Demand Estimation with Text and Image Data

Giovanni Compiani, Ilya Morozov, Stephan Seiler

CESifo

Demand Estimation with Text and Image Data

Abstract

We propose a demand estimation method that allows researchers to estimate substitution patterns from unstructured image and text data. We first employ a series of machine learning models to measure product similarity from products' images and textual descriptions. We then estimate a nested logit model with product-pair specific nesting parameters that depend on the image and text similarities between products. Our framework does not require collecting product attributes for each category and can capture product similarity along dimensions that are hard to account for with observed attributes. We apply our method to a dataset describing the behavior of Amazon shoppers across several categories and show that incorporating texts and images in demand estimation helps us recover a flexible cross-price elasticity matrix.

JEL-Codes: C100, C500, C810.

Keywords: demand estimation, unstructured data, computer vision, text models.

Giovanni Compiani
University of Chicago / IL / USA
gio1compiani@gmail.com

Ilya Morozov
Northwestern University, Evanston / IL / USA
ilya.morozov@kellogg.northwestern.edu

Stephan Seiler
Imperial College London / United Kingdom
stephan.a.seiler@gmail.com

1 Introduction

Many marketing problems such as price setting or assortment choice require firms to estimate demand for their goods. The existing work on demand estimation mostly relies on choice models that capture substitution through the similarity of product attributes. This approach requires the researcher to choose which product attributes to include when modeling demand in a particular product category and to gather data on those attributes. In this paper, we propose a novel method to estimate demand that leverages texts and images instead of observed attributes. Texts and images can provide rich information about which products consumers likely perceive to be close substitutes. Consumer reviews, for example, can reveal whether consumers talk about two products in a similar way, and product images can reveal which products are visually similar. Using these unstructured data makes the estimation approach scalable because product descriptions, customer reviews, and images are readily available in the same format across categories. Our method also allows us to take into account product attributes that are typically unobserved or hard to quantify, such as products' aesthetics, which can nevertheless be captured by product images. We show that this demand estimation approach allows us to estimate flexible substitution patterns in many product categories at a low computational cost.

To harness information from unstructured data, we first employ a series of machine learning models to calculate pairwise measures of product similarity. We measure image similarity based on several pre-trained deep learning models, borrowed from the computer vision literature, that have been shown to perform well in image classification tasks. Using these models, we translate images into low-dimensional embeddings and compute similarity between products by calculating the distance between these products' vector representations. Similarly, we employ a series of text-based models to calculate similarity based on product titles, product descriptions, product Q&As, and customer reviews. We use several models, from simple bag-of-words classifiers to more advanced BERT-based sentence classifiers that account for the context in which specific words appear. We calculate similarity from a wide range of text and image models because it is not a priori clear which similarity measures perform best at explaining substitution in a given setting.

Having constructed similarity measures, we incorporate them into a demand model and let them inform substitution patterns. Specifically, we employ a paired combinatorial nested logit model (Small, 1987; Koppelman and Wen, 2000) with an overlapping nest structure, which includes a separate nest for each product pair. We parameterize the error correlation for a given nest as a function of the image and text similarities between the two products in this nest. Thus, our model allows similarity measures to influence error correlations and therefore cross-price elasticities. Importantly, the combinatorial logit model leads to closed-form expressions for purchase probabilities, thus removing the need to perform the costly numerical integration required for many standard demand models. This makes estimation computationally light and numerically stable. Because we construct multiple similarity measures, many of which are highly correlated, we employ a forward stepwise selection algorithm to select the best-fitting similarity measures.

We apply this estimation method to a dataset constructed from the Comscore Web Behavior

panel, which describes purchases made in several categories on Amazon.com. We complement purchase data from Comscore with product images, titles, descriptions, reviews, and Q&As collected directly from Amazon's product detail pages. We first assess the performance of the different similarity measures by including each of them individually in the model. We then compare the fit of each model to that of a simple logit model, which is nested within the paired combinatorial model. Based on the Akaike Information Criterion (AIC), we find that many of our 20 similarity variables improve model fit relative to the logit model. In general, image similarities lead to the largest fit improvements. Among the text-based similarity measures, the measures based on the simple bag-of-words model perform best when these models are applied to product titles, descriptions, and reviews. By contrast, the more complex sentence classifiers perform better when applied to the texts of Q&As. This varying performance of different similarity measures illustrates that, to extract information from unstructured data, one needs to consider a range of models and data types.

Applying a forward stepwise selection algorithm, we find that the best fitting model includes two image-based similarity variables. We show that this selected model generates cross-price elasticities that are substantially different from a simple logit and that align with similarity in important product attributes. At the same time, we find that only around half of the variation in the selected pairwise variables is explained by how close the products are in the space of observed product attributes. Thus, our similarity measures partially capture substitution patterns that cannot be captured with the observed attributes one would typically use in standard demand estimation approaches.

This paper contributes to an extensive literature on estimating discrete choice demand models. The idea of introducing correlated error terms into multinomial models dates back to the early work of Hausman and Wise (1978) and McFadden (1977). Much of the existing empirical work achieves this by introducing a pre-determined nesting structure or adding random effects to an attribute-based model (Berry et al., 1995, 2004). Our work is most closely related to a relatively small set of papers that parameterize the covariance of utility errors with product similarity measures (Bresnahan et al., 1997; Pinkse and Slade, 2004; Dotson et al., 2018). For example, Bresnahan et al. (1997) estimate a demand model that allows for error correlations between products that share observed attributes, and similarly, Pinkse and Slade (2004) and Dotson et al. (2018) model cross-price elasticities as a function of brand similarities in attribute space. Our approach also parameterizes cross-price elasticities as a function of product similarities, but we compute these similarities from text and image data rather than observed attributes. Thus, our estimation method is more generalizable in that it does not require researchers to collect category-specific attributes.

Several authors estimate a pairwise combinatorial logit model, which we also adopt in our application. Small (1987) estimates this model in settings where alternatives are naturally ordered and assumes that utility correlations get weaker for options that are farther apart. Koppelman and Wen (2000) apply the same model in a setting with only three choices but allow each product pair to have a different correlation. By contrast, our approach does not require products to be ordered,

2

and it can be applied in settings with a larger numbers of alternatives because we model error term correlations as a function of lower-dimensional text and image similarity measures.

Our work is also related to the emerging literature that uses additional data from texts, images, or consumer surveys to better estimate substitution patterns.[1] Dotson et al. (2019) use image data for demand estimation but follow a different methodology. They survey participants asking to rate the appeal of each image and then include the resulting ratings in their demand model as an additional "attribute."[2] Magnolfi et al. (2022) solicit product comparisons via a survey to generate data of the form "product A is closer to B than it is to C" and use these data to compute low-dimensional product embeddings that are later included into the utility function. Although their method may work well when consumers are familiar with presented products, and therefore can easily relate them to each other, it may not work as well for new or niche products that are less known to consumers. Further, both methods require collecting category-specific survey data, which makes them less scalable than our approach that is based on widely available text and image data.

Finally, Netzer et al. (2012) use data on the co-occurrence of products mentioned in online discussion forums to estimate substitution patterns. Although their co-occurence measures resemble our text similarities, they do not incorporate these measures in demand estimation. Our approach also does not assume that any particular measure of similarity is a good proxy for the degree of substitution between products. Instead, we infer from the data which of our candidate similarity measures provide the largest explanatory power in demand estimation.

2   Model

In this section, we describe the demand model framework we use for estimation. We focus on describing how pairwise measures of product similarity enter the demand model and how they affect elasticities. In Section 3, we describe how these similarity measures are computed from product images and texts such as product descriptions and customer reviews.

2.1   Model Setup and Properties

We consider a market where each consumer has unit demand and chooses from the set of J available products. Consumer i obtains the following utility from purchasing product j:

$$u_{ij} = \bar{u}_{ij} + \varepsilon_{ij} = \delta_j - \alpha \cdot \text{price}_{ij} + \varepsilon_{ij} \tag{1}$$

where δ_j is a product fixed effect capturing unobserved quality differences, α denotes the price coefficient which is assumed to be homogeneous across consumers, price_{ij} is product j's price that

---

[1] An alternative approach, put forth by several papers, has been to use auxiliary search data to inform cross-price elasticities and analyze product substitution. (Kim et al., 2011; Armona et al., 2021; Amano et al., 2022).

[2] Sisodia et al. (2022) also extract interpretable product attributes from images, which are then used as an input to a conjoint analysis.

consumer i faces when making the choice[3], and ε_{ij} is an idiosyncratic taste shock. For simplicity, we assume that there is no outside option, so the consumer must choose one of J alternatives.

To model how consumers substitute between products, researchers typically assume that consumers have heterogeneous tastes for observed product attributes x_j (Berry et al., 1995). Under this assumption, heterogeneity in tastes over these attributes implies that products with similar attributes x_j have positively correlated utilities u_{ij}. Thus, if the price of some product j increases, consumers who would normally buy this product will disproportionately substitute to other products whose attributes x_k are similar to those of product j. Although our approach is similar in spirit, we follow a different strategy when modeling the correlations of product utilities. In particular, we do not include attributes x_j in the model and instead directly parameterize the joint distribution of taste shocks $\varepsilon_{i1}, \ldots, \varepsilon_{iJ}$ and allow it to depend on product similarity measures.

Following Small (1987) and Koppelman and Wen (2000), we assume a paired combinatorial logit model – a generalized version of the nested logit model with overlapping nests. The model includes a separate nest for every product pair. Specifically, we assume that the taste shocks ε_{ij} for each product pair (i.e., within each nest) follow a correlated extreme value distribution with its own *correlation parameter* parameter λ_{jk}. A lower value of λ_{jk} implies a stronger correlation between the taste shocks of products j and k. Thus, one can think about $(1 - \lambda_{jk})$ as measuring the degree of substitutability between products j and k. Similar to a standard nested logit model, the pairwise combinatorial logit model is consistent with utility maximization if $0 < \lambda_{jk} \leq 1$. The model becomes a simple non-nested logit model when $\lambda_{jk} = 1$ for all product pairs.

In this model, the purchase probability of consumer i for product j is given by:

$$
p_{ij} \quad = \quad \frac{\sum_{k \neq j} \exp(\bar{u}_{ij}/\lambda_{jk}) \left(\exp(\bar{u}_{ij}/\lambda_{jk}) + \exp(\bar{u}_{ik}/\lambda_{jk})\right)^{\lambda_{jk}-1}}{\sum_{l=1}^{J-1} \sum_{m=l+1}^{J} \left(\exp(\bar{u}_{il}/\lambda_{lm}) + \exp(\bar{u}_{im}/\lambda_{lm})\right)^{\lambda_{lm}}}. \tag{2}
$$

To gain intuition for this expression, we can re-write it by multiplying the probability that the consumer chooses some product pair (j, k) containing product j and the conditional probability of choosing product j from this pair. That is, we can express $p_{ij} = \sum_{k \neq j} p_{ij|jk} \times p_{ijk}$ where the sum is across all product pairs that include product j and where the two probabilities under the sum are given by:

$$
p_{ij|jk} \quad = \quad \frac{\exp(\bar{u}_{ij}/\lambda_{jk})}{\exp(\bar{u}_{ij}/\lambda_{jk}) + \exp(\bar{u}_{ik}/\lambda_{jk})}, \tag{3}
$$

$$
p_{ijk} \quad = \quad \frac{\left(\exp(\bar{u}_{ij}/\lambda_{jk}) + \exp(\bar{u}_{ik}/\lambda_{jk})\right)^{\lambda_{jk}}}{\sum_{l=1}^{J-1} \sum_{m=l+1}^{J} \left(\exp(\bar{u}_{il}/\lambda_{lm}) + \exp(\bar{u}_{im}/\lambda_{lm})\right)^{\lambda_{lm}}}. \tag{4}
$$

Because the denominator in (4) sums over all possible product pairs, the probability p_{ijk} resembles the standard logit probability if we imagine the consumer chooses a product pair (j, k) out of all

---

[3]We index prices with i because in the data, different consumers are observed at different points in time and therefore face different prices.

4

possible pairs. Similarly, the probability $p_{ij|jk}$ in (3) is the standard logit probability where the consumer chooses product j conditional on choosing a product from the pair (j, k). Therefore, intuitively, the purchase probabilities in (2) recognize that the consumer can choose product j after considering any pair that includes this option, and they compute a weighted sum of corresponding conditional probabilities. We emphasize that these purchase probabilities have closed-form expressions; therefore, we do not need to approximate them using simulations as is the case in the commonly used mixed logit specifications.

We also note that it would be straightforward to include observed attributes x_j, over which consumers have heterogeneous preferences β_i, into the utility function (1). In such a model, substitution patterns would be driven by the similarity in observed attributes as well as the similarity in product-pair-specific variables. For simplicity, we do not introduce such observed attributes and instead focus on exploring how substitution can be modeled using similarity measures derived from images and texts.

2.2 Covariance Matrix

To complete the model, we parameterize the correlation parameters λ_{jk} to be functions of pairwise similarity measures. This parameterization avoids the need to nonparametrically estimate all $J(J-1)/2$ nesting parameters λ_{jk}, which would be computationally prohibitive except for small choice sets J. More importantly, this specification enables us to model the correlations in utilities u_{ij} as a function of image-based and text-based product similarity measures. We parameterize λ_{jk} such that:

$$\lambda_{jk} = \frac{1}{1 + \exp(-\theta + w'_{jk}\beta)} \tag{5}$$

where w_{jk} denotes a column vector of pairwise similarity measures. The functional form ensures that $\lambda_{jk} \in (0, 1]$ for any parameter values, thus keeping the specification consistent with utility maximization. In practice, we normalize w_{jk} such that higher values of variables w_{jk} indicate higher similarity between products j and k and $w_{jk} = 0$ captures the case when the two products are maximally dissimilar. We do not estimate θ and instead set it to a relatively large number so that substitution patterns collapse to the logit case ($\lambda_{jk} = 1$) for products that are very dissimilar and hence have values of w_{jk} close to zero.[4] If greater similarity in w_{jk} translates into higher correlation in utilities λ_{jk}, then we expect to find $\beta > 0$. In this case, greater similarity leads products to be closer substitutes, which shifts the model farther away from the logit model with uncorrelated taste shocks ε_{ij}.

2.3 Cross-Price Elasticities

As Koppelman and Wen (2000) show, in the paired combinatorial logit model, the cross-price elasticity of demand for product j with respect to product k's price is given by:

---

[4]In practice, we set $\theta = 4$ in estimation, which generates numerical values of the nesting parameter that are close to logit ($\lambda_{jk} \approx 1$) for very dissimilar products.

$$e_{jk} = \alpha \times \text{price}_k \left(p_k + \left(\frac{1 - \lambda_{jk}}{\lambda_{jk}} \right) \frac{p_{jk} \cdot p_{j|jk} \cdot p_{k|jk}}{p_j} \right). \tag{6}$$

For dissimilar products with $\lambda_{jk} = 1$, this elasticity reduces to the well-known logit elasticity $e_{jk} = \alpha \cdot \text{price}_k \cdot p_k$ that depends only on the market share and price of product k. As products become more similar and λ_{jk} decreases, the second term in the brackets also grows, thus reflecting that these products become closer substitutes. The elasticity formula in (6) is also closely related to the elasticity formula for the nested logit model with non-overlapping nests. If products j and k are in the same nest, but neither product is included in any other nest, it follows that $p_j = p_{j|jk} \cdot p_{jk}$. In this case, the elasticity expression above simplifies to the standard nested logit elasticity $e_{jk} = \alpha \cdot \text{price}_k \cdot (p_k + (1 - \lambda_{jk})/\lambda_{jk} \cdot p_{k|jk})$.

2.4   Comparison to Other Models

Our approach has several advantages over traditional methods that model substitution via observed product attributes x_j. First, it removes the need to collect data on product attributes x_j in each category. Collecting such attributes takes considerable effort, and it is often a subjective process because researchers need to decide which attributes are the most relevant for consumers. Our approach removes this attribute selection step, thus reducing the impact of researcher's choices on demand estimates. We instead model substitution by using data on images and texts that describe each product. Because these data are not category-specific, one can easily apply our approach to any category as long as image and text data are available.

Second, because the researcher can include the same similarity variables w_{jk} regardless of the category, it is possible to pool data across multiple categories if one is willing to assume that similarity variables have the same effect on substitution patterns across these markets. Pooling data across categories can be especially beneficial when researchers have sparse purchase data in individual categories. It would be difficult to pool data in this way in standard demand models because each product category would likely have its own unique set of of choice-relevant attributes.

Third, our approach can capture product similarity along dimensions that are difficult to capture with observed product attributes x_j. For example, similar product images might imply that these products are aesthetically similar, which makes them close substitutes from the consumer's perspective. Such visual similarity may not be captured via observed attributes, which are usually confined to product's physical characteristics. Modeling substitution from images might be especially appealing in product categories where product aesthetics strongly influence consumers' choices (e.g., categories of clothing or home decorations). Textual product descriptions might also correlate with attributes that are difficult to measure. For example, consumers might write reviews describing the best uses of a product (e.g., earphones might be more suitable for music or audio books), which are only indirectly captured by observed attributes.

Several other approaches estimate substitution without relying on observed attributes or product similarity measures. Examples include the latent attributes approach used by Ruiz et al. (2017)

and the latent partitions approach by Smith et al. (2019). These approaches require more data because substitution patterns are estimated directly from the data without the help of observed attributes. By contrast, our approach uses data on image-based and text-based similarity measures, which simplifies estimation by providing a lower-dimensional representation of elasticities.

Finally, our approach is closely related to that of Magnolfi et al. (2022), who use a survey to directly ask consumers which products are similar to each other. The authors use survey data to compute product embeddings, which reflect product locations in a low-dimensional representation of the product space. They demonstrate two ways to use these embeddings in estimation: by including them in the same way as standard product attributes x_j or by using them to directly discipline the cross-elasticity parameters of a log-linear demand model. In contrast to their approach, we compute embeddings from unstructured text and image data that are widely available from the web. In this sense, our approach is more scalable because it does not require researchers to ask survey questions about each product pair (and each category) included in estimation. Further, their method is likely to work less well for new or niche products that consumers are unfamiliar with. By contrast, our method can be applied to any product category as long as researchers can collect texts or images.

3 Data and Descriptive Statistics

We combine data on online purchases on Amazon.com with the histories of daily prices of Amazon products collected from a third-party database. We also gather product images and textual descriptions information from product detail pages that consumers visit when making their choices.

3.1 Purchases and Prices

We obtain purchase data from the 2019-2020 Comscore Web Behavior Panel, which contains a sample of about two million U.S. households. We focus on Amazon.com because the Comscore dataset contains many more Amazon purchases than purchases from other online retailers. This high density of Amazon data enables us to identify several product categories where we observe a sufficient number of purchases for demand estimation. We complement the purchase data with daily price histories of products collected from the third-party database Keepa.com.

We apply our method to four categories of electronic goods: headphones, tablets, memory cards, and computer monitors. We selected these categories because they are characterized by frequent purchases as well as rich temporal variation in prices. Because these four categories can be broadly classified as "Electronics," it seems reasonable to pool data across them for estimation. Table 1 shows summary statistics for the four selected categories. We observe in total 2,749 purchases: 1,598 in the category of Headphones, 593 in Tablets, 280 in Memory Cards, and 278 in Monitors. Because we do not include the outside option in the model (see Section 2.1), we omit category visits in which consumers did not purchase one of the products we pre-selected in that category.

7

Category	Number of Products Selected	Total Purchases	Price Average Dollars	Price Std. Dev. Dollars
Headphones	28	1598	81.51	12.32
Tablets	8	593	149.31	17.89
Memory Cards	6	280	12.26	1.52
Monitors	9	278	127.00	8.59

Table 1: **Descriptive Statistics: Four Selected Categories.** The table shows summary statistics for the four categories selected into our estimation sample. The last column shows the average standard deviation of prices over time for individual products in that category.

In Appendix A, we provide additional details about the dataset and our category and product selection process.

3.2 Data on Product Images and Textual Descriptions

We augment the purchase and price data with product similarity variables. These variables fall into two categories: visual product similarities constructed from product images and text-based similarities constructed from product names, seller-provided product descriptions, discussions in the Q&A section, and customer reviews. We collect all relevant image and text data directly from product detail pages on Amazon.com.[5]

Figure 1 shows an example of a product detail page. We extract the default product image shown at the top of the product detail page. Further, we gather textual data from several fields on the product pages. We collect product titles displayed at the top of the page, and we gather the text from the bullet points describing the product's attributes, which we term a "product description". We also gather texts from the Q&A section that contains specific questions consumers asked about the product, as well as the answers posted by the seller or other consumers. Finally, we extract the texts of the 100 most recent reviews for each product. Because the HTML structure of pages somewhat varies across categories on Amazon, not all product pages contain all four textual fields, thus leading to missing data.[6] Nevertheless, we observe product titles and at least one other textual element (descriptions, Q&A, or reviews) for all product categories in our sample.

---

[5]We collected most of these data in 2022-2023, whereas our dataset of purchases covers 2019-2020. Because images, titles, and descriptions are typically provided by sellers, we do not expect these to change over time. By contrast, customer reviews and Q&As may change. Such temporal changes should not bias our estimation as long as the typical content in these textual fields is informative about choice-relevant product attributes.

[6]In estimation, whenever a pairwise variable w_{jk} is missing, we set the corresponding $w'_{jk}\beta$ to zero in Equation (5). For models with just one w_{jk} variable, this means that we take the logit case with $\lambda_{jk} = 1$ as a default when the pairwise variable is missing.

Figure 1: **Example: Image and Text Data Collected From Product Pages.**

3.2.1 Image-Based Product Similarities

To construct similarity measures from images, we employ a series of deep learning models that were originally built for object detection and classification tasks. Specifically, we use several pre-trained models that are available from Keras Deep Learning Library. We use pre-trained models because our main goal is to develop a general demand estimation method that does not require any category-specific data. Using pre-trained models also helps us avoid training custom image models for each category, which makes our estimation approach scalable and computationally light.

We use four different classification models. First, we use VGG19, a very deep convolutional neural network with 19 layers. VGG19 is one of the most popular algorithms for image classification that performed well in image classification competitions (Simonyan and Zisserman, 2015). Second, we use ResNet50, a convolutional neural network that is 50 layers deep. ResNet50 is a "residual network", a specific type of artificial neural network that forms networks by stacking residual blocks (He et al., 2016). Finally, we use InceptionV3 and Xception, convolutional neural networks that are 48 and 71 layers deep (Szegedy et al., 2016; Chollet, 2017). All four models have a high predictive accuracy of 90-94.5% on the ImageNet validation dataset.[7] Because these models perform well at distinguishing similar objects, we expect them to do well at detecting pairs of products that consumers perceive as visually similar.

Each model first transforms the original image into a lower-dimensional vector representation – an "embedding" – and then classifies an image by predicting which object it contains from the embedding. Originally, these models were trained to classify images into labeled classes of similar objects (e.g., "cup," "book," or "sofa"). However, because our aim is not to label products but to measure how similar they are, we remove the classification layer from these models and instead directly work with the embeddings they produce. Specifically, to compute similarity between the images of two products, we compute the Euclidean distance between the embeddings of these two images, take its negative, and normalize its value to be between 0 and 1. This process yields four similarity metrics $w_{jk} \in [0, 1]$, one per model, which are then used to parameterize the covariance parameters λ_{jk} in equation (5).

We do not commit to any specific model. Instead, we compute several similarity measures and let our estimation algorithm select the combination of these measures that performs best at explaining substitution patterns. We intentionally choose models with different architectures to generate variation in the image similarity metrics.

3.2.2 Text-Based Product Similarities

Next, we compute text-based similarity measures based on customer reviews, product titles, descriptions, and Q&As. The main idea behind these measures is that, if two products are similar,

---

[7]The ImageNet data (https://www.image-net.org/) is used for the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) which evaluates algorithms for object detection and image classification, a benchmark in object category classification (Russakovsky et al., 2015). Precision is defined as one minus the top5 mis-classification error (the true object class is not among the top 5 predicted classes), which is the metric used to determine the best performing model in the ILSVRC.

sellers and consumers will tend to describe them both in similar ways. For example, because seller-provided product titles often contain a brief description of key attributes (e.g., "Apple iPad 10.2-inch Retina Display, 64GB, 12MP front camera"), the similarity of product titles may indicate the two products share some choice-relevant attributes such as brand, screen size, or camera resolution. Similarly, consumers might mention in their reviews that a particular tablet is suitable for kids because it survives most drops and is easy to draw on. If two tablets are characterized in this way, the textual similarity of their reviews will indicate that these tablets might be close substitutes.

As with image similarities, we use a sequence of increasingly sophisticated models to compute text similarities. We use four models, and for each model we compute a distinct text similarity metric for each type of textual data (i.e, titles, descriptions, Q&A, and reviews). First, we use a simple bag-of-words count model, which transforms each text into fixed-length vectors by counting the number of word occurrences. This method does not take the order of words into account and only measures how often words appear in the document.

Second, we use the same bag-of-words model but with a TF-IDF vectorizer. Although similar to the previous method, this approach places a large weight on "unique" words – the words that appear frequently in a given document but infrequently in other documents. This approach leads to a larger emphasis on words that are unique to a subset of products, thus making it more likely that our text-based similarities reflect some unique attributes shared between the two alternatives.

Third, we use the Universal Sentence Encoder model (USE), which converts each sentence into a 512-dimensional sentence embedding. These embeddings are typically used for text classification, semantic similarity, clustering, and other natural language tasks. We use a pre-trained Universal Sentence Encoder model based on Cer et al. (2018). In contrast to the bag of words models, this model accounts for the order of words and the context in which they appear.

Fourth, we use the Sentence Transformer model (ST). Specifically, we use the pre-trained Sentence-BERT model made available by Reimers and Gurevych (2019). Their model is a more efficient modification of a widely used BERT network (Devlin et al., 2018), and it is trained to extract semantically meaningful sentence embeddings. Similar to the Sentence Transformer, this model also assesses the context in which words appear in sentences.

Before applying these models, we pre-process our text data as described in Appendix B. Using our text models, we then compute product distance as the Euclidean distance between the two embeddings extracted from the two products' textual descriptions. We translate these distances into similarities in the same way as for the image data (see Section 3.2.1).

3.3 Illustrative Example

To better understand what our similarity measures capture, consider an example from our data shown in Figure 2. The nodes in this graph correspond to four tablets: two Amazon Fire tablets with slightly different attributes (such as display size and storage capacity), a kid's model of Amazon Fire, and an Apple iPad. We label edges with image and description-based similarities computed

11
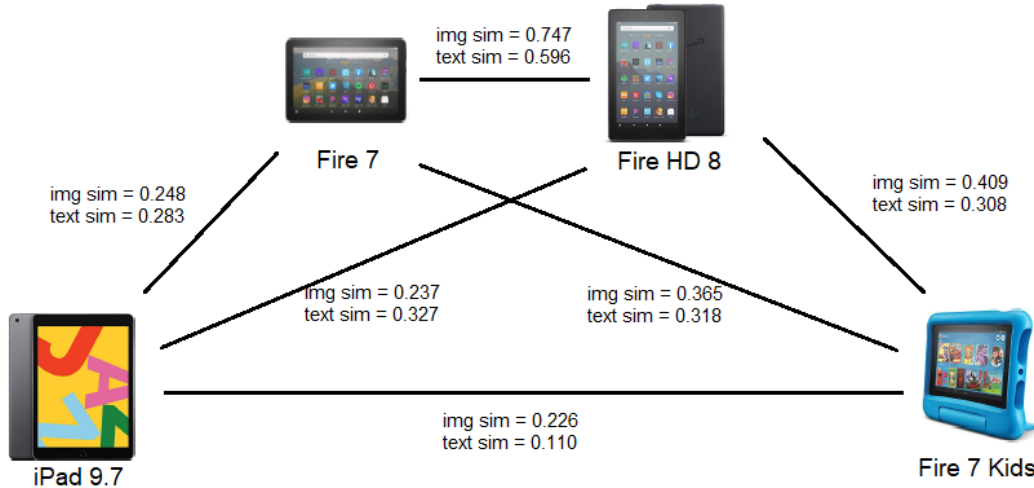
Figure 2: **Illustrative Example: Image and Product Description Similarities of Four Tablets.**

for the corresponding product pairs (see Appendix Figure A1 for the complete product descriptions of these four tablets).

We find that image and text similarities largely follow the similarity of attributes. Based on product descriptions, the most similar to the Fire 7 tablet is Fire 8 tablet, another model from the same brand offering similar technical features. The second most similar product is the Fire 7 Kids tablet, which belongs to the same brand but clearly targets families with kids. Finally, the least similar to the Fire 7 product is the tablet Apple iPad 9.7, which belongs to a different brand. Image similarities exhibit similar patterns.

Finally, in Tables 2-3 we show the image and description similarities of all eight tablets included in our sample. Overall, measures do capture product similarity well. For example, description similarities are high between the two iPads, among the three models of Amazon Fire, and between the two Fire tablets for kids. Similarity measures in these cases are around 0.6-0.7, much higher than for other pairs of products that appear more different from each other. More generally, these measures seem to perform well at detecting the pairs of products that share key attributes (e.g., HD display or screen size), target similar demographics (e.g, families with kids), or have the same brand.

4   Estimation and Results

Recall from Section 3.1 that we have data from four categories, and in each category we observe a cross-section of consumers' purchase decisions. Let $c = 1, \ldots, C$ index categories and let $\mathcal{Y}_c$ denote the set of consumers in category c. Further, assume y_{ij} is the indicator that consumer i purchased product j, P is the matrix of prices, and W is the matrix of similarity measures. Our goal is to use these data to estimate parameters $\theta = (\alpha, \beta, \delta)$ where α is price sensitivity, δ is the vector of

	Fire 7	Fire HD 8	Fire HD 10	Fire 7 Kids	iPad 10.2	Fire HD 8 Kids	Dragon Touch	iPad 9.7
Fire 7	1.000	0.747	0.729	0.365	0.505	0.369	0.435	0.248
Fire HD 8		1.000	0.764	0.409	0.486	0.402	0.402	0.237
Fire HD 10			1.000	0.444	0.498	0.449	0.439	0.262
Fire 7 Kids				1.000	0.355	0.701	0.317	0.226
iPad 10.2					1.000	0.362	0.480	0.409
Fire HD 8 Kids						1.000	0.319	0.237
Dragon Touch							1.000	0.361
iPad 9.7								1.000

Model: VGG19

Table 2: **Product Image Similarities for Tablets.**

	Fire 7	Fire HD 8	Fire HD 10	Fire 7 Kids	iPad 10.2	Fire HD 8 Kids	Dragon Touch	iPad 9.7
Fire 7	1.000	0.596	0.627	0.318	0.265	0.328	0.361	0.283
Fire HD 8		1.000	0.671	0.308	0.303	0.386	0.393	0.327
Fire HD 10			1.000	0.288	0.313	0.357	0.368	0.329
Fire 7 Kids				1.000	0.100	0.730	0.257	0.110
iPad 10.2					1.000	0.153	0.257	0.710
Fire HD 8 Kids						1.000	0.282	0.167
Dragon Touch							1.000	0.250
iPad 9.7								1.000

Model: Universal Sentence Encoder

Table 3: **Product Description Similarities for Tablets.**

product fixed-effects, and β is the vector of coefficients on product similarities in (5). We estimate parameters θ by maximizing the log-likelihood of the data computed as

$$logL(\theta|P,W) = \sum_{c=1}^{C} \sum_{i \in \mathcal{Y}_c} \sum_{j=1}^{J} y_{ij} \log(p_{ij}^c(\theta))$$

where $p_{ij}^c(\theta)$ is the purchase probability for product j in category c given by (2). We assume a uniform price coefficient α and uniform similarity coefficients β in order to pool data across the four categories. We also normalize the fixed effect of one product in each category to zero to fix the location of utility. Thus, we estimate in total 47 fixed effects for the 51 products included in the sample.

4.1 Specifications Using a Single Similarity Measure

Before turning to our main specification, we start by estimating a sequence of models, each including only one similarity measure. This analysis helps us explore the informational content of different similarity measures. The results from these specifications are reported in Table 4. We find that more than half of all similarity measures lead to a significant improvement in model fit relative to the standard logit model. However, some measures are more predictive of substitution patterns than others. For instance, image-based similarity variables lead to the largest AIC improvements. In fact, the smallest AIC improvement we get from including any image-based similarity exceeds the largest fit improvement from including any text-based similarity measure.

Among the text-based measures, the results are more sensitive to the exact model and data type used to compute similarity. For each type of textual data, there is at least one text model for which the estimated demand model is not statistically distinguishable from a simple logit model (as indicated by the p-values of the likelihood ratio statistic in column 3). For measures based on product titles and descriptions, the simple bag-of-word models lead to the largest improvement in fit. By contrast, for Q&A-based measures, the best fitting similarity measures are those from two more complex text models, the Universal Sentence Encoder and the Sentence Transformer model. This is consistent with the fact that product titles and descriptions often simply list the product specifications and are thus easily captured by bag-of-words models, whereas the more nuanced text in the Q&As requires more sophisticated models. One might expect the reviews text to also fall in this latter category, but we find that for that type of data input, all ML models perform fairly similarly (and none of them substantially improves over logit).

We also report the range of cross-price elasticities generated by each model in the last column of Table 4. To compute this statistic, we consider an increase in the price of a specific product and compute the elasticity of demand for other products in the same category with respect to this price change. We compute the range of these cross-price elasticities for a given product's price change, and we then average the result across all products and all four categories in our sample. As before, we use the simple logit model as a benchmark. Because cross-price elasticities in the logit model only depend on the price and market share of the product whose price changes, the range of cross-price elasticities — as defined above — is always zero. To interpret the magnitudes of these ranges for the different models, note that the average cross-price elasticity is roughly equal to 0.31 across all estimated models. Compared to this average, the range of cross-price elasticities induced by including some of the image and text similarities is relatively large with values of up to 1.002 and 0.709, respectively. Therefore, as Table 4 shows, several similarity measures generate elasticities that substantially deviate from the IIA-type substitution patterns of the logit model.

4.2 Selection of Similarity Measures

Although several similarity measures improve model fit relative to a simple logit model, many of them are highly correlated. In Appendix Table A1, we display correlations between the best fitting metrics based on each of the different data inputs, where the "best fit" is defined based on

Included Similarity Measure	AIC	ΔAIC	LR Test P-val	Cross- Elast. Range
Panel A. Image-based similarities				
VGG19	12928.99	-30.68	0.000	0.605
ResNet50	12938.14	-21.52	0.000	1.002
Xception	12934.74	-24.92	0.000	0.361
InceptionV3	12928.67	-31.00	0.000	0.792
Panel B. Text-based similarities (titles)				
Bag of Words	12942.30	-17.36	0.000	0.567
Bag of Words TF-ID	12956.30	-3.36	0.021	0.034
Universal Sentence Encoder	12959.11	-0.56	0.110	0.703
Sentence Transformer	12961.59	1.93	0.788	0.677
Panel C. Text-based similarities (descriptions)				
Bag of Words	12948.76	-10.90	0.000	0.150
Bag of Words TF-ID	12957.86	-1.80	0.051	0.116
Universal Sentence Encoder	12961.67	2.00	1.000	0.001
Sentence Transformer	12957.54	-2.12	0.042	0.037
Panel D. Text-based similarities (Q&A)				
Bag of Words	12961.67	2.00	1.000	0.001
Bag of Words TF-ID	12961.62	1.95	0.825	0.003
Universal Sentence Encoder	12948.46	-11.20	0.000	0.270
Sentence Transformer	12947.32	-12.34	0.000	0.188
Panel E. Text-based similarities (reviews)				
Bag of Words	12960.31	0.65	0.244	0.040
Bag of Words TF-ID	12958.33	-1.34	0.068	0.709
Universal Sentence Encoder	12961.67	2.00	1.000	0.001
Sentence Transformer	12961.67	2.00	1.000	0.001
Panel F. Logit model (benchmark):				
Simple Logit Model	12959.66			0.000

Table 4: **Combinatorial Logit Estimation Results.** Different rows in Panels A-E show estimation results for a combinatorial logit model with only one included pairwise variable. Panel F shows results from a simple logit model as a benchmark for comparison. We report AIC values (column 1), ΔAIC defined as a change in AIC relative to the simple logit model (column 2), the p-value of the likelihood ratio statistic for testing each model against the simple logit model (column 3), and the range of estimated cross-price elasticities (column 4).

the lowest AIC value reported in column 1 of Table 4. We find a positive correlation between all measures of similarity. The most strongly correlated variables are those based on product Q&As and reviews, suggesting that both data types contain somewhat similar information regarding product substitution. In Appendix Tables A2-A6, we report the correlation matrix for all similarity measures derived from the same data type. In most cases, we find fairly large positive correlations across models within a given data type.

To summarize, we find that many similarity measures are highly correlated, with correlations being particularly high for different models applied to the same data type (e.g., across image models). Moreover, not all similarity measures improve fit, and there are no clear patterns with regards to which models or data types perform best. It is therefore not obvious a priori which product similarity measures should perform best at identifying products that are close substitutes. We also cannot include all similarity measures as that would likely lead to overfitting. Taken together, these observations suggest that we need to select similarity variables in a data-driven way.

To this end, we optimally select similarity measures w_{jk} using a forward stepwise selection algorithm, a computationally efficient approximation of the best subset of similarity measures.[8] To perform this selection, we only vary which similarity metrics are included in w_{jk}, while leaving the rest of the model unchanged. We start with a simple logit model where we set $\lambda_{jk}^c = 1$ for all product pairs in all categories. We then estimate a sequence of models, each including only one similarity metric w_{jk}, and we select the best-fitting model that yields the largest AIC improvement relative to logit (this step corresponds to estimating the models in Table 4). Next, we consider adding one more similarity metric and select the one that yields the largest AIC improvement. This process is repeated until adding similarity metrics no longer improves AIC or until we run out of similarity metrics to include.

4.3 Main Specification: Optimally Selected Similarity Measures

We report results from our final specification in Table 5. Our final specification contains two image-based similarities from models VGG19 and InceptionV3. This result is unsurprising given that these two image-based similarities achieve the best fit improvement in Table 4 when included individually, and they are highly correlated with text-based measures.

In Tables 6 and 7, we illustrate the substitution patterns generated by our model for the category of tablets. We explore substitution patterns by showing the matrices of cross-price elasticities and diversion ratios, where diversion ratios measure the share of the demand reduction that is diverted to each alternative when the price of the focal product increases. In both tables, we report the demand response for the row product to changing the price of the column product. In the case of diversion ratios, the off-diagonal values add up to one in each column by definition.

---

[8]Performing the best subset selection would be infeasible because with 20 similarity metrics, we have $2^{20} = 1,048,576$ possible models to select from.

	Coeff.	S.E.
Price coefficient	-0.008	0.001
Image similarity InceptionV3	7.850	5.694
Image similarity VGG19	12.128	4.726
Product FE	Yes	

Table 5: **Estimation Results.** The table shows coefficient estimates and their standard errors from the model selected by the greedy forward algorithm described in Section 4.2.

Elasticities	Fire 7	Fire HD 8	Fire HD 10	Fire 7 Kids	iPad 10.2	Fire HD 8 Kids	Dragon Touch	iPad 9.7
Fire 7	-0.601	0.509	0.424	0.140	0.199	0.047	0.031	0.098
Fire HD 8	0.315	-1.215	0.337	0.165	0.343	0.042	0.049	0.117
Fire HD 10	0.174	0.251	-1.636	0.237	0.386	0.037	0.034	0.126
Fire 7 Kids	0.151	0.285	0.508	-1.226	0.549	0.042	0.112	0.189
iPad 10.2	0.105	0.235	0.338	0.242	-8.081	0.610	0.970	0.380
Fire HD 8 Kids	0.105	0.164	0.188	0.088	1.601	-2.893	0.578	1.372
Dragon Touch	0.093	0.244	0.204	0.294	2.606	0.661	-2.721	1.054
iPad 9.7	0.210	0.431	0.287	0.279	1.267	1.321	1.127	-13.136

Table 6: **Estimated Elasticities in the Category of Tablets.** Each entry in the table is an estimated elasticity of the demand for the row product with respect to the price of the column product.

Diversion ratios	Fire 7	Fire HD 8	Fire HD 10	Fire 7 Kids	iPad 10.2	Fire HD 8 Kids	Dragon Touch	iPad 9.7
Fire 7	-1.000	0.465	0.301	0.240	0.109	0.121	0.082	0.121
Fire HD 8	0.464	-1.000	0.263	0.229	0.131	0.108	0.117	0.136
Fire HD 10	0.247	0.217	-1.000	0.254	0.128	0.085	0.066	0.069
Fire 7 Kids	0.145	0.149	0.231	-1.000	0.140	0.058	0.132	0.087
iPad 10.2	0.058	0.079	0.104	0.122	-1.000	0.202	0.275	0.088
Fire HD 8 Kids	0.035	0.028	0.030	0.030	0.151	-1.000	0.197	0.242
Dragon Touch	0.025	0.035	0.031	0.081	0.301	0.247	-1.000	0.258
iPad 9.7	0.027	0.028	0.040	0.045	0.040	0.179	0.132	-1.000

Table 7: **Estimated Diversion Ratios in the Category of Tablets.** Each entry in the table is a diversion rate of the demand for the row product with respect to the price of the column product. Diversion ratios measure the share of the demand reduction that is diverted to each alternative when the price of the focal product increases.

17

In line with the results reported in Table 4, our selected specification generates rich substitution patterns and substantial deviations from IIA. This pattern can best be gleaned from the elasticity matrix in Table 6. For example, when Fire 7 becomes more expensive, consumers substitute to Fire HD 8 or Fire HD 10 with elasticities of 0.315 and 0.174, higher than cross-price elasticities of demand for other tablets. The lowest cross-price elasticities in that column are those for Dragon Touch (0.093), iPad 10.2 (0.105), and Fire HD 8 Kids (0.105), consistent with our intuition in the illustrative example (Section 3.3). Similar patterns hold for other tablets in the assortment. These results are in contrast to a simple logit model, in which the cross-price elasticities in each column are identical because they only depend on the market share and price of the column product (the product whose price changes). Hence, the selected model does generate large deviations from the IIA substitution patterns of a simple logit. The matrix of diversion ratios in Table 7 is an alternative way to illustrate substitutions patterns (Conlon and Mortimer, 2021). In line with cross-price elasticities, the diversion ratios also reveal large differences in substitutability across product pairs.

Finally, we note that the estimated substitution patterns do not perfectly match the patterns in the selected pairwise variables. This can be gleaned by comparing the elasticity and diversion matrices in Tables 6 and 7 with Table 2, which shows the values of one of the two similarity variables selected by our procedure. For instance, while the Fire 7 Kids tablet is most visually similar to the Fire HD 8 Kids, the elasticities and diversions between these two products are fairly small. These discrepancies are due to the fact that other factors — specifically, differences in the products' average prices and in their vertical qualities, as captured by the product fixed effects — contribute to the estimated substitution patterns. Thus, substitution patterns are affected, but not entirely determined, by the selected pairwise variables.

4.4 What Do Pairwise Variables Capture?

In this section, we explore the extent to which the pairwise variables which were selected for our preferred specification capture similarity in observed product attributes. If our similarity measures strongly correlate with the similarity in product attributes, our approach should generate similar cross-price elasticities as a standard characteristics-based demand model with random coefficients. At the same time, our similarity measures may also partially capture product aspects that are hard to measure using observed product attributes, such as aesthetic product similarity.

To analyze the role of observed attributes, we take the two image-based similarity measures that were selected by the forward stepwise selection algorithm and regress them on the similarities in different product attributes. To implement such a regression, we collect observed product attributes for all four categories used to estimate our demand model, and we compute a separate similarity measure for each attribute by taking the absolute value of the difference between the attribute values for any pair of goods. We obtain between 10 and 12 product attributes across the four categories. Notably, the sets of product attributes differ across the four categories, which implies that any product attribute-based model would require estimating different coefficients for different

categories. In contrast, our approach uses measures of product similarity that are defined for any category where image and text data are available, thus allowing us to pool information across categories. Having constructed attribute similarities, we then regress each of the two selected image similarity metrics on all attribute similarities.[9]

We present the results from these regressions in Appendix Table A7. We obtain R-squared coefficients of 0.440 and 0.513 when using each of the two selected similarity measures as the dependent variable. We conclude that roughly half of the variation in the selected product similarity measures is explained by similarity in observed product attributes. Our model therefore allows us to capture additional information about product similarity beyond the information contained in observed product attributes. We re-iterate that our model is also able to capture variation related to observed product attributes in a relatively parsimonious way. Whereas our main specification contains two similarity variables, our dataset includes 10-12 attributes per category. In this case, standard demand models would require estimating the distribution of random coefficients for each product attribute, which would involve estimating many more parameters than required by our method. Standard models would also require the researcher to perform numerical integration, which is computationally costly in models with many random coefficients.

5   Conclusion

In this paper, we propose a demand estimation method that allows researchers to estimate substitution patterns from unstructured data such as product descriptions, customer reviews, and product images. We use a series of machine learning models to obtain measures of similarity from these data sources. We then estimate a nested logit model with product-pair specific nesting parameters that depend on the image and text similarities between products. This nested logit model allows us to include text and image similarities into a micro-founded demand system that exhibits closed-form expressions for purchase probabilities. We apply our method to a dataset on choices made by Amazon shoppers across several categories, and we show that our method allows us to recover flexible substitution patterns.

---

[9]More specifically, we regress image similarities on each attribute similarity interacted with a dummy for the category for which this attribute is defined. The only two attributes that are defined for all four categories are brand and price. These two similarity measures are not interacted with category dummies.

References

AMANO, T., A. RHODES, AND S. SEILER (2022): "Flexible Demand Estimation with Search Data," Working Paper.

ARMONA, L., G. LEWIS, AND G. ZERVAS (2021): "Learning Product Characteristics and Consumer Preferences from Search Data," Working Paper.

BERRY, S., J. LEVINSOHN, AND A. PAKES (1995): "Automobile Prices in Market Equilibrium," *Econometrica*, 63, 841–890.

——— (2004): "Differentiated Products Demand System from a Combination of Micro and Macro Data: The New Car Market," *Journal of Political Economy*, 112, 68–105.

BRESNAHAN, T. F., S. STERN, AND M. TRAJTENBERG (1997): "Market Segmentation and the Sources of Rents from Innovation: Personal Computers inthe Late 1980s," *RAND Journal of Economics*, 28, S17–S44.

CER, D., Y. YANG, S. YI KONG, N. HUA, N. LIMTIACO, R. S. JOHN, N. CONSTANT, M. GUAJARDO-CESPEDES, S. YUAN, C. TAR, Y.-H. SUNG, B. STROPE, AND R. KURZWEIL (2018): "Universal Sentence Encoder," .

CHOLLET, F. (2017): "Xception: Deep Learning With Depthwise Separable Convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1251–1258.

CONLON, C. AND J. H. MORTIMER (2021): "Empirical properties of diversion ratios," *The RAND Journal of Economics*, 52, 693–726.

DEVLIN, J., M.-W. CHANG, K. LEE, AND K. TOUTANOVA (2018): "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*.

DOTSON, J. P., M. A. BELTRAMO, E. M. FEIT, AND R. C. SMITH (2019): "Modeling the Effect of Images on Product Choices," Working Paper.

DOTSON, J. P., J. R. HOWELL, J. D. BRAZELL, T. OTTER, P. J. LENK, S. MACEACHERN, AND G. M. ALLENBY (2018): "A Probit Model with Structured Covariance for Similarity Effects and Source of Volume Calculations," *Journal of Marketing Research*, 55, 35–47.

GREMINGER, R., Y. HUANG, AND I. MOROZOV (2023): "Make Every Second Count: Time Allocation in Online Shopping," *Working Paper.*

HAUSMAN, J. A. AND D. A. WISE (1978): "A conditional probit model for qualitative choice: Discrete decisions recognizing interdependence and heterogeneous preferences," *Econometrica: Journal of the econometric society*, 403–426.

HE, K., X. ZHANG, S. REN, AND J. SUN (2016): "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

KIM, J. B., P. ALBUQUERQUE, AND B. J. BRONNENBERG (2011): "Mapping Online Consumer Search," *Journal of Marketing Research*, 48, 13–27.

KOPPELMAN, F. S. AND C.-H. WEN (2000): "The paired combinatorial logit model: properties, estimation and application," *Transportation Research Part B: Methodological*, 34, 75–89.

MAGNOLFI, L., J. MCCLURE, AND A. SORENSEN (2022): "Triplet Embeddings for Demand Estimation," Working Paper.

MCFADDEN, D. (1977): "Modelling the choice of residential location," .

NETZER, O., R. FELDMAN, J. GOLDENBERG, AND M. FRESKO (2012): "Mine Your Own Business: Market-Structure Surveillance Through Text Mining," *Marketing Science*, 31, 521–543.

PINKSE, J. AND M. E. SLADE (2004): "Mergers, brand competition, and the price of a pint," *European Economic Review*, 48, 617–643.

REIMERS, N. AND I. GUREVYCH (2019): "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics.

RUIZ, F. J. R., S. ATHEY, AND D. M. BLEI (2017): "SHOPPER: A Probabilistic Model of Consumer Choice with Substitutes and Complements," *Annals of Applied Statistics*, 14, 1 – 27.

RUSSAKOVSKY, O., J. DENG, H. SU, J. KRAUSE, S. SATHEESH, S. MA, Z. HUANG, A. KARPA-THY, A. KHOSLA, M. BERNSTEIN, A. C. BERG, AND L. FEI-FEI (2015): "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, 115, 211–252.

SIMONYAN, K. AND A. ZISSERMAN (2015): "Very Deep Convolutional Networks for Large-Scale Image Recognition." in *International Conference on Learning Representations*.

SISODIA, A., A. BURNAP, AND V. KUMAR (2022): "Automatically Discovering Visual Product Characteristics," Working Paper.

SMALL, K. A. (1987): "A Discrete Choice Model for Ordered Alternatives," *Econometrica*, 55, 409–424.

SMITH, A. N., P. E. ROSSI, AND G. M. ALLENBY (2019): "Inference for Product Competition and Separable Demand," *Marketing Science*, 38, 690–710.

SZEGEDY, C., V. VANHOUCKE, S. IOFFE, J. SHLENS, AND Z. WOJNA (2016): "Rethinking the Inception Architecture for Computer Vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2818–2826.

Online Appendix

A   Data Collection

In this section, we provide additional details on the data sources used in the paper. We obtain purchase data from the 2019-2020 Comscore Web Behavior Panel, which contains a sample of U.S. households. Households install software meters on their computers and give Comscore permission to track all their Internet activity, including all online purchases they make in major online stores. Because our methodology involves scraping images and textual descriptions of products, it is practical to focus our analysis on one specific online store. We focus on purchases made on Amazon.com which accounts for almost 54% of all purchases. The second-largest store, Walmart.com, accounts only for 8.9% and the third-largest store, Target.com, only for 2.3%. We use the dataset constructed by (Greminger et al., 2023), who classify over 12 million unique Amazon products into narrowly defined categories (e.g., "laptops," "smartphones," and "tablets") by combining store-defined category labels with detailed browsing data. Their dataset also includes daily product prices collected from a third-party dataset (Keepa.com). These price data cover around 95% of all unique Amazon products in the Comscore dataset.

In estimation, we focus on four categories of electronic goods: headphones, tablets, memory cards, and computer monitors. We select categories using the following process. From the raw dataset, we drop around 5% of products for which we do not have price data. Then, to keep the sample of products manageable, in each of the 3,890 categories available in the Comscore dataset, we select products that were purchased at least 15 times. Both observing enough purchases and some temporal price variation are critical for obtaining accurate demand estimates. From the remaining 197 categories, we select the categories of durable goods with the largest number of observed purchases.[10]

B   Text Processing Steps

Before applying our text models, listed in Section 3.2.2, we pre-process our text data as follows. Working with product titles is straightforward because each title is a short text that typically includes only 10-15 words. When assessing similarity based on product descriptions, we merge the text from all bullet points, and apply our models to the merged text. For customer reviews, we transform the text of each review into a separate vector of word occurrences or an embedding, and we average these vectors or embeddings across all reviews of a given product. We then compute the distance between two products by computing the Euclidean distance between the two averaged vectors. Finally, for Q&As, we merge each question and its answer into a single text string, and we then treat each of these Q&A exchanges identically to how we treat customer reviews.

---

[10]Around 9.8% of purchases in our dataset represent consumers who re-visit the same category to make another purchase. To keep our analysis simple, we treat such repeat purchases as independent and do not account for this panel structure in estimation.

For both bag-of-words approaches, we further pre-process text data by removing stopwords and lemmatizing words. We remove stopwords using the standard dictionary of common English words in the NLTK package. We lemmatize words using the WordNet Lemmatizer from the same package, NLTK. Then, we convert each pre-processed text into a vector of word occurrences (weighted word occurrences for TF-IDF), and we compute the Euclidean distance between these vectors for each product pair. In the case of USE and ST models, we first transform each text into a lower dimensional embedding using a pre-trained model. Because both models have a built-in text pre-processing step, we apply these models directly to the unprocessed text data.

- 7" IPS display; 16 or 32 GB of internal storage (add up to 512 GB with microSD)
- Faster 1.3 GHz quad-core processor
- Up to 7 hours of reading, browsing the web, watching video, and listening to music
- Hands-free with Alexa, including on/off toggle
- 1 GB of RAM
- 2 MP front and rear-facing cameras with 720p HD video recording
- Stay on track – Check email, make video calls, update shopping lists, and set reminders. Use your favorite apps like Zoom, Outlook, and OneNote
- 90-day limited warranty
- Enjoy your favorite apps like Netflix, Facebook, Hulu, Instagram, TikTok, and more through Amazon's Appstore (Google Play not supported)

- 8" HD display, 2X the storage (32 or 64 GB of internal storage and up to 1 TB with microSD card) + 2 GB RAM. 10th generation (2020 release).
- All-day battery life - Up to 12 hours of reading, browsing the web, watching videos, and listening to music.
- Now with USB-C for easier charging. Fully charges in under 5 hours (with included cable + adapter).
- 30% faster thanks to the new 2.0 GHz quad-core processor.
- Enjoy your favorite apps like Netflix, Facebook, Hulu, Instagram, TikTok, and more through Amazon's Appstore (Google Play not supported).
- Stay on track – Check email, make video calls, update shopping lists, and set reminders. Use your favorite apps like Zoom, Outlook, and OneNote.
- Hands-free with Alexa, including on/off toggle.
- 2 MP front and rear-facing cameras with 720p HD video recording.
- Dual-band, enhanced wifi.

- 2-year worry-free guarantee: if it breaks, return it and we'll replace it for free.
- Over 20 million kids (and their parents) have enjoyed Amazon Kids+ service. Amazon Kids parental controls allow you to set educational goals, create time limits, and filter content.
- The included 1 year of Amazon Kids+ gives your kids access to thousands of apps, games, books, videos, audiobooks, and educational content from PBS Kids, Nickelodeon, Disney, and more. Your subscription will then automatically renew every month starting at just $4.99 per month plus applicable tax. You may cancel at any time by visiting the Amazon Parent Dashboard or contacting Customer Service.
- Parents can give kids access to more apps like Netflix, Minecraft, and Zoom via the Amazon Parent Dashboard.
- Amazon Kids+ includes thousands of Spanish language books, videos, apps, games, and audiobooks.
- Save up to $70 on a Fire 7 Tablet (not a toy) for kids 3-7, 1 year of Amazon Kids+, a Kid-Proof Case with built-in stand, and 2-year worry-free guarantee, versus items purchased separately.
- Stream through Wi-Fi or view downloaded content on the go with 16 GB of internal storage and up to 7 hours of reading, browsing the web, watching video, and listening to music. Add a microSD card for up to 512 GB of expandable storage.

- 10.2-inch Retina display
- A10 Fusion chip
- Touch ID fingerprint sensor
- 8MP back camera, 1.2MP FaceTime HD front camera
- Stereo speakers
- 802.11ac Wi-Fi
- Up to 10 hours of battery life
- Lightning connector for charging and accessories
- Support for Smart Keyboard and Apple Pencil

Figure A1: **Illustrative Example: Images and Product Descriptions of Tablets.**

24

	Images	Titles	Descr.	Q&A	Reviews
Image-based similarity	1.000				
Text-based similarity (titles)	0.102	1.000			
Text-based similarity (descriptions)	0.317	0.265	1.000		
Text-based similarity (Q&A)	0.313	0.313	0.060	1.000	
Text-based similarity (reviews)	0.260	0.101	0.065	0.602	1.000

Table A1: **Correlations Between Different Similarity Metrics.** The table displays correlations between the best-fitting (lowest AIC) similarity variables from each of the five groups.

C   Correlations between similarity metrics

Text-based similarities (titles)	BOW	TF-IDF	USE	BERT
Bag of Words	1.000			
Bag of Words TF-IDF	0.424	1.000		
Universal Sentence Encoder (USE)	0.011	0.461	1.000	
Sentence Transformer (BERT)	0.148	0.553	0.787	1.000

Table A2: **Correlations Between Different Text-Based Similarity Metrics (product titles).**

Text-based similarities (descriptions)	BOW	TF-IDF	USE	BERT
Bag of Words	1.000			
Bag of Words TF-IDF	0.245	1.000		
Universal Sentence Encoder (USE)	-0.142	0.441	1.000	
Sentence Transformer (BERT)	0.177	0.547	0.666	1.000

Table A3: **Correlations Between Different Text-Based Similarity Metrics (product descriptions).**

Text-based similarities (Q&A)	BOW	TF-IDF	USE	BERT
Bag of Words	1.000			
Bag of Words TF-IDF	-0.134	1.000		
Universal Sentence Encoder (USE)	-0.207	-0.047	1.000	
Sentence Transformer (BERT)	-0.198	-0.040	0.927	1.000

Table A4: **Correlations Between Different Text-Based Similarity Metrics (product Q&A).**

Text-based similarities (reviews)	BOW	TF-IDF	USE	BERT
Bag of Words	1.000			
Bag of Words TF-IDF	0.201	1.000		
Universal Sentence Encoder (USE)	0.580	0.741	1.000	
Sentence Transformer (BERT)	0.643	0.738	0.955	1.000

Table A5: **Correlations Between Different Text-Based Similarity Metrics (customer reviews).**

Image-based similarities	VGG19	ResNet50	Xception	InceptionV3
VGG19	1.000			
ResNet50	0.661	1.000		
Xception	0.380	0.663	1.000	
InceptionV3	0.538	0.661	0.706	1.000

Table A6: **Correlations Between Different Image-Based Similarity Metrics.**

D Similarity metrics and observed product attributes

VARIABLES	(1) Image Sim VGG19	(2) se	(3) Image Sim INC3	(4) se
(Pooled) similarity_brand	0.219	(0.017)	0.066	(0.013)
(Pooled) similarity_price	0.016	(0.007)	0.049	(0.006)
(Headphones) similarity_water_proof	0.034	(0.018)	0.044	(0.014)
(Headphones) similarity_color	0.018	(0.010)	0.030	(0.008)
(Headphones) similarity_connectivity	-0.006	(0.020)	-0.068	(0.015)
(Headphones) similarity_deep_bass	0.012	(0.009)	-0.004	(0.007)
(Headphones) similarity_tanglefree	0.016	(0.026)	0.100	(0.020)
(Headphones) similarity_with_microphone	0.024	(0.012)	0.010	(0.009)
(Headphones) similarity_sweat_proof	-0.046	(0.018)	-0.038	(0.014)
(Headphones) similarity_noise_reduction	-0.009	(0.009)	-0.005	(0.007)
(Headphones) similarity_number_eartips_sets	-0.002	(0.004)	0.000	(0.003)
(Tablets) similarity_memory	-0.001	(0.003)	-0.004	(0.003)
(Tablets) similarity_maximum_resolution	0.044	(0.035)	-0.112	(0.027)
(Tablets) similarity_screen_size	-0.014	(0.010)	0.020	(0.007)
(Tablets) similarity_with_case	-0.046	(0.016)	-0.053	(0.012)
(Tablets) similarity_kids_subs	0.220	(0.034)	0.016	(0.026)
(Tablets) similarity_number_of_cores	-0.006	(0.009)	-0.002	(0.007)
(Tablets) similarity_battery_life	-0.007	(0.014)	0.011	(0.011)
(Tablets) similarity_front_camera_mp	-0.007	(0.006)	-0.021	(0.005)
(Tablets) similarity_back_camera_mp	-0.006	(0.007)	0.001	(0.005)
(Memory Cards) similarity_micro_card	0.227	(0.042)	0.206	(0.033)
(Memory Cards) similarity_flash_memory_type	0.017	(0.121)	0.119	(0.093)
(Memory Cards) similarity_uhs_speed_class	0.035	(0.227)	0.275	(0.175)
(Memory Cards) similarity_personal_computer	0.048	(0.044)	0.088	(0.034)
(Memory Cards) similarity_camera	0.026	(0.045)	0.057	(0.035)
(Memory Cards) similarity_laptop	-0.122	(0.054)	0.098	(0.042)
(Memory Cards) similarity_tablet	0.012	(0.045)	-0.009	(0.035)
(Memory Cards) similarity_magnetic_proof	0.002	(0.051)	-0.027	(0.040)
(Monitors) similarity_blue_light_filter	0.013	(0.035)	-0.030	(0.027)
(Monitors) similarity_frameless	0.038	(0.028)	0.019	(0.022)
(Monitors) similarity_tilt_adjustment	0.025	(0.037)	0.007	(0.029)
(Monitors) similarity_height_adjustment	-0.168	(0.029)	-0.042	(0.023)
(Monitors) similarity_flicker_free	-0.058	(0.026)	-0.033	(0.020)
(Monitors) similarity_built_in_speaker	-0.142	(0.031)	-0.084	(0.024)
(Monitors) similarity_wall_mountable	-0.076	(0.027)	-0.030	(0.021)
(Monitors) similarity_curved_screen	-0.028	(0.027)	-0.007	(0.021)
(Monitors) similarity_adaptive_sync	-0.025	(0.026)	-0.033	(0.020)
(Monitors) similarity_refresh_rate	0.000	(0.000)	0.001	(0.000)
Observations	710		710	
R-squared	0.513		0.440	
F-Statistic	17.20		12.81	

Standard errors in parentheses

Table A7: **Regressions of image similarities selected by the model on the observed similarities in product attributes.**