Practical Lessons from Predicting New User Demographics for Ad Targeting

Musen Wen MUSEN.WEN@GMAIL.COM
eBay Inc.
San Jose, California USA

Zhen Xia OVERSNOW@GMAIL.COM
eCreditPal
Mountain View, California, USA

Deepak Kumar Vasthimal DEEPUJAIN@GMAIL.COM
eBay Inc.
San Jose, California, USA

Editors: João Vinagre, Alípio Mário Jorge, Albert Bifet and Marie Al-Ghossein

Abstract

Programmatic ad buying, the use of technology to automate and optimize the ad buying process in real-time, has been emerging to be the major form of online advertising. For each online campaign, advertisers generally want to specify a certain group of audience that they want to target at. Among these, demographics (user age and gender) is the fundamental and most common targeting option. On the other side, due to the huge volume of bid-requests flowing into the exchange, majority of those users (i.e. cookies) are either completely new to the ad platform or has too little historical behavior information to determine their demographics. In this paper, we present and discuss the methods, system and practical lessons in tackling this problem at massive scale.

Keywords: behavior targeting, advertising, ad targeting, demographics, machine learning, computational advertising

1. Introduction

Online advertising is a multi-billion dollar industry and growing rapidly in past decade. Programmatic ad buying, advertisement platforms offers advertisers an automatic way to buy for video, display and native advertisements across all devices. The ads allocation is dynamic, real time, and generally tailored to online users demographics and interest. Big data and machine learning are the backbone of such programmatic ad buying business.

User targeting (i.e. showing the right ad to the right person), and the prediction of users conversion probability (if showing them the ads) are two important components of the automated ads buying system. Accurate targeting enables the advertiser to reach the right online users (e.g. right age group, gender group, or specific interest group) to deliver their message in order to either increase their brand awareness or directly sell their products. An efficient ad buying platform enables both advertisers and publishers to achieve a “win-win”, and hence improve the efficiency of the whole marketplace.

In this paper, we focus on developing machine learning system to predict users' demographics for ad targeting in real time and on massive scale (Vasthimal et al., 2017). In most platforms, online users are represented in terms of either their web browser's "cookies" or mobile device IDs (IDFA, IMEI, e.g.). We use the term "user" universally to refer to all these possible use cases and scenarios.

There are many existing literature on predicting online user's demographics (Hu et al., 2007) (Jansen and Solomon, 2010) (Kabbur et al., 2010) (Murray and Durrell, 1999). We categorize these as "offline" methods, where we need to "offline" collect user's past behavior on the browser and mobile device, etc. before we can make the prediction. These behavior could be a log of browsing websites, searching text on the search box, clicking behaviors on ads, or following links to visit advertiser's websites, etc. With various machine learning method and user's rich past behavior, we can then predict user's age and gender group based on these information. The user ID and offline prediction results are stored on grid and cached. When this user visits a publishers website, an ad request is sent promptly to the ad buying platform in real-time. The ad server system looks up the cached user profile and identifies this users demographics (which is pre-calculated and stored). The bidding processes is conducted for those ads that satisfy the targeted demographics condition.

However, in practice, for most ad buying platform, at most up to 30% users from ad requests are old (existing) users and their demographics are predicted and cached. A large volume of users passing along the ad requests to the ad buying platform are "new" either they are just seen for the first time, or even we seen this user in our system once or twice before, but there is extremely little past behavior we knew about this user (so that its demographics was not predicted before). Thus, exploration and building a comprehensive "online" demographics prediction system is beneficial and critical, given the massive volumes of new or first-seen users.

This paper focuses on practical lessons in predicting the new user demographics. The rest of the paper is organized as follows - in Section 2, we explore and experiment various benchmark models; in Section 3, we discuss different aspects of the practical implementations; in Section 4, we described the design of the production pipeline; in Section 5, we focus on analyzing the accuracy issues and constructing the confidence score for practical applications; at the end, in Section 6, we point out some possible directions for further improving the prediction system.

2. Model Exploration

2.1. Data and Features Coverage

Unlike offline demographics prediction (for old and existing users), each new user from bid requests carries *very limited* information. These information contains 1) userID (cookie, mobile device ID, e.g.) and 2) a bunch of attributes such as URL, operating system of the user used, browser type, timestamp, etc. Further parsing these information in real time, for new user demographics prediction, we will generally have around ten refined features. Those features entered into the final model include Top-Level Domain (TLD), Domain, browser type, Hour-of-Day (HOD), etc. Notice that for some features, such as domain and TLD, they are extremely sparse. Although the number of features for each bid request varies, 99% of new users should have at least a few of most common feature types including TLD.

Table 1: AUC Comparision for GBDT, LR and NB methods

Model	**AUC Lift (relative to GBDT)**
Gradient Boost Decision Tree	-
Sparse Logistic Regression	3.35%
Naive Bayes (NB)	3.01%

2.2. Experiment Setup: Labeling

The demographics prediction can be viewed as traditional supervised learning problem, where the labels we used as "ground truth" are obtained from hundreds of millions of web users. Web mail applications for example, requires filling demographics information to complete the registration for services. A set of randomly selected anonymous users, whose age and gender are voluntarily disclosed, are used as labels. This will then join with a large volumes of historical bid requests to form the online features for these user. A traditional 75%-25% train and test (Vasthimal et al., 2019)split are used for offline comparison for different models.

2.3. Benchmark Models

Based on above experimental data set, we conducted experiments on three benchmark machine learning models: Sparse Logistic Regression, Gradient Boost Decision Tree, and Naive Bayes.

1. Logistic Regressions (LR): Large scale grid based modeling packages, such as Vowpal Wabbit (Langford et al., 2007), are able to handle billions of training samples and millions of sparse features. Although our feature types is not large, but features themselves are extremely sparse, such as TLD or Domain. Logistic regression becomes a nature candidate.

2. Gradient Boosting Decision Tree (GBDT): Another widely used family of models in industry are tree models, such as gradient boost decision tree (GDBT), which has been reported widely that it performs better for many industrial applications.

3. Naive Bayes (NB): Last, we also try with Naive Bayes, largely due to its simplicity especially on the system side implementation, as well as its prediction stability. We consider the Naive Bayes method as another nature candidate.

Table 1 gives a comparison the AUC lift for gender prediction with respect to the GBDT baseline in test data.

In our experiments, well customized logistic regression performs the best, with AUC slightly higher than Naive Bayes. In practice we consider both methods are on par - due to the tiny difference in AUC. On the other hand, since Naive Bayes has no optimization procedure involved, its prediction performance is generally more stable and controllable in production as time involves, where underlying feature distributions may shift. Last, since NB does not require additional optimization, we are able to avoid the overhead of dynamically exploring best parameters (L1, learning rate, number of passes, etc.) for optimization for logistic regression in production pipeline. In view of these, we implement the Naive Bayes as the production model for prediction.

3. Prediction model components

The fundamental of Naive Bayes (NB) is straight forward. For example, the posterior prediction for a new user being Male, given all attributes a_1, a_2,...,a_I can be expressed as,

$$\hat{P}\left(Male|a_1, a_2, \cdots, a_I\right) \propto \hat{P}\left(Male\right) \times \prod_i \hat{P}\left(a_i = k|Male\right)$$

Thus, for any new user, the final prediction of its gender is given by,

$$\underset{Male,Female}{\arg\max} \left\{\hat{P}\left(Male|a_1, a_2, \cdots, a_I\right), \hat{P}\left(Female|a_1, a_2, \cdots, a_I\right)\right\}$$

In what follows, we will discuss some practical issues we learned that will affect the performance of the model.

3.1. Feature Selection

In practice, one key practical component we considered to establish a reliable NB based prediction system is on feature selection. Even in our training data, we have hundred of thousands of different sub-domains. A large amount of these sub-domains have basically no discriminant power for user gender or age. Keeping all these sub-domain attributes will not help on anything but adding large amount of noise to the prediction system, and decrease the prediction performance significantly.

A pre-filtering of the features is conducted via χ^2 feature selection, a technique widely used in natural language processing (NLP) (Manning et al., 2008).

$$\chi^2 = \sum_i \sum_j \left(\frac{o_{ij} - e_{ij}}{e_{ij}}\right)$$

Suppose we have label $Y \in \{Male, Female\}$, and for each sample, we have a K-dimensional feature vector $\mathbf{X} = (X_1, \cdots, X_K)$, where X_i representing how many times sub-domain i is visited. We ran through a χ^2 feature selection for all these K sub-domains. Our practical rule is to select those sub-domain with χ^2 value greater than 3.84. After this process, we end up with retaining around 1/3 of the original sub-domains in our training data set.

Table 2: Binary Feature Counts

	visit sub-domain	not visit sub-domain
male	N_{00}	N_{01}
female	N_{10}	N_{11}

For any binary features, such as Table 2, we implemented a fast feature selection process via the calculation of $\chi^2 = \frac{(N_{11}+N_{10}+N_{01}+N_{00})(N_{11}N_{00}-N_{10}N_{01})}{(N_{11}+N_{01})(N_{11}+N_{10})(N_{10}+N_{00})(N_{01}+N_{00})}$, where all these calculations are conducted offline, and stored for later online prediction.

3.1.1. Add-Delta Smoothing

Further, in our design and implementation, our practice shows that the add-δ smoothing technique in calculating the conditional probabilities (Chen and Goodman, 1996) for the cells that has no observations in order to avoid an estimation of "zero" probability work the best for demographics prediction with all these spare features. In our learning, instead of the general add-one smoothing method, an extremely small δ such as 10^{-8} we find out that it will generate a much stable result from practice.

3.2. Prior Effect and Geo-partition

The Prior is critical for NB method, especially when we do not have too much features for each user. The prior can be estimated on impression (ad request) level or ID level. Each method has its own advantage and disadvantage. In general, the impression estimate has large variance - think of one user (suspicious bots) that has thousands of impressions per day; the ID level estimation will mainly be suffered to the so called "cookie churn" problems. In our practice, we pay close attention to the healthy and normal traffics and use impression level data to estimate the priors. In our practice, we also found that the prior estimates for new user should varies from group to group. In the system, we can introduce the concept of the geo-partition of users. The motivation of these is to generate more homogeneous groups of users to yield a good "local" estimation of the priors. We found that by adding such dimension, the accuracy is improved and the estimation results are more stable.

3.3. Model Re-calibration

In predicting the demographics, together with classifying any incoming cookie as Male or Female, the probability of being Male or Female for this cookie is of equally importance. For any user associated with the attributesfeatures, we can classify whether its Male or Female via the posterior probability. In Nature Language Processing and document classification, the usage of Naive Bayes method will generally result in distorted probabilities. In developing our prediction system, special attention are paid to ensure that it will not suffer too much from this issue. Due to the lack of correlation among features (TLD, etc.) and relatively small number of types of features, in practice we found that we are not suffering from such scoring bias. To illustrate this point, we conduct the offline experimentation to construct the reliability diagram. We apply Platts calibration (Casella and Berger, 2002) to the original raw probability scores. We plot the reliability diagram for both before and after calibration (the number of buckets we choose here is 15) We find that the probability is adjusted slightly after calibration, and the reliability diagram shows small difference from the raw normalized probability. It lines along the 45-degree line to the empirical probability. In our system, we calculate the calibrated scores to provide probability measurement for the prediction on individual sample level.

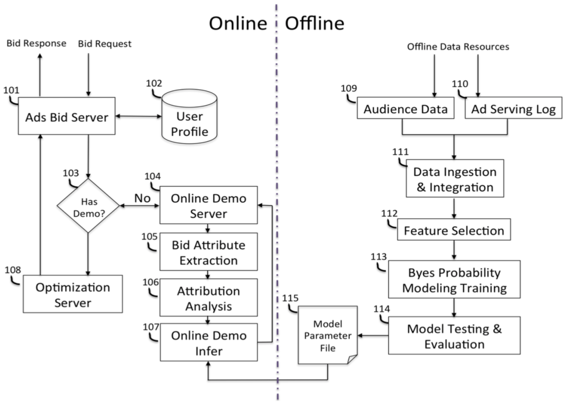

Figure 1: System Architecture and Pipeline.

4. Building the Real-Time Prediction Pipeline

The system is designed for demographic inferences including user age and gender. Broadly speaking, it contains an offline pipeline running periodically to train the model (as describe above) and an online pipeline developed for real time prediction for any bid requests. The overall design of both system and pipelines is illustrated in Figure 1. In following subsections, we describe each components in details.

4.1. Online Pipeline

As shown in Figure 1, when a bid request comes in, it is handled by the processor 101, the ads bid server. The processor 101 first looks up the user demographic information from the processor 102, online user profile server. If the user demographic does not exist in user profile server, the ads bid server then calls the online demographic prediction module by the processor 103. The online demographic prediction module starts with the processor 104, the online demo server, and then the online demo server calls processor 105 to extract available attributes during the bid-request. After attributes extraction, the online model performs analyses on the available attributes and chooses corresponding models to predict by processor 106. The online model infers user's age and gender information based on pre-cached models as described in previous sessions. This is done from processor 107. The predict user demographic information passes to processor 108, the optimization server. After this step, the selected ad returns to processor 101, the ads bid server, to make a real time bidding.

4.2. Offline Pipeline

The offline process and pipeline are in the right part of Figure 1, which is a main engine for model training. Training data resource are from processor 109 and processor 110 including 3rd party audience data, serving logs (Vasthimal et al., 2011), etc.

Massive data ingesting and integrating are performed with processor 111, which transfers different data resources to a unified data source platform. Feature availability is very limited for online predictions in our advertising system. In processor 112, and handful of features are carefully selected and tested leveraging on the model coverage and accuracy. The processor 113, the Bayesian modeling framework is carefully tuned after balancing the accuracy and efficacy. The model is trained offline by batch processing and is updated on a daily basis in the online system. Model testing and prediction performance evaluation are carried out by processor 114, where the model is tested and evaluated by large set of testing data and calibrated with 3rd party data. After thorough testing and evaluation, the model file is then pushed to the online module with processor 115 for real time forecasting.

5. Addressing Accuracy in Practice

5.1. Model Performance: Practical Accuracy

In these sections, we analyze the performance of the model from the practical aspects and discuss some practical concerns. For gender prediction, we observed a much higher accuracy than a random guess (i.e. use prior only as an uniform forecast).

For age prediction, it is not quite practical and applicable to predict the exact age (in magnitude of how many years old) of users. Instead, in accordance with most ads buying platform, we consider 7 buckets for ages: 17 & below, 18-24, 25-34, 35-44, 45-54, 55-64, 65+. The system built would then aim at predicting the users age buckets in real time. By incorporating the attribute features into the prediction system with NB method, we obtains relatively average 10 + percent accuracy lift. A very important point here for the practice and business is that - one age bucket error margin is quite normal and acceptable in real situations. This is largely due to the fact that when advertisers specify a targeted user age range. And this range normally spans across at least two adjacent age buckets (that we used in our system). For example, a targeted population could be male, and age from 25 to 45. Thus a predicted ages that fall into near by age bucket most probably still being considered as a "correct" result in practice. Subsequent analysis we conducted shows that the age forecasting system are well adequate according to most of our advertisers' targeting criteria for age.

5.2. Confidence Score: Fisher's Information Mapping

Generally a "confidence score" must be provided at the same time with the prediction. On business side, this score will be used to determine whether the prediction should be used for campaign targeting according to various advertisers' hard targeting criteria and targeting error tolerance. Motivated by the fact that a larger difference between the probability estimated for male and female from a single ads request, the higher evidence that we could place on the gender prediction. We develop a simple way to map the Fishers information to the empirical accuracy of the prediction. To formulate, notice that our prediction is for Bernoulli outcome X, where value "1" represents male, and "0" represents female. Then $X \sim Bernoulli\,(1, p)$. The variance for each estimation of p can be estimated by $var\,(p) = \hat{p}\,(1 - \hat{p})$. We map the Fisher's information $I = \frac{1}{p(1-p)}$ to the empirical accuracy of the prediction. The minimal value of Fisher's Information I is 4. Empirically we estimate the distribution of the Fisher's information for a sample illustrated data set, and divide I into evenly spaced bucket and empirically estimate the mapping of this bucket to the prediction accuracy in the testing data set.

$$I \rightarrow \widehat{PredAccur}$$

where $\widehat{PredAccur}$ is the prediction accuracy in the testing data set. This scheme provides us an empirical estimation of the accuracy for any prediction in gender and age.

5.3. Accuracy and Coverage Trade-off

In real time, when an ad-request is received, our system provides a demographic prediction and a confidence score to quantify the confidence level of this prediction. One important decision practice in ad buying system is that we need to leverage the confidence score for each prediction. For example, if we have a very high confidence that the prediction of the demographics falls in a certain age/gender bucket, then we could make a decision to consider biding on this impression, etc. for some campaigns. In general, setting up a higher confidence score threshold would disqualify more new users for being considered for bidding.

In practice, we plot a mapping of the confidence score to the coverage for any confidence level we pick (x-axis), it provides an empirical estimate of the coverage (y-axis) of total users that could be considered bidding on at that specific confidence level. This empowers vast flexibility for targeting.

6. Discussions

So far, we have discussed an end-to-end system to predict new user's demographics in real-time base on Naive Bayes method. There are a few possible extensions we may consider in the future in order to further improve the prediction performance.

The major issues for predicting new user demographics is the lack of features we do not have anything except for what the ad request passes along to the system. One way to overcome this is to consider adding the cross-features (conjunctions) under the NB framework. Either subsection selections from an enumeration of all possible cross, or use GBDT to generate a new set of binary features (He et al., 2014) seems a reasonable way. Also, deep learning for feature engineering has emerged recently, and should be well considered as a further exploration.

We already see that the geo-partition has yield a much stable result. It is undoubtedly that adding new dimension to partition the data and get an appropriate prior probability of demographics estimation is beneficial. The exploration of extra dimensions and how to optimally partition the user space should be a continuous involving process and remains an open problem.

From the experiments, we see that logistic regression (or even GBDT) has similar prediction performance to the Naive Bayes model. Will a combination of all these models (which has its own advantage and disadvantage in terms of modeling the underlying data) and the use of stacking technique help boosting the prediction performance? Apart form the resources, latency and infrastructure re-design concern, in terms of pursing a better accuracy, this will be a good exploration direction. It will be also very interesting to see if an utilizing of representation learning will help on similar problems in the future.

References

G. Casella and R. L. Berger. Statistical inference. Thomson Learning, 2002.

S. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. Proceedings of the 34th annual meeting on Association for Computational Linguistics, 1996.

X. He, J. Pan, O. Jin, T. Xu, and B. Liu. Practical lessons from predicting clicks on ads at facebook. Proceedings of the Eighth International Workshop on Data Mining for Online Advertising (ADKDD), 2014.

J. Hu, H.-J. Zeng, H. Li, C. Niu, and Z. Chen. Demographic prediction based on users browsing behavior. Proceedings of the 16th International Conference on World Wide Web, 2007.

B. J. Jansen and L. Solomon. Gender demographic targeting in sponsored search. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2010.

S. Kabbur, E. Han, and G. Karypis. Content-based methods for predicting web-site demographic attributes. 2010 IEEE 10th International Conference on Data Mining (ICDM), 2010.

J. Langford, L. Li, and A. Strehl. Vowpal wabbit. https://github.com/JohnLangford/vowpal wabbit/wiki, 2007.

C. Manning, P. Raghavan, and H. Schutze. Introduction to information retrieval. Cambridge University Press, 2008.

D. Murray and K. Durrell. Inferring demographic attributes of anonymous internet users. International WEBKDD 99 Workshop San Diego, CA, USA, August 15, 1999, 1999.

N. A. Zaidi, J. Cerquides, M. J. Carman, and G. I. Webb. Alleviating naive bayes attribute independence assumption by attribute weighting. Journal of Machine Learning Research, 2013.

D. K. Vasthimal, R. R. Shah, and A. Philip. Centralized log management for pepper. 2011 IEEE Third International Conference on Cloud Computing Technology and Science, Athens, 2011, pp. 1-3.

D. K. Vasthimal, S. Kumar, and M. Somani. Near Real-Time Tracking at Scale. 2017 IEEE 7th International Symposium on Cloud and Service Computing (SC2), Kanazawa, 2017, pp. 241-244.

D. K. Vasthimal, P. K. Srirama, and A. K. Akkinapalli. Scalable Data Reporting Platform for A/B Tests. 2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS), Washington, DC, USA, 2019, pp. 230-238.