest bid wins the match [18].

DMPs have been emerging as a central hub to seamlessly (and rapidly) collect, integrate, manage and activate large volumes of user data [10]. Such user data could be first-party data (i.e., historical user data collected by advertisers in their private customer relationship management systems), or third-party data (i.e., data provided by third-party data partners, typically each specializing in a specific type of data, e.g., demographics, credit scores, buying intentions, etc.). After advertisers utilize DMPs to tie their existing customers to their digital identities (fully anonymous without any personally identifiable information), they could run two kinds of advertising campaigns: retargeting campaigns or prospecting campaigns. The former are run to target existing customers with ads that may help them make more purchases. The latter are run to reach new audiences and convert some portion of them to customers.

In prospecting campaigns, the first and probably the most important question is to define the right audience to reach. By leveraging the data on DMPs, an advertiser can set up a specific segment (groupings of users according to some rules, e.g. “male users in California of ages 25-35”) towards which to target their advertisements. For example, to focus on performance, advertisers can target audiences that have previously visited their websites to maximize the likelihood

of conversion (e.g. product purchase, subscription, filling out a form, etc.); to promote their brand, advertisers can target audiences that are marked with a given demographic profile by some third-party data provider (e.g. “Parents of Infants” marked by an independent data provider).

Such audience segments are usually manually created for multiple reasons. Most of the time, manual generation of targeted user segments is due to advertiser policy, e.g. a sports company may only target certain age ranges, or a certain income group. Such a choice can be easily justified even if the online performance (clicks or conversions) of such segments is not that good, since the advertiser may want to reach these audiences to improve company recognition (this recognition will help offline purchases etc.). Another reason is to achieve additional filtering on top of automatically generated user audiences to improve performance and obey company policy. The final reason can be listed as simply the intuition of the advertiser, i.e. carrying of advertiser experience/know-how from different domains into the online advertising domain.

In many cases, the manually created segment only covers a small percentage of population and needs to be expanded. Here are two use cases:

- *Predictive targeting*: The advertiser identified a group of clickers. With the assumption that users with similar profiles tend to behave similarly, now they want to find other audiences having similar profiles with those clickers and deliver advertisement impressions to them.
- *Data sparsity*: It is very difficult to collect reliable demographic data on an audience. For example, though there could be plenty of “Parents of Infants” in the audience pool, only a few of them are actually marked as “Parents of Infants”. Advertisers want to find other audiences that might also be parents of infants.

In this paper we focus on the topic of what we call *audience extension*. This problem deals with how to best extend the advertiser-provided set of targeted segments so that the new set of users in the extended segment is similar to the original segments’ set of users, and performance metrics (such as click-through rate, conversion rate, or return-on-investment which is the ratio of the amount of monetary value received from clicks and conversions to the amount of money spent by the advertiser on showing impressions) are preserved or improved in the extended segments. Furthermore, by extending the initially provided audience, we increase the advertiser’s reach of users, as well as the amount of money that can be spent on impressions (spending capacity [13]) by the advertiser. The contributions of this paper are as follows:

- Formal definition of audience extension problem in the online advertising domain,
- Multiple algorithms to extend audiences initially provided by an advertiser,
- Parallel implementation details of our audience extension framework which scales to the terabyte datasets in the online advertising domain,
- Evaluation of different methodologies proposed in terms of their effect on user reach, click/conversion performance etc.

To the best of our knowledge, this is the first paper that focuses on the topic of audience extension, and fills up a sig-

nificant void in the literature on the methodical examination of this crucial industry problem.

The rest of the paper is organized as follows. We give a more detailed description as well as some explanatory examples of *audience extension* problem in § 2. Previous work in the literature for similar problems is discussed in § 3. Later, we give multiple methodologies for solving audience extension problem in § 4, and the parallel implementation details needed to scale to our domain’s large data sizes in § 5. Finally, we conclude the paper with the evaluation of the proposed audience extension methodologies in § 6, followed by the paper’s summary as well as our future work suggestions in § 7.

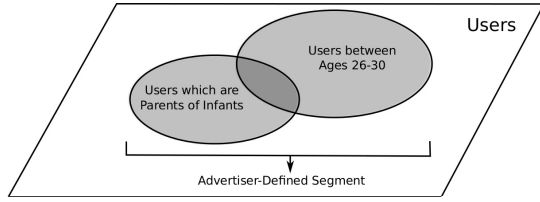
2. PROBLEM DEFINITION

Background and Motivation. In digital advertising, users are usually referred to as the audience (for advertisements). From the business perspective, advertisers wish to learn everything about their audience: their demographics, psychographics, online and offline behavior, and how they respond to different types of advertisements. In Turn’s DMP system, each user is essentially a cookie ID with an associated profile data. Examples of data are visits to specific pages or purchases, user demographics (age, gender, income), or user intent (searching for a used car on an auction site). User data can come from an online context (pixels placed on pages) or could be collected offline, through a matching process joined to online users. Data can be owned by the advertiser (first-party) or can be purchased (third-party).

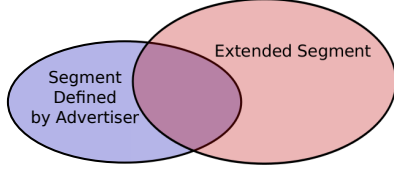
First-party and third-party data usually originate from the data source’s own cookie space and identifying the users across all these platforms remains a constant challenge. Inside Turn, we achieve this by syncing cookie IDs, which allows us to incorporate different types of data from a variety of partners and make the data actionable. For example, now advertiser “InfantMilk” might observe that users purchasing their product usually have label “YoungCouple” from a third party. They can then utilize such information so that the impressions are delivered towards this specific population. Advertisers have to pay an impression-based fee for using third-party data, which is called *data cost*; and in reality, it is another reason for audience extension. An advertiser may prefer to utilize the extended audience rules, especially if the third-party data in the extended rules have a lower data cost.

Such a holistic view of the audience data is a key factor to design of highly targeted ad campaigns that would maximize the return-on-investment (ROI). First and third-party data essentially tag users into all kinds of label categories. An advertiser can then decide the audience population for advertisement delivery by designing some rules to check whether a user has some specific labels. In this paper, we focus on how to extend such a manually-selected audience population to improve audience size and preserve the performance metrics. Each user is represented as a bag of data categories $u_i = \langle c_{i1}, c_{i2}, \dots, c_{in} \rangle$, where c_{ij} is category j assigned to user u_i from some data source.

DEFINITION 1. A category variable $I_j(u_i)$ is a boolean indicator which is 1 if user u_i has category label c_j , 0 otherwise. Sometimes we simplify $I_j(u_i)$ as c_j for brevity. An audience rule is a propositional logic formula where each variable is a category variable.



(a) Example of an advertiser-defined segment



- Given a segment **S**, find a **larger** audience that is
 - **Similar** to the audience inside **S**
 - Able to bring **good ROI**

(b) Example for audience extension problem

Figure 1: Explanative examples for advertiser-defined segments and audience extension problem.

DEFINITION 2. A segment is a portion of the online users that qualify for the advertiser-defined audience rule. Such rules are propositional logic formulas operating over user properties that can be first-party or third-party.

Two examples of advertiser-defined segments are given in Figure 1(a). In the figure, segment “Age Range 26-30” is defined as the propositional logic formula

$$(\text{Age}26 \vee \text{Age}27 \vee \text{Age}28 \vee \text{Age}29 \vee \text{Age}30),$$

which checks whether a user has one of the given age labels. Segment “Parents of Infants” is defined as

$$(\text{ParentOfInfant}),$$

which checks whether a user has label “ParentOfInfant”. Both segments are subsets of the total set of users. The intersection of these two segments can further be defined as a new segment which essentially covers young parents of infants. If an advertiser originally targets those young parents of infants and needs to expand its audience pool, *Parents of Infants* or *Age Range 26-30* will be a natural choice.

Research Problem. Given the fact that there are billions of users labeled with all kinds of data categories, advertisers usually conclude that their manually created segment is small and needs to be expanded. This *audience extension* problem can be formally defined as follows:

DEFINITION 3. Given an advertiser-defined segment rule S , Audience Extension recommends a new user population with a segment rule S' such that,

$$\begin{aligned} \text{sim}(S, S') &> \alpha, \\ \text{perf}(S') - \text{perf}(S) &> \beta, \text{ and} \\ |\text{aud}(S \vee S')| &\gg |\text{aud}(S)|. \end{aligned} \quad (1)$$

Above, *sim* is a similarity score which looks at how similar the two audiences (defined by S and S') are, *perf* is the performance metric (conversion/click rate, ROI etc.) calculated for the two audiences, and finally, $|\text{aud}(S)|$ denotes the size

(number of users that fall into the rule, or *reach*) of segment S (please note that $|\text{aud}(S \vee S')| < |\text{aud}(S)| + |\text{aud}(S')|$ in most cases due to intersection of users between two segments). Note that there could be multiple definitions to calculate *sim* and *perf*, as we will show later.

An example for the above problem definition is given in Figure 1(b). In the figure, we want the intersection between two segments to be large, and the extended segment to bring additional reach (novel set of users) and have good performance (only ROI has been given in the figure for conciseness). We will be exploring multiple methodologies that we developed for our system in § 4.

There are multiple factors that make audience extension challenging. First, it is usually infeasible to optimize towards each requirement at the same time. Users which are highly similar to the seed audience are usually rare, let alone the set of events (clicks, conversions) that they are involved in, which are necessary to compute their performance. We have to keep a balance among different criteria. Second, computational efficiency is extremely desirable in audience extension. There are tens of thousands of categories in our system and any propositional logic formula of them could be a segment. Given the huge number of possible category combinations, pre-computing the recommendation offline for each possible segment would be expensive. A fast audience extension algorithm is critical for user experience. Finally, advertisers want the extension results to be easily understood, evaluated and modified. This excludes most of black-box machine learning methods for our purpose. In this paper, the search space of S' we focus on is the logical disjunction forms of categories, i.e. $S' = (c_1 \vee c_2 \vee \dots \vee c_m)$. Like other overwhelmingly common “Find Good Items” tasks in the *recommender systems* literature [15], we would suggest a list of the recommended categories, along with predictions (scores) for how much the advertisers would like them.

3. PREVIOUS WORK

Our problem is significantly different from another audience extension definition used in online advertising [1, 2], which allows publishers to earn extra money by tagging users who visit their websites, so that this information can later be used by themselves or other advertisers to target those users even when they are outside of the publisher’s domain. As explained in the introduction section, we focus on methodically increasing the audience population based on an advertiser’s manually created rule. Research work in some other domains is related to ours and could potentially be applied.

Understanding User Interests. In order to perform effective advertisement targeting, it is critical to understand user interests. As users interact with content and advertising, their passive behavior can reveal their interests towards advertising [20]. Since there are usually plenty of clicks while other information is not available at a large scale, most existing work in behavioral targeting uses clicks on advertisements as a proxy of interest [8, 22, 27]. Maximizing clicks does not necessarily imply maximizing purchase activities or transactions which directly translate to advertiser’s revenue [20]. Recently, advertisers have been willing to share individual responses to advertisements, making it feasible to use machine learning methods to develop models that are specifically optimized for conversions [5, 4]. There have also been some efforts in the literature that approximate user

interest with demographics. For example, Joshi et al. [16] aim to generate textual data for each user based on demographic information, by which the authors aim at efficiently matching advertisements to potentially interested users.

User Clustering. User segmentation, which aims at grouping users into user segments with similar behaviors, is crucial to behavioral targeting. If users with similar purchase intentions can be automatically clustered into the same segment, advertisers may gain more profit from advertisement delivery. [24, 3] utilize Latent Dirichlet Allocation (LDA), and [26] utilizes probabilistic latent semantic analysis (pLSA) to group users for advertising, hence generating relevant audience subsets over the available users. In this way, it is possible to evaluate the benefit of each group for each advertiser. While these works are useful in matching appropriate advertisers with appropriate audiences, they do not take into account either advertiser-provided initial segments (our method is basically how to build on top of such prior information), or provide an option for advertisers to choose their audiences. As we will explain later, we provide such an option via weighting of different audience metrics.

Extending Social Networks. Research shows that users connect to both friends they already know offline and new friends they discover on social networking sites [9]. Given the size of social networking sites, finding known contacts and interesting new friends to connect with on the site can both be a challenge. It has been shown that algorithms based on text content or friendship connections are effective in providing people recommendations and can significantly increase the number of friends of a user on the site [6]. An interesting work given in [19] actually may intuitively be utilized as a very feasible way to extend audiences. The authors look at friendship information between users (collected via an online social networking platform) for targeting. It would be intuitive to have the friendship network as a graph (where nodes are online users, and edges represent friendship). Given an initial set of nodes (users) in this graph by the advertiser as the original audiences (i.e. an advertiser-provided subgraph/audience), we can generate an extended audience as the users that have a direct edge (or we can look at n -hop neighborhood depending on advertiser preference or performance metrics etc.) to the original subgraph. [19] does not follow this kind of route, and unfortunately such friendship information is often proprietary (not even available as third-party data).

Recommender Systems. Audience Extension is a special case of making recommendations. There is a wealth of research on recommender systems [15]. One of the most successful technologies for recommender systems, called collaborative filtering, has been developed and improved over the past decade to the point where a wide variety of algorithms exist for generating recommendations. It utilizes similarities of preferences among users to recommend items such as movies for a user to consume. It does not really analyze the actual content of the items, but instead require users to indicate preferences on them, usually in the form of ratings.

Since advertisers typically do not share their created segments, applying many of the existing recommender techniques could potentially be very challenging. To address Audience Extension, we can approach Collaborative Filtering from two perspectives. First, we can treat each segment as an item and find rules used by similar advertisers. This

however may raise advertiser privacy concerns. Instead, we could treat each individual audience as an item and find patterns. This is what we adopted in this paper - look into similarity to recommend audiences.

Industry Patents. In terms of patents, there is a single one which is remotely relevant to our work. The authors of [11] describe their approach of Audience Extension, which is quite different from what we propose in this paper. They suggest assigning a score (Audience Extension Score) that reflects the likelihood of each particular user to exhibit a desired behavior, such as desired TV viewing behavior or purchasing behavior. The patent is not specific on how to assign such scores, or how to optimize based on an advertiser's different preferences. Finally, two patents [12, 23] do mention the problem of audience extension, however they do not provide any details on how to recommend extended audiences.

4. METHODOLOGY

Based on the problem formulation, we will present the methodologies we applied to extend advertiser-defined segments. As it will be seen, these are either trying to maximize one single function given in the problem definition, or try to find a balance between them.

4.1 Purely Greedy Approach

As we explained before, the segment(s) provided to us by the advertiser is a propositional rule S , involving n categories c_1, \dots, c_n . This of course does not suggest that the audiences (set of targeted users) belonging to this segment only have these categories. Rather, the users within the segment may have many different properties, but they have been included in this segment due to the fact that they are tagged with the provided set of categories in the segment definition. Therefore, our first method, which we call *purely greedy*, aims to recommend the most frequently occurring set of categories in the segment's audiences, which are **not** in the segment definition. This is a variation of the greedy set cover algorithm, where we want to either completely cover the original segment's audience, or until our newly recommended segment's audience reaches a certain size. Details are given in Algorithm 1.

There are multiple disadvantages of the purely greedy approach. First is the way we generate the extension. Since the algorithm is basically the application of the set cover algorithm, the original segment is a subset of the extended audience. Other than this, the algorithm neither looks at the additional reach of the extended segment (covering the original segment does not mean the extended segment added new users to the original segment, since it can be very close in size to the original audience), nor the performance. Extension is based directly on the set cover algorithm, and at no point of this scheme do we check the return-on-investment (ROI) or any other metric of the newly recommended users. This effect can be seen with a sample we picked from our system which is given in Table 1. The original advertiser-defined segment in this example is a single rule to pick up users in the age range 20-30 (we receive this tag info from the third-party data provider "TP 1"). As it can be seen, the extended audience is not really a gain since it basically either divides the age range to partitions, or presents some obvious facts (the user is a child in the family, i.e. is not the head of household, or makes under 15K dollars a year),

Algorithm 1: Greedy approach to extend an audience

Input: S : original segment, m : desired audience size

- 1 $\Omega \leftarrow \text{aud}(S)$;
- 2 Collect $C \leftarrow c_{n+1 \rightarrow n+m}$, all categories from the users in $\text{aud}(S)$ that are not in definition of S , such that $C \cap S = \emptyset$ and,
 $\forall c_j \in C, \exists u_i \in \text{aud}(S) \text{ s.t. } u_i \in \text{aud}(c_j)$;
- 3 $\forall c_j \in C$, count the # of users in Ω that have c_j ;
- 4 Order C into C_{sorted} according to the number of users belonging to each category $c_j \in C$, descending ;
- 5 $S' \leftarrow \emptyset$ //extended audience is empty at the beginning;
- 6 **while** $|\text{aud}(S \vee S')| < m$ **and** $C \neq \emptyset$ **do**
- 7 Get $c_j \in C_{\text{sorted}}$ which is the most frequent ;
- 8 $S' \leftarrow S' \vee c_j$;
- 9 $C \leftarrow C - c_j$;
- 10 $\Omega \leftarrow \Omega - \text{aud}(c_j)$;
- 11 $\forall c_j \in C$, recalculate the number of users ;
- 12 Recalculate C_{sorted} from the new C ;
- 13 Recommend $S \vee S'$ as the new segment

Table 1: Extended audiences generated in our system for the age range 20-30 according to the purely greedy approach.

Rule Property	Data Source	Rule Parent	Rule Definition
Original Rule	TP 1	Age Range	20-30
Extended Rule	TP 1	Family Position	Child
Extended Rule	TP 1	Age Range	22-25
Extended Rule	TP 1	Age Range	26-35
Extended Rule	TP 1	Income	< 15K \$

which comes up with an audience too big, and naturally does not follow the characteristics of the original segment. Second disadvantage with this method is on the running time. Our data have the strict mapping of users to segment rules (not vice versa), hence the set cover logic takes a long processing time, since at every point we need to recalculate the set of users that belong to a segment (this is costly in both sequential and parallel implementation, since we have to go through the audience once for each step, and our user space is in the order of hundreds of millions). Please note that we need the extension to be fast since the advertiser wants to constantly generate new segments and extend them in an interactive application (which we provide in our advertising platform). The rest of the section will be our explanation of a faster algorithm we applied with a much more feasible (better similarity, reach, and performance) recommended audience extension.

4.2 Weighted Criteria-based Algorithm

As we have shown, the greedy approach presented before does not necessarily recommend extended audiences with high reach or performance. In this section, we will present our proposed approach in balancing the necessary metrics as given in the problem definition (Eq. 1). To be exact, we start with an original advertiser-defined segment (set of categories/rules), and come up with an extended audience in a fast manner, while optimizing over the following metrics:

- **Interest:** This metric measures the similarity of the extended and original segments (directly related to $\text{sim}(S, S')$ in Eq. 1),

- **Novelty:** This metric measures the additional audience size generated by the extended audience segment (directly related to $\text{aud}(S \vee S')$ in Eq. 1),
- **Quality:** This metric measures the performance (ROI, click-through rate or conversion rate) of the extended audience (directly related to function perf in Eq. 1).

Next, we will explain the calculation methods for all three of the above criteria.

4.2.1 Calculation of Interest Criterion

We have explored two methodologies for the computation of the interest criteria: (i) Category correlation, and (ii) A simple probabilistic solution (that we decided to apply in the end) which is computationally cheaper, and works well in practice.

Let's first talk about the correlation between categories. This means that if our newly recommended category (to be included in the extended segment) correlates well with the original categories then it has a high interest score. Correlation (*Pearson* correlation coefficient) of the recommended new category with an original category can be calculated as:

$$\text{corr}(c_{\text{new}}, c_{\text{old}}) = \frac{[\sum_{u_i \in U} (I_{c_{\text{new}}}(u_i) - \mu_{\text{new}})(I_{c_{\text{old}}}(u_i) - \mu_{\text{old}})]}{\sum_{u_i \in U} \sigma_{\text{new}} \sigma_{\text{old}}} \quad (2)$$

where,

$$\mu_k = \frac{\sum_{u_i \in U} I_{c_k}(u_i)}{|U|},$$

and,

$$\sigma_k = \sqrt{\frac{\sum_{u_i \in U} (I_{c_k}(u_i) - \mu_k)^2}{|U|}}.$$

Above, $I_{c_k}(u_i)$ is the indicator function that returns 1 if c_k is a category that u_i belongs to or 0 otherwise (please also see Definition 1 in § 2). Basically, the above formulation goes over all users and calculates a metric on the overlap of two categories. Please note that in the above formulation, any category can be replaced with a segment (i.e. set of categories) and the formulas can be applied as is.

Correlation is actually a pretty valid similarity criteria, but we have decided not to use it in our proposed method due to the following reasons. The first reason is data sparsity, since most users belong to a few categories, we have observed that the computed correlation values are very close to each other (and very small) for most category pairs. Furthermore, we observed that correlation values are either too small (this is the prominent case), or too high (match perfectly), in which case the extended audience would match perfectly to the original audience, hence would be redundant.

As listed above, the second possibility we explored (and the method of computation we chose) for the interest criteria is based on a simple probabilistic formula. This formulation gives the overlap between the original segment and a newly recommended category, and is as follows:

$$\text{sim}(c_{\text{new}}, S) = p(c_{\text{new}}|S) = \frac{|\text{aud}(c_{\text{new}}) \cap \text{aud}(S)|}{|\text{aud}(S)|}.$$

This formula basically answers the following question: "If a user belongs to segment S , what is the probability that





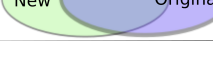
	Similarity $P(New Original)$	Novelty $P(!Original New)$	Value Good/OK/Bad?
	1	0	Bad
	1	≈ 0.5	Good
	≈ 0.5	0	Bad
	≈ 0.2	≈ 0.8	OK
	≈ 0.8	≈ 0.2	OK

Figure 2: Examples of similarity vs. novelty in original and extended audiences.

the same user also belongs to category c_{new} ?”. This value measures how well the extended audience covers the original audience. Please note that the purely greedy solution we describe in § 4.1 maximizes this, but as we will show in the rest of the section, we look at *novelty* and *quality* as well.

4.2.2 Calculation of Novelty Criterion

Novelty is directly related to the size of new audience (novel audience, i.e. that does not belong to the original audience) in the newly recommended extension. Since we may have different sizes of original or extended audiences, we need a normalized value (rather than just the absolute value of the novel audience) for the novelty of a new category, which is given as the following probability:

$$\begin{aligned}
 nov(c_{new}, S) &= p(!S|c_{new}) = 1 - p(S|c_{new}) \\
 &= 1 - \frac{|aud(c_{new}) \cap aud(S)|}{|aud(c_{new})|}.
 \end{aligned}$$

The above formulation can easily be applied to a segment, and answers the following question: “If a user belongs to our newly recommended extended category, what is the probability that it will **not** belong to the original advertiser-provided segment/audience, hence is *novel*?”. We show an interpretation of the balance between *similarity* and *novelty* in Figure 2. It can be easily seen that we need both values to be balanced, since the *goodness* of the extended audience does not depend on a single one of the metrics. Furthermore, we give some demonstration of how the metrics of *similarity* (probabilistic formula), *novelty*, and *correlation* (this is what we explored, but ended up not using, for similarity metric) look like for different segments’ extensions in Tables 2, 3, and 4. Please note that the tables are created using a set of categories generated by different first/third-party providers and do not necessarily represent ground-truth (since all providers only have access to, and can reliably tag a certain subset of all users).

4.2.3 Calculation of Quality Criterion

Quality criteria can be calculated in multiple ways, depending on an advertiser’s preferences. We can calculate

Table 2: Extended category similarity, novelty, and correlation metrics for the original advertiser-provided segment of “Interest: Gossip” (ordered by similarity).

Category	Sim	Nov	Corr
Interest: Arts & Entertainment	0.746	0.91	0.251
Gender: Female	0.37	0.912	0.173
Interest: News	0.328	0.911	0.163
Interest: TV & Video	0.231	0.966	0.077
Interest: Movies	0.189	0.964	0.073

Table 3: Extended category similarity, novelty, and correlation metrics for the original advertiser-provided segment of “State: California” (ordered by similarity). “Ethnicity: CA” represents users who were tagged by a third-party as born in California.

Category	Sim	Nov	Corr
Age: 18-34	0.642	0.721	0.415
Gender: Male	0.554	0.670	0.419
Ethnicity: CA	0.465	0.699	0.365
Politics: Democrat	0.415	0.674	0.36
Politics: Unaffiliated	0.285	0.644	0.312

click-through rate (CTR), conversion rate (CVR), or return-on-investment (ROI) for an extended category c_{new} generated for an advertiser adv as follows:

$$CTR_{c_{new}} = \frac{\sum_{u \in aud(c_{new})} click(u, adv)}{\sum_{u \in aud(c_{new})} imp(u, adv)}, \quad (3)$$

$$CVR_{c_{new}} = \frac{\sum_{u \in aud(c_{new})} conv(u, adv)}{\sum_{u \in aud(c_{new})} imp(u, adv)}, \quad (4)$$

$$ROI_{c_{new}} = \frac{\sum_{u \in aud(c_{new})} value(u, adv)}{\sum_{u \in aud(c_{new})} cost(u, adv)}. \quad (5)$$

Above, we have

- $click(u, adv)$ is the number of clicks that user u (having category c_{new}) performed on impressions provided to it by advertiser adv ,
- $imp(u, adv)$ is the number of impressions shown to user u (having category c_{new}) by advertiser adv ,
- $conv(u, adv)$ is the number of conversions that user u (having category c_{new}) performed on impressions provided to it by advertiser adv ,
- $cost(u, adv)$ is the amount of money spent by advertiser adv to show impressions to user u (having category c_{new}),

Table 4: Extended category similarity, novelty, and correlation metrics for the original advertiser-provided segment of “Financial: Makes Stock Trades” (ordered by similarity).

Category	Sim	Nov	Corr
Activity: Uses Financial Advisor	0.938	0.142	0.897
Trade Style: Online	0.845	0.24	0.8
Activity: Research Financial Prod.	0.779	0.366	0.702
Investment Type: Real Estate	0.774	0.443	0.655
Member: Credit Rewards Program	0.765	0.418	0.665

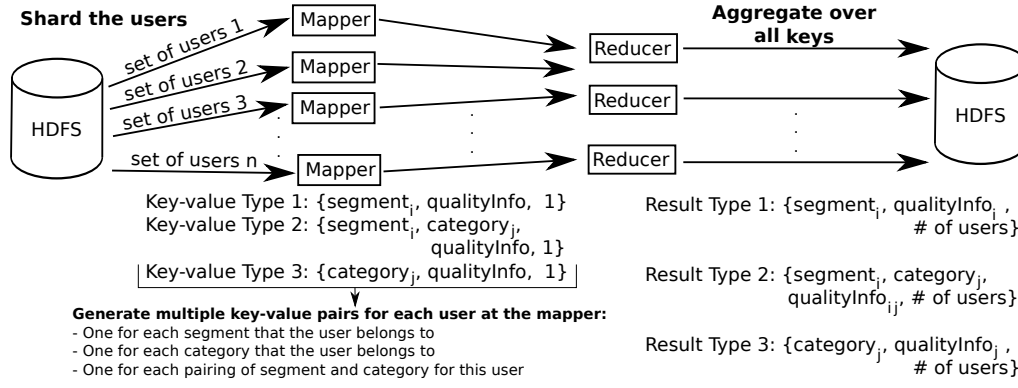


Figure 3: Implementation Details for Weighted Criteria-based Audience Extension

- $value(u, adv)$ is the amount of value (by the clicks and conversions performed) provided by user u (having category c_{new}) to advertiser adv .

Obviously, we want our new segment/audience/category to have a quality value (chosen to be any of above three) as high as possible, while having both good similarity and novelty compared to the original advertiser-provided audience/segment. The need for such balancing implies a weighted audience extension scheme, which is given next.

4.2.4 Weighted Audience Extension

Given the original segment S , our audience extension method combines the previously defined metrics *similarity* (sim), *novelty* (nov), and *quality* (q) and defines the score for a newly recommended category c_{new} as:

$$score \propto sim(c_{new}, S) \times nov(c_{new}|S) \times q(c_{new}) .$$

The intuition for the above formula is to capture the extra return (via $nov(c_{new}|S) \times q(c_{new})$) from the right audience (via $sim(c_{new}, S)$). To avoid numerical instability, we take log of both sides,

$$\begin{aligned} logScore &\propto log(sim(c_{new}, S)) + \\ &log(nov(c_{new}|S)) + log(q(c_{new})). \end{aligned}$$

To make the recommending more flexible and allow advertisers to balance between criteria, we can assign weights to the log metrics, and when we replace the metrics with the formulations from the previous section, we have:

$$\begin{aligned} logScore(c_{new}|S) &= \theta_1 log(p(c_{new}|S)) + \\ &\theta_2 log(1 - p(S|c_{new})) + \theta_3 log(q(c_{new})) . \end{aligned} \quad (6)$$

The θ values are application specific, but we can furthermore entitle different extension choices to the advertiser (according to different θ s), giving them the option to choose as they please. Also, the reason we did not elaborate on the formulation of *quality* metric above is because there might be different choices, and Equations 3, 4, or 5 can all be inserted in place of $q(c_{new})$.

Based on the formulation, we can sort different categories, and recommend the top- k ones to the user. Furthermore, it is always possible to pre-compute a segment-to-category matrix of goodness metrics, and immediately calculate the score online for an advertiser's application-use experience.

5. SYSTEM IMPLEMENTATION

In this section, we give implementation details on how we achieved large-scale audience extension at Turn; in specific, the weighted scheme presented in § 4.2. In the online advertising domain, we typically have to deal with virtual users (cookie spaces) in the orders of hundreds of millions. To be able to check the segment or category information, as well as calculate quality metrics such as ROI or CTR, we need to apply parallel algorithms.

System Architecture. Our audience extension system leverages our in-house data warehouse system, called Cheetah [7] built on top of the Hadoop framework [25]. It is designed specifically for our online advertising application to allow various simplifications and custom optimizations. User facts are stored within nested relational data tables.

We present the first step of the audience extension implementation in Figure 3. This figure gives the basic parallel process on generating the data to calculate segment to category similarity, novelty, and quality values. We distribute the whole segment and category data, as well as quality information (e.g. number of impressions as well as number of clicks for one or multiple advertisers) for each user to different mappers. Within each mapper, we generate three types of key-value pairs:

- **Key $\rightarrow \{segmentId\}$, Value $\rightarrow \{qualityInfo, 1\}$:** For each user, we create one of these keys for each segment the user belongs to. Quality info is the numerator and denominator values for one of the Equations 3, 4, or 5 for a single or multiple advertisers.
- **Key $\rightarrow \{categoryId\}$, Value $\rightarrow \{qualityInfo, 1\}$:** For each user, we create one of these keys for each category the user belongs to. Quality info is the numerator and denominator values for one of the Equations 3, 4, or 5 for a single or multiple advertisers.
- **Key $\rightarrow \{segmentId, categoryId\}$, Value $\rightarrow \{qualityInfo, 1\}$:** For each user, we create one of these keys for each (segment, category) pair the user belongs to. We can further exclude the (segment, category) pairs where the category is part of the segment (since we do not want to recommend an already utilized category in our extension). Quality info is the numerator and denominator values for one of the Equations 3, 4, or 5 for a single or multiple advertisers.

After these keys are generated, in the reducer phase we can count the number of users as well as quality information for

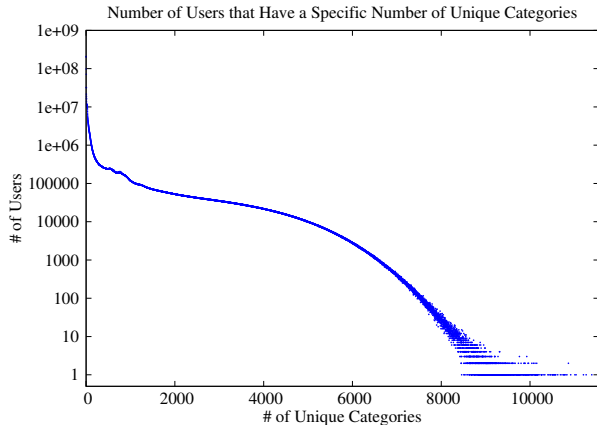


Figure 4: We count how many users have the specified number of unique categories. The plot shows a power-law degree distribution for the number of categories.

each segment, category, or segment-category intersections. These values are then used to calculate similarity, novelty, and quality information as in § 4.2. For this purpose, we need to join segment and category values on the different types of outputs of reducers, again in a parallelized manner. This way we can generate segment-to-category matrices of similarity, novelty, and quality values. At the extension time, it is trivial for each (segment, category) pair to calculate the scores according to given different θ values. We can then sort categories for each segment to be extended and recommend as possible audience extension candidates. The advertiser can then select the extension categories according to their internal metrics or *data cost* to utilize the new categories at run-time for targeting.

Handling Data Skew. Pair-wise computation is expensive in big data systems, since this potentially can involve $O(n^2)$ complexity. Skewed data could create imbalanced load on both mappers and reducers, which makes such an operation much more costly. Varying execution times result in low resource utilization and high overall execution time, since the next MapReduce cycle can only start after all mappers/reducers are done [14, 17]. Our user data is highly skewed. Without any mitigation, counting co-occurrences can lead to significantly longer job execution times and significantly lower cluster throughput.

More specifically, our data skew comes from two dimensions – users and categories. The *user skew* is caused by some long-tail users, which have significantly more category labels than others. They could be bots or spam users. We count the number of each user’s unique categories and plot the histogram in Figure 4. Like many other scientific data, a user’s count of categories roughly fits the power-law function and a small percentage of users have unusually many categories. Sometimes data providers could introduce *category skew* into the data – advertisers pay a data cost if they use the data provider’s data for deciding to deliver an impression. Data providers are motivated to tag more users. In some cases, a category could tag so many users that it loses its discriminative power. For example, the most frequent category, “Traveler” in our data covers more than 55% of our users.

We take two approaches to mitigate data skew. First, we ignore those long-tail users and categories, since they introduce little value to advertisers. In our implementation, the top 5% most frequent categories and 15% most tagged users are removed from the computation. Second, we take the load balance into account when we partition the data: before we do the actual computation, we sample the input records and compute a histogram of the underlying key space. We then partition the data so that each mapper has roughly the same load. With those two approaches implemented, the recommendation speed is improved by more than 450%*.

6. EXPERIMENTAL RESULTS

In this section, we will give some preliminary results on the extended audiences that our proposed algorithms return. As given in Eq. 6, we can assign different weights to *similarity*, *novelty*, and *quality*. This is advertiser/application-specific, i.e. there is no single correct assignment of parameters, but rather we leave it to the advertisers to choose according to their needs.

For our experiments, we have utilized 15 days of user profile data collected at Turn’s online advertising platform within 2014. From this data we can calculate how many users fall into any advertiser-defined segment, or how many users will fall into our recommended extended audience. We present our results for mainly four original segments (whose exact names and properties are not shown here due to privacy reasons):

- **Segment 1** picks up users that have bought a certain brand car in a certain state,
- **Segment 2** picks up users that have interest in a certain winter sport,
- **Segment 3** picks up users that have stayed at a specific hotel chain,
- **Segment 4** picks up users that have bought a new certain type of vehicle and are within a certain age range.

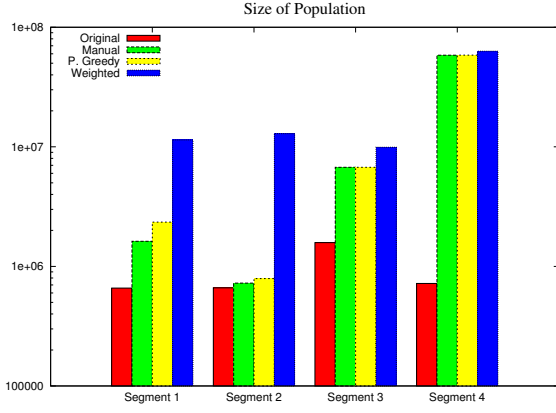
We have performed two experiments which optimize fully, either *similarity score*, or *quality score*, and we present the results below. We compare *Weighted* which is our proposed system from § 4.2 to two other methods: *Manual* gives the human-generated extension (which picks up the top k most frequent categories from the advertiser’s original segment population), and *P. Greedy* is the purely greedy approach given in § 4.1 (set-cover algorithm).

6.1 Population Recovery

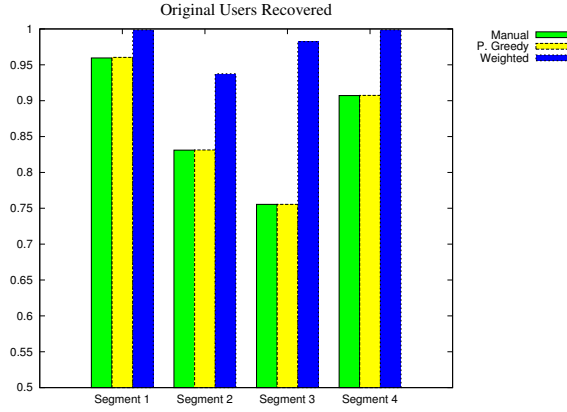
In this experiment, we are examining whether given a subset of an original advertiser-provided segment, we can extend this subset to *recover* the original segment. The exact steps of this experiment is as follows:

1. Pick up the users that belong to an advertiser-provided segment (consisting of k disjoint categories),
2. Sample n (we took this to be 50K for this instance) users from the audience,
3. Pick up the top $k/2$ categories from this user sample, and generate a new segment,

* Exact computation time is not shown here for confidentiality reasons.



(a) Sizes of audiences for the original segment and different recommended extensions.



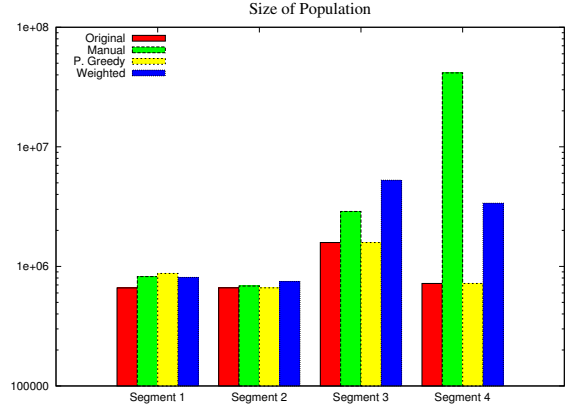
(b) Amount of original audience covered by different recommended extensions.

Figure 5: Results of Population Recovery Experiment

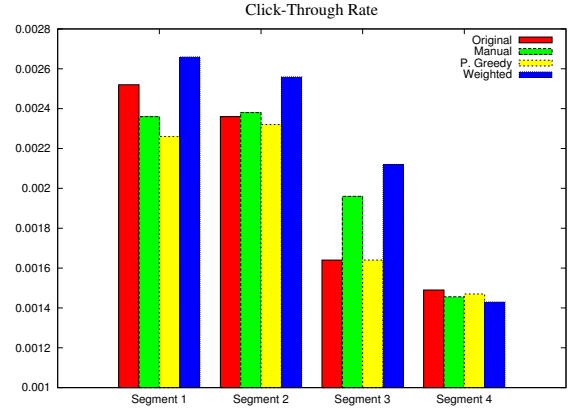
- Using different algorithms, generate $k/2$ recommended categories and check how much this new audience population covers the original segment (the one in Step 1, defined by k disjoint categories).

Results of *population recovery* experiment is given in Figure 5. In Figure 5(a), we present the audience sizes for the *original* segments (Segments 1-4 which consist of $k_{1 \rightarrow 4}$ categories), and then we present the audience sizes for different extension algorithms (extension is made on top of the subset, and our aim is to recover the original segment). For the weighted case, we have chosen to fully optimize similarity ($(\theta_1, \theta_2, \theta_3) = (1, 0, 0)$ in Eq. 6) since this is the most intuitive, and gave the best result in our experience for population recovery. It can be seen from the figure that although *Manual* and *P. Greedy* each employ a different logic to choose the extension categories, audience size results are pretty close. *Weighted* extension returns the largest audience, which also is significantly larger than the original (further boost to novelty, although we were not optimizing towards it).

Figure 5(b) gives the percentage of users (of the original audience) that were covered by the extension algorithms. Our aim for the experiment was to cover as much of the original segment as possible by extending the subset. It



(a) Sizes of audiences for the original segment and different recommended extensions.



(b) CTR values for the original segment and different recommended extensions.

Figure 6: Results of Performance Optimization Experiment

can be seen that our weighted scheme provides significantly higher coverage in all cases ($p < .001$).

6.2 Performance Optimization

In this experiment, we try to recommend an extended population with high quality scores. We chose click-through rates (CTRs, as described in Eq. 3) as our quality metric. For this purpose, we set the parameters $(\theta_1, \theta_2, \theta_3) = (0, 0, 1)$ according to Eq. 6, which fully optimizes according to *quality* score, which we take to be CTR.

The results of the *optimize CTR* experiment is given in Figure 6. We extend the four segments according to different approaches with one recommended category. We can see that our weighted approach comes up with reasonably larger audiences (Figure 6(a)), and the CTR of the extended audience is the largest in most cases (Figure 6(b)). Please note that we have modified the actual CTR values with a constant multiplier for all cases, for privacy purposes. In 3 out of 4 cases, our weighted scheme provides significantly higher CTR values than other methods ($p < .001$). Since the purely greedy approach does not look into the quality of extended audience, it usually produces audiences with a smaller click-through rate compared to the original segments.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we presented the problem of audience extension in online advertising. We provided two solutions, one purely greedy that applies a set-cover algorithm on the original audience, while the other one with different weights on similarity, novelty, and quality of the extended audience. We compared these different approaches on our real-world data set, and explained their implementation in our advertising platform at Turn. Our proposed method is computationally efficient, can flexibly adjust to different needs and shows good performance. To the best of our knowledge, this is the first paper to methodically examine the audience extension problem and possible solutions to it.

There are multiple paths of interesting future work. First, we would like to explore different formulations of the metrics that are optimized for extension, as well as different parameter settings on the weights of similarity, novelty, and quality metrics. Second, our current search space of extension is the logical disjunction forms of categories. It is possible to extend the recommendation to other simple logical formulas, if we can design some heuristics to limit the search space. Lastly, in the future we would like to mine the common rules that advertisers utilize to define a segment and leverage such rules to help recommendations by adapting transfer learning techniques [21].

Acknowledgments

We thank many talented scientists and engineers at Turn for their help and feedback in this work, and the anonymous reviewers for their valuable comments and suggestions.

8. REFERENCES

- [1] <http://digitalmarketing-glossary.com/what-is-audience-extension-definition>, 2011.
- [2] <http://www.knowonlineadvertising.com/sharing-knowledge/audience-extension/>, 2013.
- [3] A. Ahmed, Y. Low, M. Aly, V. Josifovski, and A. J. Smola. scalable distributed inference of dynamic user interests for behavioral targeting. In *Proc. ACM KDD*, pages 114–122, 2011.
- [4] M. Aly, A. Hatch, V. Josifovski, and V. K. Narayanan. Web-scale user modeling for targeting. In *Proc. ACM WWW*, pages 3–12, 2012.
- [5] N. Archak, V. S. Mirrokni, and S. Muthukrishnan. Mining advertiser-specific user behavior using adfactors. In *Proc. ACM WWW*, pages 31–40, 2010.
- [6] J. Chen, W. Geyer, C. Dugan, M. Muller, and I. Guy. Make new friends, but keep the old: recommending people on social networking sites. In *Proc. CHI*, pages 201–210, 2009.
- [7] S. Chen. Cheetah: a high performance, custom data warehouse on top of mapreduce. *Proc. VLDB Endowment*, 3(1-2):1459–1468, 2010.
- [8] Y. Chen, D. Pavlov, and J. F. Canny. Large-scale behavioral targeting. In *Proc. ACM KDD*, pages 209–218, 2009.
- [9] N. B. Ellison et al. Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication*, 13(1):210–230, 2007.
- [10] H. Elmeleegy, Y. Li, Y. Qi, P. Wilmot, M. Wu, S. Kolay, A. Dasdan, and S. Chen. Overview of turn data management platform for digital advertising. *Proc. VLDB Endowment*, 6(11):1138–1149, 2013.
- [11] J. Evans and T. Liebowitz. System and method for targeting advertisements, Nov. 8 2012. WO Patent App. PCT/US2012/032,539.
- [12] F. Falcon. Audience measurement system, Aug. 3 2011. EP Patent App. EP20,080,875,725.
- [13] S. C. Geyik, A. Saxena, and A. Dasdan. Multi-touch attribution based budget allocation in online advertising. In *Proc. ACM ADKDD*, pages 1–9, 2014.
- [14] B. Gufler, N. Augsten, A. Reiser, and A. Kemper. Handling data skew in mapreduce. In *Proc. CLOSER*, 2011.
- [15] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Information Systems*, 22(1):5–53, 2004.
- [16] A. Joshi, A. Bagherjeiran, and A. Ratnaparkhi. User demographic and behavioral targeting for content match advertising. In *Proc. ACM ADKDD*, pages 53–60, 2011.
- [17] Y. Kwon, M. Balazinska, B. Howe, and J. Rolia. Skewtune: mitigating skew in mapreduce applications. In *Proc. ACM SIGMOD*, pages 25–36, 2012.
- [18] K.-C. Lee, B. Orten, A. Dasdan, and W. Li. Estimating conversion rate in display advertising from past performance data. In *Proc. ACM KDD*, pages 768–776, 2012.
- [19] K. Liu and L. Tang. Large-scale behavioral targeting with a social twist. In *Proc. ACM CIKM*, pages 1815–1824, 2011.
- [20] S. Pandey, M. Aly, A. Bagherjeiran, A. Hatch, P. Ciccolo, A. Ratnaparkhi, and M. Zinkevich. Learning to target: what works for behavioral targeting. In *Proc. ACM CIKM*, pages 1805–1814, 2011.
- [21] C. Perlich, B. Dalessandro, T. Raeder, O. Stitelman, and F. Provost. Machine learning for targeted display advertising: Transfer learning in action. *Machine Learning*, 95(1):103–127, 2014.
- [22] F. Provost, B. Dalessandro, R. Hook, X. Zhang, and A. Murray. Audience selection for on-line brand advertising: privacy-friendly social network targeting. In *Proc. ACM KDD*, pages 707–716, 2009.
- [23] C. Sugnet, J. Shoop, P. Sutter, and K. Feldman. Audience segment selection, Aug. 6 2013. US Patent 8,504,905.
- [24] S. Tu and C. Lu. Topic-based user segmentation for online advertising with latent dirichlet allocation. In *LNCS 6441*, pages 259–269, 2010.
- [25] T. White. *Hadoop: The Definitive Guide*. O'Reilly Media, Sebastopol, CA, 2012.
- [26] X. Wu, J. Yan, N. Liu, S. Yan, Y. Chen, and Z. Chen. Probabilistic latent semantic user segmentation for behavioral targeted advertising. In *Proc. ACM ADKDD*, pages 10–17, 2009.
- [27] J. Yan, N. Liu, G. Wang, W. Zhang, Y. Jiang, and Z. Chen. How much can behavioral targeting help online advertising? In *Proc. ACM WWW*, pages 261–270, 2009.