Causally Driven Incremental Multi Touch Attribution Using a Recurrent Neural Network*

Ruihuan Du
JD.com
duruihuan@jd.com

Yu Zhong
JD.com
zhongyu5@jd.com

Harikesh S. Nair
JD.com and Stanford Univ.
harikesh.nair@jd.com

Bo Cui
JD.com
cuibo@jd.com

Ruyang Shou
JD.com
shouruyang@jd.com

ABSTRACT

This paper describes a practical system for Multi Touch Attribution (MTA) for use by a publisher of digital ads. The approach has two steps, comprising response modeling and credit allocation. For step one, we train a Recurrent Neural Network (RNN) on user-level conversion and exposure data. The RNN has the advantage of flexibly handling the sequential dependence in the data while capturing the impact of advertising intensity, timing, competition, and user-heterogeneity, which are known to be relevant to ad-response. For step two, we compute Shapley Values, which have the advantage of having axiomatic foundations and satisfying fairness considerations. The specific formulation of the Shapley Value we implement respects incrementality by allocating the overall incremental improvement in conversion to the exposed ads, while handling the sequence-dependence of exposures on the observed outcomes. The system is deployed at JD.com, and scales to handle the high dimensionality of the problem on the platform.

CCS CONCEPTS

• **Information systems** → **Online advertising**; • **Applied computing** → *Online shopping*; *Economics*;

KEYWORDS

multi touch attribution, recurrent neural networks, deep learning, Shapley values, advertising, marketing.

ACM Reference format:
Ruihuan Du, Yu Zhong, Harikesh S. Nair, Bo Cui, and Ruyang Shou. 2019. Causally Driven Incremental Multi Touch Attribution Using a Recurrent Neural Network. In *Proceedings of ACM Woodstock conference, Anchorage, Alaska USA, August 2019 (ADKDD'19),* 6 pages. https://doi.org/10.475/123_4

*The authors are part of JD Intelligent Ads Lab. An extended version is on arXiv; sample code is at https://github.com/jd-ads-data/jd-mta.

ADKDD'19, August 2019, Anchorage, Alaska USA

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

1 INTRODUCTION

The issue of Multi Touch Attribution (MTA) is one of high importance to advertisers and digital publishers. MTA pertains to the question of how much the marketing touchpoints a user was exposed to, contributes to an observed action by the consumer. Understanding the contribution of various marketing touchpoints is an input to campaign design, to optimal budget allocation and for understanding the reasons for why campaigns work. Wrong attribution results in misallocation of resources, inefficient prioritization of touchpoints, and lower return on marketing investments.

This paper develops a data-driven MTA system for a publisher of digital ads. We developed this system for JD.com, an eCommerce company, which is also a publisher of digital ads in China. Our approach has two steps. The first step ("response modeling") fits a user-level model for purchase of a brand's product as a function of the user's exposure to ads. The second ("credit allocation") uses the fitted model to allocate the incremental part of the observed purchase due to advertising, to the ads the user is exposed to over the previous T days.

In step one, our goal is to develop a response model that accommodates heterogeneously the responsiveness of user purchase behavior *to the sequence* (temporal ordering) of past own *and* competitor advertising exposures, with response driven by the intensity of those ad-exposures with possible time-decay. An MTA system that comprehensively accommodates all these features is currently missing in the literature. Sequential dependence is key to the ad-response, because what we need to capture from the data is how exposure to past touch points cumulatively build up to affect the final outcome.

Given the scale of the data, and the large number of ad-types, a fully non-parametric model that reflect these considerations is not feasible. Instead, we train a Recurrent Neural Network (RNN) for ad-response. The specific RNN

we present is architected to capture the impact of advertising intensity, timing, competition, and user-heterogeneity outlined above while accommodating sequence-dependence.

To implement step two, we focus on incrementality-based allocation for advertising. We allocate only the incremental improvement generated by advertising to an order to each exposed ad-type, ensuring that the part of observed orders that would have occurred anyway is not allocated to a focal brand's ads. To allocate credit, we compute Shapley Values (henceforth "SVs"), which have the advantage of having axiomatic foundations and satisfying fairness considerations. The specific formulation of the SVs we implement respects incrementality while handling the sequence-dependence of exposures on the observed outcomes.

Computing the SVs is computationally intensive. We present a new scalable algorithm (implemented in a distributed `MapReduce` framework) that is fast enough to make practical productization feasible. The algorithm takes predictions from the response model as an input, and allocates credit over tuples of ad-exposures and time periods. Allocation at the tuple-level has the advantage of handling the role of the sequence in an internally consistent way, which is new to the literature. Once allocation of credit at this level is complete, credit allocation over any desired bundle of exposure options (such as ad-channels like search and display) can be developed by exact aggregation, reducing aggregation biases.

We present an illustration of the framework using data from the cell-phone product category at `JD.com`. This is a single category version of the full framework. The full framework which has been deployed accommodates all product categories on the platform, and scales to handle the dimensionality (attribution of the orders of about 300M users, for roughly 160K brands, across 200+ ad-types, served about 80B ad-impressions over a typical 15-day period).

2 RELATED WORK

Previous art has developed various empirically specified ad-response models using *user-level* browsing and conversion data (e.g. [2, 3, 6, 8, 14, 17, 21]). Compared to this stream, which simplifies the sequential dependence, the RNN model used here handles complex patterns of dependence in a more flexible way, and allows different sequences of touchpoints to have differential effects on final conversion, which is novel. The setup also accommodates in one framework the role of intensity, timing, competition, and user-heterogeneity, which have not been considered together previously.

Dalessandro et al. [6], Yadagiri et al. [18] combine data-based response models with SVs for the MTA problem, but abstract away from the explicit role of the sequence and allocate the total, rather than the incremental component of orders. In contrast, we focus on incremental allocation, and the allocation of credit to an ad-slot in our model depends explicitly on its order in the temporal sequence of exposures, which is more general. Also related are versions of RNNs presented by li et al. [9], Ren et al. [12] for user response, but without an SV-based allocation scheme.

A limitation of our approach and of all the response models cited previously, is the lack of exogenous variation in user exposure to advertising. Extant papers that have trained ad-response models on data with full or quasi-randomization (e.g., [4, 10, 13, 20]) have done so at smaller scale, over limited number of users and ad-types due to the cost and complexity of such randomization, and have not considered the corresponding credit-allocation problem. The approach adopted here is to include a large set of user features into the model as synthetic control. Controlling flexibly for these observables mitigates the selection issue (e.g., [16]), albeit not perfectly.

3 MODEL FRAMEWORK

3.1 Problem Definition

Let $i = 1, .., N$ denote users; $t = 1, .., T$ days; and $b = 1, .., B$ brands. Let $k = 1, .., K$ index an "*ad-position*," i.e., a particular location on the publishers inventory or at an external site at which the user can see advertisements of a given brand. For e.g., an ad-slot showing a display ad on the top frame of the `JD` app home-page would be one ad-position. Consider an order $o(i, b, T)$ made by user i for brand b on day T. Let $\mathcal{A}_{ibT} \subseteq K$ denote the set of ad-positions at which user i was exposed to ads for brand b over the T days preceding the order (from $t = 1$ to T).

The attribution problem is to obtain a set of credit-allocations $\varrho_k(\mathcal{A}_{ibT})$ for all $k \in \mathcal{A}_{ibT}$, so that the allocation for k represents the contribution of brand b's ads at position k to the expected *incremental* benefit generated by brand b's advertising on the order. Define $v(\mathcal{A}_{ibT})$ as the change in the probability of order $o(i, b, T)$ occurring due to the user's exposure to b's ads in the positions in $\mathcal{A}_{ibT}$. We look for a set of fractions $\varrho_k(\mathcal{A}_{ibT})$ such that, $\sum_{k \in \mathcal{A}_{ibT}} \varrho_k(\mathcal{A}_{ibT}) = v(\mathcal{A}_{ibT})$.

3.2 Problem Solution

To allocate the orders on date T, in step 1, we train a response model for purchase behavior using individual user-level data observed during $t = 1$ to T. In step 2, we take the model as given, and for each order $o(i, b, T)$ observed on date T, we compute SVs for the ad-positions $k \in \mathcal{A}_{ibT}$. We set $\varrho_k(\mathcal{A}_{ibT})$ to these SVs and aggregate across orders to obtain the overall allocation for brand b on date T.

3.2.1 SVs. The SV (Shapley [15]) is a fair allocation scheme in situations of joint generation of outcomes. A fair allocation avoids waste and allocates all the benefit to the units that generated it ("allocative efficiency"). Fairness also suggests that two units that contribute the same to every possible configuration should be allocated the same credit, and a unit which contributes nothing should be allocated no credit. Fair

allocations also satisfy the "marginality principle", that the share of total joint benefit allocated should depend only on that unit's own contribution to the joint benefit (see Young [19]). The appeal of the SV is that it is the *unique* sharing rule that is efficient, symmetric and satisfies the marginality principle (Young [19]).

Defining the Expected Incremental Benefit to an Order. Let $Y_{ibT} \in (0,1)$ denote a binary random variable for whether user i purchases brand b on day T. An order $o(i,b,T)$ is a realization $Y_{ibT} = 1$ with associated own-brand ad-exposures at positions $\mathcal{A}_{ibT}$. The expected incremental benefit generated by brand b's advertising on order $o(i,b,T)$ is $v(\mathcal{A}_{ibT}) = \mathbb{E}[Y_{ibT}|\mathcal{A}_{ibT}] - \mathbb{E}[Y_{ibT}|\varnothing_b]$. Holding everything else fixed, this difference represents the expected incremental contribution of the ad-positions in $\mathcal{A}_{ibT}$ to the order. We can think of $v(\mathcal{A}_{ibT})$ as a causal effect of brand b's advertising over the past T days on user i's propensity to place the observed order on day T.

Allocating Incremental Benefit to a Position-Day Tuple. To allocate $v(\mathcal{A}_{ibT})$ to the ad-positions in $\mathcal{A}_{ibT}$, we first allocate $v(\mathcal{A}_{ibT})$ to each *ad-position-day tuple* in which i saw ads of brand b over the last T days. We then sum the allocations across days for the tuples that each ad-position $k \in \mathcal{A}_{ibT}$ is associated with, to obtain the overall allocation of $v(\mathcal{A}_{ibT})$ to that k.

To do this, let $\mathcal{N}_{ibT}$ be the set of ad-position-day combinations at which user i saw ads for brand b during the T days preceding order $o(i,b,T)$. For a given tuple $\{k,t\}$, let S denote a generic element from the power-set of $\mathcal{N}_{ibT} \setminus \{k,t\}$. Let the cardinalities of these sets be denoted $|\mathcal{N}_{ibT}|$, $|S|$.

Define the function $w(S) = \mathbb{E}[Y_{ibT}|S] - \mathbb{E}[Y_{ibT}|\varnothing_b]$, i.e., $w(S)$ represents the expected incremental benefit from user i seeing ads for brand b at ad-position-day combinations in S, holding everything else fixed. By construction, $w(\mathcal{N}_{ibT}) = v(\mathcal{A}_{ibT})$. So, by allocating $w(\mathcal{N}_{ibT})$ across the ad-position-day tuples in $\mathcal{N}_{ibT}$, we allocate the same total incremental benefit generated by brand b's advertising, as we would by allocating $v(\mathcal{A}_{ibT})$ across the ad-positions in $\mathcal{A}_{ibT}$. Also, by construction, $w(\varnothing_b) = 0$.

We need the fractions $\varrho_{\{k,t\}}(\mathcal{N}_{ibT})$, to satisfy two conditions. First, that $\sum_{\{k,t\} \subseteq \mathcal{N}_{ibT}} \varrho_{\{k,t\}}(\mathcal{N}_{ibT}) = w(\mathcal{N}_{ibT})$, so that the fractions sum to the full incremental benefit of the ads on the order (i.e., satisfy allocative efficiency). Second, that the fractions for a given tuple $\{k,t\}$ are functions of only its marginal effects with respect to $w(.)$ (i.e., satisfy the marginality principle). These are the SVs for the tuples $\{k,t\} \subseteq \mathcal{N}_{ibT}$ defined as,

$$\varrho_{\{k,t\}}(\mathcal{N}_{ibT}) = \sum_{S \subseteq \mathcal{N}_{ibT} \setminus \{k,t\}.} \frac{|S|!(|\mathcal{N}_{ibT}| - |S| - 1)!}{|\mathcal{N}_{ibT}|!} \Big[w(S \cup \{k,t\}) - w(S) \Big] \tag{1}$$

Computing the SVs requires a way to estimate the marginal effects $w(S \cup \{k,t\}) - w(S)$ in equation (1) from the data, as well as an algorithm that scales to handle the high-dimensionality of S. This is discussed in the subsequent section.

Once the SVs $\varrho_{\{k,t\}}(\mathcal{N}_{ibT})$ are computed, we sum them across all t to obtain the allocation of that order to ad-position k as, $\varrho_k(\mathcal{N}_{ibT}) = \sum_{t \subseteq \mathcal{N}_{ibT}(k)} \varrho_{\{k,t\}}(\mathcal{N}_{ibT})$, where $\mathcal{N}_{ibT}(k)$ is the set of days in $\mathcal{N}_{ibT}$ that are associated with ad-position k.

The final step is to do this across all orders observed for brand b on day T. To do this, we sum $\varrho_k(\mathcal{N}_{ibT})$ across all k and all users who bought brand b on day T. This gives the overall incremental contribution of the K ad-positions to the brand's orders. To allocate this to k, we simply compute how much ad-position k contributed to this sum. Denote $\varrho_k(\mathcal{N}_{ibT})$ above as ϱ_{ibkT} for short. We allocate to ad-position k, a proportion Ψ_{bkT},

$$\Psi_{bkT} = \frac{\sum_{\{i:Y_{ibT}=1\}} \varrho_{ibkT}}{\sum_k \sum_{\{i:Y_{ibT}=1\}} \varrho_{ibkT}} \tag{2}$$

Each element k in $\mathbf{\Psi}_{bT} = (\Psi_{b1T}, .., \Psi_{bKT})$ represents the contribution of ad-position k to the total incremental orders obtained on day T by the brand due to its advertising on the K positions. $\mathbf{\Psi}_{bT}$ thus represent a set of attributions that can be reported back to the advertiser.

Linking to a Response-Model. Let x_{ibkt} be the number of impressions of brand b's ad seen by user i at ad-position k on day t. Collect all the impressions of the user for the brand's ad across positions on day t in $\mathbf{x}_{ibt} = (x_{ib1t}, .., x_{ibKt})$; collect the impression vectors across all the brands for that user on day t in $\mathbf{x}_{it} = (\mathbf{x}_{i1t}, .., \mathbf{x}_{iBt})$; and stack the entire vector of impressions across all days and brands in a $(K \cdot B \cdot T) \times 1$ vector $\mathbf{x}_{i,1:T} = (\mathbf{x}_{i1}, .., \mathbf{x}_{iT})'$. Let p_{bt} be a price-index for brand b on day t, representing an average price for products of brand b faced by users on day t.[1] Collect the price indices for all brand on day t in vector $\mathbf{p}_t = (\mathbf{p}_{1t}, .., \mathbf{p}_{Bt})$ and stack these in a $(B \cdot T) \times 1$ vector $\mathbf{p}_{1:T} = (\mathbf{p}_1, .., \mathbf{p}_T)'$. Finally, let $\mathbf{d}_i$ represent a $R \times 1$ vector of user characteristics collected at baseline. The probability of purchase on day T is modeled as a function of user characteristics, and the ad-impressions and price-indices of brand b and all other brands in the product category over the last T days as $\mathbb{E}[Y_{ibT}] = \Pr(Y_{ibT} = 1) = \sigma\left(\mathbf{x}_{i,1:T}, \mathbf{p}_{1:T}, \mathbf{d}_i; \hat{\Omega}\right)$. The probability model $\sigma(.)$ is parametrized by vector $\hat{\Omega}$ which will be learned from the data.

We use $\mathbb{E}[Y_{ibT}]$ as above along with the definition of the marginal effects $w(S)$, to compute the SVs defined in equation (1). To obtain the marginal effects from the response model, we define an operator $\Gamma_b(.)$ on $\mathbf{x}_{i,1:T}$ that takes a

[1] We compute this as a share weighted average of the list prices of the SKUs associated with the brand on that day.

set S as defined in §3.2.1 as an input.[2] Given S, $\Gamma_b(.)$ sets all the impressions of brand b apart from those in the ad-position-day tuples in S to 0. $\Gamma_b(.)$ also leaves impressions of all brands $b' \neq b$ unchanged.

Mathematically, taking $\mathbf{x}_{i,1:T}$ and S as input, $\Gamma_b\left(\mathbf{x}_{i,1:T}, S\right)$ outputs a transformed vector $\mathbf{x}_{i,1:T}^{(b,S)}$ computed as,

$$x_{ib'kt}^{(b,S)} = \begin{cases} 0 & \text{if} \quad b' = b \text{ and } \{k,t\} \subsetneq S \\ x_{ib'kt} & \text{otherwise} \end{cases} \tag{3}$$

With $x_{ib'kt}^{(b,S)}$ as defined above, we can compute the SV using the response model as,

$$\varrho_{\{k,t\}}(\mathcal{N}_{ibT}) = \sum_{S \subseteq \mathcal{N}_{ibT} \setminus \{k,t\}.} \frac{|S|!(|\mathcal{N}_{ibT}| - |S| - 1)!}{|\mathcal{N}_{ibT}|!} \Big[\sigma\Big(\mathbf{x}_{i,1:T}^{(b,S\cup\{k,t\})}, \mathbf{p}_{1:T}, \mathbf{d}_i; \hat{\Omega}\Big) - \sigma\Big(\mathbf{x}_{i,1:T}^{(b,S)}, \mathbf{p}_{1:T}, \mathbf{d}_i; \hat{\Omega}\Big) \Big] \tag{4}$$

In effect, what we obtain in the square brackets in equation (4) is the change in the predicted probability of purchase of an order of brand b on day T by user i when the tuple $\{k,t\}$ is added to the set of ad-position-day combinations in S, holding everything else (including competitor advertising) fixed at the values observed in the data for that order. The `arXiv` version of the paper provides an example.

Efficient Algorithm for Fast, Large-Scale Computation. Exact computation of SVs as described above is computationally intensive. SVs have to be calculated separately for each ad-position-day tuple for each order (in the millions). When $|\mathcal{N}_{ibT}|$ is large, this latter step also becomes computationally intensive, requiring Monte Carlo simulation methods to approximate the calculation.

Our implementation switches between exact and approximate solutions for the SVs depending on the cardinality of $\mathcal{N}_{ibT}$, and is implemented in a MapReduce framework so it runs in a parallel, distributed environment. Algorithm 1 in the `arXiv` version of the paper presents details.

3.2.2 Response Model. The response model provides the marginal effects in equation (1). The architecture of the RNN is presented in Figure (1). Though the model training is done simultaneously across all brands, the picture is drawn only for one brand. The input vector of ad-impressions, $\mathbf{x_{it}}$, and the input vector of price-indexes, $\mathbf{p}_t$, are fed through an LSTM layer with recurrence. The user characteristics, $\mathbf{d}_i$, are processed through a separate, fully-connected layer. The outputs from the LSTM cells and the fully-connected layer jointly impact the predicted outcome $\tilde{Y}_{it}$. Combining this with the observed outcome,Y_{it}, we obtain the log-likelihood $\mathcal{L}$, which forms the loss function for the model. The RNN finds a set of parameters or weights that maximizes the log-likelihood.

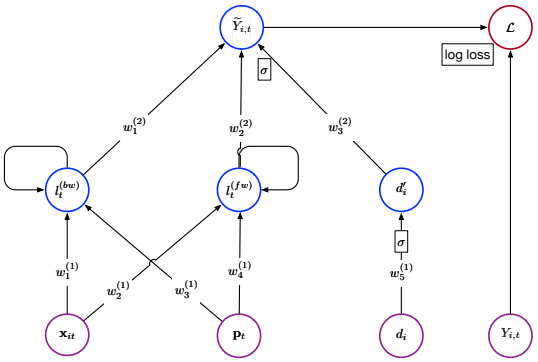


Figure 1: Computational Graph for RNN

As noted before, we utilize a bi-directional formulation. This is shown in Figure (1) where the superscript "fw" indicates forward recurrence and "bw" indicates backward recurrence. This improves the fit of the model, because the fact that a user i saw a particular set of ads in $t+1, .., T$ is useful to predict his response in period t. For example, if a user bought a brand in t, he may not search for that brand in period $t+1$, and not be exposed to search ads in $t+1$. So the knowledge that he did not see search ads in $t+1$ is useful to predict whether he will buy in period t. When computing SVs for an order, the entire sequence of ads for T periods prior to the order are known, so the use of the bi-directional RNN as a predictive model presents no conceptual difficulty. The `arXiv` version of the paper provides additional details of the implementation, including an extension to much larger scale to accommodate all product categories (not just one as above) in one unified framework.

4 EMPIRICAL APPLICATION

Code for a simulation that implements the system is provided at: `https://github.com/jd-ads-data/jd-mta`. Here, we discuss an application of the model using user-level data from the cell-phone product category on `JD.com` during a 15-day window in 2017. To create the training data, we sample users who saw during the window, at least one ad-impression related to a product in the cell-phone product category on `JD.com`. We define the *positive sample* as the set of users who purchased a product of any brand in the cell-phone category during the time window. We define the *negative sample* as the set of users who did not purchase any product in the cell-phone category during the 15-day time window. Table 1 provides summary statistics.

At the brand-level, the recall and precision of the model are `Huawei` (.340,.714), `Xiaomi` (.308,.705), `Apple` (.218,.696),

[2] Recall from §3.2.1 that we use S to refer to a sub-set of ad-position-day combinations at which the user saw ads for brand b during the T days, excluding tuple $\{k, t\}$.

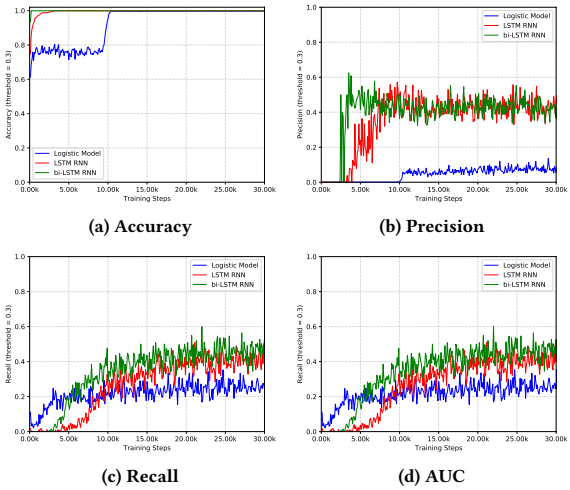
Table 1: Summary Statistics of Training Data

Number of users in Overall Sample	75,768,508
Number of users in Positive Sample	2,100,687
Number of users in Negative Sample	73,667,821
Num: of ad-impressions in category	7,153,997,856
Num: of orders in category	3,477,621
Number of orders made on day $T = 15$	175,937
Number of brands (B)	31
Number of ad-positions (K)	301

`Meizu` (.186,.727), `Xiaomi` (.119,.762), `Others` (.095,.726), showing it fits the data well (accuracy > .99 for all). Figure 2 compares the accuracy, precision, recall and AUC (area under the curve) statistics of the model against two benchmark specifications.[3] The first is a flexible logistic model (analogous to early papers like Shao and Li [14]), which specifies the probability of a user i purchasing brand b on day t as a semi-parametric logistic function of the ad-impressions and price-indexes on the same day (i.e.,$\Pr(Y_{ibt} = 1) = \text{logistic}(\mathbf{x}_{it}, \mathbf{p}_t; \Omega)$). The second is a unidirectional LSTM RNN, which is the same as the preferred model but without the forward recurrence.

Looking at the results, we see the bi-directional RNN has the highest AUC amongst the models; and has accuracy, precision and recall statistics that are comparable or higher. The poor performance of the logistic model emphasizes the importance of accounting for dependence over time to fit the data. The plots also show the speed of convergence of the models as a function of the number of training steps; the bi-directional RNN converges faster in fewer training steps. This is helpful in production, which typically requires frequent model updating.[4]

Tables 2 also benchmarks the algorithm for distributed computation of SVs. Recall this algorithm shifts from exact computation of SVs to a Monte Carlo simulation approximation when the number of ad-positions over which to allocate credit is large. This "mixed" method improves computational speed, which is important for high-frequency reporting of results in deployment. To assess the performance of this algorithm, we pick 6,000 orders from $t = T$ and run the algorithm on these data for various configurations. The experiment is repeated for each configuration 5 times, and the average across the 5 reps is reported.[5] The first row in the table reports on the number of orders we are able to attribute per minute using the three methods: the mixed method is about 2,300% faster than exact computation, and



(a) Accuracy **(b) Precision**

(c) Recall **(d) AUC**

Figure 2: Performance Benchmarks

Table 2: Processing Efficiency of SV Computation

Algorithm	Exact	Approximate	Mixed
Orders processed per minute	4.2	88.85	101.24
Error	–	0.3190	0.0064

about 14% faster than a simulation-only method. The second row documents this efficiency gain does not come at the cost of high error: the average error in the mixed method is low relative to exact computation, and a order of magnitude smaller than using full simulation.[6]

Additional details on credit allocation from the model are reported in the `arXiv` version of the paper which shows (1) that allocating only the incremental component of orders is important; and (2) that SVs have discriminatory power in picking high-contribution ad-positions. We also document there that intensity, sequence, timing and competitive effects matter for capturing ad-response in these data; and that controls for selection via user attributes have value.

Figure 3 compares the credit allocation to "Last-clicked" attribution. We show the SVs at each ad-position on the x-axis, averaged across all orders on $T = 15$ for which that position was the last clicked. The SVs are all seen to be <1; in contrast, "last-click" would allocate these ad-positions full-credit. To the extent that the SVs are all less than 0.6, the model suggests that last-clicked ads contribute a maximum of 60% to the incremental conversion due to advertising. As a gut check, cart and payment page positions which may

[3]The statistics are computed on a validation dataset that is held-out separately from the training dataset.

[4]To get a sense for this, the training times for 30,000 steps for the 3 models on our cluster are 11.21 hrs (bi-directional RNN); 9.68 hrs (unidirectional RNN); 12.48 hrs (logistic) hrs respectively.

[5]The computational environment uses a `Spark` cluster with `Spark` 2.3 by `pyspark`, running `TensorFlow` v1.6, with an 8 core CPU, 100 workers and 8 GB memory per worker without the GPU.

[6]We compute error as a mean squared difference over the $l = 1, \ldots 6000$orders in the total attributed value (`line 20` in Algorithm 1) for the evaluated algorithm (SA_l) relative to that from the exact algorithm $\left(SA_l^*\right)$, i.e. err = $\sqrt{\frac{1}{6000}\sum_{l=1}^{6000}\left(SA_l - SA_l^*\right)^2}$.

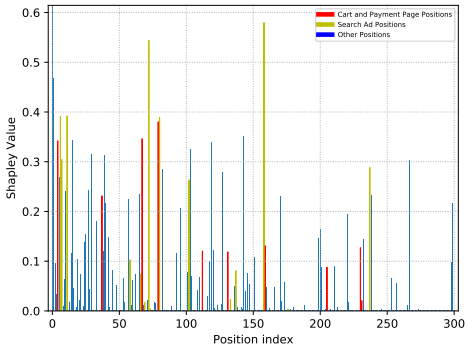
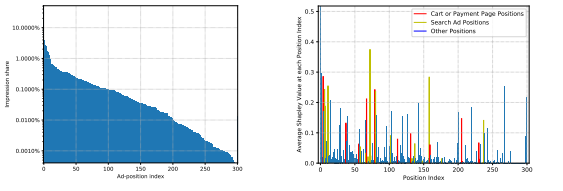


Figure 3: Benchmark Against Last-Click Attribution

get a lot of credit under "last-click" or "last-visit" attribution schemes on eCommerce sites, are seen to not be allocated a lot of credit by the model.

As a final assessment, we compare Figure 4a which shows the proportion of ad-impressions in the data across the ad-positions to Figure 4b which shows the average (across orders) of the SVs for the same positions. The ad-positions are indexed from 1 – 301 in order of their share of total impressions. Comparing the two, we can see that the distribution of SVs across positions do not follow the same pattern as that of impressions, suggesting that the model is not purely picking up differences in intensity of advertising expenditures. We observe that some positions that receive fewer ad-impressions have higher SVs than those that receive higher impressions. This suggests that advertising expenditure allocations overall may not be optimal from the advertiser's perspective, and could be improved by incorporating better attribution. A more formal assessment however also requires a method for advertiser budget allocation, which is outside of the scope of this paper.



(a) Imp: Share by Ad-Position **(b) Shapley Values by Ad-Position**

Figure 4: Shapley Values by Impression Share

5 CONCLUSIONS

A practical system for data-driven MTA for use by a publisher is presented. A bi-directional RNN for ad-response is developed as the response model, which is semi-parametric, reflects many aspects of ad-response, and able to handle high dimensionality and long-term dependence. The SV used allocates credit in a way that respects the sequential nature of ad-response and fairness considerations. An area of future research would be to use the model for campaign budget allocation and bidding (e.g., Geyik et al. [7], Pani et al. [11]). The SV takes the advertisers' actions as given. It is possible that advertisers re-optimize their advertising policies in response to the attribution. The optimal contract that endogenizes equilibrium response remains an open question (e.g., Abhishek et al. [1], Berman [5] for theory).

REFERENCES

[1] V. Abhishek, S. Despotakis, and R. Ravi. 2017. Multi-Channel Attribution: The Blind Spot of Online Advertising. *mimeo, CMU* (2017).
[2] V. Abhishek, P. Fader, and K. Hosanagar. 2015. Media Exposure through the Funnel: A Model of Multi-Stage Attribution. *mimeo, UPenn* (2015).
[3] E. Anderl, I. Becker, F. von Wangenheim, and J.H. Schumann. 2016. Mapping The Customer Journey: Lessons Learned From Graph-Based Online Attribution Modeling. *IJRM* (2016).
[4] J. Barajas, R. Akella, M. Holtan, and A. Flores. 2016. Experimental Designs and Estimation for Online Display Advertising Attribution in Marketplaces. *Marketing Science* 35, 3 (2016), 465–483.
[5] R. Berman. 2018. Beyond the Last Touch: Attribution in Online Advertising. *Marketing Science* forthcoming (2018).
[6] B. Dalessandro, C. Perlich, O. Stitelman, and F. Provost. 2012. Causally Motivated Attribution for Online Advertising *(ADKDD '12)*.
[7] Sahin Cem Geyik, Abhishek Saxena, and Ali Dasdan. 2014. Multi-Touch Attribution Based Budget Allocation in Online Advertising. In *ADKDD 2014*.
[8] H. Li and P.K. Kannan. 2014. Attributing Conversions in a Multichannel Online Marketing Environment: An Empirical Model and a Field Experiment. *Journal of Marketing Research* 51, 1 (2014).
[9] Ning li, Sai Kumar Arava, Chen Dong, Zhenyu Yan, and Abhishek Pani. 2018. Deep Neural Net with Attention for Multi-channel Multi-touch Attribution. *ADKDD* (2018).
[10] H. S. Nair, S. Misra, W.J. Hornbuckle, R. Mishra, and A. Acharya. 2017. Big Data and Marketing Analytics in Gaming: Combining Empirical Models and Field Experimentation. *Marketing Science* 36, 5 (2017).
[11] Abhishek Pani, S. Raghavan, and Mustafa Sahin. 2018. Large-Scale Advertising Portfolio Optimization in Online Marketing. *Working Paper, Univ. of Maryland* (2018).
[12] Kan Ren, Yuchen Fang, Weinan Zhang, Shuhao Liu, Jiajun Li, Ya Zhang, Yong Yu, and Jun Wang. 2018. Learning Multi-touch Conversion Attribution with Dual-attention Mechanisms for Online Advertising. In *CIKM '18*.
[13] N. Sahni. 2015. Effect of Temporal Spacing between Advertising Exposures: Evidence from Online Field Experiments. *QME* 13 (2015).
[14] X. Shao and L. Li. 2011. Data-driven Multi-touch Attribution Models. In *SIGKDD*.
[15] L.S. Shapley. 1953. *A Value for N-Person Games.* Princeton University Press, Chapter Annals of Mathematical Studies, 307–317.
[16] H. R. Varian. 2016. Causal Inference in Economics and Marketing. *PNAS* 113 (2016).
[17] L. Xu, J. A. Duan, and A. Whinston. 2014. Path to Purchase: A Mutually Exciting Point Process Model for Online Advertising and Conversion. *Management Science* 60, 6 (2014).
[18] M.M. Yadagiri, S.K. Saini, and R. Sinha. 2015. A Non-parametric Approach to the Multi-channel Attribution Problem. In *WISE*.
[19] H. P. Young. 1988. *Individual Contribution And Just Compensation.* Cambridge University Press, 267–278.
[20] D. Zantedeschi, E.M. Feit, and E. T. Bradlow. 2017. Measuring Multi-channel Advertising Response. *Management Science* 63, 8 (2017).
[21] Y. Zhang, Y. Wei, and J. Ren. 2014. Multi-touch Attribution in Online Advertising with Survival Theory. In *IEEE ICDM*.