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
Another approach is called the T-learner [2, 23, 25]. Instead of using base learners learned from first step, and then the final estimation imputed effects for the control and treatment group based on the estimations from these two base learners. Built on the T-learner, control and treatment data. The second model takes the difference of features, so there is no special role in the treatment variable and there interaction between items in session based recommendation. A neural network (NN) based method for treatment effect estimation and reward attribution. 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 be represented by a vector $s = (w_1, w_2, ..., w_k)^T$. Note that $w_i$ 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 $Y(s, x)$, where $x$ is the context including customer preference, browsing history etc. In the settings of the LTR methods, we need to train a model $M(w_i, x)$ to estimate the score for each widget $w_i$. One of the most straightforward approaches is to train $M(w_i, x)$ to approximate $E[Y(s, x)|w_i \in s]$. 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

\[
\Phi_d(w_i, w_0, x) = \sum_{b \in B} P(b)E[\sum_{i=1}^{M} Y(\{b_i\}, x) - Y(\{w_i, w_0\}, x)]
\]

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

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)

\[
\Phi_e(w_i, w_0, x) = Y((b, w_j), x) - Y((w_i, w_0), x).
\]

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

\[
\Phi_v(w_i, w_0, x) = Y((b^k, w_i), x) - Y((b^k, w_0), x).
\]

where \( b^k \) 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 \( Y((b^k, w_i), x) \) and \( Y((b^k, w_0), x) \). Then the VMTA can be calculated by

\[
\tilde{\Phi}_v(w_i, w_0, x) = \tilde{Y}((b^k, w_i), x) - \tilde{Y}((b^k, w_0), x).
\]

where \( \tilde{Y}((b^k, w_i), x) \) and \( \tilde{Y}((b^k, w_0), x) \) 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 \( M_{\text{step}}^{i+1}(b^{i+1}, w_i) \) is trained to estimate the reward \( Y((b^{i+1}, wi), x) \). After training is completed, the VMTA can be estimated by calculating \( M_{\text{step}}^{i+1}(b^{i+1}, w_i) - M_{\text{step}}^{i}(b^{i+1}, x) \). 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 \( M_{\text{slate}}(b^{i+1}, w_i, u^{N-i}) \), where \( u^{N-i} \) is a wild card vector for \( N-i \) widgets. After model training is completed, the VMTA can be estimated by calculating \( M_{\text{slate}}(b^{i+1}, w_i, u^{N-i}) - M_{\text{slate}}(b^{i+1}, u^{N-i+1}) \). 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 SlateNet 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

\[
R = h_x(a)^T x + \sum_i h_v(a_i)^T x + \sigma^T H_{uv}(a) + \sigma^T H_{vx}(a) x
\]

where \( h_x(a) \) is the coefficient for the values of each widget, \( h_v(a_i) \) is the vector of coefficients between the context and the ith widget, \( H_{uv}(a) \) is the matrix of coefficient for interaction between different widgets, \( H_{vx}(a) \) 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 \( a \), which is determined by the position of widgets.
6.1 Distribution of Reward

Due to the Pareto distribution of \( v \), 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).

\[
NMAE = \frac{\sum_i |X_i - \hat{X}_i|}{\sum_i |X_i|}
\]

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

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

Figure 2: Reward estimation and attribution MSE and NMAE

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

Figure 3: Attribution for online dataset


