Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 136 (2018) 255–264

Procedia Computer Science

www.elsevier.com/locate/procedia

7th International Young Scientist Conference on Computational Science

Adaptive look-alike targeting in social networks advertising

Artem Popov[a], Daria Iakovleva[a]

[a]ITMO University, 49 Kronverksky pr., St Petersburg, 197101, Russia

Abstract

Look-alike targeting models are based on an assumption that user similarity correlates with the probability of receiving the same positive feedback on advertising. They are successfully applied to searching and targeting of audiences in large-scale advertising networks. Usually, look-alike models lack an ability to improve their performance using experience gained during their application. Due to this, they highly depend on the initial user seed and can be thus affected by many negative factors including possible biases and noise presented in the seed. To minimize the influence of these factors, we suggest a look-alike model that is capable of adopting its targeting strategy to the feedback received. This model exploits Thompson Sampling algorithm applied to the space of users' features. We have evaluated the proposed model in real advertising campaigns in a large social network assuming that its users were described by the online communities, members of which they were. Our method has achieved average 12.5% AUC improvement in comparison to the baseline look-alike models.

Keywords: Lookalike Modelling ; Display Advertising ; Social Networks ; Behavioral Targeting ; Machine Learning.

1. Introduction

Targeted advertising is a form of advertising where online advertisers can use sophisticated methods to target the most receptive audiences with certain traits, based on the product or person the advertiser is promoting [12].

Evolved from traditional media advertising such as television, radio, magazines, and contextual advertising in search engines, targeting advertising is currently now one of the fastest advancing areas in the IT industry [20]. Businesses and services are transitioning into the online space creating a large platform for promoting products and generating a massive amount of user data which gives an opportunity to analyze audience interests and personalize user experience. The advertisement industry has created many opportunities and problems for data mining and machine

* Artem Popov. Tel.: +7-952-230-8450 ;
** Daria Iakovleva. Tel.: +7-911-034-3466 ;
E-mail address: alpopov@corp.ifmo.ru

1877-0509
10.1016/j.procs.2018.08.264

learning research. Many online advertising platforms, such as Criteo[1], Google[2], Facebook[3] (i. e. [5]) and Yahoo[4] are engaged in research to improve the quality of user behaviour comprehension and, thus, to improve the quality of life of all process participants. The research aims to improve the efficiency of ad campaigns for advertisers, the revenue of platforms and the happiness of users.

There is a variety of different online advertising platforms: search services, display advertising networks, social networks, mobile and video platforms. Each platform has its unique targeting features. In the case of a social network, search for target audience usually exploits knowledge about its users' personality and behavior. Useful knowledge can be obtained from explicit user feedback, i.e. subscriptions to communities or likes.

Different approaches exist for a searching of a target audience for online-advertising. The most common method of targeting is the search guided by a set of advertiser-defined settings. These settings are typically based on advertiser's assumptions on the target audience: sociodemographic and geo-based parameters, communities users belong to, or interests they may share. This approach may be classified as an expert-based and thus inherits all the constraints and limitations of expert-based methods [4]. Moreover, advertiser's expertise of their target audience commonly has low quality [11].

Usually, ad platforms exist in the form of an auction where advertisers are bidding for ad impressions.[20] The most common way of performing targeting automatically is to predict a conversion rate to target action (i.e. purchase, click or subscription to a page) for the user to decide, if is it profitable to show a given ad to the user or not. It is a major machine learning problem in online advertisement which is called CTR-prediction (click-through rate) or CR-prediction (conversion rate) problem. There are many different models which platforms use to solve the problem (i.e. linear models[1], decision trees ensembles[5], factorization machines[14], field-aware factorization machines[6], neural networks[21]). These models are based on a set of categorical features (i.e. user gender, age, geographical location; ad features such as domain and advertiser id) and their interactions between each others. Advertisers often have prior knowledge about related users, who already showed some kind of interest outside of advertising platform (for instance, they may be active customers or web page visitors). This data is useful and can be used for targeting automatization and improving a quality of feedback prediction algorithms. Such users can be delivered to advertisement platform via retargeting[7] mechanism.

One of the possible approaches to solve the task of building a new audience with the help of a users set that already known to be interested in an advertiser is the use of look-alike targeting models. They are based on an assumption that the user similarity correlates with the probability of receiving the same positive feedback on advertising. They allow running an advertising campaign requiring only an initial seed of users, who have already given the positive feedback. Thus, the targeting parameter selection process becomes automatic. Look-alike targeting models enable accessing to a much more complicated model of the target user that cannot be expressed in terms of simple targeting filters and, also, improve quality of CR and CTR models prediction.

In certain terms, the look-alike targeting models and recommender systems[15] solve similar problems, but there exists notable difference. For example, we denote communities in a social network as recommender system items. A new users set which was given by advertiser is a new pseudo-community and, respectively, a new item. Recommender systems usually operate in a fixed set of items and begin to recommend a new item only after whole model fitting. In look-alike models, each task is to find users who can like the new item and a result should be calculated in online mode. We cannot wait for a scheduled model re-fit. This is the reason why it is more reasonable to treat the problem as a semi-supervised learning task [23]: we need to learn a classifier given a feature description of all users and a tiny subset of users known as positive examples.

In such setting, the size of the dataset becomes the most important issue. The number of the users in an advertising network can reach up to hundreds of millions, at the same time sizes of an input seed set and an expanded set can be up to millions of users. The size of the sparse feature set, which the users are described with, can reach up to hundreds of millions features. These two factors constrain on methods that can be applied to the data and leave behind most of the efficient data preprocessing and semi-supervised techniques. Because of these constraints, static

[1] http://research.criteo.com/publications/
[2] https://ai.google/research/pubs/
[3] https://research.fb.com/
[4] https://research.yahoo.com/research-areas/advertising-science

distance-based models which estimate distance between the initial seed and every user are a common approach to the problem. This leads to high dependency of results on the quality and cleanliness of the initial seed. The success of an advertising campaign, which is targeted to the look-alike audience, can be strongly affected by a noise and different biases presented in the initial seed because of the dimensionality of the feature space, the limited number of positive examples, and the absence of vividly expressed negative examples.

A natural way to minimize the influence of those factors is to make the model learning with new experience gained with the feedback of users selected by the model on previous steps. However, the constraints described leave not much space for sophisticated models that can reflect complex dependencies. Users are represented by thousands of sparse features, only a few of which relate to their reaction on advertising, and the size of initial seed is too small to assume that feature importance evaluated on the seed can generalize well.

In this paper, we suggest a look-alike model that uses experience gained during an ongoing advertising campaign to explore, which features are truly important, and employs important features to select users, who should see an advertisement during the next stage of the campaign. Due to the exploration of sparse feature importance is performed by getting feedback from users, it collides with requirements of showing the advertisement to the users having the highest probability of the positive feedback Balancing between them is the well-known problem of exploration vs exploitation trade-off [19], and we use Thompson sampling [16] algorithm to handle it.

The contributions of this paper are the following.

1. We introduce a concept of the adaptive look-alike model, which incorporate iterative exploratory approach to look-alike modelling.
2. We suggest implementation of the model in the context of a social network advertisement.
3. We provide empirical research of the model suggested and compare the model with the baseline models on the dataset from real large-scale advertising campaigns from the advertising platform of the social network VK[5].

The rest of the paper is organized as follows. In Section 2, we overview related papers. In Section 3, we define problem formulations and terms. In Section 4 we describe proposed algorithm. Section 5 contains all results of an empirical research of the suggested model and a comparison of the suggested model with the baseline models. A conclusion is in a Section 6.

Nomenclature

S	Users seed set
U	A universal set of available users
R	Expanded set of users, based on seed similarity
$\mathcal{X}$	Set of available actions "Add user k to expanded set R"
$y_k \in \{1, 0\}$	Feedback of user k. Can be positive(1) or negative(0)
$r(y_k)$	Reward function
$M_\theta(k)$	A model with parameters θ which returns look-alike score of user k
F_k	Observed features vector of user k
f_i	Single feature from feature vector F
$P(f_i \mid S)$	A probability of observing feature i in a random user drawn from S
R^+	Frequency of observed positive feedback users
IV	Information Value, describes the power of features
p_i, q_i	The proportion of positive and negative samples, respectively, with feature i

[5] https://vk.com/

2. Related Works

Look-alike models described in several different ways and implemented in many advertising networks, including Facebook[6] and Google AdWords[7]. Quality evaluation of look-alike targeting has no common baseline and benchmark dataset. It differs in related works, so we also describe different approaches to quality estimation.

2.1. Existing Look-alike Models

The most widely spread approach to the problem is to solve an isolated problem of searching an expanded audience that consist of similar users.

Semi-supervised classification models [3, 9, 13] use suitable classifier as a base of a model, which is trained on positive examples from the initial seed. A random sample from the user set without positive feedback from previous campaigns can be used as negative examples. After the classifier is learned, all users are classified by their feedback, and top $|R|$ users with the highest probability of belonging to the positive class are taken as the result.

Rule-based models [10, 17] use special rules for assigning weights to every feature evaluated on the initial seed. All users that have best features are taken as the result.

Recommendation based models [17, 9] consider look-alike modeling as a recommendation problem and solve the problem by expanding the user set by applying similarity measures.

Hybrid models [8] combine strong sides of several approaches using pair-wise similarities and weighting feature importance at the same time. Locality-sensitive hashing (LSH) [18] is used for quick evaluation of user similarities.

Transfer learning models [22] operate in a composition with CTR prediction model, using transfer learning. These models are created specially for improving CTR-prediction performance. They do not expand any explicit user set, rather utilize knowledge about user browsing behaviour.

All the approaches described build model statically without using knowledge obtained during online campaigns. General framework for **adaptive look-alike modelling** [2] defines several methods of user profiling and look-alike modelling combined with iterative model update.

2.2. Quality Evaluation Methods

App conversion log: as a dataset, authors of [9] use app install log of 100 apps, divided to train and test parts by time. The user features include apps installed, app categories, app usage activities, and app meta information. Users, who installed an app during the train part of the log, are taken as the seed set, while others are considered as negative examples. Authors measure conversion to install rate of users. It is calculated based on expanded set users from the test part of the log.

Conversion log: authors of [8] also use a conversion log, divided by time. Unfortunately, authors do not mention specific user features, however specify that it is a set of sparse binary features with millions of features magnitude.

Online A/B tests: authors of [17] expand audiences of real advertising campaigns and measure a set of standard CTR-prediction metrics such as reach or CTR during large-scale online A/B testing.

Online display ad logs: authors of [22] use AUC and RMSE, measuring the improvement of CTR-prediction model performance on real-world dataset. It consists of two weeks of online display ad logs across different campaigns and users browsing history.

3. Look-alike Targeting Problem Formulation and Terms

We denote an initial user seed as S and a universal set of available users as U. U does not include users from S. The advertising campaign process is the environment which is able to give us a reward in a form of positive user feedback (i.e. a click, a full video view or an another target action like a purchase if an advertisement network supports cost per

[6] https://www.facebook.com/business/help/164749007013531
[7] https://support.google.com/adwords/answer/2676774?hl=en

acquisition (CPA) campaigns). Our goal is to run a campaign as effective as possible, i.e. maximize the reward from the environment on an fixed size expanded user set R using similar users search.

Estimated set sizes for a dataset:

- for $|S|$: 1, 000–100, 000.
- for $|R|$: 100, 000–1, 000, 000,
- for $|U|$: 100, 000, 000.

The set of features which represents users is distinguished by a large dimensionality and a sparsity (on average, only 100 features out of 2,000,000 features are presented for a certain user).

Features represent different indicators of an user behaviour and interests. For example, they could be indicators if the user belongs to a certain community or other relationships in a social graph for social networks. In the case of web-based advertising networks, they can be represented by visited web resources. User representation can be done via different variants of feature set factorization or using complex quantitative features (for example, weighted frequency of different interactions with community or web-resource with time-based decay coefficient). For a simplicity sake, we restrict our model to only binary features since the representation of a user in such models is a distinct topic beyond this paper.

4. Adaptive Look-alike Targeting Algorithm

The main idea of the adaptive look-alike targeting algorithm is to define the user score function via model, which has uncertainty in parameters. Thompson Sampling allows to use degree of uncertainty as managing exploration-exploitation trade-off mechanism.

We define the problem using reinforcement learning notation. The goal of the agent is to choose a sequence of actions from $\mathcal{X}$. $\mathcal{X}$ consists of actions x_k which means "Add user k to expanded set R". After adding a user we observe an output $y_k \in \{0, 1\}$ and earn a reward $r(y_k) = [y_k = 1]$, where [] is the indicator function. We find the value of k using the expression 1, where $M_\theta(k)$ is a model with parameters θ which returns look-alike score of user k.

$$k = \arg\max_{x_k \in \mathcal{X}} P(r(y_k) \mid y_k) = \arg\max_{x_k \in \mathcal{X}} M_\theta(k) \tag{1}$$

We use Bernoulli Naive Bayes as a base of the look-alike model to classify users, whether they should be in the S or U. On each step the model evaluates actions from $\mathcal{X}$ and chooses the best action x_{best}.

$$x_{best} = \arg\max_{x_k \in \mathcal{X}} M_\theta(k) \tag{2}$$

$$x_{best} = \arg\max_{x_k \in \mathcal{X}} \frac{P(k \in S \mid k)}{P(k \in U \mid k)} \tag{3}$$

$$x_{best} = \arg\max_{x_k \in \mathcal{X}} \frac{P(k \in S \mid F_k)}{P(k \in U \mid F_k)} \tag{4}$$

$$x_{best} = \arg\max_{x_k \in \mathcal{X}} \frac{\dfrac{P(F_k \mid k \in S) \cdot P(k \in S)}{P(F_k)}}{\dfrac{P(F_k \mid k \in U) \cdot P(k \in U)}{P(F_k)}} \qquad P(k \in S) \textit{ and } P(k \in U) \textit{ are constants } \forall k \tag{5}$$

$$x_{best} = \arg\max_{x_k \in \mathcal{X}} \frac{P(F_k \mid k \in S)}{P(F_k \mid k \in U)} \quad \textit{using conditional independence assumptions} \tag{6}$$

$$x_{best} = \arg\max_{x_k \in \mathcal{X}} \frac{\prod_{f_i \in F_k} P(f_i \mid S)}{\prod_{f_i \in F_k} P(f_i \mid U)} \quad \textit{applying log function} \tag{7}$$

$$x_{best} = \arg\max_{x_k \in \mathcal{X}} \Big(\sum_{f_i \in F_k} \log(P(f_i \mid S)) - \sum_{f_i \in F_k} \log(P(f_i \mid U))\Big) \tag{8}$$

We consider $P(f_i \mid k \in S)$ and $P(f_i \mid k \in U)$ as model parameters $\theta = \{\theta_{i,1|0}\}, \theta_{i,1} = P(f_i \mid k \in S), \theta_{i,0} = P(f_i \mid k \in U)$, which are estimated from given sets S and U. Each parameter is Beta distributed random variable $Beta(\alpha,\beta)$, where α is the number of observed positive outcomes and β is the number of observed negative outcomes.

The agent is initially uncertain about the value of θ and represents his uncertainty using a prior distribution. We calculate priors for every model parameter:

$$P(f_i \mid S) \sim Beta(\alpha_{\theta_{i,1}}, \beta_{\theta_{i,1}}) \qquad \alpha_{\theta_{i,1}} = |S_i| \qquad \beta_{\theta_{i,1}} = |S| - |S_i| \tag{9}$$

$$P(f_i \mid U) \sim Beta(\alpha_{\theta_{i,0}}, \beta_{\theta_{i,0}}) \qquad \alpha_{\theta_{i,0}} = |U_i| \qquad \beta_{\theta_{i,0}} = |U| - |U_i| \tag{10}$$

As observations are collected, the distribution is updated according to Bayes' rule. In particular, posterior distribution of each feature is also $Beta(\alpha,\beta)$ with parameters that can be updated according to the following rule:

$$P(f_i \mid S) \sim Beta(\alpha_{\theta_{i,1}} + F_{k,i}, \beta_{\theta_{i,1}} + (1 - F_{k,i})) \text{ if } y_k = 1 \qquad F_{k,i} = \text{value of feature i in } F_k \tag{11}$$

$$P(f_i \mid U) \sim Beta(\alpha_{\theta_{i,0}} + F_{k,i}, \beta_{\theta_{i,0}} + (1 - F_{k,i})) \text{ if } y_k = 0 \tag{12}$$

The main idea of Thompson sampling is random sampling of θ at every iteration of the algorithm instead of using $\mathbb{E}(\theta)$. We run advertising campaign using Algorithm 1 listed below. The best user is chosen at every iteration of the algorithm, but users sets of appropriate size can be chosen as well without significant modifications of the algorithm.

5. Experiments

Anonymous user data from social network VK are provided for testing the algorithm. We split a campaign feedback to two parts: train and test. Seed set consists of users from the train set that gave positive feedback. All users in test set represent U. The task is to sort users in order of simulated U traversal and maximize the reward over first $|R|$ users.

5.1. Dataset

Anonymous user data from social network VK are provided for testing the algorithm. The dataset consist of:

Users features dataset. The dataset has vectors of features for a set of social network users. Features represent subscriptions to communities pages and subscriptions to users with large number of subscribers. Specific properties of the dataset approximately:

Algorithm 1 Thompson sampling adaptive look-alike targeting algorithm

```
procedure InitializePriors
    for i ∈ F do
        α_{θ_{i,1}} ← 0
        β_{θ_{i,1}} ← 0
        α_{θ_{i,0}} ← 0
        β_{θ_{i,0}} ← 0
procedure UpdatePriors(S, U, F)
    for k ∈ S do
        for i ∈ F do
            if i ∈ F_k then
                update ← 1
            else
                update ← 0
            α_{θ_{i,1}} ← α_{θ_{i,1}} + update
            β_{θ_{i,1}} ← β_{θ_{i,1}} + (1 − update)
    for k ∈ U do
        for i ∈ F do
            if i ∈ F_k then
                update ← 1
            else
                update ← 0
            α_{θ_{i,0}} ← α_{θ_{i,0}} + update
            β_{θ_{i,0}} ← β_{θ_{i,0}} + (1 − update)
procedure SampleParameters(F)
    for i ∈ F do
        Generate θ_{i,1} : θ_{i,1} ∼ Beta(α_{i,1}, β_{i,1})
        Generate θ_{i,0} : θ_{i,0} ∼ Beta(α_{i,0}, β_{i,0})
    return θ
procedure RunCampaign(S, U, F, RSize)
    InitializePriors()
    UpdatePriors(S, U, F)
    R ← ∅
    for 1 to RSize do
        θ ← SampleParameters()
        BestUser ← arg max_{k∈U}(M(k, θ))
        if Feedback[BestUser] = 1 then
            S_update ← {BestUser}
            U_update ← ∅
        else
            S_update ← ∅
            U_update ← {BestUser}
        R ← R ∪ {BestUser}
        UpdatePriors(S_update, U_update, F)
```

1. 10^8 users
2. $3 \cdot 10^6$ features
3. 100 features per user on average.
4. 500 — minimal size of community to be represented as feature

5. 1000 — minimal subscribers count for a user to be represented as feature

Advertisement campaigns logs dataset. The log consists of records whether specific user clicked on a ad. Specific properties of the log approximately:

1. 119 real large-scale ad campaigns which were going for a two weeks in May, 2018 in VK advertising network.
2. 2000 clicks is a minimal number of clicks in each log.
3. Campaigns to test were chosen via specific criteria: campaigns with maximum reach, maximum CTR or minimal CTR.
4. 10^6 unique users with known feedback on average per campaign.

5.2. Metrics

The first metric of the algorithm quality is the ratio of the positive examples percentage of the selected users R set to the positive examples percentage of the whole set U. We call it Normalized CTR.

$$\texttt{nCTR} = \frac{|R^+|}{|U^+|} \tag{13}$$

We calculate it for a set of fixed $|R|$, expressed through a percentage of $|U|$: 1%, 5%, 10%, 50%.

The second metric is AUC, which shows the quality of users ranking and classification. It is convenient to use AUC, assuming we do not need specific users scores.

5.3. Scoring Methods

We applied the following set of methods through the evaluation.

- `BaselineIV` is method proposed in [8]. We choose this method as since the setting of experiment and data are very similar.
- `BaselineDiffLog` is `BaselineIV` with modified "Score by Feature Information Value" part:

$$IV(f, y) = \log\left(\frac{p_i + \epsilon}{q_i + \epsilon}\right) \tag{14}$$

  Additionally, we use simple feature selection technique to decrease the number of noisy and irrelevant features: only features, which were observed in $|S|$ at least $|S|/C$ times and have a positive IV remain in dataset. C is empirically chosen constant. It also can be selected via cross-validation.
- `BaselineDiffLogTopFeatures` is `BaselineDiffLog`, but score for every user calculates over top C user features with highest IV scores. C is empirically chosen constant. It also can be selected via cross-validation.
- `BaselineDiffLogActivityBias` is `BaselineDiffLogTopFeatures`, but before feature selection all IV scores are shifted on `ActivityBias` value:

$$\texttt{ActivityBias} = \frac{\dfrac{\text{number of positive features } \forall \text{ users} \in S}{|S|}}{\dfrac{\text{number of positive features } \forall \text{ users} \in U}{|U|}} \tag{15}$$

The intuition behind `ActivityBias` is the fact, that, in general, users from S are more active and subscribe to more communities than random users on average, as the advertiser found a way to collect such users. So, there are many irrelevant features with positive IV score which we filter using this heuristic.

- `AdaptiveLookAlikeStatic` is proposed model without iterations and θ sampling. Instead of sampling $\mathbb{E}(\theta)$ is used as θ. All calculations are performing using prior distribution, calculated by S an U
- `AdaptiveLookAlikeGreedy` is proposed model without θ sampling.
- `AdaptiveLookAlike` is proposed model.

5.4. Proposed method evaluation

The Proposed method evaluation results are presented in Table 1.

Table 1. The Proposed method evaluation results.

Method	AUC	nCTR (1%)	nCTR (5%)	nCTR (10%)	nCTR (50%)
BaselineIV	0.55	2.35	1.76	1.54	1.14
BaselineDiffLog	0.5838	2.39	1.95	1.73	1.23
BaselineDiffLogTopFeatures	0.5957	**3.27**	**2.29**	**1.95**	1.25
BaselineDiffLogActivityBias	0.5961	3.23	2.28	1.93	1.26
AdaptiveLookAlikeStatic	0.6067	2.50	2.07	1.80	1.30
AdaptiveLookAlikeGreedy	0.6112	2.55	2.08	1.87	1.32
AdaptiveLookAlike	**0.6191**	2.61	2.14	1.9	**1.33**

Proposed method reaches top scores by AUC and nCTR(50%), thus, it performs better at large R and continuous advertising campaigns. `BaselineDiffLogTopFeatures` reaches top scores by nCTR over small $|R|$. In case of searching for a small look-alike audience, `BaselineDiffLogTopFeatures` provides best results.

6. Conclusion

In this paper, we have presented the concept of the adaptive lookalike targeting and the model that uses a reinforcement learning technique to maintain adaptiveness of the model via working with users' features independently.

The model has achieved the improved performance in comparison to the baseline method by 12.5% in AUC using a Thompson Sampling model. This improvement encourages us to continue studying applications for reinforcement learning to the online advertising targeting problem.

It is possible to create automatic advertising campaigns using the algorithm presented, since at each iteration the algorithm is retrained and obtains new knowledge about users and their features.

6.1. Future Work

The model should respect an advertiser's budget (which is determines $|R|$) and adapt exploration/exploitation trade-off when there are few resources to explore features. The algorithm described does not use knowledge about $|R|$ and that is why it has average results by $nCTR$ over small R sizes — the algorithm lacks a stop point as a parameter. We use a quite simple variant of reinforcement learning approach. More sophisticated approaches could give better results.

As noticed above, the user representation is a pretty interesting field of research and it should be studied in connection with online reinforcement learning part.

We test the algorithm on already finished campaigns and the environment can not give users feedback immediately in real campaign scenario. Some users can see an advertising only after a while. Unfortunately, the need to update scores for all users and lack of immediate feedback make the algorithm impractical in online production environment. However, we are able to redefine the task to a closer one to online advertising auctions. Instead of solving the sorting or expanding task, we can solve the following task: predict, if it is profitable to show the ad for a specific user to reach the advertising campaign global goal, i.e cost per conversion.

Some advertising networks support negative feedback, for example, ads hiding and reporting. We definitely can use it in the model as a proper negative signal which correlates with users frustration from advertising.

The most important input (i.e. seed set) comes from the advertiser and it could be initially broken which can doom a whole campaign at the very beginning. Therefore, an input should be validated so that advertiser does not run campaigns that would gone wrong (the model knows this before the start).

Advertising networks run and store a lot of campaigns. We can use this knowledge when building the model and use previous related campaigns experience. This actually leads the whole model to fully automatic approach, depending on an object of advertising. For example, if the advertiser wants to promote a community, current members or members of similar communities can be used as a seed set. If the advertiser wants to promote some service, we could match related campaigns by ad content (or just by previous advertiser's campaigns) and collect seed from experience.

References

[1] Agarwal, A., Chapelle, O., Dudík, M., Langford, J., 2014. A reliable effective terascale linear learning system. The Journal of Machine Learning Research 15, 1111–1133.

[2] Bagherjeiran, A., Tang, R., Zhang, Z., Hatch, A., Ratnaparkhi, A., Parekh, R., 2015. Adaptive targeting for finding look-alike users. US Patent 9,087,332.

[3] Chakrabarti, D., Agarwal, D., Josifovski, V., 2008. Contextual advertising by combining relevance with click feedback, in: Proceedings of the 17th international conference on World Wide Web, ACM. pp. 417–426.

[4] Giarratano, J.C., Riley, G., 1998. Expert systems. PWS Publishing Co.

[5] He, X., Pan, J., Jin, O., Xu, T., Liu, B., Xu, T., Shi, Y., Atallah, A., Herbrich, R., Bowers, S., et al., 2014. Practical lessons from predicting clicks on ads at facebook, in: Proceedings of the Eighth International Workshop on Data Mining for Online Advertising, ACM. pp. 1–9.

[6] Juan, Y., Zhuang, Y., Chin, W.S., Lin, C.J., 2016. Field-aware factorization machines for ctr prediction, in: Proceedings of the 10th ACM Conference on Recommender Systems, ACM. pp. 43–50.

[7] Lambrecht, A., Tucker, C., 2013. When does retargeting work? information specificity in online advertising. Journal of Marketing Research 50, 561–576.

[8] Ma, Q., Wagh, E., Wen, J., Xia, Z., Ormandi, R., Chen, D., 2016a. Score look-alike audiences, in: Data Mining Workshops (ICDMW), 2016 IEEE 16th International Conference on, IEEE. pp. 647–654.

[9] Ma, Q., Wen, M., Xia, Z., Chen, D., 2016b. A sub-linear, massive-scale look-alike audience extension system, in: Proceedings of the 5th International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications.

[10] Mangalampalli, A., Ratnaparkhi, A., Hatch, A.O., Bagherjeiran, A., Parekh, R., Pudi, V., 2011. A feature-pair-based associative classification approach to look-alike modeling for conversion-oriented user-targeting in tail campaigns, in: Proceedings of the 20th international conference companion on World wide web, ACM. pp. 85–86.

[11] Nagiah, N., 2014. 72 percent of marketers cant find their target audience online — can you? http://www.huffingtonpost.com/navin-nagiah/72-of-marketers-cant-find_b_5324677.html.

[12] Plummer, J., Rappaport, S.D., Hall, T., Barocci, R., 2007. The online advertising playbook: proven strategies and tested tactics from the advertising research foundation. John Wiley & Sons.

[13] Qu, Y., Wang, J., Sun, Y., Holtan, H.M., 2014. Systems and methods for generating expanded user segments. US Patent 8,655,695.

[14] Rendle, S., 2010. Factorization machines, in: Data Mining (ICDM), 2010 IEEE 10th International Conference on, IEEE. pp. 995–1000.

[15] Ricci, F., Rokach, L., Shapira, B., 2011. Introduction to recommender systems handbook, in: Recommender systems handbook. Springer, pp. 1–35.

[16] Russo, D., Van Roy, B., Kazerouni, A., Osband, I., 2017. A tutorial on thompson sampling. arXiv preprint arXiv:1707.02038 .

[17] Shen, J., Geyik, S.C., Dasdan, A., 2015. Effective audience extension in online advertising, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM. pp. 2099–2108.

[18] Slaney, M., Casey, M., 2008. Locality-sensitive hashing for finding nearest neighbors [lecture notes]. IEEE Signal Processing Magazine 25, 128–131.

[19] Sutton, R.S., Barto, A.G., 1998. Reinforcement learning: An introduction. volume 1. MIT press Cambridge.

[20] Wang, J., Zhang, W., Yuan, S., et al., 2017a. Display advertising with real-time bidding (rtb) and behavioural targeting. Foundations and Trends® in Information Retrieval 11, 297–435.

[21] Wang, R., Fu, B., Fu, G., Wang, M., 2017b. Deep & cross network for ad click predictions. arXiv preprint arXiv:1708.05123 .

[22] Zhang, W., Chen, L., Wang, J., 2016. Implicit look-alike modelling in display ads, in: European Conference on Information Retrieval, Springer. pp. 589–601.

[23] Zhu, X., 2011. Semi-supervised learning, in: Encyclopedia of machine learning. Springer, pp. 892–897.