Prediction and Optimal Scheduling of Advertisements in Linear Television (Clypd Problem)
MPI 2015 Report

Ghan Bhatt
Tennessee State University
gbhatt@tnstate.edu

Simon Burhoe
University of Massachusetts Amherst
burhoe@math.umass.edu

Michael Capps
Colorado State University
cappsm@rams.colostate.edu

Christina Edholm
University of Nebraska-Lincoln
cedholm2@math.unl.edu

Fadoua El Moustaid
Temple University
fadoua@temple.edu

Tegan Emerson
Colorado State University
emerson@math.colostate.edu

Pak-Wing Fok
University of Delaware
pakwing@udel.edu

Nathan Gold
York University
ngold5@my.yorku.ca

Ryan Halabi
UC Davis
rghalabi@math.ucdavis.edu

Madelyn Houser
University of Delaware
mhouser@udel.edu

Peter Kramer
Rensselaer Polytechnic Institute
kramep@rpi.edu

Wayne Lee
University of North Carolina Chapel Hill
waynelee1217@gmail.com

Qingxia Li
Fisk University
qli@fisk.edu

Weiqiang Li
University of Delaware
weiqiang@udel.edu

Dan Lu
Rensselaer Polytechnic Institute
lud@rpi.edu

Mark Panaggio
Rose-Hulman Institute of Technology
panaggio@rose-hulman.edu

Yuzhou Qian
Rensselaer Polytechnic Institute
qianyuzhou1@gmail.com

Star-Lena Quintana
Temple University
starlenaq@gmail.com

Lou Rossi
University of Delaware
rossi@math.udel.edu

Deborah Shutt
Colorado School of Mines
dshutt@mines.edu

Vicky Chuqiao Yang
Northwestern University
chuqiaoyang2013@u.northwestern.edu

Yingxiang Zhou
University of Delaware
yxzhou@udel.edu

August 15, 2015

Problem Presenter: Marco Montes de Oca

1 Introduction

Advertising is a crucial component of marketing and an important way for companies to raise awareness of goods and services in the marketplace. One way for advertisers to spread their message to consumers is through "linear" (i.e. traditional, not on-demand) television. In principle, advertising through linear TV allows companies to reach large numbers of people using relatively few advertisements.

However, many advertisements do not reach their intended audience, resulting in a waste of resources for the company and a disenchanted audience. Advertisers wish to create campaigns that can reach their target demographic, within budget. Media companies (such as Comcast and Cox) or networks (such as ABC, NBC and Fox) would like to schedule advertisements in such a way as to maximize revenue. Given that television viewership (number of "impressions") is often difficult to predict, generating optimal advertising schedules is an open and interesting mathematical problem which we address in this report.

Specifically, we study three problems related to optimally creating advertising schedules:

(a) Given the number of impressions over a certain time period, predict the number of impressions in future.

(b) Given the predicted number of impressions in future and a list of orders from advertisers, satisfy these orders to maximize revenue for the media company.

(c) Given the number of impressions over a certain time period, predict the *reach* of television programs in the future.

This report is divided into 4 parts. In section 2, we attempt to predict impression numbers a few weeks in advance using realistic viewership data provided over a period of 38 weeks. In section 3, we propose an algorithm, based on a binary integer program, to optimally schedule advertisements to maximize revenue. In section 4, we develop a scheme to predict the reach (defined as the number of unique impressions) of an advertisement. Finally, in section 5, we summarize our findings and draw some conclusions.

2 Predicting Number of Impressions

Figure 1 shows a time series for the number of impressions over a period of about 270 days. Qualitatively, the signal is noisy with large spikes in the viewership. We shall use this data to predict future impressions.

2.1 Spectral Analysis

Although the data appears noisy, we show in this section that there are clear periodic trends in the data. These are mostly likely driven by the periodic nature of the channel programming. Our assumption is that the number of impressions, $S(t)$ can be decomposed into a deterministic, periodic part $P(t)$ and a stochastic part $\eta(t)$:

$$S(t) = P(t) + \eta(t). \tag{1}$$

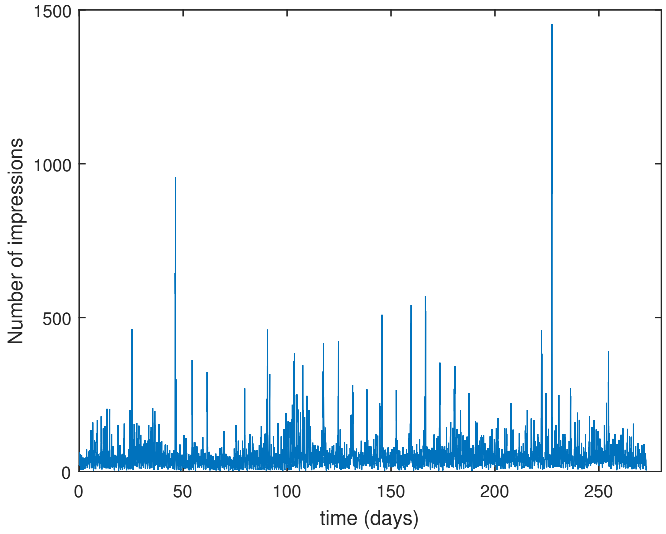

Figure 1: Number of impressions for each hour over the course of 273 days starting from 23rd October 2014 for channel 7287. Representative data is provided by Clypd.

The aim is to filter the signal to remove $\eta(t)$, leaving behind $P(t)$ only. $P(t)$ can be found by performing Fourier transform on the data and taking only the dominant modes in the power spectrum. Missing data was filled in using linear interpolation.

For the filtering scheme, we used Matlab's `fft` and `ifft` algorithms to find the power spectrum and then removed all frequencies with amplitude less than a given threshold A_{thresh}. We write the full signal as

$$S(t) = A_0 + \sum_{j=1}^{N} (A_j e^{ik_j t/T} + A_j^* e^{-ik_j t/T}), \tag{2}$$

where $N = 3275$, and the total duration of the signal is $T = 6551$ hours ≈ 273 days. A_j and A_j^* are complex amplitudes (conjugates of each other), and k_j are dimensionless wavenumbers. Impressions data is given for $t = 0, 1, \ldots, 6550$. Note that there 6551 data points $S(0), \ldots, S(6550)$, 6551 Fourier amplitudes (distinguishing between A_j and A_j^*) and the representation (2) is exact.

In our procedure, we remove all modes j such that $|A_j| \leq A_{\text{thresh}}$:

$$S(t) = \underbrace{A_0 + \sum_{j:|A_j|>A_{\text{thresh}}} (A_j e^{ik_j t/T} + A_j^* e^{-ik_j t/T})}_{\text{filtered signal, } P(t)} + \underbrace{\sum_{j:|A_j|\leq A_{\text{thresh}}} (A_j e^{ik_j t/T} + A_j^* e^{-ik_j t/T})}_{\text{noise, } \eta(t)}. \tag{3}$$

The *filtered* signal is therefore defined as

$$P(t) = A_0 + \sum_{j:|A_j|>A_{\text{thresh}}} (A_j e^{ik_j t/T} + A_j^* e^{-ik_j t/T}), \tag{4}$$

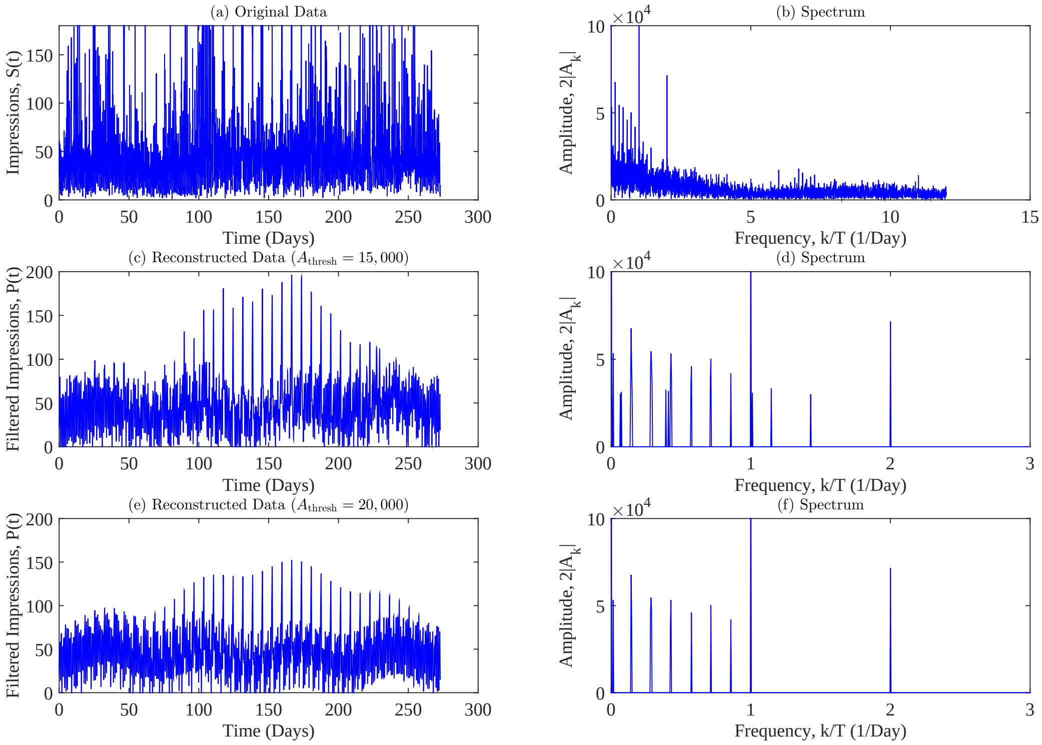

Figure 2: (a,b): Total viewership signal along with power spectrum (frequency has units of day^{-1}). (c,d): filtered signal and power spectrum with $A_{\text{thresh}} = 15,000$. (e,f): filtered signal and power spectrum with $A_{\text{thresh}} = 20,000$.

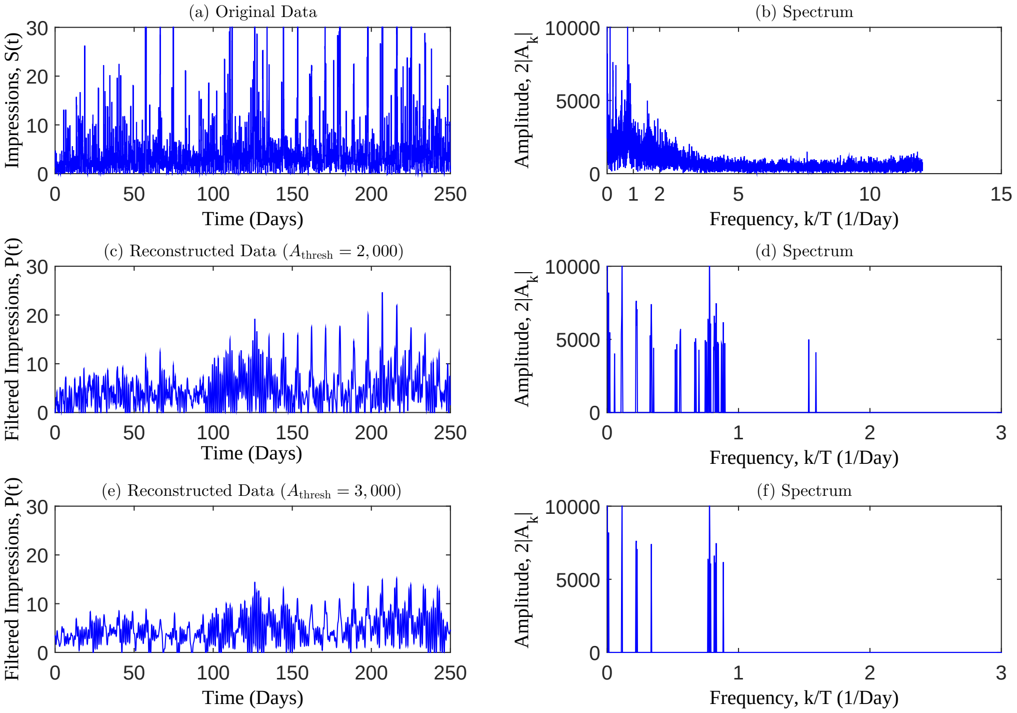

Figure 3: (a,b): Viewership of males 65 and older, along with power spectrum (frequency has units of day^{-1}). (c,d): filtered signal and power spectrum with $A_{\text{thresh}} = 2,000$. (e,f): filtered signal and power spectrum with $A_{\text{thresh}} = 3,000$.

while the *noise* is defined as

$$\eta(t) = \sum_{j:|A_j|\le A_{\text{thresh}}} (A_j e^{ik_jt/T} + A_j^* e^{-ik_jt/T}). \tag{5}$$

Which fourier modes are considered part of the signal, or part of the noise, depends solely on the cut-off A_{thresh}.

Our results are shown in Figures 2 (for the general viewership) and 3 (for males 65 and older). Power spectra are shown for rescaled wave numbers k_j/T which have units of inverse time. In Figure 2, we see that there are "spikes" at $k/T = i/7$ per day for $i = 1, \ldots, 7$. There are also spikes at $k/T = 0$ and $k/T = 2$ per day. Broadly speaking, the predictable portion of the viewership consists of 9 "subpopulations:"

1. A subpopulation that always has their television on, regardless of the time ($k/T = 0$ per day)
2. A subpopulation that watches TV once a week ($k/T = 1/7$ per day)
3. A subpopulation that watches TV twice a week ($k/T = 2/7$ per day)
4. A subpopulation that watches TV three times a week ($k/T = 3/7$ per day)
5. A subpopulation that watches TV four times a week ($k/T = 4/7$ per day)
6. A subpopulation that watches TV five times a week ($k/T = 5/7$ per day)
7. A subpopulation that watches TV six times a week ($k/T = 6/7$ per day)
8. A subpopulation that watches TV once a day ($k/T = 1$ per day)
9. A subpopulation that watches TV twice a day ($k/T = 2$ per day)

These "subpopulations" do not necessarily refer to the same members of the viewership (for example, if a subpopulation of 1000 people always leave their television, 500 people could have switched off their TV and 500 new viewers could have switched on at the same time, keeping the viewership at 1000). The frequencies above also appeared in different demographic groups. However, males and females 65 and older did not exhibit these frequencies (see Fig. 3) suggesting that older members of the population have qualitatively different viewing habits. We should note, however, that upon taking different 7-week subsets of the data, some of the spikes at $k/T = 1/7, 2/7, 3/7, \ldots, 6/7$ no longer appear. That is, only the $k/T = 0$, $k/T = 1$ and $k/T = 2$ per day modes are robust.

In summary, there is a subpopulation of viewers whose behavior is periodic over a week. The weekly pattern is shown as a solid black curve in Figure 4. This figure indicates the median number of impressions over a representative week along with confidence intervals. The distribution of impressions at a given time in the week is found from the first 38 weeks of data. 90% confidence intervals (dashed light blue) are calculated by taking the 5th and 95th percentiles of the distribution. We see that during the evening of each day, there is a rise in the viewership. Saturday seems to be the hardest to predict since it has the highest variability in impressions. The red curve shows the viewership over the 39th week, which mostly falls within the confidence intervals.

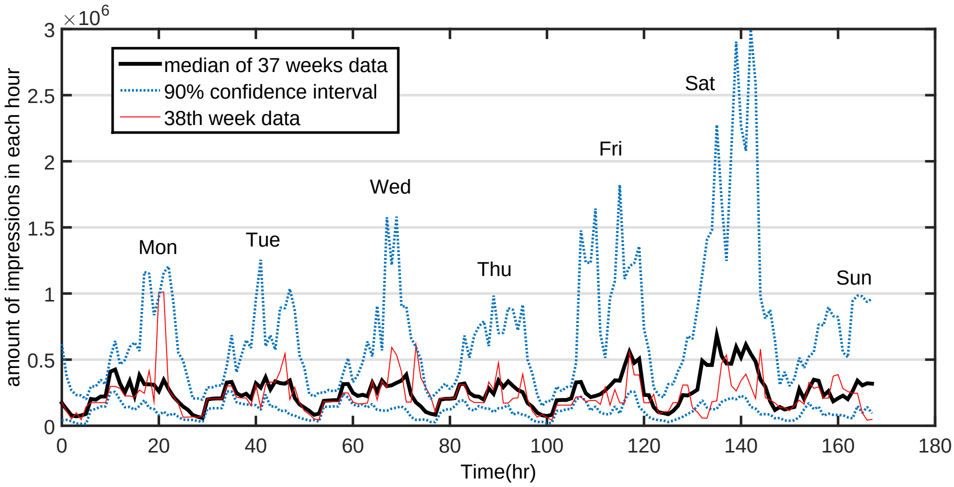

Figure 4: Predicted week-long trend for viewership of channel 7287. The signal exhibits clear periodicity with dominant wave numbers $k/T = \frac{1}{7}, \frac{2}{7}, \ldots, \frac{7}{7}$ per day and $k/T = 2$ per day.

2.2 Analysis of noise

We studied the noise $\eta(t)$, after removing the periodic signal using $A_{\text{thresh}} = 20,000$. We assume that η is a time-homogeneous random variable with a different realization for every t:

$$\text{Prob}[x \leq \eta(t) \leq x + dx] = f(x)dx. \tag{6}$$

Then we tried fitting the data to various well-known probability distributions. The two distributions that fit the data well were the Normal and the t Location-Scale distributions: see Fig 5. For a normal distribution, the data best-fit $N(\mu, \sigma^2)$ with $\mu = -2.7$ and $\sigma = 32$. For the t Location-Scale distribution with pdf

$$f(x) \quad = \quad \frac{\Gamma[(\nu+1)/2]}{\sigma\sqrt{\nu\pi}\Gamma(\nu/2)}\left[\frac{[(x-\mu)/\sigma]^2+\nu}{\nu}\right]^{-\frac{\nu+1}{2}}, \tag{7}$$

we found that the best-fit parameters were $\mu = -6.2$, $\sigma = 20$ and $\nu = 3$. The t-Location-Scale distribution seems to give a better fit than the Normal distribution.

Another important feature of $S(t)$ is the set of large spikes that seem to occur randomly in time. We define the spiking event time as the time when the impressions signal crosses a threshold from below: see Fig. 6. The threshold is defined as the 95th percentile of the total impressions data. In Figure 7, we show that the distribution of waiting times τ between consecutive spikes appears to be approximately exponentially distributed:

$$\text{Prob}[t \leq \tau \leq t + dt] = \lambda \exp(-\lambda t)dt. \tag{8}$$

This suggests that the spiking has no memory (spiking is approximately Markovian) and occurs at a Poisson rate of $\lambda \approx 0.015$ per hour, so that the mean time between spikes is about 69 hours. The analysis of spiking time was performed on unfiltered data $S(t)$. A more thorough analysis would just use the noise left over from the filtering, $\eta(t)$. We don't

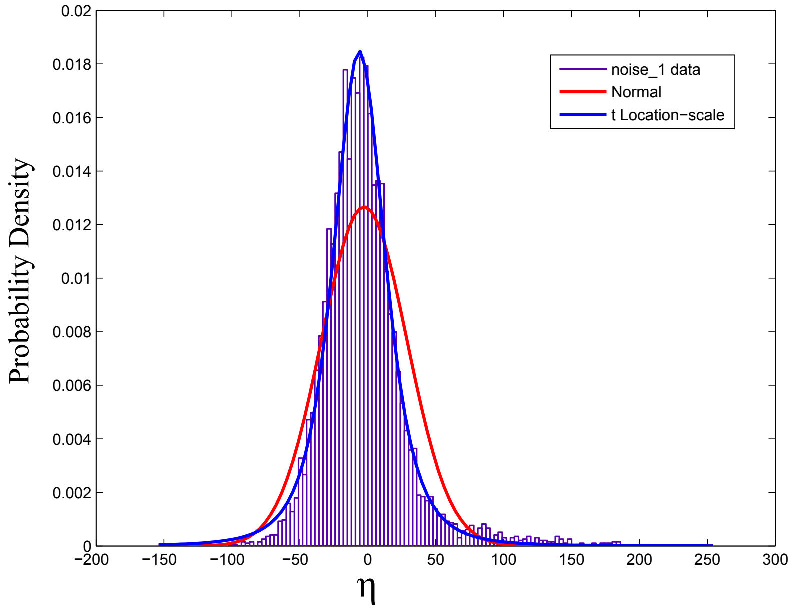

Figure 5: The noise $\eta(t)$ (obtained using the cut-off $A_{\text{thresh}} = 20,000$) is well-fit to both normal and t Location-scale distributions. The mean and standard deviation of the best-fit normal pdf are $\mu = -2.7$ and $\sigma = 32$. The parameters of the best-fit t Location-scale pdf are $(\mu, \sigma, \nu) = (-6.2, 20, 3)$.

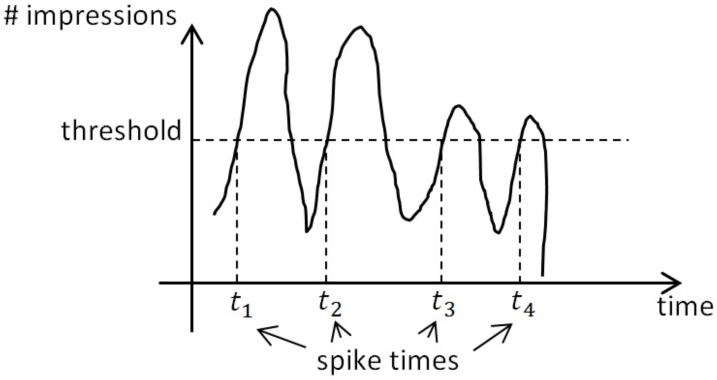

Figure 6: Spike time is defined as the beginning of a series of impressions that exceed a certain threshold. In our analysis, the threshold is set to be the 95th percentile.

anticipate a difference in the outcome however because the periodic signal generally has small amplitude: the top 5% of impressions in $S(t)$ will be very similar to the top 5% in $\eta(t)$.

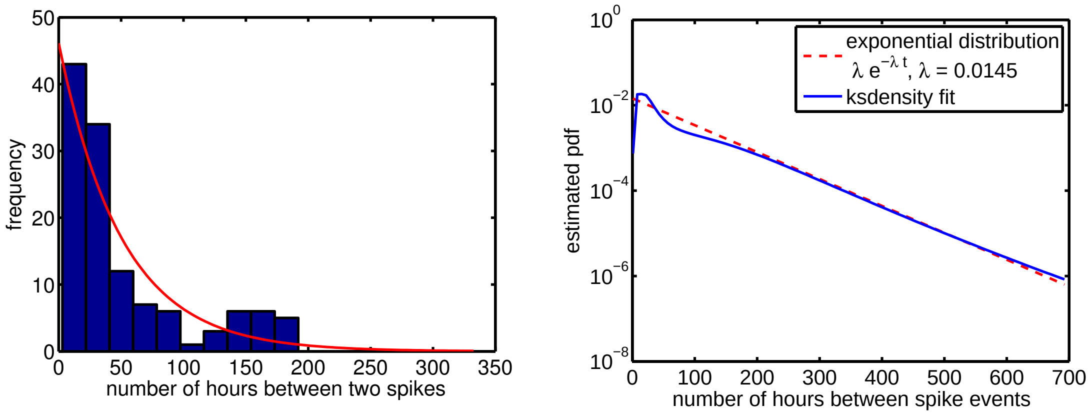

Figure 7: The time between two spikes in viewership approximately follows an exponential distribution. `ksdensity` is a Matlab routine that attempts to find the pdf associated with given data.

2.3 Machine Learning Approach

We also used a machine learning approach to predict the number of impressions in a time slot by learning from data. The machine learning task is defined with the following attributes: program ID (nominal), day of the week (nominal) and time of the day (numeric). The output class is number of impressions.

There are two categories of method used. The first treats the output class as numeric. A number of methods suitable for this task are tested: k-nearest neighbor, neural networks, linear regression, regression tree, and k-star. The best performing algorithm is 1-nearest neighbor. The root relative square error is 61% (100% would correspond to the error in naively guessing the mean of all impressions).

The second category of algorithms requires binning the output class. We binned the output class into 5 bins, and we tested 4 algorithms: decision trees, random forest, naive Bayes, and random tree. Decision tree and random forest performed equally well - the accuracy is 56% (compared with 20% from naively guessing).

A learning curve is shown in Figure 8. The fact that the curve did not plateau shows potential for more accurate prediction given more data. The fact that testing error decreases with training error shows that we didn't over-fit.

2.4 Prediction of Future Impressions by comparing to Previous Programs

Given two programs, p_1 and p_2, at times t_1 and t_2 respectively, we created a function $S(p_1, t_1, p_2, t_2)$ that returns a score relating how similar those programs are at those times

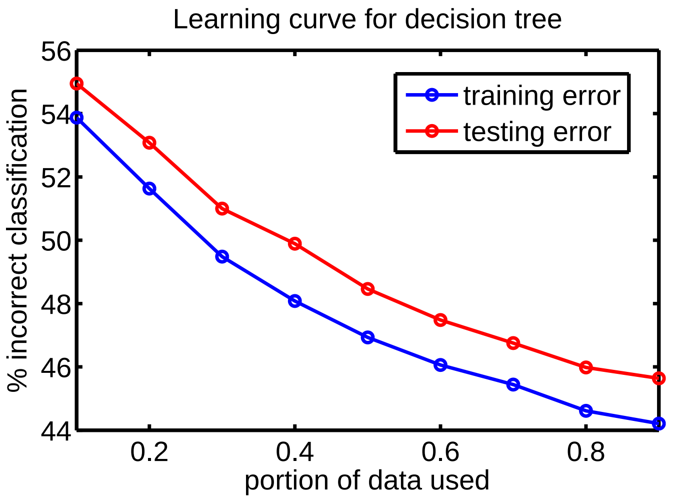

Figure 8: Learning curve for decision tree. The curve did not plateau shows potential for more accurate prediction given more data. The fact that testing error decreases as training error shows that we didn't over-fit.

in terms of their demographic ratios. The idea is that the viewership breakdown of future television shows can be predicted by studying the breakdown of similar shows that have aired in the past.

The score is computed as follows: let $D_{p,t}$ be the demographics information for program p at time t (where t includes information about day of the week and time slot during the day, ex: Monday from 3pm to 4pm).

$$D_{p,t} = (d_1, d_2, ..., d_N)$$

where d_i is the amount of impressions from the i-th demographic (ex: females ages 21 to 24) and N is the number of demographic fields in the data (in our case $N = 30$, with 15 age ranges for both males and females). Now let $I_{p,t} = \sum_{i=1}^{N} d_i$, be the total number of impressions for program p in the time slot t. Let $\hat{D}_{p,t} = \frac{D_{p,t}}{||D_{p,t}||_1} = D_{p,t}/I_{p,t}$ which is $D_{p,t}$ normalized in the ℓ_1 norm. $\hat{D}_{p,t} \in \mathbb{R}^N$ is a breakdown of the viewership by demographic for program p at time t.

Define

$$S(p_1, t_1, p_2, t_2) = ||\hat{D}_{p_1,t_1} - \hat{D}_{p_2,t_2}||_2, \tag{9}$$

which is just the Euclidean distance between $\hat{D}_{p_1,t_1}$ and $\hat{D}_{p_2,t_2}$ in $\mathbb{R}^N$. Programs that are more similar (as measured by impressions made for each demographic group) will be closer to each other in Euclidean distance.

Since we do not know where in the space our data sits, we randomly selected pairs of programs to see how far apart they are on average, keeping certain program attributes (such as program ID, day of showing, hour of showing) identical. Our results are shown in Table 1.

Program ID	Day of showing	Hour of showing	S
Random	Random	Random	0.5128
Random	Same	Same	0.4845
Same	Random	Same	0.3146
Same	Same	Random	0.2842

Table 1: Average distance score for 100,000 trials. The distance S is defined in equation (9).

For future investigation, we would like to measure clustering of these points, experiment with different similarity scores, predict the number of impressions for similar programs, and use Fourier analysis to predict the number of impressions as a function of time. The similarity measure we compute relies on the L^2 norm, however we consider a list of other possible calculations for future investigation. We could compute the difference between $D_{p,t}$ and $D_{p',t'}$ using a dot product

$$\langle \hat{D}_{p,t}, \hat{D}_{p',t'} \rangle$$

resulting in a score between 0 and 1 (after normalizing by a factor of $\frac{2}{\pi}$). We also could investigate impression prediction for a new program by taking the average impressions for some group of similar programs based on the computed score. Lastly, we can determine a Fourier series for each demographic to see how the number of impressions varies in time for each demographic. Given a new program, we can compute a similarity measure and see what similarity neighborhood it falls into. Then we can pick the program that it is closest to and suppose that it should have a similar Fourier series for impressions in each demographic to the program for which it is most similar.

2.5 Prediction of Future Impressions using Kalman Filtering

The goal was to make a prediction for the "value" of future time slots, where the "value" is measured by the number of impressions the time slot will receive.

In order to do the Bayesian estimation, we neglected any sampling issues that may be present from the data (assumption of perfect data). The number of interested viewers are fixed with a particular probability of watching the network or not, which can be modeled by a binomial distribution. Because of the large sample size of viewers, we are able to use a Gaussian distribution to approximate the binomial distribution. If the viewership was not changing, we would be able to represent it as i.i.d random variables. However, we don't want this number to be fixed, so we will just use independent random variables.

For a Bayesian estimation, a prior probability distribution and likelihood function must be assumed in order to formulate an initial prediction. After this initial prediction is made, the new data is observed and is used to update the probability distribution and gives a posterior probability distribution. For our purposes, this posterior distribution was then used as our prior for predicting the next week.

The number of impressions, S, for each week w will be modeled by a Gaussian with mean $\mu(w)$ and standard deviation σ, where we allow μ but, for simplicity, not σ to vary for each week's prediction. Under the Bayesian framework, we view $\mu(w)$ as an unknown parameter which we will represent by a subjective probability distribution. We can think of $\mu(w)$ as a measure of the popularity of the show at week w, while the actual viewership will have an unpredictable fluctuation from week to week due to issues having to do with external factors. The standard deviation σ measures this inherent variability in weekly viewership.

To find a reasonable value for the fixed σ, the available data was divided into a particular amount of bins. For each bin of data, the standard deviation was found, and then all of the standard deviations were averaged together to get an *average* standard deviation. This was done multiple times for various amounts of bins in order to choose the optimal (smallest) standard deviation. The smallest value found was then used as σ for I.

To understand the distribution of μ, a recursive Bayesian estimation was made. To start, a weak prior probability distribution was chosen for μ. After plotting histograms of the historical data, a decent prior distribution seemed to be a Gaussian using the mean and standard deviation of the data (Figure 9). After using the prior and likelihood distributions with Bayes rule and observing the actual data for the following week, the prior distribution is updated to get a posterior distribution. If the likelihood function was also taken to be approximated by a Gaussian, we were able to get a Gaussian posterior probability distribution. To predict farther into the future, the simplest choice for the prior distribution for the next week was to use the posterior distribution from the previous week. The mean of this prior distribution is our approximation for the expected number of impressions for a given network on a given day of the week for a given time.

In order to see how well the implemented model performed, we chose the first 20 weeks of data as the "training" data for the model and tested it against the last 19 weeks. For every day of the week the relative errors were found between the predicted impressions and observed number of impressions for the last 19 weeks. The root mean square (RMS) of the relative errors was calculated as the measure of error for our model. The RMS for each day of the week is found in Table 2 (which were all less than 30%). The table also includes an example of our model's prediction for a week (39^{th} week) for a particular network at 8:00pm.

Day	**Mon**	**Tues**	**Wed**	**Thurs**	**Fri**	**Sat**	**Sun**
Date	**2/2**	**2/3**	**2/4**	**2/5**	**2/6**	**2/7**	**2/8**
Model	83902	101720	65465	50160	65032	101120	63138
Actual	87204	105940	62146	39267	60376	93997	53142
Rel. Error	0.0379	0.0398	0.0534	0.277	0.0771	0.0758	0.188
RMS for Day	0.27	0.0903	0.2203	0.2016	0.0804	0.0883	0.1983

Table 2: Example of impressions estimate for February 2-8, 2015 for a given network at 8:00pm. The model estimates are given along with the observed number of impressions given from the data. The relative errors for the shown week is calculated and compared to the root mean square determined from all of the 39 weeks.

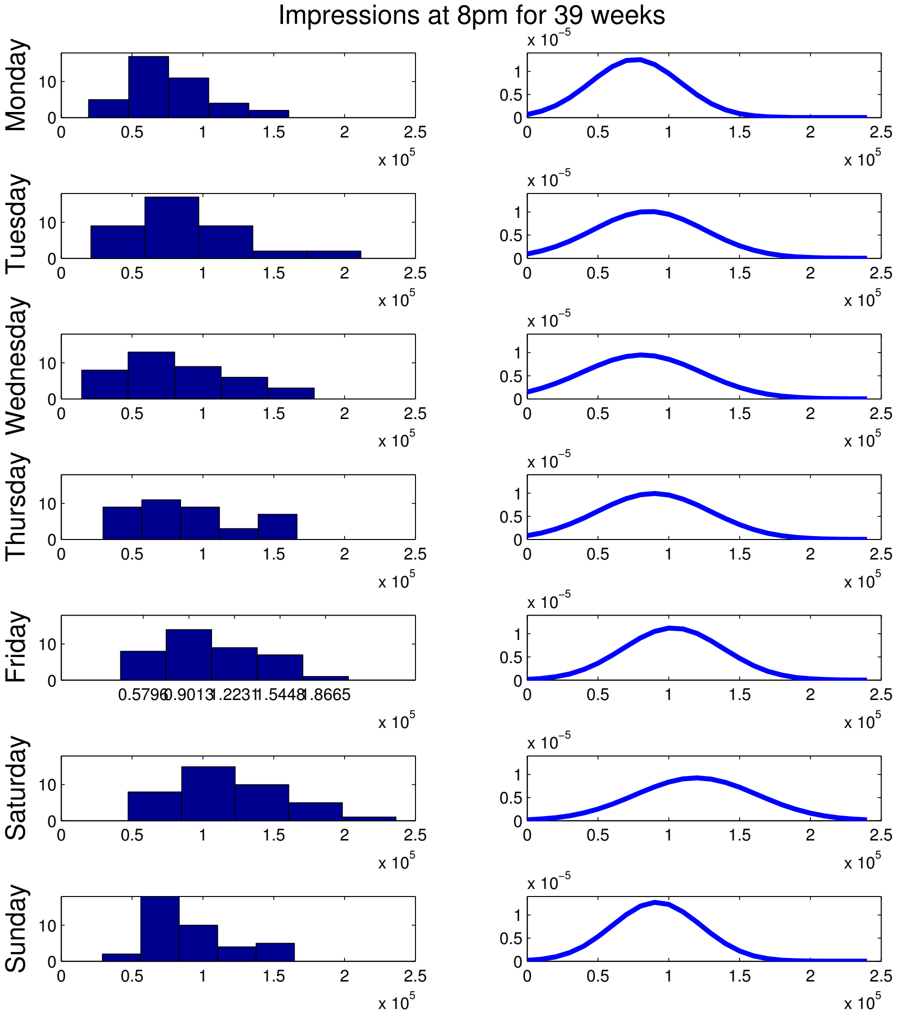

Figure 9: (Left) Histogram of available data for each day of the week. (Right) Approximated normal distribution of available data.

3 Optimal Scheduling using Integer Programming

In this section, we address the important problem of creating an optimal schedule of advertisements, given a set of orders from an advertising agency and predicted viewership numbers (which would come from the analyses in section 2).

3.1 Notation

- C Channels with index c: $1 \leq c \leq C$.
- N_c slots on channel c with index i: $1 \leq i \leq N_c$.
- A advertisers with index a: $1 \leq a \leq A$.
- $S_{c,i}^{(d)}$ number of impressions for slot i, demographic group d, on channel c.
- $\mathbf{V}^{(a)}$ binary vector with target demographics for advertiser a.
- $S_{c,i}^{(a)}$ number of impressions for slot i, in the demographics specified by advertiser a, on channel c: $S_{c,i}^{(a)} = \sum_{d \in \mathbf{V}^{(a)}} S_{c,i}^{(d)}$.
- $P_{c,i}$ price for slot i on channel c (fixed by advertiser).
- $B^{(a)}$ budget of advertising campaign a.
- $R^{(a)}$ desired impressions for campaign a.
- $X_{c,i}^{(a)}$ a 'binary matrix' indicating whether advertiser a is assigned to slot i on channel c. This represents the schedule we are trying to find.

3.2 Current Assumptions

- Fixed time slot duration.
- Programmers specify the number of time slots and the price of each time slot.
- Clypd receives orders all orders for a given week in advance and determine the schedule for one week at a time.
- Order information represents hard inequality constraints for our solution.
- All advertisers have equality priority, but we choose which orders to fill based on a measure of whether the order is likely to be satisfiable.
- Clypd makes a fixed percentage of revenue, so the goal is to maximize revenue.
- All target demographics are of equal value to the advertiser (i.e. the desired number of impressions can be satisfied by any subset of the target audience).

- There is a “good” estimate for the impressions per demographic for each time slot.

We have proposed methods for modifications of the original problem based on implementing some other constraints on the programming schedule (no consecutive ads from the same company, variable duration for ad slots and ad durations, etc.)

3.3 Data Assumptions

In order to try to find an optimal schedule our approach requires that we have accurate values for the number of impressions for each demographic that is watching a given time slot on a given channel. We have made several assumptions when selecting the data to evaluate our approach on.

- Each order is placing restrictions on the programming for a single work week (Monday-Friday) during the hours of 5:00am through 12:00am.
- We use past data from a single work week, 5:00am-12:00am, for three channels to estimate the impressions per channel at each time slot for each demographic.
- We do not take into account any uncertainty in the estimates of impressions per time slot.
- Impressions by demographic are constant over the span of an hour (if there were two half hour shows in an hour we average the impressions per demographic from both shows).

In our implementation we have proposed a way for extending our problem for orders that can be filled over a greater window of time.

3.4 Problem Formulation

3.4.1 Constraints

Budget constraint: Total cost to each advertiser must not exceed their budget $B^{(a)}$.

$$\sum_{c,i} X_{c,i}^{(a)} P_{c,i} \leq B^{(a)} \tag{10}$$

Impressions: The total number of impressions (predicted by $S_{c,i}^{(d)}$) must exceed the campaign goal $R^{(a)}$.

$$\sum_{c,i} X_{c,i}^{(a)} S_{c,i}^{(a)} \geq R^{(a)} \tag{11}$$

Note: This linear constraint only yields a feasible region if it is possible to satisfy every order. In practice, this is unlikely, so instead we solve a sequence of problems where $R^{(a)}$ are replaced with 0 for the orders we are not able to fill. We use a value function to decide which *orders to eliminate.*

No overlap: Only one advertiser can use a given slot on a given channel. Mathematically, this can be stated as

$$\sum_a X_{c,i}^{(a)} \leq 1 \tag{12}$$

This constraint can be modified to allow for variable length commercials by weighting each entry in $X_{c,i}^{(a)}$ by the commercial length for advertiser a and then changing the right hand side to include the number of 'time slots' in each commercial break.

3.4.2 Objective function

Our goal is to maximize the total revenue by the programmers (and consequently the broker).

$$\sum_a \sum_{c,i} X_{c,i}^{(a)} P_{c,i} \tag{13}$$

3.4.3 Implementation

This optimization problem involves finding a vector of binary inputs $\mathbf{X}$ which is a vectorized version of $X_{c,i}^{(a)}$ that satisfies a set of linear inequality constraints and that maximizes the value of a linear objective function. So, this can be solved using a binary integer program.

We achieve this by writing our inequality constraints as a matrix inequality

$$A\mathbf{X} \leq B$$

and our objective function as dot product

$$f(\mathbf{X}) = \mathbf{P} \cdot \mathbf{X},$$

where $\mathbf{P}$ is a vectorized version of $P_{c,i}$. This allows us to make use of MATLAB's built in algorithms from the optimization toolbox. MATLAB's built in mixed integer linear programming algorithm consists of the following three steps

(a) solve the linear programming problem without the integer valued constraints,

(b) use a heuristic algorithm to find a nearby integer feasible solution,

(c) perform branching to try to improve on the heurtistic feasible solution.

Sometimes the heuristic algorithm finds a solution which sells all of the time slots, which is the global maximum for the problem. Other times, we cannot find a feasible solution at all or we cannot fill all of the spots with the given number of orders. Alternatively, it may not be possible to satisfy all of the orders so instead we choose a subset of the orders that includes only the ones that are easiest to satisfy. In the next section we propose a method to rank orders in terms of how easy they are to fill so that we can choose which subset of orders to fill.

3.5 Value function for individual orders

Given that in general it will not be possible to satisfy every order received, we need a systematic way to rank orders in order determine which orders to reject. Below we provide a methods for doing this.

3.5.1 Step 1: Eliminating Unreasonable orders

If the number of impressions desired is more than the size of the viewership for that demographic (in the time alloted), then the order cannot be satisfied. We should reject these orders immediately.

Definition 1: Total size of target demographic

$$T^{(a)} = \sum_{d \in \mathbf{V}^{(a)}} \sum_{c,i} S_{c,i}^{(d)} \tag{14}$$

We will reject an order if $T^{(a)} < R^{(a)}$.

3.5.2 Step 2: Monte Carlo Method

Use a Monte Carlo method to estimate the number of feasible solutions for each advertiser. And then assign a value proportional to that number.

(a) First generate a random binary vector $X_{c,i}^{(a)}$ for fixed a.

(b) Check to see if it is feasible.

(c) Repeat N times.

(d) Count up the fraction of these random vectors that are feasible (satisfy the constraints).

Modification 1: Weight by excess budget. We can improve upon this model, by weighting these feasible solutions by the price the advertiser would be willing to pay (in an auction). This is related to the excess money in their budget. So, given a feasible solution $X_{c,i}^{(a)}$ (from method 1), we can weight that solution fraction of the budget that is unused (this is related to excess amount of money the agency would be willing to increase their offer). For feasible solution j, the weight would then be

$$E_j^{(a)} = \frac{B^{(a)} - \sum_{c,i} X_{c,i}^{(a)} P_{c,i}}{B^{(a)}}. \tag{15}$$

which measures the percentage of the budget that is unused. The average weight of these feasible solutions would give another measure of the value of a particular order.

Note: An advertiser should be willing to increase the bid by a factor of $1/(1 - W^{(a)}) = \frac{B^{(a)}}{\sum_{c,i} X_{c,i}^{(a)} P_{c,i}}$ *to ensure that their advertising campaign order is accepted.*

So, let $\mathbf{F}$ be an N by 1 vector indicating which candidate solutions are feasible. Let $\mathbf{E}^{(a)}$ be the weights from above. Then, we define the value to be

$$\frac{\mathbf{F} \cdot \mathbf{1}}{N}(1 - r) + \frac{\mathbf{F} \cdot \mathbf{E}^{(a)}}{N} r \tag{16}$$

where r represents the relative weight of the excess budget to the feasibility.

Modification 2: Weight by urgency. In order to take into account variable time frames for orders, this value should factor in the fact that certain orders are more urgent than others. Orders that have already been accepted will usually need to be prioritized over new orders. Some orders with short time tables will be infeasible. Other orders with short time tables will need to be completed immediately before the deadline. Orders with long time tables may be saved for later, but if they are postpone for too long, they will become infeasible. This modification was not implemented, but it warrants further exploration.

3.6 Heuristics: Greedy Method

In order to generate a 'quick' solution, we start by implementing a greedy approach. In this approach, we generate a matrix V, where the rows are indexed by the slot $\{i, c\}$ and the columns are index by the advertiser a. We assign a value to each entry of V for each advertiser. Then we choose the slot with the highest value and assign that to the advertiser who gets the most value from that slot. This process is repeated until there are no longer available slots, the orders have all been met or no advertisers can afford a slot. At this point, incomplete orders are removed and the process is repeated with the remaining orders until all orders have been satisfied or removed.

The value of slot $\{i, c\}$ is for advertiser a is equal to

$$V_{c,i}^{(a)} = \left(\frac{S_{c,i}^{(a)}}{R^{(a)}} \right) / \left(\frac{P_{c,i}}{B^{(a)}} \right) \tag{17}$$

which represents the fraction of the desired impressions that can be provided by the slot divided by the fraction of the budget that must be used for the slot.

3.7 Results

In a toy problem consisting of one slot per hour on three different channels for an entire work week, an optimal schedule can be found that satisfies the top 49 sorted orders in approximately 6.5 seconds. The optimal schedule is shown in Figure 10. Adding more time slots by expanding the time horizon to several weeks allows us to accomodate more orders, but it increases the computation time as shown in Figure 11. It appears that this method is suitable for smaller problems with fixed time horizons. Using more channels with varied viewership could greatly increase the number of orders we can accomodate in an optimal schedule. Scaling this problem requires a deeper look at tools for exploiting the structure of the optimization problem to segment into smaller pieces that could be solved in parallel.

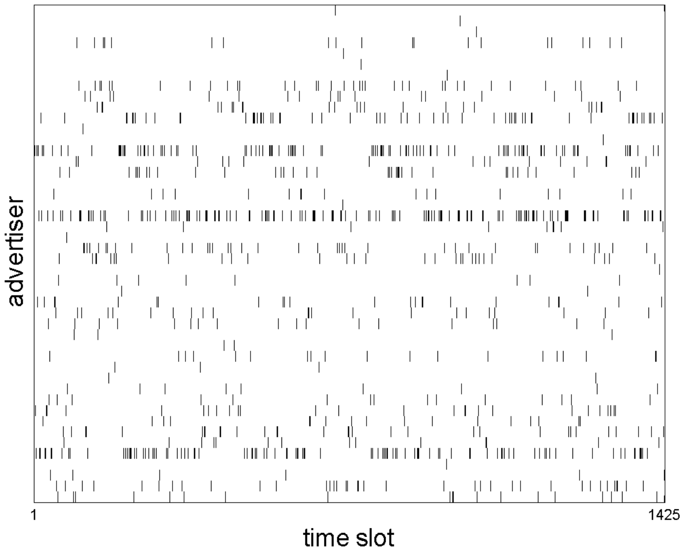

Figure 10: This figure shows the optimal schedule for a toy problem consisting of one slot per hour on three different channels for an entire work week. The vertical axis corresponds to order number while the horizontal axis corresponds to the time slot index. Black corresponds to a slot being filled.

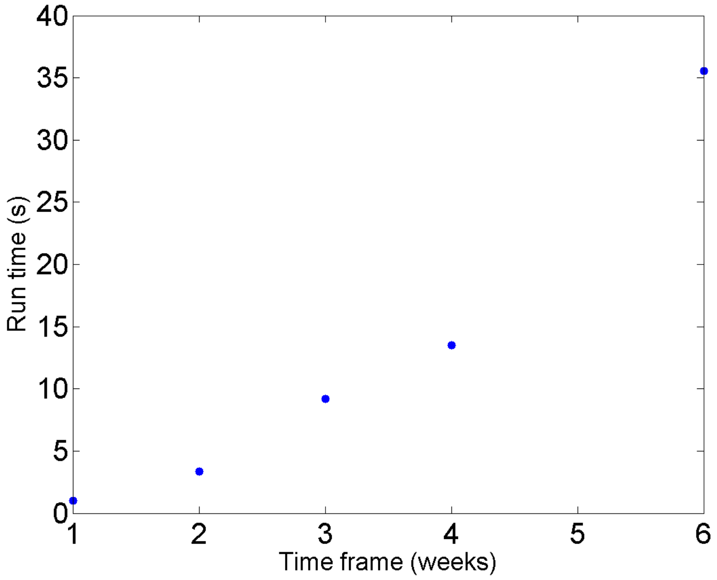

Figure 11: This figure shows how the run-time to finding an optimal schedule scales with the number of weeks for the time horizon of the orders.

3.8 Conclusions

- Determining a ranking for orders based on some value score is important for ensuring the existence of a feasible solution.
- High quality heuristics are necessary for finding feasible solutions to initialize the integer program. The development of new heuristics and improvement of current heuristics could be beneficial for scaling of the problem to more realistic sizes.
- Mixed integer programming provides a flexible framework for implementing the essential constraints while consideration of more complex constraints would require a hybrid approach involving multiple algorithmic frameworks.

4 Reach and Frequency

Advertisers can also specify goals in their bids involving *reach* and *frequency* of their advertisements. We can think of the *reach* of an advertisement as the number of unique individuals who have received at least one impression of the advertisement, and the *frequency* of an advertisement as the mean number of times the advertisement is seen by these individuals who have been exposed at least once to the advertisement. From detailed viewership data such as produced by the Nielson organization, we can express the reach of an advertisement by the following exact formula:

$$R = \sum_{j=1}^{N^{(a)}} S^{\sharp}_{i_j} \tag{18}$$

where $S^{\sharp}_{i_j}$ is number of *new* impressions made at time slot i_j of the jth airing of advertisement, and $N^{(a)}$ is the total number of times the advertisement is shown. This is to be contrasted with the formula for total number of impressions made by the advertising campaign,

$$I = \sum_{j=1}^{N^{(a)}} S_{i_j}$$

where S_{i_j} is the number of impressions (including both new and repeat viewers) made at the time slot i_j of the jth airing of the advertisement. $S^{\sharp}_{i_j}$ by contrast only counts those viewers on whom an impression was made at time slot i_j, but not on a previous airing of that advertisement. Consequently, to count reach, we must know more about the viewership than simply the statistics for the number of impressions likely to be made in each time slot. Put another way, the number of impressions S_i is only a function of the data at time slot i (and thus may be thought of as a one-dimensional marginal distribution of the viewership data), whereas the number of new impressions $S^{\sharp}_i$ depends not only on the statistics of time slot i by also on statistics of previous time slots (and thus inherently involves joint distributions of viewership data at different time slots). To make matters more complicated for the schedule optimization, the number of impressions S_i in a time slot is independent of anything other than the time slot in which the ad is scheduled, whereas the number of new impressions

$S_i^\sharp$ depends not only on the time slot i but also the previously scheduled time slots of the advertisement. The frequency fortunately is easily determined from the reach by the simple formula:

$$F = S/R.$$

In these discussions, we are imagining that the advertising campaign will take place on a specific channel, with programs tied in a 1-1 way with the time slots, and a fixed demographic group under consideration. Everything can be generalized in principle to allow multiple channels, varying programs in a given time slot, and multiple demographic groups, with accompanying complication in notation.

4.1 Predictive Scheme for Reach

As noted above, with historical viewership data, the reach of any hypothetical or actual past advertising schedule is easily calculated by a standard data processing algorithm. Predicting the reach of a proposed future advertising campaign, however, is a much more delicate matter. Consider for example the simplest situation in which we can assume future weeks will have viewership identical to past weeks, or that we otherwise have an oracle that will unfailingly predict the Nielson data for the future time frame over which the advertising campaign will take place. Then, while we can calculate the reach for any proposed schedule, this will be an expensive and unwieldy calculation. Either we must do an online processing of historical data, or we must refer to an intractably huge data structure which has precomputed reach scores for every feasible advertising schedule. Including reach into the optimization scheme for the advertising schedule will, as we show, introduce an inherent nonlinearity, and nonlinear optimization almost inevitably involves a large degree of iteration. That means many schedules will be proposed in the optimization, and thus the expensive reach computation will be invoked many times. We therefore consider a simplified way to estimate future reach. Such simplification is justified in particular because future viewership cannot be perfectly predicted, so involving a expensive and precise computation for reach in the schedule optimization algorithm would seem to be a misplaced effort.

We propose encoding the information needed for future reach calculations in a two-slot function $P_{i,i'}$ which, for $i < i'$, represents an estimate of the fraction of viewers at time slot i' who also viewed time slot i. Under our standing simpilfying assumptions, $P_{i,i'}$ could be estimated from historical data, possibly using the Kalman filtering idea in Section 2.5. If we allow future time slots to be associated with programs different from those in the past, we could try to develop an inference scheme for combining historical data on viewership of time slots with viewership of programs. But this might still be attempting too fine a resolution. Since the number of potential time slots in which a proposed advertisement could air within a typical campaign window is large, such a detailed data-driven approach would require the storage of P as an immense matrix, which would be at least $10^3 \times 10^3$ for a weeklong campaign even in our very simplified setting, and much larger in practice. A more tractable approach might be to simply treat $P_{i,i'}$ as a function of only $i - i'$, meaning essentially the time difference between slots (and possibly also a measure of difference between channels for a multichannel campaign). We might imagine that $P_{i,i\prime}$ begins as a decaying function of $|i - i'|$, but has peaks at multiples of a day and a week for patterned viewer behavior. Perhaps historical data could be fit to a sum of a small number of periodic functions with frequencies identified by the spectral analysis in Section 2.1, with decaying amplitudes.

We now assume we have in hand some scheme for estimating the two-slot function $P_{i,i'}$, and now wish to estimate the reach of a proposed scheduling of the advertising campaign in time slots $\{i_1, i_2, \ldots, i_{N^{(a)}}\}$. According to the formula (18), we need to estimate the number of new impressions made with each airing of the advertisement, and we propose to approximate this in terms of the estimated number of impressions and the two-slot function as follows:

$$S^{\sharp}_{i_j} \approx S_{i_j} \prod_{j'<j} (1 - P_{i_{j'},i_j}). \tag{19}$$

That is, the estimated number of new impressions is equal to the estimated number of impressions, discounted by factors $(1 - P_{i_{j'},i_j})$ representing the fraction of the viewers of the jth airing of the ad who did *not* also see the prior j'th airing of the ad. The approximation in Eq. (19) is conditional independence of the viewership of all previous airings of the advertisement by the viewers of the jth airing of the advertisement. For a concrete example, for $j = 3$, the *conditional* independence assumption is that whether viewers of the third airing of the advertisement watched the first airing of the advertisement is independent of whether they watched the second airing. Note that this is not the same as stating that a general viewer has an independent chance of viewing the first and second airing of the advertisement (unconditional independence). Indeed, if P_{i_1,i_3} and P_{i_2,i_3} were 0.95, then the probability model underlying the formula (19) would have a substantial positive correlation between the viewers of the first and second airing of the advertisement. The point of the conditional independence assumption is that we assume all such correlations between the viewership of the various airings of the advertisement can be well represented by an explicit model of the correlation between the viewer of each airing $j' < j$ and the airing j under consideration, with the correlations between the previous airings being implied (not neglected!) by the conditional independence assumption.

The conditional independence assumption can lead to either overestimates or underestimates of the reach. For example, if the airings occur during successive episodes of a program with a substantial committed core base who watches every episode, the number of new impressions would be underestimated by formula (19). On the other hand, if the airings of the advertisement involve some episodes repeated at different times during a week, the viewership of those airings would be more negatively correlated than the conditional independence assumption, and the number of new impressions could be overestimated by formula (19). This can be verified under a simple model in which no viewer makes repeated viewings of the same episode at different times. Some kind of conditional independence assumption seems to be necessary to reduce the reach calculation to a complexity comparable to the 2-slot statistic. Another natural way to invoke conditional independence is via a Markov chain model, which would only attempt to explicitly model the repetition in viewership between successive airings of the advertisement. Such a Markovian approach appears less suitable than the conditional independence we suggest in the previous paragraph for a couple of reasons. First of all, it seems rather unclear how to deduce the number of impressions made on the third airing of an advertisement by knowing how many viewers of the first advertisement saw the second advertisement, and how many viewers of the second advertisement saw the third advertisement. How does one infer from this the number of viewers of the third advertisement who saw neither the first nor the second airing? Moreover, the Markovian approach seems completely incapable of representing the likely strong repeat viewership of a regularly airing program from one week to the next, if advertisements are also aired in between those weekly episodes. So if the first and third airing of the advertisement took place one week apart in successive episodes, and a second airing took place in between, one would expect a large number of repeat viewers between the first and third airing, but not between the second airing and either the first or third airing.

4.2 Incorporation of Reach into Schedule Optimization

The approximate reach estimate developed in Subsection 4.1 can be expressed as a polynomial function of the schedule vector $X_{c,i}^{(a)}$:

$$R(X_{c,i}^{(a)}) = \sum_{i=1}^{N_c} X_{c,i}^{(a)} S_i \prod_{i'<i} (1 - P_{i',i} X_{c,i'}^{(a)}),$$

where N_c is the number of slots available on the channel under consideration. Constraints involving reach (or frequency) would become smooth nonlinear constraints, and after relaxation from the integer constraint, could be approached by the alternating direction method of multipliers.

4.3 Uncertainty Estimation for Reach and Frequency

For the purpose of building in safety margins in a schedule to avoid disappointing an important advertising client, we might be interested in characterizing the risk that a particular advertising campaign might miss the targets set by an advertiser's bid. The simplest characterization of uncertainty would be a standard deviation. The number of impressions I_i and the 2-slot characterization of repeat viewership, $P_{i,i'}$ are supposed to be directly estimated from historical data by one of the methods described in Section 2, and we imagine that those methods can also produce uncertainty estimates. (Kalman filtering does this automatically.) The reach and frequency are somewhat complicated function of these variables, so we describe one simple way we might translate the uncertainty estimates of these variables to an uncertainty estimate for reach and frequency.

We will begin by assuming the uncertainty in the estimates of $\{S_i\}_{i=1}^{N_c}$ and $\{P_{i,i'}\}_{1\leq i<i'\leq N_c}$ are all independent, and indicate the mean of a random variable Y as $\bar{Y}$ and its standard deviation as $\sigma(Y)$ (so variance is σ_Y^2). Because the variance of a sum of independent random variables is the sum of the variances of each term, we can therefore express the variance of the reach as a sum of the variances of the new impressions:

$$\sigma^2(R) = \sum_{j=1}^{N^{(a)}} \sigma^2(S_{i_j}^{\sharp}).$$

The independence assumption does allow the variance of the new impressions, $\sigma^2(S^\sharp_{i_a})$ to be worked out in a closed form expression in terms of the mean and standard deviations of $\{S_i\}_{i=1}^{N_c}$ and $\{P_{i,i'}\}_{1\leq i<i'\leq N_c}$, but this expression is very messy. We satisfy ourselves here with simply reporting an expression valid when the standard deviations of all the constituent random variables are small compared to their means:

$$\begin{aligned}\sigma(S_i) &\ll \bar{S}_i, \qquad 1 \leq i \leq N_c,\\ \sigma(P_{i,i'}) &\ll \bar{P}_{i,i'}, \qquad 1 \leq i < i' \leq N_c.\end{aligned}$$

Then one can conduct a small noise expansion by writing every random variable in the form $Y = \bar{Y} + \tilde{Y}$, taking a Taylor expansion up to first order in the fluctuations $\tilde{S}_i$ and $\tilde{P}_{i,i'}$, and then computing the variance. We can thereby obtain:

$$\sigma(R)^2 \approx \sum_{j=1}^{N^{(a)}} \sigma^2(S_{i_a}) \prod_{j'<j}(1-\bar{P}_{i_{j'},i_j})^2 + \sum_{j=1}^{N^{(a)}}\sum_{j''=1}^{j-1} \bar{S}_{i_j}^2 \sigma^2(P_{i_{j'},i_j}) \prod_{j'<j,j'\neq j''} (1-\bar{P}_{i_{j'},i_j})^2$$

Actually this small noise expansion can be readily generalized to allow correlations between the random variable models for $\{S_i\}_{i=1}^N$ and $\{P_{i,i'}\}_{1\leq i<i'\leq N}$; the same strategy would produce further sums involving the covariances between all pairs of these variables.

Applying the same small noise approximation to the frequency, we obtain an estimate for its standard deviation:

$$\sigma^2(F) \approx \frac{\sigma^2(S)}{\bar{R}^2} + \frac{\sigma^2(R)\bar{S}^2}{\bar{R}^4}$$

where

$$\sigma^2(S) = \sum_{j=1}^{N^{(a)}} \sigma^2(S_{i_j}).$$

5 Conclusions

In this report, we analyzed the problem of optimally scheduling advertisements using several different methods. First, we analyzed historical data to obtain trends in the viewership. We found that the viewership was strongly periodic and that deviations from the periodic signal (noise) were approximately bell-shaped. We supplemented these analyses with predictions from several machine learning algorithms for viewership, based on program attributes, and a procedure for predicting new program impressions from only the program's target demographic. Second, we developed an algorithm, based on binary integer programming, to schedule advertisements. Given orders in the form of a budget, number of impressions desired and demographic targets, the algorithm produces a binary matrix that tells the media company how to schedule advertisements in such a way as to maximize revenue. The algorithm should be initialized with a schedule generated by a greedy heuristic. Finally, we developed a theoretical framework to quickly estimate the reach (number of new impressions made) of an advertisement. This framework approximates the number of new viewers through historical impressions data and computing/estimating a *two-slot function*, which gives the fraction of viewers who watched the same advertisement in two time slots.

In summary, mathematical analysis can be an extremely useful tool for understanding how to best-schedule advertisements. Techniques from probability, statistics, data science, signal analysis and linear/non-linear programming can all be used to improve and optimize advertising campaigns, give insight into viewership trends and predict the reach of future television programs.