Machine Learning Attribution: Inferring item-level impact from slate recommendation in e-commerce

Shenghe Xu
Amazon.com
Seattle, USA
shenghe@amazon.com

Yan Zhao
Amazon.com
Seattle, USA
yzhaoai@amazon.com

Sameer Kanase
Amazon.com
Seattle, USA
kanases@amazon.com

Mitchell Goodman
Amazon.com
Seattle, USA
migood@amazon.com

Saad Khan
Amazon.com
New York, USA
saadmk@amazon.com

Brent Payne
Amazon.com
Seattle, USA
paybren@amazon.com

Patricia Grao
Amazon.com
Seattle, USA
pgrao@amazon.com

ABSTRACT

Slate level recommendation is widely adopted in online services including e-commerce, video streaming and news services. Customers may observe a set of recommended items and interact with the content accordingly. Due to the combinatorial characteristics of slate level recommendation, various current ranking models are still aiming to optimize item level scores, instead of the slate level score. One key challenge of training such models is to attribute slate level reward correctly to each item. In this paper, we propose two learning based approaches named StepNet and SlateNet to estimate item level attributed reward. In StepNet we train models from different sets of data and use the difference of the estimated scores from two models as the attributed rewards. In SlateNet, we train a model on seen portions of slate-level recommendations and use this model to infer the impact of removing seen content to determine attribution. Simulation results show that both StepNet and SlateNet achieve normalized mean absolute error lower than 0.1 for attribution results with up to 8 items per slate. Validation on real dataset also confirms that SlateNet models are able to learn the intrinsic features of content and their relationship to different types of rewards.

1 INTRODUCTION

Recommending a slate of items or widgets is a common practice nowadays for online services [20, 22, 28]. Online video or music streaming services may recommend several items for the users to choose from on the same page, and a list of articles is usually shown on news webpages to users[6, 33]. Similarly, online e-commerce sites rank sets of widgets on their webpages to provide personalized experiences to customers. Since customers see a slate of recommended widgets for each request, it is often impossible to tie the reward generated by customers back to individual items. However, due to the combinatorial nature of slate level recommendation, the action space for choosing a slate of m widgets from n candidates can become intractable online for large n, making it impractical to consider every slate level combination online. Methods such as the commonly used learning-to-rank (LTR) [4, 5, 9, 21] try to estimate the score for each item first, then return the top K items. In order to reduce the bias [1, 32, 34, 35] and correctly assign scores to each item, we need a method that could suitably assigns reward to each item.

In this paper, we focus on the problem of attribution for slate level recommendation in e-commerce. We assume each widget in the slate contains content which could impact customers' behavior. Unlike recommendations for videos or news, in e-commerce simple metrics such as click-through-rate could not be directly used as the optimization objective. Each widget may be associated with commodities of significantly different values. Clicking on a \$5 item may generate far less value than clicking on a \$1000 item. The recommendation models need to maximize objectives such as purchases, revenue or other numerical numbers related to sales.

In addition, one unique problem in e-commerce is the existence of intricate interaction between widgets in the same slate. One example is the interaction between substitutes, which are similar products that may differ only in a small number of dimensions such as brand or price. In some cases showing a more expensive substitute may boost sales of the cheaper substitute. On the slate level, the reward received would be the purchase of the cheaper substitute. But on the item level, if we attribute all the reward only to the cheaper substitute, the model will fail to consider the boost of sales brought by showing the more expensive substitute. If we attribute the reward to all the items that were shown, irrelevant items would receive reward even if they did not bring incremental value. An attribution method that could correctly consider the interactions between widgets would ultimately improve the ranking performance.

Overall, the main contributions of this paper are

- We consider the special customer behaviors for e-commerce, and introduce the view-based marginal treatment effect (VMTE) to better measure the incremental value brought by recommending a widget.
- Two methods named StepNet and SlateNet are proposed, both exploiting the customers' viewing preference to estimate VMTE. Simulation results show that both methods are able to achieve lower than 0.1 normalized mean absolute

errors. To the best of our knowledge, this is the first work to utilize such kind of customer behavior to attribute item-wise reward from a slate.
- We implemented the SlateNet based attribution method in production. Validation on real dataset confirms that SlateNet models are able to learn the intrinsic features of content and their relationship to different types of rewards.

The rest of the paper is organized as follows. Section 2 introduces the related work for treatment effect estimation and reward attribution. In Section 3, we formulate the problem and introduce VMTE. Learning based attribution methods are introduced in 4. We describe the simulation setup in Section 5. Simulation results are presented in Section 6. We further validate our method with real data in Section 7. Conclusions are drawn in Section 8.

2 RELATED WORK

The reward attribution problem in slate level recommendation can be formulated as a problem of estimating the treatment effect, which has been studied in several papers [2, 12, 12, 18, 23–25]. When estimating the treatment effect, many studies make a simple assumption that the treatment effect is the same across the population. However, this assumption is not appropriate due to heterogeneity. For example, even for the same treatment assigned to the population, the treatment effect might still be different for certain subgroups of the population due to people's differing characteristics. In order to consider the treatment effect properly, the uplift model was proposed to obtain the conditional average treatment effect (CATE) as the estimation of the heterogeneous effect [8, 13]. In terms of calculating the CATE, we have to know both of the potential outcomes (treated and untreated) simultaneously. However, for a single unit, only one of the potential outcomes could be observed. In order to approximate the CATE under the limitation of observational data, the unconfoundedness assumption [7] must be made, which will remove any potential bias arising due to different units.

Several models have been proposed to estimate the CATE or the lift effect for a treatment. The simplest one is called S-learner [12], which stands for the single uplift model. The main idea is that the covariates and treatment variables will all be treated as model features, so there is no special role in the treatment variable and there is only one estimator. Once the estimation is done, the CATE can be calculated by using estimations from the corresponding groups; and can be further improved by incorporating propensity scores. Another approach is called the T-learner [2, 23, 25]. Instead of using one estimator, the T-learner utilizes two estimators to estimate the response functions separately for the treatment and control groups. It also involves two models. For the first model, two base learners will be learnt by separately estimating the conditional expectations of the control and treatment data. The second model takes the difference of the estimations from these two base learners. Built on the T-learner, the X-learner was proposed in [18] to improve the performance on imbalanced data, where data in the treatment group is generally less than the control group. The X-learner is able to leverage information in the control to get a better estimators for the treatment or vice versa. It consists of three stages. Compared to the T-learner, it creates imputed effects for the control and treatment group based on the base learners learned from first step, and then the final estimation is the weighted average of the estimation of the imputed effects by using propensity score. [24] proposed an approach called R-learner, also called 'residual-to-residual'. The R-learner builds two models, one of which models the correlation between the treatment and the covariates and the other of which models the correlation between the response and the covariates. Then another model is built on the residuals of each model. Essentially, the R-learner assumes that once the confoundedness is removed, the correlation between the treatment and the response is the true treatment effect. In [3, 30], casual forest was proposed and it was used to estimate the heterogeneous treatment effect in online marketplace in [31].

On the other hand, several papers also consider the interaction between items in the case of session based recommendation and slate based recommendation [11, 16, 22, 29]. In [29], a convolutional sequence embedding method was proposed to consider the interaction between items in session based recommendation. A reward interaction propensity scoring based approach was proposed in [22] to better evaluate the reward for actions in session based recommendation. In [16], a method named SlateQ was proposed to decompose the reward for slate recommendation, making it tractable for reinforcement learning.

Unlike the previous papers which either focus on the definition and measurement of treatment effect [7, 8, 13] or the estimation of the CATE for single treatments. In this paper we focus on the problem of attribution for slate level recommendations, where each item in the slate corresponds to a treatment. This paper does not limit the problem setting on session based recommendation [11, 22, 29] or specific ranking methods such as reinforcement learning [16]. We study the general reward attribution problem, so that the proposed method could still be combined with current learning to rank algorithms. In addition, we also consider the specific characteristics of recommendations for e-commerce and provide a problem specific definition for treatment effect. Two neural network (NN) based methods are proposed to address the attribution problem in slate based recommendation.

3 PROBLEM FORMULATION

3.1 Problem Setting

We focus on the problem of recommending a slate of widgets to customers. A widget is a single piece of content used for recommendation. Assuming there are N candidate widgets $w_1, w_2, ..., w_N$, a slate of k widgets can represented by a vector $\boldsymbol{s} = (w_i, w_j, ...w_k)^T$. Note that w_1 shows at the top of the page, so widgets with a higher rank number will be put at lower positions on the page. The received reward each time a slate is recommended can be written as $Y(\boldsymbol{s}, \boldsymbol{x})$, where $\boldsymbol{x}$ is the context including customer preference, browsing history etc. In the settings of the LTR methods, we need to train a model $M(w_i, \boldsymbol{x})$ to estimate the score for each widget w_i. One of the most straight forward approaches is to train $M(w_i, \boldsymbol{x})$ to approximate $E[Y(\boldsymbol{s}, \boldsymbol{x})|w_i \in \boldsymbol{s}]$. However, this attribution of value fails to consider contribution of each widget to the reward of the total slate. One example may be a blank widget that is always shown with other high value widgets, even if it contributes no value, it will always receive a high score with this kind of attribution.

3.2 Definition of Treatment Effect

Instead of using the naive approach of assigning the slate level reward to all the widgets in the slate, the average marginal treatment effect (AMTE) [8, 13] could be used as a measurement for the average value brought by a widget. The AMTE for w_i can be written as

$$\Phi_a(w_i, w_0, \boldsymbol{x}) = \sum_{\boldsymbol{b} \in B} P(\boldsymbol{b}) E[Y((\boldsymbol{b}, w_i), \boldsymbol{x}) - Y((\boldsymbol{b}, w_0), \boldsymbol{x})] \quad (1)$$

where w_0 is the baseline widget that ideally should be a blank widget, $\boldsymbol{b}$ is the vector of the other widgets, $P(\boldsymbol{b})$ is the probability of $\boldsymbol{b}$ being shown, B is the set of all possible $\boldsymbol{b}$. We use $(\boldsymbol{b}, w_i)$ to represent the concatenated vector from $\boldsymbol{b}$ and w_i . The AMTE represents the expected incremental value of choosing w_i given the context of $\boldsymbol{x}$.

The AMTE evaluates a widget over all the possible combinations with other widgets. When attributing to a single data point, with a fixed recommendation of widgets, the average over other widgets is removed and we obtain the marginal treatment effect (MTE)

$$\Phi_b(w_i, w_0, \boldsymbol{x}) = Y((\boldsymbol{b}, w_i), \boldsymbol{x}) - Y((\boldsymbol{b}, w_0), \boldsymbol{x}). \quad (2)$$

One main issue of applying MTE is finding a suitable baseline treatment w_0. The most ideal solution is to send empty widgets as baseline and collect the data. However this could cause negative customer experience and potentially harm the overall performance of the recommendation system. In practice, we found that even though a slate of widgets is always recommended, the widgets are not always rendered and viewed. In fact, for one of the slates in our system, only around 50% of the recommended widgets are rendered. The widgets that are not rendered will not show on the customer's screen and will have no impact on the customer's behavior. Therefore we utilize this feature in the dataset and use the widgets that are not rendered as w_0.

In addition, due to the common pattern of customers' viewing behavior, widgets with higher positions usually get viewed before widgets with lower positions. When attributing for a widget at rank i, if the baseline assumes the widget is not viewed, in most cases widgets with lower positions will not be viewed either. Therefore we define the view-based marginal treatment effect (VMTA) as

$$\Phi_v(w_i, w_0, \boldsymbol{x}) = Y((\boldsymbol{b}^k, w_i), \boldsymbol{x}) - Y((\boldsymbol{b}^k, w_0), \boldsymbol{x}). \quad (3)$$

where $\boldsymbol{b}^k$ denotes the other widgets with higher positions than w_i. Essentially, the VMTA considers the incremental value brought by adding a widget to an existing partially ranked slate.

4 LEARNING BASED ATTRIBUTION

In real world datasets, the w_i and w_0 in VMTA cannot be observed at the same time for a single data point. Therefore we have to estimate one of them to calculate the VMTA. We proposed two different methods called StepNet and SlateNet to estimate $Y((\boldsymbol{b}^k, w_0), \boldsymbol{x})$ and $Y((\boldsymbol{b}^k, w_i), \boldsymbol{x})$. Then the VMTA can be calculated by

$$\widetilde{\Phi}_v(w_i, w_0, \boldsymbol{x}) = \widetilde{Y}((\boldsymbol{b}^k, w_i), \boldsymbol{x}) - \widetilde{Y}((\boldsymbol{b}^k, w_0), \boldsymbol{x}). \quad (4)$$

where $\widetilde{Y}((\boldsymbol{b}^k, w_i), \boldsymbol{x})$ and $\widetilde{Y}((\boldsymbol{b}^k, w_0), \boldsymbol{x})$ are the estimated value provided by the models.

4.1 Step-based Neural Network for Reward Estimation

In the step-based neural network (StepNet) reward estimation method, the dataset is divided into several subsets based on the number of widgets that have been viewed. For each of the subset D_i with the top i widgets viewed, a model $M^i_{step}((\boldsymbol{b}^{i-1}, w_i), \boldsymbol{x})$ is trained to estimate the reward $Y((\boldsymbol{b}^{i-1}, w_i), \boldsymbol{x})$. After training is completed, the VMTA can be estimated by calculating $M^i_{step}((\boldsymbol{b}^{i-1}, w_i), \boldsymbol{x}) - M^{i-1}_{step}(\boldsymbol{b}^{i-1}, \boldsymbol{x})$. In this case we are essentially using the models trained from D_{i-1} and D_i to estimate the VMTA for the widget in position i.

4.2 Slate-based Neural Network for Reward Estimation

For the slate-based neural network (SlateNet) approach, the whole dataset is used to train a single model. However, the widgets that are not viewed are represented with wild cards. For a data point with only the top i widgets viewed, a model $M_{slate}((\boldsymbol{b}^{i-1}, w_i, \boldsymbol{u}^{N-i}), \boldsymbol{x})$ is trained, where $\boldsymbol{u}^{N-i}$ is a wild card vector for $N - i$ widgets. After model training is completed, the VMTA can be estimated by calculating $M_{slate}((\boldsymbol{b}^{i-1}, w_i, \boldsymbol{u}^{N-i}), \boldsymbol{x}) - M_{slate}((\boldsymbol{b}^{i-1}, \boldsymbol{u}^{N-i+1}), \boldsymbol{x})$. In this approach, we are replacing the input of w_i with a wild card to the model to calculate the estimated baseline reward. In practice, we represent each widget with a one-hot vector, and the wild card is a vector with all zeros.

For an online dataset, the reward distribution could be different if the number of widgets that are viewed is different. We expect StepNet to estimate reward more accurately since it uses different models for different number of viewed items. However StepNet uses N models for a slate of N widgets, the SlateNet method trains one model and could potentially reduce training cost for large slates.

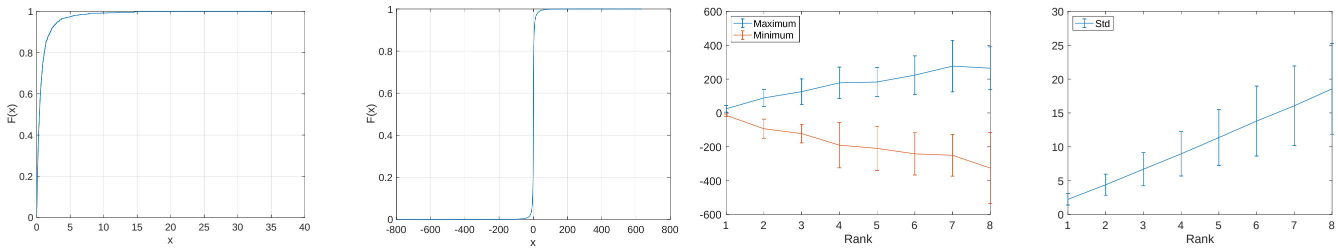
5 SIMULATION SETUP

5.1 Reward Model

Due to the nature of the recommendation problems, the ground truth reward for each item is not available in the real dataset. We evaluate the performance of our methods with simulation to better measure their performance in terms of estimation error. We adopt a similar reward model used in [10, 14] for our simulation. In our simulation, we assume that each widget w_i has an associated value v_i. Then the reward can be formulated as

$$R = \boldsymbol{h}_v(\boldsymbol{a})^T \boldsymbol{v} + \sum_i \boldsymbol{h}_x(a_i)^T \boldsymbol{x} + \boldsymbol{v}^T \boldsymbol{H}_{vv}(\boldsymbol{a}) \boldsymbol{v} + \boldsymbol{v}^T \boldsymbol{H}_{vx}(\boldsymbol{a}) \boldsymbol{x} \quad (5)$$

where $\boldsymbol{h}_v(\boldsymbol{a})$ is the coefficient for the values of each widget, $\boldsymbol{h}_x(a_i)$ is the vector of coefficients between the context and the ith widget, $\boldsymbol{H}_{vv}(\boldsymbol{a})$ is the matrix of coefficient for interaction between different widgets, $\boldsymbol{H}_{vx}(\boldsymbol{a})$ is the matrix of coefficient for interaction between widgets and context. We only focus on second order interactions since higher orders are usually negligible [14]. All of the coefficients are functions of the slate layout $\boldsymbol{a}$, which is determined by the position of widgets.



(a) Empirical CDF of v. **(b) Empirical CDF of reward** **(c) Maximum and minimum reward across rank** **(d) Reward Std across rank**

Figure 1: Empirical CDF of parameters in test datasets

5.2 Parameter Selection

We assume there are 100 candidate widgets for ranking. To capture the long tail characteristics of rewards in e-commerce, values v_i are generated from a Lomax distribution with the probability density function

$$f(x) = \begin{cases} \frac{\alpha}{(1+x)^{\alpha+1}} & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{6}$$

where we set $\alpha = 2$. For each slate 8 widgets are selected and forms a layout $\boldsymbol{a}$. We assume $\boldsymbol{H}_{vv}(\boldsymbol{a})$ is determined by the widgets in the slate, i.e. $\boldsymbol{H}_{vv}(\boldsymbol{a})_{i,j} = \boldsymbol{G}_{a_i,a_j}$ where $\boldsymbol{G}$ is a constant matrix generated from a normal distribution with mean of 0 and variance of 1. Similarly, $\boldsymbol{h}_v(\boldsymbol{a})_i = \boldsymbol{t}_{a_i}$, $\boldsymbol{h}_x(\boldsymbol{a})_i = \boldsymbol{M}_{a_i}$ and $\boldsymbol{H}_{vx}(\boldsymbol{a})_{i,j} = \boldsymbol{P}_{a_i,j}$, where $\boldsymbol{t}$, $\boldsymbol{M}$ and $\boldsymbol{P}$ are also generated from a normal distribution with mean of 0 and variance of 1. Since the context is expected to have a smaller impact on the reward, the context $\boldsymbol{x}$ is generated randomly for each sample from a normal distribution with mean of 0 and but a smaller variance of 0.1.

The models in SlateNet and StepNet all use fully connected NNs [15] with hidden layers of size 256 and 128. The NNs are trained with the Adam [17] optimizer with learning rate of 0.001. All the NNs are trained for 100 epochs. We assume there are at most 8 widgets in a slate. 4,000,000 data points are used for training and 400,000 data points are used for testing. For the training set the data is evenly distributed by the number of widgets been viewed, i.e. each D_i consists of 500,000 data points. For testing, 50,000 data points are generate initially, but each is used for testing on all 8 ranks by varying the number of viewed widgets. The simulation is repeated with 10 different randomly picked seeds.

6 SIMULATION RESULTS

6.1 Distribution of Reward

Due to the Pareto distribution of $\boldsymbol{v}$, as shown in figure 1, over 95% of the elements are less than 5, but the maximum value is over 35. Over 90% of final rewards are between -10 and 10, but the extreme values of reward can be lower than -800 or higher than 600. Such kind of extreme values are also found in realworld datasets.

In addition, as the number of viewed widgets increases, the range of rewards also increases. Figure 1 (c) shows the change of maximum and minimum rewards as more widgets get viewed. Figure 1 (d) illustrates the standard deviation (Std) change when more widgets are viewed. As more widgets are viewed, more interaction between widgets introduce more variance in reward, which could potentially make estimation of reward harder.

6.2 Performance evaluation

We evaluate StepNet and SlateNet by both slate level reward estimation accuracy and widget level attribution accuracy. Since the Std of rewards varies by the random seeds, the performance should be mainly evaluated by the normalized mean absolute error (NMAE).

$$NMAE = \frac{\sum_i |X_i - \hat{X}_i|}{\sum_i |X_i|} \tag{7}$$

where X_i is the ground truth value and $\hat{X}_i$ is the estimated value. Table 1 shows the average NMAE scores its Std across seeds. Note that for rank one no attribution is required, so we all putting zeros for the attribution errors at rank 1.

Figure 2 (a) and (b) show the model estimation errors as the rank increases. For rank one to rank three StepNet performs better than SlateNet, however for rank four and above, SlateNet achieves lower MSE and lower NMAE.

In Figure 2 (c) and (d) we evaluate the attribution error, i.e. the estimation error of VMTA for StepNet and SlateNet. With better estimation accuracy at rank one and rank two, StepNet is able to more accurately estimate VMTA at rank two. However, for rank four and above, SlateNet is able to attribute values to items more correctly.

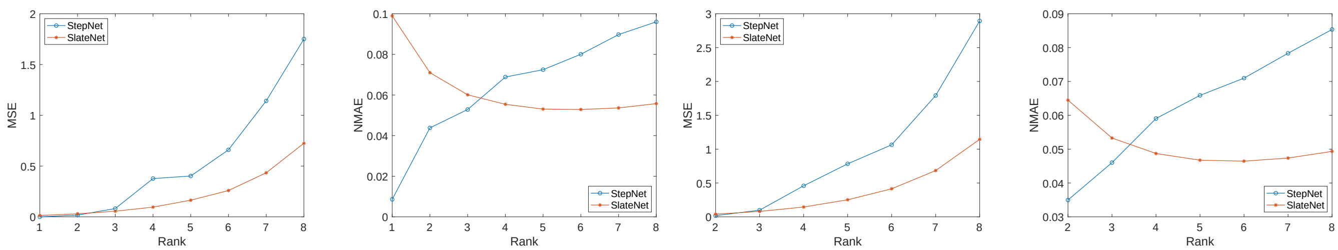
Given the change of reward distribution across ranks, one might expect StepNet to perform better than SlateNet since it uses different models to handle different sets of homogeneous data. However this is only true for rank one two and three. One possible reason is that the data points with not viewed items cause some form of data augmentation effect, like dropout [27] in neural network training or data augmentation tricks such as random cropping for image data [26]. Using such data points helps the model to learn a representation that is closer to the real function.

7 VALIDATION ON ONLINE DATASET

To further validate the performance of SlateNet. We trained a SlateNet model with online data. The dataset contains widget ids, context and slate level rewards. For each slate, 4 widgets are chosen from over

Table 1: Estimation and attribution errors

Metrics	Rank							
	1	2	3	4	5	6	7	8
StepNet Est. NMAE	0.0086	0.0438	0.0529	0.0688	0.0725	0.0801	0.0898	0.0960
StepNet Est. Std	0.0026	0.0101	0.0059	0.0184	0.0111	0.0115	0.0161	0.0166
SlateNet Est. NMAE	0.0989	0.0710	0.0601	0.0554	0.0531	0.0529	0.0536	0.0558
SlateNet Est. Std	0.0231	0.0167	0.0145	0.0135	0.0135	0.0141	0.0143	0.0148
StepNet Attr. NMAE	0.0000	0.0350	0.0460	0.0590	0.0659	0.0710	0.0783	0.0854
StepNet Attr.. Std	0.0000	0.0080	0.0058	0.0122	0.0107	0.0109	0.0133	0.0168
SlateNet Attr.. NMAE	0.0000	0.0645	0.0533	0.0487	0.0467	0.0465	0.0474	0.0493
SlateNet Attr. Std	0.0000	0.0169	0.0141	0.0129	0.0125	0.0128	0.0132	0.0136



(a) Reward estimation MSE **(b) Reward estimation NMAE** **(c) Attribution MSE** **(d) Attribution NMAE**

Figure 2: Reward estimation and attribution MSE and NMAE

327 candidates for desktop and 282 candidates for mobile. Six desktop candidates and four mobile candidates are ad widgets, the rest of the candidates are organic content that do not contain ads. Even though it is impossible to obtain ground-truth item level rewards, we are able to validate the efficiency of SlateNet attribution using the simple logic that ad related widgets should be attributed with more ad revenue. Note that the ad revenue used in this validation contains down session components, so widgets that increase down session ad revenue should also be attributed with more ad revenue.

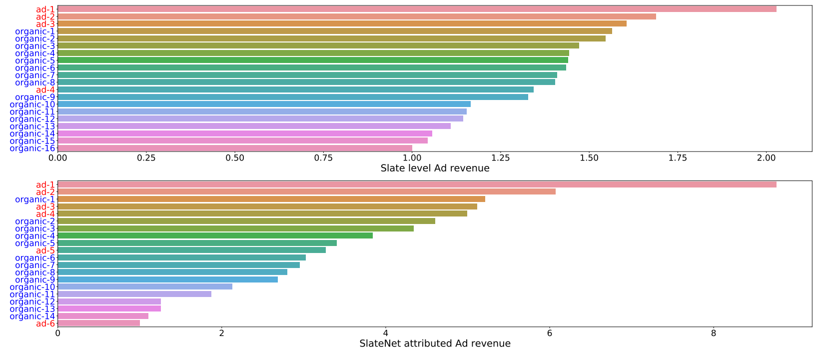
Since the production recommendation model for this dataset is a linear [19]model, we trained a linear model based SlateNet model for ads revenue attribution on 200 million training samples. Attribution was performed on a validation set with 60 million data points. Figure 3 shows the comparison between total slate level ad revenue and the SlateNet attributed ad revenue. For each subfigure, widgets are sorted by average attributed ad revenue, then the top 25 widgets are used for plotting. The values are divided by the smallest numbers to better compare the relative orders of attributed revenues. According to the results, the SlateNet based attribution model is able to attribute ad revenue more to ad widgets, for both mobile and desktop recommendations. Moreover, unlike slate-level revenues where all widgets attributions are distributed almost uniformly, SlateNet is able to draw clearer decision boundaries between ad widgets and organic widgets. Another interesting finding is some organic widgets are also pushed up by SlateNet, which seems counter-intuitive, but after diving deep into a few of them by looking at their online A/B experiments, we found that these widgets are able to drive more customer interactions on detail pages or guide customers to search pages which in turn benefit site-wise ad revenue gains.

8 CONCLUSION AND FUTURE WORK

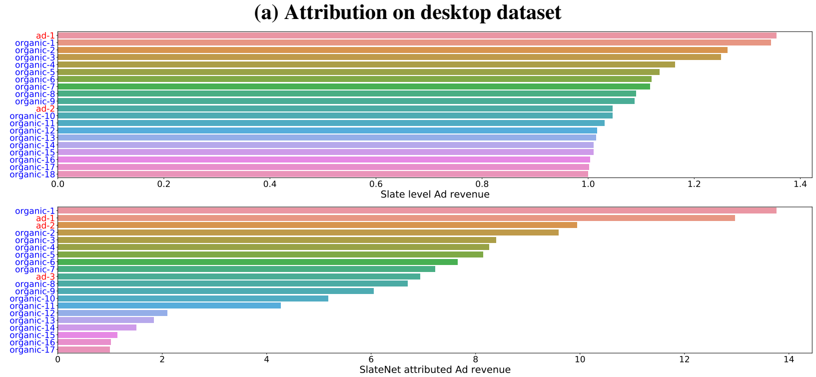
In this paper, by taking customers' viewing behavior into account, we derived a new treatment effect measurement called view-based marginal treatment effect (VMTA) for slate level recommendation reward attribution. The VMTA measures the incremental value of adding an item to the current slate. We proposed SlateNet and StepNet to estimate VMTA. According to simulation results, both StepNet and SlateNet are able to attribute widget level reward with NMAE lower than 0.1. StepNet may attribute more accurately for ranks less than four, while SlateNet could achieve better results for rank four or higher.

REFERENCES

[1] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. Controlling popularity bias in learning-to-rank recommendation. In *Proceedings of the eleventh ACM conference on recommender systems*, pages 42–46, 2017.
[2] Susan Athey and Guido W Imbens. Machine learning methods for estimating heterogeneous causal effects. *stat*, 1050(5):1–26, 2015.
[3] Susan Athey, Julie Tibshirani, and Stefan Wager. Generalized random forests. *The Annals of Statistics*, 47(2):1148–1178, 2019.
[4] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96, 2005.
[5] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136, 2007.
[6] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, et al. The youtube video recommendation system. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 293–296, 2010.
[7] Stephen G Donald, Yu-Chin Hsu, and Robert P Lieli. Testing the unconfoundedness assumption via inverse probability weighted estimators of (l) att. *Journal of Business & Economic Statistics*, 32(3):395–415, 2014.
[8] Naoki Egami and Kosuke Imai. Causal interaction in factorial experiments: Application to conjoint analysis. *Journal of the American Statistical Association*, 2018.



(a) Attribution on desktop dataset



(b) Attribution on mobile dataset

Figure 3: Attribution for online dataset

[9] Yingqiang Ge, Shuyuan Xu, Shuchang Liu, Zuohui Fu, Fei Sun, and Yongfeng Zhang. Learning personalized risk preferences for recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 409–418, 2020.

[10] Thore Graepel, Joaquin Quinonero Candela, Thomas Borchert, and Ralf Herbrich. Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine. In *ICML*, 2010.

[11] Lei Guo, Hongzhi Yin, Qinyong Wang, Tong Chen, Alexander Zhou, and Nguyen Quoc Viet Hung. Streaming session-based recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1569–1577, 2019.

[12] P Richard Hahn, Jared S Murray, Carlos M Carvalho, et al. Bayesian regression tree models for causal inference: Regularization, confounding, and heterogeneous effects (with discussion). *Bayesian Analysis*, 15(3):965–1056, 2020.

[13] Jens Hainmueller, Daniel J Hopkins, and Teppei Yamamoto. Causal inference in conjoint analysis: Understanding multidimensional choices via stated preference experiments. *Political analysis*, 22(1):1–30, 2014.

[14] Daniel N Hill, Houssam Nassif, Yi Liu, Anand Iyer, and SVN Vishwanathan. An efficient bandit algorithm for realtime multivariate optimization. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1813–1821, 2017.

[15] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[16] Eugene Ie, Vihan Jain, Jing Wang, Sanmit Narvekar, Ritesh Agarwal, Rui Wu, Heng-Tze Cheng, Tushar Chandra, and Craig Boutilier. Slateq: A tractable decomposition for reinforcement learning with recommendation sets. In *Proceedings of the Twenty-eighth International Joint Conference on Artificial Intelligence (IJCAI-19)*, pages 2592–2599, Macau, China, 2019. See arXiv:1905.12767 for a related and expanded paper (with additional material and authors).

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[18] Sören R Künzel, Jasjeet S Sekhon, Peter J Bickel, and Bin Yu. Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the national academy of sciences*, 116(10):4156–4165, 2019.

[19] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010.

[20] Shuchang Liu, Fei Sun, Yingqiang Ge, Changhua Pei, and Yongfeng Zhang. Variation control and evaluation for generative slate recommendations. In *Proceedings of the Web Conference 2021*, pages 436–448, 2021.

[21] Tie-Yan Liu. Learning to rank for information retrieval. *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, 2010.

[22] James McInerney, Brian Brost, Praveen Chandar, Rishabh Mehrotra, and Benjamin Carterette. Counterfactual evaluation of slate recommendations with sequential reward interactions. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1779–1788, 2020.

[23] Houssam Nassif, Finn Kuusisto, Elizabeth S Burnside, and Jude W Shavlik. Uplift modeling with roc: An srl case study. In *ILP (Late Breaking Papers)*, pages 40–45. Citeseer, 2013.

[24] Xinkun Nie and Stefan Wager. Quasi-oracle estimation of heterogeneous treatment effects. *Biometrika*, 108(2):299–319, 2021.

[25] Nicholas Radcliffe. Using control groups to target on predicted lift: Building and assessing uplift model. *Direct Marketing Analytics Journal*, pages 14–21, 2007.

[26] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019.

[27] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[28] Adith Swaminathan, Akshay Krishnamurthy, Alekh Agarwal, Miroslav Dudík, John Langford, Damien Jose, and Imed Zitouni. Off-policy evaluation for slate recommendation. *arXiv preprint arXiv:1605.04812*, 2016.

[29] Jiaxi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 565–573, 2018.

[30] Stefan Wager and Susan Athey. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 113(523):1228–1242, 2018.

[31] Shu Wan, Chen Zheng, Zhonggen Sun, Mengfan Xu, Xiaoqing Yang, Hongtu Zhu, and Jiecheng Guo. Gcf: Generalized causal forest for heterogeneous treatment effect estimation in online marketplace. *arXiv preprint arXiv:2203.10975*, 2022.

[32] Xuanhui Wang, Michael Bendersky, Donald Metzler, and Marc Najork. Learning to rank with selection bias in personal search. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 115–124, 2016.

[33] Xing Zhao, Qingquan Song, James Caverlee, and Xia Hu. Trailmix: An ensemble recommender system for playlist curation and continuation. In *Proceedings of the ACM Recommender Systems Challenge 2018*, RecSys Challenge '18, New York, NY, USA, 2018. Association for Computing Machinery.

[34] Xing Zhao, Ziwei Zhu, and James Caverlee. Rabbit holes and taste distortion: Distribution-aware recommendation with evolving interests. In *Proceedings of the Web Conference 2021*, WWW '21, pages 888–899, New York, NY, USA, 2021. Association for Computing Machinery.

[35] Ziwei Zhu, Yun He, Xing Zhao, Yin Zhang, Jianling Wang, and James Caverlee. Popularity-opportunity bias in collaborative filtering. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, WSDM '21, pages 85–93, New York, NY, USA, 2021. Association for Computing Machinery.