Time Series Forecasting for Conversion Prediction in Social Advertising

Davis Sorenson
February 22, 2023
Aalto University School of Science

Supervisor: Prof. Aki Vehtari, Aalto University
Advisor: PhD (Tech.) Hannu Laine, Smartly.io Solutions Oy

Download these slides: davissorenson.com/msci_presentation.pdf

Table of contents

Introduction

Introduction

- Advertising is a multi-hundred billion-dollar industry ($723 billion in 2021) [1]
- Social media advertising is a large portion of it. In the same year, Meta Inc. and Snap Inc. reported ad revenues of $115 billion and $4.1 billion respectively [2] [3]
- Advertisers expect a return on investment for all of this ad spend, in particular **conversions** (ad viewers becoming customers)
- Difficult for advertisers to know what will be an effective ad

Introduction

- Digital ads make it easy to try a lot of different strategies to see what works
- Ad budgets are then adjusted in proportion to ad effectiveness
- Making adjustments to budgets of many ads is a challenge:
    - Correctness: Which ads are actually more effective?
    - Practicality: Manually changing budgets takes a lot of time
- High opportunity cost to not adjusting budgets

Introduction

- A solution: Predictive Budget Allocation (PBA), described by Ahonen (2017) [4]
- Automatically divides up a budget between different ads based on performance
- Attempts to optimize for the lowest possible **cost per conversion**
- Changes to budgets only affect the future, so PBA relies on forecasted conversions
- In this thesis, we investigate different time series forecasting methods for predicting conversions

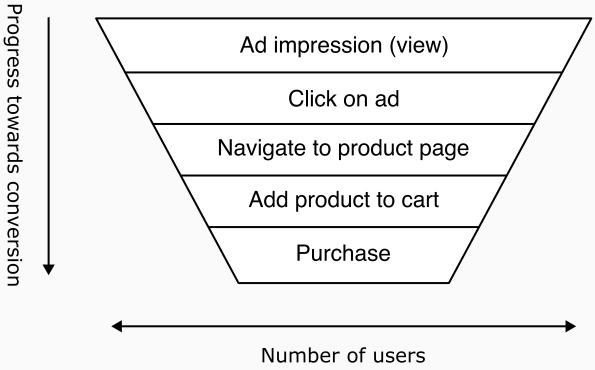
Background: Social advertising, predictive budget allocation, and conversion prediction

How ads are purchased

- Advertisers specify **daily budgets** for ads, which the ad platform spends by the end of the day
- When the ad platform finds a suitable user for the ad (according to how it is targeted), the ad platform holds an **ad auction**, where ads **bid according to their budget and bidding strategy**
- The winning ad has its bid deducted from its budget, and is then shown to a user
- Note: Modern ad auctions are won by more criteria than just the highest bid

Conversions and the conversion funnel

- A **conversion** is when a user who saw an ad becomes a paying customer
- Upon seeing an ad, a user enters the top of the so-called **conversion funnel**
- At each step of the funnel, users fall away
- The final step of the funnel is the conversion event



Predictive budget allocation (PBA) (1/2)

- To maximize conversions, **advertisers adjust budgets in proportion to performance**: Increase budgets for well-performing ads, and vice versa
- Large advertisers may have large numbers of campaigns
- Impractical to adjust budgets for everything by hand, and **need to adjust with respect to future performance**
- Ahonen (2017) [4] suggests finding optimal budgets using a Bayesian multi-armed bandit, where
  - Each arm is an ad
  - The cost is ad spend
  - The reward is conversions
- This is called **Predictive Budget Allocation (PBA)**

Predictive budget allocation (PBA) (2/2)

Initiate by allocating equal share of the budget to each ad;
while *Campaign is active* **do**
  Wait 1 day;
  Collect latest performance data;
  Calculate probability for each ad being best in terms of cost per conversion;
  Modify suggested proportions to fit constraints given by the ad platform;
  Set budgets of ads to match calculated proportions;
end

Algorithm 1: Predictive budget allocation algorithm from Ahonen (2017)

Conversion prediction (1/2)

- Any adjustments we make to the budgets only have an effect in the future
- **We have daily past performance data** (ad spend, views, clicks, conversions...)
- To predict future performance for PBA, **we treat these performance data as a time series**
- Then we can use any time series forecasting method we like

Conversion prediction (2/2)

- **Point predictive performance** is our primary interest
- In practice, point predictions are rarely completely accurate
- We would like to somehow quantify our uncertainty about the predictions
- Hence we are also interested in **probabilistic predictive performance**
- We would also like to measure robustness to two phenomena in conversion prediction: **Conversion signal delay** and **nonstationarity** of the time series

Conversion prediction phenomenon #1: Conversion signal delay

- After a user sees an ad, it may take time for them to convert
- For this reason, ad platforms report conversions with respect to an **attribution window** measured in days since the first ad impression
- Example: Two users see an ad on day 0. The first user makes a purchase immediately, and the second user waits 3 days to make a purchase.
    - With a 1-day attribution window, 1 conversion is reported
    - With a 7-day attribution window, 2 conversions are reported
- We would like a conversion prediction model that can work with whichever attribution window we choose

Conversion prediction phenomenon #2: Nonstationarity

- Under some circumstances, especially sales events, the **conversion rate** may change quickly
- This causes the time series to be nonstationary
- We would like a conversion prediction model that can give good predictions even when the parameters of the underlying data-generating process change

Research material and methods

Experiment 1/3: Effect of attribution window length

- For all experiments, we used time series of ad spend and conversions from real ad performance data (one model used additional features)
- **Attribution window data set:** Performance data with **1-day** and **7-day** attribution windows, respectively
  - The data set had two subsets, 1-day and 7-day
  - The performance data in each subset were from the same ads, so we could compare prediction results directly
- **Effect of attribution window length experiment:**
  - We ran each model we tested on 1,000 time series from the 1-day and 7-day attribution window subsets
  - The time series were chosen randomly, but were from the same ads

1-day attribution window	**7-day attribution window**
Ads 1, 2, 3. . .	Ads 1, 2, 3. . .

Experiment 2/3: Effect of nonstationarity

- **Stationarity data set:** Performance data partitioned by stationarity using the augmented Dickey-Fuller test [5] with $p = 0.05$.
  - The data set had four subsets, for each combination of attribution window and stationarity
  - The performance data in the stationary and nonstationary parts of the data set were not from the same ads, so we could only compare average prediction results
- **Effect of stationarity experiment:**
  - We ran each model we tested on performance data time series from 1,000 ads each from the stationary and nonstationary sub sets
  - We controlled for attribution window length by predicting with both lengths and averaging the results

	Stationary	**Nonstationary**
1-day attribution window	Ads 1, 2, 3...	Ads a, b, c...
7-day attribution window	Ads 1, 2, 3...	Ads a, b, c...

Experiment 3/3: Overall model performance

- **Overall point predictive performance experiment:**
  - We looked at point predictive performance across the previous two experiments
- **Overall probabilistic predictive performance experiment:**
  - We looked at probabilistic predictive performance across the previous two experiments
- 6,000 total time series predicted on for these experiments. Care was taken to weight each prediction equally.

Models (1/5): Persistence model

- Naïve baseline model: The persistence model
- Today's prediction $\hat{y}_t$ = yesterday's observation y_{t-1}
- For probabilistic predictions, just use a Poisson distribution with the prediction as the mean $\text{Pois}(\hat{y}_t)$

Models (2/5): ARIMAX

- ARIMAX (**A**uto**r**egressive **I**ntegrated **M**oving **A**verage with e**x**ogenous variable) is a classical statistical model
- Can be thought of as the persistence model taken a lot further:
  - Uses the last p observations, and also weights them
  - Differences the observations d times first to remove nonstationarity
  - Uses q weighted "innovation" terms to model the moving average
  - Uses r weighted exogenous terms
- The exogenous variable in our case was **ad spend**
- Also gives us probabilistic predictions in the form of confidence intervals

Models (3/5): Bayesian Poisson model

- Simple model written in Stan
- Idea: The number of conversions $\mathbf{y_t}$ is Poisson-distributed, with the rate parameter being an **unknown conversion rate $\boldsymbol{\lambda}$** times the **known amount of ad spend** that day $\mathbf{s_t}$
- Our task is to **estimate $\boldsymbol{\lambda}$**, which we do by summing up to the last 28 days of observed spend and conversions

$$\lambda \sim \text{Gamma}(1, 1)$$
$$\sum_{t=1}^{N} y_t \sim \text{Pois}\left(\lambda \sum_{t=1}^{N} s_t\right)$$

- Then we can predict tomorrow's conversions with tomorrow's spend and our estimate of $\boldsymbol{\lambda}$ as $\boldsymbol{y_{N+1} \sim \text{Pois}(\lambda s_{N+1})}$

Models (4/5): Bayesian Poisson model with Delays (1/2)

- We take the previous idea one step further, by modeling conversion delays due to attribution windows
- For a conversion window of length $\mathbf{W}$, any ad spend leads to conversions on the present day + the $\mathbf{W}$ following days
- These conversions arrive according to an unknown delay distribution p, which we try to estimate. For example if $W = 1$, $p = \begin{bmatrix} 0.4 & 0.6 \end{bmatrix}$ means that 40% of conversions arrive on the same day as the ad spend, and the remaining 60% arrive the following day
- We assume a Dirichlet prior for $\mathbf{p}$ and the same Gamma prior as before for $\boldsymbol{\lambda}$

Models (4/5): Bayesian Poisson model with Delays (2/2)

- We estimate $\boldsymbol{\lambda}$ and $\mathbf{p}$ with

$$y_t \sim \text{Pois}\left(\lambda s_t p_1\right) + \text{Pois}\left(\lambda s_{t-1} p_2\right) + \cdots + \text{Pois}\left(\lambda s_{t-W} p_{W+1}\right)$$

- To predict tomorrow's conversions, we calculate the rate parameters up to tomorrow using the estimated conversion rate and delay distribution (shifted forward) and known spend:

$$y_{N+1} \sim \text{Pois}\left(\lambda s_{N+1} p_1\right) + \text{Pois}\left(\lambda s_{N-1} p_2\right) + \cdots + \text{Pois}\left(\lambda s_{N-W} p_{W+1}\right)$$

Models (5/5): XGBoost

- We wanted to see if a pretrained model could give us reasonable out-of-sample predictions
- XGBoost is a tree-based model similar to a decision tree and has shown promise in the literature, Kaggle competitions, etc.
- We trained XGBoost on spend and conversions, but also impressions, clicks, and categorical features like which country the ads were run in
- We used 10,000 time series as training data
    - We trained one XGBoost model on 1-day attribution time series and one on 7-day attribution window time series
- Time-varying features were time-delay embedded

Evaluation methods: Point predictive performance

- We use mean absolute error (MAE) to measure point predictive error, but this is not enough due to the variety of time series scale
- No real consensus in the literature about what the best way to control for scale is; every method has caveats
- We chose a metric called **relative mean absolute error (rMAE)** suggested by Lago et al. [6] which normalizes any model's MAE by the MAE from the persistence model

$$\text{rMAE}(y, \hat{y}_m) = \frac{\text{MAE}(y, \hat{y}_m)}{\text{MAE}(y, \hat{y}_{\text{naïve}})}$$

Evaluation methods: Probabilistic predictive performance (1/2)

- We use **expected log predictive density (ELPD)** (Vehtari, Gelman and Gabry 2017 [7]) to evaluate probabilistic performance
- In particular, we use ELPD formulated for time series forecasting called **ELPD-LFO (leave future out)** (Bürkner, Gabry, and Vehtari 2020 [8]), here simplified for the 1-step ahead case:

$$\text{ELPD}_{\text{LFO}} = \log p(y_{t+1} \mid y_1, \ldots, y_t)$$

- For the Bayesian models, we estimate the density in the equation above using Monte-Carlo using draws from the posterior distribution

$$p(y_{t+1} \mid y_1, \ldots, y_t) \approx \frac{1}{S} \sum_{s=1}^{S} p(y_{t+1} \mid y_1, \ldots, y_t,\ \theta_1^{(s)}, \ldots, \theta_t^{(s)})$$

Evaluation methods: Probabilistic predictive performance (2/2)

- For the other models which give probabilistic predictions (persistence and ARIMAX), we calculate log-likelihoods, which are roughly comparable to ELPDs
- For the persistence model, this is $\log \mathrm{Pois}(y_t|\hat{y}_t)$
- For the ARIMAX model, we use the confidence intervals it provides to calculate the standard deviation $\boldsymbol{\sigma}$. We chose a normal distribution with mean $\hat{y}_t$ and variance $\boldsymbol{\sigma}^2$. We compute the likelihood by integrating this around the true value $\mathbf{y_t}$. In practice we use the CDF.

$$\mathrm{ELPD}_{\mathrm{ARIMAX}} = \log\left[\int_{y_t-\frac{1}{2}}^{y_t+\frac{1}{2}} \mathcal{N}\left(y \mid \hat{y}_t, \hat{\sigma}_t^2\right) \mathrm{d}y\right]$$

- Later we will see why this discretized normal distribution did not work.

Results and discussion

Overall point predictive performance

Model	MAE	rMAE
Persistence	107.68	1.00
ARIMAX	114.05	1.11
BPM	165.49	1.63
BPM+D	210.31	5.65
XGBoost	325.58	2.05

- None of our models could unambiguously outperform the naïve model
- ARIMAX, which is most similar to the persistence model, came closest
- The Bayesian Poisson models didn't do so well— we'll look at why

Overall probabilistic predictive performance

Model	Mean ELPD	MAE
Persistence	-51.25	107.68
ARIMAX	N/A	114.05
BPM	-131.51	165.49
BPM+D	-108.24	210.31

- The persistence model's naïve probabilistic predictions also seem to be the best
- The discretized normal distribution we chose for ARIMAX usually worked, but sometimes was too "certain", resulting in $-\infty$ log-likelihood

Impact of attribution window length (point predictions)

Model	MAE 1-day	MAE 7-day	rMAE 1-day	rMAE 7-day
Persistence	129.37	124.35	1.00	1.00
ARIMAX	119.67	130.97	1.08	1.12
BPM	154.28	229.50	1.24	2.04
BPM+D	199.93	213.63	6.37	3.47
XGBoost	252.46	570.74	1.74	1.85

- The Bayesian Poisson model with delays finally shows some improvement over the model without delays
- The additional complexity hurts it with short attribution windows

Impact of attribution window length (probabilistic predictions)

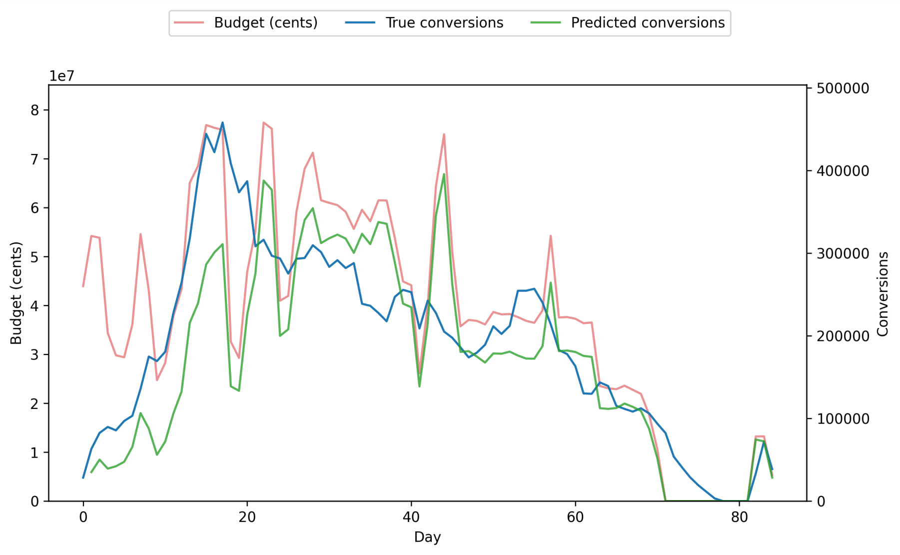
Model	Mean ELPD 1-day	Mean ELPD 7-day	ΔMean ELPD
Persistence	-77.71	-55.05	22.66
ARIMAX	N/A	N/A	N/A
BPM	-83.13	-248.17	-165.04
BPM+D	-107.84	-88.74	19.10

- BPM+D again shows its strength in the longer attribution window case

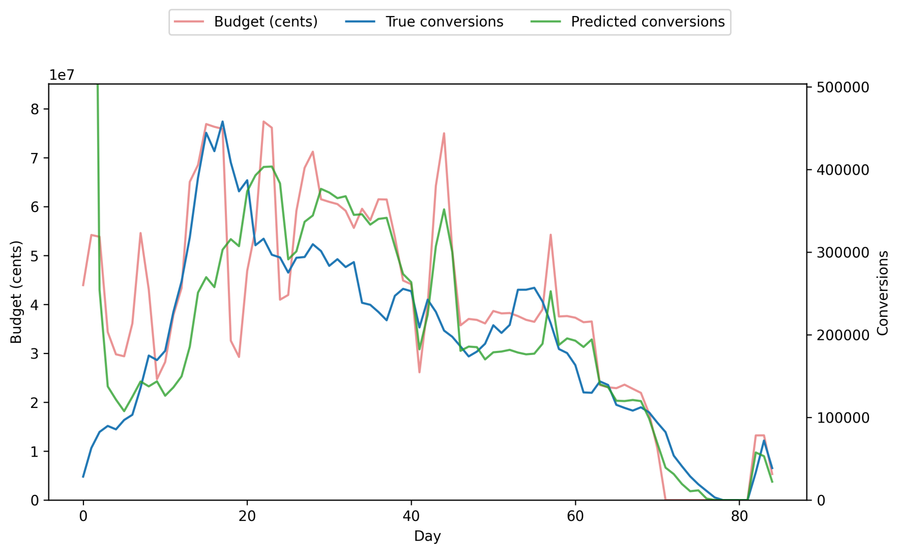
Bayesian models with long attribution windows

- The Bayesian model without delays gave predictions which were too closely tied to the day's spend
- The model with delays was able to model the decaying effect of previous days' spend
- ...But the model with delays also suffered from instability at the beginning of many time series

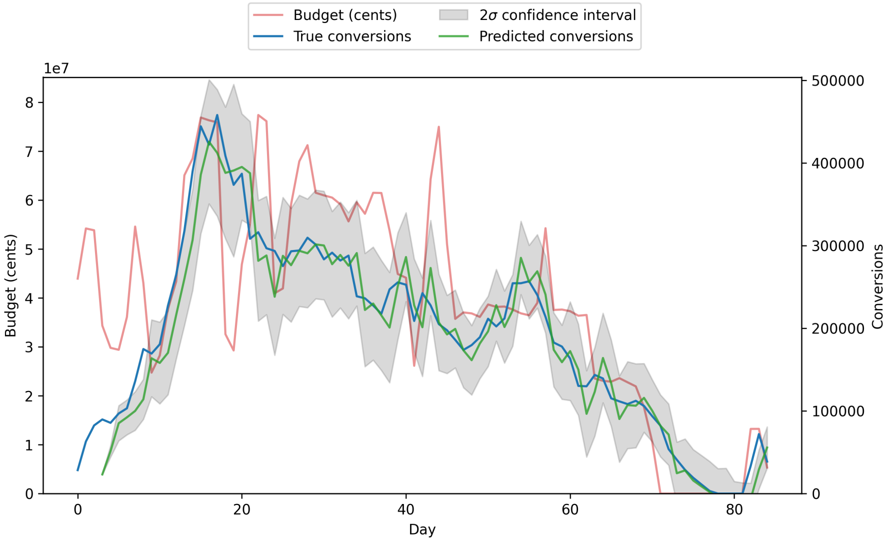
Bayesian Poisson model (no delays)



Bayesian Poisson model with delays



ARIMAX


Budget (cents)
True conversions
2σ confidence interval
Predicted conversions
1e7
Budget (cents)
Conversions
Day
0
20
40
60
80
0
1
2
3
4
5
6
7
8
0
100000
200000
300000
400000
500000

Impact of stationarity (point predictions)

Model	MAE S.	MAE N.S.	rMAE S.	rMAE N.S.
Persistence	201.60	29.40	1.00	1.00
ARIMAX	223.27	32.55	1.06	1.14
BPM	314.89	44.52	1.31	1.84
BPM+D	446.41	58.20	4.54	7.01
XGBoost	612.29	62.88	1.45	2.66

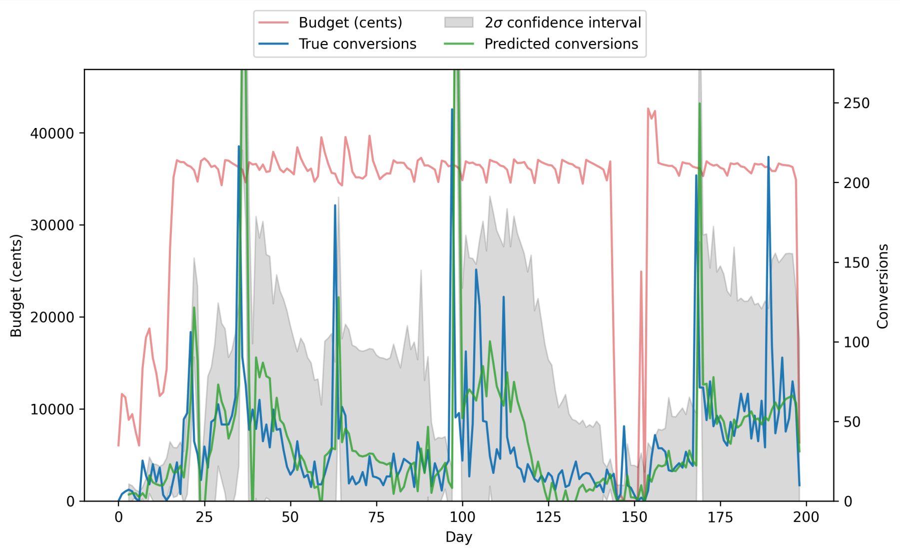
- The scale of the stationary **(S.)** time series was much larger than the nonstationary **(N.S.)** time series, so the MAE is not very relevant
- We are hesitant to draw strong conclusions from this test, but note in particular BPM+D's poor performance on nonstationary time series

Impact of stationarity (probabilistic predictions)

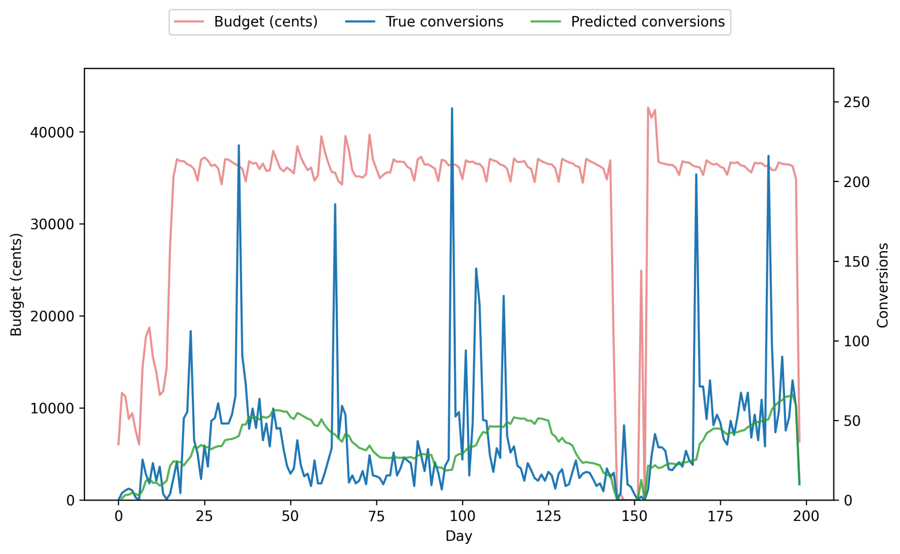
Model	Mean ELPD stationary	Mean ELPD nonstationary	ΔMean ELPD
Persistence	-72.56	-24.24	48.52
ARIMAX	N/A	N/A	N/A
BPM	-225.08	-40.79	184.29
BPM+D	-238.98	-31.23	207.76

- ELPD seems to be somewhat proportional to MAE, so again we are hesitant to draw strong conclusions from this test

Nonstationarity: ARIMAX



Nonstationarity: Bayesian Poisson model (no delays)



Future work

- The Bayesian models need to adjust to changes in conversion rate more quickly
  - Model each day's conversion rate as a random walk from the previous day
- A Poisson distribution assumes variance equal to the mean
  - A negative binomial distribution allows for more variance through the overdispersion parameter
  - We tried a negative binomial model, but the results were very unstable for the BPM+D model
- ARIMAX: To get usable ELPD, Bayesian ARIMAX from Matamoros et al. [9]
- Without probabilistic predictions, XGBoost isn't useful for us as a conversion prediction method
  - Bayesian tree-based models do exist however, see Bayesian Additive Regression Trees (BART) from Chipman, George, and McCulloch [10]

Conclusion

Conclusion

- Conversion prediction is not as straightforward as it may first appearance
  - The data are often sparse
  - Not all conversions arrive right away
- We weren't able to make a model that totally outperformed the naïve model, but each model gave us hints.
- Measuring time series forecasting performance when the time series scale varies so much is not easy!
- For advertisers, this is a problem worth solving. Managing so many budgets is a real challenge, and misspending is a big risk.

Thanks and Acknowledgements

Thank you to Prof. Aki Vehtari for supervising my thesis, to PhD (Tech.) Hannu Laine for advising, and to PhD (Tech.) Topi Paananen for help with ELPD calculations. Thank you to my employer, Smartly.io Solutions Oy, for providing me with opportunity and support to do my master's thesis.

Questions?

References i

- Advertising spending worldwide from 2000 to 2024.
https://web.archive.org/web/20220828204012/https://www.statista.com/statistics/1174981/advertising-expenditure-worldwide/.
[Online; accessed 2022-08-28].
- Meta Reports Fourth Quarter and Full Year 2021 Results.
https://web.archive.org/web/20220301170032/https://investor.fb.com/investor-news/press-release-details/2022/Meta-Reports-Fourth-Quarter-and-Full-Year-2021-Results/default.aspx.
[Online; accessed 2022-03-01].

Snap Inc. Announces Fourth Quarter and Full Year 2021 Financial Results.
https://web.archive.org/web/20220217233444/https://investor.snap.com/news/news-details/2022/Snap-Inc.-Announces-Fourth-Quarter-and-Full-Year-2021-Financial-Results/default.aspx.
[Online; accessed 2022-02-17].

Niko-Petteri Ahonen.
Applying Bayesian Bandits For Solving Optimal Budget Allocation In Social Media Marketing.
Master's thesis, Aalto University School of Science, 2017.

References iii

Wayne A Fuller.
Introduction to Statistical Time Series.
John Wiley & Sons, 2009.

Jesus Lago, Grzegorz Marcjasz, Bart De Schutter, and Rafał Weron.
Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark.
Applied Energy, 293:116983, 2021.

Aki Vehtari, Andrew Gelman, and Jonah Gabry.
Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC.
Statistics and computing, 27(5):1413–1432, 2017.

Paul-Christian Bürkner, Jonah Gabry, and Aki Vehtari.
Approximate leave-future-out cross-validation for Bayesian time series models.
Journal of Statistical Computation and Simulation, 90(14):2499–2523, 2020.

Asael Alonzo Matamoros, Cristian Cruz Torres, Andres Dala, Rob Hyndman, and Mitchell O'Hara-Wild.
bayesforecast: Bayesian Time Series Modeling with Stan, 2022.
Available at `https://cran.r-project.org/package=bayesforecast`, version 1.0.1.

References v

Hugh A. Chipman, Edward I. George, and Robert E. McCulloch.
BART: Bayesian additive regression trees.
The Annals of Applied Statistics, 4(1):266–298, 2010.