Two-Stage Audience Expansion for Financial Targeting in Marketing

Zhining Liu*, Xiao-Fan Niu*, Chenyi Zhuang, Yize Tan, Yixiang Mu, Jinjie Gu, Guannan Zhang
Ant Financial Services Group
{eason.lzn,xiaofan.nxf,chenyi.zcy,yize.tyz,yixiang.myx,jinjie.gujj,zgn138592}@antgroup.com

ABSTRACT

With the revolution of mobile internet, online finance has grown explosively. In this new area, one challenge of significant importance is how to effectively deliver the financial products or services to a set of target users by marketing. Given a product or service to be promoted and a set of users as seeds, audience expansion is such a targeting technique, which aims to find potential audience among a large number of users. However, in the context of finance, financial products and services are dynamic in nature as they co-vary with the socio-economic environment. Moreover, marketing campaigns for promoting products or services always consist of different rules of play, even for the same type of products or services. As a result, there is a strong demand for the timeliness of seeds in financial targeting. Conventional one-stage audience expansion methods, which generate expanded users by expanding over seeds, would encounter two problems under this setting: (1) the seeds would inevitably involve a number of users that are not representative for expansion, and direct expansion over these noisy seeds would dramatically deteriorate the performance; (2) one-stage expansion over fixed seeds cannot timely and accurately capture users' preferences over the currently running campaign due to the lack of timeliness of seeds.

To address the above challenges, in this paper, we present a novel two-stage audience expansion system — *Hubble*. In the first cold-start stage, a reweighting mechanism is devised to suppress the noises within seeds, which is motivated from the observation on the relationship between golden seeds and their corresponding density in the embedding space. With incrementally collecting feedbacks from users, we further include these feedbacks to guide subsequent audience expansion in the second stage. But the distribution of these feedbacks is usually biased and cannot fully characterize the distribution of all target audiences. Therefore, we propose a method to incorporate biased feedbacks with seeds in a meta-learning manner to pan for golden seeds from the noisy seed-set. Finally, we conduct extensive experiments on three real datasets and online A/B testing, which demonstrate the effectiveness of the proposed method. In addition, we release two datasets for boosting the study of this new research topic.

*Both authors contributed equally to this research.

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3412748

CCS CONCEPTS

• **Applied computing → Marketing**.

KEYWORDS

audience expansion, financial targeting, marketing, noisy labels

ACM Reference Format:

Zhining Liu*, Xiao-Fan Niu*, Chenyi Zhuang, Yize Tan, Yixiang Mu, Jinjie Gu, Guannan Zhang. 2020. Two-Stage Audience Expansion for Financial Targeting in Marketing. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3340531.3412748

1 INTRODUCTION

By the end of 2018, there are about 5.1 billion unique mobile subscribers around the world[1]. The innovative mobile economy is gaining more and more attention. In such a new market having a large number of users, the effectiveness of delivering diverse financial products and services to the right audience is crucial in market competitions [13]. Hence, audience targeting is becoming one of the key techniques in in Customer Relationship Management (CRM) [2].

Audience expansion (also known as look-alike), which has been widely applied in Ad serving systems (e.g., Yahoo [19], Linkedin [16], Pinterest [5], WeChat [17]), is such a technique to establish the connections between a campaign for promotion and the audiences who are interested in it. Given a product for promotion, as well as a set of users as seeds (known as *seed-set*), audience expansion aims to increase the reach of the product by displaying ads (campaigns) to audiences with similar attributes to the seed-set. The seed-set is usually selected based on the analysis of CRM. For instance, CRM could help us pre-define some rules for audience filtering to generate the seed-set, or retrieve target audience on similar past campaigns as seeds.

However, in the context of finance, the dynamic environment puts new requirements on CRM. Roughly speaking, we conclude with the following two aspects: (1) financial products and services are dynamic in nature as they co-vary with the socio-economic environment, which suggests that the seeds would be outdated in a short period; (2) in a competitive market, a marketing campaign of promoting products or services always consists with innovative rules of play, and these rapidly changing rules would dramatically reduce the effectiveness of the seeds. As a result, there is strong a demand for timeliness of seeds in financial targeting. Conventional one-stage based audience expansion methods, which generate expanded users by expanding over seeds, would encounter

[1]https://wearesocial.com/blog/2019/01/digital-2019-global-internet-use-accelerates

two problems under this setting: (1) the seeds would inevitably involve a number of users that are not representative for expansion, and direct expansion over these noisy seeds would dramatically deteriorate the performance; (2) one-stage expansion over the fixed seeds cannot accurately capture users' preferences over the currently running campaign due to the lack of timeliness of the seeds. Hence, given a new financial campaign running for several days, in addition to the seeds retrieved at the beginning, the targeting system should have the ability to continuously refine the audience expansion process throughout the whole campaign period.

To solve the aforementioned problems, in this paper, we present a two-stage audience expansion system for financial targeting. In the cold-start stage, we propose a reweighting method to adjust sample weights based on the observation on the relationship between golden seeds and their corresponding density in the embedding space. Then, as the expansion audience is going on, a group of users with positive (e.g., click and conversion) and negative feedbacks are included to guide subsequent audience expansion. But the distribution of these feedbacks is usually biased and cannot fully characterize the distribution of all target audiences. Therefore, we further propose a method to incorporate biased feedbacks with seeds in a meta-learning manner to pan for golden seeds from the noisy seeds. As a result, our audience expansion system is able to adjust the expansion strategy throughout the entire campaign period, which is crucial in financial targeting. In summary, our main contributions are as follows:

- **Problem.** In the context of finance, we first identify challenges in financial targeting and formally define a new problem of audience expansion with a noisy seed-set and user feedbacks .
- **Method.** We present a novel two-stage targeting system to fully take advantage of both noisy seed-set and feedbacks collected as a campaign is going on.
- **Evaluation.** We perform extensive experiments on three real datasets and online A/B testing, which demonstrates that the proposed framework achieves consistent improvement in financial targeting. In addition, we release two real-world datasets for boosting the study of this new research direction.

The rest of the paper is organized as follows. In Section 2, we first introduce the related work. Then, our proposed framework and experimental results are presented in Section 3 and Section 4, respectively, before we conclude the paper in Section 5.

2 RELATED WORK

After surveying on the related audience expansion methods, generally speaking, there exist two main categories:

Similarity-based Methods. Similarity-based methods expand a given seed-set via calculating the similarity of all pairs between seed users and candidate users. The similarity function is usually computationally efficient, such as vector-based dot product, Euclidean distance, Jaccard index, and so on. Then, on the basis of similarity scores, top-ranked users are selected as the target audience. In spite of its high efficiency, the performance of this kind of methods heavily depends on the manual feature engineering.

To tackle the shortcoming, user representation learning, which generates low-dimensional and representative embeddings based on users' demographic profiles and activity logs, is introduced, which improved the effectiveness of similarity calculation. For instance, [6] proposed an adversarial factorization auto-encoder that generates binary user representations for better computational efficiency and encodes higher-order feature interactions. [17] proposed to learn user representations using an attention-based merge mechanism under the framework of Youtube DNN [4]. On the other hand, the selection of similarity functions also attracts a number of researchers. For instance, Locality-Sensitive Hashing (LSH) [23] is a widely used method for reducing the cost of computing user similarity. On the basis of LSH, by considering feature importance and distributions of non-seed users, two improved similarity calculation methods are proposed by [18] and [5], respectively.

Model-based Methods. In this category, machine learning methods are utilized to predict users' degrees of membership of belonging to a seed-set. Given a seed set and a candidate set, positive samples are selected from the seed-set, while negative samples are commonly sampled from the difference set of the candidate and seed sets. When there are enough training samples, model-based methods usually perform better than similarity-based methods, since a machine learning model could capture more complicated correlations between the input (i.e., user raw features) and output (i.e., degree of membership). Below, we list several typical methods.

The Logistic Regression (LR) model [22] is a simple but effective model. Then, [6] explored further to replace LR with other more powerful classifiers, like one-class SVM [20], factorization machine [24]. Since the selection of positive and negative samples is crucial for a model-based method, several studies are also conducted for guidance. Among them, in terms of audience expansion performance, [12] studied the impact of sampling ratios and sampling techniques. Considering the cold-start problem with a limited number of training samples, [5] proposed a blending method, which takes both similarity-based and model-based techniques into consideration. Specifically, a similarity-based method is devised to generate more diverse audiences, which can prevent the trained model from being sensitive to the size of seed-set.

However, to the best of our knowledge, few studies are conducted on studying financial targeting with a strong demand for the timeliness of seeds. This special requirement makes conventional methods easily fail, and motivates us to carefully deal with the noisy seed-set and introduce user feedbacks to guide the expansion. Facing these new challenges raised in financial targeting, we present a two-stage targeting system in the next section.

3 THE PROPOSED METHOD

In this section, we first formally define the problem of audience expansion in financial targeting, then illustrate our proposed methods in detail.

3.1 Problem Formulation

Given a seed-set $\mathcal{S}$ and a candidate-set C, audience expansion aims to extend $\mathcal{S}$ via selecting m users $\mathcal{E}$ (usually $|\mathcal{S}| << |C|$) from C, such that the extended users $\mathcal{E}$ are similar to $\mathcal{S}$. In this problem, each user i is usually represented by a d-dimensional vector $x_i \in \mathbf{X}$ that encodes the information of users' demographic profiles and online behaviors.

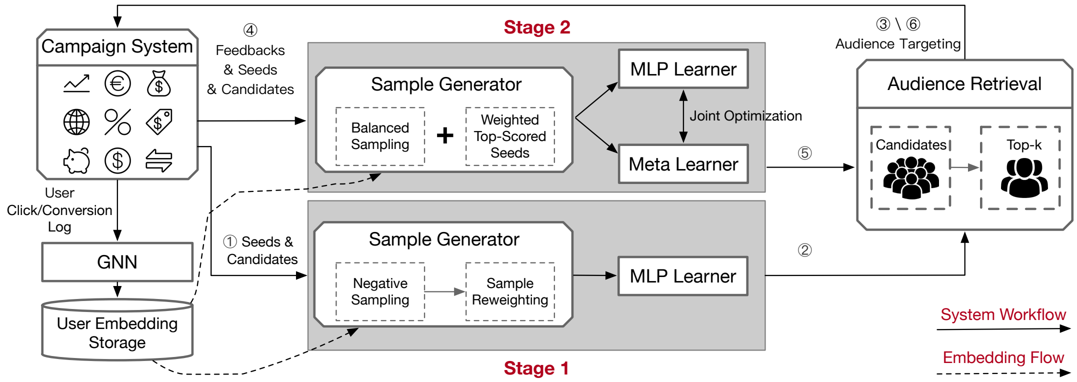


Figure 1: The architecture of our targeting system.

In the context of financial marketing, throughout a whole audience expansion process, there exist the following two stages. At the beginning of audience expansion, only the seed-set is available. Therefore with the above notations, the problem in this stage can be formally defined as follows:

PROBLEM 1. *Cold-start with Noisy Seed-set*

Input: *User representations* $\mathbf{X}$*, noisy seed-set* $\mathcal{S}$*, candidate-set* C*, the number of users to be expanded* m*.*

Output: *Expanded user set* $\mathcal{E}$*.*

As the process is going on, we can then obtain user feedbacks incrementally to guide subsequent audience expansion. As a result, we encounter the second problem:

PROBLEM 2. *Expansion with Incremental Feedbacks*

Input: *User representations* $\mathbf{X}$*, noisy seed-set* $\mathcal{S}$*, candidate-set* C*, the number of users to be expanded* m *and a group of users with feedback* $\mathcal{F}$*.*

Output: *Updated user set* $\mathcal{E}$*.*

To solve the problems above, correspondingly, we divide our solution into two stages: (1) a cold-start stage in Section 3.4; and (2) an expansion stage with incremental feedbacks in Section 3.5.

3.2 System Overview

Figure 1 describes the whole workflow of our targeting system named as *Hubble* [27]. This system is implemented and deployed on the Alibaba Cloud [2] since October 2019. In Ant Financial Services Group, in order to promote the mobile financial services (e.g., Alipay, Ant Insurance, Ant Credit, Ant Cash, etc.), the system runs about hundreds of campaigns toward billions of users every day.

In details, all the data processing jobs (e.g., data cleaning, log reflowing, graph construction for graph neural network, etc.) are implemented based on the MaxCompute [3] platform, which is powerful enough to handle interactions between one billion users and tens of thousands of campaigns. As one input of our proposed method, user embeddings are generated by a graph neural network (GNN) based model proposed in [27]. With the user-campaign-interaction graph is iteratively constructed on the basis of click and conversion logs, the user embeddings are updated asynchronously. Then, given seeds, candidates, user embeddings and the followed feedbacks, an instance of the two-stage targeting algorithm is launched for each campaign task. Considering the massive amount of interactive data and Hubble's scalability, all the algorithms are implemented by PAI [4] — an industrial-grade distributed machine learning platform. In the following subsections, we will introduce the core part, i.e., two-stage audience expansion, of our system.

3.3 Preliminary

Before illustrating the proposed solution, we first briefly introduce: (1) how the user representation, which is one of the inputs of our method, is obtained; and (2) an expandability analysis that motivates us to study the noisy seed-set.

3.3.1 User Representation Learning. To obtain high-quality user representations, a common pre-processing approach [5, 9, 17] is to generate low-dimensional user embeddings based on users' online behaviors and demographic profiles. The learned user embeddings can effectively characterize users' preferences over different campaigns and greatly improve the efficiency of audience expansion.

In our case, we use a GNN [27] to learn user representations. We build a bipartite graph by interpreting users and campaigns as nodes and interactions (i.e., click and conversion) as edges. In addition, users' demographic profiles are used as node attributes. By randomly deleting a small portion of edges and sampling the same number of non-existent edges from the graph, we can define a task of link prediction to learn user representations, and the objective function of the GNN is defined as:

$$\mathcal{L}_x = -\sum_{i,j}[\mathbb{1}(y_i^j = +1)\log\sigma(x_i^T \cdot c_j) + \mathbb{1}(y_i^j = -1)\log\sigma(1 - x_i^T \cdot c_j)], \tag{1}$$

where $\mathbb{1}(\cdot)$ denotes the indicator function, $y_i^j \in \{-1, +1\}$ is the label that denotes the occurrence of user behaviors over campaigns, and x_i and c_j are the user and campaign vectors generated from a trainable graph neural network, respectively. After training, the learned user representations are served as $\mathbf{X}$ in this paper.

[2] https://www.alibabacloud.com/
[3] https://www.alibabacloud.com/product/maxcompute
[4] https://www.alibabacloud.com/product/machine-learning

Since $\mathbf{X}$ is the input of our method, we omit the detailed information here. In addition, the DSSM [11] and Youtube DNN [4] are also good alternatives.

3.3.2 Expandability Analysis. Most audience expansion methods equally treat all seed users, but we argue that in financial targeting, users in $\mathcal{S}$ are usually with different expandability, which suggests that not every user is representative for expansion. To quantitatively measure the expandability of each user i in $\mathcal{S}$, we define a metric $E_k(i)$ as:

$$E_k(i) = \sum_{u \in \mathcal{N}_k(i)} \mathbb{1}(y_u = +1), \tag{2}$$

where $\mathcal{N}_k(i)$ denotes the k-nearest neighbors of i in the embedding space (here Euclidean distance is used), and $\mathbb{1}(y_u = +1)$ indicates whether its neighbor u is a target audience or not. Higher $E_k(i)$ means better expandability, as there are more target audience in its k-nearest neighbors. On the contrary, a user i with $E_k(i) = 0$ should be excluded in the process of expansion, since i has no expandability resulting in poor generalization ability. We consider this kind of users in $\mathcal{S}$ as noises.

To analyze the noises in the seed-set, we plot the histograms (black bars) of $E_{50}(i)$ over three online campaigns in Figure 2. The long tail distributions suggest that, owing to the timeliness and complexity of targeting in financial marketing, only a small portion of $\mathcal{S}$ is representative for expansion in a new campaign. However, in existing audience expansion techniques [12, 17], users in $\mathcal{S}$ are usually thought to having an equal expandability that is larger than zero, i.e., $\forall i, j \in \mathcal{S}, E_k(i) = E_k(j) > 0$.

In accordance with the analysis above, we present a more practical and general setting that is unlike $\forall i \in \mathcal{S}, E_k(i) > 0$, there is a certain probability of having $\exists i \in \mathcal{S}, E_k(i) = 0$ but under the constraint $\mathbb{E}_{i \in \mathcal{S}}[E_k(i)] > \mathbb{E}_{i \in C}[E_k(i)]$, which indicates that $\mathcal{S}$ should at least enjoy better expandability than C. To realize audience expansion under this new problem setting, in the following two subsections, we carefully devise two mechanisms to improve the performance of audience expansion in financial targeting.

3.4 Stage 1: Cold-start with Noisy Seed-set

By regarding all the users in $\mathcal{S}$ as positive samples and randomly sampling negative samples from the difference set of C and $\mathcal{S}$ [12], the loss of our initial model f is defined as a binary cross-entropy function:

$$\mathcal{L} = -\frac{1}{N}\Big[\sum_{i \in \mathcal{S}} \log f(x_i) + \sum_{j \sim (C - \mathcal{S})} \log(1 - f(x_j))\Big], \tag{3}$$

where N is the number of all samples.

Even though $\mathcal{S}$ is extremely noisy in our setting as shown in Figure 2, we first prove that the classification model (i.e., Eq. 3) trained on $\mathcal{S}$ tends to perform better than the one trained in a totally random way, which explains why it is still beneficial to study on the noisy seed-set.

Proposition 1. *If we assume the classifier f has a sufficiently high capacity, the classifier f_s trained on the noisy seed-set $\mathcal{S}$ has higher probability to predict the true label than f_r trained on a noisier seed-set $\mathcal{S}_r$.*

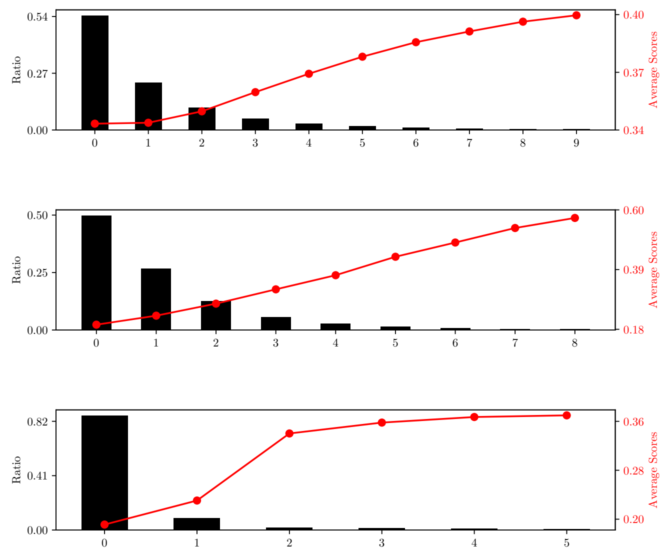


Figure 2: Histograms of expandability $E_{50}(i)$ on three real campaigns. The red curves give the average score of users in the same bin, and a large score indicates the corresponding user is located in a low-density area.

Proof. Let $\hat{y}$ denote the true labels, y^f denote the prediction labels and y denote the labels we observed. Then, let $\rho_{+1} = P(y = -1|\hat{y} = +1)$ denote the probability of true positive samples that are observed as negative, and $\rho_{-1} = P(y = +1|\hat{y} = -1)$ denote the probability of true negative samples that are observed as positive. Since the classifier f has a sufficiently high capacity of fitting its observed labels, we have $\rho_{+1} = P(y^f = -1|\hat{y} = +1)$ and $\rho_{-1} = P(y^f = +1|\hat{y} = -1)$ [3]. Then, the probability of the classifier correctly predicting the true label is

$$\begin{aligned} &P(y^f = \hat{y}) \\ &\quad = P(y^f = +1, \hat{y} = +1) + P(y^f = -1, \hat{y} = -1) \\ &\quad = P(y^f = +1|\hat{y} = +1)P(\hat{y} = +1) \\ &\qquad + P(y^f = -1|\hat{y} = -1)P(\hat{y} = -1) \\ &\quad = (1 - \rho_{+1})P(\hat{y} = +1) + (1 - \rho_{-1})P(\hat{y} = -1) \\ &\quad = 1 - \rho_{-1} + (\rho_{-1} - \rho_{+1})P(\hat{y} = +1). \end{aligned} \tag{4}$$

Note that $P(\hat{y} = +1) + P(\hat{y} = -1) = 1$. Due to that $P(\hat{y} = +1)$ is extremely small in targeting problems, $(\rho_{-1} - \rho_{+1})P(\hat{y} = +1)$ could be ignored in this equation. Since $\mathcal{S}$ contains less noises than $\mathcal{S}_r$, the possibility of labeling a true negative sample as positive (i.e., ρ_{-1} in Eq. 4) is lower. As a result, the function f_s trained on $\mathcal{S}$ will have a higher probability of predicting the true labels than f_r. □

After proving the effectiveness of using $\mathcal{S}$, by looking at Eq. 4, we aim to move one more step forward to make ρ_{-1} smaller. By observing the distributions shown in Figure 2, the users with good expandability usually are minority in comparison with the large amount of noises. This finding inspires us to further low ρ_{-1} from a perspective of rare category detection [10, 26], which propose to study these minorities based on their density in the data space. Therefore, we use the Isolation Forest algorithm [15], which is chosen for its linear time complexity and limited memory requirement, to calculate the density distribution in users' representation space $\mathbf{X}$.

With the normalized scores $\{v_i\}_{i\in\mathcal{S}}$ (a large score v_i indicates the user i is located in a low-density area) calculated by Isolation Forest, we calculate the average score of each group of users with the same $E_k(i)$, which gives the red curves in Figure 2. From the figure, we can easily observe that the group of users with large $E_k(i)$ tends to be located in the low-density areas (i.e., large v_i). It suggests that a group of users with strong personality is usually a rare category in financial targeting, which lies outside of the large community formed by the majority. Based on this observation, we design a reweighting mechanism by adaptively adjusting the weight of each positive sample. Specifically, the weights of negative samples remain constant (i.e., 1.0), while the weights of positive samples are set to $\{v_i\}_{i\in\mathcal{S}}$. Since the true negative samples in $\mathcal{S}$ are more likely to obtain lower weights, ρ_{-1} in Eq. 4 becomes smaller, which gives higher test accuracy. Hence, the final loss function in the cold-start stage is defined as:

$$\mathcal{L}_r = -\frac{1}{N}[\sum_{i\in\mathcal{S}} v_i \log f(x_i) + \sum_{j\sim(C-\mathcal{S})} \log(1-f(x_j))]. \quad (5)$$

Through optimizing the above objective function, we can use the optimal f^* to score each user in the C and select top-scored users as the expanded users.

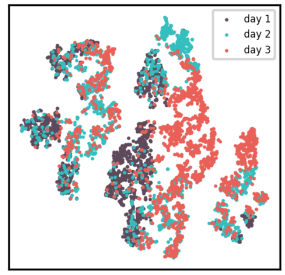
3.5 Stage 2: Expansion with Incremental Feedbacks

While a campaign is serving online, the targeting management system can collect incremental real feedbacks on the currently running campaign. Comparing with $\mathcal{S}$, these positive feedbacks usually enjoy a higher average expandability for their better ability to characterize the targeting audience. To validate this, expandability analysis on three online campaigns are presented in Table 1. Here we compare the expandability of the seed-set S and the user set $\mathcal{F}_1$ having positive feedbacks on the first day of the campaign. It is obvious that $\mathcal{F}_1$ has higher average expandability than $\mathcal{S}$. Therefore, according to Eq. 4, we can conclude that the classifier trained on the feedback is expected to have a higher test accuracy than the classifier trained on the noisy seed-set $\mathcal{S}$. From both of the positive and negative feedbacks, a balanced training data set $\mathcal{F}$ can be constructed and the corresponding loss function is defined as:

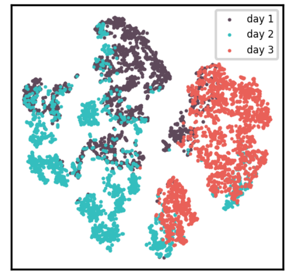
$$\mathcal{L}_d = -\frac{1}{|\mathcal{F}|}\sum_{(x_i,y_i)\in\mathcal{F}}[\mathbb{1}(y_i=+1)\log f(x_i) + \mathbb{1}(y_i=-1)\log(1-f(x_i))] \quad (6)$$

where $y_i = +1$ and $y_i = -1$ denotes positive and negative feedbacks, respectively.

However, the distribution of $\mathcal{F}$ in the first few days is usually biased due to limited collected feedbacks. Figure 3 presents two examples on how the feedbacks collected on different days distribute over the user embedding space. We can find that feedbacks collected in a single day cannot fully characterize the distribution of all potential audiences, which easily leads to a coverage bias problem.



(a) Campaign A



(b) Campaign B

Figure 3: Visualization of the users with positive feedbacks on different days of the marketing campaign in the embedding space.

Table 1: The average expandability score of $\mathcal{S}$ and $\mathcal{F}_1$.

Campaign	$\mathbb{E}_{i\in\mathcal{S}}[E_{50}(i)]$	$\mathbb{E}_{i\in\mathcal{F}_1}[E_{50}(i)]$
C1	0.497	0.649
C2	0.202	0.222
C3	0.591	0.683

This motivates us to leverage the seeds used in the cold-start stage because $\mathcal{S}$ is usually less biased for its sufficient number of samples. Therefore, how to effectively leverage the seed set together with limited feedbacks becomes a key problem in the current stage. One challenge, which we can observe from Figure 2 and Table 1, is that the quality of $\mathcal{S}$ is quite different across different campaigns. Directly incorporating the seed-set with feedbacks only leads to reduce the average expandability of positive samples, which easily deteriorate the quality of the expansion.

To adaptively deal with varied qualities of the seed-sets across different campaigns, unlike the unsupervised Isolation Forest method used in the first stage, we devise a learning to weight method to reweight seeds so that we can detect these golden seeds within $\mathcal{S}$ and incorporate into $\mathcal{F}$ to reduce ρ_{-1}. Specifically, a weighting function $\mathcal{V}$ and a classifier f are learned in a meta-learning manner [7, 8], and we divide the feedbacks $\mathcal{F}$ into the training data $\mathcal{F}_{train}$ and the meta-data $\mathcal{F}_{meta}$ along the time direction (e.g., $|\mathcal{F}_{train}| : |\mathcal{F}_{meta}| = 23:1$) to capture the change of users' preference. Then, the parameters Θ of $\mathcal{V}(x_i;\Theta)$ and the parameter θ of $f(x_i;\theta)$ are jointly optimized to maximize the accuracy of prediction on $\mathcal{F}_{meta}$, which gives the following objective function:

$$\Theta^* = \arg\min_{\Theta} -\frac{1}{|\mathcal{F}_{meta}|}\sum_{(x_i,y_i)\in\mathcal{F}_{meta}}[\mathbb{1}(y_i=+1)\log f(x_i;\theta^*) + \mathbb{1}(y_i=-1)\log(1-f(x_i;\theta^*))], \quad (7)$$

where,

$$\theta^*(\Theta) = \arg\min_{\theta} -[\frac{1}{|\mathcal{S}|}\sum_{x_i \in \mathcal{S}} \mathcal{V}(x_i;\Theta)\log f(x_i;\theta)] - \frac{1}{|\mathcal{F}_{train}|}\sum_{(x_i,y_i)\in\mathcal{F}_{train}} [\mathbb{1}(y_i=+1)\log f(x_i;\theta) + \mathbb{1}(y_i=-1)\log(1-f(x_i;\theta))] \quad (8)$$

To avoid the nested loops of optimization, we use the online strategy [25] to optimize Θ and θ, which includes the following updates:

$$\hat{\theta}^{(t)}(\Theta) = \theta^{(t)} - l_r \frac{1}{b_t} \times \sum_{i=1}^{b_t} \mathcal{V}(x_i;\Theta)\nabla_\theta l_i^{train}(\theta)\Big|_{\theta^{(t)}} \quad (9)$$

$$\Theta^{(t+1)} = \Theta^{(t)} - l_r \frac{1}{b_m}\sum_{j=1}^{b_m} \nabla_\Theta l_j^{meta}\left(\hat{\theta}^{(t)}(\Theta)\right)\Big|_{\Theta^{(t)}} \quad (10)$$

$$\theta^{(t+1)} = \theta^{(t)} - l_r \frac{1}{b_t} \times \sum_{i=1}^{b_t} \mathcal{V}\left(x_i;\Theta^{(t+1)}\right)\nabla_\theta l_i^{train}(\theta)\Big|_{\theta^{(t)}} \quad (11)$$

where l_r is the learning rate, b_t and b_m are batch size, l_i^{train} and l_j^{meta} denote the loss calculated on the sample i in $\mathcal{F}_{train}$ and j in $\mathcal{F}_{meta}$, respectively.

Optimizing Eq. 7 would require to train over $\mathcal{S}$ and $\mathcal{F}$, which is computationally intensive and harmful to the convergence due to a number of noisy seeds. Inspired by the self-training technique [1], which only includes the most confident data samples into the new training set, we apply the adaptive reweighting method on the most confident seeds $\mathcal{S}_{top} \subset \mathcal{S}$, of which scores are given by the classifier f_d trained by optimizing Eq. 6. Here, we set the number of confident seeds equal to the number of positive feedbacks in $\mathcal{F}$, i.e., $|\mathcal{S}_{top}| = \sum_{(x_i,y_i)\in\mathcal{F}} \mathbb{1}(y_i=+1)$, and further define the weighting function $\mathcal{V}$ as:

$$\mathcal{V}(x_i;\Theta) = max(w_{top}, 0) \quad (12)$$

where w_{top} is a trainable parameter. Note that the weighing function put equal weight on each sample in $\mathcal{S}_{top}$ based on the assumption that users in $\mathcal{S}_{top}$ are with similar expandability, which greatly simplify the weighting function. Here, w_{top} controls the overall weight of selected seeds. Ideally, a good quality of seed-set would corresponds to a large value of w_{top}.

4 EXPERIMENTS

In this section, we conduct extensive offline and online experiments on the proposed system to demonstrate its effectiveness by answering the following three questions:

- Q1: Does the reweighting mechanism help the first cold-start stage of targeting?
- Q2: Does the combination of the seed-set and feedbacks boost the expansion in the second stage of targeting?
- Q3: Does this two-stage audience expansion system get improvement on the performance as a whole?

Before diving into the experimental results, we first introduce the experiment settings, including datasets, comparison methods and metrics for evaluation.

Datasets. In our experiments, three datasets of financial targeting are used. For the sake of replicability of our experiments, two small datasets (i.e., A and B in Table 2) are publicly available[5]. In addition to the two public datasets, one much larger industrial datasets are also used to demonstrate the effectiveness of the proposed method. All datasets include the seed-set $\mathcal{S}$, candidate-set C, user behaviors (here users with click behaviors are considered as the target audience) with timestamps and d-dimensional user embeddings $\mathbf{X}$. We summarize the statistics of these datasets in Table 2.

Table 2: Statistics of the experimental datasets.

Dataset	$\lvert\mathcal{S}\rvert$	$\lvert C\rvert$	d
A	287560	2846323	16
B	380895	2791242	16
C	~3 million	~0.3 billion	64

Comparison Methods. We compare our method with several strong baselines:

- One-Class SVM (**OCSVM**) [20]: a model-based method audience expansion method using one-class support vector machine that only takes the seed-set as the input.
- **Pinterest** [5]: an approach presented by Pinterest. In the offline stage, a global user embedding model is trained to generate user embedding. In the online stage, an embedding-based scoring model is employed to assign the score for each user w.r.t. a given campaign.
- Multilayer Perceptron (**MLP**): a model-based method audience expansion method using MLP as the classifier. In this paper, MLP is served as the base classifier of all our methods.
- Self-Ensemble Label Filtering (**SELF**) [21]: a learning method with noisy labels, which proposed to progressively filter out the wrong labels during training.
- **Self-Training** [1]: a semi-supervised learning method, which iteratively includes the most confident data samples into the new training set and re-train the model.

Model Setup. For OCSVM, we use the RBF kernel. For Pinterest, we split the whole user embedding space into 10 random hyperplanes and repeat the entire process of determining random projections for 10 times. For MLP, there are three hidden layers with ReLU as the activation function. The number of neurons is 512, 256 and 128, respectively. We train all neural network-based methods using Adam [14] optimizer with a learning rate of 3e-4, and set the batch size to 512. In Self-Training, we only iterate for once and the final classifier is trained on $\mathcal{S}_{top} \cup \mathcal{F}$.

Metrics. In this paper, we introduce two metrics, which are Area Under Curve (AUC) and "precision at $p\%$" (prec@$p\%$), to quantitatively measure the performance of different methods. Both AUC and prec@$p\%$ are calculated on user feedback data in snapshot 1 (stage 1) and snapshot 2 (stage 2), and prec@$p\%$ is defined as:

$$prec@p\% = \frac{\sum_{i\in\mathcal{P}} \mathbb{1}(y_i=+1)}{p\% * |C|} \quad (13)$$

[5] https://tianchi.aliyun.com/dataset/dataDetail?dataId=50893

where $\mathcal{P}$ denotes the top $p\%$ scored users in C. In this paper, we set $p = 5$. The higher the two metrics, the better the corresponding method is. Since our system is divided into two stages, we evaluate the proposed methods in each stage, and Figure 4 describes the inputs to each stage and how the evaluations are conducted.

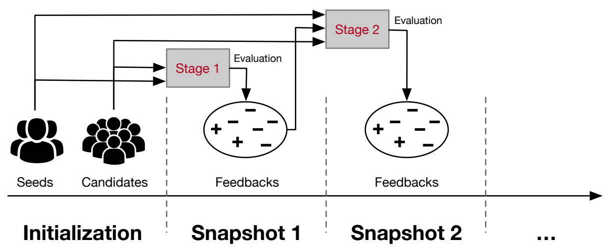


Figure 4: Evaluation procedure on the two stages.

4.1 Q1: Results on Cold-start with Noisy Seed-set

To demonstrate the effectiveness of the reweighting method, we compare it with the aforementioned methods on the three datasets. Since the reweighting would change the ratio of the weights of positive to negative samples, we first scale the weight of all positive samples to ensure the total weights of positive samples equal to negative samples. The experimental results are shown in Table 3. We can observe that our reweighting approach performs best in all datasets. As for *SELF*, we can observe the performance degradation, which indicates that the noisy label filtering technique failed while dealing with such a seed-set with extremely poor expandability.

Table 3: Offline evaluation results in the cold-start stage.

		Datasets		
	Methods	A	B	C
AUC	OCSVM	0.625	0.610	0.553
	MLP	0.681	0.646	0.607
	Pinterest	0.623	0.608	0.510
	SELF	0.681	0.630	0.581
	Reweighting	**0.692**	**0.657**	**0.627**
prec@5%	OCSVM	1.05e-2	7.27e-3	1.86e-4
	MLP	1.22e-2	1.29e-2	5.63e-4
	Pinterest	9.54e-3	9.24e-3	9.83e-5
	SELF	1.33e-2	1.53e-2	4.01e-4
	Reweighting	**1.36e-2**	**1.54e-2**	**6.09e-4**

4.2 Q2: Results on Expansion with Feedbacks

Similarly, we compare with the aforementioned baselines to demonstrate the effectiveness of our proposed method. To further compare the effectiveness between $\mathcal{S}$ and $\mathcal{F}$, we test these baselines with different combinations of inputs, i.e., $\mathcal{S}$, $\mathcal{F}$ and $\mathcal{S}+\mathcal{F}$. Here, $\mathcal{S}$ is served as positive samples, and $\mathcal{F}$ includes both positive and negative feedbacks. For those methods that only use positive samples (i.e., *Pinterest* and *OCSVM*), only $\mathcal{S}$ and positive feedbacks are used as the inputs. The experimental results are shown in Table 4, we can find that:

Table 4: Offline evaluation results in the second stage.

			Datasets		
	Methods	**Inputs**	A	B	C
AUC	Pinterest	$\mathcal{S}$	0.624	0.608	0.513
		$\mathcal{F}$	0.733	0.596	0.506
		$\mathcal{S}$+$\mathcal{F}$	0.698	0.596	0.512
	OCSVM	$\mathcal{S}$	0.451	0.572	0.573
		$\mathcal{F}$	0.605	0.560	0.585
		$\mathcal{S}$+$\mathcal{F}$	0.454	0.570	0.574
	MLP	$\mathcal{S}$	0.675	0.647	0.678
		$\mathcal{F}$	0.761	0.710	0.741
		$\mathcal{S}$+$\mathcal{F}$	0.655	0.667	0.710
	Reweighting	$\mathcal{S}$	0.669	0.656	0.689
		$\mathcal{F}$	0.746	0.659	0.727
		$\mathcal{S}$+$\mathcal{F}$	0.675	0.672	0.669
	Self-Training	$\mathcal{S}$+$\mathcal{F}$	0.759	0.710	0.750
	Our	$\mathcal{S}$+$\mathcal{F}$	**0.763**	**0.712**	**0.752**
prec@5%	Pinterest	$\mathcal{S}$	8.16e-3	1.94e-3	4.04e-5
		$\mathcal{F}$	1.72e-2	3.58e-3	8.61e-5
		$\mathcal{S}$+$\mathcal{F}$	1.22e-2	3.58e-3	4.29e-5
	OCSVM	$\mathcal{S}$	4.64e-3	2.53e-3	8.00e-5
		$\mathcal{F}$	9.00e-3	2.69e-3	9.33e-5
		$\mathcal{S}$+$\mathcal{F}$	4.69e-3	2.50e-3	1.06e-4
	MLP	$\mathcal{S}$	1.07e-2	2.75e-3	2.90e-4
		$\mathcal{F}$	1.85e-2	3.77e-3	3.69e-4
		$\mathcal{S}$+$\mathcal{F}$	6.16e-3	2.18e-3	1.99e-4
	Reweighting	$\mathcal{S}$	1.07e-2	2.89e-3	3.15e-4
		$\mathcal{F}$	1.90e-2	3.74e-3	3.76e-4
		$\mathcal{S}$+$\mathcal{F}$	1.13e-2	2.94e-3	2.95e-4
	Self-Training	$\mathcal{S}$+$\mathcal{F}$	1.82e-2	3.87e-3	3.76e-4
	Our	$\mathcal{S}$+$\mathcal{F}$	**1.91e-2**	**4.02e-3**	**3.92e-4**

- Comparing with only using $\mathcal{S}$ as the input, expansion based on $\mathcal{F}$ shows great improvements in most methods, which demonstrates the advantage of expansion with feedbacks.
- Naive combination of $\mathcal{S}$ and $\mathcal{F}$ leads to performance degradation, and *Self-Training* shows promising results on the combination of $\mathcal{S}$ and $\mathcal{F}$ except for dataset A.
- *Reweighting* over $\mathcal{S}$ can still achieve consistent improvements on the second stage, but *Reweighting* over $\mathcal{F}$ usually leads to degradation due to the biased distribution of $\mathcal{F}$.
- Our proposed method performs best among all the other baselines.

To further analyze the role of adaptive reweighting, we present the results when w_{top} is set to different fixed values (i.e., 0, 10, 30, 50, 70, 90) and the optimal weight w^*_{top} obtained from weighting function, which are shown in Table 5. We observe that the weighting function can adaptively put suitable weight on $\mathcal{S}_{top}$.

Table 5: Different weights on top-scored users.

		Datasets		
	w_{top}	A	B	C
AUC	0	0.761	0.710	0.741
	10	0.747	0.711	0.749
	30	0.745	0.712	0.752
	50	0.746	0.708	0.751
	70	0.744	0.709	0.752
	90	0.739	0.703	0.751
	w^*_{top}	0.15	10.5	12.3

4.3 Q3: Online A/B Test

To verify the effectiveness of the proposed two-stage methods, we further conducted a A/B test against a baseline using the vanilla MLP as the classifier. In the A/B test, we first randomly splited all candidates into two buckets. Then, using the same seed-set, the baseline and our approach select the same amount of top-scored users from two buckets, respectively. To quantitatively evaluate the expansion results, the following two metrics are compared:

- **Click UV**: a click indicates that a user clicks an item and it is only counted once for each user.
- **CTR**: the ratio of click UV and impressions.

In a real marketing campaign lasted for one week, we observe 9.01% increase in Click UV and 3.58% increase in CTR in total over the baseline, which validated the effectiveness of our system in financial targeting.

5 CONCLUSION

In this paper, we propose a two-stage audience expansion method for financial targeting. Due to the timeliness of financial targeting and high complexity of financial products and services, we first identify the existence of noises in the seed-set, which is ignored in conventional audience expansion methods. A quantitative metric, expandability, is proposed for delineating the ability of a user in the seed-set to expand to a new target user. With the help of this measure, an unsupervised reweighting method is introduced in the first stage to reduce the influence of noises, then given feedbacks, a supervised learning to reweight method is proposed in the second stage to combine the noisy seed-set with feedbacks. Finally, extensive experiments are conducted to demonstrate the effectiveness of the proposed methods. In addition, in response to this new problem, we release two real-world datasets for boosting the study of this new research direction. Future work will focus on using a lightweight model to extract effective information from the noisy seed-set.

REFERENCES

[1] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. 2009. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks* 20, 3 (2009), 542–542.

[2] Injazz J Chen and Karen Popovich. 2003. Understanding customer relationship management (CRM). *Business process management journal* (2003).

[3] Pengfei Chen, Ben Ben Liao, Guangyong Chen, and Shengyu Zhang. 2019. Understanding and Utilizing Deep Neural Networks Trained with Noisy Labels. In *International Conference on Machine Learning*. 1062–1070.

[4] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. ACM, 191–198.

[5] Stephanie deWet and Jiafan Ou. 2019. Finding Users Who Act Alike: Transfer Learning for Expanding Advertiser Audiences. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2251–2259.

[6] Khoa D Doan, Pranjul Yadav, and Chandan K Reddy. 2019. Adversarial Factorization Autoencoder for Look-alike Modeling. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. ACM, 2803–2812.

[7] Matthias Feurer, Jost Tobias Springenberg, and Frank Hutter. 2015. Initializing bayesian hyperparameter optimization via meta-learning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[8] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 1126–1135.

[9] Mihajlo Grbovic and Haibin Cheng. 2018. Real-time personalization using embeddings for search ranking at airbnb. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 311–320.

[10] Jingrui He. 2012. *Analysis of rare categories*. Springer Science & Business Media.

[11] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. ACM, 2333–2338.

[12] Jinling Jiang, Xiaoming Lin, Junjie Yao, and Hua Lu. 2019. Comprehensive Audience Expansion based on End-to-End Neural Prediction. In *2019 SIGIR Workshop on eCommerce, eCOM 2019*, Vol. 2410.

[13] Katie King. 2019. *Using Artificial Intelligence in Marketing: How to harness AI and maintain the competitive edge*. Kogan Page Publishers.

[14] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[15] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 413–422.

[16] Haishan Liu, David Pardoe, Kun Liu, Manoj Thakur, Frank Cao, and Chongzhe Li. 2016. Audience expansion for online social network advertising. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 165–174.

[17] Yudan Liu, Kaikai Ge, Xu Zhang, and Leyu Lin. 2019. Real-time Attention Based Look-alike Model for Recommender System. *arXiv preprint arXiv:1906.05022* (2019).

[18] Qiang Ma, Eeshan Wagh, Jiayi Wen, Zhen Xia, Robert Ormandi, and Datong Chen. 2016. Score Look-Alike Audiences. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*. IEEE, 647–654.

[19] Qiang Ma, Musen Wen, Zhen Xia, and Datong Chen. 2016. A Sub-linear, Massive-scale Look-alike Audience Extension System A Massive-scale Look-alike Audience Extension. In *Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications*. 51–67.

[20] Larry M Manevitz and Malik Yousef. 2001. One-class SVMs for document classification. *Journal of machine Learning research* 2, Dec (2001), 139–154.

[21] Duc Tam Nguyen, Chaithanya Kumar Mummadi, Thi Phuong Nhung Ngo, Thi Hoai Phuong Nguyen, Laura Beggel, and Thomas Brox. 2020. Self: Learning to filter noisy labels with self-ensembling. In *International Conference on Learning Representations*.

[22] Yan Qu, Jing Wang, Yang Sun, and Hans Marius Holtan. 2014. Systems and methods for generating expanded user segments. US Patent 8,655,695.

[23] Anand Rajaraman and Jeffrey David Ullman. 2011. *Mining of massive datasets*. Cambridge University Press.

[24] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International Conference on Data Mining*. IEEE, 995–1000.

[25] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. 2019. Meta-weight-net: Learning an explicit mapping for sample weighting. In *Advances in Neural Information Processing Systems*. 1917–1928.

[26] Dawei Zhou and Jingrui He. 2019. Gold panning from the mess: Rare category exploration, exposition, representation, and interpretation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 3213–3214.

[27] Chenyi Zhuang, Ziqi Liu, Zhiqiang Zhang, Yize Tan, Zhengwei Wu, Zhining Liu, Jianping Wei, Jinjie Gu, Guannan Zhang, Jun Zhou, and Yuan Qi. 2020. Hubble: an Industrial System for Audience Expansion in Mobile Marketing. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. In Press.