# Product2Vec: Leveraging representation learning to model consumer product choice in large assortments

Fanglin Chen
New York University
fchen@stern.nyu.edu[†]

Xiao Liu
New York University
xliu@stern.nyu.edu

Davide Proserpio
University of Southern California
proserpi@marshall.usc.edu

Isamar Troncoso
Harvard University
itroncoso@hbs.edu

Friday 1 July, 2022

[†]Author names are listed in alphabetical order.

## Abstract

We propose a method, Product2Vec, based on representation learning, that can automatically learn latent product attributes that drive consumer choices, to study product-level competition when the number of products is large. We demonstrate Product2Vec's interpretability and capability for scalable causal inference. For interpretability, first, we theoretically demonstrate that there exists a direct link between product vectors and product attributes by deriving a formal proof. Second, we use product embedding to create two metrics, complementarity and exchangeability, that allow us to distinguish between products that are complements and substitutes, respectively. For causal inference, we combine product vectors with choice models and show that we can achieve better accuracy—both in terms of model fit and unbiased price coefficients—when compared to a model based solely on observable attributes, and obtain results similar to those obtained with a more complex model that includes a fixed effect for every product.

# 1 Introduction

Discrete choice models are a staple for marketers to study product competition, discover consumer preferences, and design personalized marketing strategies. However, estimating discrete choice models has become computationally challenging for today's retail landscape because of product proliferation. For example, supermarkets often carry hundreds of SKUs in the cereal or carbonated beverage categories, and online shops often sell thousands of alternatives of headphones or vacuum cleaners. Restricting choice sets to brands instead of SKUs can mitigate the computation burden, but it is not ideal for product managers who want to understand cannibalization and design product line portfolios, or for retail managers who want to optimize the assortment.

Multiple attempts have been made to model consumer choices at the SKU level, but their limitations loom large. Aggregate models such as BLP (Berry et al. 1995) can handle a large number of SKUs with product fixed effects, but they are not suitable for personalized marketing, because the aggregation nature does not allow the econometrician to identify each individual's preference for products or price. Models that leverage individual purchase data can capture consumer heterogeneity, but they are either not scalable to large choice sets (Chintagunta and Dube 2005), or they assume that purchase decisions are driven by a relatively small number of observable product characteristics (Fader and Hardie 1996). In cases with missing product attributes, for instance when product characteristics are not readily available or when the nature of the product category requires defining attributes that are not easily measurable (e.g., healthiness of drinks or aesthetics and style of clothing), the latter models might make incorrect inferences and lead to suboptimal policy designs.

To overcome these limitations, in this paper, we propose a model based on representation learning that can automatically learn the (potentially latent) product attributes that drive consumer choices (Mikolov et al. 2013a, Turian et al. 2010, Al-Rfou et al. 2013). We rely on representation learning algorithms because they are unsupervised and can easily be applied to very large datasets.

In natural language processing, representation learning algorithms take massive collections of text as input and produce continuous word vectors—also called word embeddings—as output. The word embeddings are designed to capture semantic similarities between words: words that appear in similar contexts in the corpus of text (i.e., words that are surrounded by similar words) will be close to each other in the word vector space. Using the same logic, we treat shopping baskets as sentences and products as words and use representation learning to transform each product into a vector (Grbovic et al. 2015, Barkan and Koenigstein 2016, Gabel et al. 2019). When using text as input, word embeddings capture semantic similarities. Likewise, when using products as input, product embeddings can capture relationships among products.

To decipher how the product embeddings preserve relationships among products, we project product embeddings onto two-dimensional market structure maps. Three stylized facts arise. First, the market structure maps created with product embeddings generate product clusters. Products in the same cluster are often copurchased and often share similar product attributes. It seems that copurchases are driven by product attributes, and these clusters recover such attributes. Second, the randomness in the representation learning estimation process renders the product embedding dimensions unstable. However, the product embedding cluster memberships remain strikingly stable. Third, raw product vectors and derived similarity relationships do not reveal whether two products are complements or substitutes.

These stylized facts motivate us to theoretically understand the product embeddings and provide interpretability to the embeddings. Our interpretation analyses are decomposed in two parts. In part one, we establish a direct link between product vectors and product attributes by deriving a formal proof. The proof demonstrates a novel finding that product embedding clusters can be interpreted as product attribute combinations. If the market consists of M different segments of consumers with different preferences for product attributes, and there are M unique product attribute combinations that cater to each consumer segment, then product embeddings will create M clusters to reflect these heterogeneous consumers' preferences for the unique product attribute combinations.

In part two, we devise another way to interpret the embeddings, by defining two interpretable measures, complementarity and exchangeability, that allow us to distinguish between products that are complements and substitutes, respectively.

Having established the interpretability of the product embeddings, we next turn to causal inference and discuss how to combine embeddings and choice models to improve model accuracy while limiting the number of parameters to be estimated.

Next we test the proof predictions, the metrics we created, and the choice model application using simulated data. We start by showing that product clusters map to product attributes. We then show that complementarity and exchangeability successfully capture product complementarity and substitution, respectively, and as such, they can be valuable alternatives to studying product competition that do not require estimating any demand model. Finally, we estimate a choice model that includes a dummy for each of the product clusters identified by the product embeddings. We show that including these clusters improves the accuracy of the model—both in terms of fit and unbiased price coefficients— when compared to a model based on solely observable attributes, and leads to results similar to those obtained with a model that includes a fixed effect for each product.

We conclude the paper by testing our approach on a real dataset of consumer purchase data obtained from NielsenIQ, and obtain results consistent with those obtained with simulated data, thereby showing the practical applicability of our approach.

To summarize, this paper's contributions are threefold. First, we leverage representation learning to model consumer product choices in large assortments and provide interpretabilty for the product embedding by deriving a theoretical proof that links product attributes to product vectors. Second, we create two economically meaningful metrics, exchangeability and complementarity, to discover whether products are substitutes or complements that rely solely on product vectors and do not require estimating any demand model. Third, we show both with simulated and real data that combining product embeddings and choice models, we can correctly recover price elasticities using a limited number of parameters.

Overall, the results presented in this paper suggest that machine learning methods such as representation learning can help marketers study competition in an agnostic way, without the need for making any assumption about the data, and in a more scalable way by limiting the number of parameters to be estimated relative to a baseline product fixed effects model.

The remainder of the paper is organized as follows. In Section 2, we discuss the related literature and how our work connects to the previous research. In Sections 3 to 6, we describe our modeling framework. We start with a description of the representation learning algorithm used to obtain product vectors (or product embeddings) in Section 3 and discuss a few stylized facts in Section 4. Based on the stylized facts, we provide interpretability to the product embeddings in Section 5. Section 6 presents the causal inference framework and explains how to leverage product vectors in the estimation of these traditional choice models. Section 7 verifies the interpretability and causal inference predictions with a simulation study. Section 8 presents an empirical application. Section 9 discusses the implications of our findings and provides future research directions and concluding remarks.

# 2 Related literature

Our paper relates to two strands of the literature: choice models and applications of representation learning methods to marketing problems.

Choice models are one of the most widely used methods to understand consumer purchase decisions and study competition (see Chandukala et al. (2008) and Winer and Neslin (2014) for an extensive review of these models in marketing). These models have been used to study brand-level (Guadagni and Little 1983) and product-level competition (Fader and Hardie 1996, Chintagunta and Dube 2005). At the core of these models is the assumption that consumers make their choices by evaluating products based on their characteristics/attributes (i.e., they choose the product with characteristics that maximize their utility). In practice, product characteristics might not be fully observed by the analyst. For instance, retailers that manage large assortments might not obtain

or maintain extensive data on all their product attributes (Gabel and Timoshenko 2021), or there could be certain product categories with characteristics that are coarse or hard to quantify, for example, movies, books, or cereal (Armona et al. 2021). Omitting relevant characteristics in the specification of a choice model can lead to biased estimates. A possible solution to this omitted variable bias is to incorporate product fixed effects in the model, but this approach does not scale well when the number of products is large.

In the last few years, there has been a growing interest in finding ways to make choice models more scalable and accurate. A growing stream of research in marketing and economics leverages machine learning algorithms to model consumer choices for those purposes. For example, Jacobs et al. (2016) propose an extension to the latent Dirichlet allocation model that identifies purchase "topics" (motivations) and predicts purchase behavior more accurately. Ruiz et al. (2019) propose the SHOPPER model, a hierarchical latent variable model for sequential product choices that parameterizes latent product attributes and customer preferences, while accounting for price sensitivities and seasonal effects. Gabel and Timoshenko (2021) develop a deep neural network model that predicts customer-specific purchase probabilities in response to marketing actions (e.g., personalized coupons). In a similar vein, we present an approach that leverages representation learning, a popular machine learning approach with applications in several domains, to automatically learn product attributes (either observed or latent) that drive consumer choices.

Marketing scholars have adopted representation learning methods for various applications (see Table 12 in Appendix A for an overview). Early applications of these methods in the context of recommender systems include Grbovic et al. (2015), who apply the word2vec model to a dataset with e-mail receipt logs to improve personalized product ads, and Barkan and Koenigstein (2016), who apply the word2vec model for item-based collaborative filtering. In both cases, the authors use the word2vec model to obtain meaningful product-level embeddings that capture "similarity" between products, which is further exploited to obtain more accurate product-level purchase predictions. Similar to these papers, we also use the word2Vec model to obtain latent representations of products. However, we make three distinctive contributions. First, we provide interpretability

to the embeddings and discover a link between the embeddings and product attributes. Second, we use these embeddings to derive economically meaningful metrics, that is, complementarity and substitution between products. Third, we use the embeddings to estimate unbiased price elasticities.

Another stream of research closely related to ours uses representation learning methods to identify market structure and product/brand competition. For example, Gabel et al. (2019) propose an exploratory approach to identify market structure that applies the word2vec model to learn latent product relationships from shopping baskets. To empirically validate their approach, the authors compare their results with the market structure given by a simulated dataset. Yang et al. (2021) propose a framework to identify a fluid product-market that uses autoencoder techniques to learn latent brand relationships from social media users' brand engagement data. The authors empirically validate their approach by comparing their results with external datasets about market structure. Our work contributes to this stream of literature by theoretically validating the empirical findings of these papers.

More recently, marketing scholars have started to combine the outputs of representation learning methods with other approaches. For example, Kumar et al. (2020) propose a framework to design bundles in a large-scale cross-category retail setting that leverages product embeddings learned from purchase data and search data. More specifically, the authors exploit the purchase-based embeddings and the search-based embeddings to capture complementarity and substitutability between products and improve purchase predictions. Armona et al. (2021) propose to augment the BLP demand model (Berry et al. 1995) with latent product characteristics and consumer preferences learned from search data. The authors show that closeness in the latent product space predicts competition and that including latent product characteristics in the demand model improves post-merger predictions. Our work contributes to this stream of literature in two ways. First, different from Kumar et al. (2020), we propose metrics to capture product complementarity and substitution solely from purchase data. Second, we add to Armona et al. (2021) by illustrating the value of using product embeddings to capture cross-category elasticity.

# 3 Product2Vec: learning the vector representations of products

Similar to Grbovic et al. (2015) and Barkan and Koenigstein (2016), we adopt the framework of word2vec (Mikolov et al. 2013a) to transform products into low-dimensional vectors. To illustrate how the model works, consider a set of purchase baskets B and a set of products S. Given a product $s_i$ (focal product) in basket $b$, the goal of the model is to predict the other products $s_{i+j}$ in the same basket (context products). Thus, the objective function is

$$\mathscr{V}, \mathscr{V}' = \arg\max_{\mathscr{V}, \mathscr{V}'} \sum_{b \in B} \sum_{s_i \in b} \sum_{-c \leq j \leq c} \log \mathbb{P}(s_{i+j} | s_i; \mathscr{V}, \mathscr{V}'). \tag{1}$$

where $\mathscr{V} = \{v_s\}_{s=1}^{S}$, $\mathscr{V}' = \{v_s^{'}\}_{s=1}^{S}$ are the collections of "input" vectors $v_s$ and "output" vectors $v_s^{'}$, respectively. $c$ is the length of the context for product sequences, and $\mathbb{P}(s_{i+j}|s_i)$ is the conditional probability of observing the context product $s_{i+j}$ given the focal product $s_i$ defined by the softmax function (Mikolov et al. 2013b):[1]

$$\mathbb{P}(s_{i+j} | s_i; \mathscr{V}, \mathscr{V}') = \frac{exp(v_{s_i}^T \, v_{s_{i+j}}^{'})}{\sum_{s=1}^{S} exp(v_{s_i}^T \, v_s^{'})}. \tag{2}$$

It is impractical to directly calculate $\mathbb{P}(s_{i+j}|s_i; \mathscr{V}, \mathscr{V}')$, because the cost of computing the denominator is proportional to the total number of unique SKUs ($S$). Instead, word2vec employs the negative sampling technique to approximate the log probability of the softmax (Mikolov et al. 2013b):

$$\log \mathbb{P}(s_{i+j} | s_i; \mathscr{V}, \mathscr{V}') = \log \sigma(v_{s_i}^T \, v_{s_{i+j}}^{'}) + \sum_{k=1}^{K} \mathbb{E}_{s_k} \log \sigma(-v_{s_i}^T \, v_{s_k}^{'}). \tag{3}$$

In Equation 3, $s_k$ is a SKU randomly drawn from the whole training set based on the distribution of purchase frequency, $K$ is the number of negative samples for each focal product, and $\sigma()$ is the sigmoid function. Equation 3 consists of two components: the first component maximizes the probability that the current product occurs together with its context products, while the second

[1] If multiple units of a SKU are purchased, we treat it as if the SKU appears only once. We do not consider purchase quantity because it has little to do with relationships among products.

minimizes the likelihood that the current product appears along with some randomly selected, irrelevant products. In other words, this objective function distinguishes observations from noise.

The current specification has a desirable property, in that products purchased in similar shopping baskets have similar product vectors. To see this, suppose there are two SKUs $s_{i_1}$ and $s_{i_2}$ that appear in different baskets but have the same third SKU $s_c$ in their contexts. According to Equation 1, the conditional probabilities $\mathbb{P}(s_c|s_{i_1})$ and $\mathbb{P}(s_c|s_{i_2})$ are maximized and close to each other. From Equations 2 and 3, the probabilities above depend on only the vector representations of $s_{i_1}, s_{i_2}, s_c$, and noise. Since the product vector of $s_c$ is unique, it implies that the product vectors of $s_{i_1}$ and $s_{i_2}$ will be similar. In other words, the similarity between product vectors reflects the extent to which the corresponding products are found in similar baskets.

# 4 Stylized facts

How can we use the product embeddings to understand consumer preferences for products and price? This section provides three stylized facts of product embeddings.

First, the market structure maps created with product embeddings generate product clusters. Products in the same cluster are often copurchased and often share similar product attributes. It seems that copurchases are driven by product attributes, and these clusters recover such attributes. For instance, Gabel et al. (2019) find that clusters are driven by distinct product attributes and include products designed to appeal to specific consumer segments, such as clusters of organic, vegetarian, lactose free and gluten free products. In our own application (discussed in Section 8.2), we find embedding clusters of beverages that preserve packages and size.

Second, the random elements in the representation learning estimation process render the product embedding dimensions unstable, but the product embedding cluster memberships remain strikingly stable. As explained in the previous section, estimating product vectors requires negative sampling and downsampling of frequent products. When different random number generator seeds are used, the resulting embeddings will be drastically different. However, we (see Figure 9) as well

as Gabel et al. (2019) find that product maps created by product embeddings are highly consistent across different seeds for the random number generator.

Third, raw product vectors and derived similarity relationships do not reveal whether two products are complements or substitutes. Although product similarity captures when two products share similar attributes and therefore are likely to be *related*, it cannot distinguish whether such a relationship is of complementarity or substitution. To see this, consider that two complementary products, A and B, are usually copurchased and share the same context products, and therefore have similar product vectors. Nevertheless, two strong substitutes, A and C, that are never copurchased might still share similar context products. For instance, baskets might be identical in all products except for product A or C (because only one of them will be in the basket); hence, A and C will also have similar context products and product vectors.

The three stylized facts discussed above seem to suggest that due to their randomness, embeddings themselves cannot be mapped to any observable economic constructs or model primitives. However, the clusters of embeddings could, as they reflect stable relations among products. Moreover, naive similarities between product embeddings, that is, their pairwise distance, cannot help us differentiate complements from substitutes, but there may alternative ways to leverage product embeddings to capture complementarity and substitution patterns. In the next section, we formalize these intuitions.

# 5 Interpretation

We develop two ways to interpret the product embeddings. Section 5.1 directly interprets product embeddings as product attribute combinations using a formal proof. Section 5.2 indirectly transforms product embeddings into interpretable economic measures, complementarity and exchangeability.

## 5.1 Proof

In this section, we present a theoretical proof that demonstrates that the product embeddings generated with the Product2Vec model have economic interpretations and that their clusters can replace the time-invariant product fixed effects in choice models.

The intuition behind the proof is as follows. Because product attributes drive copurchase behaviors, in a particular category, two products with the same attribute combinations will have the same cross-category copurchase patterns. And because copurchase patterns are the object fuction for deriving product fixed effects in choice models as well as product embeddings in product2vec, therefore, when two products have the same copurchase patterns, they will share the same product fixed effects and the same product embedding cluster membership. Consequently, product fixed effects in the choice model can be replaced by embedding cluster fixed effects.

We start by introducing a model for consumers' purchase decisions and data generating process (DGP) in Section 5.1.1. Then in Section 5.1.2, we show the relationship between the number of individual-specific time-invariant product utilities and the number of unique copurchase patterns in the product fixed effects model. Finally, we connect the number of product embedding clusters with the number of unique copurchase patterns in Section 5.1.3.

### 5.1.1 Data generating process

We model consumers' purchase decisions as a sequential process, in which consumers first decide which categories to buy, then decide which product to buy within a given category.

#### 5.1.1.1 Category choice

For category choices, we use a multivariate probit specification, in which the latent category utility $z_{ict}$ that consumer $i \in \{1,\ldots,I\}$ derives from buying a product in category $c \in \{1,\ldots,C\}$ during

purchase occasion $t \in \{1,\ldots,T\}$ is given by

$$\begin{aligned} z_{ict} &= \bar{z}_c + \omega_{ict} \\ \overrightarrow{\omega_{it}} &= (\omega_{i1t},\ldots,\omega_{iCt})^T, \overrightarrow{\omega_{it}} \sim N(0,\Omega), \end{aligned} \tag{4}$$

where $\bar{z}_c$ is the base category utility and captures the overall popularity of category $c$, and $\omega_{ict}$ is the individual- and time-specific category utility. The covariance matrix $\Omega$ can accommodate category complements (when $\text{Cov}(c,c') > 0$, e.g., cereal and yogurt) and substitutes (when $\text{Cov}(c,c') < 0$, e.g., tea and coffee).

Let $y_{ict} \in \{0,1,\ldots,J_c\}$ be the product choice of consumer $i$ in category $c$ at time $t$, where $J_c$ is the number of products in category $c$, and $J = \sum_{c=1}^{C} J_c$ is the total number of products across all categories. The consumer's category choice rule is

$$y_{ict} \begin{cases} > 0 & \text{if } z_{ict} > 0 \\ = 0 & \text{otherwise} \end{cases} \tag{5}$$

That is, the consumer buys a product in category $c$ if the category utility $z_{ict}$ is positive.

#### 5.1.1.2 Product choice

To model product choices within a given category, we use a multinomial logit specification. Following Singh et al. (2005), we conceptualize that products are bundles of attributes and that the consumer's utility is affected by product attributes and price. The intrinsic utility for a product is a function of underlying attributes, and the attributes are common across categories. The latent product utility $u_{ijt}$ that consumer $i$ derives from buying a product $j \in \{1,\ldots,J_c\}$ during purchase occasion $t$ is given by

$$u_{ijt} = \sum_k \alpha_{ik} X_{jk} + \delta_i Q_j + \beta_i P_{jt} + \varepsilon_{ijt} \tag{6}$$

When $y_{ict} > 0$ in Equation 5, the consumer's product choice rule is

$$y_{ict} = j \text{ if } u_{ijt} = \max_{j' \in \{1,\ldots,J_c\}} \left\{ u_{ij't} \right\} \tag{7}$$

That is, given buying in category $c$, the consumer buys the product with the highest product utility $u_{ijt}$ within the category.

This model setup can generate the shopping basket data $\overrightarrow{Y_{it}} = \{y_{i1t}, \ldots, y_{iCt}\}$. Next we discuss each component in Equation 6.

**Product attributes $\mathbf{X_{jk}}$** $X_{jk}$ is the value for attribute $k$ of product $j$ and there are $K$ attributes in total. We use these attributes $X_{jk}$ to capture products' horizontal differentiation, and they can be either observed or unobserved by the researcher. For instance, an observed attribute could be package size, whereas an unobserved attribute can be whether a product is healthy or not. Importantly, we follow the prior literature on multi-category brand choice models (Singh et al. 2005, Prasad et al. 2008) and assume that these attributes are the same across product categories. For instance, consider a consumer who wants to purchase products in two categories, cereal and yogurt. She has limited storage space and is highly health conscious; therefore, she prefers small-size and low-sugar products in both categories.

**Quality $\mathbf{Q_j}$ and price $\mathbf{P_{jt}}$** Quality $Q_j$ represents products' vertical differentiation (high quality vs. low quality), and $P_{jt}$ corresponds to the price of product $j$ during the purchase occasion $t$. Consumers could have different price sensitivities; in other words, some consumers may prefer high-quality and high-price products, whereas other consumers prefer low-quality and low-price products.

**Price decomposition** We decompose the total price $P_{jt}$ into the average price $\bar{P}_j$ and the cross-time price variation $\Delta P_{jt}$; that is, $P_{jt} = \bar{P}_j + \Delta P_{jt}$. We assume that the cross-time price variation $\Delta P_{jt}$ follows a normal distribution $\Delta P_{jt} \sim N\left(0, \sigma_j^2\right)$ with mean 0 and standard deviation $\sigma_j$. Importantly, we assume that the price coefficient $\beta$ applies to both the average price $\bar{P}_j$ and the cross-time price variation $\Delta P_{jt}$. Then Equation 6 becomes

$$u_{ijt} = \sum_{k=1}^{K} \alpha_{ik} X_{jk} + \delta_i Q_j + \beta_i \left(\bar{P}_j + \Delta P_{jt}\right) + \varepsilon_{ijt} \tag{8}$$

**Other components** $\varepsilon_{ijt}$ is the idiosyncratic error term that follows the type-1 extreme value distribution. The parameters $\alpha_{ik}$, $\delta_i$, and $\beta_i$ capture the heterogeneous consumer preferences for product attributes, quality, and price, respectively.

**Vectorization** We combine the consumer preference parameters $\{\alpha_{ik}\}_{k=1}^{K}, \delta_i, \beta_i$ into a vector $\overrightarrow{\theta_i}$, and name it as consumer $i$'s attribute preference vector. We assume that there are $S$ distinct segments of consumers, and consumers within the same segment share the same attribute preference vector. So, if both consumer $i$ and $i'$ belong to segment $s$, then $\overrightarrow{\theta_i} = \overrightarrow{\theta_{i'}} = \overrightarrow{\theta_s}$. In addition, because $\{X_{jk}\}_{k=1}^{K}, Q_j, \bar{P}_j$ are all time-invariant product characteristics, we combine them into a vector $\overrightarrow{A_j}$ and name it as the attribute vector for product $j$. We can now rewrite Equation 8 as

$$u_{ijt} = \overrightarrow{\theta_i}^T \overrightarrow{A_j} + \beta_i \Delta P_{jt} + \varepsilon_{ijt} \tag{9}$$

Because this model assumes that consumer choices are driven by product attributes and consumer preferences for these attributes, we call it the attribute model. Note that "attributes" are defined loosely here and include not only regular product attributes but also quality and average price.

#### 5.1.1.3 Product attributes and consumer purchase

**Intuition of dimension reduction** We assume that consumers have limited cognitive capability to consider all possible products separately. Instead, they make purchase decisions by considering only a limited number of unique product attribute combinations (Fader and Hardie 1996). Mathematically, let the cardinality of $\overrightarrow{A_j}$ be $\|A\|$. And even though product attributes can be continuous variables instead of discrete ones, we assume that consumers' bounded rationality only allows them to consider discrete levels of product attributes. We assume that $\|A\| \ll J_c$; that is, the number of unique product attribute combinations is much smaller than the total number of products in a particular category. Our goal is to reduce the number of dimensions in the product fixed effects model from $J_c$ to $\|A\|$.

**Products with the same attributes lead to within-category substitution** In one category, many products can share the same attributes. Based on the multinomial logit specification in our model, these products have positive cross-price elasticities and are substitutes; in other words, they satisfy similar desires and are used in place of each other. For example, in the carbonated beverage category, Fanta and Crush are substitutes because they have the same attributes of orange flavor and soda taste.

**Products with the same attributes lead to cross-category copurchase/complement** Looking across categories, our model implies that products with the same attributes in different categories are more likely to be copurchased by the same segment of consumers, hence are complements. For example, a health-conscious consumer could purchase fat-free milk, high-fiber bread, low-sugar chewing gum, and Diet Coke.

This happens because if a consumer in segment $s$ purchases product $j$ in category $c$, then Equation 7 implies that the utility of product $j$ is the highest within the category. Looking at Equation 9, if we assume that product attributes, rather than the cross-time price variation or the idiosyncratic shocks, are the primary drivers of product utilities and thus consumer choices, then the inner product of segment $s$'s preference vector and product $j$'s attribute vector ($\overrightarrow{\theta_s}^T \overrightarrow{A_j}$) is higher than that of other products in category $c$. If product $r$ in another category $c'$ shares the same attributes as product $j$ (i.e., $\overrightarrow{A_j} = \overrightarrow{A_r}$), then the inner product of segment $s$'s preference vector and product $r$'s attribute vector ($\overrightarrow{\theta_s}^T \overrightarrow{A_r}$) will also be higher than that of other products in category $c'$. Thus, product $r$ yields the highest product utility within the category $c'$, and $j$ and $r$ will be copurchased by segment $s$.

In Appendix B.1, we illustrate the intuition behind this model using a simplified example.

#### 5.1.1.4 Unique copurchase patterns

Now we consider the number of unique cross-category copurchase patterns when there are multiple consumer segments with different attribute preferences. We define a a consumer $i's$ cross-category

copurchase pattern of a focal product as the conditional purchase probabilities of all products in all categories excluding those in the same category as the focal product. Mathematically, the copurchase pattern for the focal product $y_{ict}$ is $\{Pr(y_{ic't}|y_{ict})\}_{c'\neq c}$, where $Pr(y_{ic't}|y_{ict})$ denotes the purchase probability of product $y_{ic't}$ conditional on purchasing product $y_{ict}$ in the same shopping basket. We can show that the number of unique copurchase patterns for each consumer segment, denoted as $H$, equals the minimum of $S$ and $\|A\|$, and the proof is in Appendix B.2.

**Proposition 1.** *Consider all $J_c$ products in a focal product category c, the number of unique copurchase patterns for each consumer segment, denoted as H, is equal to the minimum of the number of consumer segments S and the number of unique attribute combinations $\|A\|$.*

$$H = \min(S, \|A\|) \tag{10}$$

Without loss of generality, from now on we assume that the number of consumer segments $S$ is equal to the number of unique attribute combinations $\|A\|$, and denote it as the number of consumer distinguishable attribute combinations $\Lambda$; that is, $S = \|A\| = \Lambda$.

It is worth noting that when we consider cross-category copurchase, some attribute combinations may not exist in all categories. For example, the flavor attribute will apply to food- and beverage-related categories, but not to home cleaning products. Thus, to be precise, $\|A\|$ should be the *maximum* number of unique attribute combinations among all categories. Without loss of generality, we assume that all categories have the same number of unique attribute combinations; that is, $\|A_c\| = \|A\|, \forall c$.

Next, we consider two models to fit the simulated data: (1) the product fixed effects model, and (2) the Product2Vec model. We derive the number of unique copurchase patterns in these two models and demonstrate that we could use product clusters generated from product embeddings to replace product fixed effects.

### 5.1.2 Number of unique copurchase patterns in the product fixed effects model

When we estimate the product fixed effects model, where each product is represented with a two-way fixed effect to capture unobserved individual- and product-specific factors, the utility function in Equation 9 can be written as

$$u_{ijt} = \gamma_{ij} + \beta_i \Delta P_{jt} + \varepsilon_{ijt}, \tag{11}$$

where $\gamma_{ij} = \overrightarrow{\theta_i}^T \overrightarrow{A_j}$ represents the individual-specific time-invariant product utility. It contains the consumer's utility derived from three components: product attributes, quality, and average price.

Next, we discuss two kinds of relationships between products, within-category substitution and cross-category copurchase, and link the number of unique individual-specific time-invariant product utilities with the number of unique copurchase patterns.

#### 5.1.2.1 Within-category substitution

If two products $j$ and $l$ in the same category (i.e., $c_j = c_l$) share the same attribute combination (i.e., $\overrightarrow{A_j} = \overrightarrow{A_l}$, where $\overrightarrow{A_j} = \left(X_{j1}, \ldots, X_{jK}, Q_j, \bar{P}_j\right)^T$), only the cross-time price variation is different (i.e., $\Delta P_{jt} \neq \Delta P_{lt}$), then they will have the same individual-specific time-invariant product utility (i.e., $\gamma_{ij} = \gamma_{il}$, where $\gamma_{ij} = \overrightarrow{\theta_i}^T \overrightarrow{A_j}$). We can show that, under certain circumstances, their overall (cross-time) purchase probabilities are identical (i.e., $\mathrm{Pr}_{ij} = \mathrm{Pr}_{il}$, where $\mathrm{Pr}_{ij}$ denotes the average probability of consumer $i$ purchasing product $j$ over time). Formally,

**Lemma 1.** *If two products in the same category share the same attribute combination, and their price variations follow the same normal distribution, then they will have the same purchase prob-*

*ability when the number of time periods goes to infinity.*

$$\begin{gathered} \Pr_{ij} = \lim_{t\to\infty}\sum_t \Pr_{ijt} = \lim_{t\to\infty}\sum_t \Pr_{ilt} = \Pr_{il} \\ when \;\; c_j = c_l, \\ \overrightarrow{A_j} = \overrightarrow{A_l}, \\ \Delta P_{jt}, \Delta P_{lt} \sim N(0, \sigma_j^2) \end{gathered} \tag{12}$$

Lemma 1 implies that within one category, the number of unique purchase probabilities is equal to the number of unique individual-specific time-invariant product utilities $\|\gamma\|$. We refer the reader to Appendix B.3 for the formal proof.

#### 5.1.2.2 Cross-category copurchase/complement

From the objective function of the product fixed effects model (see Appendix B.4), the number of unique copurchase patterns for all segments (i.e., the dimensionality of the objective function) is equal to the number of segments $S$ multiplied by the squared value of the number of unique attribute combinations (i.e., $S * \|A\|^2$).

Consider three products, $j$, $l$, and $r$, where $j$ and $l$ are in the same category and $r$ is in a different category (i.e., $c_j = c_l \neq c_r$), and they share the same product attribute combination (i.e., $\overrightarrow{A_j} = \overrightarrow{A_l} = \overrightarrow{A_r}$). Suppose consumer $i$ in segment $s$ buys product $r$ together with either $j$ or $l$. Because both $j$ (or $l$) and $r$ are the chosen products in their corresponding categories, their attribute vectors $\overrightarrow{A_j}$ (or $\overrightarrow{A_l}$) and $\overrightarrow{A_r}$ have the highest inner product with the attribute preference vector $\overrightarrow{\theta_s}$ for segment $s$. So, the four vectors $\overrightarrow{A_j}, \overrightarrow{A_l}, \overrightarrow{A_r}, \overrightarrow{\theta_s}$ are highly similar, and the corresponding individual-specific time-invariant product utilities $\gamma_{ij}, \gamma_{il}, \gamma_{ir}$ are also similar. Thus, we can reduce the cardinality of the objective function $S * \|A\|^2$ to $S * \|A\|$, which is also the cardinality of individual-specific time-invariant product utility $\|\gamma\|$, and have the following result:

**Proposition 2.** *The number of unique individual-specific time-invariant product utilities* $\|\gamma\|$ *is equal to the number of unique copurchase patterns for all segments, that is, the number of segments*

*S multiplied by the number of unique copurchase patterns for each segment H.*

$$\|\gamma\| = S * H \tag{13}$$

#### 5.1.2.3 Price coefficient

In Equation 11, we assume that the price coefficient $\beta_i$ is applied to the cross-time price variation $\Delta P_{jt}$ instead of the price $P_{jt}$, and the average price $\bar{P}_j$ is incorporated as part of the individual-specific time-invariant product utility $\gamma_{ij}$. As a consequence, one dimension of the time-invariant product attributes $\overrightarrow{A_j}$ captures the average price $\bar{P}_j$. We show that when the price coefficients for $\bar{P}_j$ and $\Delta P_{jt}$ are estimated separately, they will be equal. Formally,

**Proposition 3.** *Given the data generating process* $u_{ijt} = \sum_{k=1}^{K} \alpha_{ik} X_{jk} + \delta_i Q_j + \beta_i \left(\bar{P}_j + \Delta P_{jt}\right) + \varepsilon_{ijt}$, *when the price coefficients for* $\bar{P}_j$ *and* $\Delta P_{jt}$ *are estimated separately in* $u_{ijt} = \sum_{k=1}^{K} \alpha_{ik} X_{jk} + \delta_i Q_j + \beta_{i1} \bar{P}_j + \beta_{i2} \Delta P_{jt}$, *they will both converge to the true price coefficient.*

$$\hat{\beta}_1, \hat{\beta}_2 \rightarrow \beta \text{ as } N \rightarrow \infty \tag{14}$$

We refer the reader to Appendix B.5 for a formal proof of Proposition 3. Thus, it is sufficient to estimate only the price coefficient for $\Delta P_{jt}$ because it converges to the price coefficient for $P_{jt}$. Therefore, there is no need to debiase or revise the product embeddings to get the unbiased estimate of the price coefficient.

### 5.1.3 Number of unique copurchase patterns in the Product2Vec model

If two products $j$ and $l$ in the same category share the same context products and the same negative samples, then they are copurchased with the same products and have the same copurchase pattern within each consumer segment. From the objective function of the Product2Vec model (see Appendix B.6), their embeddings are also the same and belong to the same cluster. However, in the shopping baskets data, only product pairs that are copurchased frequently (in other words, salient copurchase patterns) can be captured by the Product2Vec model. Using the logic we have discussed in Section 5.1.1.3 and Appendix B.2, each segment $s$ has one preferred attribute com-

bination $\overrightarrow{A_j}$, such that the inner product of segment $s$'s preference vector and product $j$'s attribute vector ($\overrightarrow{\theta_s}^T \overrightarrow{A_j}$) is higher than that of other products in the same category.

Therefore, only one copurchase pattern for each segment can be reflected in the product embeddiings, and the number of product embedding clusters $M$ will be equal to the number of consumer distinguishable attribute combinations $\Lambda$. Because negative sampling has a random component, the products with the same copurchase pattern might have slightly different embeddings. However, the randomness is mitigated by clustering the embeddings. Thus, we have the following result:

**Proposition 4.** *The number of product clusters generated by product embeddings $M$ is equal to the number of consumer distinguishable attribute combinations $\Lambda$.*

$$M = \Lambda \tag{15}$$

Note that when we consider cross-category copurchase, some product clusters may not exist in all categories, and some categories may have fewer clusters than other categories. Thus, the number of unique copurchase patterns is equal to the *maximum* number of product clusters among all categories. Without loss of generality, we assume that all categories have the same number of product clusters; that is, $M_c = M, \forall c$.

Combining Proposition 2 and Proposition 4, we have

**Proposition 5.** *The number of unique individual-specific time-invariant product utilities $\|\gamma\|$ is equal to the number of consumer segments S multiplied by the number of product clusters generated by product embeddings M. Thus, product clusters generated by product embeddings could replace product fixed effects.*

$$\|\gamma\| = S * M \tag{16}$$

We can show that the findings above still hold in the presence of price endogeneity, and a formal proof is available in Appendix B.7. Also, a full list of notations used in the proof is provided in Appendix B.8.

The above proof demonstrates that the product clusters from embeddings obtained with Product2Vec can replace the time-invariant product fixed effects in choice models. Both product embed-

dings and product fixed effects can capture the information of consumers' preferences for product attributes, and their dimensionalities are equal to the number of unique copurchase patterns.

## 5.2 Measuring complementarity and exchangeability among products

In this section, we transform the product embeddings into interpretable economic measures, namely, *complementarity* and *exchangeability*, to identify complements and substitutes. These metrics are inspired by those proposed in Ruiz et al. (2019).

It is worth noting that these measures are not based on the classical marketing definition that uses cross-price elasticities, but they instead use the latent representations of products. Roughly speaking, two products have high complementarity if the conditional probability of buying one given the other is high, and two products have high exchangeability if they predict similar purchase patterns for the rest of the products in the store. We provide the formal definitions of complementarity and exchangeability next, and demonstrate their ability to provide the same insights that could have been learned using cross-price elasticity in Section 7.

**Complementarity** We consider two products A and B to have high complementarity if the conditional probability of purchasing A (or B) given B (or A) being already in the basket is high. Put simply, two products have high complementarity if they are very likely to be purchased together.

Note that by using the conditional purchase probability of one product given another rather than the joint purchase probability of two products, complementarity is not affected by the base purchase frequency or popularity of one product.

Our definition of *complementarity* is based on *copurchase*, rather than negative cross-price elasticities in the economic sense. This means that, if diapers and beer tend to be purchased in the same baskets by young parents, we expect these two products to have high complementarity. Copurchase may also be driven by special occasions, such as birthday cake, candles, and balloons, and we expect these products to have high complementarity. Copurchase could also occur because products happen to have the same purchase cycle, e.g., people buy both egg and milk once per

week, so egg and milk would have high complementarity in our case. However, we expect these cases to be relatively infrequent compared to true complementary cases. And our empirical data also confirms it. We think this definition of complementarity based on copurchase is valuable for retailers because they can discover products that can be placed adjacently, despite the fact that retailers should be cautioned to not design co-promotion strategies based on it.

Formally, we compute the complementarity between products A and B as

$$\begin{aligned} C_{AB} &= \frac{1}{2} \cdot (P(A|B) + P(B|A)) \\ &\approx \frac{1}{2} \cdot \left(\sigma(v_B^T \cdot v'_A) + \sigma(v_A^T \cdot v'_B)\right), \end{aligned} \tag{17}$$

where v and v' are the input and output vectors, as shown in Equation 3. Note that, consistent with the negative sampling technique, Equation 17 approximates the conditional purchase probabilities $P(i|j)$ with the sigmoid function $\sigma(v_j^T \cdot v'_i)$.

**Exchangeability** We consider two products A and B to be exchangeable if the conditional probabilities of buying other products in the store are similar. In other words, A and B are exchangeable if they interact similarly with other products. Formally, we measure the exchangeability between products A and B as

$$E_{AB} = -||p(\cdot|A) - p(\cdot|B)||, \tag{18}$$

where $p(\cdot|A)$ and $p(\cdot|B)$ are the vectors of the conditional purchase probability of all the other products in the store given $A$ or $B$ being purchased. Based on Equation 2, we can approximate $p(k|A)$ and $p(k|B)$ with the corresponding product vectors as $\sigma(v_{j=1}^T \cdot v'_A)$ and $\sigma(v_{j=1}^T \cdot v'_A)$, respectively.

Based on this definition, we expect products that are likely to be substitutes to have a high exchangeability score. This is because two products that are substitutes (e.g., two different brands of chips) will have very similar probabilities of being co-purchased with other similar products in the store (e.g., both are likely to be copurchased with ketchup and carbonated beverages, and unlikely to be copurchased with other brands of chips). However, as discussed in Ruiz et al. (2019), two products that are likely to be complements might also have a high exchangeability score. This is because two products that are frequently purchased together tend to appear in similar baskets and

therefore tend to have similar interactions with other products. To exclude possible complements, we define the *penalized exchangeability* as

$$PE_{AB} = E_{AB} - \lambda C_{AB}, \tag{19}$$

where the parameter $\lambda$ is chosen to minimize the correlation between $PE_{AB}$ and $C_{AB}$.

# 6 Causal inference: Incorporating product vectors into the choice model

This section discusses how to use product embeddings for causal inference, more specifically, estimating unbiased price coefficients in choice models.

As hinted in Section 5.1, discrete choice models typically assume that consumers choose the product in the choice set that maximizes their utility. Formally, consider a set of products $j \in \{1,...,J\}$ characterized by a set of attributes $\{X_{jk}\}_{k=1}^{K}$ that are fixed over time, and price $P_{jt}$, which can vary over time. For each consumer $i$, the utility of choosing alternative $j$ in period $t$ is

$$u_{ijt} = \sum_{k} \alpha_{ik} X_{jk} + \beta_i P_{jt} + \varepsilon_{ijt} \tag{20}$$

Equation 20 specifies a choice model that uses product attributes to characterize a large set of products in a frugal manner, which we label as the "Attribute Model." Despite using fewer parameters than the product fixed effect model, Fader and Hardie (1996) show that it can achieve good results if the attributes chosen are able to characterize consumer choices. [2]

In many cases, the set of time-invariant product characteristics $\{X_{jk}\}_{k=1}^{K}$ in Equation 20 is not fully observed by the analyst (Armona et al. 2021, Gabel and Timoshenko 2021). We deliberately omit one or more attributes in Equation 20 and label it as the "Missing attribute Model." We include this model to examine to what extent other choice models could outperform this model.

[2]We do not compare with Chintagunta and Dube (2005) because they use market share data to control for price endogeneity and require information about total market size for each category, which is not available in our data. Nonetheless, using the mixed logit model with the price instrument, we can control for consumer heterogeneity and price endogeneity.

To overcome this limitation, one solution is to incorporate a consumer-product fixed effect $\alpha_{ij}$ to summarize the average intrinsic valuation for each product ($\sum_k \alpha_{ik} X_{jk}$).

$$u_{ijt} = \alpha_{ij} + \beta_i P_{jt} + \varepsilon_{ijt} \tag{21}$$

We label this model the "Product Fixed Effect Model." It is the best possible model because it includes a (heterogeneous) parameter for every product. We also call this model the "target" model. However, this approach does not scale well when the number of products is large. A better approach would be to identify the attributes that drive consumer choices (observed or unobserved) and use these attributes to estimate the choice model. This is exactly what we propose to do with product vectors.

Section 5.1 demonstrates that product embedding clusters can replace product fixed effects in choice models. Therefore, we create the model specification below, as the "Embedding Cluster Model."

$$u_{ijt} = \alpha_{im} + \beta_i P_{jt} + \varepsilon_{ijt} \tag{22}$$

In Equation 22, $\alpha_{im}$ is the fixed effect for consumer $i$ and cluster $m$ ($m = 1, ..., M$) which product $j$ is a member of. The cluster membership relationship is established by conducting clustering on the product embeddings. Because consumer choices are often based on a limited number of attributes, this approach guarantees that the number of clusters is relatively small compared to the number of products, thus reducing the computational burden associated with estimating choice models with many parameters. Moreover, because these embedding clusters can represent both observed and unobserved product attributes (because they can replace product fixed effects as shown in Section 5.1), including them in the model rather than simply relying on observable attributes leads to better results.

We also allow the model specification to accommodate price endogeneity, as price might be determined by some unobserved (to the analyst) demand-shocks ($\xi_{jt}$). To address such endogeneity, we adopt the control function approach (Petrin and Train 2010)[3] and use the classical BLP

[3]We apply the same endogeneity solution to the attribute model, missing attribute model, and product fixed effect model, too.

instrument (Berry et al. 1995), that is, the average price for the same product and week in other stores of the same chain. So, the full specification for the "Embedding Cluster Model" is

$$U_{ijt} = \alpha_{im} + \beta_i P_{jt} + \lambda \xi_{jt} + \varepsilon_{ijt} \tag{23}$$

# 7 Simulation

In this section, we conduct a simulation exercise used to empirically apply the models proposed in Section 5 and Section 6. In what follows, we first describe the simulation setup in Section 7.1, and then we demonstrate two ways of interpreting product embeddings. Specifically, Section 7.2 shows that embeddings can reflect product attribute combinations, and Section 7.3 shows that embeddings can identify complements and substitutes. Finally, section 7.4 shows that the embedding cluster model can recover unbiased price coefficients.

## 7.1 Setup

We start by simulating consumer choices using the model described in Section 5.1.1, but allowing for price endogeneity. To do so, we use the same category utility specification in Equation 4, and modify the product utility specification in Equation 6 to be as follows:

$$u_{ijt} = \sum_k \alpha_{ik} X_{jk} + \delta_i Q_j + \beta_i P_{jt} + \lambda \xi_{jt} + \varepsilon_{ijt}, \quad \varepsilon_{jt} \sim N(0, \sigma), \tag{24}$$

where $\varepsilon_{jt}$ is a product-specific time varying component that correlates with price. Specifically, we assume that

$$P_{jt} = \delta \text{Cost}_{jt} + \xi_{jt}, \tag{25}$$

where $\text{Cost}_{jt}$ is an exogenous cost component.

We use a simple setting where there are two attributes ($k = 1, 2$), and both attributes, as well as quality $Q_j$ and average price $\bar{P}_j$, consist of two levels (low vs. high). To incorporate consumer heterogeneity, we consider discrete heterogeneity in preferences for product attributes. Specifically, we consider 8 latent segments using the preference parameters described in Table 1. Note that each

segment either likes high levels ($\alpha_{sk}$ = 6) or low levels ($\alpha_{sk}$ = -6) of the two binary attributes $k$, and it is either price oriented ($\delta_s$ = 3 and $\beta_s$ = -6) or quality oriented ($\delta_s$ = 6 and $\beta_s$ = -3). For instance, segment 1 consumers prefer the first attribute ($\alpha_{11}$ = 6), do not prefer the second attribute ($\alpha_{12}$ = -6), and are price oriented ($\delta_1$ = 3 and $\beta_1$ = -6). The specification of discrete heterogeneity allows all the simulated products to be desirable for at least one segment of the population.

[insert Table 1 around here]

We simulate 10,000 consumers who make decisions over a period of 50 weeks. We consider a total of 10 categories with 100 products each, for a total of 1,000 products.[4]

To train the Product2Vec model and obtain the product embeddings, we use the following hyperparameters: dimension = 30, iterations = 50, window size = 10 (which equals the number of categories, assuming a consumer can buy at most one item in each category).

In what follows, we present several analyses based on the resulting product embeddings.

## 7.2 Product embeddings capture attributes

First, we explore whether product embeddings can recover meaningful attributes using visualization and clustering analyses.

**Product map** We first explore the resulting product embeddings of all products and categories visually. To do so, we create product maps that reduce the original dimensionality of the embeddings using Principal Component Analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE). We illustrate the resulting product maps in Figure 1. Using the first two PCA components, the product map displays 8 product clusters, each one of them corresponding to one possible combination of product attributes. Thus, each product cluster corresponds to exactly one segment that has a strong preference for a specific type of products.

[insert Figure 1 around here]

[4] In Appendix C.1, we replicate this exercise with larger datasets—using 300 and 500 products per category (for a total of 3,000 and 5,000 products, respectively)—and obtain similar results

We obtain similar results using t-SNE. In this case, the product map shows 10 product clusters, most of them corresponding to one product category (this is consistent with Gabel et al. 2019). Moreover, within each category cluster, it is possible to identify sub-clusters of products that correspond to different possible combinations of product attributes. Therefore, using PCA we are able to recover attributes, whereas with t-SNE we can recover both categories, and attributes within categories.

**Clustering analysis** We then test whether what we observe visually can be recovered automatically by implementing a clustering analysis. We use the K-means algorithm based on euclidean distance and determine the optimal number of clusters in the product map using the CH index (Caliński and Harabasz 1974) and the Silhouette score (Rousseeuw 1987).[5] Recall that, given our setup, the optimal number of clusters should be 8, and each cluster should group products with one possible attribute combination. Based on both metrics (CH index and Silhouette score), we find it is indeed the case, and each one of the eight clusters groups products with the same attribute combination.

Overall, the results presented in this section suggest that products with similar characteristics are bought together and therefore are close to each other in the embedding space. This means that product embeddings can capture attributes that drive product copurchases.

## 7.3 Exchangeability and complementarity

Next, we explore whether our proposed complementarity and penalized exchangeability metrics (defined in Section 5.2) can successfully capture product complementarity and substitution, respectively. To do so, we compare our penalized exchangeability score with the true cross-price elasticities in the same category,[6] and compare our complementarity score with the true copur-

[5] Using the Hierarchical Agglomerate algorithm and cosine similarity as a distance metric gives similar results.

[6] We obtain the average of the segment-specific cross-price elasticities between product $i$ and $j$, computed as $-(\bar{P}_j/s_j)\cdot(\beta_s \cdot s_i \cdot s_j)$, where $s_k$ is the choice probability of product $k$ and $\beta_s$ is the segment-specific price coefficient. The cross-price elasticities for two products in different categories are set to zero.

chase probabilities for products in different categories.[7] We also consider other embedding-based metrics previously suggested in the literature, namely similarity of the product embeddings and co-occurrence scores (Gabel et al. 2019), as benchmarks.

To explore the validity of the two metrics, we first examine their ability to capture expected patterns. Complementarity should have high values for products with similar attributes but in different categories (i.e., complementary categories), and low values for products in the same category. In Figure 2, we show that complementarity can successfully recover such a pattern.

[insert Figure 2 around here]

Similarly, penalized exchangeability should have high values for products with similar attributes in the same category, and low values for products in different categories. In Figure 3, we show that the penalized exchangeability metric can successfully recover such a pattern.

[insert Figure 3 around here]

To further validate our metrics and their advantage over other embedding-based metrics, we measure their correlations with the true copurchase probabilities and cross-price elasticities. We report these results in Table 2. We observe that complementarity has the highest correlation with the true pairwise copurchase probabilities. Similarly, penalized exchangeability has the highest correlation with the true pairwise cross-price elasticities. In addition, we observe that exchangeability captures complementarity, emphasizing the need to penalize this score to effectively distinguish substitutes from complements. Finally, these results illustrate that our metrics are more accurate than other embedding-based metrics, namely similarity and co-occurrence scores.

[insert Table 2 around here]

Overall, the results presented in this section suggest that complementarity and exchangeability can be valuable alternatives to study product competition. This is especially true in cases where the number of products is very large and estimating choice models to measure cross-price elasticities becomes computationally challenging.

[7] To be consistent with our data generating process, we obtain the co-purchase probability of product $i$ in category A and product $j$ in category B as $P(y_{iat} = i) \cdot P(y_{ibt} = j) \cdot P(z_{iat} > 0 \text{ and } z_{ibt} > 0)$. The copurchase probabilities for two products in the same category are set to zero.

## 7.4 Choice model

In this section, we show how we can leverage these embeddings in choice models to obtain more accurate estimates of price elasticities. The idea is that, because product clusters capture meaningful (and potentially unobservable) product attributes that drive consumer choices, incorporating them into the choice model specification can help to reduce omitted variable bias. In addition, because the number of product clusters is generally smaller than the number of products in the choice sets, using product clusters could be a more scalable solution than using product fixed effects.

To formally explore the benefits of incorporating product clusters, we estimate a choice model using one product category and a subsample of 1,000 customers. Consistent with the data generating process, we estimate a latent class multinomial logit model, and estimate different specifications that vary in the type of product attributes that are observed by the researcher.

We report these results in Table 3. All the results are obtained using the control function approach and the simulated costs as instruments. To simplify comparisons across models, we report only the estimates of the price coefficient, along with model fit, hit rates, and running time.

In column 1 of Table 3, we consider the case where the researcher observes all product attributes. As expected, this "true model" recovers the true parameters used to generate the data.

[insert Table 3 around here]

Columns 2 and 3 of Table 3 allow us to quantify the impact of unobserved product attributes. In column 2, we consider the case where the researcher does not observe the quality attribute Q. Although it is still possible to identify eight customer segments (from the 2x2x2 variation in the two binary attributes and the two price levels), the estimated price coefficients are substantially biased due to the positive correlation between the missing quality attribute and price. In column 3, we consider the case where the researcher does not observe one of the binary attributes, say $X_1$. In this case, it is only possible to identify four customer segments (from the 2x2 variation in the observed binary attribute and the two price levels), and the estimated price coefficients are also biased. As expected, both specifications have a worse model fit and hit rates than the "true model."

Columns 4 and 5 of Table 3 incorporate product embedding clusters (our proposed approach) and product fixed effects, respectively. Both estimates are closer to the true parameters and provide similar results (estimated price coefficient, model fit, and hit rates) to those obtained by the "true model." Thus, both approaches allow us to mitigate omitted variable bias. Nevertheless, the number of estimated parameters is much smaller when using product embedding clusters than when using product fixed effects, translating into a substantially lower running time (17 min vs. 19 hrs).

## 7.5 Robustness checks

The results presented so far are based on a single set of product embeddings, estimated with the word2vec skip-gram model Mikolov et al. (2013b) and a specific set of hyperparameters. There are two concerns with this approach. First, the performance of the word2vec algorithm can vary considerably with the choice of hyperparameters (Caselles-Dupré et al. 2018). Second, the estimation procedure contains random elements (negative sample and downsampling of frequent products) that can significantly impact the resulting embeddings. In other words, the embedding representation of the same product might change considerably across different rounds of estimation using the same dataset, even after holding the hyperparameters constant. In this section, we explore whether and how each of those concerns impacts our findings.[8]

**Performance with different hyperparamaters** To explore whether and how the randomness in the word2vec training process can impact our main findings, we sequentially vary the value of the following hyperparameters: (1) the dimension of the product embeddings (30, 50, and 100); (2) the number of iterations for training (50, 100, and 200); (3) the number of negative samples (1, 5, and 10); (4) the exponent used to shape the negative sampling distribution (0, 0.75, and 1); and (5) the threshold for configuring which higher-frequency products are randomly downsampled (0, 1e-3, and 1e-5).

[8] We thank an anonymous reviewer for raising this point.

We monitor four different outcomes: (1) the loss function; (2) the embeddings' ability to recover category-level relationships given by the $\Omega$ matrix in Equation 4; (3) the relationships between products in the embedding space, measured as the pairwise cosine similarity and Euclidean distance; and (4) the ability to recover the optimal number of product clusters. We report these results in Appendix C.2. We find that all outcomes are relatively stable with respect to changes in hyperparameters. The only parameter that seems to affect outcomes is the downsampling threshold, which we recommend keeping at its default value of 0.001.

**Randomness in the training process** To explore whether the randomness in the word2vec training process can impact our main findings, we obtain 10 different sets of product embeddings using 10 different random number generator seeds, and keeping constant the hyperparameters of the model.

In Table 4, we show that, as expected, the same product can have considerably different numerical representations. Nevertheless, the pairwise cosine similarities and vector distances are highly correlated across different seeds (see Table 5 and Table 6), suggesting that despite differences in individual product representations, the relationships between products are stable across seeds.

[insert Table 4 around here]

[insert Table 5 around here]

[insert Table 6 around here]

Finally, in Figures 4 and 5, we show that the number of optimal clusters is always 8 for all the different sets of product embeddings, and in each case, clusters group products with the same attributes. This means that product clusters are resilient to different seeds, and consequently, the random nature of the word2vec training process does not affect the estimates of the choice model.

[insert Figure 4 around here]

[insert Figure 5 around here]

# 8 Real world data

We next show that our approach generates good results when applied to a dataset of real consumer choices obtained from NielsenIQ.

## 8.1 Data

We test our model using consumer panel data and retail scanner data from NielsenIQ.[9] The consumer panel data contains 9,045,132 shopping baskets purchased by 61,381 households in 2018. These shopping baskets contain 718,063 products of 118 categories and come from 761 retailers and 49 states. The retail scanner data contains information about (1) marketing mix variables: price, feature, and display; as well as (2) product attributes: brand, flavor, type, formula, container, and size.

The consumer panel data in the grocery shopping context is suitable for training the Product2Vec model because most baskets contain a large number of products, which provide rich information about copurchase patterns. The distribution of basket size (i.e., the number of unique products in each basket) is shown in Figure 6. Because our model makes use of product co-occurrences to learn product vectors, it is important that the majority of baskets contain more than one product. In our data, 66% of baskets satisfy this condition.

[insert Figure 6 around here]

Figure 7 plots the histogram of category size (i.e., the number of unique products in each category). Among all 118 categories, 107 contain more than 500 unique products. Therefore, a category-level competitive analysis using choice models with product fixed effects is computationally expensive. We show that by leveraging clusters obtained from product embeddings, we can

[9] Researchers' own analyses calculated (or derived) based in part on data from Nielsen Consumer LLC and marketing databases provided through the NielsenIQ Datasets at the Kilts Center for Marketing Data Center at The University of Chicago Booth School of Business. The conclusions drawn from the NielsenIQ data are those of the researchers and do not reflect the views of NielsenIQ. NielsenIQ is not responsible for, had no role in, and was not involved in analyzing and preparing the results reported herein.

obtain results that are comparable to using product fixed effects, but with a substantial reduction in the number of parameters to estimate.

[insert Figure 7 around here]

We divide this data into three sets: training, estimation, and test.[10] We use the training set to train the Product2Vec model and derive the product vectors. We then use the estimation set to estimate the coefficients of the choice model. Finally, we use the test set to evaluate the out-of-sample accuracy of the demand predictions with parameters obtained from the choice model.

We randomly split all households in the consumer panel data into two subsamples and use the shopping baskets from 40% households as the training set. We drop the products with fewer than 160 occurrences [11] since their embeddings would be less accurate if they appeared in only a few shopping baskets. We also drop the products with undefined categories.

For the 60% of households left, we take their last shopping baskets as the test set, and the remaining (i.e., from first to second to last) shopping baskets as the estimation set. In other words, we use earlier purchases to estimate the choice model and predict later purchases for the same households. We further reduce the size of the estimation and test set in two ways. First, to avoid the influence of different product assortments across different retailers and states, for example, different retailers may have their own private label products, we use shopping baskets from only one retailer in one state [12], where the highest number of shopping baskets are observed. Second, since each category has its category-specific product attributes, we focus on a single category when applying the choice model to the estimation set and the test set. In this paper, we use the carbonated beverage category; we explain the reason for selecting this category in Appendix D. Summary statistics are reported in Table 7.

[insert Table 7 around here]

[10] Such division is important for two reasons. First, we need to use two different sets of data for training and estimation to make sure the model fit is credible. Otherwise, choice models estimated using product vectors that were trained on the same dataset will fit the data well simply because they use of the same information. Second, we need the test set to calculate the out-of-sample accuracy, which is the most important measure of model fit.

[11] We choose the threshold $n = 160$ such that 80% shopping baskets are kept after dropping the niche products.

[12] We do not limit data to one retailer in one state when training word2vec, because we need a sufficient number of shopping baskets to derive meaningful copurchase patterns and get product embeddings.

To estimate choice models and predict consumer purchases, we need to control for marketing mix variables and product attributes. These variables are described in Table 8. Products with a larger size tend to have a higher price, so we use unit price (i.e., price divided by size) to obtain comparable prices. Since the distribution of unit price is right skewed, we apply the log transformation. As we already show in Section 5.1, clusters generated from embeddings reflect the average prices of products. So we further subtract the average (log unit) price and use the demeaned price, named as the "delta log unit price" in the cluster model.

[insert Table 8 around here]

## 8.2 Product embeddings capture attributes

As we did with the simulated data, we plot the product map for the carbonated beverages category in Figure 9. We apply hierarchical clustering to 128-dimensional product embeddings and use the Calinski-Harabasz Index to automatically determine the optimal number of clusters, which is 9 in our case. [13] We then apply t-SNE to these product embeddings and derive a 2-dimensional product map.

In Figure 9, each point denotes a product, the size of the point represents the average price of the corresponding product, and the color of the point represents the cluster that the product belongs to, labeled with different cluster numbers. The two subplots come from two random seeds, and they demonstrate consistent cluster memberships, despite slightly different locations of individual points (possibly due to the randomness in the embedding training process).

[insert Figure 9 around here]

We can interpret the clusters by looking at the products in the same cluster. Table 9 lists the number of products, some exemplar products, and the common attribute(s) shared by most products in each cluster. For instance, cluster 3 contains 20 products, such as "Pepsi Max Diet Cola 6 Pack" and "Mountain Dew Diet Lemon/Lime/Citrus 6 Pack," and products in this cluster are mostly soft drinks that come in six-pack bottles.

[13] We use hierarchical clustering instead of k-means for real data because the Calinski-Harabasz Index for k-means always increases with the number of clusters, and therefore it is not clear what is the optimal number.

[insert Table 9 around here]

## 8.3 Exchangeability and complementarity

As we also did with the simulated data, we explore product substitution and complementarity across multiple categories using our proposed product embeddings-based metrics, namely, penalized exchangeability and complementary scores (defined in Section ). In Table 10, we report the results for six focal products (Focal UPC) from categories where we intuitively expect some strong substitution and complementarity relationships (e.g., breakfast and related products). For each focal product, we list the top 5 products with the highest penalized exchangeability and the top 5 products with the highest complementarity scores. For comparison purposes, we also include the cosine similarity between the embeddings for each pair of product. The product descriptions are our interpretation of the original UPC descriptions in the Nielsen data, as illustrated in Table 16 of Appendix E.

The results in Table 10 suggest that our proposed metrics can capture intuitive substitution and complementarity relationships among a large number of products. For instance, for the first focal product, a private label bread, the top 5 products with the highest exchangeability scores (i.e., most likely substitutes) are other types of private label bread, and the top 5 products with the highest complementarity scores (i.e., most likely complements) include products such as sliced cheese and meats. Similarly, for the second focal product, Quaker Oats, the top 5 products with the highest complementarity scores (i.e., most likely substitutes) include intuitive pairs such as yogurt and milk.

[insert Table 10 around here]

## 8.4 Choice model

Similar to the simulation section, we compare choice models estimated using different specifications described in Section 6. In the attribute model, we use five product attributes: brand fixed effects, flavor fixed effects, type fixed effects, whether the product is a regular or diet drink, and

whether the container is bottle or can. In the missing attribute model, we include only price-related variables. It is the simplest model we could ever use to estimate price elasticities and predict consumer purchases, and we would like to see how this basic model performs.

As discussed in Section 8.1, we estimate the mixed logit choice models with a dataset of 180 carbonated beverage products. As shown in Section 8.2, these products are classified into 9 clusters. We report the estimates in Tables 11.[14] Columns 1 to 4 report the estimates of the embedding cluster model, the product fixed effect model (target model), the attribute model, and the missing attribute model, respectively.

[insert Table 11 around here]

We observe that the price coefficient in the choice model with cluster dummies is closer to the target model than those from the attribute model and the missing attribute model. This is because the vectors capture product characteristics that are not observable in the data. Although Fader and Hardie (1996) control for as many observable product characteristics as possible, our results suggest that similar or better price coefficient estimates can be achieved by controlling for latent product dimensions (i.e., product clusters). This finding makes our approaches particularly appealing in cases where product characteristics are not readily available or when the nature of the product category requires defining attributes that are not easily measurable (e.g., healthiness of drinks or aesthetics and style of clothing, etc.).

Turning to model fit, we observe that our model using embedding clusters obtains a better fit in terms of log likelihood, AIC, and BIC than the missing attribute model. We also compare both the in-sample and out-of-sample hit rates. Compared with the missing attribute model, the cluster model (with the complementarity and exchangeability measures) achieves higher in-sample and out-of-sample hit rates. With a 25.4% out-of-sample hit rate, it outperforms the missing attribute model (24.1%) by 5.4%.

Finally, the running time of the target model is about 4 hours and 50 minutes, whereas the cluster model take 70% less time, or about 20 minutes. We can see that the running time is directly

[14] In Appendix F, we report the first-stage results. The F-statistics of the first stage are very large, which suggests that the instrument satisfies the relevance condition.

related to the number of parameters in each model, and, compared with the attribute model, our approach produces a less biased price coefficient estimate and a shorter running time.

Overall, the results obtained with the NielsenIQ data are similar to those we obtained with simulated data in Section 7.

# 9 Conclusions

This paper proposed Product2Vec, an approach that leverages representation learning algorithms that can learn product attributes in an unsupervised way.

We start by theoretically demonstrating that clusters of product embeddings generated using consumer purchase as input map to product attributes that drive these purchases. Then, we show that product embedding can be used to create measures of complementarity and substitution that do not require the estimation of a demand model. Finally, we empirically demonstrate—with simulated and real consumer purchase data—that combining product vectors with choice models we can obtain precise estimates of price elasticities while limiting the number of model parameters.

Overall, our approach can help firms learn about important attributes that drive consumer choices at scale, and learn about the relationships among products without the need to estimate a demand model. In addition, when it comes to estimating a demand model, our approach perform as well as a model that includes an intercept for every product in the choice set, but with considerably fewer parameters.

This paper opens up several opportunities for future research. Our method uses the co-occurrences of products in shopping baskets to study competitive relationships at the product level. A natural extension is to apply the same representation learning technique to both product co-occurrences and product-related text, such as product descriptions and reviews, and combine the insights from both sources. Another direction for future research is to model consumers as vectors to capture consumer heterogeneity. Currently, we incorporate unobserved consumer heterogeneity using the random coefficient model but do not model each individual explicitly. Future work might also use

representation learning to uncover both product embeddings and user (consumer) embeddings at the same time, thus improving model accuracy and predictions.

In the last few years, we have seen a growing body of marketing research incorporating machine learning methods with the aim of creating models that scale to large datasets, estimating parameters more accurately and producing better predictions. Our work adds to this literature by focusing on consumer choices. We hope that our work will inspire other marketing researchers to explore this new and promising area of research.

# References


Al-Rfou R, Perozzi B, Skiena S (2013) Polyglot: Distributed word representations for multilingual NLP. arXiv preprint arXiv:1307.1662.

Armona L, Lewis G, Zervas G (2021) Learning product characteristics and consumer preferences from search data. *Available at SSRN 3858377* .

Barkan O, Koenigstein N (2016) Item2vec: neural item embedding for collaborative filtering. *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, 1–6 (IEEE).

Berry S, Levinsohn J, Pakes A (1995) Automobile prices in market equilibrium. *Econometrica: Journal of the Econometric Society* 841–890.

Caliński T, Harabasz J (1974) A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods* 3(1):1–27.

Caselles-Dupré H, Lesaint F, Royo-Letelier J (2018) Word2vec applied to recommendation: Hyperparameters matter. *Proceedings of the 12th ACM Conference on Recommender Systems*, 352–356.

Chandukala SR, Kim J, Otter T, Allenby GM (2008) *Choice models in marketing: Economic assumptions, challenges and trends* (Now Publishers Inc).

Chintagunta PK, Dube JP (2005) Estimating a stockkeeping-unit-level brand choice model that combines household panel data and store data. *Journal of Marketing Research* 42(3):368–379.

Fader PS, Hardie BG (1996) Modeling consumer choice among skus. *Journal of marketing Research* 33(4):442–452.

Gabel S, Guhl D, Klapper D (2019) P2v-map: Mapping market structures for large retail assortments. *Journal of Marketing Research* 56(4):557–580.

Gabel S, Timoshenko A (2021) Product choice with large assortments: A scalable deep-learning model. *Management Science* .

Grbovic M, Radosavljevic V, Djuric N, Bhamidipati N, Savla J, Bhagwan V, Sharp D (2015) E-commerce in your inbox: Product recommendations at scale. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1809–1818 (ACM).

Guadagni PM, Little JD (1983) A logit model of brand choice calibrated on scanner data. *Marketing science* 2(3):203–238.

Jacobs BJ, Donkers B, Fok D (2016) Model-based purchase predictions for large assortments. *Marketing Science* 35(3):389–404.

Kumar M, Eckles D, Aral S (2020) Scalable bundling via dense product embeddings. *arXiv preprint arXiv:2002.00100* .

Manchanda P, Ansari A, Gupta S (1999) The "shopping basket": A model for multicategory purchase incidence decisions. *Marketing science* 18(2):95–114.

Mikolov T, Chen K, Corrado G, Dean J (2013a) Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013b) Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 3111–3119.

Petrin A, Train K (2010) A control function approach to endogeneity in consumer choice models. *Journal of marketing research* 47(1):3–13.

Prasad A, Strijnev A, Zhang Q (2008) What can grocery basket data tell us about health consciousness? *International Journal of Research in Marketing* 25(4):301–309.

Rousseeuw PJ (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics* 20:53–65.

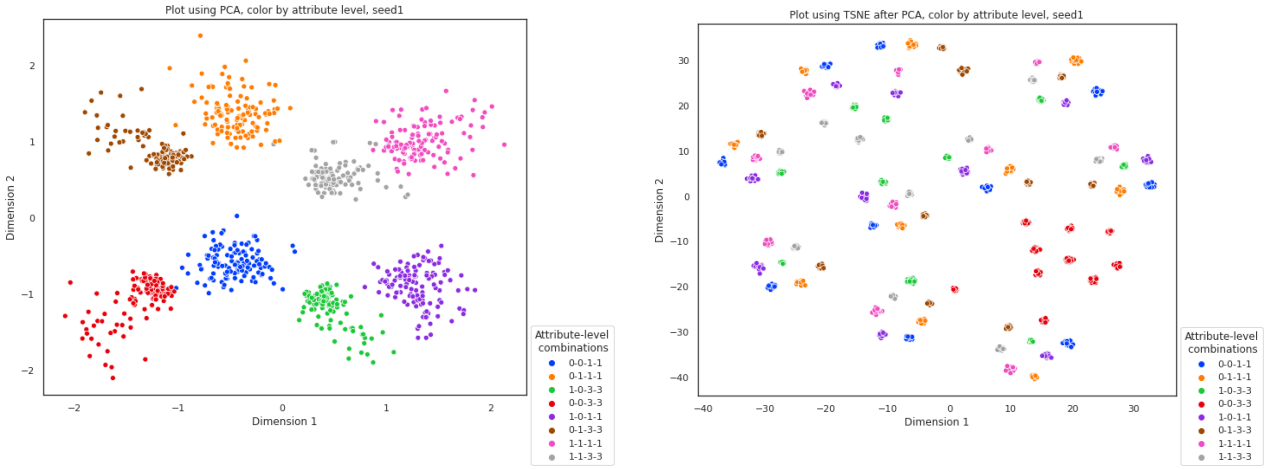
Ruiz FJ, Athey S, Blei DM (2019) SHOPPER: A probabilistic model of consumer choice with substitutes and complements. Annals of Applied Statistics.

Singh VP, Hansen KT, Gupta S (2005) Modeling preferences for common attributes in multicategory brand choice. *Journal of Marketing Research* 42(2):195–209.

Turian J, Ratinov L, Bengio Y (2010) Word representations: a simple and general method for semi-supervised learning. *Proceedings of the 48th annual meeting of the association for computational linguistics*, 384–394 (Association for Computational Linguistics).

Winer RS, Neslin SA (2014) *The history of marketing science*, volume 3 (World Scientific).

Yang Y, Zhang K, Kannan P (2021) Identifying market structure: A deep network representation learning of social engagement. *Journal of Marketing* 00222429211033585.




(a) PCA

(b) T-SNE

Figure 1: Product Map By Attribute Combinations

Note: Panel (a) shows the product map based on PCA mapping of the product embeddings, and Panel (b) shows the product map based on t-SNE mapping of the product embeddings. Colors represent the possible attribute-level combinations of the simulated product attributes.

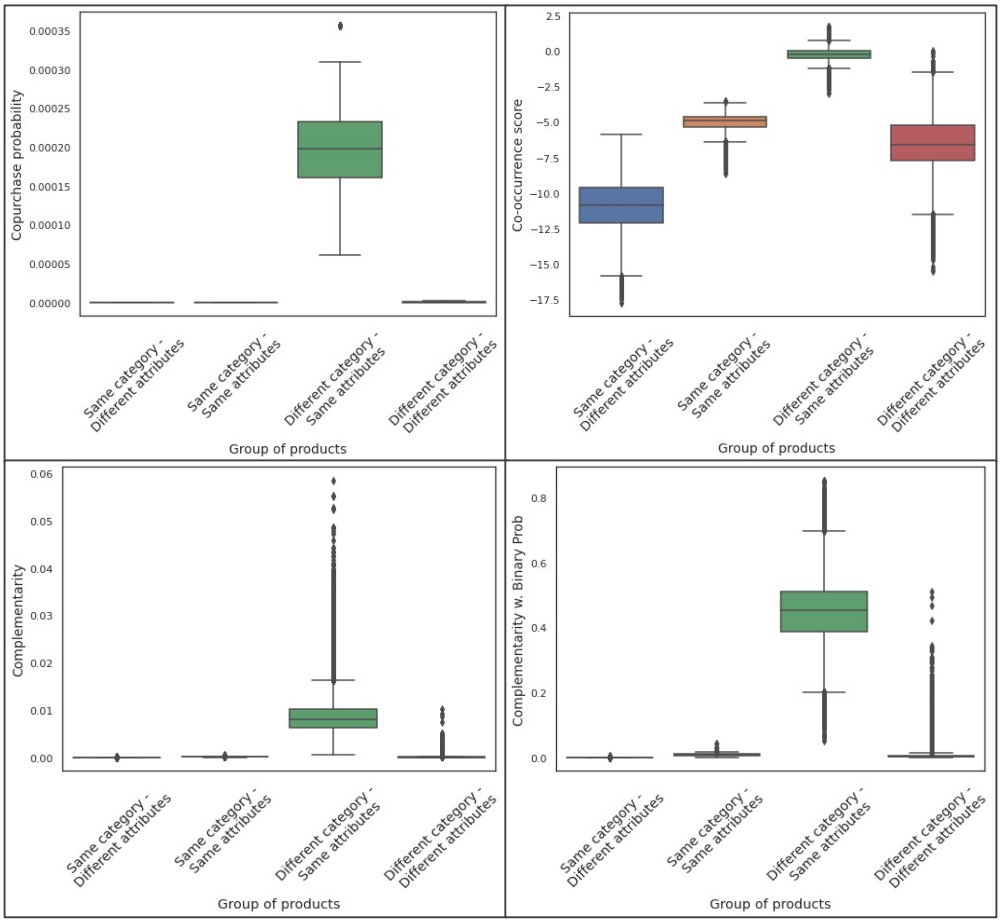



Figure 2: Copurchase, Cooccurence, and Complementarity by Product Groups

Note: The upper-left and upper-right panels illustrate the true pairwise co-purchase probabilities and the pairwise co-occurrence scores, respectively. The bottom-left and bottom-right panels illustrate the pairwise complementarity metrics with different approximations for the co-purchase probabilities. In the bottom-left panel, $P(A|B)$ and $P(B|A)$ are computed with the softmax function, while in the bottom-right panel, $P(A|B)$ and $P(B|A)$ are computed with the sigmoid function.

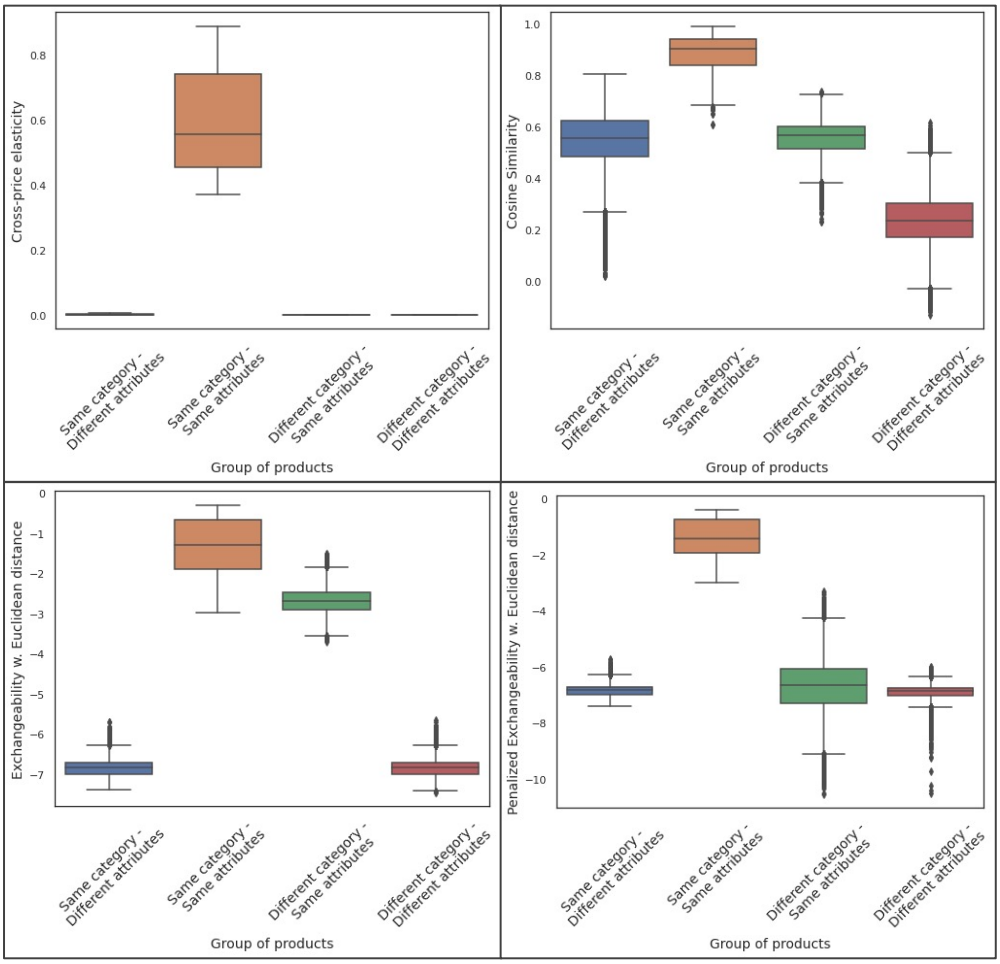



Figure 3: Cross-price elasticity, Cosine similarity, Exchangeability, and Penalized Exchangeability by Product Groups

Note: The upper-left and upper-right panels illustrate the true pairwise cross-price elasticities and the pairwise cosine similarities, respectively. The bottom-left and bottom-right panels illustrate the pairwise exchangeability and the penalized exchangeability scores, respectively.

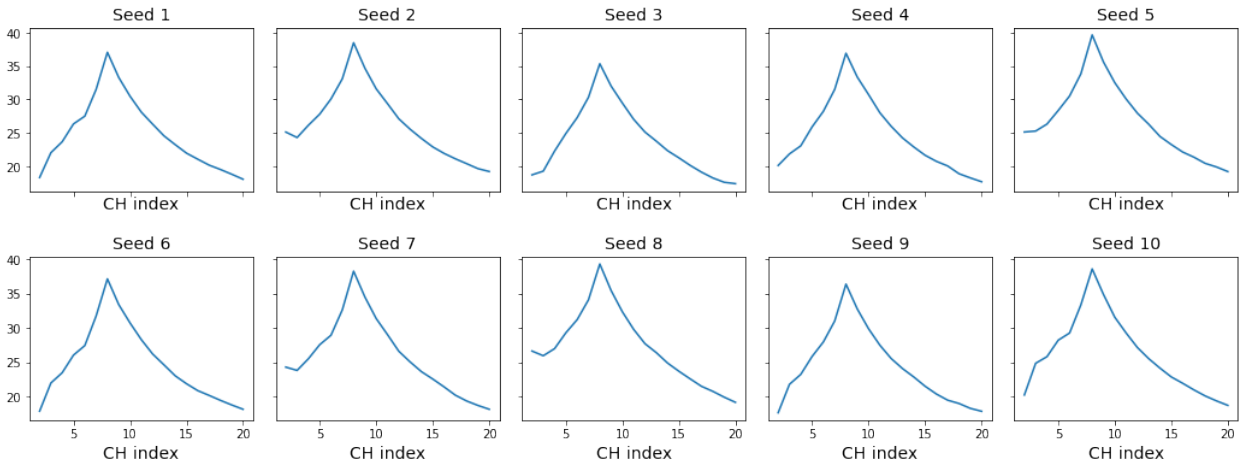



Figure 4: CH Index By Number of Clusters and Across Seeds

Note: Each figure illustrates the value of the CH index as a function of the number of clusters. Across seeds, the CH index is maximized when the number of clusters is equal to eight.

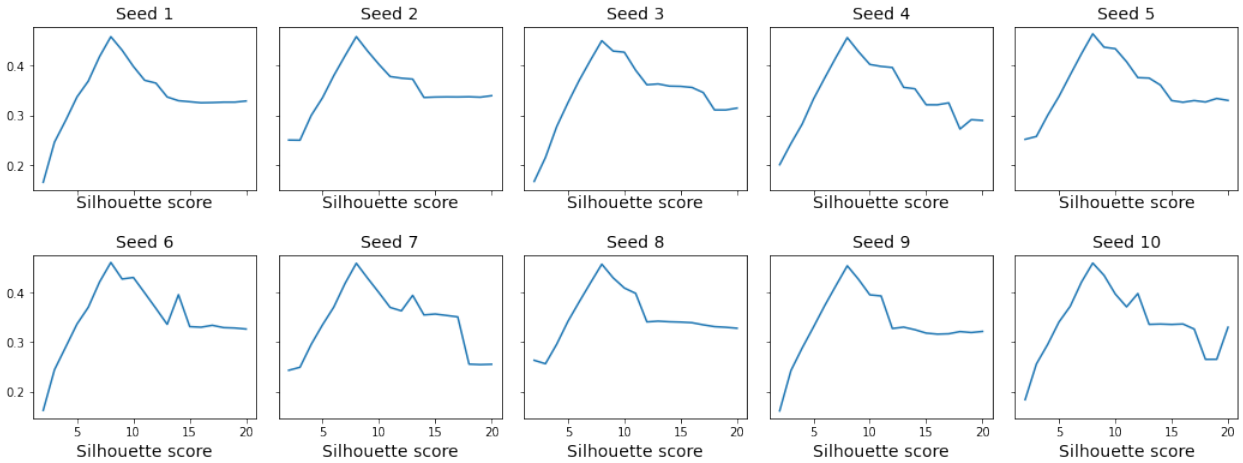



Figure 5: Silhouette Score by Number of Clusters and Across Seeds

Note: Each figure illustrates the value of the Silhouette scores as a function of the number of clusters. Across seeds, the Silhouette score is maximized when the number of clusters is equal to eight.

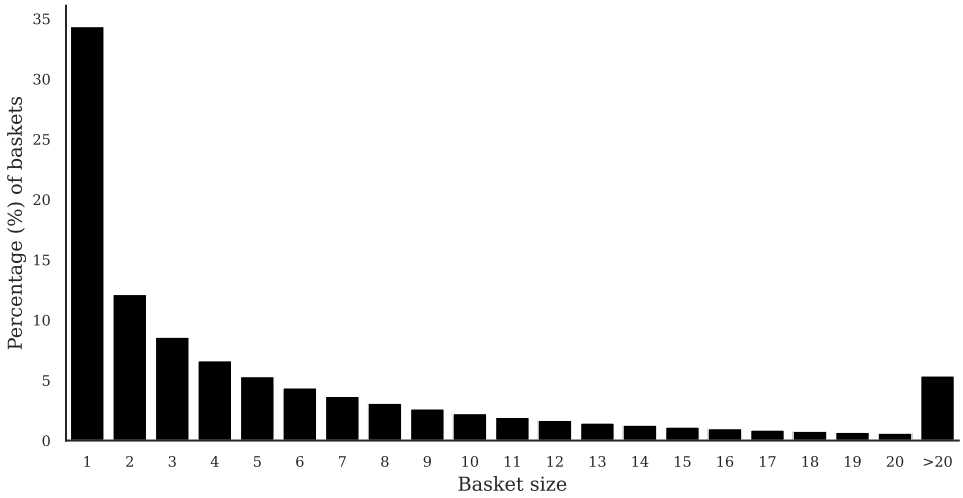



Figure 6: Distribution of Basket Size

Note: Baskets with more than 20 unique products are combined into the >20 bucket.

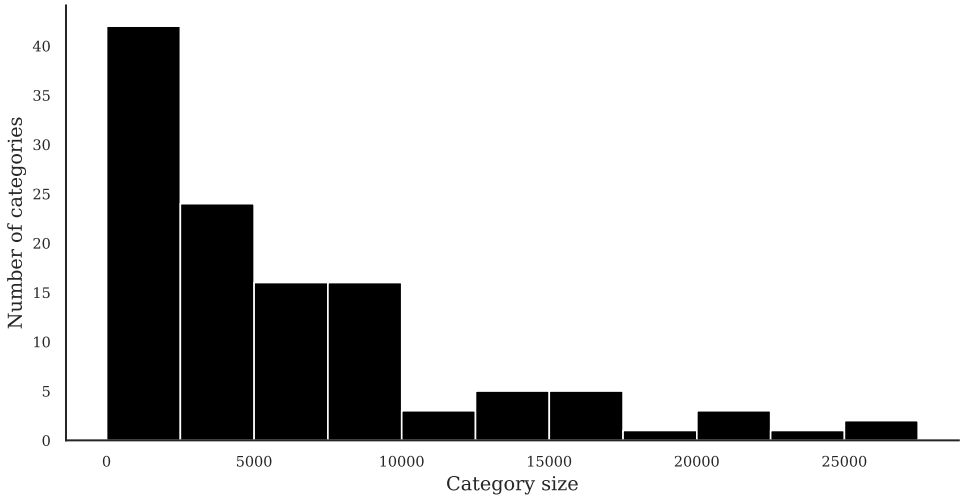



Figure 7: Distribution of Product Categories by the Number of Products They Contain.

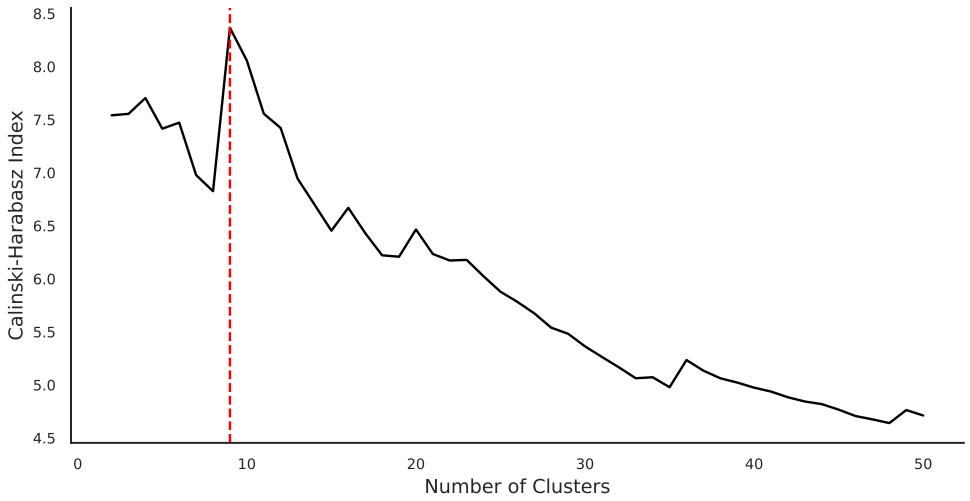
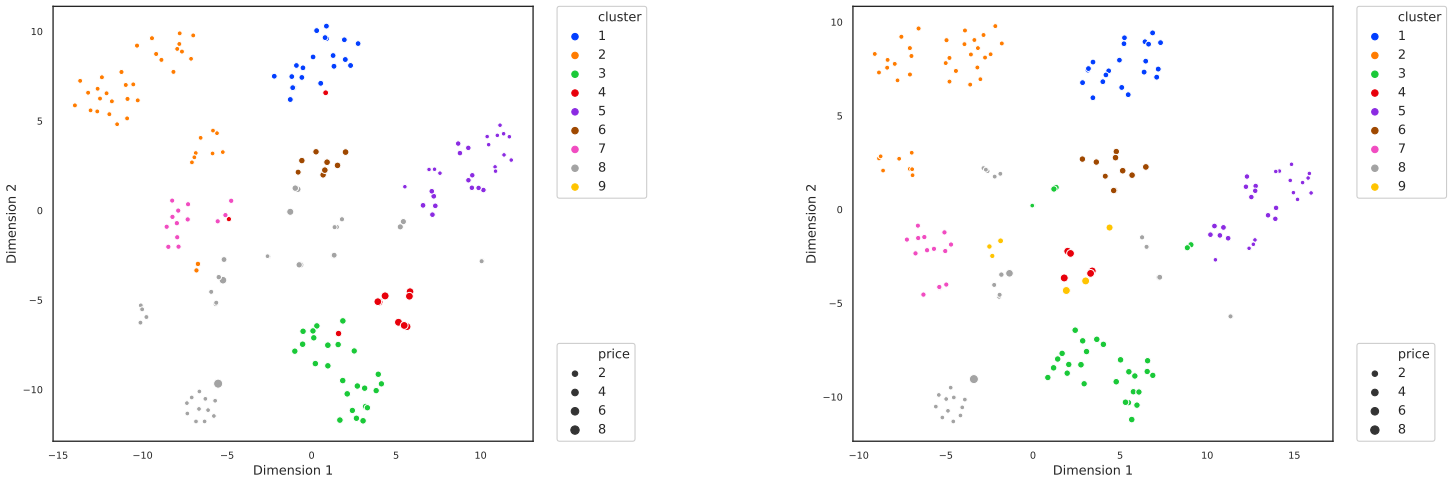



Figure 8: CH Index By Number of Clusters




(a) Seed 1

(b) Seed 2

Figure 9: Product Clusters

Table 1: Preference parameters and interpretation of the eight simulated segments

| Segment | $\alpha_{s1}$ | $\alpha_{s2}$ | $\delta_s$ | $\beta_s$ | Interpretation |
|---|---|---|---|---|---|
| 1 | 6 | -6 | 3 | -6 | Likes high levels of $X_1$, low levels of $X_2$, and it is price oriented |
| 2 | 6 | -6 | 6 | -3 | Likes high levels of X1, low levels of X2, and it is quality oriented |
| 3 | 6 | 6 | 3 | -6 | Likes high levels of X1, high levels of X2, and it is price oriented |
| 4 | 6 | 6 | 6 | -3 | Likes high levels of X1, high levels of X2, and it is quality oriented |
| 5 | -6 | 6 | 3 | -6 | Likes low levels of X1, high levels of X2, and it is price oriented |
| 6 | -6 | 6 | 6 | -3 | Likes low levels of X1, high levels of X2, and it is quality oriented |
| 7 | -6 | -6 | 3 | -6 | Likes low levels of X1, low levels of X2, and it is price oriented |
| 8 | -6 | -6 | 6 | -3 | Likes low levels of X1, low levels of X2, and it is quality oriented |

Table 2: Correlation between different complementarity and substitution metrics

| | Cross-price Elasticity | Copurchase Probability |
|---|---|---|
| Co-occurrence | 0.071 | 0.679 |
| Complementarity | −0.044 | 0.957 |
| Cosine Similarity | 0.448 | 0.398 |
| Exchangeability | 0.498 | 0.801 |
| Penalized Exchangeability | 0.902 | 0.000 |

Table 3: Price coefficients for 100 simulated products

| | (1) True Attributes | (2) Missing Q | (3) Missing A | (4) Embedding Clusters | (5) Product FEs |
|---|---|---|---|---|---|
| Price Segment 1 | $-6.186^{***}$ | $-6.053^{***}$ | $-4.148^{***}$ | $-6.110^{***}$ | $-6.611^{***}$ |
| | (0.129) | (0.124) | (0.058) | (0.127) | (0.141) |
| Price Segment 2 | $-6.157^{***}$ | $-6.231^{***}$ | $-4.543^{***}$ | $-5.729^{***}$ | $-5.979^{***}$ |
| | (0.137) | (0.139) | (0.059) | (0.109) | (0.143) |
| Price Segment 3 | $-6.224^{***}$ | $-6.136^{***}$ | $-2.836^{***}$ | $-5.925^{***}$ | $-6.231^{***}$ |
| | (0.129) | (0.126) | (0.045) | (0.133) | (0.132) |
| Price Segment 4 | $-6.003^{***}$ | $-5.795^{***}$ | $-2.514^{***}$ | $-5.818^{***}$ | $-6.249^{***}$ |
| | (0.114) | (0.110) | (0.045) | (0.121) | (0.127) |
| Price Segment 5 | $-2.838^{***}$ | $0.652^{***}$ | | $-2.997^{***}$ | $-3.180^{***}$ |
| | (0.062) | (0.021) | | (0.065) | (0.074) |
| Price Segment 6 | $-3.036^{***}$ | $0.676^{***}$ | | $-2.962^{***}$ | $-3.008^{***}$ |
| | (0.073) | (0.022) | | (0.077) | (0.079) |
| Price Segment 7 | $-3.150^{***}$ | $0.943^{***}$ | | $-2.937^{***}$ | $-2.631^{***}$ |
| | (0.079) | (0.027) | | (0.072) | (0.070) |
| Price Segment 8 | $-2.957^{***}$ | $0.665^{***}$ | | $-2.778^{***}$ | $-3.071^{***}$ |
| | (0.064) | (0.020) | | (0.061) | (0.068) |
| Loglike | -34,032 | -43,079 | -47,848 | -34,051 | -34,017 |
| AIC | 68,158 | 86,236 | 95,733 | 68,260 | 69,665 |
| BIC | 68,526 | 86,542 | 95,882 | 68,880 | 76,054 |
| In-sample HR | 0.451 | 0.311 | 0.257 | 0.418 | 0.455 |
| Out-of-sample HR | 0.404 | 0.274 | 0.217 | 0.385 | 0.402 |
| Running time (hh:mm:ss) | 00:09:32 | 00:08:14 | 00:02:46 | 00:17:00 | 19:17:00 |

*Significance levels:* * p<0.1, ** p<0.05, *** p<0.01.

*Note:* Models in column 1, 2, and 3 use log_unit_price. Models in column 4 and 4 use delta_log_unit_price.

Table 4: Embedding values for a given product across different seeds

| | dim 1 | dim 2 | dim 3 | dim 4 | dim 5 | ⋯ | dim 26 | dim 27 | dim 28 | dim 29 | dim 30 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| seed 1 | 0.032 | 0.035 | −0.389 | 0.534 | 0.399 | ⋯ | 0.628 | −0.901 | −1.059 | −0.100 | −0.205 |
| seed 2 | 0.886 | 0.837 | 0.177 | 0.417 | 0.218 | ⋯ | −0.725 | −0.387 | 0.199 | −0.230 | −0.723 |
| seed 3 | −0.517 | −0.766 | −0.366 | −0.543 | −0.482 | ⋯ | −0.041 | −0.867 | −0.986 | 0.377 | 1.415 |
| seed 4 | −0.734 | 0.106 | −0.234 | 1.345 | −0.146 | ⋯ | 0.550 | −0.794 | −0.044 | −0.717 | −1.598 |
| seed 5 | −0.845 | 0.101 | 0.007 | 0.806 | 0.087 | ⋯ | −0.279 | 1.433 | −1.033 | 0.047 | −0.423 |
| seed 6 | −1.588 | 0.194 | 0.177 | −0.457 | −1.311 | ⋯ | 0.352 | 0.772 | 0.547 | −0.996 | −1.457 |
| seed 7 | 1.023 | 0.652 | 0.587 | −1.718 | −0.236 | ⋯ | −1.146 | 0.157 | −0.011 | −0.055 | −0.075 |
| seed 8 | −0.170 | 0.512 | 0.823 | −0.813 | 0.159 | ⋯ | 0.626 | 0.706 | −0.383 | 0.585 | 1.278 |
| seed 9 | −0.280 | 0.750 | 0.212 | −0.415 | 0.926 | ⋯ | 1.013 | −0.165 | −0.629 | −0.327 | −0.317 |
| seed 10 | 0.564 | −1.023 | 0.640 | −1.037 | −0.252 | ⋯ | 0.198 | −0.048 | 0.190 | −0.338 | −0.096 |

Table 5: Correlation between pairwise cosine similarity between product embeddings across seeds

| | seed 1 | seed 2 | seed 3 | seed 4 | seed 5 | seed 6 | seed 7 | seed 8 | seed 9 | seed 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Seed 1 | 1.000 | 0.913 | 0.919 | 0.940 | 0.930 | 0.931 | 0.919 | 0.885 | 0.927 | 0.923 |
| Seed 2 | 0.913 | 1.000 | 0.888 | 0.928 | 0.939 | 0.908 | 0.927 | 0.916 | 0.900 | 0.942 |
| Seed 3 | 0.919 | 0.888 | 1.000 | 0.933 | 0.891 | 0.933 | 0.897 | 0.857 | 0.927 | 0.903 |
| Seed 4 | 0.940 | 0.928 | 0.933 | 1.000 | 0.926 | 0.937 | 0.928 | 0.883 | 0.931 | 0.921 |
| Seed 5 | 0.930 | 0.939 | 0.891 | 0.926 | 1.000 | 0.925 | 0.940 | 0.921 | 0.919 | 0.942 |
| Seed 6 | 0.931 | 0.908 | 0.933 | 0.937 | 0.925 | 1.000 | 0.917 | 0.890 | 0.949 | 0.916 |
| Seed 7 | 0.919 | 0.927 | 0.897 | 0.928 | 0.940 | 0.917 | 1.000 | 0.926 | 0.909 | 0.927 |
| Seed 8 | 0.885 | 0.916 | 0.857 | 0.883 | 0.921 | 0.890 | 0.926 | 1.000 | 0.897 | 0.916 |
| Seed 9 | 0.927 | 0.900 | 0.927 | 0.931 | 0.919 | 0.949 | 0.909 | 0.897 | 1.000 | 0.918 |
| Seed 10 | 0.923 | 0.942 | 0.903 | 0.921 | 0.942 | 0.916 | 0.927 | 0.916 | 0.918 | 1.000 |
| Attribute Similarity | 0.587 | 0.569 | 0.601 | 0.601 | 0.567 | 0.609 | 0.572 | 0.563 | 0.597 | 0.576 |

Table 6: Correlation between pairwise euclidean distances between product embeddings across seeds

| | seed 1 | seed 2 | seed 3 | seed 4 | seed 5 | seed 6 | seed 7 | seed 8 | seed 9 | seed 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Seed 1 | 1.000 | 0.920 | 0.921 | 0.938 | 0.921 | 0.924 | 0.928 | 0.903 | 0.928 | 0.929 |
| Seed 2 | 0.920 | 1.000 | 0.922 | 0.941 | 0.930 | 0.926 | 0.925 | 0.923 | 0.923 | 0.931 |
| Seed 3 | 0.921 | 0.922 | 1.000 | 0.929 | 0.903 | 0.925 | 0.918 | 0.891 | 0.930 | 0.922 |
| Seed 4 | 0.938 | 0.941 | 0.929 | 1.000 | 0.932 | 0.935 | 0.935 | 0.915 | 0.929 | 0.936 |
| Seed 5 | 0.921 | 0.930 | 0.903 | 0.932 | 1.000 | 0.923 | 0.931 | 0.926 | 0.914 | 0.934 |
| Seed 6 | 0.924 | 0.926 | 0.925 | 0.935 | 0.923 | 1.000 | 0.923 | 0.915 | 0.941 | 0.929 |
| Seed 7 | 0.928 | 0.925 | 0.918 | 0.935 | 0.931 | 0.923 | 1.000 | 0.926 | 0.916 | 0.930 |
| Seed 8 | 0.903 | 0.923 | 0.891 | 0.915 | 0.926 | 0.915 | 0.926 | 1.000 | 0.907 | 0.922 |
| Seed 9 | 0.928 | 0.923 | 0.930 | 0.929 | 0.914 | 0.941 | 0.916 | 0.907 | 1.000 | 0.931 |
| Seed 10 | 0.929 | 0.931 | 0.922 | 0.936 | 0.934 | 0.929 | 0.930 | 0.922 | 0.931 | 1.000 |
| Attribute Similarity | −0.549 | −0.552 | −0.556 | −0.554 | −0.541 | −0.569 | −0.552 | −0.572 | −0.572 | −0.566 |

Table 7: Dataset summary statistics

| | Whole Set | Training Set | Estimation Set | Test Set |
|---|---|---|---|---|
| # of baskets | 9,045,132 | 2,708,128 | 1,982 | 299 |
| # of households | 61,381 | 24,551 | 418 | 299 |
| # of products | 718,063 | 21,596 | 180 | 180 |
| # of categories | 118 | 111 | 1 | 1 |
| # of retailers | 761 | 730 | 1 | 1 |
| # of states | 49 | 49 | 1 | 1 |

Table 8: Summary statistics for the estimation set

(a) Continuous Variables

| Variable | Explanation | Mean | SD | Min | Median | Max |
|---|---|---|---|---|---|---|
| log_unit_price | unit price, take log | -3.559 | 0.814 | -13.816 | -3.650 | -1.391 |
| log_iv_unit_price | unit price instrument, take log | -3.547 | 0.800 | -6.435 | -3.621 | -1.411 |
| delta_log_unit_price | unit price, take log, demean | 0.000 | 0.293 | -11.048 | 0.006 | 1.519 |
| delta_log_iv_unit_price | unit price instrument, take log, demean | 0.000 | 0.205 | -2.255 | 0.007 | 1.215 |

(b) Categorical Variables

| Variable | Explanation | # Unique Values |
|---|---|---|
| upc_enc | product dummy, from 1 to 180 | 180 |
| cluster | cluster dummy, from 1 to 9 | 9 |
| brand | e.g., Coca-Cola | 12 |
| flavor | e.g., orange | 8 |
| type | e.g., soft drink | 3 |
| formula | regular or diet | 2 |
| container | bottle or can | 2 |

Table 9: Product examples and common attributes by cluster

| Cluster ID | Number of Products | Original UPC Description | Interpretation | Common Attribute(s) |
|---|---|---|---|---|
| 1 | 27 | CTL BR DT CH/CL CN 12P<br>CTL BR R LN/LM CN 12P | Private Label Diet Cherry/Cola Can 12 Pack<br>Private Label Regular Lemon/Lime Can 12 Pack | private label |
| 2 | 25 | PSI DT CL CN 12P<br>S DT OR CN CLP 12P | Pepsi Diet Cola Can 12 Pack<br>Sunkist Diet Orange Can 12 Pack | soft drink, can |
| 3 | 20 | PEPSI MAX DT CL NBP 6P<br>MT DI R LN/LM/CITR NBP 6P | Pepsi Max Diet Cola 6 Pack<br>Mountain Dew Regular Lemon/Lime/Citrus 6 Pack | soft drink, bottle, 6 pack |
| 4 | 20 | BRQ R RTBR NBP<br>SP R LN/LM CF NBP CT | Barq's Regular Root Beer<br>Sprite Regular Lemon/Lime Caffeine-Free | soft drink, regular |
| 5 | 19 | PSI R CL NBP<br>COKE DT CL NBP CT | Pepsi Regular Cola<br>Coke Diet Cola | soft drink, bottle |
| 6 | 13 | MT DI R LN/LM/CITR NBP<br>CRUSH R OR CF NBP | Mountain Dew Regular Lemon/Lime/Citrus<br>Crush Regular Orange Caffeine-Free | soft drink, bottle, mixed flavor |
| 7 | 11 | PSI R CL CN 24P<br>FANTA R OR NBP 6P | Pepsi Regular Cola Can 24 Pack<br>Fanta Regular Orange 6 Pack | soft drink, can, cola flavor |
| 8 | 7 | PSI R CL NBP 8P<br>COKE DT CL NBP 8P | Pepsi Regular Cola 8 Pack<br>Coke Diet Cola 8 Pack | soft drink, bottle, 8 pack |
| 9 | 39 | AMP R E-D CITR CN<br>SP IC DT SK AW BLK/RS NBP | AMP Regular Citrus Can<br>Sparkling Ice Diet Sparkling Water Black/Rapsberry | all others |

Table 10: Top exchangeable and complementary products in the Nielsen data

| **Focal UPC** | **Top Exchangeables** | | | **Top Complements** | | |
|---|---|---|---|---|---|---|
| Description | Description | Cosine Similarity | Penalized Exchangeability | Description | Cosine Similarity | Complementarity |
| PL Bread Wheat Split Top 20 OZ | PL Bread Wheat No Cholesterol 20OZ | 0.676 | -10.408 | PL Processed American Sliced Cheese 10.67OZ | 0.670 | 0.671 |
| | PL Bread Wheat 20OZ | 0.891 | -11.189 | PL Crackers Flaked Soda Salted 16OZ | 0.783 | 0.659 |
| | PL Bread Oat Bran 20OZ | 0.648 | -11.384 | PL Crackers Flaked Soda Salted 16OZ | 0.862 | 0.638 |
| | Dark Chocolate Mint Bar 4.4OZ | 0.572 | -11.510 | PL Non-Fat Greek Yogurt 5.3OZ | 0.779 | 0.626 |
| | Fresh Lettuce Roma Heart | 0.545 | -11.670 | PL Breast Turkey Black Forest Sliced 16OZ | 0.817 | 0.623 |
| Quaker Quick Oats Regular 18OZ | Oreo Mega Stuf Sandwich Cookies 13.2OZ | 0.319 | -5.983 | Yoplait Whips! Vanilla Cream Mousse 4OZ | 0.494 | 0.420 |
| | Thomas Bagel Everything NY Fresh 22OZ | 0.336 | -6.016 | 9 Lives Cat Food Wet Type 5.5OZ | 0.486 | 0.338 |
| | Snyder's of Hanover Salted Pretzel 12OZ | 0.337 | -6.080 | PL Milk 2% RD Fat Vitamin A/D Plastic 64OZ | 0.338 | 0.336 |
| | Sunshine Cheez-IT Salted Crackers 12.4 OZ | 0.385 | -6.096 | Purina Cat Chow Naturals 18PO | 0.442 | 0.311 |
| | Rold Gold Pretzel Tiny 16OZ | 0.368 | -6.110 | Quaker Oats Regular Old Fashioned 18OZ | 0.756 | 0.281 |
| Starbucks Frappuccino Liquid Coffee Mocha Bottle 15P | Nescafe Taster's Choice House Blend 12OZ | 0.427 | -12.98 | Columbus Bacon Hickory-Hardwood-Smoked Thigh Sliced Turkey 40OZ | 0.554 | 0.894 |
| | Folgers Classic Roast 16OZ | 0.513 | -13.357 | Hershey's Kisses Chocolate 56OZ | 0.684 | 0.855 |
| | Aleve Naproxen Sodium Regular Strength Caplet | 0.572 | -13.39 | Kellogg's Breakfast Bars Assorted Flavors 62.4OZ | 0.587 | 0.853 |
| | Pepperidge Farm Milano Tamper Seal 20P 0.75OZ | 0.443 | -13.681 | Quaker Oatmeal Squares 58OZ | 0.534 | 0.849 |
| | New World Farms Fresh Beans Organic | 0.497 | -13.752 | Nestle Nesquik Skim Milk 1% Low Fat Vitamin A/D Chocolate Plastic 15P 8OZ | 0.581 | 0.843 |
| Florida Natural Orange Juice Carton 59OZ | Simply Naked Pita Chips | 0.335 | -8.245 | PL Fresh Mushrooms | 0.499 | 0.941 |
| | Simply Orange Orange Juice Pulp Free Plastic | 0.410 | -8.301 | PL Bread Whole Wheat Premium | 0.498 | 0.920 |
| | Tropicana Orange Juice Plastic | 0.236 | -8.348 | Florida's Natural Orange Juice | 0.712 | 0.906 |
| | Oreo Sandwich Cookies Golden | 0.235 | -8.351 | Eat Smart Stir Fry Vegetable Mix 12OZ | 0.587 | 0.846 |
| | Mentos Pure Fresh Sugar Free Gum Fresh Mint | 0.202 | -8.353 | Village Farms Tomatoes 10OZ | 0.585 | 0.805 |
| PL Milk 1% Low Fat Vitamin A/D Plastic 128OZ | PL Milk 2% RD Fat Vitamin A/D Plastic 64OZ | 0.826 | -7.536 | PL Low Fat Yogurt Blackberry 6OZ | 0.791 | 0.734 |
| | PL Milk 2% RD Fat Vitamin A/D Plastic 64OZ | 0.880 | -7.743 | PL Frozen Lasagna 32OZ | 0.606 | 0.725 |
| | PL Milk 2% RD Fat Vitamin A/D Plastic 128OZ | 0.619 | -8.894 | Peach Bag 2PO | 0.591 | 0.722 |
| | PL Milk Whole Vitamin A/D Plastic 64OZ | 0.797 | -9.232 | PL Granola Bar Chocolatey Covered Chewy 6.5OZ | 0.797 | 0.716 |
| | PL Large White Eggs 12Q | 0.501 | -10.156 | PL Ground and Whole Bean Coffee Ind. Cups 12Q | 0.726 | 0.714 |
| Gold Peak Liquid Tea Sweet Plastic 89OZ | Maxwell House Coffee Breakfast Blend Light Roast Jar 25.6OZ | 0.164 | -6.48 | Gold Peak Liquid Tea Raspberry Plastic 52OZ | 0.594 | 0.665 |
| | Hershey's Syrup Chocolate Squeeze Bottle 24OZ | 0.261 | -6.656 | Gold Peak Liquid Tea Lemonade Plastic 52OZ | 0.541 | 0.649 |
| | Hillshire Farm Deli Select 7OZ | 0.214 | -6.664 | PL Milk 2% RD Fat Vitamin A/D Plastic 128OZ | 0.412 | 0.621 |
| | Hillshire Farm Deli Select 7OZ | 0.241 | -6.704 | Bimbo Bread White 24OZ | 0.418 | 0.514 |
| | Folgers Coffee Med-Dark Roast 100% Colombian 24.2 OZ | 0.207 | -6.734 | Gold Peak Liquid Tea Unflavored Plastic 89OZ | 0.545 | 0.476 |

Table 11: Choice model results for the carbonated beverages category

| | (1) Embedding Cluster | (2) Product FE [15] | (3) Attribute | (4) Missing Attribute |
|---|---|---|---|---|
| Price | $-2.187^{***}$ | $-2.601^{***}$ | $-0.781^{***}$ | $-0.100$ |
| | (0.185) | (0.276) | (0.070) | (0.051) |
| Loyal | $3.874^{***}$ | $3.408^{***}$ | $3.407^{***}$ | $4.636^{***}$ |
| | (0.081) | (0.085) | (0.070) | (0.073) |
| Residual | $3.386^{***}$ | $4.279^{***}$ | $1.936^{***}$ | $1.184^{***}$ |
| | (0.179) | (0.292) | (0.091) | (0.067) |
| sd.Price | $0.322^{***}$ | $2.745^{***}$ | $0.159^{***}$ | $0.254^{***}$ |
| | (0.046) | (0.139) | (0.027) | (0.055) |
| sd.Loyal | $1.342^{***}$ | $1.295^{***}$ | $1.031^{***}$ | $1.443^{***}$ |
| | (0.120) | (0.102) | (0.086) | (0.119) |
| sd.Residual | $2.005^{***}$ | $-0.202^{**}$ | $1.885^{***}$ | $1.538^{***}$ |
| | (0.086) | (0.078) | (0.074) | (0.065) |
| Fixed Effects | Cluster | Product | | |
| N | 356760 | 356760 | 356760 | 356760 |
| Loglike | $-7048.669$ | $-6294.620$ | $-6681.031$ | $-7594.428$ |
| AIC | 14141.338 | 13317.240 | 13462.062 | 15200.856 |
| BIC | 14264.359 | 15352.678 | 13741.655 | 15234.407 |
| In-sample HR | 0.359 | 0.450 | 0.397 | 0.339 |
| Out-of-sample HR | 0.254 | 0.301 | 0.291 | 0.241 |
| Running time (hh:mm:ss) | 00:20:42 | 04:49:57 | 00:59:35 | 00:05:14 |
| Number of parameters | 22 | 364 | 50 | 6 |

*Significance levels:* * p<0.1, ** p<0.05, *** p<0.01.

*Note:* The Embedding Cluster and Product FE models use delta_log_unit_price, and the Attributes and Missing Attributes models use log_unit_price.

# Online appendix

# A Literature Comparison

Table 12: Comparison with relevant literature

| Paper | Model | Application | Data | Causal Inference | Interpretable |
|---|---|---|---|---|---|
| Grbovic et al. (2015) | Prod2vec, bagged prod2vec, SGNS | Predict next product to purchase, product recommendation | Email receipt | N | N |
| Barkan and Koenigstein (2016) | SGNS | Recommender system (find similar item) | product orders | N | N |
| Gabel et al. (2019) | SGNS | market structure | Product orders | N | N |
| Gabel et al. (2021) | Autoencoder | Demand prediction, elasticity, coupon targeting | loyalty, basket, coupon | Y | Y |
| Armona, Lewis and Zervas (2021) | Bayesian personalized ranking | learn latent product attributes and consumer preferences from search data; combine with demand estimation to predict demand; post merger demand estimation | Search + aggregate demand | Y | N |
| Padilla and Ascarza (2021) | SGNS | Customer segmentation, CLV prediction | Transaction, Marketing action, Acquisition characteristics | Y | N |
| Kumar, Eckles and Aral (2020) | SGNS | Bundling | Order, search | N | N |
| **This Paper** | **SGNS** | **Demand estimation, identifying competitors and brand alliance, targeted pricing** | **Product orders** | **Y** | **Y** |

# B Additional proof details

## B.1 Figure illustration

Figure 10 illustrates the intuition behind this model using a simplified example. Suppose there are three product categories, $c_1$, $c_2$, and $c_3$, where $c_1$ and $c_2$ are complements, $c_2$ and $c_3$ are complements, whereas $c_1$ and $c_3$ are substitutes. For instance, $c_1$, $c_2$, and $c_3$ can stand for Coffee, Bread, and Tea, where Bread and Coffee are complements, Bread and Tea are complements, and Coffee and Tea are substitutes. Each category is represented with a big box on the top of the figure, with substitutes in the same color (green) and complements in another color (blue).

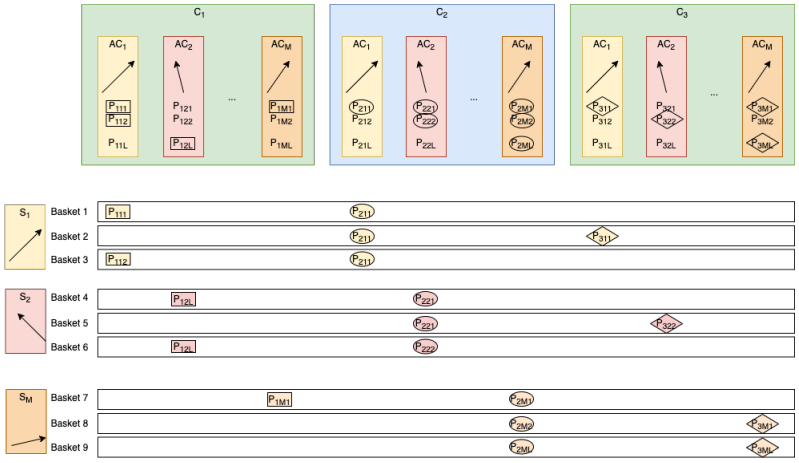



Figure 10: Model Illustration

Within each product category, there are $\|A\|$ unique attribute combinations, $\overrightarrow{AC_1}, \ldots, \overrightarrow{AC_{\|A\|}}$. The three examples, $\overrightarrow{AC_1}$, $\overrightarrow{AC_2}$, and $\overrightarrow{AC_3}$ are shown as the yellow, pink, and orange boxes, respectively. Each unique attribution combination $\overrightarrow{AC_\tau}, \tau \in \{1, \ldots, \|A\|\}$ is a $k$-dimensional vector of product attributes, that is, $\overrightarrow{AC_\tau} = \left(X_1^\tau, \ldots, X_K^\tau, Q^\tau, \bar{P}^\tau\right)^T$. [16]

[16] We use the superscript $\tau$ here instead of the subscript $j$ in Equation 6 to differentiate the index for unique attribute combinations from the index for individual products. The different indexes allow us to capture scenarios when two products $a$ and $b$ share the same attribute combination $\overrightarrow{AC_\tau}$; that is,

$$(X_{a1}, \ldots, X_{ak}, Q_a, \bar{P}_a) = (X_{b1}, \ldots, X_{bk}, Q_b, \bar{P}_b) = (X_1^\tau, \ldots, X_k^\tau, Q^\tau, \bar{P}^\tau) \tag{26}$$

Because an attribute combination $\overrightarrow{AC}$ represents the $k$-dimensional vector of product attributes, in Figure 10, we use an arrow to visualize each attribute combination. For example, $\overrightarrow{AC_1}$ is visualized by an arrow that points to the top right, whereas $\overrightarrow{AC_2}$ is visualized by an arrow that points to the top left. All products that share the same attribute combination are presented in the same colored box. For example, products $j_{111}, j_{112}, \ldots, j_{11L}$ have the same attribute combination $\overrightarrow{AC_1}$. The three subscripts for a product $j_{c\tau p}$, namely $c$, $\tau$, and $p$, represent the category $c$, attribute combination $\tau$, and the $p$th product, respectively.

As illustrated in the bottom half of Figure 10, each segment $s$ is represented by a $k$-dimensional attribute preference vector $\overrightarrow{\theta_s} = (\alpha_{s1}, \ldots, \alpha_{sk}, \delta_s, \beta_s)^T$. The segment-specific attribute preference vectors are visualized by the arrows in the boxes on the left.

## B.2 Proof of Proposition 1

In this appendix, we consider how the number of unique cross-category copurchase patterns for each consumer segment is related to the number of consumer segments $S$ and the number of unique attribute combinations $\|A\|$. We discuss three cases when the number of consumer segments $S$ is the same as, greater than, and less than the number of unique attribute combinations $\|A\|$.

1. Case 1: $S = \|A\|$

   When $S = \|A\|$, then each consumer segment will have a unique cross-category copurchase pattern; in other words, each consumer segment prefers a unique attribute combination across all product categories.

   For instance, segment 1 ($s_1$) consumers prefer attribute combination $\overrightarrow{AC_1}$ because the utility generated from products with attribute combination $\overrightarrow{AC_1}$ (the inner product of the segment attribute preference vector $\overrightarrow{\theta_{s_1}}$ and the attribute combination vector $\overrightarrow{AC_1}$) is the highest among all alternative attribute combinations. Given segment 1 ($s_1$) consumers' preference, Figure 10 shows $s_1$'s three representative shopping baskets: Basket 1, Basket 2, and Basket 3. In Basket 1, (complementary) products $j_{111}$ and $j_{211}$ are copurchased; in Basket 2, (comple-

mentary) products $j_{211}$ and $j_{311}$ are copurchased; and in Basket 3, (complementary) products $j_{112}$ and $j_{211}$ are copurchased.

A concrete example could be a set of consumers who have limited storage space and who are extremely health conscious. Therefore, they buy small-size, organic, private-label coffee and small-size, sugar-free bread in Basket 1, small-size, sugar-free bread and small-size, sugar-free tea in Basket 2, and small-size, organic, national-brand coffee and small-size, sugar-free bread in Basket 3. Segment 1 (s1) consumers do not or rarely purchase products with other attribute combinations, and thus the copurchase probabilities when one or both products have other attribute combinations are close to zero. So, the number of unique cross-category copurchase patterns for segment 1 (and similarly, for each segment) is equal to the number of unique attribute combinations $\|A\|$.

2. Case 2: $S < \|A\|$

   When $S < \|A\|$, there are multiple distinct attribute combinations that are most preferred for the same consumer segment. For example, two distinct attribute combinations $\overrightarrow{AC_\tau}$ and $\overrightarrow{AC_{\tau'}}$ are both the most preferred attribute combination for a particular consumer segment $s$. In this case, they will have the same copurchase pattern. Thus, the number of unique copurchase patterns for each segment is $S$, not $\|A\|$.

3. Case 3: $S > \|A\|$

   When $S > \|A\|$, it means that multiple consumer segments prefer the same attribute combination. For example, even if two consumer segments $s$ and $s'$ have different attribute preferences (i.e., $\overrightarrow{\theta_s} \neq \overrightarrow{\theta_{s'}}$), they might have the same preferred attribute combination. These two segments of consumers will end up with the same purchase pattern because the market is not fine-grained enough to provide different products for the different needs of the two segments of consumers. Consequently, the number of unique copurchase patterns for each segment is $\|A\|$, not $S$.

Combining the three cases above, we can conclude that the number of unique copurchase patterns for each segment equals the minimum of $S$ and $\|A\|$, which is Proposition 1.

## B.3 Proof of Lemma 1

If two products $j$ and $l$ in the same category (i.e., $c_j = c_l$) have the same individual-specific time-invariant product utility (i.e., $\gamma_{ij} = \gamma_{il}$), we prove that, under certain circumstances, their overall (cross-time) purchase probabilities are identical (i.e., $\Pr_{ij} = \Pr_{il}$).

*Proof.*

$$\Pr_{ij} - \Pr_{il} = \sum_t \Pr_{ijt} - \sum_t \Pr_{ilt} \tag{27a}$$

$$= \sum_t \frac{\exp\left(\gamma_{ij} + \beta_i \Delta P_{jt}\right)}{\sum_{j' \in c_j} \exp\left(\gamma_{ij'} + \beta_i \Delta P_{j't}\right)} - \sum_t \frac{\exp\left(\gamma_{il} + \beta_i \Delta P_{lt}\right)}{\sum_{j' \in c_l} \exp\left(\gamma_{ij'} + \beta_i \Delta P_{j't}\right)} \tag{27b}$$

$$= \sum_t \frac{1}{\sum_{j' \in c_j} \exp\left(\gamma_{ij'} + \beta_i \Delta P_{j't}\right)} * \left[\exp\left(\gamma_{ij} + \beta_i \Delta P_{jt}\right) - \exp\left(\gamma_{ij} + \beta_i \Delta P_{lt}\right)\right] \tag{27c}$$

Note that from 27b to 27c, we use the conditions that $c_j = c_l$ and $\gamma_{tj} = \gamma_{il}$.

$$\begin{aligned} &\text{Let } A := \gamma_{ij}; \\ &\quad x_t := \beta_i \Delta P_{jt}, x_t \sim N(0, \sigma_j^2); \\ &\quad x'_t := \beta_i \Delta P_{lt}, x'_t \sim N(0, \sigma_j^2); \\ &\quad B_t := \sum_{j' \in c_j} \exp\left(\gamma_{ij'} + \beta_i \Delta P_{j't}\right) \end{aligned} \tag{28}$$

$$\begin{aligned} &\because \sum_t \exp\left(A + x_t\right) = T * \exp\left(A + \frac{\sigma_j^2}{2}\right) = \sum_t \exp\left(A + x'_t\right) \\ &\therefore \Pr_{ij} - \Pr_{il} = \sum_t \frac{\exp\left(A + x_t\right) - \exp\left(A + x'_t\right)}{B_t} = 0 \end{aligned} \tag{29}$$

Note that in Equation 29, we apply L'Hôpital's Rule and take the limits for the numerator and denominator respectively. Thus, $\Pr_{ij} = \Pr_{il} = T \frac{\exp\left(\gamma_{ij} + \frac{\sigma_j^2}{2}\right)}{\sum_{j' \in c_j} \exp\left(\gamma_{ij'} + \frac{\sigma_{j'}^2}{2}\right)}$, where $\beta_i \Delta P_{jt} \sim N\left(0, \sigma_j^2\right)$.

Although in each time period $t$, their purchase probabilities are different, $\frac{\exp(\gamma_{ij}+\beta_i \Delta P_{jt})}{\sum_{j' \in c_j} \exp(\gamma_{ij'}+\beta_i \Delta P_{j't})} \neq \frac{\exp(\gamma_{il}+\beta_i \Delta P_{lt})}{\sum_{j' \in c_l} \exp(\gamma_{ij'}+\beta_i \Delta P_{j't})}$, after averaging over time, the overall (cross-time) purchase probabilities become the same.

When the number of time periods is finite, if we further assume that the sum of the exponential of deterministic utilities across all products in this category keeps constant over time (i.e., $B_t \equiv B, \forall t$ in Equation 28), and the cross-time price variations of the two products $j$ and $l$ have the opposite patterns across the finite number of time periods (i.e., $\sum_t [\exp(A+x_t) - \exp(A+x'_t)] = 0$ in Equation 29), then their price variations will cancel out when we calculate the overall (cross-time) purchase probabilities (i.e., $\Pr_{ij} = \Pr_{il}$). □

## B.4 Objective function of the product fixed effects model

In this appendix, we write out the objective function (i.e., log likelihood) of the product fixed effects model, link it to product copurchase probability, and represent it at the individual or segment level. Then we examine the dimensionality of the objective function, which is equivalent to the number of unique copurchase patterns in the product fixed effects model.

**Log likelihood** We decompose the log likelihood of copurchasing multiple products in one basket into two parts: (1) copurchasing multiple categories that these products belong to; and (2) purchasing each of these products, given having decided to purchase in the corresponding cate-

gory.

$$
\begin{aligned}
\mathscr{LL} &= \log \prod_i \prod_t \Pr\left(\overrightarrow{Y_{it}}\right) \\
&= \log \prod_i \prod_t \Pr\left(y_{i1t}, \ldots, y_{iCt}\right) \\
&= \log \prod_i \prod_t \int (2\pi)^{-\frac{C}{2}} \det(\Omega)^{-\frac{1}{2}} e^{-\frac{1}{2}\overrightarrow{\omega_{it}}'\Omega^{-1}\overrightarrow{\omega_{it}}} d\overrightarrow{\omega_{it}} \prod_{c=1}^{C} \prod_{j_c=1}^{J_c} \left[\Pr\left(y_{ij_ct}\right)^{\mathbb{1}\left(y_{ij_ct}=1\right)}\right] \\
&= \sum_i \sum_t \left\{ \log \left[ \int (2\pi)^{-\frac{C}{2}} \det(\Omega)^{-\frac{1}{2}} e^{-\frac{1}{2}\overrightarrow{\omega_{it}}'\Omega^{-1}\overrightarrow{\omega_{it}}} d\overrightarrow{\omega_{it}} \right] + \sum_{c=1}^{C} \sum_{j_c=1}^{J_c} \log\left[\Pr\left(y_{ij_ct}\right)\right] \mathbb{1}\left(y_{ij_ct}=1\right) \right\} \\
&= \mathscr{LL}_{\text{category}} + \mathscr{LL}_{\text{product}},
\end{aligned}
\tag{30}
$$

where $\omega_{it}$ is the individual- and time-specific category utility in Equation 4 for all categories, whether they are chosen or not; $\int (2\pi)^{-\frac{C}{2}} \det(\Omega)^{-\frac{1}{2}} e^{-\frac{1}{2}\overrightarrow{\omega_{it}}^T\Omega^{-1}\overrightarrow{\omega_{it}}} d\overrightarrow{\omega_{it}}$ is the integration of the density function of the multivariate normal distribution $N(0,\Omega)$ over the area of $\bigcap_{c=1}^{C}\{\omega_{ict} > 0 \text{ if } y_{ict} > 0; \text{ otherwise, } \omega_{ict} \leq 0\}$ (Manchanda et al. 1999); $y_{ij_ct}$ is a dummy variable indicating whether or not product $j_c$ is chosen by consumer $i$ at time $t$ in category $c$; $\Pr\left(y_{ij_ct}\right)$ is the probability of purchasing product $j_c$ in category $c$; and $\mathbb{1}(\cdot)$ is the indicator function. [17]

**Linking log likelihood to product copurchase probability** From now on, we do not consider the first part of Equation 30, the category copurchase probability, but focus on the second part of Equation 30, the log likelihood of product choices and link it to the copurchase probability of each

[17] Note that the partition of a multivariate normal distribution is also multivariate normal, and we use this property in the simulation.

pair of products.

$$\mathscr{LL}_{\text{product}} = \sum_i \sum_t \sum_{c=1}^{C} \sum_{j_c=1}^{J_c} \left\{ \log \left[ \Pr\left(y_{ij_ct}\right) \right] \mathbb{1}\left(y_{ij_ct} = 1\right) \right\} \tag{31a}$$

$$= \sum_{b_{it} \in B} \left\{ \sum_{j=1}^{J} \log \left[ \Pr\left(y_{ijt}\right) \right] \mathbb{1}\left(j \in b_{it}\right) \right\} \tag{31b}$$

$$= \sum_{b_{it} \in B} \left\{ \frac{1}{2d_b} \sum_{j \in b_{it}} \left\{ 2d_b * \log \left[ \Pr\left(y_{ijt}\right) \right] \right\} \right\} \tag{31c}$$

$$= \frac{1}{2d_b} \sum_{b_{it} \in B} \sum_{l,r \in b_{it}} \left\{ \log \left[ \Pr\left(y_{ilt}\right) \right] + \log \left[ \Pr\left(y_{irt}\right) \right] \right\} \tag{31d}$$

$$= \frac{1}{2d_b} \sum_{b_{it} \in B} \sum_{l,r \in b_{it}} \log \left[ \Pr\left(y_{ilt}\right) * \Pr\left(y_{irt}\right) \right] \tag{31e}$$

where $b_{it}$ is the basket $b$ that consumer $i$ buys during purchase occasion $t$, $B$ is the set of all baskets $b$, and $d_b$ is the number of context products in basket $b$; that is, the basket size of basket $b$ minus one.

Note that from 31c to 31d, the multiplier $\frac{1}{2d_b}$ is moved out of the bracket, and the sum over all individual products $j$ for $2d_b$ times is equivalent to the sum over all possible pairs of products $l$ and $r$. For example, for a basket that contains three products $A, B, C$, $d_b = 2$. In 31c, each of the three products is counted $2d_b = 4$ times, and in 31d, each of the six pair of products $A\&B, A\&C, B\&C, B\&A, C\&A, C\&B$ is counted once, and each product is also counted 4 times. In this way, we transform the log likelihood of purchasing each individual product $j$ into the combined log likelihood of copurchasing each pair of products $l$ and $r$.

**Log likelihood at the individual or segment level** We use the product utility in Equation 11 to represent the product purchase probability $\Pr\left(y_{ijt}\right)$ in Equation 31:

$$\Pr\left(y_{ijt}\right) = \frac{\exp\left(\gamma_{ij} + \beta_i \Delta P_{jt}\right)}{\sum_{j' \in c_j} \exp\left(\gamma_{ij'} + \beta_i \Delta P_{j't}\right)} \tag{32}$$

We then represent the log likelihood at the individual or segment level:

$$\begin{aligned}\mathscr{LL}_{\text{product}} &= \frac{1}{2d_b}\sum_{b_{it}\in B}\sum_{l,r\in b_{it}}\log\left[\frac{\exp\left(\gamma_{il}+\beta_i\Delta P_{lt}\right)}{\sum_{l'\in c_l}\exp\left(\gamma_{il'}+\beta_i\Delta P_{l't}\right)} * \frac{\exp\left(\gamma_{ir}+\beta_i\Delta P_{rt}\right)}{\sum_{r'\in c_r}\exp\left(\gamma_{ir'}+\beta_i\Delta P_{r't}\right)}\right] \\ &= \frac{1}{2d_b}\sum_i\sum_t\sum_{l,r\in b_{it}}\log\left[\frac{\exp\left(\gamma_{il}+\beta_i\Delta P_{lt}\right)}{\sum_{l'\in c_l}\exp\left(\gamma_{il'}+\beta_i\Delta P_{l't}\right)} * \frac{\exp\left(\gamma_{ir}+\beta_i\Delta P_{rt}\right)}{\sum_{r'\in c_r}\exp\left(\gamma_{ir'}+\beta_i\Delta P_{r't}\right)}\right] \\ &\quad \text{(Individual level)} \\ &= \frac{1}{2d_b}\sum_s N_s\left\{\sum_t\sum_{l,r\in b_{it}}\log\left[\frac{\exp\left(\gamma_{sl}+\beta_s\Delta P_{lt}\right)}{\sum_{l'\in c_l}\exp\left(\gamma_{sl'}+\beta_s\Delta P_{l't}\right)} * \frac{\exp\left(\gamma_{sr}+\beta_s\Delta P_{rt}\right)}{\sum_{r'\in c_r}\exp\left(\gamma_{sr'}+\beta_s\Delta P_{r't}\right)}\right]\right\} \\ &\quad \text{(Segment level)},\end{aligned} \tag{33}$$

where $N_s$ is the number of consumers in the segment $s$.

**Number of unique copurchase patterns** Next we examine the dimensionality of Equation 33. Because there are $\|A\|$ unique attribute combinations, there are $\|A\|^2$ unique values of copurchase probabilities $\left(\frac{\exp(\gamma_{sl}+\beta_s\Delta P_{lt})}{\sum_{l'\in c_l}\exp\left(\gamma_{sl'}+\beta_s\Delta P_{l't}\right)} * \frac{\exp(\gamma_{sr}+\beta_s\Delta P_{rt})}{\sum_{r'\in c_r}\exp\left(\gamma_{sr'}+\beta_s\Delta P_{r't}\right)}\right)$. And because there are $S$ unique segments, the number of unique copurchase probabilities for all segments is equal to the number of segments $S$ times the squared value of the number of unique attribute combinations (i.e., $S*\|A\|^2$).

$$\begin{aligned}\mathscr{LL}_{\text{product}} &= \frac{1}{2d_b}\sum_s N_s\left\{\sum_t\sum_{l,r\in b_{it}}\log\left[\frac{\exp\left(\gamma_{sl}+\beta_s\Delta P_{lt}\right)}{\sum_{l'\in c_l}\exp\left(\gamma_{sl'}+\beta_s\Delta P_{l't}\right)} * \frac{\exp\left(\gamma_{sr}+\beta_s\Delta P_{rt}\right)}{\sum_{r'\in c_r}\exp\left(\gamma_{sr'}+\beta_s\Delta P_{r't}\right)}\right]\right\} \\ &= \sum_{h=1}^{S*H}\log\left[\text{CP}_h\right] * \text{CPFreq}_h,\end{aligned} \tag{34}$$

where $\text{CP}_h = \frac{\exp(\gamma_{sl}+\beta_s\Delta P_{lt})}{\sum_{l'\in c_l}\exp\left(\gamma_{sl'}+\beta_s\Delta P_{l't}\right)} * \frac{\exp(\gamma_{sr}+\beta_s\Delta P_{rt})}{\sum_{r'\in c_r}\exp\left(\gamma_{sr'}+\beta_s\Delta P_{r't}\right)}$ is a unique value of copurchase probability and $\text{CPFreq}_h = \frac{1}{2d_b}\sum_s N_s\sum_t\sum_{l,r\in b_{it}}\mathbb{1}\left(\Pr\left(y_{ilt}\right) * \Pr\left(y_{irt}\right) = \text{CP}_h\right)$.

If two products $j$ and $l$ in the same category (i.e., $c_j = c_l$) have the same individual-specific time-invariant product utility (i.e., $\gamma_{sj} = \gamma_{sl}$), they will have the same copurchase probability with

products in another category $r$ for all segments. This can be easily seen as

$$\begin{aligned}
\mathrm{CP}_{j,r} &= \frac{\exp\left(\gamma_{sj}+\beta_s\Delta P_{jt}\right)}{\sum_{l'\in c_l}\exp\left(\gamma_{sl'}+\beta_s\Delta P_{l't}\right)} * \frac{\exp\left(\gamma_{sr}+\beta_s\Delta P_{rt}\right)}{\sum_{r'\in c_r}\exp\left(\gamma_{sr'}+\beta_s\Delta P_{r't}\right)} \\
&= \frac{\exp\left(\gamma_{sl}+\beta_s\Delta P_{lt}\right)}{\sum_{l'\in c_l}\exp\left(\gamma_{sl'}+\beta_s\Delta P_{l't}\right)} * \frac{\exp\left(\gamma_{sr}+\beta_s\Delta P_{rt}\right)}{\sum_{r'\in c_r}\exp\left(\gamma_{sr'}+\beta_s\Delta P_{r't}\right)} \\
&= \mathrm{CP}_{l,r}
\end{aligned} \tag{35}$$

**Proof by contradiction** Next, we prove that products with the same copurchase patterns for all segments will have the same segment-specific time-invariant product utility.

We prove this by contradiction. Suppose two products within the same category $j$ and $l$ have different individual-specific time-invariant product utilities (i.e., $\gamma_{ij} \neq \gamma_{il}$); then their copurchase probability with product $r$ in another category will be

$$\begin{aligned}
\mathrm{CP}_{j,r} &= \frac{\exp\left(\gamma_{sj}+\beta_s\Delta P_{jt}\right)}{\sum_{l'\in c_l}\exp\left(\gamma_{sl'}+\beta_s\Delta P_{l't}\right)} * \frac{\exp\left(\gamma_{sr}+\beta_s\Delta P_{rt}\right)}{\sum_{r'\in c_r}\exp\left(\gamma_{sr'}+\beta_s\Delta P_{r't}\right)} \\
\mathrm{CP}_{l,r} &= \frac{\exp\left(\gamma_{sl}+\beta_s\Delta P_{lt}\right)}{\sum_{l'\in c_l}\exp\left(\gamma_{sl'}+\beta_s\Delta P_{l't}\right)} * \frac{\exp\left(\gamma_{sr}+\beta_s\Delta P_{rt}\right)}{\sum_{r'\in c_r}\exp\left(\gamma_{sr'}+\beta_s\Delta P_{r't}\right)},
\end{aligned} \tag{36}$$

which are not equal. Thus, these two products will have different copurchase probabilities. By contradiction, products with the same copurchase patterns for all segments will have the same individual-specific time-invariant product utility.

## B.5 Proof of Proposition 3

Based on Equations 8 and 32, we can calculate the log of the odds ratio for choosing product $j$:

$$\begin{aligned}
\log \mathrm{OR}_{ijt} &= \log\left[\frac{\Pr(y_{ijt}=1)}{\Pr(y_{ijt}=0)}\right] \\
&= \gamma_{ij}+\beta_i\Delta P_{jt} \\
&= \sum_{k=1}^{K}\alpha_{ik}X_{jk}+\delta_i Q_j+\beta_i\left(\bar{P}_j+\Delta P_{jt}\right)
\end{aligned} \tag{37}$$

which is a linear function of $\bar{P}_j$ and $\Delta P_{jt}$. We could run a linear regression between the log of the odds ratio and the two components of price:

$$\log \mathrm{OR} = \alpha+\beta_1\bar{P}+\beta_2\Delta P+\varepsilon \tag{38}$$

And the solutions are

$$\begin{aligned}\hat{\beta}_1 &= \frac{\left(\Sigma\left(\Delta P\right)^2\right)\left(\Sigma\left(\bar{P}\cdot\log \mathrm{OR}\right)\right)-\left(\Sigma\left(\bar{P}\cdot\Delta P\right)\right)\left(\Sigma\left(\Delta P\cdot\log \mathrm{OR}\right)\right)}{\left(\Sigma\left(\bar{P}\right)^2\right)\left(\Sigma\left(\Delta P\right)^2\right)-\left(\Sigma\left(\bar{P}\cdot\Delta P\right)\right)} \\ \hat{\beta}_2 &= \frac{\left(\Sigma\left(\bar{P}\right)^2\right)\left(\Sigma\left(\Delta P\cdot\log \mathrm{OR}\right)\right)-\left(\Sigma\left(\bar{P}\cdot\Delta P\right)\right)\left(\Sigma\left(\bar{P}\cdot\log \mathrm{OR}\right)\right)}{\left(\Sigma\left(\bar{P}\right)^2\right)\left(\Sigma\left(\Delta P\right)^2\right)-\left(\Sigma\left(\bar{P}\cdot\Delta P\right)\right)},\end{aligned} \tag{39}$$

where the summation is across all $N$ observations of $i, j, t$. If we assume that $\bar{P}$ and $\Delta P$ are uncorrelated—that is, $\mathrm{Cov}\left(\bar{P}, \Delta P\right) = \mathrm{E}\left[\bar{P}\cdot\Delta P\right] - \mathrm{E}\left(\bar{P}\right)\cdot\mathrm{E}\left(\Delta P\right) = \mathrm{E}\left[\bar{P}\cdot\Delta P\right] = 0$—then $\frac{1}{N}\Sigma\left(\bar{P}\cdot\Delta P\right) \to 0$ as $N \to \infty$. After replacing $\log \mathrm{OR}$ with $\alpha + \beta_1\bar{P} + \beta_2\Delta P + \varepsilon$, we have

$$\hat{\beta}_1, \hat{\beta}_2 \to \beta \text{ as } N \to \infty \tag{40}$$

We can also show this through an example. Suppose we have $N$ observations of $\log \mathrm{OR}$ and price for two products, with average prices $\bar{P}_1$ and $\bar{P}_2$, respectively, which are represented in Figure 11. When we estimate $\beta$ using price $P$, all the blue and yellow points are used. When we fix $\Delta P$ and estimate $\beta_1$ using average price $\bar{P}$ only, only the two big points are used. When we fix $\bar{P}$ and estimate $\beta_2$ using price variation $\Delta P$ only, $\Delta P$ for both products are pooled together. In all three cases, the slope does not change, so the estimated $\beta$, $\beta_1$, and $\beta_2$ will be the same. Suppose we have not only two products, but more products along the line; the same conclusion will still hold.

## B.6 Objective function of the Product2Vec model

The objective function of the Product2Vec model can be written as below:

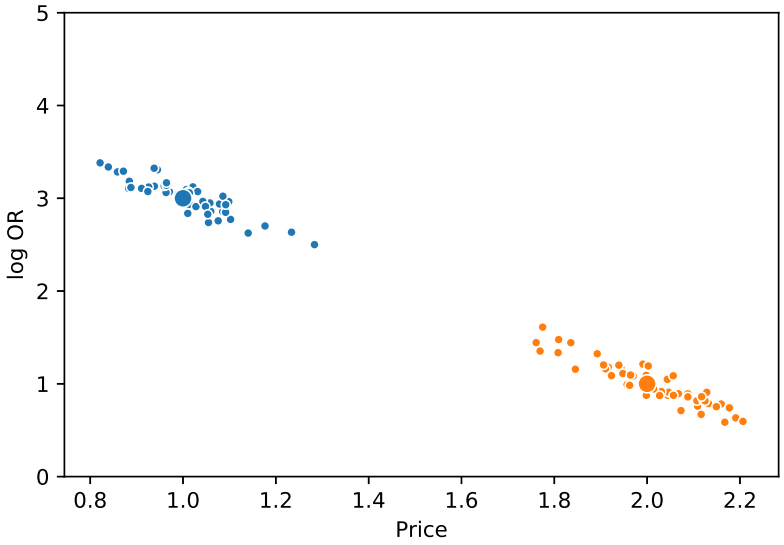



Figure 11: Debiasing Illustration

$$
\begin{aligned}
\mathscr{L}\mathscr{L}_{\text{Product2Vec}} &= \sum_{b\in B}\sum_{i\in b}\sum_{-c\le j\le c}\log\left[\Pr\left(s_{i+j}|s_i\right)\right] \\
&= \sum_{b_{it}\in B}\sum_{l\in b_{it}}\sum_{-c\le d_r-d_l\le c}\log\left[\Pr\left(y_{irt}|y_{ilt}\right)\right] \\
&= \sum_{b_{it}\in B}\sum_{l\in b_{it}}\sum_{-c\le d_r-d_l\le c}\log\sigma\left(v_{y_{ilt}}^{T}v_{y_{irt}}^{'}\right)+\sum_{g=1}^{G}\mathbb{E}_{y_g}\log\sigma\left(-v_{y_{ilt}}^{T}v_{y_{igt}}^{'}\right) \\
&= \sum_{i}\sum_{t}\frac{1}{2c}\left\{\sum_{l\in b_{it}}\sum_{-c\le d_r-d_l\le c}\log\sigma\left(v_{y_{ilt}}^{T}v_{y_{irt}}^{'}\right)+\sum_{g=1}^{G}\mathbb{E}_{y_g}\log\sigma\left(-v_{y_{ilt}}^{T}v_{y_{igt}}^{'}\right)\right\} \\
&\quad \text{(Individual level)} \\
&= \sum_{s}N_s\left[\sum_{t}\frac{1}{2c}\left\{\sum_{l\in b_{it}}\sum_{-c\le d_r-d_l\le c}\log\sigma\left(v_{y_{slt}}^{T}v_{y_{srt}}^{'}\right)+\sum_{g=1}^{G}\mathbb{E}_{y_g}\log\sigma\left(-v_{y_{slt}}^{T}v_{y_{sgt}}^{'}\right)\right\}\right] \\
&\quad \text{(Segment level)},
\end{aligned}
\tag{41}
$$

where $d_j$ is the location of product $j$ in basket $b$, $-c \le d_r - d_l \le c$ means that products $l$ and $r$ belong to the same basket; in other words, product $r$ is the context product of product $l$, and $c = d_b$ is the length of context window, that is, the number of context products in basket $b$.

We can see that product embeddings are purely determined by context products and negative samples; that is, if two products $j$ and $l$ have exactly the same context products and negative samples, they will have the same embeddings.

## B.7 Incorporating price endogeneity

### B.7.1 Change in data generating process

When we incorporate price endogeneity into the product utility, Equation 6 is changed to

$$u_{ijt} = \sum_{k=1}^{K} \alpha_{ik} A_{jk} + \delta_i Q_j + \beta_i P_{jt} + \xi_{jt} + \varepsilon_{ijt}, \tag{42}$$

where $\xi_{jt}$ is the unobserved demand shock and not independent of $P_{jt}$, which results in the price endogeneity issue.

Following Petrin and Train (2010), we use a control function approach to handle this. The model is estimated in two steps. First, the endogenous variable, price $P_{jt}$, is regressed on product attributes $A_{jk}$, quality $Q_j$, and the instrument variable $z_{jt}$. The residual of this regression $\mu_{jt}$ is retained.

$$P_{jt} = \lambda_z z_{jt} + \sum_{k=1}^{K} \lambda_k A_{jk} + \lambda_q Q_j + \mu_{jt} \tag{43}$$

Second, the choice model is estimated with the residual $\mu_{jt}$ entering as an extra variable.

$$u_{ijt} = \sum_{k=1}^{K} \alpha_{ik} A_{jk} + \delta_i Q_j + \beta_i P_{jt} + \rho_i \mu_{jt} + \varepsilon_{ijt} \tag{44}$$

### B.7.2 Change in the objective function of product fixed effect model

$$\begin{aligned}
&\mathscr{LL}_{\text{product}} \\
&= \frac{1}{2d_b} \sum_s N_s \left\{ \sum_t \sum_{l,r \in b_{it}} \log \left[ \frac{\exp(\gamma_{sl} + \beta_s \Delta P_{lt} + \xi_{lt})}{\sum_{l' \in c_l} \exp(\gamma_{sl'} + \beta_s \Delta P_{l't} + \xi_{l't})} * \frac{\exp(\gamma_{sr} + \beta_s \Delta P_{rt} + \xi_{rt})}{\sum_{r' \in c_r} \exp(\gamma_{sr'} + \beta_s \Delta P_{r't} + \xi_{r't})} \right] \right\} \\
&= \sum_{h=1}^{S*H} \log[\text{CP}_h] * \text{CPFreq}_h,
\end{aligned} \tag{45}$$

where $\text{CP}_h = \frac{\exp(\gamma_{sl}+\beta_s\Delta P_{lt}+\xi_{lt})}{\sum_{l'\in c_l}\exp(\gamma_{sl'}+\beta_s\Delta P_{l't}+\xi_{l't})} * \frac{\exp(\gamma_{sr}+\beta_s\Delta P_{rt}+\xi_{rt})}{\sum_{r'\in c_r}\exp(\gamma_{sr'}+\beta_s\Delta P_{r't}+\xi_{r't})}$ is a unique value of copurchase probability and $\text{CPFreq}_h = \frac{1}{2d_b}\sum_s N_s \sum_t \sum_{l,r\in b_{it}} \mathbb{1}\left(\Pr\left(y_{ilt}\right) * \Pr\left(y_{irt}\right) = \text{CP}_h\right)$.

Since $\xi_{jt}$ and $\Delta P_{jt}$ are both product- and trip-specific, if $\Delta P_{jt}, \Delta P_{lt} \sim N(0,\sigma_1^2), \xi_{jt}, \xi_{lt} \sim N(0,\sigma_2^2)$, and $\Delta P, \xi$ are independent, then $\beta_i\Delta P_{jt}+\xi_{jt}, \beta_i\Delta P_{lt}+\xi_{lt} \sim N(0,\beta_i^2\sigma_1^2+\sigma_2^2)$, and we can use the same logic as in Section 5.1.2.1 and have the lemma below:

**Lemma 2.** *If two products in the same category share the same attribute combinations, their price variations follow the same normal distribution with mean 0 and standard deviation $\sigma_1$, and their unobserved demand shocks also follow the same normal distribution with mean 0 and standard deviation $\sigma_2$; then they will have the same purchase probability when the number of time periods goes to infinity.*

$$\begin{gathered}\lim_{t\to\infty}\Pr_{ij} = \lim_{t\to\infty}\sum_t \Pr_{ijt} = \lim_{t\to\infty}\sum_t \Pr_{ilt} = \lim_{t\to\infty}\Pr_{il} \\ \textit{when}\ \ c_j = c_l, \\ \overrightarrow{A_j} = \overrightarrow{A_l}, \\ \Delta P_{jt}, \Delta P_{lt} \sim N(0,\sigma_1^2), \\ \xi_{jt}, \xi_{lt} \sim N(0,\sigma_2^2)\end{gathered} \tag{46}$$

Thus, when we incorporate price endogeneity, the number of unique copurchase patterns for all segments is equal to the number of unique individual-specific time-invariant product utilities; in other words, Proposition 2 still holds.

### B.7.3 Change in the objective function of Product2Vec model

After adding $\xi_{jt}$ to the utility function, the objective function of Product2Vec is not influenced, so Proposition 4 still holds and Proposition 5 follows. So, after incorporating price endogeneity into the product utility, we still have that the number of individual-specific time-invariant product utilities is equal to the number of consumer segments multiplied by the number of product clusters generated by product embeddings.

## B.8 Notations

For ease of reference, Table 13 shows a complete list of the definitions of notations used in Sections 5.1.

Table 13: Notations

| Symbol | Definition | Symbol | Definition |
|---|---|---|---|
| $I$ | Number of consumers | $i$ | Consumer |
| $K$ | Number of attributes | $k$ | Attribute |
| $A_k$ | Value for attribute $k$ | $\|\|A_k\|\|$ | Number of distinct values for attribute $k$ |
| $\|\|A\|\|$ | Total number of attribute combinations | | |
| $J$ | Number of products | $j$ | Product |
| $J_c$ | Number of products in category $c$ | | |
| $C$ | Number of categories | $c$ | Category |
| $c_j$ | Category that product $j$ belongs to | | |
| $M$ | Number of embedding clusters | $m$ | Cluster |
| $H$ | Number of unique copurchase probabilities | $h$ | Copurchase probability |
| $G$ | Number of negative samples | $g$ | Negative sample |
| $S$ | Number of segments | $s$ | Segment |
| $N_s$ | Size of segment $s$ | | |
| $B$ | Set of baskets | $b$ | Basket |
| $\gamma_{ij}$ | Individual-specific time-invariant product utility | $\|\|\gamma\|\|$ | Number of unique $\gamma_{ij}$ |
| $\Lambda$ | Consumer differentiable attribute combinations | | |

# C Additional simulation results

## C.1 Product maps, clusters, and choice model estimates with higher number of products

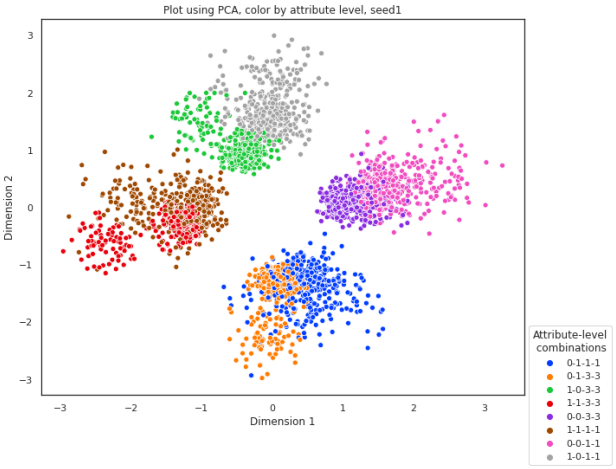
To analyze the scalability of our proposed approach, we simulate two additional scenarios with 300 products per category (3,000 in total) and 500 products per category (5,000 in total). We use the same number of categories, $\Omega$ matrix, attribute levels, and customer preferences as the ones used in the main manuscript (please refer to Section 7). To take into account the randomness of the training process, for each case we estimate the product embeddings and their corresponding product maps and product clusters using 10 different seeds.

**300 products** In Figure 12, we present the product maps resulting from the 300 products case. Similar to the results presented in Section 7, we observe that products with similar attribute-level combinations are closer in the embedding space. Different from the results presented in Section 7, we observe that the map obtained with TSNE for dimensionality reduction captures a mix of category structure and attribute-level combinations. Nevertheless, we find that the optimal number of clusters is 8, and each cluster groups products with one and only one attribute-level combination. These results are consistent across the 10 different seeds.

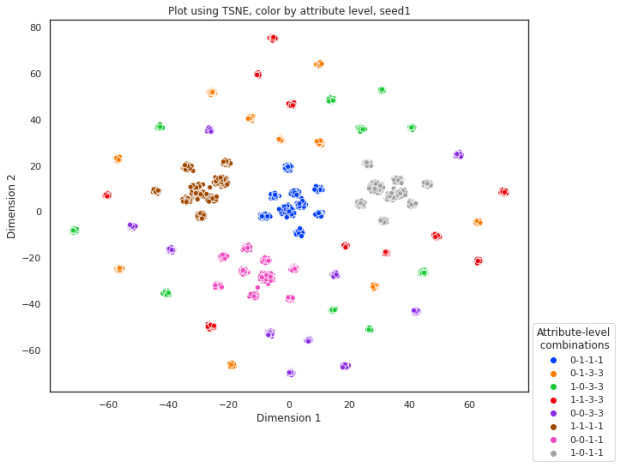
In Table 14, we illustrate the results of the different choice model specifications described in Section 7. Similar to the findings for the 100 product case used in the main manuscript, we observe that the cluster model allow us to obtain similar results to those obtained by the true model.[18]

**500 products** In Figure 13, we present the product maps resulting from the 500 products case. We generally observe very similar patterns to those obtained for the 300 product case. Moreover,

[18] Given that the running time for the product fixed-effects model was already above 18 hours for the 100 products case, we did not estimate this model specification for the 300 products and the 500 products case.



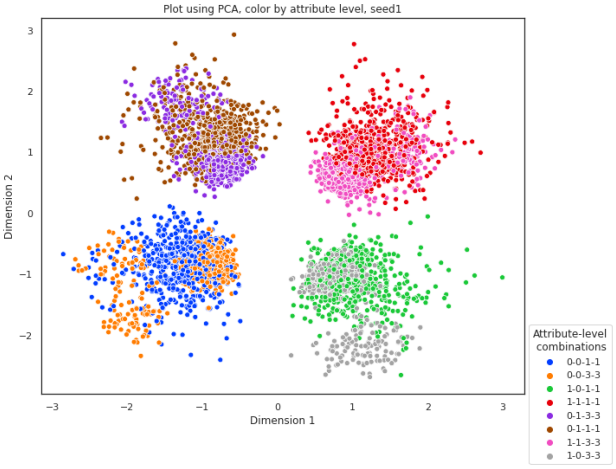

(a)



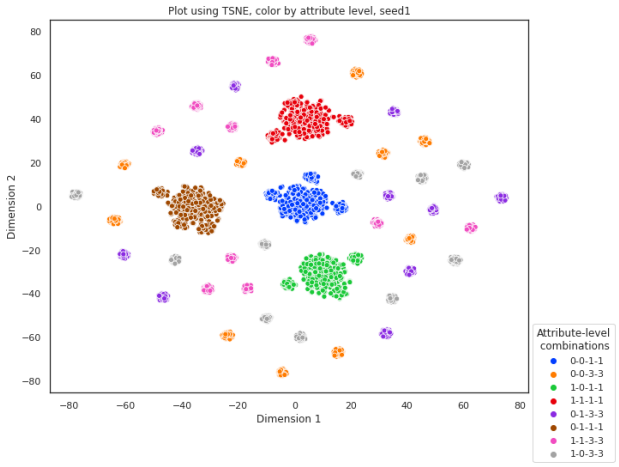

(b)

Figure 12: Panel (a) shows the product map based on PCA mapping of the product embeddings obtained with seed 10, and Panel (b) shows the product map based on t-SNE mapping of the product embeddings obtained with seed 10. Product maps do not change significantly across seeds.

we find that the optimal number of clusters is 8, and each cluster groups products with one and only one attribute-level combination. These results are consistent across the 10 different seeds.




(a)




(b)

Figure 13: Panel (a) shows the product map based on PCA mapping of the product embeddings obtained with seed 10, and Panel (b) shows the product map based on t-SNE mapping of the product embeddings obtained with seed 10. Product maps do not change significantly across seeds.

In Table 15, we illustrate the results of the different choice model specifications described in Section 7. Similar to the other cases, we observe that the cluster model allow us to obtain similar results to those obtained by the true model.

Table 14: Price estimates for 300 simulated products

| | (1) True Attributes | (2) Missing Q | (3) Missing A | (4) Embedding Clusters |
|---|---|---|---|---|
| Price Segment 1 | −5.915*** | −6.070*** | −5.051*** | −5.996*** |
| | (0.099) | (0.102) | (0.055) | (0.101) |
| Price Segment 2 | −6.136*** | −6.144*** | −4.741*** | −6.153*** |
| | (0.101) | (0.101) | (0.052) | (0.102) |
| Price Segment 3 | −5.995*** | −5.998*** | −3.043*** | −5.621*** |
| | (0.100) | (0.100) | (0.046) | (0.094) |
| Price Segment 4 | −5.911*** | −5.909*** | −2.658*** | −5.541*** |
| | (0.095) | (0.095) | (0.042) | (0.089) |
| Price Segment 5 | −3.127*** | 0.165*** | | −3.106*** |
| | (0.067) | (0.017) | | (0.066) |
| Price Segment 6 | −2.997*** | 0.630*** | | −2.960*** |
| | (0.065) | (0.021) | | (0.065) |
| Price Segment 7 | −2.922*** | 0.987*** | | −2.872*** |
| | (0.061) | (0.026) | | (0.060) |
| Price Segment 8 | −2.938*** | 1.317*** | | −2.714*** |
| | (0.065) | (0.037) | | (0.064) |
| Loglike | -49,852 | -60,878 | -63,076 | -50,055 |
| AIC | 99,798 | 121,834 | 126,190 | 100,267 |
| BIC | 100,167 | 122,139 | 126,339 | 100,886 |
| In-sample HR | 0.296 | 0.221 | 0.184 | 0.273 |
| Out-of-sample HR | 0.322 | 0.245 | 0.192 | 0.302 |
| Execution time (hh:mm:ss) | 34 min 37 sec | 26 min 25 sec | 6 min 2 sec | 39 min 36 sec |

*Significance levels:* * p<0.1, ** p<0.05, *** p<0.01.

*Note:* Models in column 1, 2, and 3 use log_unit_price. Models in column 4 and 4 use delta_log_unit_price.

## C.2 Examining the role of different hyperparameters

To analyze how some of the main hyperparameters of word2vec impact our findings, we use different values to train the product embeddings and monitor four main outcomes: (i) the loss function, (ii) the embeddings' ability to recover category level complementarity—in our case, given by the $\Omega$ matrix, (iii) the relationship between the product in the embedding space (measured as the pairwise cosine similarity and Euclidean distance), and (iv) the ability to recover the optimal number of product clusters.

Table 15: Price estimates for 500 simulated products

| | (1) True Attributes | (2) Missing Q | (3) Missing A | (4) Embedding Clusters |
|---|---|---|---|---|
| Price Segment 1 | $-6.136^{***}$ | $-5.765^{***}$ | $-5.305^{***}$ | $-6.071^{***}$ |
| | (0.09) | (0.087) | (0.054) | (0.085) |
| Price Segment 2 | $-6.122^{***}$ | $-6.039^{***}$ | $-5.176^{***}$ | $-6.122^{***}$ |
| | (0.094) | (0.093) | (0.05) | (0.097) |
| Price Segment 3 | $-6.008^{***}$ | $-6.048^{***}$ | $-3.067^{***}$ | $-6.095^{***}$ |
| | (0.084) | (0.085) | (0.045) | (0.09) |
| Price Segment 4 | $-5.961^{***}$ | $-6.374^{***}$ | $-2.928^{***}$ | $-6.043^{***}$ |
| | (0.094) | (0.101) | (0.041) | (0.093) |
| Price Segment 5 | $-3.109^{***}$ | $9.856^{***}$ | | $-3.167^{***}$ |
| | (0.070) | (0.150) | | (0.061) |
| Price Segment 6 | $-3.105^{***}$ | $4.688^{***}$ | | $-3.165^{***}$ |
| | (0.061) | (0.071) | | (0.061) |
| Price Segment 7 | $-2.904$ | $8.100^{***}$ | | $-3.065^{***}$ |
| | (0.061) | (0.119) | | (0.070) |
| Price Segment 8 | $-3.028^{***}$ | $6.176^{***}$ | | $-2.915^{***}$ |
| | (0.061) | (0.091) | | (0.061) |
| Loglike | -57,290 | -114,558 | -70,295 | -57,805 |
| AIC | 114,675 | 229,194 | 140,627 | 115,769 |
| BIC | 115,043 | 229,500 | 140,776 | 116,388 |
| In-sample HR | 0.254 | 0.192 | 0.155 | 0.233 |
| Out-of-sample HR | 0.293 | 0.229 | 0.194 | 0.274 |
| Execution time (hh:mm:ss) | 55 min 10 sec | 34 min 50 sec | 11 min 53 sec | 1 hr 23 min 16 sec |

*Significance levels:* * p<0.1, ** p<0.05, *** p<0.01.

*Note:* Models in column 1, 2, and 3 use log_unit_price. Models in column 4 and 4 use delta_log_unit_price.

**100 products** In Figure 19, we illustrate the loss function obtained under different specifications of the hyperparameters. We observe that the loss function is generally stable after 50 or 100 iterations. We also note that, among all the hyperparameters we explore, the loss function seems more sensitive to the number of negative samples.

In Figure 20, we illustrate the category-level co-occurrence score obtained under different specifications of the hyperparameters. We observe that the category-level co-occurrence score is generally stable across different hyperparameters but sensitive to the downsampling threshold.

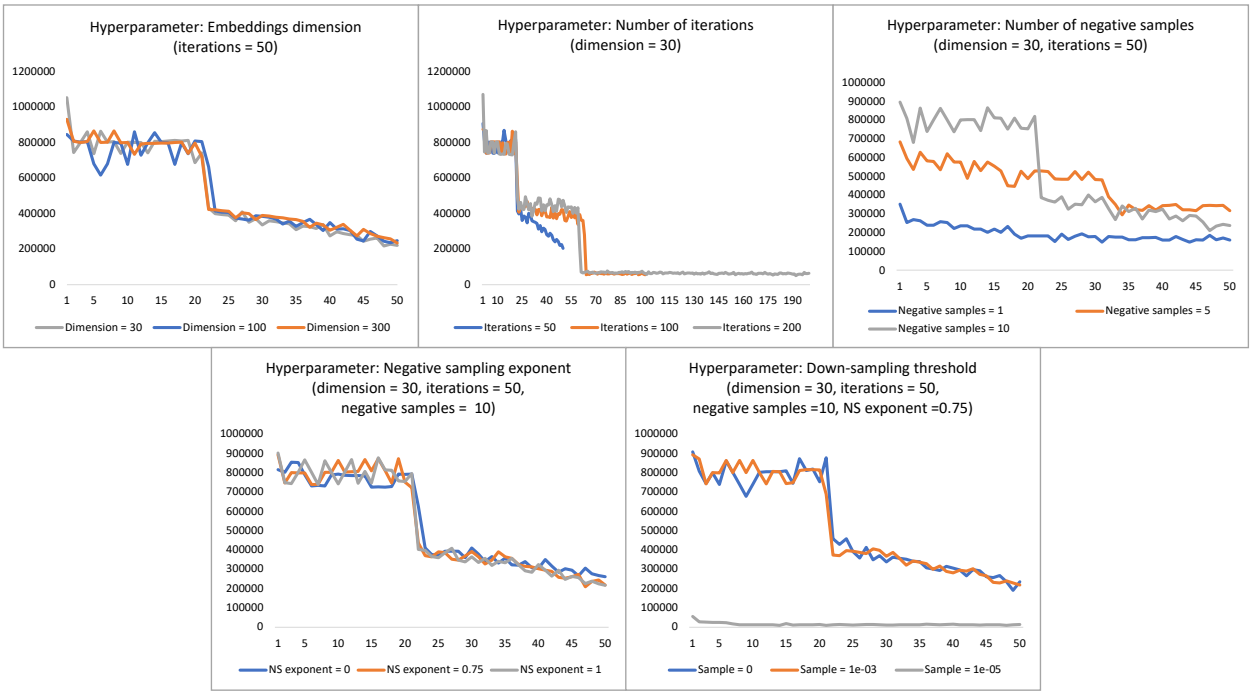



Figure 14: Loss function (y-axis) over iteration number (x-axis) obtained with different hyperparameter values for simulation with 300 products per category.

In terms of the pairwise cosine similarities and Euclidean distances, we find that the values obtained using different hyperparameters are highly correlated. The minimum observed correlation was obtained when varying the downsampling threshold parameter. More specifically, the correlation between the pairwise cosine similarities obtained with a downsampling threshold equal to 1e-5 and 0 is 0.90. Similarly, the correlation between the pairwise Euclidean distances obtained with a downsampling threshold equal to 1e-5 and 0 is 0.85. For all the other hyperparameters, such correlations were above 0.90.

Finally, the optimal number of product clusters is eight for all the hyperparameter values tested. As expected, each cluster groups products with the same simulated attributes in every case.

**300 products** In Figure 16, we illustrate the loss function obtained under different specifications of the hyperparameters. We observe that the loss function is generally stable after 50 or 100 iterations. We also note that, among all the hyperparameters we explore, the loss function seems more sensitive to the number of negative samples.

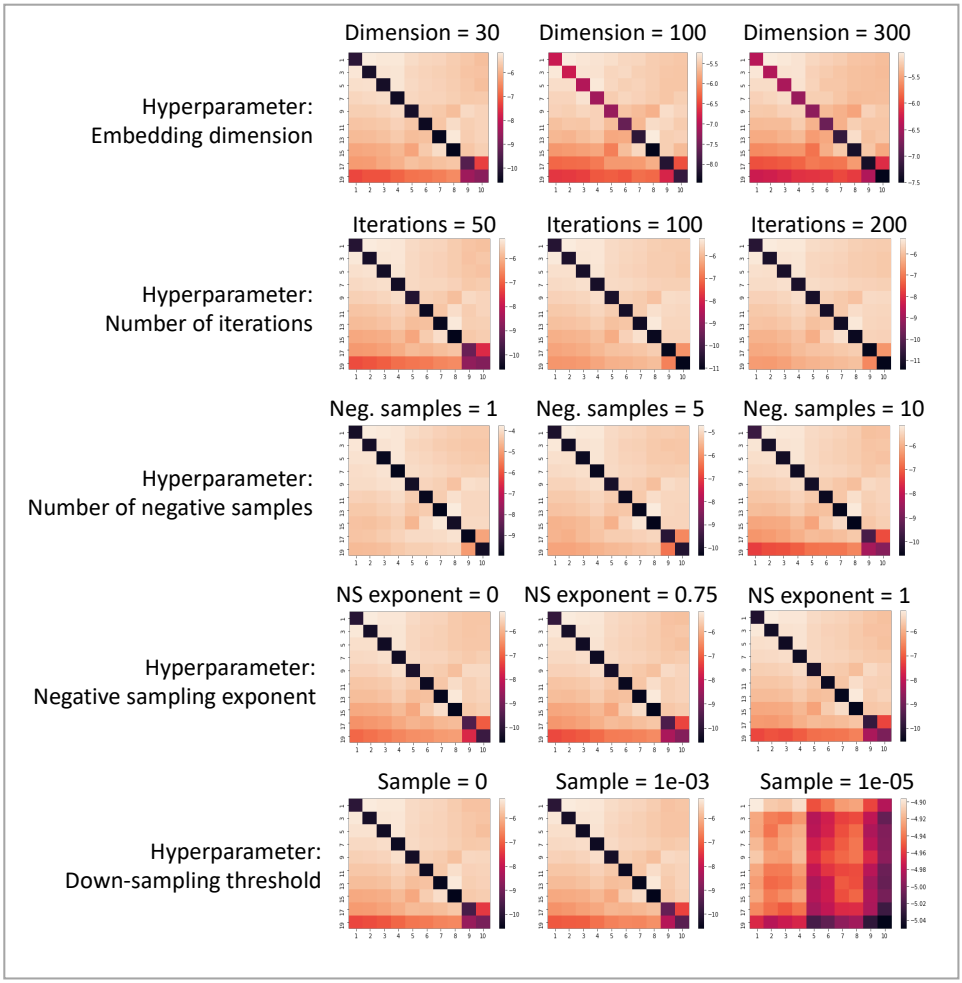



Figure 15: Co-occurrence matrix for embeddings obtained with hyperparameter values for simulation with 300 products per category. Co-occurrence is computed as the dot product between the input and output category vectors, defined as the average of the embeddings for products in the same category.

.

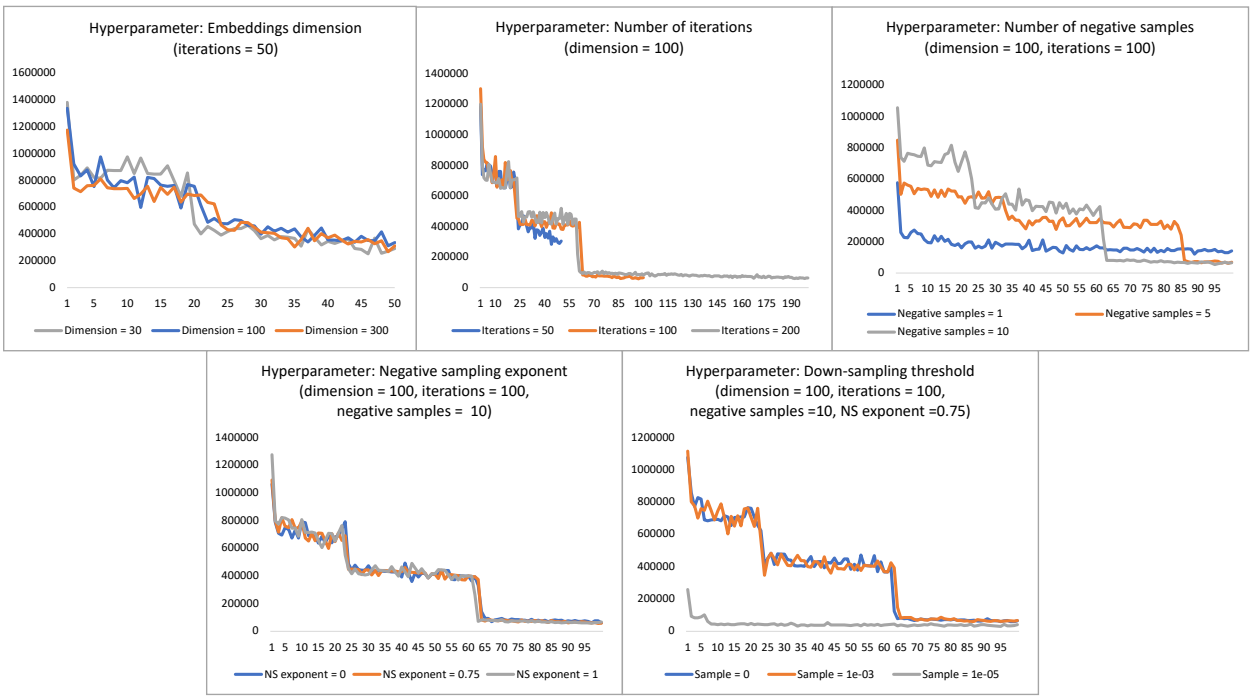



Figure 16: Loss function (y-axis) over iteration number (x-axis) obtained with different hyperparameter values for simulation with 300 products per category.

In Figure 20, we illustrate the category-level co-occurrence score obtained under different specifications of the hyperparameters. We observe that the category-level co-occurrence score is generally stable across different hyperparameters but sensitive to the downsampling threshold. Moreover, in this setting with a higher number of products per category, increasing the dimension of the embeddings to at least 100 seems beneficial.

In terms of the pairwise cosine similarities and Euclidean distances, we find that the values obtained using different hyperparameters are highly correlated. The minimum observed correlation was obtained when varying the dimension of the embeddings. More specifically, the correlation between the pairwise cosine similarities obtained with embeddings of dimension 30 and embeddings of dimension 300 is 0.84. For all the other hyperparameters, such correlations were above 0.90. Finally, based on the Silhouette score, the optimal number of product clusters is eight for all the hyperparameter values tested. Based on the CH index, the optimal number of product clusters is eight for most of the hyperparameter values tested, but two in a few cases such as the one illus-

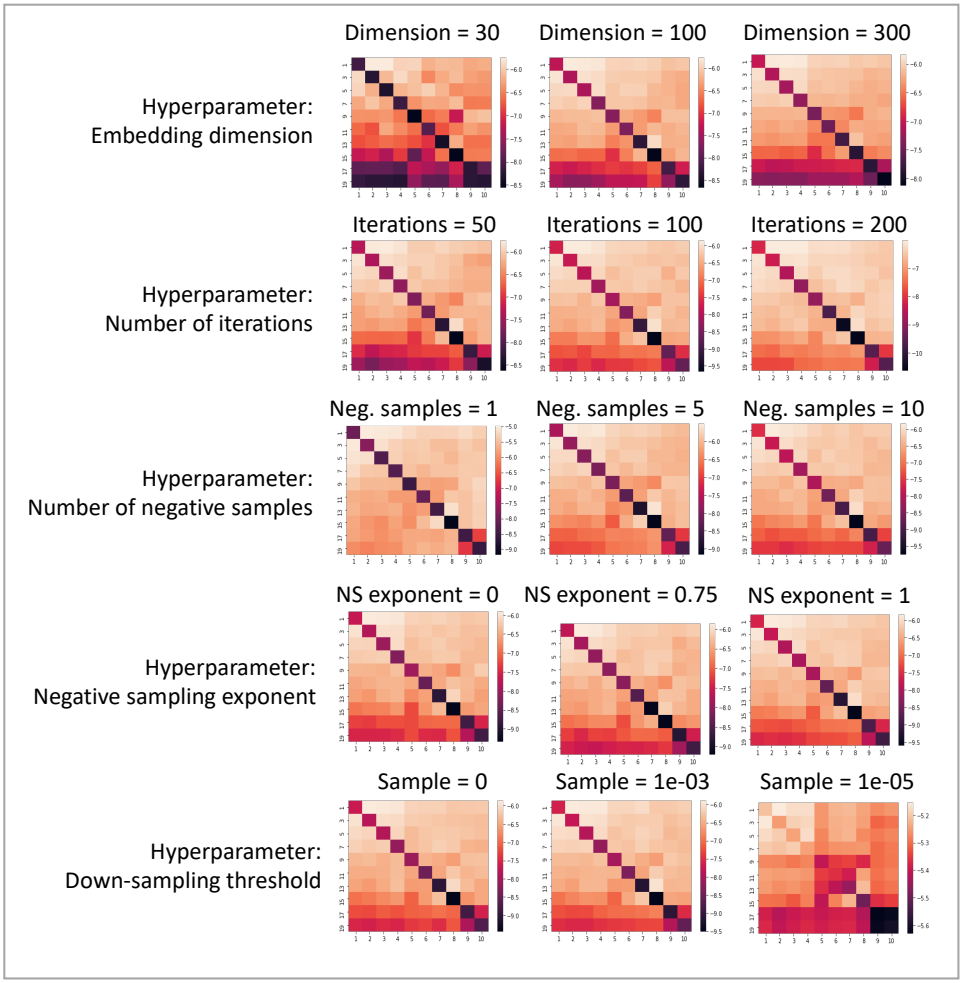



Figure 17: Co-occurrence matrix for embeddings obtained with hyperparameter values for simulation with 300 products per category. Co-occurrence is computed as the dot product between the input and output category vectors, defined as the average of the embeddings for products in the same category.

.

trated in Figure 18. As expected, when the optimal number of clusters is eight, each cluster groups products with the same simulated attributes in every case.

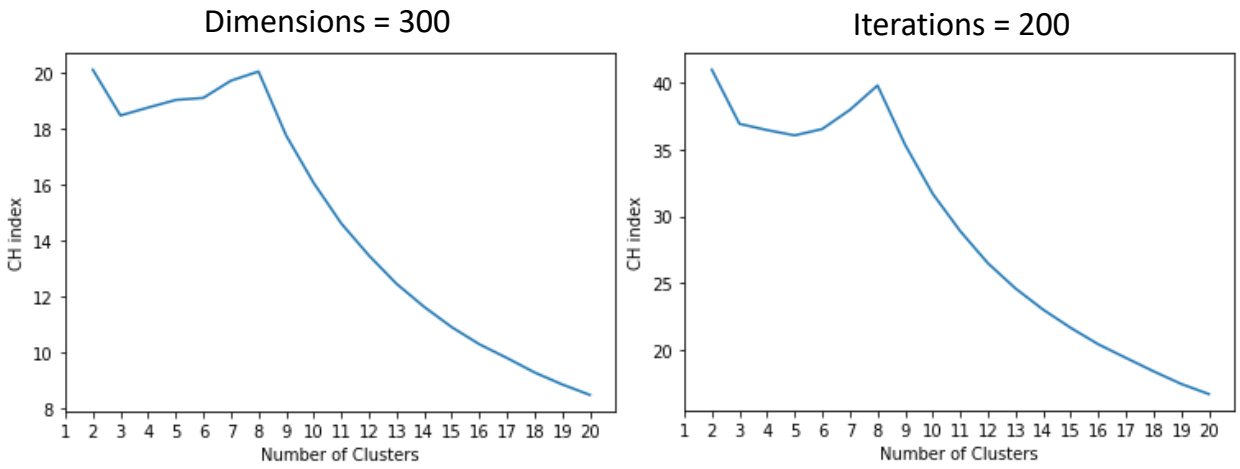



Figure 18: The figure on the left-panel illustrates the CH index as a function of the number of clusters obtained with embeddings of dimension 300 after 500 iterations. The figure on the right-panel illustrates the CH index as a function of the number of clusters obtained with embeddings of dimension 100 after 200 iterations. Note that, despite the fact that the CH index achieves its global maximum when the number of clusters is two, it also achieves a local maximum when the number of clusters is eight.

**500 products** In Figure 19, we illustrate the loss function obtained under different specifications of the hyperparameters. We observe that the loss function is generally stable after 50 or 100 iterations. We also note that, among all the hyperparameters we explore, the loss function seems more sensitive to the number of negative samples.

In Figure 20, we illustrate the category-level co-occurrence score obtained under different specifications of the hyperparameters. We observe that the category-level co-occurrence score is generally stable across different hyperparameters but sensitive to the downsampling threshold. And as in the previous case with 300 products per category, increasing the embeddings dimensions also helps to obtain a more accurate representation of the category-level co-occurrence.

In terms of the pairwise cosine similarities and Euclidean distances, we find that the values obtained using different hyperparameters are highly correlated. The minimum observed correlation was obtained when varying the dimension of the embeddings. More specifically, the correlation

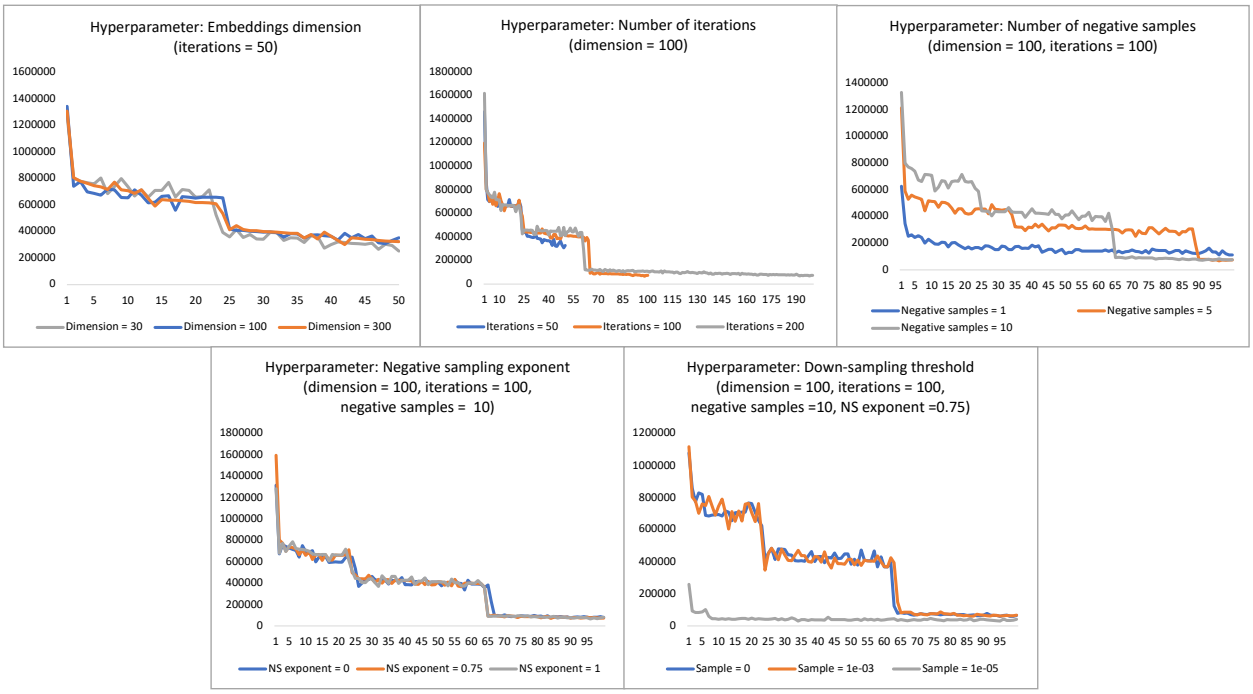



Figure 19: Loss function (y-axis) over iteration number (x-axis) obtained with different hyperparameter values for simulation with 500 products per category.

between the pairwise cosine similarities obtained with embeddings of dimension 30 and embeddings of dimension 300 is 0.81. For all the other hyperparameters, such correlations were above 0.88.

Finally, based on the Silhouette score, the optimal number of product clusters is eight for all the hyperparameter values tested. Based on the CH index, the optimal number of product clusters is eight for most of the hyperparameter values tested, but two in a few cases (similar to the situation with 300 products). As expected, when the optimal number of clusters is eight, each cluster groups products with the same simulated attributes in every case.

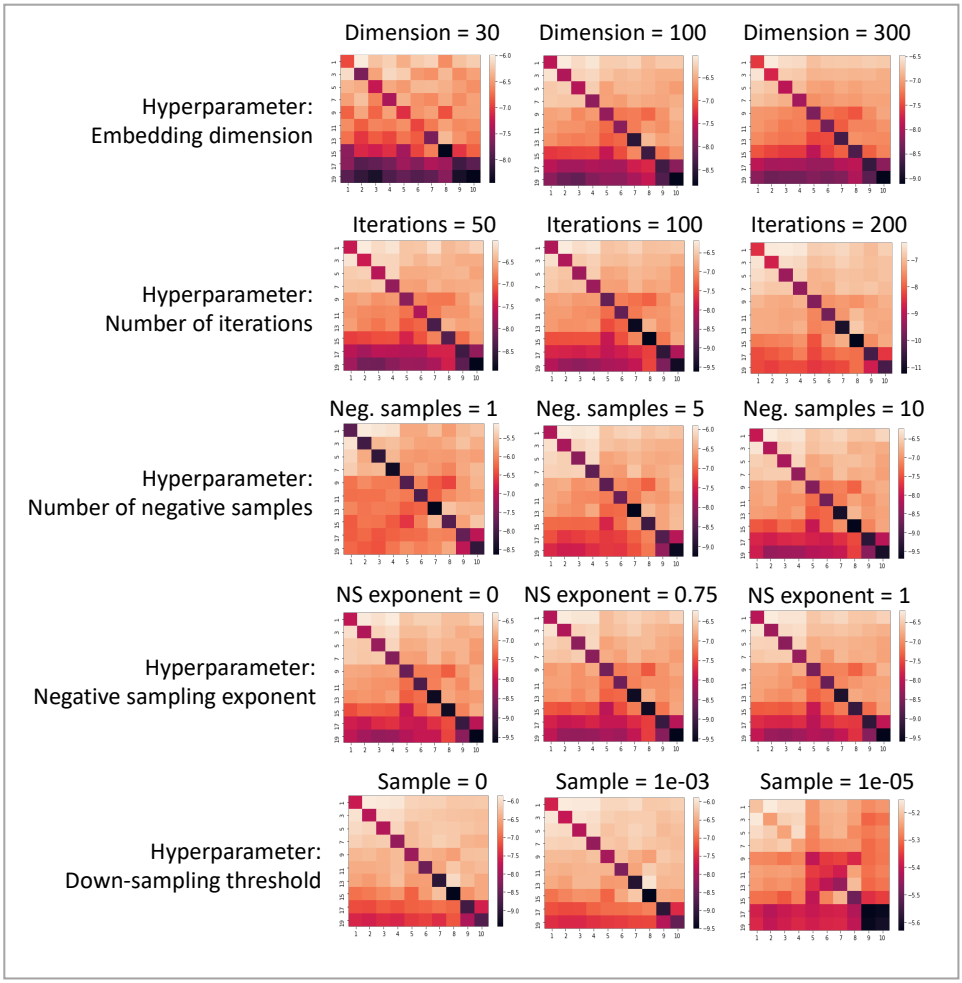



Figure 20: Co-occurrence matrix for embeddings obtained with hyperparameter values for simulation with 500 products per category. Co-occurrence is computed as the dot product between the input and output category vectors, defined as the average of the embeddings for products in the same category.

.

# D Product category selection

We choose the carbonated beverages category to estimate choice models, for two reasons.

First of all, we filter products and shopping baskets in several steps to get the estimation and test sets, and the carbonated beverages category has the most number of products after this process. Specifically, we (1) keep products that have embeddings, that is, with at least 160 occurrences in the training set, (2) keep products that appear at least once in the chosen retailer and state, to make sure they are available to consumers, and (3) keep shopping baskets that contain one and only one of such products due to the requirement of discrete choice models.

Second, we use the carbonated beverages category because the number of clusters is reasonable and each cluster is interpretable, as shown in Figure 9 and Table 9.

# E Interpretation of Nielsen UPC descriptions

The table below provides the mapping between the original UPC description in the NielsenIQ data and our interepretaion, for the products used in Table 10.

# F First stage estimates

In the first stage of the control function approach, we regress the endogenous variable, log_unit_price (or delta_log_unit_price), on the price instrument as well as other exogenous variables in the choice models. We report these results in Table 17. Across all models, the coefficients of the price instrument are significantly positive and the F-statistics are very large, suggesting that the relevance condition for the price instrument is satisfied.

Table 16: Interpreting Nielsen UPC description

| Original UPC description | Interpretation |
|---|---|
| CTL BR BRD WHE BTR ST F | PL Bread Wheat Split Top 20 OZ |
| CTL BR BRD WHE SND NC F | PL Bread Wheat No Cholesterol 20OZ |
| CTL BR BRD W-HNY F | PL Bread Wheat 20OZ |
| CTL BR BRD O-B F | PL Bread Oat Bran 20OZ |
| DK CH MINT BAR | Dark Chocolate Mint Bar 4.4OZ |
| Q-V LTC ROM HRT GRN F | Fresh Lettuce Roma Heart |
| Q QK OT RG | Quaker Quick Oats Regular 18OZ |
| NOMS SND CH | Oreo Mega Stuf Sandwich Cookies 13.2OZ |
| TMS BG EVRYTHNG NY F | Thomas Bagel Everything NY Fresh 22OZ |
| SOH PTZL ROD BG | Snyder's of Hanover Salted Pretzel 12OZ |
| SNSHN CZ-I CHS W-CHD | Sunshine Cheez-IT Salted Crackers 12.4 OZ |
| RG PTZL TWT TY BG | Rold Gold Pretzel Tiny 16OZ |
| ST-FR LQ CF MC B 15P | Starbucks Frappuccino Liquid Coffee Mocha Bottle 15P |
| NTC FRZ DRD RH RST | Nescafe Taster's Choice House Blend 12OZ |
| FLGR SPR DRD CLRST | Folgers Classic Roast 16OZ |
| ALEVE NX-SDM RS CPT | Aleve Naproxen Sodium Regular Strength Caplet |
| PF REM MILANO TAMPER SEAL 20P | Pepperidge Farm Milano Tamper Seal 20P 0.75OZ |
| NW-FRM FR BN ORG WH F | New World Farms Fresh Beans Organic |
| FLA-N GS OJ NC PU U M-P C R | Florida Natural Orange Juice Carton 59OZ |
| SY PT CP SNKD B BG | Simply Naked Pita Chips |
| S-O OJ 1%SNC U EC/C PF PL R | Simply Orange Orange Juice Pulp Free Plastic |
| TR PP OJ NVC U NP PL R | Tropicana Orange Juice Plastic |
| NBC OREO SND GLDN | Oreo Sandwich Cookies Golden |
| MNT-P-FRS SF C/G FM PC | Mentos Pure Fresh Sugar Free Gum Fresh Mint |
| CTL BR M 1% LF VAD PL F | PL Milk 1% Low Fat Vitamin A/D Plastic 128OZ |
| CTL BR M 2%RF VAD PL F | PL Milk 2% RD Fat Vitamin A/D Plastic 64OZ |
| CTL BR M 2%RF VAD PL F | PL Milk 2% RD Fat Vitamin A/D Plastic 128OZ |
| CTL BR M WH V-D PL F | PL Milk Whole Vitamin A/D Plastic 64OZ |
| CTL BR EGGS AA LG NGH | PL Large White Eggs 12Q |
| GOLD PEAK LT SWT PL | Gold Peak Liquid Tea Sweet Plastic 89OZ |
| M HSE AP BFT BLN L-R J | Maxwell House Coffee Breakfast Blend Light Roast Jar 25.6OZ |
| HRS CHOC SYP SQZ B | Hershey's Syrup Chocolate Squeeze Bottle 24OZ |
| HFD RS BF LO B/CC DTB R | Hillshire Farm Deli Select 7OZ |
| HFD PK SA HRD DTB R | Hillshire Farm Deli Select 7OZ |
| FGR AP MED DR 100%CL M-GR CNS | Folgers Coffee Med-Dark Roast 100% Colombian 24.2OZ |
| CTL BR PR AM IM SL CH | PL Processed American Sliced Cheese 10.67OZ |
| CTL BR | PL Crackers Flaked Soda Salted 16OZ |
| CTL BR BLG CK&PK SL W R | PL Crackers Flaked Soda Salted 16OZ |
| CTL BR NF GK FOB Y ST | PL Non-Fat Greek Yogurt 5.3OZ |
| CTL BR BST TRK B-F SL W R | PL Breast Turkey Black Forest Sliced 16OZ |
| YP WHPS Y VC MSS WHP | Yoplait Whips! Vanilla Cream Mousse 4OZ |
| 9-L WT S-S 4P | 9 Lives Cat Food Wet Type 5.5OZ |
| CTL BR M 2%RF VAD PL F | PL Milk 2% RD Fat Vitamin A/D Plastic 64OZ |
| PCCN D CKN SM | Purina Cat Chow Naturals 18PO |
| Q OT RG O-F | Quaker Oats Regular Old Fashioned 18OZ |
| CLMB BCN UNC H-H-S TRK TS | Columbus Bacon Hickory-Hardwood-Smoked Thigh Sliced Turkey 40OZ |
| HRSH KIS CH | Hershey's Kisses Chocolate 56OZ |
| KNG B-B CL/AST-F 48CT | Kellogg's Breakfast Bars Assorted Flavors 62.4OZ |
| QKR OM SQ RTE 2'S | Quaker Oatmeal Squares 58OZ |
| NNQ SM 1% L/F V-AD CH PL 15P | Nestle Nesquik Skim Milk 1% Low Fat Vitamin A/D Chocolate Plastic 15P 8OZ |
| CTL BR MSHRM WH F | PL Fresh Mushrooms |
| CTL BR BRD W-W PREM RCP F | PL Bread Whole Wheat Premium |
| FLA-N OJ NC U M-P C | Florida's Natural Orange Juice |
| E-S S-F VG MX F | Eat Smart Stir Fry Vegetable Mix 12OZ |
| VFRM TOMATOES HVSM GH F | Village Farms Tomatoes 10OZ |
| CTL BR LF FT Y BB | PL Low Fat Yogurt Blackberry 6OZ |
| CTL BR LG/MT SC | PL Frozen Lasagna 32OZ |
| P-E-I-B PEACH CA BAG F | Peach Bag 2PO |
| CTL BR GB CHCVCW CHCP 6CT | PL Granola Bar Chocolatey Covered Chewy 6.5OZ |
| CTL BR AM DR IC | PL Ground and Whole Bean Coffee Ind. Cups 12Q |
| GOLD PEAK LT RS PL | Gold Peak Liquid Tea Raspberry Plastic 52OZ |
| GOLD PEAK LT LMD PL | Gold Peak Liquid Tea Lemonade Plastic 52OZ |
| CTL BR M 2%RF VAD PL F | PL Milk 2% RD Fat Vitamin A/D Plastic 128OZ |
| BIMBO BRD WHI LG NC F | Bimbo Bread White 24OZ |
| GOLD PEAK LT SF UN PL | Gold Peak Liquid Tea Unflavored Plastic 89OZ |

Table 17: First stage results for the carbonated beverages category

| | (1) Embedding Cluster | (2) Product FE | (3) Attribute | (4) Missing Attribute |
|---|---|---|---|---|
| Price | $0.602^{***}$ | $0.602^{***}$ | $0.909^{***}$ | $0.951^{***}$ |
| | (0.002) | (0.002) | (0.001) | (0.001) |
| Intercept | 0.000 | $-0.003$ | $-0.340^{***}$ | $-0.187^{***}$ |
| | (0.001) | (0.006) | (0.011) | (0.002) |
| N | 356760 | 356760 | 356760 | 356760 |
| $R^2$ | 0.176 | 0.176 | 0.874 | 0.872 |
| F | 8488.068 | 424.348 | $1.078e+05$ | $2.426e+06$ |
| Number of parameters | 10 | 181 | 24 | 2 |

$^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$

Note: The Embedding Cluster and Product FE models use delta_log_unit_price, and the Attributes and Missing Attributes models use log_unit_price.