Web-to-Voice Transfer for Product Recommendation on Voice

Rongting Zhang
rongtz@amazon.com
Amazon
Seattle, USA

Jie Yang
jiy@amazon.com
Amazon
Seattle, USA

ABSTRACT

While product recommendation algorithms on the Web are well-supported by a vast amount of interaction data, the same is not true on Voice. A promising approach to mitigate the issue is transfer learning, i.e., transferring the knowledge of customers’ shopping behaviors learned from their shopping activities on the Web to Voice. Such a Web-to-Voice transfer is challenging due to customers’ distinct shopping behaviors on Voice: customers are inclined to purchase more low-consideration products and are more likely to purchase certain products repeatedly. This paper presents TransV, a novel Web-to-Voice neural transfer network that allows for effective transfer of customers’ shopping patterns from the Web to Voice, while taking into account customers’ distinct purchase patterns on Voice. Our method extends the state-of-the-art self-attention neural architecture with a multi-level tri-factorization neural component, which allows to explicitly capture the similarity and dissimilarity of customers’ shopping patterns on the Web and Voice. To model repeated purchases, TransV adopts a recency-based copy mechanism that considers the impact of the recency of historical purchases on customers’ behavior of repeated purchases. Extensive validation on multiple real-world datasets, including two cross-platform datasets from Amazon.com and Amazon Alexa, shows that our method is able to improve voice-based recommendation substantially by 26.8% as compared with non-transfer learning methods.

CCS CONCEPTS

• Information systems → Recommender systems; **• Computing methodologies → Transfer learning**; **Neural networks**.

KEYWORDS

Voice-based recommendation; Web-to-Voice transfer; Repeated purchase

ACM Reference Format:
Rongting Zhang and Jie Yang. 2020. Web-to-Voice Transfer for Product Recommendation on Voice. 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/10.1145/3397271.3401164

1 INTRODUCTION

In the last few years, there has been an explosion of interest in voice-based technology across industry, with an estimated 144.3 million shipments of voice-enabled devices last year [43]. An important use case for voice-enabled devices is voice shopping, which has become an increasingly important shopping scenario. A recent survey shows that up to 43% of voice-enabled device owners use their device to shop and by 2022, voice is expected to be a $40 billion channel for shopping [9].

A fundamental task in voice shopping is product recommendation, where the goal is to recommend relevant products to customers by inferring their preferences. While a growing body of research has addressed the voice-based recommendation problem from the dialogue perspective, i.e., improving the effectiveness of question-answering between the customer and system [7, 8, 11, 23, 25, 44, 45], relatively little work has been focused on addressing the specific challenges arising in recommendation on Voice [40].

Due to the unique characteristics of voice interfaces (e.g., narrow information channel), customers tend to explore fewer products and choose fewer long-tail products, as compared to Web-based channels [45]. Consequently, products purchased through Voice are much more limited in terms of both quantity and diversity. Due to the fact that voice interfaces are new and not yet widely adopted, customers on Voice do not have a long history as compared with Web. These problems pose a bigger-than-ever data sparsity challenge that impedes effective recommendation. To mitigate the data sparsity issue, a promising approach is transfer learning: a customer is likely to share similar shopping behaviors on the Web and Voice in terms of favored product types and purchase patterns; by transferring the shared shopping patterns from Web, the system can readily generate recommendations for customers with limited historical purchases on Voice, or even those new to Voice.

Despite its obvious potential, transfer learning from Web to Voice is non-trivial due to customers’ distinct shopping behaviors on Voice [19]. For example, customers are more inclined to purchase low-consideration products (e.g., paper towel and toothpaste) than high-consideration ones (e.g., computer monitors) on Voice. This is in part due to the recency of Voice as a shopping medium that customers are not used to making complex shopping decisions by voice, in part due to the lack of technology for supporting effective interactions. On the other hand, the convenience of voice interactions triggers a strong tendency of repeated purchases in voice shopping: many products are purchased by the same customers over and over, especially those consumables that need to be purchased on a regular basis. We note that repeated purchase behaviors have also been observed on the Web, which is mainly driven by customers’ loyalty to certain brands [5, 10, 42].

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

ACM ISBN 978-1-4503-8016-4/20/07.
https://doi.org/10.1145/3397271.3401164

In the recommendation literature, transfer learning has been implemented by extending recommendation models, e.g., factorization models [28, 29, 33] or neural networks [14, 21, 26], with shared user[1] representations for cross-domain recommendation tasks. While being different in the underlying recommendation models (see Section 2 for a detailed discussion), both classes of methods are designed for transferring user representations with less emphasis on transferring the interaction patterns, which has to be carefully considered in our Web-to-Voice transfer context. More recent work [18] attempts to address the problem by modeling the transfer of the purchase patterns from one domain to another using a linear transformation. However, it fails to capture the relationships of interaction patterns across domains, such as similarity and dissimilarity, which are essential for Web to Voice transfer.

This paper introduces TransV, a novel neural network-based recommendation method for transferring customers' shopping behaviors from Web to Voice while considering the uniqueness of voice shopping. TransV is designed to learn shared customer and product representations across both channels and to carefully transfer the purchase patterns from Web to Voice by modeling their relationships explicitly. Specifically, our method is built on the state-of-the-art self-attention neural architecture [41, 49] to learn customer and product representations. To enable effective transfer of customers' purchase patterns from Web to Voice, TransV adopts a multi-level tri-factorization approach that models web and voice purchase patterns with both shared and separated neural parts. By doing so, TransV distinguishes channel-specific purchase patterns from channel-independent ones. To account for customers' repeated purchase behaviors that are prevalent in voice shopping, TransV adopts a recency-biased copy mechanism, which leverages the copy mechanism [16, 36] and extends it by considering the impact of the recency of historical purchases on repeated purchases.

TransV generates recommendations from a mixture of general and repeated purchase probability, thereby unifying Web-to-Voice transfer learning with repeated purchase modeling in a holistic neural model. It can be trained in an end-to-end manner that automatically learns the importance of general and repeated purchases for generating the most relevant product recommendations on Voice. In summary, we make the following key contributions:

- We introduce the problem of Web-to-Voice transfer learning for effective recommendation in voice shopping.
- We propose a multi-level tri-factorization approach that allows for effective Web-to-Voice transfer while taking into account customers' behaviors on Voice.
- We present a unified neural transfer network that orchestrates both transfer learning and repeated purchase modeling for voice-based recommendation.

To the best of our knowledge, this is the first work to study transfer learning for voice-based recommendation. Extensive validation on multiple real-world datasets, including two cross-channel datasets from Amazon.com and Amazon Alexa (one of today's major voice shopping channels), shows that our method is able to improve the quality of voice-based recommendation by 26.8% as compared with non-transfer learning methods measured by NDCG@1.

2 RELATED WORK

This section discusses relevant work from the emerging field of voice-based recommendation, and then reviews existing methods related to ours in transfer learning and repeated purchases.

2.1 Voice-based Recommendation

With the rapid increase of personal assistants, a considerable amount of literature has grown up around conversational recommendation. The focal point of research efforts has been enabling the system to effectively and efficiently infer users' intents and satisfy their information needs [35, 47]. Due to the complexity of the problem, it has been studied by several research communities including natural language processing [11, 23, 25], human-computer interaction [7, 45], and information retrieval (including recommender systems) [8, 44]. Existing work mainly takes a dialogue perspective with the goal of improving the question-answering process, i.e., asking the most relevant questions to collect user feedback.

From the recommendation perspective, most existing work assumes a cold-start setting that ignores long-term preferences of users. For example, Zhang et al. [47] studies the effect of *in-session* aspect-based questions for product recommendation using memory networks [39]. Li et al. [25] introduce a neural dialogue model that classifies the sentiment of a user with respect to movies discussed *in the conversation session*, and based on that, it generates movie recommendations with a pre-trained autoencoder recommender [37]. A recent paper by Sun et al. [40] shows that the integration of users' past purchasing behaviors boosts the effectiveness of voice-based recommendation. Their work, however, concentrates on methods for integrating recommendation techniques into the dialogue system. Our work takes a step back and aims at bridging the conventional recommendation techniques with the recommendation task on Voice, with a specific focus on Web-to-Voice transfer which is of key importance for successful voice shopping in practice.

2.2 Transfer Learning for Recommendation

Transfer learning has been a popular approach for tackling the data sparsity problem by transferring the knowledge (e.g., user preferences) in a source domain to the task in the target domain [6, 24]. In recommendation, transfer learning is generally implemented through multi-task learning [48], i.e., joint model training for recommendation in source and target domains, with a specific focus on transferring user and item representations, or interaction patterns across the domains.

Early work focuses on adapting matrix factorization techniques for transfer learning [28, 29, 33]. Liu et al. [28] introduce a model based on collective matrix factorization [38], where user latent factors are shared across different domains. The observed interactions in the source domain help to train better latent factors, thus transferring the knowledge to the target domain. Pan et al. [32, 33] propose to model the interaction patterns in different domains as independent parameter matrices. Factorization models, however, only learn latent factors and parameter matrices in a linear fashion, which are oversimplified in capturing the complex user-item interaction patterns. More importantly, these methods fail to capture the relationship between user interactions in different domains.

[1] We use "user" as a generic term and "customer" to specifically refer to the user in shopping contexts; similar for "item" vs. "product", and "interaction" vs. "purchase".

Neural network-based methods are more capable of learning nonlinear latent representations for users and items and potentially also their interactions (see a recent survey [46]). Specific to neural transfer learning for cross-domain recommendation, Elkahky et al. [14] introduce a multi-view deep learning method that learns shared user representations from user-item interactions in different domains. Lian et al. [26] propose to incorporate content information into the multi-view neural network. Kanagawa et al. [21] go further in this direction and formulate cross-domain recommendation as extreme multi-class classification, where only content features are used for adapting a classifier trained in the source domain to the target domain. These methods focus on learning better representations for users and items, while emphasizing less learning their interaction patterns, which is important for voice shopping.

A recent paper [18], perhaps the most closely related work to ours, models the transfer of interactions across multiple domains as a linear transformation using cross-stitch networks [30]. In the context of Web-to-Voice transfer, since the same products appear in both Web and Voice, we adopt a tri-factorization approach [12, 31] that fixes the product representations across channels, and extend the approach into a multi-level scheme, which allows to capture channel-independent and channel-specific interaction patterns.

2.3 Repeated Purchase

Due to the importance for business profitability, repeated purchases are an important customer behavior studied in marketing as a signal of brand loyalty [20, 34]. In Web-based information systems, repeated item consumption has been shown to be most affected by the recency and quality of the item, with recency being more critical [1]. Benson et al. [4] study such a behavior in more detail and reveals the increasing inter-arrival gaps of repeated item consumption that eventually lead to abandonment. In online shopping, Bhagat et al. [5] study repeated purchases for consumable products, e.g., toothpaste and diapers, and propose a prediction model that helps increase the product click-through rate. Our work extends the study to voice shopping and the model to collaborative filtering that recommends both repeated and novel products.

Repeated purchases have only been considered in recommender system literature recently. Wan et al. [42] introduce a recommendation algorithm, adaLoyal, a personalized grocery recommender. In adaLoyal, the repeated purchase is leveraged in a post-processing procedure to adapt the prediction of purchase probability over an item using the customer' historical purchases of the item. The adaptation is implemented through a posterior calculation that accounts for both probabilities of general and repeated purchases. Ren et al. [36] propose a unified neural network model that jointly learns repeated and novel consumption for session-based recommendation. Our method is different in that we consider the general sequential recommendation scenario and further model the impact of recency of historical purchases for voice-based recommendation.

3 THE TRANSV MODEL

This section introduces our proposed method `TransV` for transferring customers' shopping patterns from Web to Voice. The overall structure of `TransV` is illustrated in Figure 1(a). It is composed of the following modules: 1) Sequence Encoder, which learns high-quality customer and product representations *shared* across channels given historical customer-product interaction records from both channels; 2) Multi-level Interaction Module, which models the purchase patterns on the Web and Voice as well as their *relationships* using a multi-level tri-factorization approach; 3) Repeated Purchase Module, which captures customers' repeated purchase behaviors on both channels; and 4) Mixture Output Module, which generates the output from a mixture of general and repeated purchase probability.

In the following, we start by formalizing the problem before introducing each of the modules in a separate subsection.

Problem Statement. We model voice-based recommendation as a sequential recommendation problem, i.e., given a sequence of products a customer has previously purchased on the Web and Voice, we predict the next purchases. Formally, let $\mathcal{U}$ and $\mathcal{V}$ be the set of customers and products, respectively; let $S_u = [v_{u,1}, v_{u,2}, \ldots, v_{u,n_u}]$ and $T_u = [t_{u,1}, t_{u,2}, \ldots, t_{u,n_u}]$ denote the sequence of products customer $u \in \mathcal{U}$ purchased on both channels and the corresponding timestamps of the purchases. Note that a timestamp is represented as the difference with respect to a reference time point in terms of the number of weeks; this allows us to map the timestamps to embeddings, so as to model the temporal effect of past purchases.

Suppose t_k is the time of prediction ($t_{u,n_u} \leq t_k$), our sequential recommendation problem with Web-to-Voice transfer is formulated as predicting the purchase probability of any product on both channels at t_{k+1}:

$$P(v^d_{t_{k+1}} = v | S_u, T_u), \quad v \in \mathcal{V}, d \in \mathcal{D}, \tag{1}$$

where $\mathcal{D}$ is the set of channels (i.e., Web and Voice in our case).

3.1 Sequence Encoder

The sequence encoder takes as input the sequence of products historically purchased by a customer and generates customer representations. It starts by encoding the products and the corresponding timestamps of purchases, then encodes the sequence of embeddings with transformer layers that take into account the dependencies between the purchases, and finally generates customer representations through a self-attention pooling layer.

Embedding Products and Timestamps. Item encoding is implemented as a single embedding lookup layer. In our case, we also consider the category taxonomy of products available to obtain higher quality embeddings. Specifically, we define the embedding of a product $v_j \in \mathcal{V}$ as the sum of embeddings of its affiliated categories at different levels of the taxonomy:

$$\mathbf{v}_j = \sum_{l=1}^{L} \mathbf{v}_j^{(l)}, \tag{2}$$

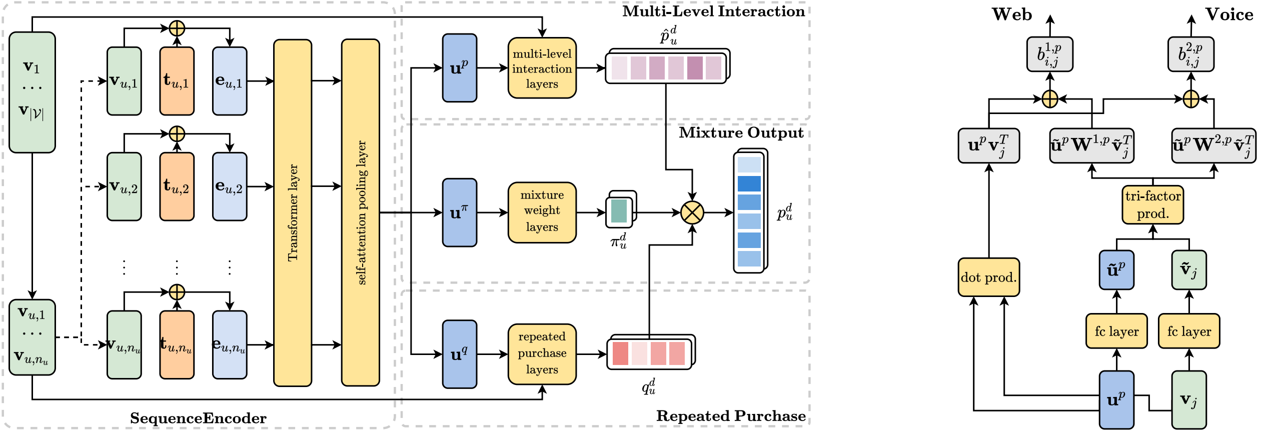
where $\mathbf{v}_j^{(l)} \in \mathbb{R}^m$ is the embedding of v_j's category at the l-th level.

To capture the temporal dynamics of S_u, we embed the timestamp of a purchase $t_{u,i} \in T_u$:

$$\mathbf{t}_{u,i} = \boldsymbol{\tau}(t_k - t_{u,i}), \tag{3}$$

where $\boldsymbol{\tau}$ represents a embedding lookup layer such that $\mathbf{t}_{u,i} \in \mathbb{R}^m$. The final representation of a product purchase is given as the sum of the item embedding and time embedding:

$$\mathbf{e}_{u,i} = \mathbf{v}_{u,i} + \mathbf{t}_{u,i}. \tag{4}$$



(a) Architecture of TransV

(b) Multi-level Interaction layers

Figure 1: The architecture of TransV (a) and the zoomed-in details of the multi-level interaction layers (b). TransV employs the Sequence Encoder to learn shared product and customer representations across channels from the customer's historical purchases. These representations are on the one hand, fed to the Multi-level Interaction Module to learn the purchase probability of the customer for the product on both Web and Voice, and on the other hand, fed to the Repeated Purchase Module to learn the repeated purchase probability on the two channels. The output from both modules are combined by the Mixture Output Module to generate the final purchase probability. To transfer the customer's purchase patterns, the Multi-level Interaction Module distinguishes channel-independent purchase patterns, i.e., $\mathbf{u}^p\mathbf{v}_j^T$, from channel-specific ones, i.e., $\tilde{\mathbf{u}}^p\mathbf{W}^{d,p}\tilde{\mathbf{v}}_j^T$.

Transformer Layers. To learn high-quality customer representations, we leverage the transformer layers [41] that capture the dependency between the purchases based on semantic affinity in the embedding space. To do so, we adopt the multi-head self-attention mechanism [27, 41], which allows jointly attending to information from different parts of the purchase sequence. Formally, let $\mathbf{E} = [\mathbf{e}_{u,1}, \mathbf{e}_{u,2}, \ldots, \mathbf{e}_{u,n_u}]^T \in \mathbb{R}^{n_u \times m}$ be the output from the embedding layer, we construct new representations using h attention heads, each learning a specific dependency relationship within the sequence as follows:

$$\mathrm{MH}(\mathbf{E}) = [head_1, head_2, \ldots, head_h]\mathbf{W}^O, \tag{5}$$

$$head_i = \mathrm{Attention}(\mathbf{E}\mathbf{W}_i^Q, \mathbf{E}\mathbf{W}_i^K, \mathbf{E}\mathbf{W}_i^V, m/h), \tag{6}$$

where $\mathbf{W}_i^Q, \mathbf{W}_i^K, \mathbf{W}_i^V \in \mathbb{R}^{m \times m/h}$ are projection matrices for each attention head and $\mathbf{W}^O \in \mathbb{R}^{m \times m}$ is the output projection matrix for the heads combined. The attention function is given by:

$$\mathrm{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}, m) = \mathrm{Softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{m}})\mathbf{V}. \tag{7}$$

In addition, we apply a position-wise feed-forward layer to the output of the multi-head self-attention layers. The output of the feed-forward layer is calculated as

$$\mathbf{A} = \mathrm{LayerNorm}(\mathbf{E} + \mathrm{Dropout}(\mathrm{MH}(\mathbf{E}))), \tag{8}$$

$$\mathbf{S} = \mathrm{Dropout}(\sigma^F(\mathbf{A}\mathbf{W}_1^F + \mathbf{b}_1^F))\mathbf{W}_2^F + \mathbf{b}_2^F, \tag{9}$$

$$\mathbf{F} = \mathrm{LayerNorm}(\mathbf{A} + \mathrm{Dropout}(\mathbf{S})), \tag{10}$$

where $\mathbf{W}_1^F$ and $\mathbf{W}_2^F$ are parameter matrices and $\mathbf{b}_1^F$ and $\mathbf{b}_2^F$ are bias terms; $\sigma^F(x) \doteq \mathrm{Softrelu}(x) = \log(1 + e^x)$ is applied element-wise. Here we adopt residual connection [17] and layer normalization [2].

Self-Attention Pooling Layer We then construct three types of customer representations, $\mathbf{u}^p, \mathbf{u}^q, \mathbf{u}^\pi$, to model the general customer preference, repeated purchase preference and their relative importance for recommendation, respectively. To do so, we apply an attention-based pooling layer [27]:

$$\begin{aligned} \mathbf{u} \doteq& [\mathbf{u}^p, \mathbf{u}^q, \mathbf{u}^\pi]^T \\ =& \mathbf{F}^T\mathrm{Dropout}(\mathrm{Softmax}(\mathrm{Dropout}(\sigma^P(\mathbf{F}\mathbf{W}_1^P))\mathbf{W}_2^P)), \end{aligned} \tag{11}$$

where $\sigma^P \doteq \mathrm{Softrelu}$, $\mathbf{W}_1^P \in \mathbb{R}^{m \times m}$ and $\mathbf{W}_2^P \in \mathbb{R}^{m \times 3}$ are parameter matrices.

3.2 Multi-Level Interaction Module

Now we consider the problem of modeling customer-product interactions induced from customers' general preference (i.e., non-repeated purchase) on different channels.

A standard approach to model interactions on different domains when entities are shared is matrix tri-factorization [31], which models the interaction scoring function on a specific domain d as:

$$b_{u,j}^{d,p} = \mathbf{u}^p\mathbf{W}^{d,p}\mathbf{v}_j^T, \tag{12}$$

where $\mathbf{W}^{d,p}$ is a channel-specific parameter matrix capturing customer-product interaction patterns. Such a formulation, however, cannot capture the similarity between interaction patterns across channels.

We introduce the multi-level interaction layers which model customer-product interaction as the sum of two factors, the *channel-independent* affinity of the product to the customer's taste and the *channel-specific one*. Formally, we calculate the interaction score by:

$$b_{i,j}^{d,p} = \mathbf{u}^p\mathbf{v}_j^T + \tilde{\mathbf{u}}^p\mathbf{W}^{d,p}\tilde{\mathbf{v}}_j^T, \tag{13}$$

where $\tilde{\mathbf{u}}^p, \tilde{\mathbf{v}}_j$ are low-dimensional vectors derived from $\mathbf{u}^p, \mathbf{v}_j$ through fully connected layers with Softrelu activation. The first term $\mathbf{u}^p \mathbf{v}_j^T$ captures customers' purchase patterns across channels, while the second term $\tilde{\mathbf{u}}^p \mathbf{W}^{d,p} \tilde{\mathbf{v}}_j^T$ captures channel-specific patterns. The details of our multi-level interaction layers are depicted in Figure 1(b).

We then obtain the interaction probability over all the products using softmax:

$$\hat{p}_u^d = \mathsf{Softmax}([b_{u,1}^{d,p}, \ldots, b_{u,|\mathcal{V}|}^{d,p}]), \tag{14}$$

where $\hat{p}_{u,j}^d$ models the probability that the customer u would purchase a specific product j among all the products on channel d.

3.3 Repeated Purchase Module

We now introduce our method for modeling the specific interaction pattern of repeated purchases of the same product. The key idea is to learn a probability of repeating a historical purchase for each customer. To do so, we adopt the copy mechanism [16], which models repeated purchases as copying purchases from the past. In the specific context of product recommendation, the probability is dependent not only on the customer's repeated purchase preferences, but also on the recency of the historical purchases. We therefore extend the copy mechanism by considering the impact of the recency of historical purchases on repeated purchases.

Since the repeated purchase behavior can vary across different channels, we model the repeated purchase score of product $v_{u,i} \in S_u$ for customer $u \in \mathcal{U}$ by:

$$b_{u,(u,i)}^{d,q} = \mathbf{u}^q \mathbf{v}_{u,i}^T + \tilde{\mathbf{u}}^q \mathbf{W}^{d,q} \tilde{\mathbf{v}}_{u,i}^T, \tag{15}$$

where $\mathbf{v}_{u,i}$ is the vector representation of $v_{u,i}$ and $\tilde{\mathbf{u}}^q, \tilde{\mathbf{v}}_{u,i}$ are low-dimensional vectors derived from $\mathbf{u}^q, \mathbf{v}_{u,i}$ through fully connected layers with Softrelu activation, respectively. Note that here we consider each purchase in the sequence, i.e., $v_{u,i} \in S_u$, rather than each product, as a candidate for repetition. The repeated purchase probability for the same product will be summed up.

In practice, customers tend to repeatedly purchase products that are recently purchased. We therefore, consider the impact of a historical purchase on current purchase decision as affected by purchase recency. Given the timestamp of the i-th purchase $t_{u,i}$ and the time of prediction t_k, we define a bias term for modeling time recency as:

$$g_{u,i} = T(t_k - t_{u,i}), \tag{16}$$

where $T(\cdot)$ is a learnable scalar function which can be represented by a single-dimensional embedding lookup layer. With such a bias term, the distribution of repeated purchase probability over historical purchases is given by:

$$q_{u,(u,i)}^d = \frac{\exp(b_{u,(u,i)}^{d,q} + g_{u,i})}{\sum_{l=1}^{n_u} \exp(b_{u,(u,l)}^{d,q} + g_{u,l})}. \tag{17}$$

Consequently repeated purchase probability of product $v_j \in \mathcal{V}$ is

$$q_{u,j}^d = \sum_{i=1}^{n_u} \mathbb{1}_{\{v_j = v_{(u,i)}\}} q_{u,(u,i)}^d, \tag{18}$$

where $\mathbb{1}_{\{\cdot\}}$ is the indicator function returning 1 if the statement is True and 0 otherwise. We note that in a degenerate case where the repeated purchase score and time recency bias for all purchased products are equal, the resulting distribution of repeated purchase probability is equivalent to the empirical purchase frequency.

3.4 Mixture Output and Loss Function

The overall probability of a customer purchasing a product is modeled as a mixture of the general purchase probability and the repeated purchase probability, i.e., $\hat{p}^d$ and q^d:

$$p_u^d = \pi_u^d q_u^d + (1 - \pi_u^d) \hat{p}_u^d. \tag{19}$$

where π_u^d is the mixture weight that describes the importance of repeated purchase in customers' shopping decision.

The mixture weight is considered to be dependent on specific customers and the recency of their last purchases. Formally, considering the timestamp of the last purchase t_{u,n_u}, we define the time recency bias as:

$$h_u = S(t_k - t_{u,n_u}), \tag{20}$$

where $S(\cdot)$ is a learnable scalar function. The mixture weight π_u^d is given by:

$$\pi_u^d = \frac{\exp(\mathbf{w}^{d,\pi} \mathbf{u}^\pi + b^{d,\pi} + h_u)}{1 + \exp(\mathbf{w}^{d,\pi} \mathbf{u}^\pi + b^{d,\pi} + h_u)}, \tag{21}$$

where $\mathbf{w}^{d,\pi} \in \mathbb{R}^m$ and $b^{d,\pi} \in \mathbb{R}$ are parameters to be learned.

Loss Function. Our model generates recommendations on both Web and Voice. To consider the importance of recommendation relevance on both channels for model training, we introduce a hyperparameter α^d as the weight for loss on channel $d \in \mathcal{D}$. Specifically, let $c_{u,j}^d$ be the normalized empirical frequency of observed purchases on channel d for customer $u \in \mathcal{U}$ and product $v_j \in \mathcal{V}$, the training loss is given by:

$$\mathcal{L} = \sum_{d=1}^{|\mathcal{D}|} \alpha^d \sum_{u \in \mathcal{U}} \sum_{j=1}^{|\mathcal{V}|} c_{u,j}^d \log(p_{u,j}^d). \tag{22}$$

4 EXPERIMENTS AND RESULTS

In this section, we perform experiments to evaluate the performance of TransV.We aim to answer the following questions:

- **Q1**: How much benefit does modeling repeated purchases bring to recommendation performance?
- **Q2**: How well does our proposed Web-to-Voice transfer learning perform for recommendation on Voice?
- **Q3**: How effective is TransV in uncovering the impact of historical purchase recency on reorder and weighting the importance of repeated purchases in voice shopping?

In addition, we investigate the impact of data sparsity on the effectiveness of our method. In the following, we start by introducing our experimental setup, before answering each of the above questions in a separate subsection.

4.1 Experimental Setup

Datasets. We evaluate our proposed model on four real-world shopping datasets. Among them, two datasets are publicly available and each contains customers' purchase records on a single channel. These datasets allow us to evaluate the effectiveness of TransV in modeling repeated purchases.

Table 1: Basic statistics of public datasets. Statistics marked by NA are not applicable.

Datasets	#item	#user	#purchase	#pur./#user	#department	#category	#subcategroy	N_w	#train sample	#test sample	repeat rate
Dunnhumby	20,248	2,493	1,777,413	712.96	27	292	1,680	30	25,374	8,443	0.3718
Instacart	22,889	34,486	8,556,249	248.11	21	134	*NA*	28	174,784	63,209	0.6280

Table 2: Basic statistics of Amazon datasets. Δ repeat rate is defined as the relative increment of repeated purchase rate on Voice with respect to that on Web.

Datasets	#item	#purchase per user	#depart.	#categ.	N_w	Δ repeat rate
Grocery	49,419	25.01	293	1,959	41	0.2829
Home	52,092	7.24	129	1,216	27	0.1568

- **Dunnhumby:** This is a grocery shopping dataset released by Dunnhumby.[2] It contains shopping history of 2,500 households for around two years.
- **Instacart:** This is another grocery shopping dataset released by Intacart.[3] It contains shopping history of more than 200 thousand users. This dataset only records the gap between two consecutive transactions with a cutoff at 30 days where gaps greater than 30 days are logged as 30 days. Thus, we only keep users whose maximum gap between two consecutive orders is less than 30 days so all her shopping timestamps can be reconstructed.

To evaluate our model in transfer learning, we construct two cross-channel datasets from Amazon.com[4] and Amazon Alexa[5] that contain customers' purchase records on both Web and Voice.[6]

- **Amazon Grocery:** This is a proprietary grocery shopping dataset collected from Amazon. It contains a sample of customers' grocery shopping histories on both Web and Voice.
- **Amazon Home:** This is a proprietary home products shopping dataset collected from Amazon (e.g., water filter, coffee maker). It contains a sample of customers' shopping histories of home products on both Web and Voice.

For every user, we create sliding windows of $N_w + 2$ weeks with a step size of 2 weeks from her historical shopping record. TransV consumes purchases from the first N_w weeks and generates recommendations for the last 2 weeks. Each sliding window of a user is considered as a sample. We split the samples into a training and a test set based on the timestamps of the purchases such that the purchases in the last 2 weeks of the test samples do not appear in the training samples. Afterward, we filter the dataset by only keeping items purchased by more than 10 unique customers. For the Amazon datasets, we only keep samples with at least one purchase on Voice in the test set. Key statistics from public and proprietary datasets are presented in Table 1 and 2 respectively.

Comparison Methods. To demonstrate the effectiveness of our method in modeling repeated purchases, we compare with the following state-of-the-art methods: 1) **Bi-LSTM** [15]: a deep learning model based on bi-directional LSTM, 2) **Transformer** [22, 41]: a recommendation model based on Transformer that adopts the self-attention mechanism, 3) **adaLoyal Transformer**: a variant of Transformer where we apply **adaLoyal** [42] on top of **Transformer** for modeling repeated purchases, 4) **RepeatNet** [36]: an RNN based method that models repeated purchases using an explore-repeat mode switch which integrates a copy mechanism that chooses item from historical purchases. For fair comparison, we use the same attention based pooling layer and the inner product interaction layer for all the above comparison methods and our model. To ablate the effect of transfer learning, the compared methods are all trained on combined samples from Web and Voice – as repeated purchases are a cross-platform behavior – without learning channel-specific interactions. As an additional baseline, we also compare with **user-itemPop** which only relies on the customer-wise empirical frequency of products for recommendation.

To investigate the effectiveness of our multi-level interaction module for transfer learning, we compare the following variants of TransV: 1) **Non Transfer**: the variant that only utilizes voice data for recommendation, 2) **Direct Transfer**: the variant that does not learn channel-specific interactions, i.e., the same configuration used in comparing methods for modeling repeated purchases, 3) **Tri-Factor**: the variant where tri-factorization is used to model interactions across Web and Voice, 4) **TransV**: the variant where our proposed multi-level interaction module is used for the transfer.

Evaluation Protocols. We measure the performance of the compared methods using two metrics 1) Normalized Discounted Cumulative Gain (NDCG) at $K = \{1, 5, 10\}$ and 2) Area Under the ROC Curve (AUC). AUC captures the overall ranking performance by comparing customer purchased products with non-purchased ones in a pairwise manner, regardless of the position of the compared pair in the generated ranking list. Unlike AUC, NDCG@K weights the top-ranking positions as more important than the others. Such a difference makes NDCG@K (especially when $K = 1$) a more suitable metric for voice-based recommendation, where the top recommendations need to be highly precise due to the narrow information channel. We note that results reported in this section for the two cross-channel datasets are all obtained from Voice.

Parameter Settings. We empirically set optimal parameters based on a head-out validation set that contains 10% of the test data. For all methods, the dimension of the embedding is set to 64. For the learning rate and regularization weight, we apply a grid search in $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. The dropout rate is selected from the set $\{0.0, 0.1, 0.2, 0.3\}$. To find the optimal default loyalty l_0 for adaLoyal, we apply grid search in $\{0.1, 0.2, ..., 0.9\}$. For Transformer models, we set the number of attention heads to 8. To keep the most recent purchases, we set the maximum sequence length of historical purchases as follows: $N = 300$ for Dunnhumby, $N = 200$ for Instacart and $N = 60$ for Amazon Grocery and Home. All the models

[2] https://www.dunnhumby.com/careers/engineering/sourcefiles

[3] https://www.instacart.com/datasets/grocery-shopping-2017

[4] https://www.amazon.com

[5] https://developer.amazon.com/alexa

[6] To protect customers, all data from Amazon.com and Alexa were anonymized and un-identified before the experiment and analysis were performed.

Table 3: Repeated purchase recommendation performance of all comparison methods on the two public datasets. The best performance is boldfaced.

Dataset	Metric	user-itemPop	Bi-LSTM	Transformer	adaLoyal Transformer	RepeatNet	TransV
Dunnhumby	NDCG@1	0.4843	0.3896	0.4252	0.4777	0.4892	**0.4972**
	NDCG@5	0.3871	0.3018	0.3245	0.3838	0.3991	**0.4060**
	NDCG@10	0.3357	0.2579	0.2745	0.3341	0.3468	**0.3523**
	AUC	0.6823	0.8328	0.8321	**0.8481**	0.8313	0.8313
Instacart	NDCG@1	0.6712	0.5549	0.5892	0.6666	0.6795	**0.6951**
	NDCG@5	0.5825	0.4415	0.4728	0.5826	0.5906	**0.6066**
	NDCG@10	0.5372	0.3919	0.4167	0.5383	0.5465	**0.5616**
	AUC	0.8137	0.9584	0.9593	**0.9659**	0.9542	0.9575

Table 4: Repeated purchase recommendation performance of all comparison methods on the two cross-channel datasets tested with voice purchases. Results of `TransV` are obtained with direct transfer. The numbers are relative ratio of the performance with respect to that of user-itemPop. The best performance is boldfaced.

Dataset	Metric	user-itemPop	Bi-LSTM	Transformer	adaLoyal Transformer	RepeatNet	TransV (direct transfer)
Grocery	NDCG@1	1.0000	1.0051	1.0115	1.0186	1.0572	**1.0823**
	NDCG@5	1.0000	0.8969	0.9101	1.0085	1.0525	**1.0674**
	NDCG@10	1.0000	0.8751	0.8878	1.0075	1.0487	**1.0610**
	AUC	1.0000	1.0701	1.0704	1.0794	1.0795	**1.0801**
Home	NDCG@1	1.0000	1.1129	1.1097	1.3270	1.4346	**1.4715**
	NDCG@5	1.0000	0.8791	0.8675	1.1086	1.2180	**1.2252**
	NDCG@10	1.0000	0.8683	0.8642	1.1003	1.2020	**1.2061**
	AUC	1.0000	1.4614	1.4672	1.4701	1.4818	**1.4838**

are implemented with *MXNet*[7]. Model training is performed using Adagrad [13] with mini-batches of size 128. All the gradients are clipped between −10 and 10 to prevent exploding [3].

4.2 Results on Repeated Purchase (Q1)

We start by investigating the effectiveness of our proposed repeated purchase module by comparing it against user-itemPop, Bi-LSTM, Transformer, adaLoyal Transformer and RepeatNet. Results on public and proprietary datasets are reported in Table 3 and 4, respectively. In Table 4, the performance is shown in terms of relative ratio with respect to that of user-itemPop tested on voice purchases.

We observe that Bi-LSTM and Transformer achieve better performance than user-itemPop when measured by AUC however are outperformed by user-itemPop when measured by NDCG@K. Recall that Bi-LSTM and Transformer are general collaborative filtering approaches that do not explicitly model customers' repeated purchase behaviors. The result indicates that while these methods are generally effective in recommendation, they are not suitable for precise recommendation where the top-ranked products need to be highly relevant. Unlike Bi-LSTM and Transformer, methods that consider repeated purchases such as RepeatNet and `TransV`, achieve higher performance than user-itemPop. Such a comparison clearly demonstrate the need to account for repeated purchases in recommendation.

Among these methods, we observe that RepeatNet and `TransV` outperform adaLoyal Transformer on Amazon Grocery and Amazon Home across both types of metrics; and on Dunnhumby and Instacart, they outperform adaLoyal Transformer when performance is measured by NDCG@K. Recall that adaLoyal Transformer takes a post-processing approach for modeling repeated purchases: it re-ranks the result from the general collaborative filtering method Transformer by considering customers' historical product purchases; RepeatNet and `TransV`, on the other hand, consider repeated purchases as an integral part in modeling customers' purchase behaviors. The comparison results demonstrate the benefit of the latter approach for precise recommendation. This can be explained by its capability of accurately capturing the effect of repeated purchases in customers' purchase behaviors, which helps generate recommendations that are more relevant.

Our proposed method `TransV` achieves the best NDCG@K across all the datasets and highest AUC on the two cross-channel datasets. In particular, `TransV` consistently outperforms RepeatNet on both datasets across both metrics. This is mainly due to the effectiveness of considering time bias in modeling repeated purchases, which we discuss further in section 4.4. Overall, `TransV` outperforms RepeatNet by 1.97% for the Web datasets (averaged over NDCG@K for $K = \{1, 5, 10\}$; p-value $< .001$, WilCoxon signed-rank test) and by 2.47% for voice-based recommendation on the cross-channel datasets in terms of NDCG@1 (p-value $< .001$ on Grocery and $< .01$ on Home, WilCoxon signed-rank test).

[7] https://mxnet.apache.org/

Table 5: Transfer learning performance on the two proprietary datasets. The numbers are relative ratio of the performance with respect to that of user-itemPop. The best performance is boldfaced.

Dataset	Metric	No Transfer	Direct Transfer	Tri-factor	TransV
Grocery	NDCG@1	0.9966	1.0823	1.0902	**1.1041**
	NDCG@5	0.8969	1.0674	1.0806	**1.0848**
	NDCG@10	0.8665	1.0610	1.0725	**1.0764**
	AUC	1.0368	1.0801	**1.0829**	1.0823
Home	NDCG@1	1.0707	1.4715	1.4989	**1.5295**
	NDCG@5	0.8595	1.2252	1.2889	**1.3063**
	NDCG@10	0.8649	1.2061	1.2840	**1.2976**
	AUC	1.3738	1.4838	1.4938	**1.4949**

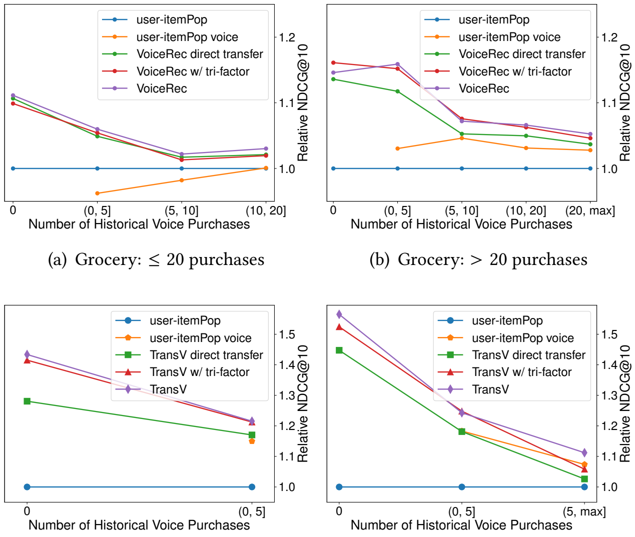
4.3 Results on Transfer Learning (Q2)

We evaluate our proposed transfer learning module. To understand the impact of the relative sparsity of Web data with respect to that of voice data on the effectiveness of transfer learning, we first divide customers into groups of different historical numbers of purchases on Web; then for each of the group, we further divide customers into groups according to their number of purchases on Voice.

The overall results on the two cross-channel datasets, i.e., Amazon Grocery and Home, are reported in Table 5. We observe that Direct Transfer is outperformed by Tri-factor, which is further outperformed by our proposed approach TransV. The result signifies the importance of transferring the interaction patterns between customers and products for voice-based recommendation. More importantly, the superior performance obtained by TransV compared with Tri-factor signifies the advantage of multi-level interaction in Web-to-Voice transfer for taking into account the distinct purchase patterns of customers on Voice.

Compared with non-transfer learning (Table 5), TransV substantially improves voice-based recommendation by 26.81%, 36.47%, and 37.13% for NDCG@K for $K = \{1, 5, 10\}$, respectively.

Impact of Data Sparsity. Results on customer groups with different numbers of purchases on Web and on Voice are depicted in Figure 2. We first note that applying user-itemPop on Voice (user-itemPop voice in the figure) results in different performance for voice-based recommendation than regular user-item Pop which is applied on purchases from both channels; moreover, it generally has better performance. This confirms that customers' shopping behavior on Voice is different from that on Web and in particular, they tend to have more repeated purchases on Voice. Comparing the transfer learning performance on customers with different numbers of historical purchases on Voice (subgroups in Figure 2(a-d)), we observe that transfer learning is most beneficial for the customers with the least number of historical purchases on Voice. Finally, we observe that for customers with a similar number of voice purchases, transfer learning is more beneficial for customers with more purchases on the Web (Figure 2(a) vs. (b), and (c) vs. (d)). These results demonstrate that Web data indeed can largely alleviate the data sparsity issue in voice-based recommendation.



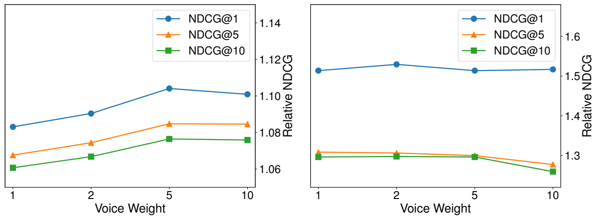
(a) Grocery: ≤ 20 purchases

(b) Grocery: > 20 purchases

(c) Home: ≤ 5 purchases

(d) Home: > 5 purchases

Figure 2: Transfer learning performance on customer groups with different number of historical voice purchases. The performance is measured by relative ratio of NDCG@10 with respect to that of user-itemPop. Note that the method "user-itemPop voice" does not apply to customers without voice purchase history.

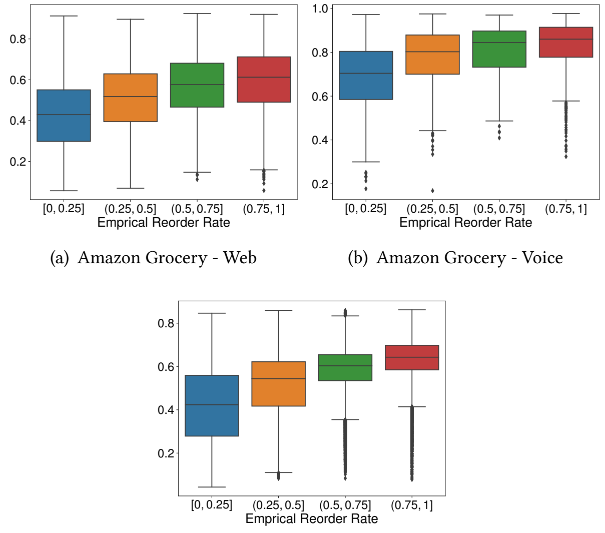


(a) Grocery

(b) Home

Figure 3: Recommendation performance with different voice weights.

Impact of Voice Weight. In transfer learning, TransV weights the importance of the recommendation tasks for different channels with the hyperparamter α^d in its loss function. To investigate the impact of the weight, we conduct an experiment with different weight values on Amazon Grocery and Home datasets. We fixed the task weight for Web as 1 and apply a grid search in $\{1, 2, 5, 10\}$ for the weight for recomendation on Voice. Figure 3 shows the performance measured by NDCG@K. We observe that as the weight for voice increases, the performance first increases then decreases. The best performance is achieved when weight for Voice is 5 for Amazon Grocery and 2 for Amazon Home. This result suggests that with an appropriate setting for the weight of recommendation on Voice, our approach can effectively transfer customers' purchase patterns from Web-to-Voice while taking into account the unique characteristics of voice shopping. Besides, the similarity in performance variation across α^d values on the two datasets shows the robustness of TransV.



(a) Amazon Grocery - Web

(b) Amazon Grocery - Voice

(c) Instacart

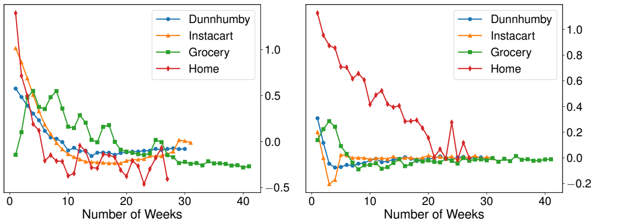
Figure 4: Predicted mixture weight against empirical repeated purchase rate. Note that we only include samples with no less than two purchases to reduce noise.

4.4 Properties of TransV (Q3)

To further show how TransV works, we conduct an in-depth analysis of the properties of TransV. We show how TransV can strike a balance between modeling general and repeated purchases for effective recommendation and can uncover a time decay phenomenon of the effect of historical purchases on repeated purchases.

Learning Mixture Weight. TransV learns the mixture weight π to capture the importance of repeated purchases in customers' shopping decisions. Being able to learn such mixture weight is important for generating recommendations of high-relevance. For comparison, Figure 4 shows the learned mixture weights against the empirical repeated purchase rate for Amazon Grocery and Instacart on the test dataset. We observe that the learned mixture weight correlates positively with the empirical repeated purchase rate. This demonstrates the effectiveness of TransV in striking a balance between modeling general and repeated purchases for recommendation. In particular, we observe that for Amazon Grocery the predicted mixture weight on Voice is higher than that on Web, which corresponds well with the statistics of the dataset (Table 2). As a remark, we note though that the predicted mixture weight does not reflect the exact empirical repeated purchase rate. This is likely due to the fact that general preference also contributes to repeated purchases and the current model cannot capture all the discriminant factors. We leave the improvement to future work.

Learning Time Bias. TransV learns time bias functions $T(\cdot)$ and $S(\cdot)$ to weight the importance of the recency of historical purchases on repeated purchases. $T(\cdot)$ captures the *relative* importance within the historical purchases, while $S(\cdot)$ captures the importance with respect to general collaborative filtering. On our experimental dataset, these functions are learned as piecewise constant functions on equal-spaced bins each representing a week.



(a) Time bias function $T(\cdot)$

(b) Time bias function $S(\cdot)$

Figure 5: The learned time bias functions $T(\cdot)$ (a) and $S(\cdot)$ (b) from the four experimental datasets. The time axis describes the number of weeks passed from the historical purchase till the prediction time.

Figure 5(a) shows the learned time bias $T(\cdot)$. We observe a general pattern that the importance of historical purchases decreases when the purchase occurs further in the past. The result implies that recently purchased products are more likely to be repeatedly purchased. We also notice that the time bias function for Amazon Grocery shows a slightly different pattern: it first increases then decreases. This can be explained by the fact that grocery products are generally consumed for a few weeks before being repeatedly purchased, e.g., trash bags, drinks. Interestingly, we observe a specific periodic pattern on the two cross-channel datasets, and the local peaks occur on a monthly basis. This is due to the subscription feature provided by Amazon where customers can elect to subscribe to products monthly.

Similar results are observed for $S(\cdot)$ from Figure 5(b): rrecently purchased products are more likely to be repeatedly purchased, thus playing a more important role in the recommenation generation. We note that $S(\cdot)$ of Instacart decreases for the first several weeks and then approaches zero. This is because customers with greater than the maximum gap (30 days) between two consecutive purchases are filtered out. This results in a skewed distribution of time since the last purchase, leading to the diminishing of $S(\cdot)$ after several weeks.

5 CONCLUSION

We presented TransV, a neural transfer network that addresses the data sparsity issue of voice-based recommendation by transferring customers' shopping patterns from the Web to Voice. It employs multi-level tri-factorization to capture the similarity and dissimilarity of customers' shopping patterns on the Web and Voice, thereby allowing effective Web-to-Voice transfer, while taking into account distinct voice shopping patterns. TransV is seamlessly integrated with a recency-based copy mechanism to capture the prevalent behavior of repeated purchases on Voice. Our extensive evaluation on multiple real-world datasets, including two cross-channel datasets from Amazon, shows that TransV significantly improves the performance of voice-based recommendation. Our analysis further offers valuable insights into customers' voice shopping behaviors, e.g. recent purchases are more likely to be repeated. As future work, we plan to study how to integrate TransV with natural language generation for conversational recommendations.

REFERENCES

[1] Ashton Anderson, Ravi Kumar, Andrew Tomkins, and Sergei Vassilvitskii. 2014. The dynamics of repeat consumption. In *Proceedings of the 23rd International Conference on World Wide Web*. ACM, 419–430.

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).

[3] Yoshua Bengio, Nicolas Boulanger-Lewandowski, and Razvan Pascanu. 2013. Advances in optimizing recurrent networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 8624–8628.

[4] Austin R Benson, Ravi Kumar, and Andrew Tomkins. 2016. Modeling user consumption sequences. In *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 519–529.

[5] Rahul Bhagat, Srevatsan Muralidharan, Alex Lobzhanidze, and Shankar Vishwanath. 2018. Buy It Again: Modeling Repeat Purchase Recommendations. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 62–70.

[6] Bin Cao, Nathan N Liu, and Qiang Yang. 2010. Transfer learning for collective link prediction in multiple heterogenous domains. In *Proceedings of the 27th International Conference on Machine Learning*. Citeseer, 159–166.

[7] Konstantina Christakopoulou, Alex Beutel, Rui Li, Sagar Jain, and Ed H Chi. 2018. Q&R: A two-stage approach toward interactive recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 139–148.

[8] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards conversational recommender systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 815–824.

[9] OC&C Strategy Consultants. 2018. Voice Shopping Set to Jump to $40 Billion By 2022, Rising From $2 Billion Today. *[online; posted 28-Febuary-2018]* (2018).

[10] John Dawes, Lars Meyer-Waarden, and Carl Driesener. 2015. Has brand loyalty declined? A longitudinal analysis of repeat purchase behavior in the UK and the USA. *Journal of Business Research* 68, 2 (2015), 425–432.

[11] Bhuwan Dhingra, Lihong Li, Xiujun Li, Jianfeng Gao, Yun-Nung Chen, Faisal Ahmed, and Li Deng. 2016. Towards end-to-end reinforcement learning of dialogue agents for information access. *arXiv preprint arXiv:1609.00777* (2016).

[12] Chris Ding, Tao Li, Wei Peng, and Haesun Park. 2006. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 126–135.

[13] John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *The Journal of Machine Learning Research* 12 (2011), 2121–2159.

[14] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 278–288.

[15] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. 2013. Hybrid speech recognition with deep bidirectional LSTM. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE, 273–278.

[16] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393* (2016).

[17] 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.

[18] Guangneng Hu, Yu Zhang, and Qiang Yang. 2018. Conet: Collaborative cross networks for cross-domain recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 667–676.

[19] Amir Ingber, Arnon Lazerson, Liane Lewin-Eytan, Alexander Libov, and Eliyahu Osherovich. 2018. The Challenges of Moving from Web to Voice in Product Search. In *Proceedings of the 1st International Workshop on Generalization in Information Retrieval*.

[20] Jacob Jacoby and David B Kyner. 1973. Brand loyalty vs. repeat purchasing behavior. *Journal of Marketing research* 10, 1 (1973), 1–9.

[21] Heishiro Kanagawa, Hayato Kobayashi, Nobuyuki Shimizu, Yukihiro Tagami, and Taiji Suzuki. 2019. Cross-domain recommendation via deep domain adaptation. In *European Conference on Information Retrieval*. Springer, 20–29.

[22] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining*. IEEE, 197–206.

[23] Tom Kenter and Maarten de Rijke. 2017. Attentive memory networks: Efficient machine reading for conversational search. *arXiv preprint arXiv:1712.07229* (2017).

[24] Bin Li, Qiang Yang, and Xiangyang Xue. 2009. Transfer learning for collaborative filtering via a rating-matrix generative model. In *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 617–624.

[25] Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards deep conversational recommendations. In *Advances in Neural Information Processing Systems*. 9725–9735.

[26] Jianxun Lian, Fuzheng Zhang, Xing Xie, and Guangzhong Sun. 2017. CCCFNet: a content-boosted collaborative filtering neural network for cross domain recommender systems. In *Proceedings of the 26th International Conference on World Wide Web companion*. International World Wide Web Conferences Steering Committee, 817–818.

[27] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130* (2017).

[28] Nathan N Liu, Evan W Xiang, Min Zhao, and Qiang Yang. 2010. Unifying explicit and implicit feedback for collaborative filtering. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. ACM, 1445–1448.

[29] Babak Loni, Yue Shi, Martha Larson, and Alan Hanjalic. 2014. Cross-domain collaborative filtering with factorization machines. In *European Conference on Information Retrieval*. Springer, 656–661.

[30] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. 2016. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3994–4003.

[31] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A Three-Way Model for Collective Learning on Multi-Relational Data.

[32] Weike Pan, Nathan N Liu, Evan W Xiang, and Qiang Yang. 2011. Transfer learning to predict missing ratings via heterogeneous user feedbacks. In *Twenty-Second International Joint Conference on Artificial Intelligence*.

[33] Weike Pan, Evan Wei Xiang, Nathan Nan Liu, and Qiang Yang. 2010. Transfer Learning in Collaborative Filtering for Sparsity Reduction. In *Twenty-fourth AAAI Conference on Artificial Intelligence*, Vol. 10. 230–235.

[34] Pascale Quester and Ai Lin Lim. 2003. Product involvement/brand loyalty: is there a link? *Journal of Product & Brand Management* 12, 1 (2003), 22–38.

[35] Filip Radlinski and Nick Craswell. 2017. A theoretical framework for conversational search. In *Proceedings of the 2017 Conference on Human Information Interaction and Retrieval*. ACM, 117–126.

[36] Pengjie Ren, Zhumin Chen, Jing Li, Zhaochun Ren, Jun Ma, and Maarten de Rijke. 2018. RepeatNet: A Repeat Aware Neural Recommendation Machine for Session-based Recommendation. *arXiv preprint arXiv:1812.02646* (2018).

[37] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th International Conference on World Wide Web*. ACM, 111–112.

[38] Ajit P Singh and Geoffrey J Gordon. 2008. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 650–658.

[39] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in Neural Information Processing Systems*. 2440–2448.

[40] Yueming Sun and Yi Zhang. 2018. Conversational recommender system. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 235–244.

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[42] Mengting Wan, Di Wang, Jie Liu, Paul Bennett, and Julian McAuley. 2018. Representing and Recommending Shopping Baskets with Complementarity, Compatibility and Loyalty. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 1133–1142.

[43] Adam Wright, Jitesh Ubrani, and Michael Shirer. 2019. Double-Digit Growth Expected in the Smart Home Market, Says IDC. *IDC [online; posted 29-March-2018]* (2019).

[44] Liu Yang, Minghui Qiu, Chen Qu, Jiafeng Guo, Yongfeng Zhang, W Bruce Croft, Jun Huang, and Haiqing Chen. 2018. Response ranking with deep matching networks and external knowledge in information-seeking conversation systems. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 245–254.

[45] Longqi Yang, Michael Sobolev, Christina Tsangouri, and Deborah Estrin. 2018. Understanding user interactions with podcast recommendations delivered via voice. In *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 190–194.

[46] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *Comput. Surveys* 52, 1 (2019), 5.

[47] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W Bruce Croft. 2018. Towards conversational search and recommendation: System ask, user respond. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 177–186.

[48] Yu Zhang and Qiang Yang. 2017. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114* (2017).

[49] Chang Zhou, Jinze Bai, Junshuai Song, Xiaofei Liu, Zhengchao Zhao, Xiusi Chen, and Jun Gao. 2017. ATRank: An Attention-Based User Behavior Modeling Framework for Recommendation. *arXiv preprint arXiv:1711.06632* (2017).