Forecasting Granular Audience Size for Online Advertising

Ritwik Sinha
Adobe Research

Dhruv Singal
Adobe Research

Pranav Maneriker
Adobe Research

Kushal Chawla
Adobe Research

Yash Shrivastava
IIT Kharagpur

Deepak Pai
Adobe Systems

Atanu R Sinha
Adobe Research

ABSTRACT

Orchestration of campaigns for online display advertising requires marketers to forecast audience size at the granularity of specific attributes of web traffic, characterized by the categorical nature of all attributes (e.g. {US, Chrome, Mobile}). With each attribute taking many values, the very large attribute combination set makes estimating audience size for any specific attribute combination challenging. We modify Eclat, a frequent itemset mining (FIM) algorithm, to accommodate categorical variables. For consequent frequent and infrequent itemsets, we then provide forecasts using time series analysis with conditional probabilities to aid approximation. An extensive simulation, based on typical characteristics of audience data, is built to stress test our modified-FIM approach. In two real datasets, comparison with baselines including neural network models, shows that our method lowers computation time of FIM for categorical data. On hold out samples we show that the proposed forecasting method outperforms these baselines.

CCS CONCEPTS

• **Applied computing** → **Forecasting**; • **Mathematics of computing** → **Time series analysis**; • **Information systems** → **Data mining**; *Online advertising*;

KEYWORDS

Display advertising, forecasting, frequent itemset mining, digital marketing, time series

ACM Reference format:
Ritwik Sinha, Dhruv Singal, Pranav Maneriker, Kushal Chawla, Yash Shrivastava, Deepak Pai, and Atanu R Sinha. 2018. Forecasting Granular Audience Size for Online Advertising. In *Proceedings of ADKDD'18, London, United Kingdom, August 20, 2018,* 6 pages.
https://doi.org/10.475/123_4

1 INTRODUCTION

The online display advertising (hereafter, display ad) ecosystem has many players that intermediate between publishers and marketers [13]. For targeting ad campaigns to consumers it is imperative for a marketer to estimate the number of visitors satisfying a set of targeted attribute values in a future time period. Consider one such target-{Country:US, Browser:Chrome, Device:Mobile}. The marketer may be interested in predicting the number of advertising bid requests for this target flowing into the Demand Side Platform(DSP) in the following week. This helps optimize spend allocation among various targets in a campaign, as well as helps manage the marketing budget across campaigns.

Forecasting granular-level audience size poses a considerable data mining challenge because of the explosion in the number of possible categorical attribute value combinations. One of our two real world datasets contains 10 attributes, each taking many values (even 100 or more), resulting in $\sim 10^{20}$ unique targets. While not all combinations are observed in the data, it is still infeasible to store data for all observed combinations and apply time series estimation methods. Notably, forecasting audience size for web traffic is an under-researched area, although programmatic advertising is the subject of growing research, with inroads in diverse topics like bid optimization [23], targeting [5] as well as estimating conversion rate [11] and click-through rate [24].

In proposing a practicable solution, we develop a three stage approach: first, bringing the problem to a tractable scale using frequent itemset mining (FIM); second, using conditional probability to extend to unobserved targets and third, leveraging time series analysis methods to forecast. Our approach is evaluated on two datasets: first, bid requests data received by a DSP and second, web analytics data of a US publisher. The DSP receives bid requests from multiple Ad Exchanges and serves multiple advertisers. The web analytics data, although from a single publisher, is more feature-rich than the bid requests data. While the two settings are different, the forecasting problem has important commonalities: both datasets comprise historical *time stamped events of consumers* (representing bid requests and page views in respective settings), where each event is defined using values for a set of categorical *attributes*. For each dataset, we forecast the number of events occurring in a given time period for a specific target set defined using values of categorical attributes. Our solution for the first dataset computes and stores the support for 5×10^5 frequent itemsets out of a possible 3.84×10^{18} and only about 100 time series models, and yet, is more accurate than baselines.

Online audience estimation requires forecasts (1) be available for any arbitrary attribute value combination, (2) be frequently updated, and (3) account for temporal variations. Historical time stamped events are used to estimate number of events with specified attribute values in a future time period. Notably, all attributes of

ADKDD'18, August 20, 2018, London, United Kingdom

ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.475/123_4

web traffic data are categorical and most attributes show long-tailed univariate distributions (Figure 1). Under this premise, our contributions are: One, we leverage the categorical nature of attributes to efficiently mine frequent attribute combinations from the event database (Section 3.1) by modifying a leading FIM algorithm to include categorical constraints. This improves performance time and helps meet (2). Two, the mined frequent item (attribute) sets (FIS) are only a small portion of attribute combinations used by firms for targeting. For the non-FIS, which is a very large set, we offer a scalable method for forecasting since the cost of storing all data is prohibitive. Our solution uses an approximation based on conditional probability, storing only on relatively few attribute sets. Three, given a target set definition and a time period in the testing phase, we select an appropriate time series model for predictions and then use information obtained from FIM to obtain estimates of audience size, which meet (3). The approach also estimates for non-FIS, thereby providing estimates for any arbitrary itemset (Section 3.2) and satisfies (1). Four, contributing to the FIM literature concerned with categorical variables, we introduce a simulation framework to stress test FIM algorithms.

2 RELATED WORK

The curse of cardinality in web traffic attribute combinations manifests in adverse query time and massive cost of storing temporal data. Websites need forecasts at granular level of attribute combinations and updated often. While existing FIM algorithms may handle the curse by extracting FIS, that fails to meet the website's needs for forecasts for most other non-frequent itemsets. We bring tools from probability and time series to address these issues. The forecasting problem considered in this work has been explored earlier by Agarwal et al [1]. However, they use domain knowledge in display advertising to build time series models for a subset of attribute combinations; we use FIM to build a generalizable approach.

The Apriori algorithm [2] has been extended to Eclat [7], FP-Growth [22], and LCM [21] algorithms. The latter three are considered better off-the-shelf algorithms for association rule mining problems [3]. In further development, [17, 19] adds category-based constraints to Apriori [4]. Advancing the work, we add categorical constraints to Eclat and show better performance against other state-of-the-art algorithms.

Time series forecasting is not new [6]. Recent attention to search through a class of models to provide forecasts based on best performing models includes Exponential Smoothing [9], Automatic ARIMA models [8] and Prophet [20]. We explore these three and a Neural Net based approach in our experiments.

Our introduction of a new framework for stress testing FIM algorithms draws upon statistical copula [14], to capture statistical dependencies among categorical variables. Existing data sets for testing FIM algorithms are not built for categorical variables. This approach to the FIM literature is expected to help in testing and comparing suitability of algorithms for data with categorical variables, which are common in web traffic.

3 APPROACH

Let us define a set of attributes $A = \{A_1, A_2, \ldots, A_k\}$, where each A_l takes one of a possible set of values, V_l. Let the set of events be $\mathcal{D} = \{d_1, d_2, \ldots, d_n\}$, where each event or *transaction* is defined by value assignments for each attribute, $d_i = \{d_{i1}, d_{i2}, \ldots, d_{ik}\}$ where $d_{il} \in V_l$. Additionally, each transaction has a timestamp associated with it. We define a *target definition* as $T = \{t_1, t_2, \ldots, t_k\}$, where $t_l \in V_l \cup \{u_l\}$, where u_l is a special marker indicating that t_l can take any value in V_l. This marker defines targets where some attributes are left unspecified. A transaction d_i *satisfies* the target definition T if for all $l, d_{il} = t_l$ where $t_l \neq u_l$. The audience estimation problem is formally stated as: given a historic dataset $\mathcal{D}$, estimate the number of events d_i satisfying T, in a future time range.

3.1 Frequent Itemset Mining

In frequent itemset mining(FIM), the events could be transactions, as in the case of purchase, or occurrences of audience member on a publisher site, as in our case. The problem is formally stated as follows [3]. For the set of *transactions* $\mathcal{D}$, such that each transaction is a set of *items*, denote the set of all possible items as $\mathcal{I} = \{\iota_1, \iota_2, \ldots, \iota_m\}$. Hence, each transaction is an *itemset*, $d_p \subseteq \mathcal{I}$. The cover $K(I) \subseteq \mathcal{D}$ of itemset $I \subseteq \mathcal{I}$ is the set of all transactions $\{d_p\} \in \mathcal{D}$ such that $I \subseteq d_p$. The *support* $s(I)$ is the size of $K(I)$, $s(I) = |K(I)|$. The problem is to find all itemsets $\{I_1, I_2, \ldots, I_m\}$ in $\mathcal{D}$ with support more than a threshold κ. Additional constraints allow more efficient enumeration of frequent itemsets [19]. A constraint is a mapping from the power set of items to a boolean value, $C : 2^{\mathcal{I}} \rightarrow \{True, False\}$. FIM algorithms exploit properties of the support constraint $(s(I) > \kappa)$.

A characteristic of online traffic is that the A_l^{th} attribute of a transaction d_p takes only one of the values in V_l. This implies that any itemset which has two or more values for the same attribute must have a zero count, which we encode as the categorical constraint (CC). We modify Eclat by checking for CC during the candidate set generation stage. Note that LCM and FP-Growth have both the horizontal and vertical representations of the transactions (explicitly in case of LCM and as the FP-Tree in FP-Growth) [3], thus cannot benefit from the inclusion of CC. Formally, $CC(I) = True$ iff $i_l \in V_l \; \forall l$, where I is the transaction $(i_1, i_2, \ldots, i_k)$, and V_l is as defined in Section 3. Constraints can be characterized by some properties such as anti-monotone, succinct, and convertible [15]. We state the definitions of two such properties here.

Definition 3.1. **Anti Monotone:** A constraint $C(\cdot)$ defined on sets is anti-monotone iff for all itemsets $S \subseteq S'$, $C(S) = False \implies C(S') = False$.

Definition 3.2. **Succinct:** A constraint $C(\cdot)$ defined on sets is succinct iff for all itemsets I: $C(I)$ can be expressed as $\forall e \in I : r(e) = True$ for a predicate r.

CC is anti-monotone and succinct [4]. Anti-monotone constraints can be applied to a level-wise algorithm, at each level successively [4]. Moreover, if a constraint is succinct, it is also *pre-counting pushable*. While [4] applied CC to Apriori, we extend CC to Eclat. This is done by *pre-counting pruning*, that is, CC can be pushed to the stage post the candidate generation phase and prior to support related checks, discarding ineligible candidates. For Eclat, the check is pushed to the stage prior to applying intersections of transaction lists of generated candidates (see Algorithm 1).

Algorithm 1: Eclat-CC

```
// Define t(I): Transaction ID list for itemset I
// Initial call:
    F ← ∅,  P ← {({i}, t({i})) : i ∈ I,  |t({i})| ≥ κ}
Function EclatCC(P, κ, F)
    Result: F, the set of frequent itemsets
    forall (X_a, t(X_a)) ∈ P do
        // X_a is a frequent itemset
        F ← F ∪ {(X_a, s(X_a))}, P_a ← ∅
        forall (X_b, t(X_b)) ∈ P, with X_b > X_a do
            X_ab = X_a ∪ X_b
            // Pre-counting pruning
            if CC(X_ab) then
                t(X_ab) = t(X_a) ∩ t(X_b)
                if s(X_ab) ≥ κ then
                    P_a ← P_a ∪ {(X_ab, t(X_ab))}
            end
        end
        // Recursive call
        if P_a ≠ ∅ then EclatCC(P_a, κ, F)
    end
```

3.2 Audience Estimation

The previous section described generation of FIS from $\mathcal{D}_{\tau-l,\tau}$ containing historical transactions in time $(\tau - l, \tau]$. The mined FIS provide $s_{\tau-l,\tau}(T)$, T being a target set satisfying the threshold κ. The interest lies in the support of T in a future time period $(\tau, \tau + m]$, that is, $s_{\tau,\tau+m}(T)$. While FIM obtains $s_{\tau-l,\tau}(T)$ for many target sets, forecasting for each requires maintaining highly granular time series data for each, making this infeasible for arbitrary targets, including for non-FIS targets. Our approach requires maintaining a granular time series only for a small number of univariate (single item) targets, and for these targets performing time series forecast that captures seasonal and trend patterns.

Denote the univariate time series targets as $\mathcal{U}$. Given $\mathcal{D}_{\tau-l,\tau}$, T, and a future time period $(\tau, \tau+m]$, we estimate the expected number of events in $(\tau, \tau + m]$. The FIS from $\mathcal{D}_{\tau-l,\tau}$ are stored along with their support. We compute the *best* univariate time series U (see below) to generate predictions for the target T, subject to $T \subseteq U$ and $U \in \mathcal{U}$. The predictions for $s_{\tau,\tau+m}(T)$ are as follows:

$$s_{\tau,\tau+m}(T) = P_{\tau,\tau+m}(T \mid U) \times s_{\tau,\tau+m}(U) \approx P_{\tau-l,\tau}(T \mid U) \times s_{\tau,\tau+m}(U), \tag{1}$$

where we use the empirical estimate for $P_{\tau-l,\tau}(T \mid U)$, given by

$$\hat{P}_{\tau-l,\tau}(T \mid U) = \hat{P}_{\tau-l,\tau}(T \cap U)/\hat{P}_{\tau-l,\tau}(U) = s_{\tau-l,\tau}(T)/s_{\tau-l,\tau}(U), \tag{2}$$

since $T \subseteq U$. In equation (1) we make the assumption that $P_{\tau,\tau+m}(T \mid U) \approx P_{\tau-l,\tau}(T \mid U)$, that is, the conditional probability of T given U remains (almost) constant from the training to the forecasting period. We tested this assumption on the FIS empirically from the two real datasets we work with, and get Pearson correlation > 0.99 between these two quantities for both.

We approximate $P_{\tau,\tau+m}(T \mid U)$ as $\prod_{i=1}^{k} P_{\tau,\tau+m}((u_1, \ldots, t_i, \ldots, u_k) \mid U)$ when T is not frequent, where u_j denotes that the j^{th} attribute takes any value in its support. In other words, we assume conditional independence among the attributes and compute the joint probability as the product of marginal probabilities.

When $(u_1, \ldots, t_i, \ldots, u_k)$ is frequent, we can use the formulation described in equation (2). In the other case, we use a threshold probability estimate $\kappa/s_{\tau-l,\tau}(U)$, where κ is the support threshold used for FIM. This is an upper bound on the empirical estimate for this itemset (using equation (2)).

To estimate the second term in equation (1), we explore multiple classes of time series models to generate the forecast $\hat{s}_{\tau,\tau+m}(U)$ along with standard deviation for all elements in $\mathcal{U}$ (details in section 4.2). The granularity of forecasts depends on the granularity of the input data. We generate hourly forecasts.

Now, from the set of candidate univariate time series for each target T, that is, those which satisfy: (1) $T \subseteq U$, (2) $s_{\tau-l,\tau}(U) \geq \kappa$, we choose the time series with the least error in prediction. From this limited set of univariate time series we still generate good predictions, as shown in our experiments. We preselect the univariates at the time of computing the frequent itemsets and choose univariates which satisfy (1) and (2). We choose from possible candidate time series, at prediction time, by minimizing the standard error of the estimate $\sigma\left(\hat{s}_{\tau,\tau+m}(T)\right) = \sigma\left(\hat{P}_{\tau,\tau+m}(T \mid U) \times \hat{s}_{\mathcal{D}_{\tau,\tau+m}}(U)\right)$.

4 EXPERIMENTS

Statistical Copula for a Simulation Framework: The current FIM literature offers synthetic datasets [2] which do not meet the need of emulating categorical nature of web traffic. Our framework fills this gap by creating synthetic data with two important properties: first, the marginal distributions follow structure typically seen in audience data, such as many attributes depicting a long tailed distribution(Figure 1); second, the strong dependence structure common in web traffic be maintained. For example, a type of browser is more likely to be used on a certain operating system. We achieve this by introducing statistical copula [14] into the FIM literature. A copula is a function that joins the multivariate distribution function to their one-dimensional marginals. This approach allows arbitrary marginal distributions while controlling the level of dependence between attributes.

We construct a Gaussian copula from a multivariate normal distribution over $\mathbb{R}^k$, by first specifying a correlation matrix $\mathbf{R}$. We simulate the random vector $\boldsymbol{x} = (x_1, \cdots, x_k)'$ with the multivariate Gaussian cumulative distribution function (CDF) $\Phi_R(\cdot)$ (with correlation matrix $\mathbf{R}$). Then, the vector $(\Phi(x_1), \cdots, \Phi(x_k))'$ (where $\Phi(\cdot)$ is the univariate normal CDF) has marginal distributions which are uniform in $[0, 1]$ and a Gaussian copula which captures the dependence. Finally, to achieve the target distributions $F_i(\cdot)$, we perform the transformation $\boldsymbol{y} = (F_1^{-1}(\Phi(x_1)), \cdots, F_k^{-1}(\Phi(x_k)))'$, where $F_i^{-1}(\cdot)$ is the inverse CDF corresponding to $F_i(\cdot)$. The resulting vector $\boldsymbol{y}$ has the desired marginals with a given dependence structure.

We are still left with deciding two quantities, $\mathbf{R}$ and $F_i(\cdot)$. Experiments with long tailed distributions show a good way to select $F_i(\cdot)$ - base it on the observed multinomial distribution of attribute values. We base the marginals on typically observed distributions in real data (Figure 1). To choose $\mathbf{R}$, we make use of the structure

Table 1: Distinct values for attributes in the real datasets

BRD		PVD	
Attribute	Unique Values	Attribute	Unique Values
ad_exchange	8	browser	876
browser	100	color_depth	8
country	233	country	233
device_family	23141	domain	61684
device	3	language	153
os	55	os	257
region	2598	ref_type	7
slot_size	693	region	1043
slot_visibility	3	resolution	448
		visit_number	12884

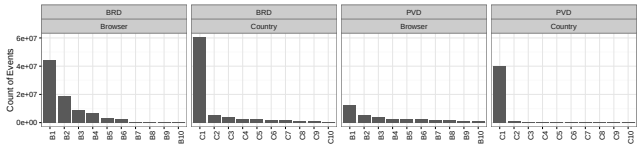

Figure 1: Attribute value frequencies in BRD and PVD

of the observed data. We ensure that the association matrices for the real and simulated data follow a similar pattern.

With the goal of testing the robustness of FIM approaches and for comparison among them, we vary the following parameters in the synthetic data. First, the number of attributes (k) is 8, 16 or 32. Next, the association is either as observed in real data or the off-diagonal elements are half of their values (these are referred to as 'high' or 'low' correlation). Next, we modify the multinomial marginal distributions to either be long-tailed ('steep') or uniform ('flat').

Bid Request Dataset (BRD): This dataset arises in the ecosystem where the publisher seeks competitive bids using Ad Exchanges and a Real-Time Bidding (RTB) platform. The publisher delivers the consumer's information, comprising attributes, for real time bidding by marketers seeking consumers matching those attributes. The training data comprise logs from 26 March to 31 March 2017, and the testing data comprise logs for 1 April, 2017. Around 97 million bid request events are present, large enough for valid experiments. We have 86 million and 11 million bid requests in the training and the testing periods respectively. Each event has 9 attributes (Table 1), a time stamp, and most attributes have a substantial number of distinct values. The number of possible attribute combinations is 3.84×10^{18}. The histogram for two attributes is presented in Figure 1. A similar long tailed distribution exists across all attributes.

Page View Dataset (PVD): This dataset comes from a publisher, where the publisher sells the consumer's information directly to marketers based on contractual pricing [18]. For each page view, the publisher matches the consumer's attributes to those desired by marketers and then offers it to a matched marketer. The contractual mechanism is less studied. Our work applies to both competitive bidding and contractual pricing. This second dataset affords generalization of our approach. The training data comprise 48 million page views from 31 March to 5 April 2017, and the testing data comprise 8 million for 6 April. We refer to this dataset as PVD.

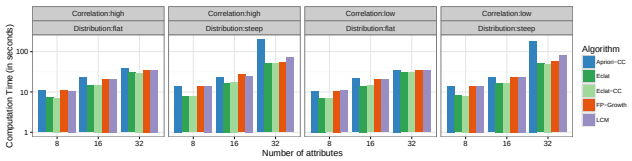

Figure 2: Computation time of FIM algorithms on synthetic data. Average time from three runs, presented for different number of attributes, correlation across attributes and marginal distributions, for support of 10% (other support levels not displayed in the interest of space).

The dataset has 10 attributes, some with a large number of distinct values (Table 1), leading to a total 1.67×10^{26} possible itemsets. As in BRD, attributes display a long-tailed distribution (Figure 1).

4.1 Frequent Itemset Mining

We perform experiments on the synthetic data and two real datasets. The experiments are carried out on a machine with 16GB RAM and 3.5GHz CPU running a Linux distribution. The algorithms included in our analysis are Apriori-CC [4], Eclat [22], Eclat-CC, LCM [21] and FP-Growth [7]. We follow or extend the implementation of Borgelt [3] for these algorithms and record the computation time averaged over 3 runs.

The methods are first compared on the synthetic data (Figure 2). We present results for two levels of correlation and two univariate distribution patterns, across three different number of attributes. For each combination, 10 million events are generated. We make a few broad observations. First, as expected, a higher number of attributes makes the problem more challenging, as reflected in increased computation times. Second, lower correlation leads to a limited decrease in the computation times. Third, having a steep distribution in the univariates leads to higher running times than having flat (equally likely) marginals. This happens because steep distribution and higher correlation lead to higher number of itemsets meeting the threshold, and hence leading to longer run times.

In comparing the algorithms, some of the findings are: one, Eclat-CC performs better than unconstrained Eclat, on average; which itself performs better than Apriori-CC. Considering average ranks across different scenarios, the performance of algorithms in decreasing order is – Eclat-CC, Eclat, LCM, FP-Growth, Apriori-CC. Thus, incorporating CC into Eclat, leads to an algorithm that performs better than the other state-of-the-art algorithms.

On the real data BRD, we find that (Table 2) Eclat-CC is between 2% and 7% better than the next best algorithm, and between 30% and 50% faster than FP-Growth and LCM. On the other real data PVD, Eclat-CC is the best algorithm on a support of 5%, while being close to the best algorithm (Eclat) on a support of 10%. Moreover, in the case of low support (1%), Eclat-CC performs somewhere in between the best algorithm (Apriori-CC) and Eclat. Thus, using categorical constraints into FIM algorithms leads to more efficient implementations in audience size estimation. It is worth noting that the training data for BRD contains 86 million events, larger than the other datasets analyzed, suggesting that the gains for incorporating CC may be more pronounced for larger datasets. We

Table 2: Comparison of FIM algorithms. The average computation time (in seconds) from three runs of the algorithm.

Dataset	Support	Apriori-CC	Eclat	Eclat-CC	FP-Growth	LCM
PVD	1%	**70.7**	83.2	76.8	71.7	72.5
	5%	75.5	46.6	**46.0**	66.4	66.2
	10%	71.8	**41.9**	42.1	59.9	59.1
BRD	1%	160.1	112.0	**105.4**	165.1	167.5
	5%	167.9	80.1	**78.7**	154.3	151.0
	10%	155.6	73.7	**73.0**	135.8	137.6

test this hypothesis with a simulated dataset of 200 million events and 8 attributes, and find that Eclat-CC is 9% better than Eclat and 25% better than FP-Growth.

4.2 Audience Forecasting

To evaluate the accuracy of forecasts, we compare our approach with a naïve, but feasible baseline (FB), an accurate, but also an infeasible baseline using individual target time series (TS) and a machine learning based method. This comparison across both BRD and PVD datasets, is done on two different target sets - FIS and IFIS(Infrequent-FIS). For *FIS*, the support or threshold value is set at 0.01% of the dataset size throughout. We find 0.5 million and 0.7 million FIS in BRD and PVD, respectively. We sample 500 FIS from each dataset with a probability proportional to the support of the itemset, ensuring that itemsets of varying supports are included in the sample.For *IFIS*, we sample 500 infrequent itemsets, among those with less than 0.01% support. We now describe our baseline approaches.

Individual Target Time Series based infeasible baseline (TS): Entire time series is stored for all 500 itemsets in FIS and in IFIS. Forecasts are generated directly by modeling the time series for each itemset, without using conditional probabilities and univariate. This baseline is not bounded by computation time or storage requirements for time series for millions of itemsets. We use it as a boundary condition baseline to compare our approach.

Feasible Baseline (FB): We find all univariate itemsets satisfying a given threshold (of 0.5%). For each of these, we obtain the hourly counts as a percentage of the global counts for that hour. For such time series, we train a model, so that we can forecast the fraction of hourly global count represented by the respective univariate itemset. We also maintain the global time series, for target $G = (u_1, \cdots, u_k)$, where u_l denotes the l^{th} attribute taking any value. For a target $T = (t_1, t_2, ..., t_k)$ we predict the hourly count estimate as $\{P((t_1, u_2, \ldots, u_k)) \times \ldots \times P((u_1, u_2, \ldots, t_k)) \times \hat{s}(G)\}$. Thus, the estimate is obtained by multiplying the global time series forecast by the estimates of percentages of each univariate, obtained from the time series. For univariate values where we do not have a time series, we assume that the percentage varies up to the threshold value used (which in our case corresponds to the interval $[0, 0.005]$). This gives us a ranged estimate, which we average to get the point estimate.

Machine Learning Baseline (ML): We modify the datasets to remodel audience forecasting task as a supervised learning problem. To achieve this, we create a training set by sampling 5,000 itemsets from FIS mined at 0.01% from both PVD and BRD, by sampling with probability proportional to the support, ensuring that itemsets of varying supports are included in the sample (similar to FIS target sets). We collect hourly counts for these itemsets, throughout the training and testing periods. Each row of each data set consists of the itemset, hour of the day and a count of transactions (page views/bid requests) satisfying the itemset in that hour. We drop the day of the week attribute, since our data set is limited to a single week and capturing weekly seasonality is not possible in such a situation. Following the construction of this derived dataset, the forecasting problem is reduced to a regression problem, with a categorical input (itemset, hour) and the output being count of transactions. However, since the total number of levels across various attributes is large (ranging into a few thousands), it is intractable for machine learning models to capture interactions among attributes. Hence, we group all attribute values for which we do not have univariate time series, i.e. present in less than 0.5% of the dataset, into a new level.

The model is first trained on a subset of sampled FIS, and the trained model is used to make predictions for the same benchmark set as other baselines. The model is a multi-layer fully connected network, with dropout, and with an additional embedding layer in the input, implemented in PyTorch [16]. Categorical inputs are mapped to columns of the embedding layer, and then passed through the network to make predictions to minimize MAPE. Hyperparameters are chosen optimally using `hyperopt`[1] library, by considering hyperparameter space spanning embedding layer dimension $\in \{32, 64, \ldots, 128\}$, dropout $\in [0.0, 0.8]$, and number of layers $\in \{1, 2, 3, 4\}$.

Each parameter is sampled uniformly from the corresponding parameter space. We use 6 days of data to train each model, and optimize the hyperparameters according to the MAPE for the 7^{th} day using the TPE algorithm for guiding the search over the hyperparameter space across 1000 trials, with 10 epochs per trial. This search leads to a 3 layer model, with layer dimensions 384, 192, and 64, a dropout of 0.05 and an embedding dimension of 128. With this model, we generate predictions for the ML baseline, and the results are shown in Figure 3. We see that the model obtained by this process performs worse than our approach across all experiments.

We generate time series forecasts for univariate targets in $\mathcal{U}$ (from Section 3.2) using four methods – Exponential Smoothing (ETS), Automatic ARIMA (ARIMA), Neural Network Autoregression (NNAR) and Prophet. We use the respective R packages to automatically choose the best hyper parameters for our time series methods. We use 6 days of hourly data to train and offer hourly

[1] https://github.com/hyperopt/hyperopt

Table 3: MAPEs for univariate time series[a]

Method	BRD	PVD	Method	BRD	PVD
ETS	23.2	13.6	NNAR	24.4	17.2
ARIMA	32.2	17.6	Prophet	23.6	26.5

[a]We also explored a Long Short-Term Memory (LSTM) based time series model, but this failed to provide acceptable accuracy.

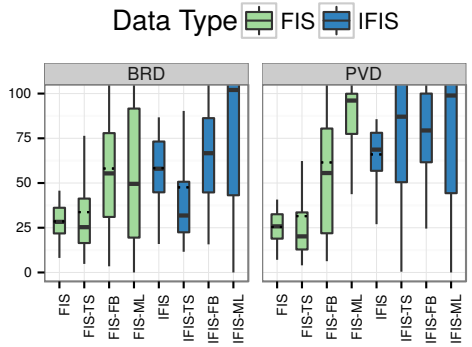

Figure 3: MAPEs (Y-axis) for forecasting: Solid and dashed horizontal lines are median and mean. Comparison is relevant across boxes of same color.

forecasts for the seventh day, capturing daily seasonality. The methods are evaluated using average Mean Absolute Percentage Error (MAPE). Based on superior performance of ETS (Table 3), we decide to use it as the time series model for all evaluations.

Figure 3 shows the results. In box plots, bars of the same color denote results on the same target sets, by data set. Mean and median of MAPE across all itemsets are denoted by dashed and solid horizontal lines, respectively. Mean MAPEs for FIS in BRD and PVD, 29% and 25%, are lower than Mean MAPEs for FIS-TS in both data, although not for medians, reflecting higher variability of FIS-TS (higher spread in box plot). Hence, we claim that the proposed approach is better in terms of mean MAPEs, than the infeasible baseline. Similarly, the proposed approach always performs better than the feasible, but naïve baseline (FB), for both FIS and IFIS; the effect being stronger for FIS. The bad performance of IFIS-TS for PVD may be due to fewer data points of page views for infrequent itemsets. The higher MAPEs for IFIS vs. FIS is due to IFIS itemsets having at most 31 events every hour on average, a small sample to obtain good estimates. Surprisingly, even in small itemsets, our approach that assumes conditional independence, compares reasonably with IFIS-TS.

MAPEs are benchmarked against [10] where ETS produces MAPEs between 10 and 20% for time series in M3 competition [12]. Our MAPEs for univariate time series targets, tasks comparable to the competition, are 14 to 23%. The audience estimation task is more challenging since forecasts are for thousands of attribute combinations, without recording the time series for each. Our MAPE values under 30% is likely to be acceptable in practice.

5 CONCLUSION

Knowing the likely size of audience segments for web traffic can help websites better plan their ad campaign. Audience forecasting is challenging because of the combinatorial explosion in attribute values, each of which could be a relevant target audience. We address this problem with a combination of frequent itemset mining and time series modeling. We are able to achieve good accuracy levels on real datasets from two use cases within online display ad and compare our results with three baseline approaches. We also give a novel FIM approach, specific to categorical characteristics of audience data. We demonstrate the superior performance of this approach over state of the art algorithms by proposing a new simulation framework.

REFERENCES

[1] Deepak Agarwal, Datong Chen, Long-ji Lin, Jayavel Shanmugasundaram, and Erik Vee. 2010. Forecasting high-dimensional data. In *ACM SIGMOD 2010*.
[2] Rakesh Agrawal and Ramakrishnan Srikant. 1994. Fast Algorithms for Mining Association Rules in Large Databases. In *VLDB '94*.
[3] Christian Borgelt. 2012. Frequent item set mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 2, 6 (2012).
[4] Tien Dung Do, Siu Cheung Hui, and Alvis Fong. 2003. Mining Frequent Itemsets with Category-Based Constraints. In *Discovery Science: International Conference*.
[5] Avi Goldfarb and Catherine Tucker. 2011. Online Display Advertising: Targeting and Obtrusiveness. *Marketing Science* 30, 3 (2011), 389–404.
[6] J.D. Hamilton. 1994. *Time Series Analysis*. Princeton University Press.
[7] Jiawei Han, Jian Pei, and Yiwen Yin. 2000. Mining frequent patterns without candidate generation. In *ACM SIGMOD*.
[8] Rob Hyndman and Yeasmin Khandakar. 2008. Automatic Time Series Forecasting: The forecast Package for R. *Journal of Statistical Software, Articles* 27, 3 (2008).
[9] Rob Hyndman, Anne B Koehler, J Keith Ord, and Ralph D Snyder. 2008. *Forecasting with exponential smoothing: The state space approach*. Springer.
[10] Rob J Hyndman, Anne B Koehler, Ralph D Snyder, and Simone Grose. 2002. A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of Forecasting* 18, 3 (2002), 439–454.
[11] Kuang-chih Lee, Burkay Orten, Ali Dasdan, and Wentong Li. 2012. Estimating Conversion Rate in Display Advertising from Past Performance Data. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.
[12] Spyros Makridakis and Michele Hibon. 2000. The M3-Competition: results, conclusions and implications. *International journal of forecasting* 16, 4 (2000).
[13] S. Muthukrishnan. 2009. Ad Exchanges: Research Issues. In *Proceedings of the 5th International Workshop on Internet and Network Economics (WINE '09)*.
[14] Roger B Nelsen. 1999. Introduction. In *An Introduction to Copulas*. Springer, 1–4.
[15] Raymond T. Ng, Laks V. S. Lakshmanan, Jiawei Han, and Alex Pang. 1998. Exploratory Mining and Pruning Optimizations of Constrained Associations Rules. In *In the 1998 ACM SIGMOD International Conference on Management of Data*.
[16] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. In *NIPS-W*.
[17] Jian Pei, Jiawei Han, and L. V. S. Lakshmanan. 2001. Mining frequent itemsets with convertible constraints. In *International Conference on Data Engineering*.
[18] Guillaume Roels and Kristin Fridgeirsdottir. 2009. Dynamic revenue management for online display advertising. *Journal of Revenue and Pricing Management* (2009).
[19] Ramakrishnan Srikant, Quoc Vu, and Rakesh Agrawal. 1997. Mining Association Rules with Item Constraints. In *KDD'97*.
[20] Sean J Taylor and Benjamin Letham. 2017. Forecasting at scale. *The American Statistician* (2017).
[21] Takeaki Uno, Tatsuya Asai, Yuzo Uchida, and Hiroki Arimura. 2004. *An Efficient Algorithm for Enumerating Closed Patterns in Transaction Databases*.
[22] Mohammed Zaki, Srinivasan Parthasarathy, Mitsunori Ogihara, Wei Li, et al. 1997. New Algorithms for Fast Discovery of Association Rules.. In *KDD*.
[23] Weinan Zhang, Shuai Yuan, and Jun Wang. 2014. Optimal Real-time Bidding for Display Advertising. In *KDD'14*.
[24] Weinan Zhang, Tianxiong Zhou, Jun Wang, and Jian Xu. 2016. Bid-aware Gradient Descent for Unbiased Learning with Censored Data in Display Advertising. In *KDD '16*.