GMCM: Graph-based Micro-behavior Conversion Model for Post-click Conversion Rate Estimation

Wentian Bao*
wentian.bwt@alibaba-inc.com
Alibaba Group

Hong Wen*
qinggan.wh@alibaba-inc.com
Alibaba Group

Sha Li
Shal2@illinois.edu
University of Illinois
Urbana-Champaign

Xiao-Yang Liu
xl2427@columbia.edu
Columbia University

Quan Lin
tieyi.lq@alibaba-inc.com
Alibaba Group

Keping Yang
shaoyao@alibaba-inc.com
Alibaba Group

ABSTRACT

Purchase-related micro-behaviors, e.g., *favorite, add to cart, read reviews*, etc., provide implicit feedback of users' decision-making process. Such informative feedback can lead to fine-grained post-click conversion rate (CVR) modeling of the buying process. However, most existing works on CVR estimation either neglect these informative feedback, or model them as a sequential pattern with Recurrent Neural Networks. We argue such modeling could be inappropriate since different orders of micro-behaviors may represent similar user buying intention, and micro-behaviors often correlate with each other.

To this end, we propose to represent user micro-behaviors as a Purchase-related Micro-behavior Graph (PMG). Specifically, each node stands for one micro-behavior, and edge weights denote the connection strength. Based on this graph representation, we frame CVR estimation as a graph classification problem over the PMG instances. We propose a novel CVR model, namely, Graph-based Micro-behavior Conversion Model (GMCM), that utilizes Graph Convolutional networks (GCN) to enhance the conventional CVR modeling. In addition, we adopt multi-task learning and inverse propensity weighting to tackle two well-recognized issues in CVR estimation: data sparsity and sample selection bias. Extensive experiments on six large-scale production datasets demonstrate that the proposed methods outperform the state-of-the-art CVR methods under industrial setting.

CCS CONCEPTS

• **Information systems** → **Personalization**; **Information retrieval diversity**; **Recommender systems**.

*Both authors contributed equally to this work.

SIGIR '20, July 25–30, 2020, Virtual Event, China

ACM ISBN 978-1-4503-8016-4/20/07…$15.00
https://doi.org/10.1145/3397271.3401425

KEYWORDS

conversion rate estimation; graph convolutional networks; multi-task learning; recommender systems

ACM Reference Format:
Wentian Bao, Hong Wen, Sha Li, Xiao-Yang Liu, Quan Lin, and Keping Yang. 2020. GMCM: Graph-based Micro-behavior Conversion Model for Post-click Conversion Rate Estimation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), July 25–30, 2020, Virtual Event, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3397271.3401425

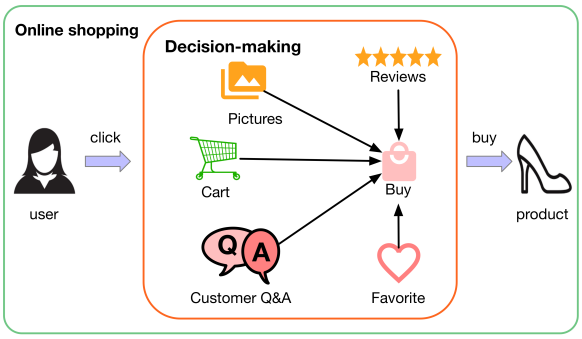

Figure 1: An illustration of the user decision-making process after clicking a product on online shopping app.

1 INTRODUCTION

E-commerce recommender systems help users find products of interest and also help business promote their products to the target costumers. Two tasks are essential for building such systems: click through rate (CTR) prediction and post-click conversion rate (CVR) prediction. CTR measures the transition probability from exposures to clicks, and CVR measures the probability of a product being purchased given it is clicked.

Despite the recent advances in CTR and CVR estimation [18, 32, 34, 35], the industrial-level CVR estimation is still challenging. In practice, the estimated CVR scores may deviate from the true values by large [17, 32], which results in unsatisfactory recommendation. Multiple reasons could lead to the performance degradation, and make the CVR estimation inaccurate. In general, we identify three

practical issues that make the industrial CVR estimation challenging: (i) limited purchase-related feedback; (ii) data sparsity; and (iii) sample selection bias.

The user decision-making process is complicated by nature, while limited purchase-related feedback are utilized to model such buying process. In practice, we notice that conversion labels alone are often too sparse to help the model accurately infer users' buying intention. For example, we have 3.6 billion exposures in our production dataset, while only 27M conversions. Besides, the users' decision-making process is complicated, and not buying a product does not necessarily mean that the user has no interest of it. Intuitively, to better understand such users' buying process, more purchase-related feedback from users should be taken into account. In practice, we find that before making a deal, users tend to have some purchase-related micro-behaviors, e.g., *add to cart, read reviews,* etc. As shown in Figure 1, these micro-behaviors often indicate users' buying intention, and reflect their decision-making process. In our systems, we find more than 70% of the purchases at least relate to one of the micro-behaviors, which emphasizes the importance of incorporating purchase-related feedback with CVR modeling.

However, very few existing works have investigated how to make use of these valuable feedback. To the best of our knowledge, [37] first proposed to utilize user micro-behaviors in e-commerce recommendation, and modeled such behaviors as a sequence. Such an attempt is meaningful but may be inappropriate in some cases. Firstly, different orders of micro-behaviors may represent similar user buying intention. For example, when the user is considering buying a product, the order of reading reviews and costumer Q&A does not make much of a difference. Secondly, micro-behaviors often correlate with each other, while sequential models fail to capture such interconnection. Therefore, we propose to represent the micro-behaviors as a purchase-related micro-behavior graph (PMG). The comparative study of graph-based and sequence-based modeling of micro-behaviors can be seen in Section 5.6.

Except for the lack of purchase-related feedback, conventional CVR modeling is also hindered by two well-recognized issues, i.e., data sparsity and sample selection bias. Data sparsity refers to the fact that training samples for CVR models may not be sufficient for fitting the large amount of model parameters. Conventional CVR models are trained using clicked samples, while clicks are relatively rare compared with exposures. Besides, for industrial CVR estimation, the embedding size of some sparse features could be millions or even billions, which makes the training require more data to converge. In our production datasets, we have 604 million clicked samples vs. over 14 billion embedding parameters (most of them are product sparse embedding). Sample selection bias comes from the fact that conventional CVR models are trained using clicked samples, while the inference is made on all exposed samples. The discrepancy between the distributions of click and exposure space biases the conventional CVR models, and makes the estimated CVR scores deviate from the true values by large [4, 18, 27].

To address the three issues above, we propose to represent the user micro-behaviors as a Purchase-related Micro-behavior Graph (PMG). Specifically, each node in graph stands for a micro-behavior type, e.g., *add to cart, read reviews*, etc, and edge weights represent the connection strength between behaviors. Based on this graph representation, we frame CVR estimation as a graph classification problem over PMG instances, and propose a novel CVR model, namely, Graph-based Micro-behaviors Conversion Model (GMCM), that utilizes the Graph Convolutional Networks (GCN) to take advantage of user micro-behaviors. To alleviate sample selection bias, GMCM adopts inverse propensity weighting, which inversely weights the CVR loss with estimated propensities (i.e., CTR scores), and leverage multi-task learning framework by jointly train CTR and CVR tasks together to address the data sparsity problem.

To summarize, the key contributions of this work are as follows:

- We propose to represent the user micro-behaviors as a Purchase-related Micro-behavior Graph (PMG). Based on this graph representation, we take a novel perspective to frame the CVR estimation as a graph classification problem.
- We propose a novel CVR model, i.e., Graph-based Micro-behavior Conversion Model (GMCM), to take advantage of the PMG, and capture the correlation between user micro-behaviors. Mutli-task learning frame work is leveraged to mitigate the severe data sparsity problem, and inverse propensity weighting technique is adopted, to alleviate the sample selection bias.
- We conduct extensive experiments on six production datasets that collected from the e-commerce platform Taobao. On all datasets, the proposed method GMCM consistently outperforms the baselines and the state-of-the-art models, in terms of several widely-adopted metrics.

2 PRELIMINARIES

In this section, we briefly review the basics of Graph Convolutional Networks (GCN) and inverse propensity weighting (IPW) technique.

2.1 Graph Convolutional Networks

GCN works with a spectral representation of the graphs and has been successfully applied in many applications such as node classification, social networks, and recommender systems [25, 30, 31]. GCN defines the graph convolution operation in Fourier domain by computing the eigen-decomposition of graph Laplacian [36]. Let $x \in \mathbb{R}^N$ with a filter $g_\theta = \text{diag}(\theta)$ parameterized by $\theta \in \mathbb{R}^N$:

$$g_\theta \star x = \mathbf{U} g_\theta(\mathbf{\Lambda}) \mathbf{U}^T x, \tag{1}$$

where $\star$ denotes the graph convolution operation, $\mathbf{U}$ and $\mathbf{\Lambda}$ denote the eigenvectors and eigenvalues' matrix of the normalized graph Laplacian $\mathbf{L} = \mathbf{I}_N - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, and $\mathbf{I}_N$, $\mathbf{D}$, $\mathbf{A}$ are the identity matrix, degree matrix, adjacency matrix, respectively. [13] generalizes the operation to signal $\mathbf{X} \in \mathbb{R}^{N \times C}$ with C input channels and F filter as follow,

$$\mathbf{Z} = \tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}} \mathbf{X} \mathbf{\Theta}, \tag{2}$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$, $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$, $\mathbf{\Theta} \in \mathbb{R}^{C \times F}$ denotes the matrix of filter parameters and $\mathbf{Z} \in \mathbf{R}^{N \times F}$ is the convolved signal matrix.

2.2 Inverse Propensity Weighting

Inverse propensity weighting (IPW) inversely weights the prediction error for each observed sample with the propensity of observing that sample [22]. Let $\mathcal{U} = (u_1, u_2, u_3, ..., u_N)$ be a set of users

and $\mathcal{I} = (i_1, i_2, i_3, ..., i_M)$ be a set of items. $\mathcal{D} = \mathcal{U} \times \mathcal{I}$ denotes the user-item pairs. Let $\mathbf{R} \in \mathbb{R}^{N \times M}$ be the true rating matrix and $\hat{\mathbf{R}} \in \mathbb{R}^{N \times M}$ be the prediction matrix where each entry $\hat{r}_{u,i}$ is a predicted rating computed by the prediction model. If the true rating matrix $\mathbf{R}$ is fully observed, the prediction inaccuracy, $\mathcal{P}$, over all user-item pairs can be measured as follow,

$$\mathcal{P} = \mathcal{P}(\mathbf{R}, \hat{\mathbf{R}}) = \frac{1}{|\mathcal{D}|} \sum_{(u,i) \in \mathcal{D}} e(r_{u,i}, \hat{r}_{u,i}), \tag{3}$$

where $e(r_{u,i}, \hat{r}_{u,i})$ denotes the prediction error. However, in practice, the true rating matrix $\mathbf{R}$ could only be partially observed. Let $\mathbf{O} \in \{0, 1\}^{N \times M}$ be the indicator matrix where each entry $o_{u,i}$ is an observation indicator, and let $\mathbf{R}^{obs}$ and $\mathbf{R}^{mis}$ be the set of observed and missing entries in the true rating matrix $\mathbf{R}$. The estimated prediction inaccuracy could be calculated as follow,

$$\begin{aligned} \mathcal{E}^{obs} &= \mathcal{E}(\mathbf{R}^{obs}, \hat{\mathbf{R}}) \\ &= \frac{1}{|\mathbf{O}|} \sum_{(u,i) \in \mathcal{D}} o_{u,i} e(r_{u,i}, \hat{r}_{u,i}), \end{aligned} \tag{4}$$

The IPW-based methods use inverse propensity to weight the prediction error, which leads to,

$$\begin{aligned} \mathcal{E}^{IPW} &= \mathcal{E}(\mathbf{R}, \hat{\mathbf{R}}) \\ &= \frac{1}{|\mathcal{D}|} \sum_{(u,i) \in \mathcal{D}} \frac{o_{u,i} e(r_{u,i}, \hat{r}_{u,i})}{\hat{p}_{u,i}}. \end{aligned} \tag{5}$$

It can be proved that given the propensity is accurate, the IPW-based estimations are unbiased, i.e., $|\mathbb{E}_{\mathbf{O}}[\mathcal{E}] - \mathcal{P}| = 0$. In this work, we adopt IPW technique to address selection bias issue in CVR estimation. More details are discussed in Section 4.

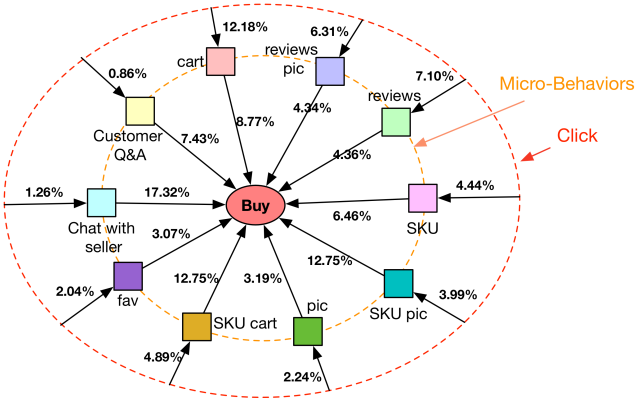

Figure 2: Transition probability among clicks, micro-behaviors and purchases. the outer circle denotes the click event, and inner circle denotes 10 types of micro-behaviors. The arrow stands for transition probability. Probabilities are calculated based on over 400 million transaction logs in our production systems.

3 PURCHASE-RELATED MICRO-BEHAVIOR GRAPH

In this section, we define the purchase-related micro-behavior graph (PMG) over user micro-behaviors after clicking a product. we start with discussion of what purchase-related micro-behaviors are, and their high relevance with the user purchase decision. Then we give a formal definition of PMG. Lastly, we discuss how to frame CVR estimations as a graph classification problem over PMG instances.

Purchase-related micro-behaviors denote the users' purchase-related actions after clicking a product, e.g., *add to cart, read reviews*, etc. Such implicit feedback provides us with fine-grained understanding of the user decision-making process, and helps CVR models better infer the users' buying preference. In Figure 2, we show 10 kinds of micro-behaviors and their transition probability to the final purchase. Figure 2 reflects the strong connection between micro-behaviors and the user purchase decision, and highlights the importance of micro-behaviors modeling in the CVR estimation.

Then we formally define the purchase-related micro-behavior graph (PMG). Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denotes the undirected graph PMG with N nodes $v_i \in \mathcal{V}$, M edges $(v_i, v_j) \in \mathcal{E}$. Define the adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, and degree matrix $\mathbf{D}$, where each entry A_{ij} stands for the edge weight of (v_i, v_j), and each entry $\mathbf{D}_{ii} = \sum_j A_{ij}$. Let $\mathcal{M}$ be the set of all micro-behavior events collected by the log engine. Then we have $\mathcal{V} \subseteq \mathcal{M}$, indicating that every node in PMG stands for one type of micro-behavior events. To determine the edge weight, let concurrency matrix of micro-behaviors be $\mathbf{C} \in \mathbb{R}^{N \times N}$ and concurrent probability matrix be $\mathbf{P} \in \mathbb{R}^{N \times N}$, where each entry c_{ij} = #(concurrency of micro-behavior i and j in logs), and each entry $p_{ij} = \frac{c_{ij}}{\sum_j c_{ij}}$. Then for each edge $(v_i, v_j) \in \mathcal{E}$, $p_{ij} \in \mathbf{P}$ denotes the initial weight of the edge (v_i, v_j). To further enable the model adjust edge weights during the training process, we define a learnable edge bias matrix $\mathbf{B} \in \mathbb{R}^{N \times N}$, and let $\mathbf{A} = \mathbf{P} + \mathbf{B}$. Figure 3 shows an instance of PMG. Note that the buy button in Figure 3 only denotes the micro-behavior of clicking the buy button, which does not necessarily lead to the final conversion. SKU stands for stock keeping unit, which represents the different type of a product, e.g., *size, color*, etc.

Next we frame CVR estimation as a graph classification problem over the PMG instance. Let $\mathcal{Y}^{\text{CVR}}$ and $\mathcal{Y}^{\text{node}}$ be the label set of conversions and micro-behaviors in PMG. Suppose the input feature matrix is $\mathbf{X}$, and a PMG instance is $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Then the CVR prediction matrix over the PMG instance is defined as,

$$\hat{\mathbf{R}} = f(\mathcal{E}, \mathcal{V}, \mathbf{X}), \tag{6}$$

where f is a differentiable predicting function, $\hat{\mathbf{R}} = [\hat{\mathbf{R}}^{\text{node}} | \hat{\mathbf{R}}^{\text{CVR}}]$, and $|$ denotes the matrix concatenation. Given the prediction matrix, the loss of estimated node scores and CVR are,

$$\begin{aligned} L^{\text{node}} &= l(\hat{\mathbf{R}}^{\text{node}}, \mathcal{Y}^{\text{node}}) \\ L^{\text{CVR}} &= l(\hat{\mathbf{R}}^{\text{CVR}}, \mathcal{Y}^{\text{CVR}}), \end{aligned} \tag{7}$$

where l denotes the loss function, and the loss of CVR estimation over the PMG instance is then defined as,

$$L^{\text{PMG}} = h(L^{\text{node}}, L^{\text{CVR}}), \tag{8}$$

where h is the merging function of two losses.

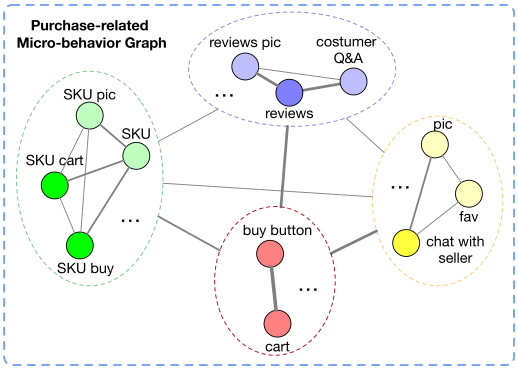

Figure 3: Illustration of the Purchase-related Micro-behavior Graph (PMG). Each node denotes one micro-behavior type, and thickness of edge represents the connection strength. SKU stands for stock keeping unit. Note that the clustering of nodes is only for drawing convenience.

4 GRAPH-BASED MICRO-BEHAVIORS CONVERSION MODEL

In this section, we propose the graph-based micro-behavior conversion model (GMCM). We firstly present the motivation, then discuss the key components of GMCM in detail. Figure 4 depicts the architecture of the proposed method.

4.1 Motivation

GMCM is devised to address three practical issues in conventional CVR modeling, i.e., limited purchase-related feedback, data sparsity and sample selection bias.

To address the first issue, GMCM takes advantage of user micro-behaviors by modeling them over PMG instances, and adopts GCN to capture their correlations. In practice, we find that these micro-behaviors are highly-related to the user purchase decision, thus may serve as a clue of the user buying intention. Besides, we observe that micro-behaviors often correlate with each other, and different orders of micro-behaviors may indicates similar user buying intention. Therefore, we represent the micro-behaviors as a graph rather than a sequence. To address data sparsity and sample selection bias issues, we leverage mutli-task learning framework, and adopt inverse propensity weighting technique. Specifically, GMCM trains CTR and CVR task together, and shares embedding layer between the two tasks. Therefore, the large amount of ID embedding parameters could be trained with both exposed and clicked samples. Besides, GMCM inversely weights the CVR prediction error with estimated propensities (i.e., CTR scores), to alleviate sample selection bias. Note that IPW-based methods are theoretically unbiased contingent on the propensities are accurately estimated. In practice, such constraint is often too restricted. However, existing works [32, 33] show that even if the propensities are accurately estimated, IPW-based method still can mitigate sample selection bias issue to some degree.

4.2 Multi-task Learning Module

GMCM chains CTR and CVR tasks together and trains them in parallel. Concretely, the two tasks have independent input layer, and a shared embedding layer. Let the shared embedding matrix be $\mathbf{W}_e \in \mathbb{R}^{d_h \times d_e}$, where d_h denotes the hush bucket size and d_e denotes the embedding dimension. Define the ID feature matrices $\mathbf{X}_s^{\text{CTR}} \in \mathbb{R}^{B \times F_s^{\text{CTR}}}$, $\mathbf{X}_s^{\text{CVR}} \in \mathbb{R}^{B \times F_s^{\text{CVR}}}$, where B is the mini batch size, F_s is the input dimension of ID features. Then the embedded ID features are represented as,

$$\begin{aligned} \mathbf{X}_e^{\text{CTR}} &= f_h(X_s^{\text{CTR}}, W_e) \\ \mathbf{X}_e^{\text{CVR}} &= f_h(X_s^{\text{CVR}}, W_e) \end{aligned} \tag{9}$$

where $X_e \in \mathbb{R}^{B \times F_e}$, f_h denotes the hash function, and F_e is the embedding dimension of ID feature matrix X_s. Then define dense feature matrices $\mathbf{X}_d^{\text{CTR}} \in \mathbb{R}^{B \times F_d^{\text{CTR}}}$, $\mathbf{X}_d^{\text{CVR}} \in \mathbb{R}^{B \times F_d^{\text{CVR}}}$, where F_d denotes the input dimension of all dense features. Then we have the concatenated feature representation,

$$\begin{aligned} X_{\text{concat}}^{\text{CTR}} &= [X_d^{\text{CTR}} | \mathbf{X}_e^{\text{CTR}}] \\ X_{\text{concat}}^{\text{CVR}} &= [X_d^{\text{CVR}} | \mathbf{X}_e^{\text{CVR}}] \end{aligned} \tag{10}$$

where $|$ denotes the matrix concatenation, and $X_{\text{concat}} \in \mathbb{R}^{B \times (F_d + F_e)}$ is concatenated input features after the shared embedding layer.

The remaining part of CTR networks is several layers of MLP with the leaky Relu activation function. Suppose that the logits of CTR output layer is $\mathbf{o} \in \mathbb{R}^B$, then for each sample the predicted CTR is given by,

$$\hat{r_i}^{\text{CTR}} = \text{sigmoid}(\mathbf{o_i}), \tag{11}$$

and the loss of CTR networks is,

$$L^{\text{CTR}} = \frac{1}{B} \sum_{i=1}^{B} l(y_i^{\text{CTR}}, r_i^{\text{CTR}}), \tag{12}$$

where $y_i \in \mathcal{Y}^{\text{CTR}}$ is the CTR label, and l stands for cross entropy loss.

4.3 Graph-based CVR Networks

The CVR networks consist of multiple MLP layers, a node embedding layer, and GCN.

4.3.1 MLP Layers. Let $H_k \in \mathbb{R}^{B \times h_k}$ and h_k denote the hidden representation and the dimension of k^{th} layer. Then we can define a single layer of MLP as,

$$\mathbf{H}_k^{\text{MLP}} = \sigma(\mathbf{H}_{k-1}^{\text{MLP}} \Theta_k^{\text{MLP}}) \tag{13}$$

where σ is leaky relu activation function, and Θ_k is parameters of k^{th} layer. By stacking multiple MLP layers, GMCM produces common input representation for the node embedding layer.

4.3.2 Node Embedding Layer. GMCM makes use of user micro-behaviors to enhance conventional CVR modeling. For each node v_i, a single layer of MLP is utilized to generate the node embedding. Suppose that there are N nodes in PMG, and $\mathbf{H}^{\text{MLP}}$ denotes the output of final layer of MLP with hidden dimention h_K. For each node v_i, the node embedding is formulated as,

$$\mathbf{e}_i = \sigma(\mathbf{H}^{\text{MLP}} \Theta_i^{\text{node-MLP}}), \mathbf{e}_i \in \mathbb{R}^{d_{\text{node}}} \tag{14}$$

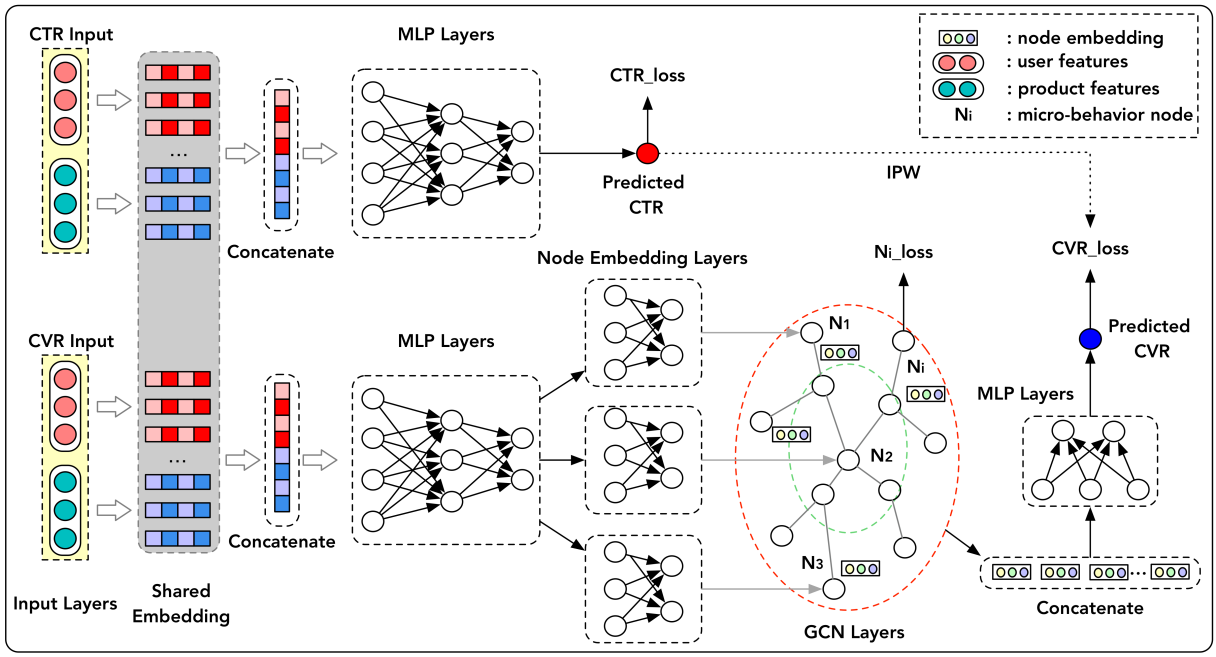

Figure 4: The architecture of Graph-based Micro-behaviors Conversion Model (GMCM).

where d_{node} is the embedding dimension, and $\Theta_i^{\text{node-MLP}} \in \mathbb{R}^{h_K \times d_{\text{node}}}$ denotes parameters of node embedding layer for node v_i.

4.3.3 Graph Convolutional Networks. GMCM exploits GCN to convolve the node embedding and capture their correlations. Define an instance of undirected graph PMG as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with N nodes $v_i \in \mathcal{V}$ and M edges $(v_i, v_j) \in \mathcal{E}$. Define the adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, and degree matrix $\mathbf{D}$, where each entry A_{ij} denote the edge weight of (v_i, v_j), and each entry $\mathbf{D}_{ii} = \sum_j A_{ij}$. Recall that by introducing the concurrent probability matrix $\mathbf{P}$ and edge bias matrix $\mathbf{B}$, the adjacency matrix $\mathbf{A} = \mathbf{P} + \mathbf{B}$. thus a single layer of GCN can be defined as,

$$\mathbf{H}_l^{\text{GCN}} = \sigma(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}_{l-1}^{\text{GCN}}\Theta_l^{\text{GCN}}), \tag{15}$$

where $\mathbf{H}_l^{\text{GCN}} \in \mathbb{R}^{B \times N \times h_l}$ and $\Theta_l^{\text{GCN}} \in \mathbb{R}^{h_{l-1} \times h_l}$ denotes the hidden representation and parameters of the l^{th} GCN layer, $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$, and $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$.

For each node v_i, GMCM outputs a predicted score $\hat{r}_i^{\text{node}}$, indicating the probability of the user having that micro-behavior after clicking the product. Besides, the hidden representations of all nodes are aggreated to generate the predicted CVR $\hat{r}^{\text{CVR}}$. Formally, suppose that the convolved feature tensor of the last GCN layer is $\mathbf{H}^{\text{GCN}} \in \mathbb{R}^{B \times N \times h^{\text{GCN}}}$. For each sample, let $\mathbf{h}_i^{\text{GCN}} \in \mathbb{R}^{h^{\text{GCN}}}$ be the representation of node v_i. Then the predicted micro-behavior score $\hat{r}_i^{\text{node}}$ is calculated as,

$$\hat{r}_i^{\text{node}} = \text{sigmoid}(\mathbf{h}_i^{\text{GCN}}\Theta_i^{\text{GCN-logit}}), \tag{16}$$

where $\Theta_i^{\text{GCN-logit}} \in \mathbb{R}^{h^{\text{GCN}}}$ is the parameters of logit layer for node v_i. And for predicted CVR $\hat{r}^{\text{CVR}}$ we have,

$$\hat{r}^{\text{CVR}} = \text{sigmoid}(f_{\text{output}}(\mathbf{H}^{\text{GCN}})\Theta_{\text{CVR}}^{\text{GCN-logit}}), \tag{17}$$

where $\Theta_{\text{CVR}}^{\text{GCN-logit}} \in \mathbb{R}^{h^{\text{GCN}}}$ is the parameters of CVR logit layer, and f_{output} denotes the output function, e.g., *concatenation, sum pooling, mean pooling*, etc.

4.3.4 Loss Layer. Given the estimated CVR scores $\hat{\mathbf{r}}^{\text{CVR}} \in \mathbb{R}^B$, micro-behaviors scores $\hat{\mathbf{R}}^{\text{node}} \in \mathbb{R}^{B \times N}$, and estimated CTR scores $\hat{\mathbf{r}}^{\text{CTR}} \in \mathbb{R}^B$, the loss of graph-based CVR networks over the PMG instance is defined as,

$$L^{\text{PMG}} = \frac{1}{B}\sum_{i=1}^{B}\frac{1}{\hat{r}_i^{\text{CTR}}}(l(y_i^{\text{CVR}}, \hat{r}_i^{\text{CVR}}) + \frac{1}{N}\sum_{j=1}^{N} l(y_{ij}^{\text{node}}, \hat{r}_{ij}^{\text{node}})), \tag{18}$$

where l denotes the cross entropy loss, $y_i^{\text{CVR}} \in \mathcal{Y}^{\text{CVR}}$ and $y_{ij}^{\text{node}} \in \mathcal{Y}^{\text{node}}$ denote the label of CVR and micro-behaviors. Note that the CVR loss is weighted by inverse predicted CTR scores, to alleviate sample selection bias. In practice, the inverse CTR scores may lead to numerical stability problem. Thus we self-normalize the predicted CTR scores when weighting CVR loss [24]. Lastly, the loss of GMCM can be defined as,

$$L^{\text{GMCM}} = L^{\text{CTR}} + L^{\text{PMG}}. \tag{19}$$

5 EXPERIMENTS

In this section, we evaluate the performance of GMCM with six production datasets collected from Mobile Taobao, the leading e-commerce platform in China. The experiments are intended to answer the following questions:

- **Q1**: Does GMCM outperform other state-of-art CVR estimation methods?
- **Q2**: Does GCN help to better utilize the micro-behaviors feedback?

Table 1: Statistics of six production datasets.

Dataset	total	# clicks	# buys	# users	# products	# training	# testing
SM-A	1.3B	215M	10M	41M	99M	1.1B	200M
SM-B	2.5B	412M	19M	63M	124M	2.2B	200M
SM-C	3.6B	604M	27M	80M	143M	3.4B	200M
GUL-A	390M	77M	7M	29M	44M	330M	59M
GUL-B	744M	147M	13M	47M	58M	685M	59M
GUL-C	1.1B	220M	20M	63M	68M	1B	59M

- **Q3**: How is the performance of GMCM affected by hyper-parameters of GCN?

5.1 Datasets

We collect six production datasets from the Mobile Taobao. Table 1 summarizes the statistics of all 6 subsets. All datasets contain 258 features, including 230 dense features, e.g., product price, click times, conversion times, etc., and 28 sparse features, e.g., user age, user gender, etc. The datasets contain 3-week consecutive logs (2020-02-15 to 2020-03-07) from two different recommendation applications: Guess You Like inShop(GUL), and Shopping More inShop(SM). For SM datasets, there are totally 3.6 billion data samples, including 604 million clicks, 27 million conversions, and 80 million users. For GUL datasets, there are totally 1.1 billion data samples, including 220 million clicks, 20 million conversions and 63 million users. We further split these 3-week logs of SM and GUL into 6 subsets:

- **Set SM-A** contains data from 2020-03-01 to 2020-03-06 as training set, and data in 2020-03-07 as testing set. Set SM-A contains approximately 30% of the SM data samples.
- **Set SM-B** contains data from 2020-02-23 to 2020-03-06 as training set, and data in 2020-03-07 as testing set. Set SM-B contains approximately 70% of the SM data samples.
- **Set SM-C** contains data from 2020-02-15 to 2020-03-06 as training set, and data in 2020-03-07 as testing set. Set SM-C contains all SM data samples.
- **Set GUL-A** contains data from 2020-03-01 to 2020-03-06 as training set, and data in 2020-03-07 as testing set. Set GUL-A contains approximately 30% of the GUL data samples.
- **Set GUL-B** contains data from 2020-02-23 to 2020-03-06 as training set, and data in 2020-03-07 as testing set. Set GUL-B contains approximately 70% of the GUL data samples.
- **Set GUL-C** contains data from 2020-02-15 to 2020-03-06 as training set, and data in 2020-03-07 as testing set. Set GUL-C contains all GUL data samples.

5.2 Compared Methods

To evaluate the performance of proposed method, We compare GMCM with the following models. Note that all compared models are based on multi-task learning framework and share the embedding layer. Hence, the data sparsity issue is mitigated to some degree.

- **Base** is a co-trained CTR and CVR networks. It has separate input layer and a shared embedding layer.
- **Division** [18] estimates CVR by CTCVR = CTR × CVR. Different from [18], we adapt Division in our experiments with co-trained CTR and CVR networks and shared embedding layer, rather than individually trained networks.
- **ESMM** [18] reduces the CVR estimation to two auxiliary tasks, i.e., CTR task and CTCVR task. By successfully adopting the multi-task learning and entire space modelling, ESMM could tackle both data sparsity and sample selection bias issues, thus is deemed as the sate-of-the-art CVR model in practice.
- **ResNet** [7] Residual Neural Networks (ResNet) introduces identity shortcut connection that allows one or more layers being skipped. ResNet has achieved great success in areas like computer vision. We adapt ResNet for CVR estimation by co-training CTR and CVR tasks with ResNet.
- **DCN** [26] Deep and Cross Networks (DCN) introduces a novel cross networks that is more efficient in learning certain bounded-degree feature interactions. We adapt DCN for CVR estimation by co-training CTR and CVR tasks with DCN.

5.3 Evaluation Metrics

We adopt three widely used metrics in our experiments: AUC, GAUC and MSE. Generally, AUC and GAUC indicates the ranking performance, and MSE reflects the prediction inaccuracy.

In CTR and CVR estimations, area under receiver operator curve (AUC) is a widely used metric [5]. One interpretation of AUC in the context of ranking system is that it denotes the probability of ranking a random positive sample higher than a negative sample. Meanwhile, we also adopt Group AUC (GAUC) in our assessments [35]. GAUC extends AUC by calculating the weighted average of AUC grouped by a certain property, e.g., user, page, etc. GAUC can be calcuated as,

$$\text{GAUC} = \frac{\sum_i w_i \times \text{AUC}_i}{\sum_i w_i}, \tag{20}$$

where w_i is the weight of certain group. GAUC is commonly recognized as a more indicative metric for ranking performance [35]. Besides, we utilize MSE to measure the prediction inaccuracy. MSE is defined as,

$$\text{MSE} = \frac{1}{|O|} \sum_{(u,i) \in O} (r_{u,i} - \hat{r}_{u,i})^2 \tag{21}$$

where O is the observed user-item rating pairs.

5.4 Hyper-parameter Settings

5.4.1 Common Settings. all competitors in the experiments use batch size of 2048, Adam Optimizer [12] with $\beta_1 = 0.9$,$\beta_2 = 0.999$. Learning rate is tuned and set to be 0.0001, and same embedding dimension of sparse features is used for all competitors. All models except for GMCM shares the network architecture of $(512, 256, 128, 64)$ for both CTR and CVR tasks. To stabilize the training process, all models use batch normalization [8].

5.4.2 Individual Settings. For DCN, five cross layers are used besides the MLP, following the setting in [26]. For GMCM, we set the architecture of CTR networks as $(512, 256, 128, 64)$, same with other baselines. For CVR networks, we adopt a three layers of MLP with the shape of $(512, 256, 128)$, and a node embedding layer with dimension of 64. We search for the number of GCN layers in $(2, 3, 4)$, and the dimension of hidden units in $(16, 32, 64, 128, 160)$. The best performance is reported with 2 GCN layers and 64 hidden units for each layer.

Table 2: Model performance evaluated by AUC/GAUC on six production datasets. The best scores are in boldface, and the second best scores are underlined. RI indicates the relative improvement of GMCM compared with baseline models in permillage. Note that to save space, lower case for AUC, GAUC, CVR, CTCVR are used in this table.

Model	SM-A(1.3B)				SM-B(2.5B)				SM-C(3.6B)			
	cvr auc	ctcvr auc	ctcvr gauc	RI	cvr auc	ctcvr auc	ctcvr gauc	RI	cvr auc	ctcvr auc	ctcvr gauc	RI
Base	0.8151	0.8278	0.7032	9.71‰	0.8249	0.8357	0.7063	7.28‰	0.8255	0.8374	0.7103	8.83‰
ESMM	0.8163	0.8283	0.7051	8.11‰	0.8263	0.8359	0.7077	5.94‰	0.8305	0.8412	0.7119	4.53‰
Division	0.8155	0.8271	0.6987	12.05‰	0.8208	0.8317	0.7072	10.15‰	0.8243	0.8344	0.7117	9.91‰
ResNet	0.8146	0.8230	0.6959	15.46‰	0.8233	0.8322	0.7002	12.27‰	0.8272	0.8364	0.7037	11.69‰
DCN	0.8169	0.8304	0.7021	8.47‰	0.8250	0.8349	0.7047	8.29‰	0.8313	0.8420	0.7115	4.08‰
GMCM	**0.8183**	**0.8332**	**0.7164**	-	**0.8300**	**0.8417**	**0.7123**	-	**0.8340**	**0.8446**	**0.7157**	-
Model	GUL-A(390M)				GUL-B(744M)				GUL-C(1.1B)			
	cvr auc	ctcvr auc	ctcvr gauc	RI	cvr auc	ctcvr auc	ctcvr gauc	RI	cvr auc	ctcvr auc	ctcvr gauc	RI
Base	0.7982	0.8335	0.7620	9.98‰	0.8059	0.8447	0.7746	8.83‰	0.8110	0.8435	0.7723	10.37‰
ESMM	0.7973	0.8298	0.7634	11.22‰	0.8066	0.8421	0.7728	10.32‰	0.8080	0.8441	0.7767	9.46‰
Division	0.7885	0.8268	0.7591	18.08‰	0.8030	0.8405	0.7692	14.04‰	0.8047	0.8423	0.7686	15.07‰
ResNet	0.7934	0.8321	0.7609	13.02‰	0.8050	0.8402	0.7683	13.76‰	0.8112	0.8491	0.7738	7.39‰
DCN	0.7969	0.8337	0.7639	9.59‰	0.8080	0.8451	0.7732	8.41‰	0.8088	0.8433	0.7721	11.47‰
GMCM	**0.8067**	**0.8422**	**0.7687**	-	**0.8180**	**0.8521**	**0.7767**	-	**0.8183**	**0.8529**	**0.7807**	-

Table 3: Model performance evaluated by MSE on six production datasets. The best scores are in boldface, and the second best scores are underlined. RI indicates the relative improvement of GMCM compared with baseline models in permillage.

Model	SM-A(1.3B)			SM-B(2.5B)			SM-C(3.6B)		
	CVR MSE	CTCVR MSE	RI	CVR MSE	CTCVR MSE	RI	CVR MSE	CTCVR MSE	RI
Base	0.04496	0.008125	11.02‰	0.04423	0.008097	4.25‰	0.04395	0.008094	7.89‰
ESMM	0.04513	0.008143	13.98‰	0.04417	0.008103	3.94‰	0.04394	0.008090	7.485‰
Division	0.04470	0.008111	7.32‰	0.04454	0.008156	11.35‰	0.04400	0.008103	8.96‰
ResNet	0.04482	0.008141	10.49‰	0.04431	0.008108	5.84‰	0.04381	0.008081	5.37‰
DCN	0.04434	0.008101	2.65‰	0.04444	0.008102	6.88‰	0.04345	**0.008057**	-0.12‰
GMCM	**0.04411**	**0.008099**	-	**0.04395**	**0.008079**	-	**0.04340**	0.008068	-
Model	GUL-A(390M)			GUL-B(744M)			GUL-C(1.1B)		
	CVR MSE	CTCVR MSE	RI	CVR MSE	CTCVR MSE	RI	CVR MSE	CTCVR MSE	RI
Base	0.0778	0.01739	13.41‰	0.0770	0.01731	21.98‰	0.0758	0.01717	9.73‰
ESMM	0.0774	0.01745	12.17‰	0.0774	0.01737	26.44‰	0.0762	0.01721	13.06‰
Division	0.0811	0.01764	40.51‰	0.0787	0.01745	36.30‰	0.0783	0.01740	32.08‰
ResNet	0.0783	0.01749	18.97‰	0.0765	0.01727	17.73‰	**0.0745**	**0.01706**	-2.16‰
DCN	0.0782	0.01745	17.84‰	0.0768	0.01732	21.11‰	0.0764	0.01739	19.89‰
GMCM	**0.0763**	**0.01726**	-	**0.0746**	**0.01708**	-	0.0748	0.01708	-

5.5 Analysis on Overall Performance (Q1)

In this section, we analyze the overall performance of the proposed method and other compared models. The experiment results are summarized in table 2 and table 3. Based on the results, we have following observations:

- GMCM consistently outperforms other baseline models across all six datasets in term of AUC/GAUC, and achieves the lowest MSE in most of the experiments. Recall that GMCM is devised to better tackle three practical issues in conventional CVR estimations, i.e., limited purchase-related feedback, data sparsity and sample selection bias. In the experiments, all models are based on multi-task learning framework and share embedding layers between CTR and CVR tasks, thus the data sparsity issue is mitigated to some degree. Compared with baseline models, however, GMCM makes use of users' micro-behaviors feedback, and leverages GCN to better model the user decision-making process. Besides, the IPW technique is adopted to alleviate sample selection bias. Therefore, we believe the performance boost of GMCM owing to the better modeling of micro-behaviors with GCN, and also the adoption of IPW method.
- ESMM is a strong competitor, and deemed as the state-of-the-art CVR model in practice. ESMM proposed a novel method of entire space modeling to mitigate the sample selection bias issue. For GMCM, we adopt the IPW technique, which is proved theoretically unbiased, and empirically tested better than ESMM in [32].
- DCN is another strong baseline and achieves decent performance in most of the experiments. For dataset SM-C, DCN even outperforms GMCM in term of CVR MSE, and has a pretty close CTCVR MSE to GMCM. Recall that DCN utilizes

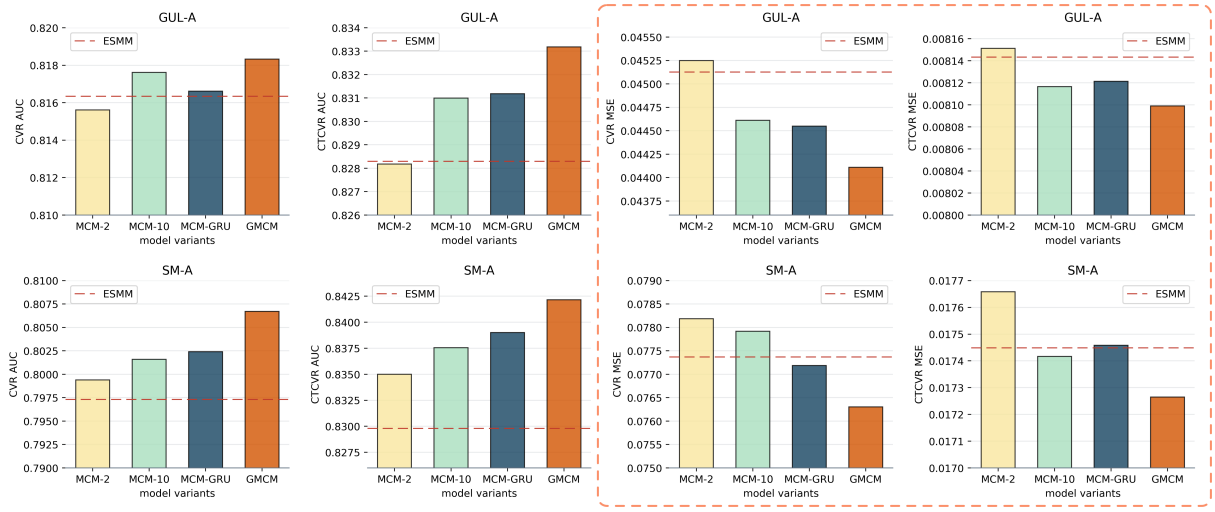

Figure 5: Ablation study of GMCM and two variant models on GUL-A and SM-A. The red dashed line denotes the performance of ESMM. Note that figures in the orange dashed box are evaluated by MSE, which means the best performance is represented by the shortest bar.

cross networks to automatically capture the feature interactions and enhance the representation ability of vanilla MLP. Generally, the good performance of DCN in our experiments indicates that changing vanilla MLP with other types of networks such as DCN, may further boost the performance of a co-trained CVR networks.

- We observe that in nearly all experiments, the model performance is improved as more training data is used. Moreover, the performance boost achieved by using more data is sometimes greater than that of using different models. We reason that in production datasets hundreds of features are used, and the samples contain millions of users and products. Therefore, the training generally requires more data to converge. Such observation suggests that in industry-level CVR estimation, the size of training data plays a key role in enhancing the model performance, and collecting more training data is as important as choosing the right model.

5.6 Ablation Study (Q2)

To further verify the benefits of utilizing user micro-behaviors, and the effectiveness of graph-based modeling, we conduct ablation experiments to compare GMCM with the following variant models. Note that MCM stands for Micro-behaviors Conversion Model, a simple variant of GMCM without using the GCN module. In MCM, all the node embedding is simply concatenated to predict the CVR, hence neither sequential information nor the correlation of nodes are taken into account.

- MCM-2: Only the two most easily accessible micro-behaviors are used, i.e., add to cart and add to favorite list.
- MCM-10: 10 micro-behaviors are used, same as GMCM.
- MCM-GRU: It replaces GCN with a sequence model. Specifically, after the node embedding layer, it adopts GRU to capture the sequential information of micro-behaviors, and uses hidden representation of the last GRU layer to predict CVR. Note that MCM-GRU is used to compare graph-based and sequence-based modeling of micro-behaviors.

Figure 5 shows the performance of GMCM and different variants. the state-of-the-art CVR model ESMM is also included as a baseline. Based on the experiment results, we have the following observations:

- Compare MCM-10 with MCM-2, we find that the model performance in terms of all three metrics is improved as more micro-behaviors are taken into account. MCM-2 only utilizes the feedback of adding to cart and favorite list, while MCM-10 incorporates more feedback such as reading reviews, zooming in pictures, etc. The results of MCM-10 and MCM-2 indicates that providing the model with more purchase-related feedback from users will generally boost the model performance.
- GMCM outperforms MCM-10 and MCM-GRU in all the experiments. Compared with MCM-10 and MCM-GRU, GMCM adopts GCN to model the micro-behaviors and capture their correlation. In practice, we find that different orders of micro-behaviors may represent similar user buying intention. Besides, the behaviors often correlate with each other. For example, the user who reads more reviews has a high probability also reads the customer Q&A. Hence, the micro-behaviors are better represented as a graph rather than a sequence.

5.7 Hyper-parameter Sensitivity (Q3)

We conduct experiments on how the performance of GMCM is affected by two important hyper-parameters of GCN, i.e., the number of GCN layers and hidden units. We search for the number of GCN layers in (2, 3, 4), and the number of hidden units in (16, 32, 64, 128, 160). The experiment results are summarized in Figure 6. From the figures we could easily observe that GMCM has the best performance when GCN has two layers, and hidden units number is between

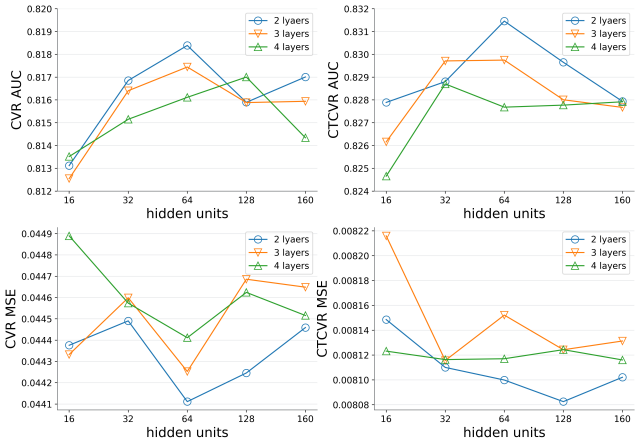

Figure 6: Experiment results of hyper-parameter sensitivity on SM-A. Note that the lower two figures are evaluated by MSE, which indicates the bottom plots have the best performance.

64 and 128. We could clearly see the performance drops when the GCN layer number increases, and when too many or too few hidden units are used. We reason that since the edge weights are learned during the training phase, too many GCN layers may make the representation of each node aggregate too much information from the uncorrelated nodes, hence introducing noise during training.

6 RELATED WORK

GMCM is devised to make use of purchase-related micro-behaviors, and tackle the practical issues of data sparsity and sample selection bias in CVR estimation. In what follows, we briefly review the existing works that most relates to these aspects.

6.1 General Approach to CVR Estimation

Due to the inherent similarity, many CTR models are adopted for CVR estimation in practice, such as logistic regression [11, 21], factorization machines (FM) [9, 10, 20], deep neural networks (DNN) [1–3, 6], methods that combines DNN with FM [15, 16], and recently, sequence models that utilize user historical behaviors [34, 35]. Few literatures directly study on the CVR estimation task. Lee *et al.* [14] proposed a hierarchical CVR model with separate binomial distributions for different feature levels and integrated these individual estimators via logistic regression. Shan *et al.* [23] proposed combined regression and triplet-wise ranking method (CRT) to jointly consider regression loss and triplet-wise ranking loss for CVR estimation. Wen *et al.* [29] proposed a deep cascade tree structure to ensemble Gradient Boost Decision Tree (GBDT) and improve the feature representation ability.

6.2 Data Sparsity and Sample Selctioin Bias

Two practical issues, i.e., data sparsity and sample selection bias, may degrade CVR estimators under industrial setting. Several studies have been carried out to tackle these challenges [18, 19, 28, 32, 33]. For example, Gary *et al.* [28] proposed the oversampling method that copied rare class samples to mitigate the data sparsity issue. Pan *et al.* [19] proposed All Missing As Negtive(AMAN) which applied random sampling on un-clicked samples. Zhang *et al.* [33] proposed to achieve unbiased CTR estimation via reject sampling on the underlying distribution of the observation. Recently, Ma *et al.* proposed Entire Space Multi-task Model (ESMM) [18], which aimed for addressing both data sparsity and sample selection bias issues. Specifically, ESMM exploited the sequential pattern of online shopping, "exposure → click → purchase", and reduced the CVR estimation to two auxiliary tasks, i.e., CTR estimation and click-through and conversion rate (CTCVR) estimation. By sharing the embedding layer between tasks, data sparsity issue is mitigated. It also alleviated the sample selection bias via entire space modeling. Zhang *et al.* [32] provided a causal perspective of CVR estimation. Two causal estimators, i.e., Multi-IPW and Multi-DR were proposed to tackle the data sparsity and sample selection bias in concert.

6.3 Micro-behaviors Based Methods

Very few existing works study on how to utilize user micro-behaviors in recommender systems. To our knowledge, [37] is the very first attempt to make use of such implicit feedback, and RNN-based method was adopted to model micro-behaviors. However, we argue that the sequence-based modeling could be inappropriate in some cases. Firstly, different orders of micro-behaviors may represent similar user buying intention. For example, when the user considering buying a product, the order of reading reviews and costumer Q&A does not make much difference. Besides, we find micro-behaviors are often inter-connected, while the sequence model could fail to capture such correlations. In this work, we propose to represent the micro-behaviors as a purchase-related micro-behavior graph (PMG), and adopt GCN to take advantage of such graph. The comparative study of graph-based and sequence-based modeling of micro-behaviors can be found in Section 5.6.

7 CONCLUSION AND FUTURE WORK

In this work, we propose GMCM, a novel graph-based CVR estimator, to tackle three challenging issues in industrial CVR estimation: limited purchase-related feedback, data sparsity and sample selection bias. We propose to represent the user micro-behaviors as a purchase-related micro-behavior graph (PMG), and based on this graph representation, we frame the CVR estimation as a graph classification problem over PMG instances. Mutli-task learning and inverse propensity weighting are also adopted to mitigate the data sparsity and sample selection bias. Experiments on six production datasets demonstrate that GMCM consistently outperforms other strong competitors including the state-of-the-art CVR methods.

Two meaningful directions deserve more attention in the future works. Firstly, what kind of micro-behaviors should be utilized. In our experiments, top 10 behaviors that are most related to the purchase are selected. However, more than a hundred are observed in our logs. How to choose the right micro-behaviors deserves further study. Secondly, the losses of micro-behaviors, CTR and CVR are simply added for optimization in this work. More sophisticated fusion strategy may further improve the model performance.

REFERENCES

[1] Junxuan Chen, Baigui Sun, Hao Li, Hongtao Lu, and Xian-Sheng Hua. 2016. Deep ctr prediction in display advertising. In *Proceedings of the 24th ACM International Conference on Multimedia*. 811–820.

[2] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. 7–10.

[3] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. 191–198.

[4] Arnaud De Myttenaere, Bénédicte Le Grand, Boris Golden, and Fabrice Rossi. 2014. Reducing offline evaluation bias in recommendation systems. *23rd annual Belgian-Dutch Conference on Machine Learning* (2014).

[5] Tom Fawcett. 2006. An introduction to ROC analysis. *Pattern Recognition Letters* 27, 8 (2006), 861–874.

[6] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence* (2017).

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.

[8] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37 (ICML'15)*. JMLR.org, 448–456.

[9] Yuchin Juan, Damien Lefortier, and Olivier Chapelle. 2017. Field-aware factorization machines in a real-world online advertising system. In *Proceedings of the 26th International Conference on World Wide Web Companion*. 680–688.

[10] Yuchin Juan, Yong Zhuang, Wei-Sheng Chin, and Chih-Jen Lin. 2016. Field-aware factorization machines for CTR prediction. In *Proceedings of the 10th ACM Conference on Recommender Systems*. 43–50.

[11] Muhammad Junaid Effendi and Syed Abbas Ali. 2017. Click Through Rate Prediction for Contextual Advertisment Using Linear Regression. *International Journal of Computer Science and Information Security (IJCSIS)* (2017).

[12] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *3rd International Conference for Learning Representations* (2014).

[13] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. *Fivth International Conference on Learning Representations* (2017).

[14] Kuang-chih Lee, Burkay Orten, Ali Dasdan, and Wentong Li. 2012. Estimating conversion rate in display advertising from past erformance data. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 768–776.

[15] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1754–1763.

[16] Weiwen Liu, Ruiming Tang, Jiajin Li, Jinkai Yu, Huifeng Guo, Xiuqiang He, and Shengyu Zhang. 2018. Field-aware probabilistic embedding neural network for ctr prediction. In *Proceedings of the 12th ACM Conference on Recommender Systems*. 412–416.

[17] Quan Lu, Shengjun Pan, Liang Wang, Junwei Pan, Fengdan Wan, and Hongxia Yang. 2017. A practical framework of conversion rate prediction for online display advertising. In *Proceedings of the ADKDD'17*. 1–9.

[18] Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, and Kun Gai. 2018. Entire space multi-task model: An effective approach for estimating post-click conversion rate. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1137–1140.

[19] Rong Pan, Yunhong Zhou, Bin Cao, Nathan N Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. 2008. One-class collaborative filtering. In *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 502–511.

[20] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International Conference on Data Mining*. IEEE, 995–1000.

[21] Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th International Conference on World Wide Web*. 521–530.

[22] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as treatments: debiasing learning and evaluation. *Proceedings of the 33rd International Conference on Machine Learning* (2016).

[23] Lili Shan, Lei Lin, and Chengjie Sun. 2018. Combined Regression and Tripletwise Learning for Conversion Rate Prediction in Real-Time Bidding Advertising. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 115–123.

[24] Adith Swaminathan and Thorsten Joachims. 2015. The self-normalized estimator for counterfactual learning. In *Advances in Neural Information Processing Systems*. 3231–3239.

[25] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. *Sixth International Conference on Learning Representations* (2018).

[26] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*. 1–7.

[27] Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. 2019. Doubly Robust Joint Learning for Recommendation on Data Missing Not at Random. *Proceedings of Machine Learning Research* (2019).

[28] Gary M. Weiss. 2004. Mining with Rarity: A Unifying Framework. *SIGKDD Explor. Newsl.* 6, 1 (June 2004), 7–19. https://doi.org/10.1145/1007730.1007734

[29] Hong Wen, Jing Zhang, Quan Lin, Keping Yang, and Pipei Huang. 2019. Multi-Level Deep Cascade Trees for Conversion Rate Prediction in Recommendation System. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 338–345.

[30] Le Wu, Peijie Sun, Richang Hong, Yanjie Fu, Xiting Wang, and Meng Wang. 2018. Socialgcn: An efficient graph convolutional network based model for social recommendation. *arXiv preprint arXiv:1811.02815* (2018).

[31] Jiani Zhang, Xingjian Shi, Shenglin Zhao, and Irwin King. 2019. STAR-GCN: stacked and reconstructed graph convolutional networks for recommender systems. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. AAAI Press, 4264–4270.

[32] Wenhao Zhang, Wentian Bao, Xiao-Yang Liu, Keping Yang, Quan Lin, Hong Wen, and Ramin Ramezani. 2020. A Causal Perspective to Unbiased Conversion Rate Estimation on Data Missing Not at Random. *the Web Conference* (2020).

[33] Weinan Zhang, Tianxiong Zhou, Jun Wang, and Jian Xu. 2016. Bid-aware gradient descent for unbiased learning with censored data in display advertising. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 665–674.

[34] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Deep interest evolution network for click-through rate prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 5941–5948.

[35] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1059–1068.

[36] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2018. Graph neural networks: A review of methods and applications. *arXiv preprint arXiv:1812.08434* (2018).

[37] Meizi Zhou, Zhuoye Ding, Jiliang Tang, and Dawei Yin. 2018. Micro behaviors: A new Perspective in E-commerce Recommender Systems. In *Proceedings of the eleventh ACM International Conference on Web Search and Data Mining*. 727–735.