Product Choice with Large Assortments: A Scalable Deep-Learning Model

by

Sebastian Gabel* and Artem Timoshenko†

June 2020

Abstract

Personalized marketing in retail requires a model to predict how different marketing actions affect product choices by individual customers. Large retailers often handle millions of transactions daily, involving thousands of products in hundreds of categories. Product choice models thus need to scale to large product assortments and customer bases, without extensive product attribute information. To address these challenges, we propose a custom deep neural network model. The model incorporates bottleneck layers to encode cross-product relationships, calibrates time-series filters to capture purchase dynamics for products with different interpurchase times, and relies on weight sharing between the products to improve convergence and scale to large assortments. The model applies to loyalty card transaction data without predefined categories or product attributes to predict customer-specific purchase probabilities in response to marketing actions. In a simulation, the proposed product choice model predicts purchase decisions better than baseline methods by adjusting the predicted probabilities for the effects of recent purchases and price discounts. The improved predictions lead to substantially higher revenue gains in a simulated coupon personalization problem. We verify predictive performance using transaction data from a large retailer with experimental variation in price discounts.

Keywords: Product Choice Model; Neural Networks; Deep Learning; Cross-Category Choice; Retail Analytics

* SO1 GmbH and Humboldt University Berlin, email: gabel@so1.ai

† Kellogg School of Management, Northwestern University, email: artem.timoshenko@kellogg.northwestern.edu

1. Introduction

Many retailers leverage personalization to promote products and categories, stimulate incremental purchases, and improve customer retention. For example, retailers can offer personalized pricing and promotions (Dubé and Misra 2017; Zhang and Wedel 2009), target different advertising campaigns to different customer segments (Ansari and Mela 2003), provide personalized product recommendations (Linden et al. 2003), or customize online and offline in-store experience (Hauser et al. 2009; Lu et al. 2016).

Coupon personalization is an important example of personalized marketing. In 2018, U.S. retailers distributed over 256.5 billion coupons for consumer packaged goods (NCH Marketing Services 2019). Less than 1% of the issued coupons were redeemed, and customers often redeemed coupons for products which they would have purchased at a regular price (Forrester 2017). To increase coupon profitability, retailers adopt solutions to provide personalized coupons to individual customers: CVS provides personalized coupons at the store entrance through kiosk systems, Food Lion (Ahold Delhaize) offers coupons for the next visit at the checkout, and Whole Foods distributes coupons via its mobile application. Such personalization solutions typically rely on loyalty card transaction data and require a model to predict how different marketing actions affect the product choices by individual customers (Arora et al. 2008).

In our conversations with major retailers and solution providers in the U.S. and Europe, practitioners frequently emphasized that implementing models to predict purchase behavior across the entire assortment can be challenging. Current product choice models used for coupon optimization generally adopt a brand perspective and focus on a single product category (e.g., Rossi et al. 1996; Johnson et al. 2013). Models require careful calibration; a modeler must delineate categories, prune input data, define choice sets, and collect product attributes. However, large retailers handle millions of transactions per day and stock tens of thousands of products across hundreds of product categories (Walmart 2005, 2016). Implementing and maintaining separate models for categories or individual products is hardly feasible. Even if retailers were able to implement the models in parallel, ignoring cross-category product relationships would lead to suboptimal targeting decisions (Smith et al. 2019).

In this paper, we develop a scalable product choice model that predicts customer-specific purchase probabilities for all products in the assortment in response to personalized coupon discounts. The model is based on a custom deep learning architecture which inputs purchase histories of individual customers and coupon assignments to predict product choice. The inputs can additionally incorporate information relevant for targeting, such as customer demographics, shopping trip data, and other marketing mix

variables. Our work directly addresses three practical challenges outlined above: the model (1) applies to raw transaction data from loyalty programs, without requiring category structure or product attribute information, (2) scales to large product assortments and customer bases, and (3) accounts for cross-product relationships and the effects of marketing interventions within and across categories.

To achieve scalability to large product assortments, the proposed neural network architecture keeps most transformations product-specific and shares weights between the by-product transformations (Alain and Bengio 2014). The parsimonious model architecture has a regularization effect and improves training speed and convergence. We incorporate flexible by-product transformations to enable weight sharing. For example, the model automatically calibrates time-series filters to efficiently summarize sparse purchase history inputs for products with different interpurchase times.

The proposed model captures cross-product relationships using the bottleneck layers. The bottleneck layers create dense product representations and help the model adjust the predicted probabilities for cross-product effects of discounts and category-level purchase patterns. For example, the model automatically infers that discounts for Coke and Pepsi have similar effects on the purchase likelihoods of other soft drinks and no effect on the purchase likelihood of detergents. It also infers that purchasing detergent today affects purchase likelihoods for the entire detergent product category next week.

We evaluate the proposed product choice model using synthetic and empirical data. The synthetic data facilitates in-depth analyses of the proposed model by comparing the predicted probabilities to the true data generating process. The empirical data verifies performance improvements in a real-life application. Both settings leverage experimental variation in price discounts at the customer level, such that both products and the depth of the discounts are randomized.

Our simulation generates a retailer with many products across multiple categories. The purchase decisions follow a two-stage process: Customers decide whether to purchase a product from a category, then choose products within the selected categories. We assume customer heterogeneity and category-specific consumption dynamics. Customers receive coupons with price discounts in each time period. Each coupon affects own-product purchase probability, purchase probabilities of other products within a category, and purchase probabilities for related categories. We calibrate the simulation study such that it closely resembles the empirical data and includes mechanisms relevant for product choice modeling and coupon optimization. We only use information about customer purchases and coupon assignments to train the product choice model and assume that other information such as category structure is unobserved ex ante.

The simulation study validates that our model (1) accurately predicts purchase probabilities for all products in the assortment, (2) dynamically adjusts the predicted probabilities for customer-specific recent purchases and consumption patterns, and (3) recovers own- and cross-product coupon effects.

We further use the simulation to demonstrate the value of the proposed product choice model for coupon personalization. The higher predictive accuracy of our product choice model leads to substantially higher revenue and purchase incidence gains through coupon personalization. Coupon personalization leverages better estimates of the own-product and cross-product discount effects, and our model is especially effective when the cross-category discount effects are more pronounced.

We complete the model validation by evaluating the predictive performance of the model with transaction data from a leading German grocery retailer. The retailer randomly distributed coupons with product discounts to a small fraction of customers. Experimental data allows training and evaluating the model without endogeneity concerns. In line with the results obtained from the simulated data, the model achieves higher out-of-sample predictive accuracy than baseline models. The outperformance margins, relative to the reference models, are particularly large for observations shortly after a category purchase and for observations with discounts.

The proposed product choice model offers high practical value for retail analytics. The model is scalable to large assortments and customer bases. The implementation does not depend on category definitions or product attributes and can incorporate additional purchase-related information. Furthermore, the model generates predictions for new observations quickly, in a single feed-forward pass. These characteristics provide a basis for retail analytics problems that require quantifying how marketing decisions affect business performance based on transaction data (Hanssens 2014).

The paper proceeds in Section 2 with a review of related literature. We introduce the proposed product choice model in Section 3. Section 4 describes the simulation setup. We use simulated data to evaluate the predictive performance of the proposed model and demonstrate its value for coupon personalization in Section 5. Section 6 validates the predictive performance using empirical data. We summarize our findings and suggest directions for future research in Section 7.

2. Related Literature

Our research relates to three streams of literature: product choice modeling, methods for targeting and coupon optimization, and deep learning applications in marketing. We next discuss each of these areas and highlight our respective contributions.

2.1. Product Choice Modeling

Product choice models quantify how marketing actions affect business outcomes, such as market share or profits; Winer and Neslin (2014) provide a comprehensive overview of the product choice literature. The ability to predict the effects of marketing activities forms a basis for efficient resource allocation (Hanssens 2014).

Traditionally, product choice models estimate purchase decisions for a single product, brand, or category, using logit (Guadagni and Little 1983), nested logit (Kamakura and Russell 1989), or random coefficient logit specifications (Chintagunta 1993). For example, Fader and Hardie (1996) propose a latent class multinomial logit model to predict customer choices for 56 products in the fabric softener category. They represent products as combinations of attributes (e.g., brand, package size) and demonstrate that their model significantly outperforms a model specification with 55 product fixed effects. Attribute-based choice models can achieve better predictive performance, but they also require retailers to maintain comprehensive product attribute data for each category-level model—a complex and laborious task, especially for large assortments. Our proposed product choice model infers product similarities directly from customer-level transaction data. The neural network represents products using low-dimensional vectors (embedding), and a common product embedding space makes products comparable. This approach does not require manual definitions of product attributes.

Models that study multi-category product choice include multivariate probit and multivariate logit models (Manchanda et al. 1999; Russell and Peterson 2000). Multivariate choice models incorporate cross-category relationships using additional parameters for combinations of categories. The number of parameters needed to capture cross-category relationships in the multivariate choice models limits their scalability. For example, Manchanda et al. (1999) and Russell and Peterson (2000) each study four product categories. Our proposed model instead encodes product relationships within and across categories, using real-valued parameters of the bottleneck layers. With the dense representations, we can model hundreds of product categories simultaneously and scale the model to the size of typical retail applications (Amano et al. 2019).

Machine learning approaches for product choice modeling are gaining more attention in marketing. For example, Jacobs et al. (2016) propose LDA-X to predict customer-specific purchase probabilities for products in the assortment of an online retailer. LDA-X first infers small-dimensional customer embeddings from the data through MCMC, and then uses the customer embeddings to inform predictions of future purchases. Ruiz et al. (2018) propose the SHOPPER model, which sequentially predicts purchase probabilities for products from multiple product categories, given the current content of a shopping cart. SHOPPER describes products through latent attributes (embeddings) that capture product

characteristics and product relationships. Both LDA-X and SHOPPER account for customer heterogeneity and are more scalable than classic discrete choice models. Our model is specifically designed to estimate individual responses to marketing actions. The model inputs marketing mix variables and individual purchase histories to predict customer-specific conditional purchase probabilities. The proposed neural network architecture supports implementation in established deep learning frameworks, so retailers with prior expertise in deep learning can easily adopt our model to their practice. For example, the retailer that provided the data for our empirical application uses neural networks in supply chain management and is likely to apply our approach to marketing problems.

2.2. Coupon Personalization and Targeting

Our product choice model is motivated by the coupon personalization problem (Fader 2012; Peppers and Rogers 1997). Coupon personalization and targeting are important topics in marketing research and practice (Bradlow et al. 2017; Grewal et al. 2017). Rossi et al. (1996) propose a model to derive profit-maximizing coupon personalization policies and highlight the value of household purchase histories for optimizing coupon profitability. Johnson et al. (2013) add a temporal dimension to coupon personalization and validate the proposed method in several brand-management applications. Zhang and Wedel (2009) jointly model purchase incidence, product choice, and quantity decisions in online and offline stores to maximize brand profit through promotion customization. Dubé and Misra (2017) propose a machine learning approach for price personalization and apply it to subscription pricing at an online recruiting company.

Coupon personalization solutions require a product choice model and an optimization approach. The product choice model predicts how different combinations of coupons affect individual purchasing behavior, and the coupon optimization approach allocates coupons, given the predicted effects. Our research develops a product choice model that predicts the impact of coupons on purchasing probabilities for the entire assortment of a large retailer, including within- and cross-category coupon effects. In Section 5.4, we evaluate our model by estimating the expected profits of a simulated retailer that allocates coupons according to different underlying product choice models.

We can compare our product choice model to model-based recommender systems. Model-based recommender systems identify the products which customers are most likely to consider or purchase by predicting customer behavior conditional on different product recommendations (Breese et al. 2013). Our model predicts how customer-specific purchase probabilities change in response to marketing actions by a retailer, which can serve as input to model-based recommender systems. In Section 5, we compare our model with two baselines incorporating cosine similarity between the customer and product

representation as a feature, which is a standard approach in recommender systems literature (Koren et al. 2009; Levy and Goldberg 2014).

The basis for training and evaluation of the models in our paper is experimental data. Our simulation and the empirical application assign coupons to customers at random. Random coupon assignment allows training the prediction model without endogeneity concerns. We validate the coupon optimization approaches in the simulation using a randomization-by-policy experimental design (Simester et al. 2019b). In particular, we evaluate coupon personalization by implementing different algorithms to assign coupons to different groups of customers (or equivalently using independent simulation runs).

2.3. Deep Learning Applications in Marketing

The proposed product choice model is based on a neural network. Neural network models have achieved remarkable performance in computer vision and natural language processing applications (LeCun et al. 2015). Marketing researchers have recently started to apply deep neural networks to marketing problems. For example, Liu, Lee, and Srinivasan (2017) develop an approach to automatically extract content information from online product reviews and predict conversion. Timoshenko and Hauser (2019) propose a deep learning framework to enable firms to identify customer needs from online reviews more efficiently. Zhang and Luo (2018) use deep learning to extract sentiments from photos and reviews posted on Yelp and find that sentiments predict restaurant survival, even after controlling for other covariates. Liu, Dzyabura, and Mizik (2018) apply deep convolutional neural networks to social media images with a goal to measure the consumers' perception of brands. Gabel et al. (2019) propose a neural network-based method to map market structures in grocery retailing based on market basket data.

Deep neural networks are particularly well-suited for applications involving loyalty card data. First, deep learning methods can handle large volumes of training data (Goodfellow et al. 2016). Large retailers process millions of transactions daily, which creates an enormous amount of data for model calibration. Second, deep learning models can effectively operate with high-dimensional inputs. Consider purchasing histories as a single input. If there are 2,500 products in a retail assortment and a 30-week history window, a purchase history for a single customer contains 75,000 values, comparable to 256×256 images often used in computer vision applications (Krizhevsky et al. 2012). The sequential nature of purchasing histories also resembles the structure of words in texts in natural language processing tasks (Collobert et al. 2011).

Our contribution is a custom neural network architecture to model dynamic customer-specific purchase probabilities for products in large assortments. We incorporate time-series filters to process sparse purchase history information. Calibrating time-series filters within a model allows using the same summary statistics for products with very different interpurchase times. In Section 5.3.6, we demonstrate

that this approach performs better than using manually-defined frequency-based measures (Fader et al. 2005). The proposed architecture also leverages the bottleneck layers to create low-dimensional representations for purchase histories and coupon allocations and capture cross-product relationships. The encoder-decoder structure of the bottleneck layers relates to autoencoder approaches (Vincent et al. 2008). However, our model uses the bottleneck layers to improve prediction of the purchase events, without reconstructing the encoder inputs. We provide an in-depth evaluation of different components of the model architecture in the nested model analysis.

3. A Proposed Cross-Category Product Choice Model

We present our model in a context of a coupon personalization problem and discuss other potential retail analytics applications in Section 3.4.

3.1. Overview

Consider a retail store operating J products. The products may be related both in terms of cross-price elasticities and purchase co-incidence (Manchanda et al. 1999). The relationship among the products is unknown *ex ante*.

There are I customers who shop at the store. For ease of exposition, we assume that the customers visit the store every time period (e.g., week, day) but may leave the store without making a purchase. We use a binary vector $\boldsymbol{b}_{it} = [b_{it0}, \dots, b_{itJ}] \in \{0,1\}^{J\times 1}$ to denote purchase decisions by customer i at time t. The binary indicator $b_{itj} \in \{0,1\}$ represents whether customer i purchased product j at time t. We summarize information about the past purchase behavior of customer i using a purchasing history of length T and product purchasing frequencies over the entire available time horizon. We denote the purchasing history of length T for customer i at time t by $B_{it}^{T} = [\boldsymbol{b}_{i,t}, \boldsymbol{b}_{i,t-1}, \dots, \boldsymbol{b}_{i,t-T+1}] \in \{0,1\}^{J\times T}$ and the vector of product-specific purchasing frequencies for customer i over the entire customer purchasing history available at time t by $B_{it}^{\infty} = [\bar{b}_{it1}, \dots, \bar{b}_{itJ}] \in [0,1]^{J\times 1}$.

Customers receive personalized, product-specific coupons before each shopping trip (e.g., via email, mobile app, or in-store kiosk). Each coupon provides a percentage discount on a product at checkout. We denote personalized coupons by $D_{it} = \left[d_{it1}, \dots, d_{itJ}\right] \in [0,1]^{J\times 1}$, where $d_{itj} \in [0,1]$ indicates a size of the coupon (i.e., discount) received by customer i in time t for product j.

The proposed product choice model predicts the probabilities $P_{i,t+1} = [p_{i,t+1,1}, \dots, p_{i,t+1,J}]$ that customer i purchases product j at time $t+1$ for every product $j \in \{1, \dots, J\}$, given the coupon assignment $D_{i,t+1}$, the purchasing history B_{it}^{T}, the purchasing frequencies B_{it}^{∞}, and the model parameters θ:

$$P_{i,t+1} = f\left(D_{i,t+1}, B_{it}^{T}, B_{it}^{\infty}; \theta\right).$$

The vector $P_{i,t+1}$ contains the probabilities for the (binary) purchase events for all products j:

$$p_{i,t+1,j} = \mathbb{P}\left(b_{i,t+1,j} = 1\right).$$

Including both B_{it}^{∞} and B_{it}^{T} as input to the model serves two purposes. First, the model uses B_{it}^{∞} to learn the customer's base preferences, whereas it models purchase patterns over time based on B_{it}^{T}. Separating the information already at the model input simplifies the learning process and speeds up the training. Second, providing B_{it}^{∞} in addition to B_{it}^{T} reduces the dimensionality of the input data. Our model could learn B_{it}^{∞} directly from B_{it}^{T} if the window length were set to infinity (i.e., $T = \infty$). However, only recent purchases are relevant to model purchase timing, so we reduce dimensionality by considering a smaller window T and including B_{it}^{∞} as a summary of older purchases.

Figure 1 Proposed Neural Network Architecture for the Product Choice Model.

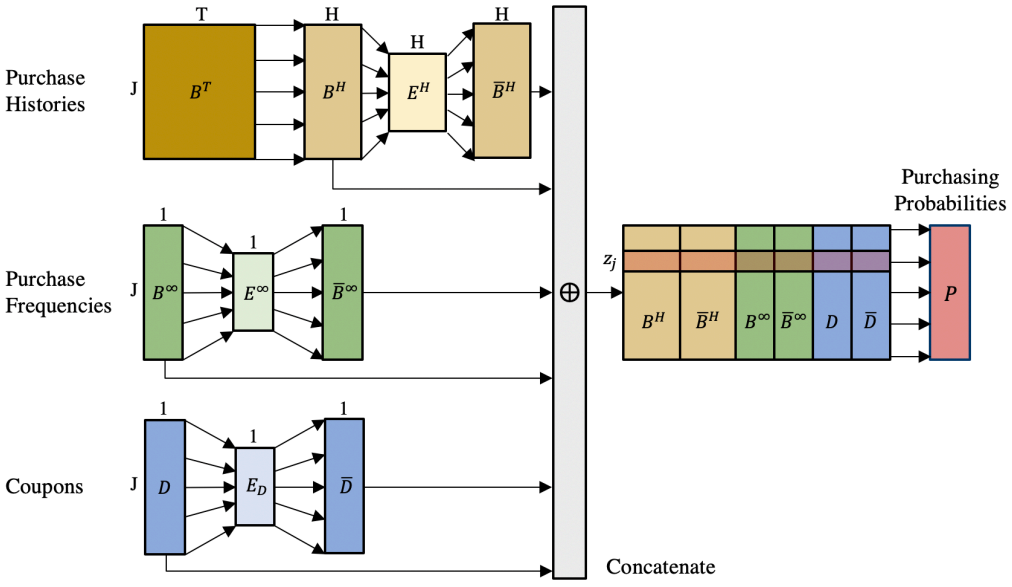


Figure 1 summarizes the proposed model architecture. The model is nonparametric and based on a neural network. Each observation in our model is a customer–time pair (i, t). For every training sample, the model transforms the inputs (i.e., $D_{i,t+1}, B_{it}^{T}, B_{it}^{\infty}$) to create product-specific feature maps $\boldsymbol{z}_{i,t+1,j} \in R^{Kx1}$, which are then used to predict the purchasing probabilities $p_{i,t+1,j}$ for every product in the assortment:

$$p_{i,t+1,j} = p(\boldsymbol{z}_{i,t+1,j}; \theta_P), \text{ with}$$

$$\boldsymbol{z}_{i,t+1} = [\boldsymbol{z}_{i,t+1,1}, \ldots, \boldsymbol{z}_{i,t+1,J}] \in R^{J \times K} \text{ and}$$

$$\boldsymbol{z}_{i,t+1} = Z(D_{i,t+1}, B_{it}^{T}, B_{it}^{\infty}; \theta_z).$$

The feature maps $\boldsymbol{z}_{i,t+1,j}$ summarize information about coupons and information about the customer purchasing behavior into customer-product-specific K-dimensional vectors. The model architecture infers cross-product relationships directly from the transaction data.

The proposed neural network provides a flexible functional form to approximate customer purchase behavior. The standalone parameters of the model have no behavioral or economic interpretation, but the model effectively predicts purchase events, which is a foundation for targeting applications. For example, the model can adjust customer-specific predicted probabilities to account for consumption dynamics and cross-product discount effects, as we demonstrate in Section 5.

3.2. Model Architecture

The inputs to the model are a customer purchasing history B_{it}^{T}, product purchasing frequencies B_{it}^{∞}, and a coupon assignment $D_{i,t+1}$. The model first transforms the purchasing histories B_{it}^{T}, which are sparse in a retail setting. We apply H different real-valued time-series filters $\boldsymbol{w}_h \in R^{T\times 1}$ and a leaky ReLU activation function:

$$B_{it}^{H} = [\sigma(B_{it}^{T} \cdot \boldsymbol{w}_1), \ldots, \sigma(B_{it}^{T} \cdot \boldsymbol{w}_H)] \in R^{J\times H},$$

where $\sigma(\cdot)$ is a leaky ReLU activation function (Xu et al. 2015):

$$\sigma(x) = \begin{cases} x & \text{for } x \geq 0 \\ 0.2x & \text{for } x < 0 \end{cases}.$$

The filters apply the same transformations to the purchasing histories of every product and create H product-specific summary statistics that represent information about recent purchases in a dense form. We calibrate the weights of the filters using the training data.

Our non-parametric approach for summarizing timing information is more flexible than manually defined transformations of the purchasing histories (e.g., weighted averages). This flexibility is important. Retail products vary substantially in their interpurchase times. For example, customers typically purchase milk every few days but detergent only once every few weeks. Observing a purchase of milk or detergent in period t thus requires different adjustments to the probability predictions in period $t+1$, and manually defining transformations suited for all products in the assortment is challenging. The time filters automatically calibrate these transformations, according to purchase patterns in the training data.

Purchasing frequencies B_{it}^{∞}, the purchasing histories B_{it}^{H} and coupon assignments $D_{i,t+1}$ are product-specific. We use linear bottleneck layers at the neural network to share information across products. In particular, we apply the following transformations:

$$E_{i,t}^{\infty} = W_{\infty}^{In} \cdot B_{it}^{\infty},\ \bar{B}_{it}^{\infty} = W_{\infty}^{Out} \cdot E_{i,t}^{\infty}$$

$$E_{i,t}^{H} = W_{H}^{In} \cdot \ B_{it}^{H},\ \bar{B}_{it}^{H} = W_{H}^{Out} \cdot E_{i,t}^{H}$$

$$E_{i,t+1}^{D} = W_{D}^{In} \cdot D_{i,t+1},\ \bar{D}_{i,t+1} = W_{D}^{Out} \cdot E_{i,t+1}^{D}$$

where W_{∞}^{In}, W_{H}^{In}, and W_{D}^{In} are $(L \times J)$ weight matrices, and W_{∞}^{Out}, W_{H}^{Out}, and W_{D}^{Out} are $(J \times L)$ weight matrices with $L \ll J$. We denote $W_{\infty} = (W_{\infty}^{In}, W_{\infty}^{Out})$, $W_{H} = (W_{H}^{In}, W_{H}^{Out})$, and $W_{D} = (W_{D}^{In}, W_{D}^{Out})$. The bottleneck layer encodes the inputs into low-dimensional representations $E_{i,t}^{\infty}$, $E_{i,t}^{H}$, and $E_{i,t+1}^{D}$. For example, in Section 4, we simulate a retailer with $J = 250$ products, and we estimate the model with $L = 30$. The model infers the weight matrices W_{∞}, W_{H}, and W_{D} during training.

The bottleneck layers are the basis for modeling cross-product relationships. Consider an illustrative example: Customer i is indifferent between Coke and Pepsi and purchases one of the two products when the combined stock of soft drinks at home is low. When the customer purchases Coke or Pepsi at time t, the retailer needs to adjust estimates of the probabilities that the customer will purchase these soft drinks at time $t + 1$. The adjustment in probabilities is independent of which brand was purchased at time t. The model recognizes this by creating similar L −dimensional representations of the purchase histories B_{it}^{H} and the purchasing frequencies B_{it}^{∞} for the two different scenarios (Coke or Pepsi). These L −dimensional representations are then expanded back to J dimensions to keep further operations at the by-product level.

Applying the bottleneck layer to the discounts $D_{i,t+1}$ captures a different type of relationship between products. A coupon for a soft drink (Coke or Pepsi) increases an overall consideration of the soft drink category and decreases purchasing probabilities for other products in the category; that is, the coupon creates own-category incidence and substitution effects. The coupon can also affect purchases in related product categories, such as iced tea or salty snacks. The bottleneck layer applied to the discounts $D_{i,t+1}$ encodes the discounts for similar products into similar vectors,

$$E_{i,t+1}^{D}(\text{Discount for Coke}) \approx E_{i,t+1}^{D}(\text{Discount for Pepsi}),$$

and then expands the representations to J dimensions $\bar{D}_{i,t+1} = W_{D}^{Out} \cdot E_{i,t+1}^{D}$. The first step ensures that coupons for similar products affect the purchase probabilities of other products similarly; the second step calculates a product-specific cumulative discount effect.

We combine the inputs and outputs of the bottleneck layers to create feature maps $\boldsymbol{z}_{i,t+1}$:

$$\boldsymbol{z}_{i,t+1} = \left[\mathbf{1}^{J\times 1}, D_{i,t+1}, \bar{D}_{i,t+1}, B_{it}^{\infty}, \bar{B}_{it}^{\infty}, B_{it}^{H}, \bar{B}_{it}^{H}\right] \in R^{J\times K},$$

where $K = 2H + 5$. Combining the inputs and outputs of the layer is a standard method to improve the predictive performance of the neural networks (Orhan and Pitkow 2017). We input the feature maps

$\boldsymbol{z}_{i,t+1,j}$ to a sigmoid layer to predict purchasing probabilities $P_{i,t+1} = [p_{i,t+1,1}, \ldots, p_{i,t+1,J}]$ for every product in the assortment:

$$p_{i,t+1,j} = \frac{\exp\left(\theta_p \, \boldsymbol{z}_{i,t+1,j}\right)}{1 + \exp\left(\theta_p \, \boldsymbol{z}_{i,t+1,j}\right)}.$$

Feature maps $\boldsymbol{z}_{i,t+1,j}$ summarize relevant information about customer purchasing behavior and the coupon assignment from the inputs, and the sigmoid layer uses $\boldsymbol{z}_{i,t+1,j}$ as the input to predict the purchasing probability for customer i and product j at time t. The parameters θ_p are shared between products.

The functional form of the sigmoid layer is similar to a binary logit model, but they are conceptually different. Traditional binary logit models assume category-specific weights and variation in the product attributes. The product attributes are defined by the researchers. Our model encodes product differences and cross-product effects in the feature maps $\boldsymbol{z}$ and keeps the weights shared among all products across categories. The model infers the feature maps $\boldsymbol{z}$ from the transaction and coupon assignment data.

3.3. Model Calibration

The parameters of the model are the time filters $\boldsymbol{w}_h$, the bottleneck layer parameters W_∞, W_H, and W_D, and the parameters of the sigmoid layer θ_P:

$$\theta = (\theta_z; \theta_p), \theta_z = (\boldsymbol{w}_{h=1..H}; W_\infty; W_H; W_D).$$

We calibrate the parameters by minimizing the binary cross-entropy loss

$$\theta^* = argmin_\theta \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{t=1}^{T} L(b_{i,t+1,j}, \hat{p}_{i,t+1,j}),$$

with

$$L\left(b_{i,t+1,j}, \hat{p}_{i,t+1,j}\right) = -\left(b_{i,t+1,j} \log \hat{p}_{i,t+1,j} + \left(1 - b_{i,t+1,j}\right) \log\left(1 - \hat{p}_{i,t+1,j}\right)\right). \quad (1)$$

We use the adaptive moment estimation (Adam; Kingma and Ba 2014) algorithm with mini-batches to optimize the parameters. Training the model in mini-batches allows distributed computing and does not require having all training data in memory. With the proposed neural network architecture, we can efficiently compute gradients through backpropagation. Model calibration is therefore feasible even with many customers I and a large assortment J. We provide a complete specification of the optimization algorithm in Appendix A.

The model also can be trained in stages. Retailers often have rich market basket data with no customer identifiers. Our model can leverage these data to better identify cross-product relationships. In particular, the unlabeled market basket data can be used to train product embeddings (Gabel et al. 2019), and then the model can initialize the bottleneck layer parameters with the embeddings. Initialization with pretrained parameters improves the predictive performance of the neural network models and leads to faster convergence (Bengio et al. 2007).

3.4. Applications in Retail Analytics

The proposed product choice model can be applied to retail analytics problems beyond coupon personalization, with simple adjustments to the modular structure of the neural network. For example, many targeting applications require a prediction of total revenue or basket size in response to marketing actions. Our model estimates purchase probabilities for every product in the assortment, which can be combined to derive basket-level outcomes by an additional output summation layer in the model architecture. Category management applications also can aggregate the output probabilities for products in a specific category.

Another important characteristic of the model is the dimensionality of the marketing actions $D_{i,t+1}$. In the coupon optimization problem, $D_{i,t+1}$ is a J-dimensional vector that indicates the depth of the discounts for all products in the assortment. Product recommendation and assortment optimization applications can include $D_{i,t+1}$ as J-dimensional binary representations of combinations of products. If a retailer decides which customers should receive a promotion, $D_{i,t+1}$ can be represented as a binary variable corresponding to *mail* and *no mail* decisions (Simester et al. 2019a). The model can then be calibrated without applying a bottleneck layer W_d to the actions $D_{i,t+1}$, such that the feature map $\boldsymbol{z}_{i,t+1,j}$ would not include $\overline{D}_{i,t+1}$.

Furthermore, the model can be extended to incorporate additional information relevant for targeting. For example, retailers might leverage information about the timing of the shopping trip, the location of the store, or customer demographics, as well as unstructured data such as product reviews (Archak et al. 2011) or images (Zhang and Luo 2018). These data can be preprocessed by additional (or pretrained) neural network layers and added to the feature maps $\boldsymbol{z}_{i,t+1,j}$ by concatenation:

$$\boldsymbol{z}^{*}_{i,t+1,j} = [\boldsymbol{z}_{i,t+1,j}, I_{itj}].$$

This extension increases the number of parameters θ_P but the optimization of the model stays tractable.

Finally, the proposed neural network can efficiently summarize information in the loyalty card data as input for downstream applications. For example, output predictions and product relationships encoded in the hidden layers can be incorporated in customer segmentation, churn management, or purchase

incidence modeling solutions (Lemmens and Gupta 2020; Rossi et al. 1996). Similar to pretrained models in natural language processing and computer vision problems, researchers can train the full neural network, and then replace the last layers or fine-tune the weights of the calibrated model to improve its performance in downstream applications (Mikolov et al. 2017; Tajbakhsh et al. 2016).

4. Simulation Setup

The proposed deep neural network aims to approximate customer purchasing behavior and predict future purchases. We use simulated data to evaluate model performance in a controlled environment. We draw on prior research in marketing to design the data generating process (Manchanda et al. 1999; Fader and Hardie 1996; McFadden 1974). An important benefit of using a simulation is that the true purchase probabilities and the parameter of the data generating process are known. We can thus better evaluate the model's performance and decompose performance gains.

We simulate a retailer with I customers and an assortment of J products. The products are grouped into C product categories of equal size. Customers visit the store every period and make purchase decisions in two stages: (1) whether to buy a product in a category and (2) which product to buy in each selected category (Neslin and van Heerde 2009). The purchase probability of customer i and product j (in category c) at time t is given by

$$p_{itj} = p_{itc}^{(1)} \cdot p_{itj}^{(2)} ,$$

where $p_{itc}^{(1)}$ is the category purchase incidence probability (Section 4.1), and $p_{itj}^{(2)}$ is the product choice probability, conditional on the category incidence (Section 4.2). Note that we use the simulation to create synthetic loyalty card data, and our proposed neural network product choice model aims to approximate customer behavior without identifying the underlying parameters of the data generating process.

4.1. Stage 1: Category Purchase Incidence

We model the category incidence as a multivariate probit model (Manchanda et al. 1999). Customer i's utility of category c depends on a customer-specific base preference, the average coupon discount in the category c, average coupon discounts in other categories $k \neq c$, and the current inventory:

$$u_{itc} = \gamma_c + \gamma_{ic} + \gamma_{ic}^{p} \bar{d}_{itc} + \sum_{\mathrm{k}\neq\mathrm{c}} \gamma_{ick}^{p} \bar{d}_{itk} + \gamma_{ic}^{Inv} Inv_{ic}^{t} + \varepsilon_{itc}.$$

Here, $\gamma_c + \gamma_{ic}$ is the (customer-specific) base utility, $\bar{d}_{itc}$ is the average coupon discount in category c (Nijs et al. 2001), and Inv_{ic}^{t} is customer i's inventory (i.e. stock) in category c at time t. Assuming that the random noise has a standard normal distribution, $\varepsilon_{itc} \sim N(0,1)$, the purchase incidence probability becomes

$$p_{itc}^{(1)} = \mathbb{P}(y_{itc} = 1) = \Phi(u_{itc}),$$

where Φ is the cumulative density function of the standard normal distribution, and y_{itc} indicates the category purchase incidence, that is the purchase of any product j in c:

$$y_{itc} = \mathbb{I}_{\{\sum_{j \in c} b_{itj} > 0\}}.$$

Customers are characterized by latent taste preferences Θ_i, and we model $\gamma_{ic} = \Gamma_c \Theta_i$. Parameters Γ_c define purchase coincidence between product categories (Manchanda et al. 1999); that is, customers tend to purchase categories c and c' together if Γ_c and $\Gamma_{c'}$ are similar.

Products within the categories have different purchasing frequencies (see Section 4.2). In some product categories, a few products account for most sales. We thus weight the coupon discounts by the customer's purchase share of each product, that is

$$\bar{d}_{itc} = \frac{\sum_{j \in C} p_{itj}^{(2)} \, d_{itj}}{|C|}.$$

Inventory dynamics are determined by the customer-specific consumption rates, $Cons_{ic}$. The inventory is aggregated to the category level, and consumption rates differ between categories:

$$Inv_{ic}^{t} = Inv_{ic}^{t-1} + \sum_{j \in C} b_{itj} - Cons_{ic}.$$

4.2. Stage 2: Product Choice

Product choice within a category follows a multinomial logit model (Guadagni and Little 1983; McFadden 1974). We assume the following form of customer i's utility for product j at time t:

$$u_{itj} = \beta_{ij}^{0} - \beta_i^{p}(1 - d_{itj}) \cdot price_j + \varepsilon_{itj},$$

where β_{ij}^{0} indicates customer i's base utility for product j, β_i^{p} is customer-specific price sensitivity, $price_j$ is a (regular) price of the product j, and d_{itj} is the size of the coupon provided to customer i for product j at time t. Assuming that the error term ε_{itj} follows a Gumbel extreme value distribution, the probability that customer i chooses product j in category c becomes

$$p_{itj}^{(2)} = \mathbb{P}\big(b_{itj} = 1 \big| y_{itc} = 1\big) = \frac{\exp\{u_{itj}\}}{\sum_{k \in C} \exp\{u_{itk}\}}.$$

The base utility β_{ij}^{0} is customer and product specific. We define $\beta_{ij}^{0} = B_j \Theta_i$, where Θ_i is the customer taste characteristic vector from Stage 1. Customer i's price sensitivity β_i^{p} is constant across categories.

We also assume that product prices, $price_j$, are constant over time, and coupons are the only source of price variation.

4.3. Simulation Calibration

We simulate a retailer with $J = 250$ products grouped into $C = 25$ categories and $I = 100{,}000$ customers. We also tested the proposed model with more products ($J > 1{,}000$) and categories ($C > 100$). The substantive findings reported in Section 5 are robust.

For every customer, we draw taste characteristics from the multivariate normal distribution $\Theta_i \sim MVN(0^{h\times 1}, h^{-1}\, I^{h\times h})$, where h is the dimensionality of the latent tastes. We simulate 130 burn-in periods to allow the inventory to converge, and we simulate an additional 60 periods for model training and 10 periods for model evaluation.

Customers receive coupons every time period. For model training and evaluation, we assume that the coupons are assigned randomly, with discounts that range from 10% to 40%. We benchmark the predictive performance of the proposed product choice model in a simulation with five coupons per customer. Random coupon assignment in the simulation is consistent with the empirical application.

We define the parameters of the category purchase incidence model ($\gamma_c, \Gamma_c, \gamma_{ic}^{p}, \gamma_{ick}^{p}, \gamma_{ic}^{Inv}, Cons_{ic}$) and the product choice model (B_j, β_i^{p}) to balance customer heterogeneity, consumption dynamics, and coupon and inventory effects on product purchasing rates. We calibrate the data generating process, such that the characteristics of the synthetic data are similar to the empirical transaction data in Section 6. The sampling distributions and parameter values are in Appendix B.

5. Evaluation of the Proposed Model Using Synthetic Data

To evaluate the performance of the proposed product choice model on a holdout test data, we simulate 70 time periods, train our model and the baseline models using the first 60 periods, and then evaluate predicted purchase probabilities for the following 10 periods. The models never access data from the last 10 time periods during training. For more details on the holdout test set construction, see Appendix C.

We repeat model training and evaluation with 30 simulated data sets. The datasets are based on different parameter draws for the data generating process. We report the average performance and calculate standard errors across simulated data sets throughout this section.

5.1. Baseline Models

We compare the performance of our model against three baselines. The first baseline is a binary logit model (hereafter Binary Logit). We train and apply the Binary Logit model by-product. For each product,

the independent variables are the current discount $d_{i,t+1,j}$, customer-specific purchasing frequency $\bar{b}_{itj}$, number of recent product purchases with multiple time-windows h (Fader et al. 2005), and a cosine similarity between the customer embedding and product representation based on the Product2Vec model (Gabel et al. 2019). We use these independent variables to predict the purchase decision $b_{i,t+1,j}$.

We use LightGBM as a second baseline (Ke et al. 2017). LightGBM is an efficient implementation of the gradient boosting decision tree algorithm. We calibrate a separate LightGBM model for every product in the assortment with an extended set of independent variables: the product-specific variables in the Binary Logit model, product purchasing history $[b_{itj}, \dots, b_{i,t-T+1,j}]$, current discounts for all other products in the assortment, and binary indicators for individual products. The LightGBM model extends Binary Logit by incorporating cross-product discounts and the entire product-specific purchasing history. Using the purchasing histories for all products is not feasible in the LightGBM model due to high dimensionality and data sparseness. For a complete description of the Binary Logit and LightGBM independent variables, see Appendix D.

Our third baseline is a hierarchical multinomial logit model (Hierarchical MNL; Rossi et al. 2012). The Hierarchical MNL model relies on the predefined category structure and estimates purchase probabilities by category. We provide true category definitions from the simulation to the model. For every category C, the model specification includes product random effects, heterogeneous price effects $(1 - d_{itj}) \cdot price_j$ (for $j \in C$), and heterogeneous effects of the number of recent category purchases with multiple time-windows h:

$$B_{itC}^{h} = \frac{1}{h} \sum_{k=1}^{h} \sum_{j \in C} b_{i,t-k+1,j}$$

For computational reasons, we subsample the training data and estimate the Hierarchical MNL with 8,000 customers. The model achieves no substantial improvement with more training examples.

The Hierarchical MNL depends on information about category structure that is not incorporated into the other baselines or the proposed neural network model. In our simulation, the category incidence probability depends on the customer's inventory (Inv_{ic}^{t}) for the category, and the discounts always affect other products within the same category. The inputs to the Hierarchical MNL thus are well-suited to capture dynamic purchase behavior in the simulation. We show in Section 5 that our model yields similar or better performance than the Hierarchical MNL across the benchmarks without category information from the data generating process. In practical applications, category information in retail varies substantially across the retailers, and predefined category structures often cannot perfectly isolate within-category consumption or substitution patterns (Smith et al. 2019).

The proposed neural network model extends the baselines, by using full information about all products to *simultaneously* predict purchasing incidence for all products in the assortment. Leveraging rich high-dimensional information for all products is possible with the proposed parsimonious model architecture, including the time filters to reduce purchase history sparsity, bottleneck layers to encode cross-product relationships and weight sharing to reduce the number of parameters and regularize the model.

5.2. Aggregate Predictive Performance

Table 1 evaluates the predictive performance of the proposed neural network using simulated data with five random coupon discounts per customer. We report the average binary cross-entropy loss and KL divergence calculated using the holdout data in 30 simulation data sets. The binary cross-entropy measure is equivalent to negative log-likelihood and indicates how well the predicted probabilities approximate binary purchasing decisions. The KL divergence compares the predicted probabilities to the true simulated probabilities. Our model achieves better average predictive performance than the reference models on both metrics.

Table 1 Aggregate Predictive Performance (Simulation)

	Cross-Entropy Loss	**KL Divergence**
True Probabilities	0.0778	0.000
Our Model	0.0810	0.0035
Binary Logit	0.0914	0.0329
LightGBM	0.0813	0.0052
Hierarchical MNL	0.0857	0.0125

Notes: The table reports the average aggregate predictive performance across 30 simulated data sets. All differences are significant at $p < 0.01$.

Results presented in Table 1 are conditional on the number of training examples. In Appendix E, we evaluate aggregate predictive performance as a function of the size of the training data. Our model outperforms the baseline models with at least 4,000 customers in the training data, which is equivalent to 120,000 training observations (4,000 customers times 30 training weeks). Many retailers have larger customer bases in their loyalty programs. For example, the empirical application in Section 6 leverages transaction data from a grocery retailer with more than 150,000 registered regular customers.

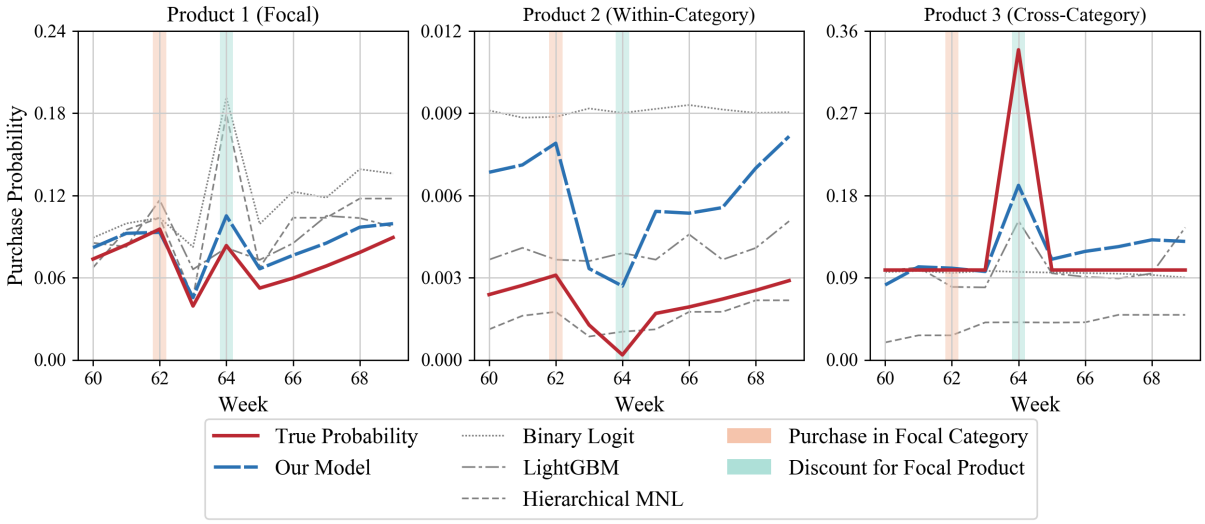
5.3. Predictive Performance Decomposition

For effective personalization applications, product choice models need to capture time dynamics in product choice (e.g., individual consumption patterns) and predict the effects of marketing actions. We therefore provide a more detailed evaluation next.

5.3.1. Product Choice Dynamics

The time dynamics of purchase probabilities in our simulation are determined by category inventory dynamics and coupon assignments. Figure 2 depicts purchase probabilities of three products for one customer over ten holdout periods. Products 1 and 2 belong to the same product category, and Product 3 is from a different product category.

Figure 2 Time-Series Prediction (Hold-Out Set)



There are two observations of interest. First, Figure 2 illustrates inventory dynamics. A customer purchases Product 1 at time $t = 62$. Once the purchase happens, the simulation increases the (latent) category inventory for the customer, and purchase probabilities decrease for all products in the category in the following few time periods. Inventory effects are category specific; they do not affect purchase behavior in other categories (Product 3). Our model captures such inventory dynamics for the focal and within-category products and correctly predicts no purchase effect across categories. Recall that category definitions are not provided as inputs to our model. To predict within-category and the lack of cross-category effects of the purchase events, the model automatically infers the category structure from transaction data.

The second observation is about the effects of a coupon for Product 1 at time $t = 64$. The simulation assumes that coupons affect the purchase probability of the focal product and other products within the

category, as well as of products in related categories. In Figure 2, we observe a substantial positive effect on Product 1 (focal) and a negative effect on Product 2 (within-category). Product 3 belongs to a complimentary product category, so we observe a positive cross-category coupon effect. The neural network model adjusts the probabilities for own- and cross-product effects. Similar to inventory dynamics, identifying cross-product coupon effects without category structure information is a challenging problem; for example, the LightGBM baseline in Figure 2 captures the positive effect on Product 3 but not the negative effect on Product 2.

The next two sections unfold this illustrative example by a more-structured analysis of time dynamics (Section 5.3.2) and discount effects (Section 5.3.3).

5.3.2. Time Dynamics and Inventory Effects

To quantify how well the models capture the time dynamics of purchase probabilities, we estimate the correlation of the predicted probabilities and the true probabilities over time (i.e., for 10 holdout weeks) for every customer–product pair:

$$\rho^{time} = \frac{1}{IJ}\sum_{ij} corr_t^{ij},$$

with

$$corr_t^{ij} = \frac{cov_t\left(\hat{p}_{itj}, p_{itj}^{true}\right)}{\sigma_{\hat{p}}\,\sigma_p},$$

where $\sigma_{\hat{p}}$ and σ_p are the standard deviations of the predicted and true probabilities over time. To ensure numerical stability in the computation, we add random noise $\eta \sim U(0,10^{-6})$ to both predicted and true probabilities. We compute correlation metrics for simulated data with coupons and without coupons. The first data set includes both sources of probability variation over time: the effect of consumers' inventories and the coupon effects. The second data set isolates the inventory effect.

Table 2 reports the average time correlation score ρ^{time} for our proposed model and three baselines. The neural network architecture achieves an average time correlation of $\rho^{time} = 0.48$ with coupons and $\rho^{time} = 0.35$ without them. These values are significantly higher than the time correlation scores for Binary Logit and LightGBM models and similar to the Hierarchical MNL results.

Recall that the Hierarchical MNL incorporates category structure information, and our simulation assumes consumption dynamics at the category level. Our model yields similar time correlation performance without the predefined categories as input.

Table 2 Time Series Correlation Scores for Model Predictions

	Data with Coupons	**Data without Coupons**
True Probabilities	1.00	1.00
Our Model	0.51	0.35
Binary Logit	0.15	-0.01
LightGBM	0.17	0.00
Hierarchical MNL	0.51	0.37

Notes: The table reports average time series correlations across 30 simulated data sets. The differences between our model and the Hierarchical MNL are not statistically significant; all other differences are significant at $p < 0.01$.

5.3.3. Own- and Cross-Product Coupon Effects

The simulation setup implies that a coupon for a product affects the purchase probability of that product (own-product effect), other products in the same category (within-category effects), and products in other product categories (cross-category effects). Cross-category effects can be positive, negative, or null. We evaluate whether the model can recover coupon effects in the holdout data by comparing the true coupon discount elasticities in the simulation with the predicted elasticities.

To calculate the true discount elasticities, we save the simulation after 10 holdout periods and calculate purchasing probabilities for each customer–product combination (i, j) in the next period for two scenarios:

1) The retailer does not provide coupons to customers.
2) All customers receive a 30% discount for product j_c.

We repeat this process for all products $j_c \in \{1, \dots, J\}$, average the probabilities across the customers and calculate product-specific discount elasticities

$$\varepsilon_{j,j_c} = \frac{p_j^{30\%} - p_j^{0\%}}{0.3 \cdot p_j^{0\%}},$$

where $p_j^{30\%}$ is the average purchasing probability for product j given a 30% discount on product j_c, and $p_j^{0\%}$ is the average purchasing probability for product j without coupons. This evaluation process yields a $J \times J$ matrix of own- and cross-product elasticities. To compute elasticities for different models, we replace true purchasing probabilities with the predicted values.

Table 3 reports the true and predicted discount elasticities. All models capture a positive effect of price discounts on own-product purchase probabilities. Estimating the cross-price effects is a more challenging benchmark. The Binary Logit model does not incorporate cross-price effects, so all cross-product elasticity estimates within and across categories are zero. The Hierarchical MNL model estimates significant within-category cross-price elasticities and assumes null cross-category effects. The LightGBM model infers non-zero cross-category elasticities, but substantially underestimates the magnitude of the discount effects. The proposed neural network model closely approximates the cross-product discount effects within and across categories.

Table 3 True and Estimated Discount Elasticities (Average Across Products)

	Own-Product	**Within-Category**	**Cross-Category**		
			Positive	**Negative**	**Zero**
True Probabilities	4.433	-0.193	0.133	-0.120	0.000
Our Model	4.467 [0.167]	-0.161 [0.070]	0.134 [0.064]	-0.102 [0.059]	0.009 [0.038]
Binary Logit	4.402 [0.386]	0.000 [0.193]	0.000 [0.133]	0.000 [0.120]	0.000 [0.000]
LightGBM	4.165 [0.331]	-0.059 [0.137]	0.028 [0.107]	-0.022 [0.100]	0.002 [0.005]
Hierarchical MNL	4.180 [0.464]	-0.117 [0.077]	0.000 [0.133]	0.000 [0.120]	0.000 [0.000]

Notes: The table reports true and estimated own-product and cross-product discount elasticities, with the mean absolute errors for the estimated elasticities in the brackets, averaged across 30 simulated data sets. Our model achieves the lowest mean absolute errors; and all differences between the models are significant at $p < 0.01$.

In Appendix F, we compare our model to a LightGBM baseline with additional inputs: average coupon price discount in all product categories, and frequency of recent purchases in the own-product category with different time windows. These features are defined using the category structure from the true data generating process in the simulation. The extended LightGBM model can better capture cross-product discount effects and yields higher time correlation scores than the initial LightGBM specification in Section 5.1, and the proposed neural network model outperforms the extended baseline in all benchmarks.

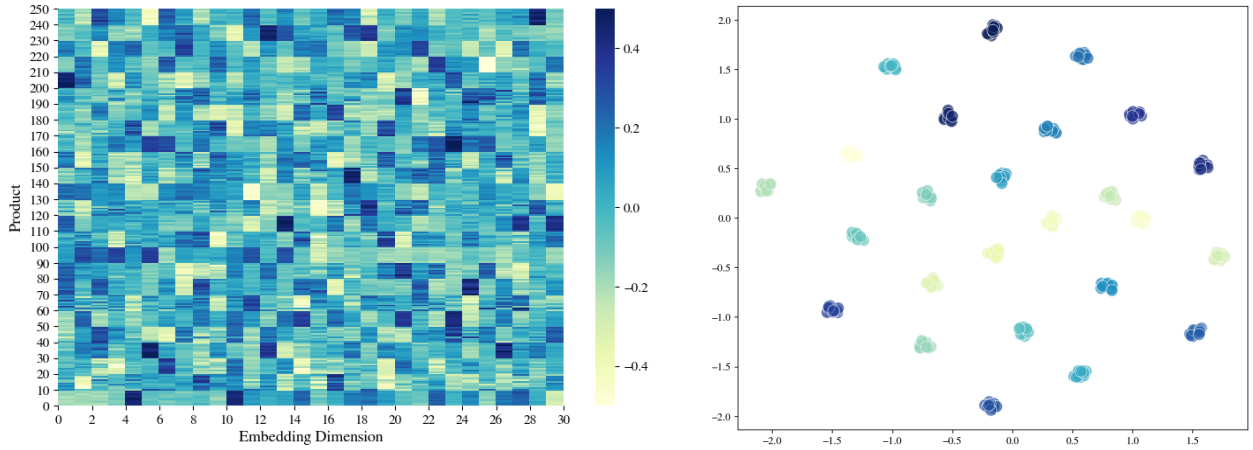
5.3.4. Identifying Product Category Structure

Our analysis of inventory time dynamics and cross-product coupon effects indicates that our model identifies cross-product relationships within and across categories. However, the model does not require specifying the product categories *ex ante* and infers cross-product relationships from the training data.

The cross-product relationships are encoded in the parameters of the bottleneck layers. In Figure 3a, we plot the heatmap of the bottleneck layer weight matrix W_H^{In}. The weight matrix W_H^{In} has 250 rows, corresponding to $J = 250$ products in the simulated data. We order products by product categories, such that the first 10 products correspond to the first product category, the next 10 products correspond to the second category, and so forth. The heatmap reveals 25 groups of 10 similar rows in the matrix W_H^{In}, which correspond to $C = 25$ product categories. The rows of matrix W_H^{In} can be considered as *product embeddings*, as they incorporate information about product similarities. Products from the same categories have similar product embeddings.

Figure 3b depicts the two-dimensional t-SNE projections (Maaten 2014; Maaten and Hinton 2008) of product embeddings based on W_H^{In}. Each dot represents one product, and the true (simulated) categories are indicated by different colors. The products form clusters, corresponding to different categories, and the clusters are perfectly separated, which confirms that the trained product embeddings encode information about the product category structure. We document similar results for the discount bottleneck weight matrix W_D^{In} in Appendix G.

Figure 3 Heatmap and t-SNE Projection of Product Embeddings, based on $\boldsymbol{W_H^{In}}$.



(a) (b)

5.3.5. Nested Model Analysis

To demonstrate how different components of the proposed neural network architecture affect the predictive performance of the model, we sequentially remove modules from the full architecture and

evaluate the nested specifications. The nested model analysis explicates the value of the discount bottleneck layer W_D, the purchase history bottleneck layer W_H, and the time-series filters $\boldsymbol{w}_{h=1..H}$. We discuss the evaluation results below and provide details in Appendix H.

The nested model analysis confirms that the discount bottleneck layer W_D is critical to estimating within- and cross-category discount effects. In our simulation, we order products by product categories (see Section 5.3.4). Replacing the discount bottleneck layer with a block-diagonal matrix with C main-diagonal blocks, such that $\overline{D}_{i,t+1} = diag(W_D^1, \dots, W_D^C) \cdot D_{i,t+1}$, eliminates the ability of the model to infer cross-category discount elasticities. Furthermore, the model estimates null within- and cross-category discount elasticities when we completely remove the discount bottleneck layer.

When we remove the discounts $D_{i,t+1}$ from inputs to the neural network model, the model cannot estimate any discount effects but yields time correlation scores on the data without discounts similar to the full model specification. Retailers often have historical transaction data without experimental variation in coupon allocations. Our model can leverage these data to pretrain parameters of the neural network related to the purchase histories, such as W_H, W_∞, $\boldsymbol{w}_{h=1..H}$ and θ_z; costly experimental data can then be used to calibrate W_D and fine-tune the model.

The purchase history bottleneck layer W_H helps to adjust predicted probabilities for the effects of recent purchases. Recall that our simulation assumes that individual purchase probabilities depend on recent purchases aggregated to the category level. The model cannot account for purchases of other products within the focal category without the bottleneck layer W_H. In contrast, incorporating the block-diagonal constraint on the purchase history transformation $\overline{B}_{it} = diag(W_H^1, \dots, W_H^C) \cdot B_{it}$ helps the model recognize zero cross-category effects of recent purchases in the simulation, and thus improves time correlation scores and elasticity estimates.

The nested model analysis also demonstrates the value of the time-series filters $\boldsymbol{w}_{h=1..H}$. The model calibrates the filters using training data to represent sparse purchase histories B_{it}^T in a dense form. Flexible calibration allows to capture purchase patterns in assortments with high variation in product interpurchase times. In Appendix H, we observe that replacing the time-series filters with manually-defined transformations of purchase histories leads to a loss of information and results into substantially lower time correlation scores. Time correlation score is an important indicator of the model's ability to optimize marketing actions in time.

5.3.6. Scalability of the Proposed Product Choice Model

The proposed product choice model estimates in approximately one hour for a data set with $I = 100{,}000$ customers, $T = 30$ training periods and $J = 250$ products on a server with an Intel(R) Core(TM) i7 CPU, 256GB RAM, and a TITAN X (Pascal) graphics card.

The estimation scales to large customer bases and product assortment. The model has no customer-specific parameters, so larger customer bases provide more training data without affecting the model specification. Inference for large customer bases can be implemented on distributed systems. The cross-product relationships in our model are captured by dense product representations in the parameters of the neural network. For a given dimensionality of product representations, the number of parameters of the model scales linearly with product assortment. We demonstrate in Appendix I that estimation time also linearly depends on the number of products, and the model effectively infers the product category structure in a simulation with many products ($J = 900$; our empirical data includes 418 products).

5.4. Performance Gains for Coupon Optimization

We conclude the evaluation of the proposed product choice model by demonstrating how the improved predictive performance translates into the efficiency gains for coupon personalization. For every customer, we use our model and the baseline models to select one coupon that yields the highest predicted revenue uplift. We then evaluate coupon allocations using the true data generating process in the simulation. Formally, we solve the following maximization problem for every customer i:

$$D_{it}^{*} = argmax_{D=[d_1,\dots,d_J]} \sum_j price_j \cdot \left[\left(1 - d_j\right) \cdot \hat{p}_{itj}(D) - \hat{p}_{itj}(\mathbf{0})\right]$$

$$\text{s.t. } D \in \{0, 0.3\}^{J\times 1} \text{ and } \textstyle\sum_j I\left(d_j > 0\right) = 1,$$

where $\hat{p}_{itj}(D)$ and $\hat{p}_{itj}(\mathbf{0})$ are the predicted purchase probability for customer i and product j at time t with the coupon assignment D and without coupons, respectively. The predictions are based on individual purchase histories B_{it}^{T} and purchase frequencies B_{it}^{∞}.

We also evaluate coupon personalization with a different objective function, as an additional test of the model's predictive performance. In particular, we maximize the focal product revenue uplift, i.e. we provide a discount for a product with the highest predicted own-product revenue gain. The focal product uplift optimization does not account for the cross-product discount effects and only leverages customer-specific purchase probability estimates for the focal product.

Table 4 presents coupon personalization results with total uplift and focal product uplift optimization objectives. We derive the optimal coupon allocations for each model in 30 simulated data sets and report the average revenue uplifts per customer.

Table 4 Coupon Optimization Results

	Revenue		Purchase Incidence	
	Total	**Focal Product**	**Total**	**Focal Product**
True Probabilities	\$2.22	\$1.20	0.52	0.35
Our Model	\$0.86	\$0.99	0.34	0.30
Binary Logit	\$0.26	\$0.83	0.21	0.28
LightGBM	\$0.47	\$0.95	0.26	0.29
Hierarchical MNL	\$0.70	\$0.82	0.27	0.27
Best Uniform	\$0.41	\$0.24	0.10	0.08
Random	\$0.01	\$0.10	0.02	0.04

Notes: The table reports the expected revenue and purchase incidence uplifts for coupons allocated using different prediction models across 30 simulated datasets. *Total* and *Focal Product* correspond to optimizing gains from the total market basket or the focal product only. All differences are significant at $p < 0.01$.

Random coupon allocation defines the lower bound for coupon personalization performance; coupon policies that optimize revenue should outperform this lower bound. When we compare the coupon allocation based on the product choice models with the random coupon assignment, all optimized methods outperform the lower bound for both total revenue and focal product revenue optimization.

A second reference point is a mass marketing coupon policy, which provides the same revenue-maximizing discount to all customers (Best Uniform). As expected, this policy leads to a larger revenue uplift than the random coupon allocation and is outperformed by all four models that personalize coupons.

The proposed neural network model significantly improves coupon personalization over the Binary Logit, LightGBM and Hierarchical MNL baselines. A policy based on our model yields a 1.2x-3.3x higher total revenue uplift than the baseline models. Our model also outperforms the baselines in a focal product optimization problem, but the relative gains are smaller.

These coupon personalization results are consistent with the analysis in Sections 5.3. Compared with the baseline models, the proposed product choice model yields similar estimates for the own-price elasticities and substantially better cross-price elasticities (see Table 3). We thus expect that our model is particularly valuable for retail applications that account for cross-product effects of marketing actions, and coupon personalization indeed results in greater outperformance margins when the outcomes are aggregated across the entire assortment.

To better understand the coupon optimization gains, we additionally study a simulated data set without cross-category discount effects, i.e. $\gamma_{ick} = 0 \; \forall c \neq k$. Detailed results are available in Appendix J. The

proposed neural network model yields higher revenue uplifts than all reference models for the focal product revenue optimization. For the total revenue optimization, our model achieves higher revenue uplifts than the Binary Logit and the LightGBM baselines, but not the Hierarchical MNL. The Hierarchical MNL correctly assumes zero cross-category effects and leverages perfect category information so this result is not surprising. If product categories are completely unrelated, product choice models can benefit from modeling categories independently or explicitly constraining cross-category effects to zero. Our product choice model can incorporate information about the category structure in the bottleneck layers (Section 5.3.5). In practical applications, and especially in grocery retailing, we expect that many categories are either complements or substitutes and that category definitions do not perfectly delineate.

As a robustness check, we replicate the coupon personalization analysis optimizing purchase incidence instead of revenue (Table 4). Total purchase incidence is equivalent to the expected market basket size, an important metric in retail analytics. Purchase incidence allows to evaluate coupon personalization gains without re-weighting the predicted probabilities by the product prices, which is a more direct evaluation of the product choice model predictive performance. The results are consistent with the findings from revenue optimization.

We conclude that the improved predictive performance of the proposed product choice model yields substantial performance gains in coupon personalization in our simulation. The model accounts for both own-product and cross-product discount effects, leading to better targeting decisions, and the relative gains are larger when the cross-product effects are more pronounced.

6. Evaluation of Predictive Performance Using Empirical Data

6.1. Data

We validate the predictive performance of our model using transaction data provided by a leading German grocery retailer. The data set comprises three data sources: loyalty card data, market basket data, and coupon data. Loyalty card data have a panel structure and contain transactions by loyalty card holders; market basket data include information about purchases without customer identifiers; and coupon data contain information about the coupons provided to the customers (including non-redeemed coupons). Overall, the data set spans 80 weeks (2015–2016) and includes 22,740,377 purchases with customer identifiers and 73,048,605 shopping baskets with no customer information.

The retailer provided coupons to promote category–brand combinations, thereby grouping stock-keeping units of the same package size and price range. We adopt the retailer's product grouping as the level of aggregation for our analysis, and we focus on 45 product categories for which the retailer distributed

coupons during the period of the analysis. The categories include food products (e.g., milk, bread, and chocolate bars), and non-food products (e.g., shampoo, fabric softener, and toothbrushes). We provide a complete list of the product categories and their interpurchase times in Appendix K.

For a small share of customers, the retailer provided coupons randomly. The coupons were distributed via kiosks at the store entrance for the current shopping trip. Our empirical analysis only considers customers with randomly assigned coupons, which allows us to avoid endogeneity concerns in model training and validations.

Table 5 Summary Statistics: Empirical Application

Data Set	Variable	Value
Loyalty Card	# of users	150,094
	# of weeks (date range)	80 (2015/06 - 2016/12)
	# of brands (# of retailer categories)	418 (45)
	# of stores	155
Coupon	# of coupons	524,972
	Avg. # of coupons per customer (SD)	3.50 (6.21)
	Discount range	[5%, 50%]
	Avg. redemption rate (SE)	7.83% (0.04%)
	Avg. discount (SD)	22.5% (9.6%)
Market Basket	# of baskets	73,048,605
	# of months (First year of loyalty card data)	12
	Avg. # of products per basket	4.91

Notes: The data includes only frequent shoppers, large brands, and in-store coupons.

We apply three additional data preprocessing steps. First, most customers visit the store no more than once a week, and the median time between two shopping trips is two weeks, so we aggregate the data to a weekly level. Second, the retailer introduced a loyalty card program at the beginning of the observation period. We drop the first 10 weeks of data and customers with less than 15 shopping trips, to account for non-stationarity due to the early adoption. Third, for every product category, we aggregate the least frequently purchased products into an *Other Product* option, so that the total number of products per category never exceeds ten. This assumption reduces data sparsity and allows us to estimate the Hierarchical MNL baseline. Our model and the two by-product baselines can be implemented on the full

data set; the substantive conclusions presented in Section 6.2 are robust. We provide summary statistics for the focal data set in Table 5.

6.2. Evaluation Results

For the model evaluation, we follow the approach used in the simulation study and create a holdout test set by splitting the data on the time dimension. The first 60 weeks serve for model training, and the last 10 weeks comprise the test data. We predict the purchase probabilities for all products and 1,000 customers.

To train the proposed product choice model, in the first stage, we apply P2V-MAP (Gabel et al. 2019) to the market basket data to derive product embedding. We use the pretrained embedding to initialize W_H, W_d, and W_∞. Pretraining is a common step in machine learning applications; it helps avoid local minima in supervised learning and facilitates generalizability (Erhan et al. 2010). Initializing product embedding in our neural network with the output of P2V-MAP also reduces the number of training iterations required to achieve model convergence by approximately 25%, with concomitant decreases in the required training time. Then in the second stage, we initialize the parameters of the bottleneck layers with product embedding, fine-tune the parameters, and train the rest of the parameters of the neural network by minimizing the binary cross-entropy loss (see Section 3.3). We calibrate the hyperparameters of the model by comparing the binary cross-entropy loss for a validation set over a small number of randomly sampled hyperparameter configurations. Random search is a common approach to configuring neural networks and typically produces solutions that are as good as grid-searched results, at a small fraction of the computation time (Bergstra and Bengio 2012). We initialize the hyperparameter search with the values from the simulation; a larger embedding size ($L = 50$) improves the test loss, and the model converges in fewer epochs ($n_{epoch} = 10$). For the other hyperparameters, the random search did not yield a significant loss improvement, so we use the same values as in the simulation. Because we calibrate the simulation to mimic the behavior of the empirical data, the similarities between the simulation study and the empirical application are not surprising. In line with the results of the simulation study, the parsimonious architecture prevents overfitting.

An important difference from the simulation study is that true purchase probabilities are unknown for the empirical application. We therefore focus on the binary cross-entropy loss metric, which compares predicted purchase probabilities with observed (binary) purchases, and report the evaluation results in Table 6. For interpretability, recall that the binary cross-entropy loss is equivalent to the negative log-likelihood score. The ranking of the models based on their predictive performance is in line with the results obtained from the simulated data, and our proposed model achieves a lower cross-entropy loss (higher log-likelihood) than the reference methods.

Table 6 Aggregate Predictive Performance (Empirical Application)

	Cross-Entropy Loss	Difference to Our Model
Our Model	0.01150	-
Binary Logit	0.01314	14.3%
LightGBM	0.01215	5.7%
Hierarchical MNL	0.01177	2.3%
Best Uniform	0.01712	48.9%

Notes: All differences are significant at $p < 0.01$, based on standard errors computed using a non-parametric bootstrap with 30 replications.

We conduct an additional regression analysis to understand performance differences between our model (DNN) and the baseline models. Specifically, we compute the binary cross-entropy loss for each observation (customer, product, time) in the test set for the DNN and the reference model, then compare the loss values using a linear regression:

M1: $L_{ijtm} = \alpha_0 + \alpha_c + \delta_{DNN}$

M2: $L_{ijtm} = \alpha_0 + \alpha_c + \delta_{DNN} + \delta_C + \delta_P + IPT_{ic} + \delta_{DNN} \times \delta_C + \delta_{DNN} \times \delta_P + \delta_{DNN} \times IPT_{ic}$,

where m indexes the model (DNN or reference model), α_0 is the regression intercept, α_c are category-level fixed effects, and IPT_{ic} is the average customer-level category interpurchase time computed on the training data. The regression includes three indicator variables:

$$\delta_{DNN} = \mathbb{I}_{ijtm}(m = DNN),$$

$$\delta_C = \mathbb{I}_{ijtm}(d_{itj} > 0),$$

$$\delta_P = \mathbb{I}_{ijtm}([\textstyle\sum_{k \in C} b_{i,t-1,k}] > 0),$$

i.e., δ_{DNN} identifies loss values corresponding to our model, δ_C marks observations (i, j, t) with a coupon, and δ_P is an indicator for observations with a category purchase in a previous period. We use the retailer's category definition to compute IPT_{ic} and δ_P.

The three interaction terms with the indicator variable for the neural network observations, δ_{DNN}, indicate whether the model loss is particularly low for the given data partitions (low is good). For readability, we multiply all loss values by a factor of 100. We repeat the analysis for the Binary Logit, LightGBM, and the Hierarchical MNL models, which produces six sets of regression coefficients (two nested model specifications and three model comparisons).

Table 7 presents the regression results. The results for the three models are consistent. The neural network achieves a significantly lower binary cross-entropy loss than the baseline models (δ_{DNN}). On average, predicting probabilities is more challenging for observations with coupons (δ_C), observations with a category purchase in the last week (δ_P), and smaller interpurchase times (IPT_{ic}).

The interaction terms reveal that the DNN model improves predictions for observations with a recent category purchase, beyond the average improvement measured by δ_{DNN}. The DNN binary cross-entropy loss also tends to be smaller than the loss in the reference models for coupon observations, but the effect is not statistically significant.

We conclude that the evidence from the empirical application confirms our analysis in the simulation study (Section 5). The proposed product choice model achieves significantly better predictive performance than the baseline models, and the outperformance margins are greater for observations with recent purchase or coupon discount events.

Table 7 Binary Cross-Entropy Loss Comparison

	Our Model vs. Binary Logit		**Our Model vs. LightGBM**		**Our Model vs. Hierarchical MNL**	
	(M1)	**(M2)**	**(M1)**	**(M2)**	**(M1)**	**(M2)**
Intercept α_0	1.549*** (0.015)	4.436*** (0.059)	1.430*** (0.015)	4.232*** (0.056)	1.416*** (0.015)	4.305*** (0.057)
DNN δ_{DNN}	-0.182*** (0.022)	-0.165*** (0.031)	-0.063*** (0.021)	-0.077*** (0.030)	-0.049*** (0.021)	-0.014 (0.030)
Coupon δ_C		1.529*** (0.099)		1.344*** (0.094)		1.454*** (0.095)
Category Purchase δ_P		2.051*** (0.039)		1.845*** (0.037)		2.204*** (0.037)
Interpurchase Time IPT_{ic}		-0.002*** (0.001)		-0.002*** (0.001)		-0.002*** (0.001)
DNN × Coupon		-0.118 (0.140)		0.067 (0.134)		-0.044 (0.134)
DNN × Category Purchase		-0.173*** (0.054)		0.039 (0.052)		-0.323*** (0.052)
DNN × Interpurchase Time		0.0003 (0.001)		0.0004 (0.001)		0.0002 (0.001)
Category Fixed Effects	Yes	Yes	Yes	Yes	Yes	Yes

Notes: To simplify the exposition, we scaled the binary cross-entropy loss values by a factor of 100.

7. Conclusion

Retailers collect high-quality data about customer choices and the effectiveness of marketing actions, but leveraging these data to support targeted marketing is challenging. Large assortments and customer bases require prediction and optimization methods that can scale, to both the number of products and the amount of training data, without extensive product attribute information.

In this paper, we develop a nonparametric model to predict product choice for the entire assortment of a large retailer in response to marketing actions. The model is motivated by the coupon personalization problem. Given coupon assignments and customer purchasing histories, our model predicts customer-specific product choice probabilities for every product in the assortment. The model is based on a custom deep neural network architecture and can be applied directly to transaction data gathered from loyalty card programs. Our model thereby eliminates the need for product attributes and assumptions about the category structure or cross-product effects.

In simulations, the proposed model significantly outperforms established benchmarks in predictive performance. The model identifies product similarities from the training data and approximates own- and cross-product coupon and inventory effects out-of-sample. The greater predictive accuracy leads to better performance in a simulated coupon personalization application. The coupon optimization methods achieve substantially higher revenue gains when using the purchase probabilities predicted by our model, compared to the baseline predictions. The model is particularly valuable if cross-category discount effects are more pronounced.

The empirical study based on data from a leading German grocery retailer verifies the predictive accuracy results from the simulation study. The improvements are particularly large for observations (i.e., customer, time, and product combinations) with a recent category purchase and observations with coupon assignments (though the latter is not statistically significant).

Our product choice model offers high practical value for retail analytics, beyond coupon personalization. The model applies to raw transaction data from loyalty programs, can scale to large product assortments and customer bases, and predicts customer purchase behavior in response to marketing actions. The modular neural network architecture allows simple modifications and extensions for different domain applications. These characteristics are the basis for targeting in retail.

Target marketing and product choice modeling accordingly provide a rich setting for future research. Our product choice model is based on a neural network. We expect the proposed architecture to further improve with continued advancements of the machine learning methods. For example, new methods might enable the model to accommodate changes in the product assortment, new product forecasting, or

learning from infrequent purchases. Qualitatively different inputs might also enhance the predictive accuracy of product choice models. Image data and customer reviews are promising information sources to extend modeling inputs and better guide targeting decisions.

Another direction for research might involve investigating the performance of the proposed product choice model if only historical transaction data are available. Experiments are costly for retailers, so developments and applications from causal inference and multi-armed bandit literatures might enable model calibration with less experimental data and practical implementations that are robust to changes in customer behavior (e.g., due to seasonality). Finally, further model refinements and new insights regarding the performance of the product choice model could stem from more applications to targeted marketing, including pricing, product recommendations, and promotion personalization.

References

Alain, G. and Bengio, Y. (2014). What Regularized Auto-Encoders Learn from the Data-Generating Distribution. *The Journal of Machine Learning Research*, 15(1):3563–3593.

Amano, T., Rhodes, A. and Seiler, S. (2019). Large-scale Demand Estimation with Search Data. *Working Paper*.

Ansari, A., and Mela, C. F. (2003). E-customization. *Journal of Marketing Research,* 40(2), 131–145.

Archak, N., Ghose, A., and Ipeirotis, P. G. (2011). Deriving the Pricing Power of Product Features by Mining Consumer Reviews. *Management Science*, 57(8):1485– 1509.

Arora, N., Dreze, X., Ghose, A., Hess, J. D., Iyengar, R., Jing, B., Joshi, Y., Kumar, V., Lurie, N., Neslin, S., et al. (2008). Putting One-to-One Marketing to Work: Personalization, Customization, and Choice. *Marketing Letters*, 19(3):305.

Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. (2007). Greedy Layer-Wise Training of Deep Networks. In *Advances in Neural Information Processing Systems*, pages 153–160.

Bergstra, J. and Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, 13(Feb):281–305.

Bradlow, E. T., Gangwar, M., Kopalle, P., and Voleti, S. (2017). The Role of Big Data and Predictive Analytics in Retailing. *Journal of Retailing*, 93(1):79–95.

Breese, John S., David Heckerman, and Carl Kadie. "Empirical analysis of predictive algorithms for collaborative filtering." *arXiv preprint* arXiv:1301.7363 (2013).

Chintagunta, P.K., 1993. Investigating purchase incidence, brand choice and purchase quantity decisions of households. *Marketing Science*, 12(2), pp.184-208.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural Language Processing (Almost) From Scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Dubé, J.-P. H. and Misra, S. (2017). Scalable Price Targeting. Available at SSRN: https://ssrn.com/abstract= 2992257.

Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P., and Bengio, S. (2010). Why Does Unsupervised Pretraining Help Deep Learning? *Journal of Machine Learning Research*, 11(Feb):625–660.

Fader, P. (2012). *Customer Centricity: Focus on the Right Customers for Strategic Advantage*. Wharton Digital Press.

Fader, P. S. and Hardie, B. G. (1996). Modeling Consumer Choice Among SKUs. *Journal of Marketing Research*, 33(4), pp.442-452.

Fader, P.S., Hardie, B.G. and Lee, K.L., (2005). RFM and CLV: Using iso-value curves for customer base analysis. *Journal of Marketing Research*, 42(4), pp.415-430.

Forrester (2017). Demystifying Price and Promotion: Shoppers Bust Long Held Myths on Pricing and Promotions. http://www.parkeravery.com/media/forrester- study-demystifying-price-and-promotion.pdf (Accessed 2019-04-03).

Gabel, S., Guhl, D. and Klapper, D., 2019. P2V-MAP: Mapping Market Structures for Large Retail Assortments. *Journal of Marketing Research*, 56(4), pp.557-580.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Grewal, D., Roggeveen, A. L., and Nordfält, J. (2017). The Future of Retailing. *Journal of Retailing*, 93(1):1–6.

Guadagni, P. M. and Little, J. D. (1983). A Logit Model of Brand Choice Calibrated on Scanner Data. *Marketing Science*, 2(3):203–238.

Hanssens, D. M. (2014). Econometric Models. In *The History of Marketing Science*, pages 99–128. Singapore: World Scientific Publishing.

Hauser, J.R., Urban, G.L., Liberali, G. and Braun, M. (2009). Website morphing. *Marketing Science*, 28(2), pp.202-223.

Jacobs, B. J., Donkers, B., and Fok, D. (2016). Model-Based Purchase Predictions for Large Assortments. *Marketing Science*, 35(3):389–404.

Johnson, J., Tellis, G. J., and Ip, E. H. (2013). To Whom, When, and How Much to Discount? A Constrained Optimization of Customized Temporal Discounts. *Journal of Retailing*, 89(4):361–373.

Kamakura, Wagner A., and Gary J. Russell (1989). A probabilistic choice model for market segmentation and elasticity structure. *Journal of Marketing Research* 26(4): 379-390.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems*, pages 3146–3154.

Kingma, D. P. and Ba, J. (2014). Adam: A Method for Stochastic Optimization. arXiv Preprint arXiv:1412.6980.

Koren, Y., Bell, R. and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8), pp.30-37.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep Learning. *Nature*, 521(7553):436–444.

Lemmens, A. and Gupta, S., 2020. Managing churn to maximize profits. *Marketing Science*, Forthcoming.

Levy, O. and Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems* (pp. 2177-2185).

Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendation: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1), 76–80.

Liu, L., Dzyabura, D., and Mizik, N. (2018). Visual Listening In: Extracting Brand Image Portrayed on Social Media. Available at SSRN: https://ssrn.com/abstract=2978805.

Liu, X., Lee, D., and Srinivasan, K. (2017). Large Scale Cross Category Analysis of Consumer Review Content on Sales Conversion Leveraging Deep Learning. Available at SSRN: https://ssrn.com/abstract=2848528.

Lu, S., Xiao, L. and Ding, M. (2016). A video-based automated recommender (VAR) system for garments. *Marketing Science*, 35(3), pp.484-510.

Maaten, L. v. d. and Hinton, G. (2008). Visualizing Data Using t-SNE. *Journal of Machine Learning* Research, 9(Nov):2579–2605.

Maaten, L., v. d. (2014). Accelerating t-SNE using tree-based algorithms. *The Journal of Machine Learning Research*, 15(1), pp.3221-3245

Manchanda, P., Ansari, A., and Gupta, S. (1999). The Shopping Basket: A Model for Multicategory Purchase Incidence Decisions. *Marketing Science*, 18(2):95–114.

McFadden, D. (1974). The Measurement of Urban Travel Demand. *Journal of Public Economics*, 3(4):303–328.

Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C. and Joulin, A., 2017. Advances in pre-training distributed word representations. *arXiv preprint arXiv:1712.09405*.

NCH Marketing Services (2019). 2018 Year-End Coupon Facts at a Glance. https:// www.nchmarketing.com/2018-year-end-coupon-facts-at-a-glance.aspx (Accessed 2019-04-03).

Neslin, S. A., Van Heerde, H. J., et al. (2009). Promotion Dynamics. *Foundations and Trends in Marketing*, 3(4):177–268.

Nijs, V.R., Dekimpe, M.G., Steenkamps, J.B.E. and Hanssens, D.M. (2001). The Category-Demand Effects of Price Promotions. Marketing Science, 20(1), pp.1-22.

Orhan, A. E. and Pitkow, X. (2017). Skip Connections Eliminate Singularities. arXiv Preprint arXiv:1701.09175.

Peppers, D. and Rogers, M. (1997). The One to One Future: Building Relationships One Customer at a Time. Currency-Doubleday.

Rossi, P. E., McCulloch, R. E., and Allenby, G. M. (1996). The Value of Purchase History Data in Target Marketing. *Marketing Science*, 15(4):321–340.

Rossi, P.E., Allenby, G.M., and McCulloch, R. (2012). Bayesian statistics and marketing. John Wiley & Sons.

Ruiz, F.J., Athey, S. and Blei, D.M., 2020. Shopper: A Probabilistic Model of Consumer Choice With Substitutes and Complements. *Annals of Applied Statistics*, 14(1), pp.1-27.

Vincent, P., Larochelle, H., Bengio, Y. and Manzagol, P.A., (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning* (pp. 1096-1103).

Russell, G. J. and Petersen, A. (2000). Analysis of Cross Category Dependence in Market Basket Selection. *Journal of Retailing*, 76(3):367–392.

Simester, D., Timoshenko, A., and Zoumpoulis, S. I. (2019a). Targeting Prospective Customers: Robustness of Machine Learning Methods to Typical Data Challenges. *Management Science* (forthcoming).

Simester, D., Timoshenko, A., and Zoumpoulis, S. I. (2019b). Efficiently Evaluating Targeting Policies: Improving Upon Champions vs. Challenger Experiments. *Management Science* (forthcoming).

Smith, A.N., Rossi, P.E. and Allenby, G.M. (2019). Inference for Product Competition and Separable Demand. *Marketing Science*, 38(4), pp.690-710.

Tajbakhsh, N., Shin, J.Y., Gurudu, S.R., Hurst, R.T., Kendall, C.B., Gotway, M.B. and Liang, J., 2016. Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE transactions on medical imaging*, 35(5), pp.1299-1312.

Timoshenko, A. and Hauser, J. R. (2019). Identifying Customer Needs from User-Generated Content. *Marketing Science*, 38(1):1–20.

Walmart (2005). Our Retail Divisions. http://corporate.walmart.com/_news_/ news-archive/2005/01/07/our-retail-divisions (Accessed 2016-12-30).

Walmart (2016). Walmart.com's History and Mission. https://help.walmart.com/ app/answers/detail/a_id/6 (Accessed 2016-12-30).

Winer, R. S. and Neslin, S. A. (2014). *The History of Marketing Science*. World Scientific.

Xu, B., Wang, N., Chen, T., and Li, M. (2015). Empirical Evaluation of Rectified Activations in Convolutional Network. arXiv Preprint arXiv:1505.00853.

Zhang, J. and Wedel, M. (2009). The Effectiveness of Customized Promotions in Online and Offline Stores. *Journal of Marketing Research*, 46(2):190–206.

Zhang, M. and Luo, L. (2018). Can User Generated Content Predict Restaurant Survival: Deep Learning of Yelp Photos and Reviews. Available at SSRN: https://ssrn.com/abstract=3108288.