Computational Television Advertising

Suhrid Balakrishnan, Sumit Chopra, David Applegate and Simon Urbanek
AT&T Labs Research
180 Park Ave., Florham Park, NJ 07932
Email: {suhrid,schopra,david,urbanek}@research.att.com

***Abstract*—Ever wonder why that Kia Ad ran during Iron Chef? Traditional advertising methodology on television is a fascinating mix of marketing, branding, measurement, and predictive modeling. While still a robust business, it is at risk with the recent growth of online and time-shifted (recorded) television. A particular issue is that traditional methods for television advertising are far less efficient than their counterparts in the online world which employ highly sophisticated computational techniques. This paper formalizes an approach to eliminate some of these inefficiencies by recasting the process of television advertising media campaign generation in a computational framework. We describe efficient mathematical approaches to solve for the task of finding optimal campaigns for specific target audiences. In two case studies, our campaigns report gains in key operational metrics of up to 56% compared to campaigns generated by traditional methods.**

***Keywords*-Television Advertising; Computational Advertising; Optimization; Media Campaign Generation**

I. Introduction

Some of the most memorable advertisements people come across are via television. Arguably the dominant channel for advertising in the past few decades, television advertising remains a robust business[1]. Indeed, while the trends appear to be a decline in revenue associated with print media advertising and a strong growth in revenue for online advertising, television advertising revenues have remained relatively stable, due to televisions' popularity and effectiveness. However, the recent emergence of online television clearly poses a threat to the established mode of advertising. Growing numbers of people are moving away from consuming content via traditional broadcast model television, to consuming content on demand/recorded/online television. This "on demand" medium is more user friendly, as people can choose the shows they want to watch and when they want to watch them. This mode of delivery also provides interesting new opportunities to television advertisers; advertisers can target individual users who they think will have a higher propensity towards their products. The rise of alternatives to the broadcast model make it imperative to develop better techniques for effective advertisement on broadcast television. This need is further bolstered by the the fact that certain programs like sporting events, the news and special events, ceremonies etc., will always be better viewed live.

[1]http://www.emarketer.com/PressRelease.aspx?R=1008788

Perhaps mostly due to when television advertising came about and rose to prominence, the current methodology behind techniques to create television advertising campaigns is a mix of marketing, sales, and some analytics. Considerable human effort is involved in the process of generating an advertising campaign. By contrast, the online advertising world is at the other end of the spectrum. Companies such as Google Inc with Adwords, Yahoo Inc with Yahoo! Publisher Network, and DoubleClick etc., have created large scale advertising networks. These ad networks match customers to advertisers via their interactions online using sophisticated algorithms and models, broadly grouped under the umbrella term computational advertising. We feel that there is a tremendous opportunity to do something similar in the domain of broadcast television. We would like to be able to borrow ideas from computational advertising and apply these algorithms to the television domain.

Towards this end, the primary goal of this paper is to bring television advertising campaign creation into the computational age. By casting the problem of television media campaign creation as an optimization problem, we are motivated to eliminate/reduce otherwise wasted budget, by more precise targeting. We also push towards the automation of campaign creation process. Automatically creating television campaigns will itself have very high impact as it will enable more campaigns to be run faster. Furthermore, quick campaign turn-arounds will enable quicker tuning/modifications to subsequent campaigns based on near immediate feedback.

Our contributions in this paper are threefold. **1.** We propose optimization problems for the task of television media campaign generation for a target audience. **2.** We propose efficient algorithms to solve these tasks and discuss their properties and limitations. **3.** We validate our claims about the efficiency gained by operating in our framework using two real case studies. This is achieved by comparing our proposed techniques against the ones used to generate a traditional campaign.

As an example of the types of gain in efficiency we achieve in this work, we show an up to 56% improvement over traditional campaign design for a particular ad-campaign in terms of a key operational metric, CPM (Cost-per-Mille: which measures the value per thousand impressions) and an up to 18% gain in Reach (which measures the number of unique customers who view the ad).

II. Traditional Campaign Design for Television

The basic goal of any advertising campaign is to provide target customers with relevant advertisements, and television campaigns are no different in this respect. The process begins with advertisers who research their prospective customers/target group for their product/service. The television ad-network (the people who create the ad campaigns) then tries to match the target customer group, as best as they can, to various networks/channels or individual television programs. The goal of this process is to place ads on channels/programs where there is a high chance that the target customers will see, and hopefully act on the ads.

Stated this way, the key influence of television ratings information is quite evident. Ratings information, namely channel/program audience size and its composition (demographic characteristics, such as, age, gender etc.) are the main source of quantitative data used in campaign creation. This data is often used in conjunction with various business rules and more qualitative information to design ad campaigns. Some examples of such qualitative information are, the perception of alignment between the advertiser and the proposed network/channel/program, or any restrictions on times of day to show the ad etc.

This rather manual and iterative nature of traditional campaign design process leaves the door open to more efficient techniques for campaign creation. The algorithms we propose in this paper are able to target users much more efficiently. In other words, we are able to achieve better performance than traditional campaigns, at a lower cost.

A. Ad Campaign Specification

A typical television ad campaign is delivered as a document called a *Media Plan*. A Media Plan specifies the channels and times at which to book the ad spots (also called ad insertions) for the campaign. An excerpt from a fictitious Media Plan is given in Table I.

Table I
EXCERPT FROM A FICTITIOUS MEDIA PLAN.

Campaign Dates: 5.14.2012-6.10.2012

Channel	Day	Date	Daypart	# Insertions
A&E	Mon	5/14/2012	LateNite	3
A&E	Tue	5/15/2012	Fringe	2
A&E	Tue	5/15/2012	Primetime	1
A&E	Wed	5/16/2012	LateNite	3

As Table I shows, each line in the plan specifies the channel, date, and day-part where a (typically) 30-second ad will be placed. *Dayparts* are just a coarse partition of the week by time (see Table II). For the purpose of brevity, we will call such {daypart, channel} combinations *Slots*, in the rest of the paper. The column labeled the number of *Insertions* (# Insertions), specifies the number 30-second ad spots purchased, and to be inserted in that Slot.

A few constraints for television ad campaigns are worth keeping note of. The first is the campaign budget, which is simply the total dollar amount available to spend on the ad-insertions in the plan. A Slot typically has a going rate for each desired 30-second ad insertion. This Slot rate is affected by many factors, such as, time of the day (for any channel, ad insertion in a LateNight daypart is often cheaper than in the Primetime daypart), the popularity of the channel (popular channels which on an average have a large audience are typically more expensive to insert on as compared to channels with small audience) etc. Clearly, the budget of the campaign determines how many total ad spots can be purchased.

A second constraint is that media campaigns are usually specified to run for 4 or 6 weeks. It is possible to concentrate ad insertions on some particular weeks, say, for example, by ramping up insertions before a product launch. Alternatively, some advertisers may desire spreading the plans out uniformly over the entire 4 to 6 week period. In this paper, we remain agnostic to these changes and operate on a weekly basis, using a weekly budget that allocated taking these campaign preferences into account.

Third, inventory constraints, and not budget constraints, often dictate the maximum number of ad insertions on Slots. Many television carriers are allocated a small number of advertising insertions on particular channels in particular Slots as part of their service contract. For instance, four 30-second ad insertions per hour. The length of the Slot (size of its daypart) thus constrains the total number of possible ad insertions for that Slot. For example, the Fringe Slot on any channel, which spans $3-7$ PM every weekday is 4 hours long. Assuming inventory of 4 insertions per hour, this implies that the maximum number of insertions possible in such a Slot is $16\,(4 \times 4)$. In reality however, an advertiser would probably never desire to place a maximal number of ads in any single Slot. Not only is such a large amount of inventory never available (the market place for television advertising is very robust and inventory is almost never unsold), maxing out a Slot would probably result in conflicts in content adjacency business rules - it would be quite annoying for a user to watch the same ad twice or thrice in a row! For these reasons the maximum allowable number of ad insertions per Slot is set to some small value. For this paper, we experiment with values in the range of $2-6$.

The fourth constraint has to do with the fact that only certain channels may be available to place ads on. These channels are also called “Insertable” channels. Examples of non-insertable channels are premium cable channels which are not ad supported (e.g. HBO).

Finally, creating a Media Plan customized for a target audience group requires at least one more specification, namely, the metric to evaluate on. Two metrics which are very popular in the television industry are:

- **CPM (Cost-per-Mille):** reflects the cost associated per thousand estimated views for an ad. Lower CPM is better (you want more impressions on your target customers at a lower cost).
- **Reach:** counts the number of unique target users who are exposed to an ad. Higher reach is better (a larger number of your target customers see your ad).

We next define the problem more concretely by giving an overview of our methodology, and discuss the data we have that enables our solution.

III. Computational Campaign Design for Television

The idea behind the approach proposed in this paper is as follows: We observe and analyze properly anonymized historical television interaction data from our target audience. This data is used to train forecasting/predictive models for the viewership of these audience. The viewership predictions in turn feed optimization algorithms which enables us to create campaigns targeted to optimize various metrics that are important for the specific advertisers.

We obtain the television viewership data of the target audience through a major television provider, and it consists of anonymized Set-Top-Box (STB) interaction data. An STB is a device cable companies provide to individual users through which they access television programming. The interaction data we have access to includes, among other things, which channel the STB is tuned to, and at what time of day. This is not explicitly labeled data, in the sense that users do not confirm whether they were watching the television when the STB was tuned to a particular channel. Hence, there is some amount of inference that needs to be done to determine to a reasonable degree whether to credit a user for having watched a particular channel at a particular point in time. The next section (Section IV) outlines these and other issues pertaining to the data.

Once we have inferred which user watches what channel and at what time, we build predictive models for their expected viewership into the future. The definition of the future is given by the time period in which the advertiser wants to place its ads. As discussed before, these typically correspond to $4-6$ weeks ahead. This step can be thought of as estimating the ratings information, but precisely tuned to the target audience. The predictive models for the target audience is then used to drive the optimization algorithms. We optimize the two most popular metrics used in the television industry, resulting in two Media Plans, one optimized for CPM, and the other optimized for Reach. Furthermore, we propose two types of Reach optimized campaigns: Binary Reach, and Fractional Reach.

The details of the models which generate these different campaigns are given in Sections V and VI. In the next section we give the detailed description of the data set used.

IV. Set-Top-Box Interaction Data

The raw data that we have access to consists of the interaction of anonymous users with their STBs. This stream of user-STB interaction data, features timestamped channel-tune events, such as, channel tunes/changes, change in volume etc.

Given the sensitivity of the data, we took several steps to ensure the privacy of individuals was not compromised. First, only anonymous data was used in this study. In particular, all personal information identifying the users was removed and users were referred to only by a hashed index value. Second, all our results are presented as aggregates and no individual user was singled out for the study, resulting in further privacy protection of individuals.

Recall that our objective is to obtain the STB viewership information from this raw data stream. In other words, from this interaction data, we need to infer which channels do the target audience watch and at what times.

We base our viewership determination and predictive modeling on STB *Impression* counts (a term from online advertising). Impressions, in turn, are defined based on filtered STB channel tune events. These select tune events meet certain activity and inactivity thresholds, which are simple rules based on the time that transpires between channel tunes/changes that users execute using their STB remote control. In particular, we use two filters to identify valid channel tune events. The first filter, called the Activity Filter, removes any viewership that is too small to actually indicate engagement. We do not credit viewership to any STB tuned to a channel for less than 20 seconds in length. The second filter, called the Inactivity Filter, stops giving credit to channel tunes that have a large time interval from the last known remote event (channel change or tune). This filter is crucial because people routinely fail to turn off their STBs when done watching - some turn off just their televisions, and some just never turn their televisions off at all. Our Inactivity Filter stops giving credit to channel tunes after 1.5 hrs of no activity using the remote control. In other words, we count a slot as being viewed by an STB only if there is at least some STB activity (events using the remote control) in the preceding 1.5 hrs.

A. *Impressions data*

After filtering the raw target STB data to obtain valid channel tune data, we further divide all continuous time into contiguous 15-minute blocks. The aggregate channel tune information for any STB (including non-contiguous viewership) on any channel in a block is what we use to define impressions.

Define a Subslot as a particular combination of $\{$15-minute block, channel$\}$. We convert the Subslot viewership amounts, i.e., the number of seconds tuned in, out of the 15 minutes in that Subslot, to impressions via thresholding. In particular, we count a single impression if a particular STB watches more than 5 minutes (non-contiguous, aggregate) of a particular 15 minute Subslot.

Once we have STB impression data, we aggregate it in two different ways to facilitate the two different predictive models we build. A common dimension of aggregation is time. In our campaigns, we have the ability to buy spots for ad placement only at the daypart granularity. This is mainly due to the fact that traditional campaigns have typically operated at this level of granularity. Table II provides the definitions of the dayparts we use in this paper.

Table II
DAYPART DEFINITION

Daypart	Time Period	
Daytime	M-F	6AM - 3PM
Fringe	M-F	3PM - 7PM
Late Night	M-Su	12AM - 6AM
Prime	M-Su	7PM - 12AM
Weekend	Sa-Su	6AM - 7PM

Our first aggregated dataset that we use for forecasting is the Slot level impressions dataset. As a reminder, a Slot, is defined as a $\{$daypart, channel$\}$ combination, and thus aggregates over the many Subslots contained in that Slot. In the Slot level dataset, weekly files hold the Slot impression values aggregated over the users. This dataset is used in generating CPM-optimized media planning (Section V).

Our second dataset used for forecasting, is the anonymized user level impressions dataset. It is a weekly dataset holding the user/STB slot aggregated impression values. This dataset is used to build Reach-optimized media plans (Section VI).

A few other points worth mentioning about the data: First, we only pull usage data on insertable channels. As mentioned previously, these are the channels where a cable provider will have available inventory and the ability to place ads. Second, we "normalize" the raw user viewership to local times. This aligns 10 AM regardless of the time zone in which the ad is to be shown. This makes sense for two reasons: 1. The cable television provider can only insert ads locally. 2. We want to devise national campaigns.

The next section describes media campaigns and gives the details of modeling and optimization problems.

V. CPM Optimized Media Plan

CPM is defined as cost per thousand impressions. At the level of a campaign, given an advertising budget, CPM is the ratio of the campaign cost (in dollars) to the number of impressions (in thousands) that the campaign achieves. Similarly, one can also define the CPM at the level of a Slot. CPM of a Slot is the cost of a single ad insertion in that Slot, divided by the number of impressions the ad achieves in that Slot.

The algorithm for creating a CPM optimized Media Plan for the target group can be seen as a two step process:

Step 1. Using the past viewership history of the target audience, learn a model to forecast the number of target impressions in all the insertable Slots in the time period for which the media campaign is sought.

Step 2 Select Slots for the media campaign which optimize the campaign CPM using these predicted Slot target impressions estimates.

Once we have accurately estimated the number of impressions for all the insertable Slots, creating a Media Plan that optimizes the CPM (Step 2 above) simply involves sequentially picking slots that provide the best value for impressions. Indeed it is not hard to show that the following algorithm is optimal:

Figure 1. The algorithm optimizing CPM

Input: predicted Slot impressions $y(s)$
  Slot cost $c(s)$; budget B; max. insertions m
Output: A Media Plan MP_{cpm}
Initialize $MP_{cpm} = []$ $\{$empty Media Plan$\}$
Sort the Slots in ascending order of their Slot-CPMs
  $c(s)/y(s)$.
while $B \geq 0$ $\{$Budget not exhausted$\}$ **do**
  Pick the next Slot s in the sorted list
  Add m insertions of s to the Media Plan MP_{cpm}
  Set $B = B - mc(s)$ $\{$update the Budget$\}$
end while

The only unknown in the above algorithm is the predicted number of target impressions per Slot. The performance of the CPM optimized model strongly hinges on the accuracy of these predictions.

As stated in Section II-A, we handle the Media Plan for each week independently. We make a further independence assumption for the Slot impression prediction task as well. In particular, for each week, we train an independent model to forecast the number of impressions for Slots in that week. In other words, the number of forecasting models we train is equal to the number of weeks for which the media campaign is sought. As an example, for a 4 week campaign, we would create 4 different predictive models. The first model takes as input the historical viewership data of target audience (training data), and predicts Slot impressions 1-week ahead from the last date of the training data (for the first week of the plan). Similarly, the second model predicts the impressions 2-weeks ahead from the last date of the training data (for the second week of the plan), and so on till week 4. Matching intuition, it gets harder and harder (and then levels off) the further out we are trying to predict. Thus one would expect that our 1-week ahead model will do better predictively than the 4-week ahead model. Indeed, the fact that we expect different amounts of uncertainty depending on the prediction horizon was the reason why we chose to model each week separately.

Next, we describe the regression based predictive model for the target Slot impressions prediction task that we use (Step 1 above).

A. Predictive Model for Target Slot Impressions

We use Ridge regression [1] to predict the Slot target impressions. Essentially, we model the number of target impressions in each Slot as a linear combination of the number of target impressions in the same Slot in the previous k weeks. k is a hyper-parameter, whose value we choose via cross-validation.

Recall, that we treat each week independent of the other weeks, and have a separate weekly prediction model, which predicts the impressions of all Slots in that week.

Let i be an index over the dayparts in a week. Then from Table II, i ranges from $i \in [1 \ldots 26]$. Let there be M insertable channels, and let j be an index over them: $j \in [1 \ldots M]$. Then any Slot $s_{i,j}$ corresponding to the daypart i, and channel j, in a week t is identified by s_{ijt}. Let $y_{s_{ijt}}$ denote the number of target impressions in the slot s_{ijt}. Mathematically, we estimate these number of impressions as:

$$\hat{y}_{s_{ijt}} = \sum_{m=t-k}^{t-1} w_i y_{s_{ijm}}, \tag{1}$$

where w_i's are the parameters of the model we need to estimate. Let W denote the vector of parameters $[w_1, w_2, \ldots, w_k]$. In order to learn the weights W, we create a training dataset from the historical target impressions of the current Slot $s_{i,j}$, and use Ridge regression to compute W. Ridge regression lets us add a shrinkage factor to the weights via a tunable parameter λ.

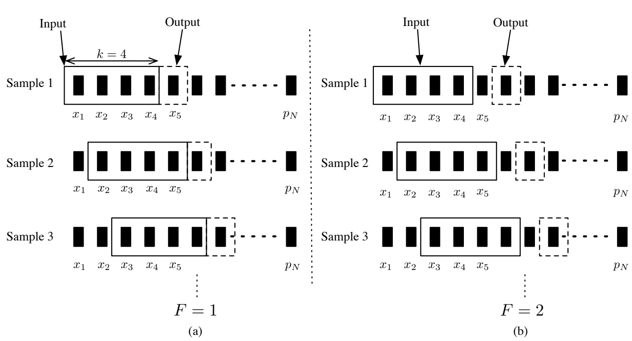

Figure 2. *Figure describing how the training dataset is generated for each Slot $s_{i,j}$, for $k = 4$ for different values of F, namely $F = 1$, and $F = 2$. The total number of such samples (input:output pairs) generated will be equal to $N - k - F$.*

We now describe how we build the training data to learn the parameters W. We encourage the readers to refer to Figure 2. Assume for the current Slot s_{ij}, we have the target impression counts for the past N weeks. To simplify notation let us denote these impressions by $x_1, \ldots, x_N$. Thus $x_1 = y_{s_{ij1}}, x_2 = y_{s_{ij2}}$, and so on. Let us also assume that we are looking back at the window of size k, and we are forecasting $F(= 1)$ one week ahead. Then for any response week t ($t \in [k + F, \ldots, N]$), we associate k input weeks indexed by $t - F, t - F - 1, \ldots, t - F - (k - 1)$, and create a single training sample (an input-output pair) $(\{x_{t-F}, \ldots, x_{t-F-(k-1)}\} : x_t)$. The input of this training sample consists of the impressions from the past k weeks, which is paired with the impressions of the response week t. We repeat this process for all values of t such that $t \in [k + F, \ldots, N]$. Let the number of such pairs generated be m. Then we define a $m \times k$ matrix X, such that its i-th row stores the impressions associated with the input weeks corresponding to the i-th training sample. We also define a vector Y of size m, which stores the impressions of the corresponding response (output) weeks. Then the loss L we minimize in order to obtain the weights W is:

$$L = (Y - XW)^T (Y - XW) + \lambda W^T W. \tag{2}$$

The closed form solution of the above system [1] is:

$$W = (X^T X + \lambda I)^{-1} X^T Y, \tag{3}$$

where I is an identity matrix of size $k \times k$, and λ is the regularization parameter whose value is computed using a validation process. Any linear algebra package will be able to handle the above system, and since we estimate these parameters per slot, the systems of equations themselves are very small. Once these parameters are estimated, we use equation 1 to predict the target impressions for any Slot $s_{i,j,t}$ in week t.

There are a number of advantages associated with this prediction model. First, it is simple. While it is more general than an auto-regressive model, it is less sophisticated than many other time series models which take into account other factors, such as, seasonality. This simplicity also leads to very fast training and deployment of the model. Second, since the matrix X does not change across the various $1, 2, 3$ and 4 week ahead predictions, the matrix inverse in Eq. 3 only needs to be evaluated once for all the 4 models. The only variable that changes is Y. Third, since each Slot per predictive week is handled independently of the other, the model is highly parallelizable.

The independence assumptions we make for computational convenience does result in a few limitations of our model though. Weeks are treated independently, as are the Slots. Consecutive dayparts on the same channel are also considered independent. Presumably, an approach that will make less drastic independence assumptions, and will allow to share data across the Slots and the weeks, would lead to more accurate predictions. Improved model predictions will lead to improvements in the generated Media Plans.

VI. Algorithms for Reach-optimal Media Plans

While CPM as an objective aims to optimize the number of cost-effective impressions, it doesn't explicitly take into account what fraction of the target STBs your campaign ads will be shown to. As an example, consider two Slots (Slot 1 and Slot 2) that have the same ad insertion cost. Suppose the program that airs on Slot 1 has a medium-sized fan base, but these fans are die-hard fans. It is viewed by 1000 target STBs, who always watch all of this program, translating to roughly 8 impressions per STB. This results in 1000×8 = 8K impressions for Slot 1. Slot 2 features a more popular program, but one where people constantly tune in and tune out. So, say 4000 target STBs each roughly having 2 impressions each, also resulting in 8K target impressions for the Slot. In terms of CPM, Slot 1 and Slot 2 are identical, but in terms of number of unique STBs to whom the ad will be shown (our definition of reach), Slot 2 reaches far more STBs (4 times as many).

It seems evident that there will be instances when clients would like the ability to optimize a Media Plan for target reach (and so would prefer Slot 2 in the example above). This might occur, for example, during the launch of a new product/service, when the client is interested in generating awareness about their new product/service.

Optimizing reach, of course, entails a very different kind of optimization problem. We next describe the reach optimization algorithm for a binary viewership data case.

A. Binary viewership

In the simpler binary viewership data setting, the target users are all assumed to either completely watch Slots or not. In what follows, let U be the set of all users (STBs), S be the set of all Slots, and for $u \in U, s \in S$, let $y(u,s)$ denote the viewership data for user u of Slot s. In the restricted binary viewership setting, we threshold this viewership quantity, with a tunable threshold scalar parameter τ, setting $y(u,s)$ to 0 or 1 depending on whether the corresponding viewership amount is greater or less than τ (in the next subsection we describe an alternative probablistic way to create the data).

Now in this simple binary case, a non-zero $y(u,s)$ implies we have high confidence in our prediction that a particular user has a high propensity to watch that particular channel in that daypart. Correspondingly, if we place an ad in this Slot, we would expect to reach that user (i.e., we expect the user to see an ad we place in that Slot).

Note that even if we simplified the problem by assuming we knew which Slots everyone would watch (thus removing the need for viewership predictions in this simplified scenario), picking the Slots that would maximize reach given a budget is an instance of the budgeted maximum coverage problem [2]. Even with unit Slot costs, this is an NP-hard problem, by a straightforward reduction from the set cover problem.

While integer programming provides one feasible path towards obtaining optimal or near optimal Media Plans, we instead choose to apply a greedy heuristic which provides a provable worst-case approximation ratio of $(1 - 1/\sqrt{e})$ [2], but in practice performs significantly better. This algorithmic choice makes even more sense when we consider the fact that we only have access to an an estimate of Slot viewership per target user, which is imperfect/approximate in itself. Our approximate solution is shown in Figure 3.

Figure 3. The greedy heuristic for binary reach

Input: Binary $y(u,s)$ user,Slot viewership data
  Slot cost $c(s)$; budget B
Output: Media Plan MP_{reach}
Initialize $MP_{reach} = []$ {Empty Media Plan}
Initialize $r(u) = 1$ for $u \in U$ {$r(u)$ = remaining reach}
while $B \geq 0$ {Budget not exhausted} **do**
  Pick the Slot s that maximizes remaining reach divided by cost: $(\sum_{u \in U} r(u) y(u,s))/c(s)$
  Add a single insertion of s to the Media Plan, MP_{reach}
  For all $u \in U$, if $y(u,s) = 1$, then set $r(u) = 0$
  Set $B = B - c(s)$
end while

At a high level, the greedy solution just sequentially picks Slots that maximize the remaining reach per cost. So, the first Slot picked is the Slot that has the highest reach per cost. Once this Slot is chosen, we remove all the users from the dataset that we assume will be covered by an ad placed in this Slot. We next examine the remaining users data, and pick the next Slot that maximizes reach per cost on these users, and so on. Thus iterations consisting of picking Slots to maximize reach, and modifying the data assuming those users are reached. Although the worst case bound on this algorithm is a factor of $(1 - 1/\sqrt{e})$, in practice on our datasets the performance is extremely good and actually very close to a linear programming-based lower bound.

We next describe the fractional version of the same reach optimization problem. As stated before, the main difference is that the viewership data we have per user is the probability that they will watch a particular Slot (and not the hard thresholded binary version of this data above).

B. Fractional viewership

In the fractional reach problem, $y(u,s)$ will represent user Slot viewership probabilities. Given any user viewership in a Slot, these probabilities sum to one over all the Slots in the same time interval. As a first step, we will need to estimate the effect of placing an ad in a particular Slot on reaching a particular user.

Suppose user u watches a Slot with probability $y(u,s)$, and we place an ad in that Slot. Then the probability that they do not watch that ad is $1 - y(u,s)$. Continuing in this vein, suppose we place multiple ads in Slots $s_1, s_2, ..., s_n$. The probability that u has not watched any of these ads is $\prod_{i=1}^{n}(1 - y(u,s_i))$. This is because the event that u has not been reached is exactly the compound event that she is not reached by any ad individually. Thus, the probability that u

is reached is $1 - \prod_{i=1}^{n}(1 - y(u, s_i))$, which, summed over all users, is what we call the "fractional reach".

This lays the ground for the intuition behind the fractional reach optimization algorithm. Because of the non-linear product $\prod_{i=1}^{n}(1 - y(u, s_i))$, fractional reach is not easily modeled as an integer linear program, and we again use a greedy heuristic. The algorithm is shown in Figure 4. The

Figure 4. The greedy heuristic for fractional reach

Input: Fractional $y(u, s)$ user,Slot view probabilities
Slot cost $c(s)$; budget B; max. insertions δ
Output: Media Plan $MP_{fracReach}$
Initialize $MP_{fracReach} = []$ {Empty Media Plan}
Initialize $r(u) = 1$ for $u \in U$
{$r(u)$ = prob. that u has not yet watched an ad}
while $B \geq 0$ {Budget not exhausted} **do**
Pick the Slot s that has not yet been inserted δ times, and that maximizes $(\sum_{u \in U} r(u)y(u, s))/c(s)$ (remaining reach divided by cost)
Add an insertion of s to the Media Plan, $MP_{fracReach}$
For all $u \in U$, set $r(u) = r(u)(1 - y(u, s))$
Set $B = B - c(s)$
end while

core algorithm is quite similar to the binary case, with the exception that now after selecting any particular Slot s with fractional reach, the remaining reach $r(u)$ for each user may be reduced, but doesn't become zero unless $v(u, s) = 1$. As a further consequence, the same Slot now has the chance to be selected more than once, provided the gain in fractional reach per cost it provides is more than the fractional reach provided by any other Slot (per cost). We give the details about the predictive model for user level Slot impressions.

C. Predictive model for user-based Slot viewership

We use an extremely straightforward model for user based Slot viewership prediction. We predict Slot viewership for a user using a simple windowed average:

$$\hat{y}_{s_{ijt}} = \frac{1}{k} \sum_{m=t-k}^{t-1} y_{s_{ijm}}. \tag{4}$$

This simple model is justified in our context for at least three reasons. First, people tend to be quite consistent in their television viewership, and this model forms an extremely simple, and yet powerful baseline estimator. Second, there are potentially a very large number of target STBs which will all need individual models (millions of STBs in the target is not at all uncommon). This places practical restrictions on the computational burden user level impressions modeling can impose. Finally, the main focus of our work is on the optimization formulation of media campaign generation and user or aggregate Slot impressions prediction. Any model for individual STBs that makes better predictions will only strengthen our campaign performance results.

VII. Experiments and Results

We now examine two case studies in an attempt to quantify the benefits of the computational targeted media campaigns that we propose.

A. Case Studies

We consider two advertising campaigns from earlier this year (2012). In both of these cases, traditional campaigns were created around target audience specifications in the customary manner. This involved matching the channels to the demographic attributes of the target audience (see Section II). Client sensitivities prevent us from divulging the details of both the campaign specifications. However, both the clients are large companies, one being an upscale car manufacturer, and the second being a large investment/retirement planning firm. In the rest of the paper, these clients along with their campaigns will be referred to as CAR and BANK respectively.

The target audience for both campaigns is similar, in that both the clients are interested in higher income customers of an overlapping, but slightly different age demographic. The number of target STB sets for both the clients were in the hundreds of thousands. In addition, the CAR client specified the target to be males. Consequently the number of target STBs for it were a little smaller. Furthermore, the objectives for both campaigns were different. The BANK was seeking a campaign to maximize CPM over the target audience, while the CAR was seeking a campaign which maximized Reach instead. Both campaigns were 4 weeks long and the budgets for both the campaigns are in the same ball-park, with the BANK having a slightly higher budget. Both the clients desired an even campaign, and accordingly we allocated the total budget equally to all four weeks.

Since these were traditional television campaigns we had an opportunity to create a variety of optimization based campaigns, and evaluate against them. Our experimental methodology for both case studies is similar to back-testing in investment analysis. We use the historical viewership data of the target STBs to generate the two types of data described in Section IV, namely, Slot-based data, and User-based data. Note that we only use viewership data from prior to the start of the campaign. We then train our prediction models and generate various campaigns corresponding to different values of the hyper-parameters. In particular, we explored two hyper-parameters: 1. the maximum number of insertions allowed in each Slot, and 2. the two objective functions, namely, CPM and Reach. We then "deploy" our campaigns in parallel to the traditional campaign. After the campaign period is over, we collect the ground truth target STB viewership data. This data is also converted to impressions data, against which we evaluate the performance of all our campaigns, as well as that of the traditional campaigns.

B. Results

We first evaluate our predictive models, as these form the basis for our CPM and reach optimization procedures. As can be seen in Figure 5, our per Slot predictions are very good and show no obvious biases. There are a few Slots for which our impressions predictions are very different than the ground truth number of impressions. Upon examination however, we found that these Slots mostly corresponded to isolated and unpredictable events in television programming, such as, the broadcast of a popular college game in a Slot where there is usually sports news. Our user level predictive results while acceptable, are clearly not quite as accurate. There is more unexplained variance/error in the predictions. This is to be expected, as aggregation of Slot impressions from all target users, week to week, is bound to smooth out some user level noise.

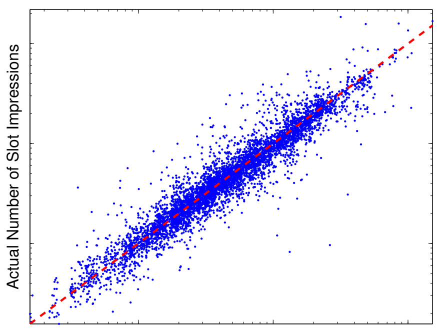

Figure 5. *Scatterplot showing Slot predictions vs. ground truth number of impressions for the campaign length. The plot is on a log-log scale, and the $y = x$ line is shown (ideal predictions would all lie on this line). Axis label values have been removed from the plot to protect sensitive information.*

We next proceed to the results of the media campaigns themselves. Table III shows the results for the CAR case study, and the corresponding table for the BANK case study is Table IV. In addition to our predictive model based results, using CPM, Binary Reach, and Fractional Reach as the objective functions, we also show results for the traditional campaign (with a maximum of 2 insertions per Slot), and three other retrospectively optimal campaigns. These campaigns are marked with a "*" symbol beside their names.

These "*"-ed campaigns (namely CPM*, FRAC*, REACH*), are computed after the campaign dates have elapsed, and are essentially derived from the same optimizations we propose. The only difference is that the target impression data used in these campaigns is not based on forecasting. Instead, we use the ground truth data to infer the number of impressions and probability for watching a Slot. For instance, CPM* is the optimal campaign for the time period in terms of CPM. These campaigns provide a nice set of bounds for our methods - we cannot hope to do as well as these campaigns, since we are additionally using predictive estimates, but the bounds do allow us to gauge how well we are doing with respect to the best possible campaigns in hindsight.

The following are the descriptions of the abbreviations in the two tables.

ALG. The algorithm used to create the Media Plans. CPM refers to the CPM optimized algorithm described in Section V, REACH refers to greedy binary reach algorithm of Section VI, FRAC is the greedy fractional reach algorithm from Section VI-B, and the traditional campaign is referred to by TRAD. There are also the starred retrospectively optimal versions of CPM, FRAC and REACH algorithms.

INS. This is another hyper-parameter we explored while generating the campaigns and it refers to the maximum number of ad insertions allowed per Slot. We experimented with values $2, 4$, and 6.

CPM. For all the campaigns we report the test/ground truth cost per thousand impressions.

NEU. This is a proxy for reach, and is defined as the number of exposed users. This is the number of unique target users who watched a Slot where an ad was placed. To protect sensitive information, CPM and NEU for the traditional campaigns were scaled to 10 and 100,000 respectively. This results in a CPM scaling factor and an NEU scaling factor. All other CPM and NEU numbers are scaled with their corresponding scaling factor.

NI. Defined as Number of Insertions, it refers to the number of ad insertions a media campaign specifies in the four week period.

AE. The Average Exposures (AE) is defined as the average over target users of the number of advertised/chosen Slots the user watched.

EA. The Expected number of Ads seen, is simply the average over users of the expected number of ads seen by any user. We obtain this expectation assuming a binomial model with the average test/ground truth probability of watching a Slot for that user and taking the number of exposures as the number of trials.

The results validate our motivating claim that there is indeed substantial room to improve traditional campaign creation strategies. Our algorithms deliver superior performance on both CPM and reach. For instance, we can make gains in CPM up to 56% (for 6 insertions) if you consider aggressive inventory schemes. More practically, gains from $7-30\%$ are very easily achieved by simply switching to CPM-based optimization for campaign creation.

We further see that while our algorithms perform very well, there is also some room for further research on the ratings forecasting, with the difference in CPM between our proposal and CPM* being roughly 15%.

Our results for Reach are equally promising. Looking at NEU, using the binary reach algorithm we can cover around 18% more target STBs. Assuming target groups in the hundreds of thousands, this translates to being able to expose an additional tens of thousands of target customers to the advertisement. Somewhat surprisingly to us, the fractional reach optimization formulation results in much smaller increased reach and more closely resembles results from CPM optimization, rather than the binary reach optimization. We are currently investigating this further.

There appears to be a distinct trade-off between minimizing CPM (low cost impressions) and exposing more unique customers (Reach). By and large, in the optimization steps, you are forced to choose between inserting maximally on Slots which have high Slot-CPM "piling up", vs. trying to cover all the target users, and thus choose a diverse set of "thinly spread" out Slots. Each could be advantageous depending on the specific campaign priorities.

Table III
CAR RESULTS. PLEASE SEE THE TEXT FOR A DESCRIPTION OF THE QUANTITIES IN THE COLUMNS.

ALG	INS	CPM	NEU	NI	AE	EA
TRAD	2	10.00	100000	2856	14.35	2.16
CPM*	2	7.98	103729	2431	16.43	2.48
CPM*	4	5.60	95176	3969	25.82	3.88
CPM*	6	4.44	89309	5328	35.16	5.28
CPM	2	9.28	100867	2607	15.41	2.30
CPM	4	6.53	89679	4277	25.22	3.76
CPM	6	5.09	84304	5889	34.78	5.18
FRAC	2	9.66	105504	2604	8.05	1.20
FRAC	4	6.85	97970	4078	6.86	1.02
FRAC	6	5.46	96737	5380	6.63	0.99
FRAC*	2	8.34	111767	2425	9.42	1.42
FRAC*	4	6.15	110092	3701	8.78	1.32
FRAC*	6	5.03	109516	4913	8.57	1.29
REACH	2,4,6	15.23	115652	1708	10.09	1.49
REACH*	2,4,6	12.89	118114	1702	10.80	1.61

Table IV
BANK RESULTS. PLEASE SEE THE TEXT FOR A DESCRIPTION OF THE QUANTITIES IN THE COLUMNS.

ALG	INS	CPM	NEU	NI	AE	EA
TRAD	2	10.00	100000	2983	14.61	2.22
CPM*	2	8.13	105325	2595	16.16	2.46
CPM*	4	5.70	96341	4156	25.89	3.93
CPM*	6	4.52	91111	5506	35.02	5.31
CPM	2	9.42	101778	2799	15.29	2.32
CPM	4	6.61	90792	4339	24.99	3.77
CPM	6	5.13	85054	5977	35.66	5.36
FRAC	2	9.82	107286	2704	8.15	1.23
FRAC	4	7.00	100058	4128	6.87	1.04
FRAC	6	5.59	98877	5431	6.62	1.00
FRAC*	2	8.49	113793	2506	9.34	1.42
FRAC*	4	6.30	112237	3783	8.87	1.35
FRAC*	6	5.16	111693	5004	8.67	1.32
REACH	2,4,6	15.23	117874	1721	10.12	1.52
REACH*	2,4,6	12.95	120222	1720	10.81	1.63

Lastly, Figure 6 shows the percentage of budget dollars saved by different CPM optimized Media Plans against the traditional campaign, for the same number of target impressions. Clearly, the CPM optimized campaigns have a significant monetary benefit over the traditional campaigns.

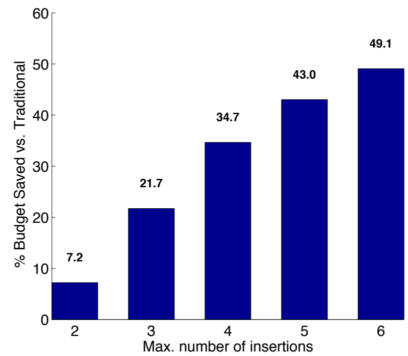

Figure 6. *The percentage of the budget saved by the CPM optimized CAR campaign versus the traditional campaign, for the same number of target impressions.*

VIII. RELATED WORK

To the best of our knowledge, we are not aware of any work in the literature that addresses computational approaches to television media campaign creation. While certain companies do offer products which appear to be very related[2], the algorithms are proprietary. However, there is work on a few related strands of research.

The first of these is television ratings prediction. This is a well studied problem and a number of approaches have been discussed in the literature. Numerous companies are in the business of ratings estimation, the most prominent of whom is probably Nielsen. Unlike our work, Nielsen bases its ratings estimates primarily using data from a random sample of US households. These households are paid to be part of the ratings system. This allows for explicit data collection, as it is incumbent on the participants to carefully report exactly what they watch. While this is extremely clean data, it does suffer from sample selection bias. Furthermore, since the number of sampled households is only in the thousands, it cannot fully utilize the amount of information available at the scale of a larger content provider, such as a national television provider. Other companies report ratings using a mix of implicit data measured on users, and explicit data. An example is TiVo, whose content delivery model is very much like video on demand. Ratings prediction work in the literature has typically been via time-series modeling approaches, with the competitive model being a classic reference [3], [4]. These approaches do adjust for external factors, such as, seasonality. A more contemporary take on the time series approach is due to Meyer et. al., in [5].

[2]For example Nielsen's website where they advertise: Optimize your TV plans to maximize reach for less spend

However, as we stated before, our main contribution is media campaign creation and not purely ratings prediction. This is why we explicitly separate these two problems in our work, and aim to improve our results by using the existing research for ratings prediction. Tellis et al [6], use regression analysis on ratings information to examine which specific ads to place (ad-effectivity), when to place them in a week etc. Their methodology and scope is very different than ours however, because they do not have STB level data and are limited to a post-ad-placement analysis of viewership.

Another stream of related work deals with online television [7], [8]. The key motivation in this line of work is to leverage the technology and algorithms behind existing web advertising networks. Since they deal exclusively with online television delivery and not broadcast television, the issues they deal with and research challenges they face are of a very different nature.

IX. Discussion and Future Work

The media campaign optimization strategies we suggest also open up a few interesting areas of future work. As seen in the slot prediction results and in the gap between CPM* and CPM in in Section VII-B, there is an opportunity for improved predictions that would result in even better media plans. This is particularly true in the case of reach based optimization (both binary and fractional), where the user Slot prediction model has (by nature) fairly high variance.

An alternative strategy, would be to explicitly model this uncertainty, and account for it in a principled manner via portfolio optimization. We would first approximately characterize the distribution of Slot impression counts, say for example, by the first two moments, i.e., by an expected value and variance. Given these Slot count mean and variance estimates, the idea behind the application of portfolio optimization would be to explicitly pick a limit on the amount of uncertainty tolerable in a media plan. We can then solve for an optimal plan with this additional limit on uncertainty applied as a constraint. The intuition is that optimizing CPM assuming slot predictions having equal/no uncertainty, is risky. There might be cases where we favor a less optimized but more stable solution - one that chooses to not favor slots that have highly variable week to week viewership, for instance.

A second opportunity that we are currently investigating is principled methods to actively trade-off CPM for Reach and vice versa. Essentially CPM and Reach are objectives that are at odds with each other. CPM favors piling up on low cost high impression Slots, whereas Reach (binary) favors spreading thin to cover the users. If you imagine placing points for Media Plans on the plane spanned by CPM on one axis and Reach on the other, the two points achieved by applying the binary reach algorithm and the CPM algorithm lie in opposite corners. An interesting idea of immense practical value would be to develop techniques that would allow one to systematically explore the 'efficient frontier' of solutions between these points. In other words, find the set of dominant solutions between the points. This would allow practitioners to systematically and near optimally trade-off reach for CPM or vice-versa and design campaigns that precisely meet their expectations.

X. Conclusion

In this paper, we propose a new way to advertise on broadcast television. We formulate the task of generating a media campaign as a combination of prediction and optimization problems. We propose novel models which are trained to generate media plans which are either CPM optimal or Reach optimal. We validate the superiority of our approach by comparing the performance of media plans generated by our models against the ones generated by the traditional methodology on two real-life case studies.

Acknowledgment

The authors would like to thank the AdPlacement team and particularly, Micheal Stack and Michael McCarthy for numerous discussions, ideas and comments on the manuscript. The authors would also like to thank the anonymous reviewers for their valuable feedback.

References

[1] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 42, no. 1, pp. 80–86, 2000.

[2] S. Khuller, A. Moss, and J. S. Naor, "The budgeted maximum coverage problem," *Inf. Process. Lett.*, vol. 70, no. 1, pp. 39–45, Apr. 1999.

[3] D. Gensch and P. Shaman, "Models of competitive television ratings," *Journal of Marketing Research*, vol. 17, no. 3, pp. 307 – 315, 1980.

[4] ——, "Predicting tv ratings," *Journal of Marketing Research*, vol. 20, no. 4, pp. 85 – 92, August 1980.

[5] D. Meyer and R. J. Hyndman, "Rating forecasts for television programs," *Monash Econometrics and Business Statistics Working Papers*, no. 1/50, March 2005.

[6] G. J. Tellis, R. K. Chandy, D. J. MacInnis, and P. Thaivanich, "Modeling the microeffects of television advertising: Which ad works, when, where, for how long, and why?" *Marketing Science*, vol. 24, no. 3, 2005.

[7] Y. Interian, Kaustuv, I. Navernioukk, P. J. Opalinski, S. Dorairaj, and D. Zigmond, "Do viewers care? understanding the impact of creative on tv viewing behavior," *Re:think 2009: The ARF Annual Conference*, 2009.

[8] Y. Interian, S. Dorai-Raj, I. Naverniouk, P. Opalinski, Kaustuv, and D. Zigmond, "Ad quality on tv: predicting television audience retention," *Proceedings of the Third International Workshop on Data Mining and Audience Intelligence for Advertising*, pp. 85 – 91, 2009.