Aiding Television Media Planning Through Bayesian Inference and Forecasting

Matthew Tiger

Towson University

May 2018

Problem

TV Advertising Buying and Selling

Motivating Example

Baseball example

Formal Statement of Problem

Bayesian Inference

Baseball example

Types of Data

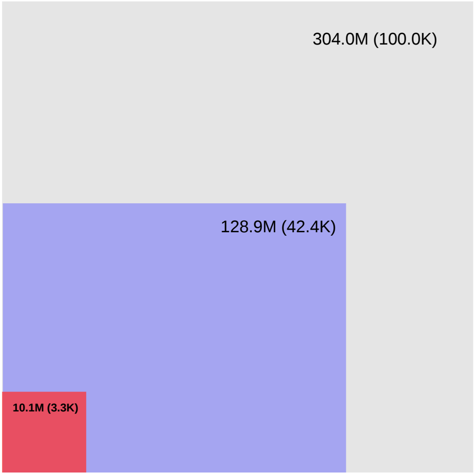
Programming Schedule

Forecasted Impressions

Audience Measurement

Training Data

Audience Size Comparison



Units of Observation and Analysis

Covariates

- *Time-based* covariates and **Program-based** covariates

Covariates

- *Time-based* covariates and **Program-based** covariates
- Derived from Media Schedule
    - *Broadcast Month*
    - *Day of Week*
    - *Stratified Hour*
    - **Content**
    - **Lead-in Content**

Covariates

- *Time-based* covariates and **Program-based** covariates
- Derived from Media Schedule
  - *Broadcast Month*
  - *Day of Week*
  - *Stratified Hour*
  - **Content**
  - **Lead-in Content**
- Derived from Audience Measurement Data
  - **Genre**
  - **Live-program**
  - **First-run**

Assumptions

- We assume that the response variables y_i are *exchangeable* given the parameters of the model and the covariates of the unit of observation.

Assumptions

- We assume that the response variables y_i are *exchangeable* given the parameters of the model and the covariates of the unit of observation.
- A sequence of random variable is exchangeable if the "joint probability density $p(y_1, \ldots, y_k)$ is invariant to permutations of the indexes."

Assumptions

- We assume that the response variables y_i are *exchangeable* given the parameters of the model and the covariates of the unit of observation.
- A sequence of random variable is exchangeable if the "joint probability density $p(y_1, \ldots, y_k)$ is invariant to permutations of the indexes."
- This allows us to model the data as independently and identically distributed given the covariates and unknown parameters.

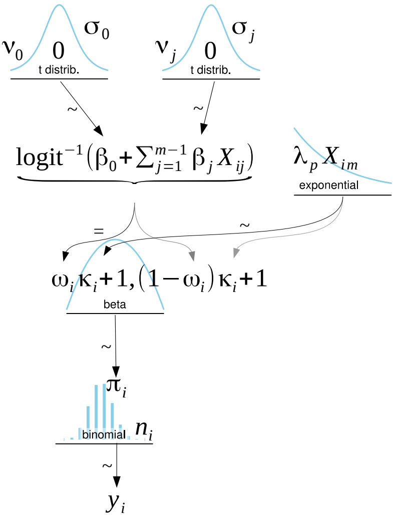
Model Description

Define model $\mathcal{M}$ to be

$$
\begin{aligned}
y_i | X_i, n_i, \pi_i, \omega_i, \kappa_i &\sim \text{Bin}(n_i, \pi_i) \\
\pi_i | \omega_i, \kappa_i &\sim \text{Beta}\left(\omega_i \kappa_i + 1, (1 - \omega_i)\kappa_i + 1\right) \\
\omega_i &= \text{logit}^{-1}\left(\beta_0 + \sum_{j=1}^{m-1} \beta_j X_{ij}\right), \quad \beta_j \sim t_4(0, \sigma_j^2) \\
&\text{for } 0 \leq j \leq m \\
\kappa_i | X_{im} &\sim \text{Exp}(\lambda_p X_{im}), \quad \text{for } p = 0, 1,
\end{aligned}
$$

where $\text{logit}^{-1}(\alpha) = \frac{\exp \alpha}{1 + \exp \alpha}$.

Model Description



Prior Distribution Choice

- Prior distributions for coefficients β_j and concentration parameter κ_i are chosen to be *weakly informatative*.

Prior Distribution Choice

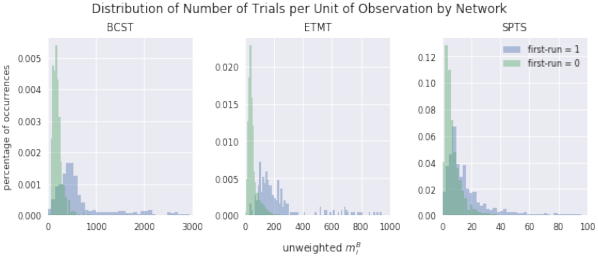
- Prior distributions for coefficients β_j and concentration parameter κ_i are chosen to be *weakly informatative*.
- For the coefficients, this means that $\mu_j = 0, \nu_j = 4$ for all j and that $\sigma_j = 2.5$ if $1 \leq j \leq m$ otherwise $\sigma_0 = 5$.

Prior Distribution Choice

- Prior distributions for coefficients β_j and concentration parameter κ_i are chosen to be *weakly informatative*.
- For the coefficients, this means that $\mu_j = 0, \nu_j = 4$ for all j and that $\sigma_j = 2.5$ if $1 \leq j \leq m$ otherwise $\sigma_0 = 5$.
- For the concentration parameter, this means that $\lambda_p = 10^{-4}$.

Prior Distribution Choice

- Prior distributions for coefficients β_j and concentration parameter κ_i are chosen to be *weakly informatative*.
- For the coefficients, this means that $\mu_j = 0, \nu_j = 4$ for all j and that $\sigma_j = 2.5$ if $1 \le j \le m$ otherwise $\sigma_0 = 5$.
- For the concentration parameter, this means that $\lambda_p = 10^{-4}$.



Computation

- Inference was computed using pymc3, a probabilistic programming language and library for Python.

Computation

- Inference was computed using pymc3, a probabilistic programming language and library for Python.
- The library is powered by the No U-Turn Sampler (NUTS) which is a variant of Hamiltonian Monte Carlo (HMC).

Computation

- Inference was computed using pymc3, a probabilistic programming language and library for Python.
- The library is powered by the No U-Turn Sampler (NUTS) which is a variant of Hamiltonian Monte Carlo (HMC).
- Parameters used for sampling:
  - `target_accept`: 0.95
  - tuned samples: 3000
  - drawn samples: 500
  - number of chains: 4

Convergence

- Approximate convergence to posterior distribution is measured through the *Gelman-Rubin* statistic, denoted by $\hat{R}$.

Convergence

- Approximate convergence to posterior distribution is measured through the *Gelman-Rubin* statistic, denoted by $\hat{R}$.
- Another convergence check is the number of effective samples produced by the simulation, denoted by $\hat{n_{\text{eff}}}$.

Convergence

- Approximate convergence to posterior distribution is measured through the *Gelman-Rubin* statistic, denoted by $\hat{R}$.
- Another convergence check is the number of effective samples produced by the simulation, denoted by $\hat{n_{\text{eff}}}$.
- If $\hat{R}$ is close to 1 then we may assume we have approximate convergence. Further it is recommended that $\hat{n_{\text{eff}}} \geq 10M$ where M is the number of sampled Markov chains for all model parameters.

Convergence

- Approximate convergence to posterior distribution is measured through the *Gelman-Rubin* statistic, denoted by $\hat{R}$.
- Another convergence check is the number of effective samples produced by the simulation, denoted by $\hat{n_{\text{eff}}}$.
- If $\hat{R}$ is close to 1 then we may assume we have approximate convergence. Further it is recommended that $\hat{n_{\text{eff}}} \geq 10M$ where M is the number of sampled Markov chains for all model parameters.
- For each network model, we have that $0.99 \leq \hat{R} \leq 1.01$ and $\hat{n_{\text{eff}}} > 400$ for all model parameters.

Posterior Predictive Checks

- "If the model fits, then replicated data under the model should look similar to observed data."

Posterior Predictive Checks

- "If the model fits, then replicated data under the model should look similar to observed data."
- Generating data using the posterior density and checking some aspect of the generated data set is called a *posterior predictive check*.

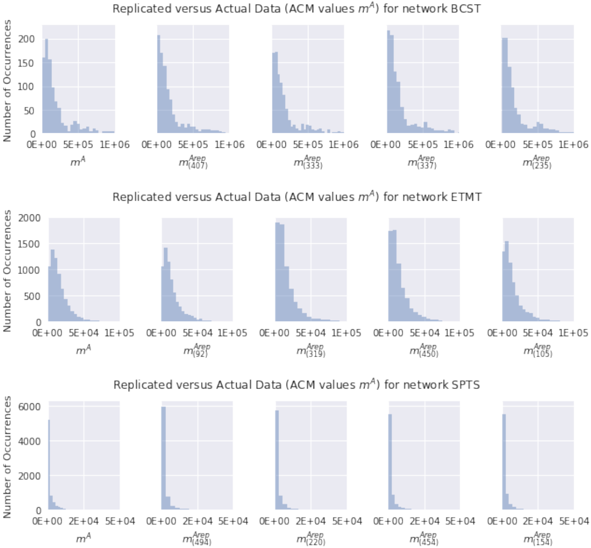
Posterior Predictive Checks

- "If the model fits, then replicated data under the model should look similar to observed data."
- Generating data using the posterior density and checking some aspect of the generated data set is called a *posterior predictive check*.

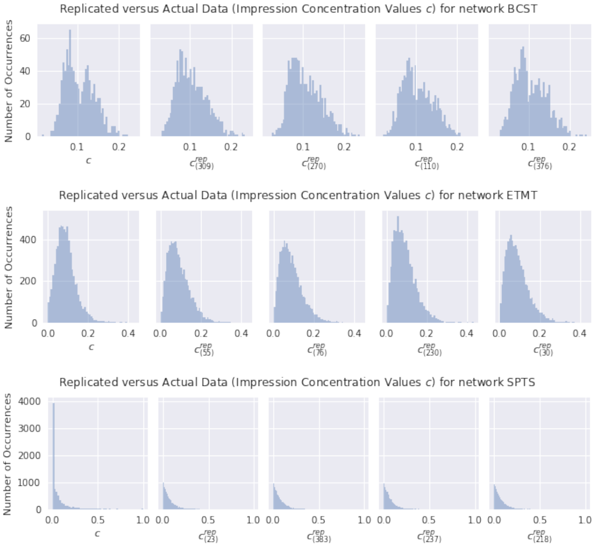
Let y be the observed data and θ be the vector of model parameters. Define y^{rep} to be the replicated data that could have been generated given θ, i.e.

$$p(y^{\text{rep}}|y) = \int p(y^{\text{rep}}|\theta)p(\theta|y)d\theta. \quad (1)$$

Replicated versus Actual Data



Replicated versus Actual Data



Test Statistics

- We can quantify model discrepancies by defining a test quantity $T(y, \theta)$ and then measuring the discrepancy between the observed data and the replicated data.

Test Statistics

- We can quantify model discrepancies by defining a test quantity $T(y, \theta)$ and then measuring the discrepancy between the observed data and the replicated data.
- Formally, we can compute a posterior predictive p-value defined as

$$p_B = \Pr\left(T(y^{\text{rep}}, \theta) \geq T(y, \theta) | y\right).$$

Test Statistics

- We can quantify model discrepancies by defining a test quantity $T(y, \theta)$ and then measuring the discrepancy between the observed data and the replicated data.
- Formally, we can compute a posterior predictive p-value defined as

$$p_B = \Pr\left(T(y^{\text{rep}}, \theta) \geq T(y, \theta) | y\right).$$

- Since we use simulated values of the posterior density, we have that the estimated p-value for S simulations is given by:

$$\hat{p_B} = \frac{1}{S} \sum_{i=1}^{S} [T(y_{(i)}^{\text{rep}}, \theta_{(i)}) \geq T(y, \theta_{(i)})]. \tag{2}$$

Test Statistics - Definition

We define the following test quantities to use in evaluating the fit of model $\mathcal{M}$:

- $T_1(y, \theta) := \min(y)$,

Test Statistics - Definition

We define the following test quantities to use in evaluating the fit of model $\mathcal{M}$:

- $T_1(y, \theta) := \min(y)$,
- $T_2(y, \theta) := \overline{y} = \frac{1}{N} \sum_{i=1}^{N} y_i$,

Test Statistics - Definition

We define the following test quantities to use in evaluating the fit of model $\mathcal{M}$:

- $T_1(y, \theta) := \min(y)$,
- $T_2(y, \theta) := \overline{y} = \frac{1}{N} \sum_{i=1}^{N} y_i$,
- $T_3(y, \theta) := \max(y)$,

Test Statistics - Definition

We define the following test quantities to use in evaluating the fit of model $\mathcal{M}$:

- $T_1(y, \theta) := \min(y)$,
- $T_2(y, \theta) := \overline{y} = \frac{1}{N} \sum_{i=1}^{N} y_i$,
- $T_3(y, \theta) := \max(y)$,
- $T_4(y, \theta) := \text{std}(y) = \sqrt{\frac{\sum_{i=1}^{N} (y_i - \overline{y})^2}{N-1}}$.

Test Statistics - Evaluation - BCST network

Test quantity	$T(y,\theta)$	95% int. for $T(y^{\text{rep}},\theta)$	p_B
$T_1(y,\theta)$ (min)	3701	[6245, 14270]	0.99
$T_2(y,\theta)$ (mean)	227457.84	[2266852.49, 236367.09]	0.95
$T_3(y,\theta)$ (max)	4311038	[3443885, 4989241]	0.34
$T_4(y,\theta)$ (std)	334052.86	[325128.37, 364859.10]	0.90

Test Statistics - Evaluation - ETMT network

Test quantity	$T(y, \theta)$	95% int. for $T(y^{\text{rep}}, \theta)$	p_B
$T_1(y, \theta)$ (min)	0	[9, 182]	1.0
$T_2(y, \theta)$ (mean)	16357.80	[16705.39, 17489.11]	1.0
$T_3(y, \theta)$ (max)	452762	[307901, 760822]	0.78
$T_4(y, \theta)$ (std)	17686.89	[20021.24, 23205.09]	1.0

Test Statistics - Evaluation - SPTS network

Test quantity	$T(y,\theta)$	95% int. for $T(y^{\text{rep}},\theta)$	p_B
$T_1(y,\theta)$ (min)	0	[0, 0]	1.0
$T_2(y,\theta)$ (mean)	3972.45	[3714.91, 4559.66]	0.73
$T_3(y,\theta)$ (max)	526816	[607186, 2239365]	0.99
$T_4(y,\theta)$ (std)	22300.18	[20012.59, 39808.44]	0.88

Residual Analysis

- For a model with unknown parameters θ and predictors x_i, the *predicted* value is $\mathrm{E}(y_i|x_i, \theta)$ and the *residual* is $r_i = y_i - \mathrm{E}(y_i|x_i, \theta)$.

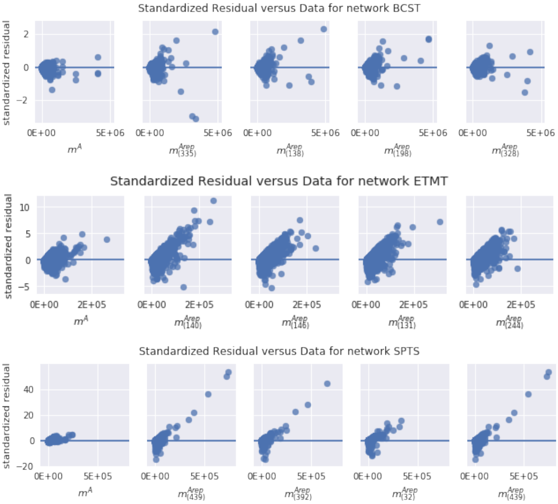
Residual Analysis

- For a model with unknown parameters θ and predictors x_i, the *predicted* value is $\mathrm{E}(y_i|x_i,\theta)$ and the *residual* is $r_i = y_i - \mathrm{E}(y_i|x_i,\theta)$.
- The *standardized residual* is given by $r_i/\mathrm{std}(y)$.

Residual Analysis

- For a model with unknown parameters θ and predictors x_i, the *predicted* value is $\mathrm{E}(y_i|x_i, \theta)$ and the *residual* is $r_i = y_i - \mathrm{E}(y_i|x_i, \theta)$.
- The *standardized residual* is given by $r_i/\mathrm{std}(y)$.
- Using the simulated posterior density, we can compute $\mathrm{E}(y_i|x_i, \theta)$ to be the mean of the replicated hold-out data itself.

Residual Analysis - Actual versus Replicated



Residual Analysis - Test Statistic Evaluation

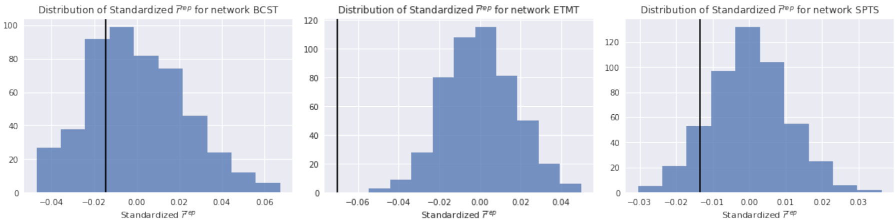
We can measure the residual mis-fit through the following test statistic:

$$T(y, \theta, x) = \frac{\bar{r}}{\text{std}(y)}.$$

Residual Analysis - Test Statistic Evaluation

We can measure the residual mis-fit through the following test statistic:

$$T(y, \theta, x) = \frac{\bar{r}}{\text{std}(y)}.$$



Units of Observation

Quantiled Media Plans

Sample frame title