2024 IEEE 20th International Conference on Automation
Science and Engineering (CASE)
August 28 - September 1, 2024. Bari, Italy

Coupon Personalization: Leveraging Click Data with Deep Learning for Behavioral Insights

Ruotong Sun[1] and Chen Wang[2*]

Abstract— **This paper proposes a deep learning (DL) framework that leverages customer multidimensional click sequence data on e-commerce platforms to predict purchase probabilities and optimize personalized coupon issuance policy. Our study aims to bridge the gap in the existing literature that focuses only on page view and purchase data, thus overlooking the nuanced customer behaviors captured through actions such as adding products to carts and marking them as favorites. We construct a customer-product bipartite graph in the framework and apply heterogeneous Graph Neural Network (GNN) techniques to accommodate individual differences between customers and products. We employ the Hidden Markov Model (HMM) to unravel the latent psychological processes underlying customer purchasing decisions. The two matrices in HMM serve as an enhanced embedding to provide more accurate predictions (about 10% enhancement) with higher interpretability. Lastly, we employ the Bellman equation to formulate an optimal coupon issuance policy. We use click data of cosmetics and snacks on a particular e-commerce platform to demonstrate the interpretability of our model. Our findings indicate that HMM's hidden transition matrix effectively reflects customer loyalty towards cosmetics and extensive browsing patterns in the snacks category. Furthermore, we observe that the revenue increase from each customer after coupon personalization is proportionate with the probabilities of different clicking actions implied by the HMM.**

I. INTRODUCTION

E-commerce has digitalized retailing practices, significantly enhancing our ability to collect rich interaction data. This wealth of information can provide online platforms with deeper insights into customer preferences, playing a pivotal role in forecasting customer behavior and boosting revenue generation. This accumulation of data is not just in sheer volume but extends across multiple dimensions in customer trajectories. Specifically, from the initial page view to the eventual purchase, customers navigate through products back and forth, adding items to their cart or marking them as favorites.

Prior research has introduced several models to analyze the click data, including the cascade model [1][2], the click chain model [3], the Markov chain choice model (MCCM) [4], and the generalized Markov chain choice model (GMCCM) [5]. These models are designed to utilize the rich data generated by e-commerce interactions. Early models tend to assume a linear progression of clicks, e.g., customers view products from the top to the bottom of pages[1], which fails to capture the back-and-forth browsing behaviors exhibited by users. The Markov chain-based models, which are frequently applied in assortment optimization and pricing strategies [6][7][8][9], consider the transition probabilities among products, allowing customers to revisit the previous products. MCCM assumes that a customer will purchase a product and leave the platform if it is within the assortment; if not, the customer will continue browsing the available products. Conversely, GMCCM expands on this by considering the probability of a customer purchasing a product whenever it is available.

However, these approaches mainly concentrate on the coarsest click data (including only page view data and purchasing data), neglecting the preferences revealed through intricate actions such as adding to carts, marking as favorites, or mere page views. This oversight suggests a gap in the current literature, pointing to the need for more complex models to capture online consumer behavior's complexity fully.

Recent literature has showcased the application of machine learning (ML) and deep learning (DL) based choice models across various domains, including prediction enhancement, coupon personalization, pricing, and assortment optimization [10][11][12]. These models often integrate product information and customer demographic details as crucial inputs. However, the challenge for large e-commerce retailers lies in exploiting the vast array of product attributes because maintaining comprehensive product attribute data is a complex and laborious task, especially for large assortments [12]. Furthermore, customer demographic data, while helpful, pose privacy concerns as third parties can potentially deduce them through observed price adjustments in the pricing system [13]. This suggests a limitation in the current scope of ML/DL models, which may not adequately factor in the depth of consumer psychology or the full spectrum of purchasing behaviors.

In this study, we aim to leverage customer click sequence data in e-commerce platforms to predict purchase probabilities and issue personalized coupons based on a deep understanding of customer behavior.

On the one hand, we build a customer-product bipartite graph to address the individual differences in customer behaviors and apply the GNN techniques for enhanced analysis. GNNs are powerful in generating detailed product recommendations and improving click-through rates by capturing the nuanced interactions between users and products [14][15][16]. Specifically, we simplify the Heterogeneous Graph Attention Network (HAN) [17] for our purposes.

[1]Ruotong Sun is with Weiyang College, Tsinghua University, Beijing, 100084, China (e-mail: sunrt21@mails.tsinghua.edu.cn)
[2]Chen Wang is with the Department of Industrial Engineering, Tsinghua University, Beijing, 100084, China (e-mail: chenwang@tsinghua.edu.cn)
*Correspondence author.

2024 IEEE 20th International Conference on Automation Science and Engineering (CASE) | 979-8-3503-5851-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/CASE59546.2024.10711788

The feature spaces of consumers and products can be completely different. Our focus is on understanding the distinct characteristics of consumers and products rather than their differences. Compared to GNNs designed for homogeneous graphs, which deal with a single node type (e.g., all nodes are customers), heterogeneous GNNs offer a more suitable approach for analyzing the customer-product bipartite graph due to their ability to handle multiple types of relationships. The strength of GNN lies in its ability to predict customer preferences across a broad range of products, even if the customer has only interacted with a few items.

On the other hand, we use HMM[18] to capture the complexity of customer behavior while considering the unobservable dynamics of their intentions. The HMM is a tool effectively applied to understand customer actions [19][20] by modeling the hidden psychological processes that influence purchasing decisions. It distinguishes between hidden states and actions. The former represents the psychological aspects of consumer decision-making, encompassing *willing to buy, interested*, and *forgotten*. Later, actions such as *adding items to carts, marking favorites, purchasing*, or *leaving the site* are included.

The main contributions of this paper include:

- Propose a DL framework using click sequence data to enhance customer behavior prediction and personalized revenue management.
- Leverage click sequence data by HMM to improve interpretability regarding customers' behavior transitions.
- Utilize real-world data to perform extensive numerical experiments to demonstrate the potential of coupon personalization for improving revenues.

II. Methodologies

A. Terminologies and Panorama

This section introduces a "predict-then-optimize" framework for behavior prediction and personalized coupon issuance. In the upcoming parts, we use *product trajectories* to specify the sequence of page views across various products within the assortment under consideration. Conversely, we employ *(customer) behavior trajectories* to describe a customer's comprehensive click behavior except for the page view of a single product, capturing the multifaceted interactions. The embedding for customers or products generated by GNN is called *customer* or *product embedding*. They are *node embeddings* of the graph since both customers and products are nodes on the customer-product bipartite graph. Besides, we use *instant purchase rate* to represent the probability of a customer purchasing a product instantly without browsing any other products.

Our approach begins by tailoring the product transition matrix $\boldsymbol{\rho}$ for each customer, which describes page view behaviors among products, named *(customer) product transition*. Then, we utilize HMM to model their within-product transitions by *behavior trajectories*. This method produces high-quality, interpretable embedding for customers and products, referred to as *HMM embedding* in this research. With HMM embedding, we can more precisely estimate the purchase probability μ for each customer-product combination. Following this estimation, we optimize the coupon issuance for each customer, taking into account both $\boldsymbol{\rho}$ and μ. The configuration of this research is shown in Fig. 1. Notice that our study focuses exclusively on analyzing short-term customer behaviors, assuming that all purchase patterns remain valid for the given period.

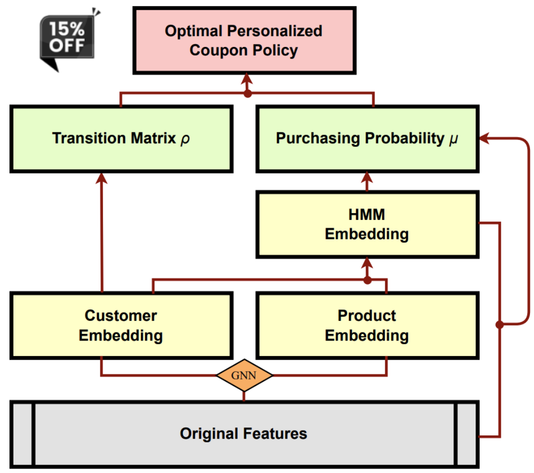

Fig. 1: Configuration of the Proposed Method

B. GNN for Graph Embedding

In this study, we construct a customer-product bipartite graph (hence undirected) $\mathcal{G} = \{\mathcal{U}, \mathcal{V}, \mathcal{E}\}$ utilizing customers' purchase histories. Here, $\mathcal{U}$ represents the customer vertices, $\mathcal{V}$ denotes the product vertices, and $\mathcal{E}$ encompasses the edges between them. Notably, we focus solely on the product trajectories between products, disregarding subsequent actions such as cart additions and favorites, which do not influence the edge weights or formation of new connections.

We leverage the Graph Attention Network (GAT), a variant within the GNN family, to generate more representative embeddings because aggregate-level features extracted from customer behavior history do not adequately capture the relationship between customers and products. For instance, while we can quantify the frequency of a customer's engagements with various products, this does not provide insights into the specific products a customer has interacted with or identify products commonly viewed by any two given customers. However, the resulting embedding by using GNN offers a more informative understanding of these interactions. GAT incorporates attention mechanisms to dynamically assign significance to the features of neighboring nodes during the feature aggregation phase, thus facilitating adaptive learning of features in graph-structured data. Employing multi-head attention further enhances the stability of the attention mechanism's learning process. Specifically, the equation for updating the embedding at layer l for a node v in a multi-head GAT is given by:

$$h_v^{(l)} = \|_{k=1}^{K} \sigma \left(\sum_{u \in N(v)} \alpha_{vu}^k W^{(l)} h_u^{(l-1)} \right), \quad (1)$$

where we have K heads attention, and α_{vu} represents the normalized attention weight, indicating the relative importance of node u to node v. For a bipartite graph, the neighborhood of v comprises vertices of the same type, effectively reaching v through 2-hops on the graph. In the e-commerce scenario, for example, if Ame and Bob both bought milk, they are considered neighbors and their normalized attention weight can be calculated. In our research, since we have many customers and products, incorporating additional layers into the GAT allows the model to aggregate information from more distant nodes. This expansion enhances the model's understanding of broader relational contexts. For this reason, a two-layer GAT is employed to extract the embedding of customers and products.

C. Predicting the Product Transition

We consider an assortment of n products offered on the e-commerce platform, which we denote as $\mathcal{N} = \{1, \cdots, n\}$. Since $\mathcal{N}$ is a subset of the entire product range available to customers, we introduce a virtual product, labeled 0, to act as an absorbing state. It stands for the no-purchase option and the aggregate of all other potential choices outside $\mathcal{N}$, where $\bar{\mathcal{N}} = \mathcal{N} \cup \{0\}$. In this research, each product is treated as a state of an underlying Markov chain, the same as MCCM and GMCCM in the literature. The transition probability ρ_{ij} denotes the likelihood of a customer moving from considering product i to product j, with $\sum_{j \in \bar{\mathcal{N}}} \rho_{ij} = 1$. ρ_{ij} is the entry of transition matrix $\boldsymbol{\rho} \in \mathbb{R}^{(n+1)\times(n+1)}$. Customers direct their attention to products in the assortment $\mathcal{N}$ with arrival rates $\boldsymbol{\lambda} = [\lambda_1, \cdots, \lambda_n]$ and follow the Markov chain before making a purchase decision or leaving.

Suppose we have T product trajectories $\{(i_1^t, i_2^t), \cdots, (i_{n_t-1}^t, i_{n_t}^t)\}_{t=1}^T$, from different customers where i denotes a single product in the product trajectories. The estimation problem is:

$$\max_{\boldsymbol{\rho}} \quad \frac{1}{T} \sum_{t=1}^{T} \frac{1}{n_t} \sum_{w=1}^{n_t-1} \log \rho_{i_w, i_{w+1}} \quad (2)$$

$$s.t. \quad \sum_{j \in \bar{\mathcal{N}}} \rho_{ij} = 1 \qquad \forall i \in \mathcal{N} \quad (3)$$

$$\rho_{0i} = 0, \rho_{00} = 1 \qquad \forall i \in \mathcal{N} \quad (4)$$

$$\rho_{ij} \geq 0 \qquad \forall i \in \mathcal{N}, \forall j \in \bar{\mathcal{N}} \quad (5)$$

Here, $\boldsymbol{\rho}$ is derived by a neural network $\boldsymbol{\Phi} : \mathbb{R}^d \mapsto \mathbb{R}^{(n+1)\times(n+1)}$, where d is the dimension of the aggregate-level customer features. The customer embedding, denoted as $h_{customer}$, is generated through GNN, specifically tuned via backpropagation within our framework. To reduce the complexity of our model, we apply a rank constraint to the matrix before softmax. This restriction serves a dual purpose: it curbs overfitting and reflects the variability in customer transition behaviors, which is affected by the range of products offered. We employ a softmax layer to transform the $(n+1) \times (n+1)$ matrix with rank r into the same size product transition matrix. To align with model constraints (3)-(5), we specifically configure the softmax layer to set its first row to zeroes and ones appropriately. Note that we assume the transition matrix $\boldsymbol{\rho}$ is independent of the coupon issuance, which is in line with literature [5]. This assumption makes sense because the platform can issue coupons to customers after they click on the product.

D. HMM for Predicting Instant Purchase Rate

In the preceding section, our analysis was confined to utilizing merely page view data. For predicting the purchase probability, however, page views and purchase data do not reflect the entire decision-making process of customers when interacting with specific products.

To address this gap, we leverage HMM, which focuses on the behavior trajectories following a page view and captures customers' psychological states. The hidden states $\mathcal{S}$ —"willing to purchase", "interested", and "forgotten"—represent customer interest levels. The action states $\mathcal{A}$—"order", "cart", "fav", "leave"—represent the click behavior a customer makes on a specific product, which constitutes the behavior trajectories. Accordingly, $\mathcal{S} = \{\text{W (willing to purchase)}, \text{I (interested)}, \text{F (forgotten)}\}$ and action $\mathcal{A} = \{\text{order, cart, fav, leave}\}$. The model is parameterized by the initial distribution vector $\boldsymbol{\pi} = [\pi_1, \pi_2, \pi_3]$, the hidden transition matrix $H \in \mathbb{R}^{3\times3}$ that represents the probabilities of transitioning between hidden states, and the emission matrix $E \in \mathbb{R}^{3\times4}$, which denotes the probabilities of transitioning from hidden states to observable actions. These parameters are collectively denoted as $\Lambda = \{\boldsymbol{\pi}, H, E\}$. Given these parameters, the probability of a behavior trajectories $A = \{a_1, \cdots, a_T\}$ is given by:

$$P(A|\Lambda) = \sum_{s_1, \cdots, s_T \in \mathcal{S} \times \cdots \times \mathcal{S}} \pi_{s_1} E_{s_1 a_1} H_{s_1 s_2} \cdots E_{s_T a_T}. \quad (6)$$

We employ the forward-backward algorithm to evaluate probabilities in the HMM. Other algorithms include the Viterbi algorithm, Baum-Welch algorithm, etc.[18].

We make the following assumptions to depict the customers' psychological activities for simplicity of modeling.

Assumption 1: Customers initially view a product out of interest, leading to the initial state probability distribution $\boldsymbol{\pi} = \{1, 0, 0\}$.

Assumption 2: If a customer intends to purchase or is interested in a product, she will not leave the page immediately. Conversely, if a customer forgets to continue the interaction, she will not proceed to purchase the product.

Assumption 3: The hidden transition matrix is a function of the product's attributes, whereas the emission matrix depends entirely on the customer's characteristics.

Assumption 1 posits that a customer's initial product engagement is driven by interest. This simplification bolsters the stability and interpretability of our learning process, reducing the parameter set of the HMM to $\Lambda = \{H, E\}$. Assumption 2 indicates that a product's features primarily influence a customer's emotional responses during the purchase process, while the customer's inherent traits are more decisive in determining their subsequent actions. Assumption 3 stipulates that the $(1,4),(2,4)$, and $(3,1)$ entries in the emission matrix are all zeros.

To estimate the parameters of HMM, we employ another neural network depicted in Fig. 2 (right). We create embedding for products and customers via the GNN mentioned above. Following Assumption 2, two matrices in HMM are derived from these embeddings. The loss function is formulated as follows:

$$l = -\frac{1}{K}\sum_{k=1}^{K}\log P(A_k|\{\hat{H},\hat{E}\}) \tag{7}$$

K is the count of a customer's behavior trajectories with a product, A_k symbolizes a specific behavior trajectory ending with "leave" or "order", and $\hat{H}$,$\hat{E}$ means the estimated matrices. The sequence probability calculation is analogous to (6). Note that the parameters of the GNN and the FC layers in Fig. 2 (right) are updated through backward propagation using this loss.

Post parameter estimation, the two HMM matrices are flattened into vectors named *hidden vector* and *emission vector*, respectively. In a later discussion, we refer to *HMM embedding* to the flattened two matrices in HMM.

To estimate the customer's instant purchase rate μ and discern how it responds to price variations, we concatenate the customer's and product's original features with the emission vector and the hidden vector, respectively. Then we input the result into a Multi-Layer Perceptron (MLP). The MLP is designed with a positive weight for the price feature, which reflects the common market principle that lower prices generally increase purchase rates. The customer-product pairs are collectively represented by the set $\mathcal{B}$, encompassing each pair b.

Given that each product in our dataset experiences minor price fluctuations, we can ensure that the estimated purchase rates are accurate around the average price during the observed period.

The true instant purchase rate is labeled by calculating the proportion of behavior trajectories that terminate in an "order" out of the total number of behavior trajectories for each customer-product pair. The Mean Squared Error (MSE) is used to measure the loss of predictions:

$$l = \frac{1}{|\mathcal{B}|}\sum_{b\in\mathcal{B}}(\tilde{\mu}_b - \bar{\mu}_b)^2 \tag{8}$$

To mitigate the impact of absolute pricing differences across various products on the instant purchase rate, we incorporate a control sample for each customer-product pair representing a zero price and a 100% instant purchase rate. This addition helps normalize the data and provides a baseline for understanding how price influences customer behavior.

E. Coupon Personalization

In this section, we use the transition matrix and instant purchase rate introduced in previous sections to optimize the coupon issuance policy. Consider a set of coupons, $\mathcal{C}$, with each element α representing a discount level within the $(0,1)$ range. The optimal coupon issuance policy for a specific customer can be modeled using a finite Markov Decision Process (MDP). In this model, the state i represents a product, and the action α corresponds to the discount level assigned to the present product of interest. The reward r is defined as zero if there is no purchase activity; otherwise, it is defined as the product price multiplied by the discount minus the cost c if the customer decides to purchase the product at the given discount α. Here, we set the decaying rate $\gamma = 1$. The probability mass function (PMF) of the dynamic function $\mathcal{P}(j,r|i,\alpha)$ is listed in TABLE I, with the notation following section 2C.

Dynamic Function	Probability
$\mathcal{P}(j,ap_i|i,\alpha),\quad j\in\mathcal{N}$	0
$\mathcal{P}(0,ap_i|i,\alpha)$	$\mu_i(\alpha)$
$\mathcal{P}(j,0|i,\alpha),\quad j\in\mathcal{N}$	$(1-\mu_i(\alpha))\rho_{ij}$
$\mathcal{P}(0,0|i,\alpha)$	$(1-\mu_i(\alpha))\rho_{i0}$

TABLE I: PMF of Dynamic Function

We take both "purchase" and "leave" as the terminal state 0. Policy $\pi(\alpha|i)$ for this MDP is the issued coupon level for each product, which can be either deterministic or stochastic. The state value function $v_\pi(i)$ is the expected return starting from product i following personalized coupon policy π. The total revenue earned can be represented as $\mathcal{R}_\pi = \sum_{i\in\mathcal{N}}\lambda_i v_\pi(i)$, where λ_i is the arrival rate. We assume $v_\pi(0)=0$. By the Bellman equation,

$$v_\pi(i) = \sum_{\alpha\in\mathcal{C}}\pi(\alpha|i)\sum_{j,r}\mathcal{P}(j,r|i,\alpha)\left[r+v_\pi(j)\right] =$$
$$\sum_{\alpha\in\mathcal{C}}\pi(\alpha|i)\left((\alpha p_i - c)\mu_i(\alpha) + (1-\mu_i(\alpha))\sum_{j\in\mathcal{N}}\rho_{ij}v_\pi(j)\right) \tag{1}$$

We use value iteration [21] to find the optimal policy for this MDP model. The result is independent of the arrival rate. Algorithm 1 shows the process of value iteration. By assuming that $\rho_{i0} > 0, \forall i\in\mathcal{N}$, setting the decaying rate $\gamma = 1$ will not diverge the result since the contraction mapping still holds (see Appendix). The assumption means a customer always has a positive probability of leaving the system directly without purchasing any product. This value iteration algorithm is the discrete version of the pricing algorithm in [6].

III. Numerical Experiments

In this section, we investigate the efficacy of the proposed coupon issuance policy by experimenting with click sequence data from a particular e-commerce platform in China. During the training process, through hyperparameter tuning on the training data, we set the dimensions of both

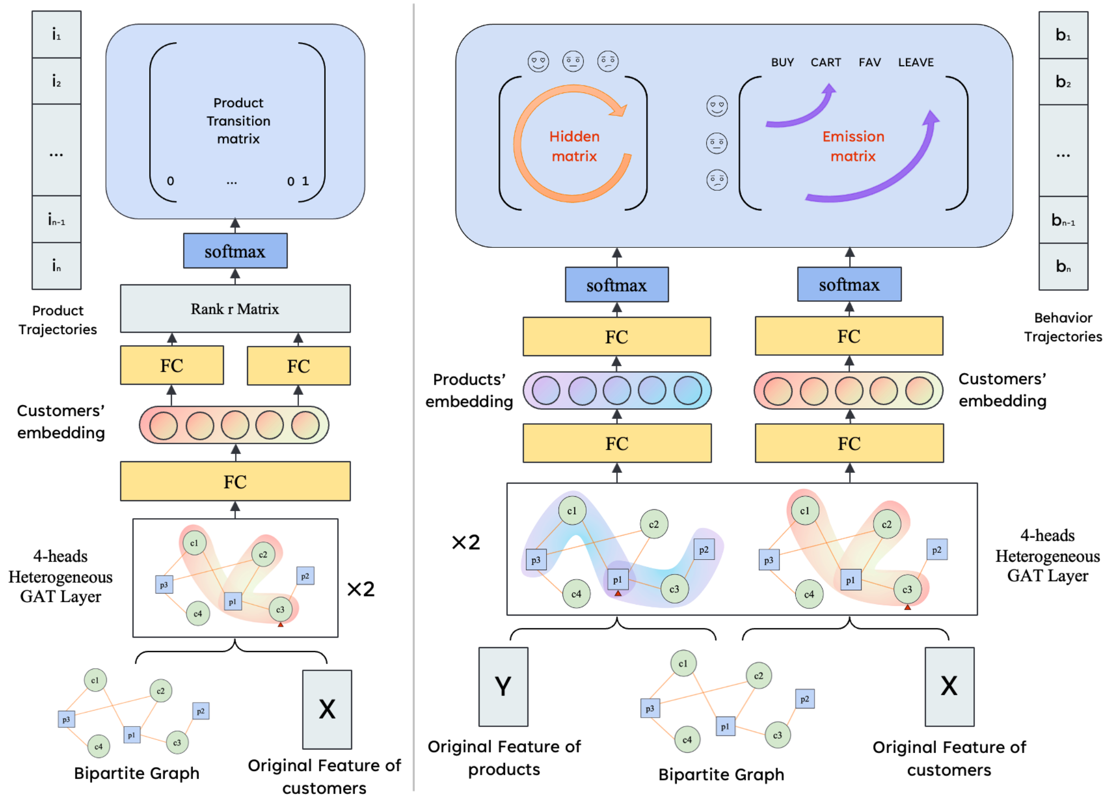

Fig. 2: Framework of Customizing Transition Matrix (left) and Learning HMM Embeddings (right)

Algorithm 1 Value Iteration for Personalized Coupon Policy

Initialize $V(i)$ arbitrarily for all $i \in \mathcal{N}$
Initialize ε (threshold)
repeat
    $\Delta \leftarrow 0$
    for each $i \in \mathcal{N}$ **do**
        $v \leftarrow V(i)$
        Define $R(\alpha, i) \leftarrow \mu_i(\alpha)(\alpha p_i - c)$
        Define $T(\alpha, i) \leftarrow (1 - \mu_i(\alpha)) \sum_{j \in \mathcal{N}} \rho_{ij} v_\pi(j)$
        $V(i) \leftarrow \max_{\alpha \in \mathcal{C}} [R(\alpha, i) + T(\alpha, i)]$
        $\Delta \leftarrow \max(\Delta, |v - V(i)|)$
    end for
until $\Delta < \varepsilon$
Output: $V^*(i) \approx V(i)$ for all $i \in \mathcal{N}$

customer and product embeddings as 36, with a learning rate of 5e-3, and 100 epochs. We believe that the dimension of 36 is enough to capture the features of the customers and products according to the result in this section.

A. Data Preprocessing

The dataset comprises four tables detailing seven days of purchase information alongside multifaceted click data. It encompasses identifiers for products and customers, timestamps, and types of customer actions. The actions recorded include "order", "page view" (pv), "add to cart" (cart), and "mark as favorite" (fav). The dataset has 221,403 customers, 85,778 products, and 7,293,362 click records. To select a proper assortment with a substitution effect and a large interaction sample, we choose 70 snacks and 30 cosmetics with the largest number of purchases. Then, we select 22,530 customers with more than two interactions with the given assortment, which takes 10% of our customer set. Moreover, the action "order" takes 19% of this click data, which provides us with a nice sample to learn customers' purchase behavior.

An essential preprocessing step is to ensure logical sequence in the actions, particularly that a page view precedes any other action. A "pv" action is inserted accordingly, where this sequence is not followed. Furthermore, actions separated by a gap of more than 12 hours are treated as distinct interaction trajectories, reflecting separate shopping sessions.

We then extract the aggregate-level features from the click data, identifying 15 features for customers (such as repeating pv rate, average purchase amount in one order, etc.) and 18 for products (such as the price, the click purchase ratio, etc.).

To get the customized transition matrix $\boldsymbol{\rho}$, we select the product trajectories for customers with the assortment. This is realized by first grouping the data by customers, then only counting for the "pv" behavior and separating it into different trajectories by a 12-hour time interval. On average, we have 2.91 products in each trajectory.

To learn the HMM embedding for the customer-product pairs, we group the data by customer and product, extracting the consecutive click behavior as behavior trajectory. Each behavior trajectory starts with page view and ends with either purchase ("order") or leave without purchasing anything ("leave"), such as {"pv", "cart", "order"} and {"pv", "cart", "leave"}. The action "leave" means either the customer makes a subsequent page view or just leaves the assortment.

B. Results and Analysis

We tailor the transition matrix $\boldsymbol{\rho}$ for each customer using the method detailed in section 2C. Our model is benchmarked against a traditional approach that derives an average-level transition matrix $\boldsymbol{\rho}_{\text{MLE}}$ by straightforwardly maximizing the likelihood function (2). To evaluate the performance of both approaches, we employ the Cross-Entropy (CE) metric and top-k accuracy, assessing whether the model can accurately predict a customer's following chosen product j by determining if it falls within the top-k highest probability products for transitioning from the current product i. Since 100 products are selected in the experiment, we set the matrix rank before the softmax operation to be 10, 20, and 30, which corresponds to $\boldsymbol{\rho}_{10}$, $\boldsymbol{\rho}_{20}$, and $\boldsymbol{\rho}_{30}$. If we set a large rank, the parameters will be too numerous, increasing the risk of overfitting. Choosing a small rank may result in the model being underfitted. For this analysis, we use 80% of the samples for the training set. The outcomes of the test set are presented in TABLE II.

TABLE II: Result Comparison between Traditional Method and Proposed Method (with Different Ranks) on Estimating Transition Matrix $\boldsymbol{\rho}$

Metrics	CE	top-1	top-3	top-5	top-7
Training Set					
$\boldsymbol{\rho}_{10}$	2.99	0.32	0.59	0.67	0.70
$\boldsymbol{\rho}_{20}$	2.69	0.31	0.63	0.73	0.77
$\boldsymbol{\rho}_{30}$	2.62	0.32	0.63	0.73	0.78
$\boldsymbol{\rho}_{\text{MCCM}}$	3.01	0.34	0.59	0.65	0.67
Test Set					
$\boldsymbol{\rho}_{10}$	2.99	0.31	0.59	0.67	0.70
$\boldsymbol{\rho}_{20}$	2.78	0.30	**0.61**	0.70	0.75
$\boldsymbol{\rho}_{30}$	**2.76**	0.29	0.61	**0.71**	**0.75**
$\boldsymbol{\rho}_{\text{MCCM}}$	3.05	**0.33**	0.60	0.68	0.71

The dataset demonstrates the efficacy of tailored transition matrices $\boldsymbol{\rho}$, with $\boldsymbol{\rho}_{30}$ outperforming others by achieving the lowest CE at 2.76 and securing the highest accuracies for top-5 and top-7 predictions. This indicates that a matrix rank that is too low before applying softmax falls short of adequately capturing customer behaviors. Besides, the performance of rank-20 and 30-rank matrices are similar, indicating that setting the rank between 20 and 30 is adequate. Additionally, it's essential to acknowledge that since all trajectories must conclude with a terminal state 0, transitions to the highest probability product invariably lead to the virtual product 0. This explains why traditional estimates of the overall level transition matrix might be able to achieve the highest 33% accuracy rate for the top-1 metric.

We use the behavior trajectories for each customer-product pair to train the HMM matrices. As depicted in Fig. 3, the output embedding of the products by GNN and HMM embedding after 2D-tsne shows that the model trained with multifaceted data can successfully cluster and differentiate the products.

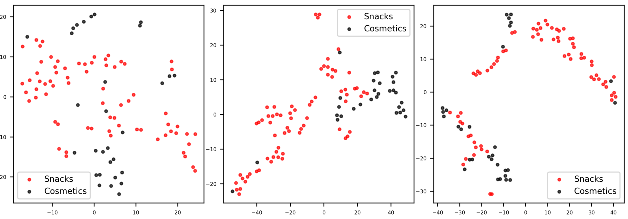

Fig. 3: 2D-tsne Visualization of Aggregate-level Features (left), Embedding Generated by GNN (middle), and HMM Embeddings (right)

The two HMM matrices also have strong interpretability. Since the hidden transition matrix only depends on the features of the products, we take the means of the hidden transition matrices for snacks and cosmetics, respectively, as shown in TABLE III. Since the emission matrix depends only on the features of customers, we average the emission matrices of all $22,530$ customers, shown in TABLE IV.

TABLE III: Hidden Transition Matrix on Average. Snacks (left) and Cosmetics (right)

	W	I	F
W	0.52	0.15	0.33
I	0.19	0.37	0.44
F	0.07	0.71	0.22

	W	I	F
W	0.63	0.17	0.20
I	0.27	0.29	0.44
F	0.11	0.50	0.39

	Order	Cart	Fav	Leave
W	0.7641	0.0020	0.2339	0
I	0.2954	0.0056	0.6990	0
F	0	0.2606	0.1358	0.6036

TABLE IV: Mean Emission Matrix

We use W, I, F to represent "Willing to purchase", "Interested", and "Forgotten", respectively. Table III reveals distinct consumer behaviors towards cosmetics and snacks. For cosmetics, the transition probabilities suggest strong consumer loyalty, with a 63% likelihood of maintaining preference ("Willing to purchase") and a lower tendency towards forgetting ("Forgotten"), at just 20%. This illustrates consumer engagement with cosmetics. In contrast, snacks display a more fluid consumer attitude, evidenced by a 71% probability of moving from "Forgotten" to "Interested", indicating higher substitutability and a lack of solid preference. This data underlines that cosmetics elicit a stronger, more definitive response from consumers, manifesting in either continued preference or an apparent disengagement. In contrast, with their higher substitutability, snacks are less likely to command the same level of distinct consumer loyalty or aversion.

The mean emission matrix reveals that consumers are highly likely to purchase ("Order") when they are already willing to buy the product (76.41%), corresponding to the profile of high-quality users. Those in an "Interested" stance have a significant inclination towards favoring the product ("Fav") at 69.90%, suggesting a strong coherence. The high probability (60.36%) of moving from "Forgotten" to "Leave" also shows strong accordance between hidden state and action.

Next, we show that prediction accuracy increases in the face of HMM embeddings. We use the support vector machine (SVM) and tree-based models including random forest (RF), AdaBoost, and XGBoost to classify if a customer purchases or leaves without purchasing anything. The benchmark approach marked as "Original" only uses the aggregate-level features as input, while "HMM" refers to the case with HMM embeddings only. "HMM+" is when the input concatenates the aggregate-level features and the HMM embeddings.

	RF	AdaBoost	XGBoost	SVM RBF
Original	67.07%	70.57%	69.15%	70.51%
HMM	**78.33%**	75.99%	**78.66%**	71.94%
HMM+	77.64%	**76.94%**	78.04%	**74.11%**

TABLE V: Performance Improvement in Predicting Purchase Behavior with HMM Embeddings

To get the purchase rates at different levels of personalized coupons, we first set the coupon issuance level as no discount, 5%-off, 10%-off, 15%-off, 20%-off, 25%-off, 30%-off. After integrating the aggregate-level features with the HMM embeddings, we trained the MLP introduced at the end of section 2D. Recognizing the potential volatility in neural network training, we conducted the training process 50 times, excluding ten instances where the loss exhibited irregular patterns. On average, the Mean Squared Error (MSE) loss on the test set was 0.064. Fig. 4 illustrates the purchase rates of the first customer to coupon issuance across the first ten products. Here, the line represents the average instant purchase rate, while the shaded area delineates the 95% confidence interval for the instant purchase rate.

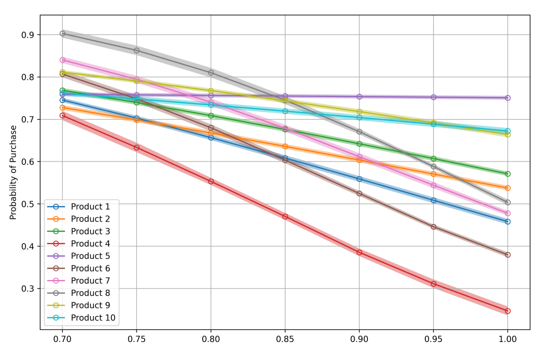

Fig. 4: 1st Customer's Purchase Rates under Different Coupon Levels

Utilizing Algorithm 1, we assigned personalized coupons for each customer-product pair, setting the cost at 60% of the average price observed over seven days. The resulting revenue enhancement varied significantly, achieving a maximum increase of 63.15 and a minimum of zero, with an average uplift of 24.49 per customer. The percentage increase of revenue is about 2.23% per customer. Given the large number of customers (22,530) and the 60% cost, this represents a significant rise. The distribution of optimized prices (after adjustment by coupons) is illustrated in Fig. 5, noting that prices have been anonymized to protect sensitive data.

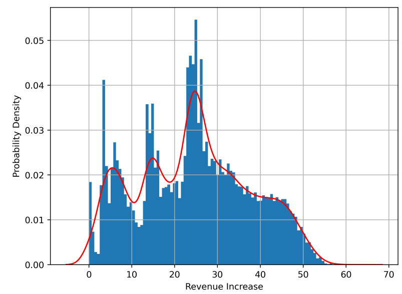

Fig. 5: Distribution of Revenue increased for all (22,530) Customers under the optimal Personalized Coupon Policy

Fig. 6 indicates the satisfactory interpretability of our HMM embeddings. As the emission matrices are only decided by customer features, we calculate the probability of each first step action under uniform initial distribution (i.e., $[1/3, 1/3, 1/3]$). We plot the revenue increase after issuing personalized coupons to each customer against the four actions respectively in Fig. 6. The revenue increase is directly proportional to the probability of marking as a favorite and the probability of leaving while inversely proportional to the probability of purchasing and adding items to the cart. This is because a high probability of purchase or cart addition indicates that the customer already has a strong intention to buy, which means there is little room for profit from issuing coupons. On the other hand, "leaving" shows low purchase intent, and marking it as a favorite often means high prices hinder interest. Issuing coupons can effectively turn this interest into purchases, especially when price is the main barrier. The derived HMM embedding offers a strong interpretative power to the results, effectively summarizing consumer behavior and using multidimensional click data.

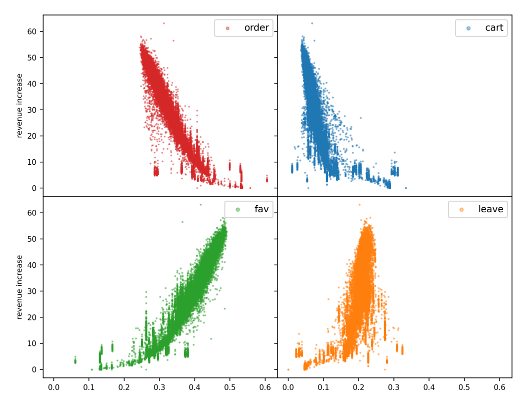

Fig. 6: Revenue Increase against Parameters in HMM

IV. Conclusion

In this paper, we introduce a framework for coupon personalization. The results demonstrate that our model effectively leverages the value of multidimensional click data and has interpretability to customers' behavior. Specifically, although we do not aim to classify the products, the HMM embedding clustered the different products well after fitting the behavior trajectories. Besides, the HMM matrices reflect customers' preferences for various kinds of products and have a strong relationship with the eventual revenue increase. Meanwhile, our research has some limitations: we assume the transition matrix $\boldsymbol{\rho}$ does not depend on coupon issuance, the instant purchase rate μ does not depend on other products, and we overlook products outside the selected assortment. In addition, to verify actual revenue increases and evaluate our model's ability to characterize customer behavior, A/B tests on the e-commerce platform are needed.

APPENDIX

A. Proof in Section 2E

From the value iteration, we can define an operator T, such that $v_\pi^{k+1} \triangleq T v_\pi^k$. Each elements $v_\pi^{k+1}(i)$ is mapped to:

$$\max_{\alpha} \left\{ \mu_i(\alpha)(\alpha p_i - c) + (1-\mu_i(\alpha)) \sum_{j\in\mathcal{N}} \rho_{ij} v_\pi^k(j) \right\}$$

Suppose that we have two value functions v_π and $v_{\pi'}$,

$$\begin{aligned}
&\|Tv_\pi - Tv_{\pi'}\|_\infty = \\
&\max_i \left| \max_\alpha \left\{ (1-\mu_i(\alpha)) \sum_{j=1}^n \rho_{ij} v_\pi(j) + \mu_i(\alpha)(\alpha p_i - c) \right\} \right. \\
&\left. - \max_{\alpha'} \left\{ (1-\mu_i(\alpha')) \sum_{j=1}^n \rho_{ij} v_{\pi'}(j) + \mu_i(\alpha')(\alpha' p_i - c) \right\} \right| \\
&\le \max_{i,\ \alpha} \left| (1-\mu_i(\alpha)) \sum_{j=1}^n \rho_{ij} v_\pi(j) + \mu_i(\alpha)(\alpha p_i - c) \right. \\
&\left. -\ (1-\mu_i(\alpha)) \sum_{j=1}^n \rho_{ij} v_{\pi'}(j) + \mu_i(\alpha)(\alpha p_i - c) \right| \\
&\le \max_{i,\ \alpha} (1-\mu_i(\alpha)) \sum_{j=1}^n \rho_{ij} \max_k |(v_\pi(k) - v_{\pi'}(k))| \\
&< \max_k |(v_\pi(k) - v_{\pi'}(k))| \\
&= \|v_\pi - v_{\pi'}\|_\infty
\end{aligned}$$

The last inequality holds for $\rho_{i0} > 0$, so the operator T corresponds to a contraction mapping. By Banach fixed point theory, the solution of the iteration exists and is unique.

REFERENCES

[1] N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey, "An experimental comparison of click position-bias models," in *Proceedings of the 2008 International Conference on Web Search and Data Mining*, 2008, pp. 87–94.

[2] X. Gao, S. Jasin, S. Najafi, and H. Zhang, "Joint learning and optimization for multi-product pricing (and ranking) under a general cascade click model," *Management Science*, vol. 68, no. 10, pp. 7362–7382, 2022.

[3] F. Guo, C. Liu, A. Kannan, T. Minka, M. Taylor, Y.-M. Wang, and C. Faloutsos, "Click chain model in web search," in *Proceedings of the 18th International Conference on World Wide Web*, 2009, pp. 11–20.

[4] J. Blanchet, G. Gallego, and V. Goyal, "A markov chain approximation to choice modeling," *Operations Research*, vol. 64, no. 4, pp. 886–905, 2016.

[5] M. Liu, J. Cao, and Z.-J. M. Shen, "Pricing under the generalized markov chain choice model: Learning through large-scale click behaviors," *Available at SSRN*, 2022.

[6] J. Dong, A. S. Simsek, and H. Topaloglu, "Pricing problems under the markov chain choice model," *Production and Operations Management*, vol. 28, no. 1, pp. 157–175, 2019.

[7] A. J. Kleywegt and H. Shao, "Revenue management under the markov chain choice model with joint price and assortment decisions," *arXiv preprint arXiv:2204.04774*, 2022.

[8] J. B. Feldman and H. Topaloglu, "Revenue management under the markov chain choice model," *Operations Research*, vol. 65, no. 5, pp. 1322–1342, 2017.

[9] A. Désir, V. Goyal, D. Segev, and C. Ye, "Constrained assortment optimization under the markov chain–based choice model," *Management Science*, vol. 66, no. 2, pp. 698–721, 2020.

[10] Z. Cai, H. Wang, K. Talluri, and X. Li, "Deep learning for choice modeling," *arXiv preprint arXiv:2208.09325*, 2022.

[11] H. Wang, X. Li, and K. Talluri, "Transformer choice net: A transformer neural network for choice prediction," *arXiv preprint arXiv:2310.08716*, 2023.

[12] S. Gabel and A. Timoshenko, "Product choice with large assortments: A scalable deep-learning model," *Management Science*, vol. 68, no. 3, pp. 1808–1827, 2022.

[13] X. Chen, D. Simchi-Levi, and Y. Wang, "Privacy-preserving dynamic personalized pricing with demand learning," *Management Science*, vol. 68, no. 7, pp. 4878–4898, 2022.

[14] J. Wang, P. Huang, H. Zhao, Z. Zhang, B. Zhao, and D. L. Lee, "Billion-scale commodity embedding for e-commerce recommendation in alibaba," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 839–848.

[15] F. Li, Z. Chen, P. Wang, Y. Ren, D. Zhang, and X. Zhu, "Graph intention network for click-through rate prediction in sponsored search," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019, pp. 961–964.

[16] Z. Li, X. Shen, Y. Jiao, X. Pan, P. Zou, X. Meng, C. Yao, and J. Bu, "Hierarchical bipartite graph neural networks: Towards large-scale e-commerce applications," in *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 2020, pp. 1677–1688.

[17] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, "Heterogeneous graph attention network," in *The World Wide Web Conference*, 2019, pp. 2022–2032.

[18] L. Rabiner and B. Juang, "An introduction to hidden markov models," *IEEE ASSP Magazine*, vol. 3, no. 1, pp. 4–16, 1986.

[19] K. Yada and N. Sano, "Customer behavior modelling using radio frequency identification data and the hidden markov model," in *2012 Annual SRII Global Conference*. IEEE, 2012, pp. 509–514.

[20] H. Kawazu, F. Toriumi, M. Takano, K. Wada, and I. Fukuda, "Analytical method of web user behavior using hidden markov model," in *2016 IEEE International Conference on Big Data (Big Data)*. IEEE, 2016, pp. 2518–2524.

[21] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT press, 2018.