DESCRIBING ROSÉ:

AN EMBEDDING-BASED METHOD FOR MEASURING PREFERENCES

Anirban Mukherjee

Cornell University

Hannah H. Chang

Singapore Management University

Date: 3 June 2021

ABSTRACT

Many products and services are best (and typically) described in prose. In extant preference-measurement methods, however, due to the challenge of numerically representing prose in econometric models, products can only be described to participants and portrayed in the utility model as a list of attributes. In this research, the authors develop an embedding-based utility model and preference method that addresses this limitation; in it, products are described to participants in (unstructured) prose. The proposed method provides three benefits: (1) in it, products can be described more completely, (2) it improves study realism, and (3) it enables a more detailed measurement of preferences. The authors employ the proposed method to measure consumer preferences in Australia, New Zealand, and the United States for wines made in 427 wine-growing regions in 44 wine-growing countries, from 708 wine-grape varietals. They find the proposed model has superior in-sample fit and generates better out-of-sample predictions than benchmark models. Importantly, the method is able to capture differences in consumers' valuation for wines (products) that are observationally equivalent in categorical attributes, and therefore indistinguishable in classical categorical variable-based analysis. The use of the proposed model as a decision support system for marketing activities is demonstrated.

Keywords: preference measurement; embedding; machine learning; marketing.

"Aromas of white spring flower, Bartlett pear and citrus waft out of the glass. The racy, refreshing palate is full of energy, offering crisp yellow-apple, lemon drop and orange zest flavors balanced by vibrant acidity. A perlage of small, refined and continuous bubbles provides the silky backdrop."

—Kerin O'Keefe, Wine Enthusiast *(2019), on its top-ranked wine of 2019,*
Nino Franco NV Rustico Brut (Valdobbiadene Prosecco Superiore)

Many products[1] are best described (and typically described) in prose. Prominent examples include products that consumers choose, buy, and use primarily for the experience provided (Holbrook and Hirschman 1982), such as entertainment (e.g., rock concert), travel (e.g., tourism packages to India), and hospitality (e.g., upscale restaurants). As consumer experiences are rich and nuanced, descriptions of these products are best delivered in prose—restaurants describe their ambience, travel operators describe the sight-seeing experience at an exotic locale, and theme parks describe the exhilaration of a rollercoaster ride. Similarly, tasting notes of wines, as seen in the opening quote, are presented in prose to capture the sensory nature of wine consumption.

In extant preference-measurement methods (e.g., conjoint analysis), however, due to the challenge of numerically representing prose in econometric models, products can only be described to participants and portrayed in the utility model as a list of product attributes (Green et al. 2001, Toubia et al. 2004). In this paper, we develop and propose a novel embedding-based preference-measurement method and utility model that addresses this limitation; in our proposed method, products are described to participants in prose.

Our primary contribution lies in developing an embedding-based utility model to infer participants' valuations of product attributes and attribute-levels from their responses to unstructured prose product descriptions. The model is based on the mathematical theory of an embedding—an injective map (i.e., a one-to-one function) from a set of objects to points on a

[1] We refer to products, services, and other market offerings as "products".

normed vector space, the axes of which encode important and relevant information about the objects—as applied to products. To construct a product embedding, we develop and describe embedding algorithms that we apply to prose product descriptions. We incorporate the product embedding in the utility model to form an embedding-based utility model. We estimate the utility model on participants' choices to infer preferences. Relative to extant preference-measurement methods, our proposed method provides three key benefits.

First, it enables products to be described in prose, and hence more completely than if they were described using categorical variables. Specifically, for many products, a product description in prose is more complete than a product description using categorical variables (Chung and Rao 2012, Toubia et al. 2019). For example, the opening quote describes the experience of tasting Nino Franco NV Rustico Brut, a sparkling white wine from Italy. It would be difficult to construct a categorical system to capture the intricate flavors and sensory properties of this wine and all the other wines in the market. Even if such a categorical system could be derived for wines, it would have far too many attributes and attribute-levels to be useful in a preference-measurement study and would need to be simplified considerably. As such, categorical systems in preference measurement lead to a trade-off between completeness and complexity in data collection and analysis (pp. 233–237, Toubia et al. 2007a). Thus, for example, Toubia et al. (2007b) conduct a real-world field study for a "Napa Valley-based [wine bottle] closure manufacturer and cooperating U.S. wineries" that includes only two categorical attributes to account for the taste of a wine: the type of wine ("dry white," "aromatic white," "dry red," or "blush red") and the country/region of origin ("Australia/New Zealand," "France," "Sonoma/Napa," or "Chile/Argentina").

Second, describing products in prose enhances study realism. Extant studies in consumer behavior have shown that presenting the same product information in different formats (e.g., numerically vs. verbally, matrix display vs. written sentences) alters consumers' decision processes (Kleinmuntz and Schkade 1993, Payne et al. 1993). For example, when product information is presented numerically, consumers use more compensatory processing (Stone and Schkade 1991) and more within-attribute comparisons (Huber 1980), and they exhibit greater attraction effect in choices (Frederick et al. 2014), than when the same information is presented non-numerically (verbally). Specific to our context, in choice-based conjoint, the magnitude of partworth estimates is dramatically influenced by the mode in which information is presented to participants (Hauser et al. 2019). Therefore, for participant responses in a preference measurement study to reflect real-world behavior, products should be presented in the same manner in a study as they are presented in the real-world marketplace (see Morales et al. 2017). Many products are described in prose in the real-world marketplace when that is the more suitable and complete way to convey information (e.g., wines, as illustrated in the opening quote). For such products, our method enhances study realism, thereby enhancing the accuracy and precision of the research.

Third, our method enables a more detailed measurement of consumer preferences than is feasible using extant methods, as extant methods are limited by the number of attributes and attribute-levels that can be included in a study (Toubia et al. 2007a). Specifically, if products are described to participants using categorical variables, to curb respondent load, there is a natural limit to how many attributes and attribute-levels can be included in a study (cf. Malhotra 1984, Exhibit 1). Whilst much progress has been made in the development of more efficient study designs including partial profile conjoint designs (Netzer et al. 2008), the addition of many categorical attributes dramatically increases the amount of participant data required for the study. These issues are particularly salient for products with many nuanced, sensory characteristics (such as wines), where a more complete description of the product requires a categorical system with multiple attributes and attribute-levels. Importantly, in our method, the vector space of the product embedding is constructed to capture product differentiation. This enables the specification of a much more parsimonious embedding-based utility model than the canonical categorical-variable based utility model, thereby reducing data requirements and enhancing cost-effectiveness.

To establish and showcase these benefits, we use our method to investigate consumer preferences for wines made in 427 wine-growing regions in 44 wine-growing countries, from 708 wine-grape varietals. We situate our study in the wine industry because it exemplifies the information needs in industries where firms want to know how consumers value products with complex attributes that have many attribute-levels (Jaeger et al. 2009). Other examples of industries with similar information needs include hospitality, entertainment, and tourism. We conduct a large-scale preference-measurement study in which 1,000 participants from Australia, New Zealand, and the United States (henceforth US) chose a preferred wine between 32 randomly selected pairs of prose wine descriptions.

In particular, wines are bought, stocked, and sold globally on the basis of three categorical product attributes—(1) the region and (2) country where the wine was made, and (3) the varietal (or blend) used to make the wine (Arias-Bolzmann et al. 2003). It is common for wine distributors and retailers to carry an extensive assortment of wines from many regions and countries, which are made from many varietals. For example, wine.com (a popular US-based internet wine retailer) carries 408,103 different wines from 26 countries. Furthermore, each country has a myriad of wine regions where wine is made from a multitude of varietals. For instance, of the 82,129 different French wines carried by wine.com, 48,612 are red wines that are made from 26 different red wine grape varietals; 28,745 are white wines that are made from 25 different white wine grape varietals; the rest are sparkling, rosé, or dessert wines that are made from other grape varietals.

In order to make effective marketing decisions (e.g., to determine the product assortment), wine brands, distributors, and retailers (such as wine.com) need to know how consumers in different countries value all levels of these attributes. Due to cost concerns, however, it is impractical for firms to use extant methods to measure preferences (i.e., partworths) for 1,179 attribute-levels. Instead, it is typical for firms to aggregate attribute-levels, thereby reducing data-collection costs but also reducing the specificity, and hence the information value, of the research. In contrast, our method allows us to cost-effectively estimate individual-level partworths for all attribute-levels. Thus, our method directly addresses the substantive challenge and managerial need identified in prior papers of developing practical and feasible methods for preference measurement in products with many attributes and attribute-levels (Park et al. 2008, Scholz et al. 2010, Chung and Rao 2012).

Even though these three categorical attributes are far too detailed for use without simplification in extant preference-measurement methods, the attributes may still be insufficiently detailed to adequately discriminate among wines. In general, many products (e.g., experiential products) that are observationally identical in common categorical descriptors often differ in important and relevant ways. For example, in the restaurant industry, it is common for restaurants that are geographically co-located and serve the same style of cuisine to differ in other respects including ambience, service quality, and food quality (Athey et al. 2018). Central to our research, such differences are reflected in the (complete) prose description of these products, which enables our embedding-based approach to assess consumer preferences more accurately and in greater detail.

To empirically investigate this issue, we consider four prominent "types"[2] of iconic red wines—(1) red wine from Bordeaux (France); (2) cabernet sauvignon wines from California (US); (3) sangiovese wines from Tuscany (Italy); and (4) syrah wines from

[2] We use "type" to indicate the wines are from the same region and country, and made from the same varietal.

Washington state (US). These four types of wines are economically and oenologically significant. For example, Bordeaux produces more than 500 million bottles of wine annually, about half of which is exported out of France and consumed globally; the wine industry is the largest employer and the largest industry in the Gironde department of Aquitaine, where Bordeaux is located (Ashenfelter 2010).

Our findings show large and important differences in consumers' valuations of different wines of the same type. Importantly, these differences would be missed by conventional categorical variable-based approaches, which would classify these wines as identical. Moreover, the extent to which the categorical attributes accurately and precisely capture consumer preferences varies across consumers—some consumers have relatively similar valuations, while others have relatively dissimilar valuations, of different wines of the same type.

Because we specify a choice model, our model and method can form the basis for decision support tools that assist managers in many marketing activities such as segmentation, targeting, and promotions. To showcase the managerial benefits of the increased specificity of our proposed embedding-based utility model, we construct a decision support system from the perspective of a brick-and-mortar wine retailer (as brick-and-mortar represents about 90% of wine retail sales in the US; Briscoe 2020). We identify wines that are likely to sell best, and that are therefore optimal for the wine retailer to carry. To do so, we compute participants' valuations of the wines of each type and use that to choose among wines. Importantly, this analysis cannot be conducted effectively utilizing extant preference-measurement methods that employ a categorical-based utility model as these methods only measure how much participants value each type of wine and do not capture differences among wines of the same type. We find that the wines selected by our method are much more likely to be chosen by consumers than wines selected at random of each type.

The rest of our paper is organized as follows. In the next section, we define and describe the mathematical theory of embeddings, develop the embedding algorithms that we utilize in our research, and outline a utility model that incorporates a product embedding. We then describe our study, in which we employ our embedding-based method to investigate consumer preferences for wines. The last section concludes.

METHOD

We propose a new preference-measurement method in which we present participants with two products that are described in prose and ask participants to choose between them. The key modeling challenge in our research is to derive a utility model that can be estimated on participants' choices between (unstructured) prose descriptions. To address this challenge, we draw on two distinct literatures—the literature on embeddings in computer science and machine learning, and the literature on choice modeling in economics and marketing—to develop a novel embedding-based utility model. As the development of our utility model requires the definition and description of embeddings and embedding algorithms, we organize this section as follows. First, we discuss the mathematical theory of embeddings as it pertains to our research question. Next, we describe the embedding algorithms that we apply to prose product descriptions to construct product embeddings. Finally, we develop our utility model, which incorporates a product embedding.

Embedding

A vector space is a set of vectors on which the operations of addition and multiplication are defined. A normed vector space is a vector space that is equipped with a norm. An embedding is an injection or an injective function (i.e., a one-to-one function) from a collection of objects to a normed vector space. A product embedding is an injection from products to a normed vector space. The location on the vector space corresponding to a product is its vector representation.

Importantly, not all vectors form a vector space and not all vector representations constitute an embedding. For example, it is typical in marketing and economics to use dummy variable vectors to represent categorical variables. Dummy variable vectors are not closed under element-by-element addition. Furthermore, the distance (e.g., the Euclidean distance) between two dummy variable vectors is not meaningful and does not constitute a norm. Therefore, the dummy variable vectors do not form a normed vector space, and a dummy variable representation is not an embedding.

In this research, we advocate using a product embedding to incorporate non-numerical product attributes in a choice model for two reasons. First, the restriction that the numerical (vector) representations of non-numerical product attributes form a normed vector space is crucial, as the operations of addition and multiplication combined with the existence of a norm is the minimal structure required to specify a general mathematical function on a set of vectors. Second, an embedding provides a complete account of the prose product descriptions in the utility model, which enables us to infer participants' preferences from their choices.

To construct a product embedding, we develop and apply embedding models to the prose product descriptions shown to participants. Embedding models map objects (e.g., words, documents, etc.) to locations in a normed vector space where direction and magnitude correspond to an object's meaning. For example, word-embedding models construct an embedding such that the vector space location of a word corresponds to its semantic meaning. Consider the words *king* and *queen*. Embedding models locate these words in a vector space (denote the word-to-vector-space mapping as "Word_Vector") such that Word_Vector (*king*) – Word_Vector (*man*) + Word_Vector (*woman*) = Word_Vector (*queen*).

Other embedding models operate on sequences of words. For example, consider two documents (sequences of words) describing the *biography of a king* and the *biography of a queen*. Embedding models locate these documents in the vector space (denote the document-to-vector-space mapping as "Document_Vector") such that Document_Vector (*biography of a king*) – Word_Vector (*man*) + Word_Vector (*woman*) = Document_Vector (*biography of a queen*). Thus, document-embedding models develop a representation whereby the meaning of a document is encoded in its vector representation.

By applying embedding models to prose product descriptions, we construct a product embedding where the location on the vector space that represents a product corresponds to its attributes as detailed in its prose description. This enables us to specify products' vector space locations as their numerical (vector) representations in the utility model.[3]

Embedding models that operate on sequences of words can be classified into non-neural models, feedforward neural models, and recurrent neural models. The three classes of models differ in both the flexibility and the data requirements of the generative model. This has important downstream consequences for the utility model. Specifically, on the one hand, more flexible embedding models are able to encode more information in the vector space. Therefore, they may lead to a more precise capture of preferences. On the other hand, more flexible embedding models also require more training data. If the training data in a specific application is a bottleneck, less flexible models may outperform more flexible models. Therefore, in our empirical study, we adapt a state-of-the-art model from each class of models and compare on the basis of fit and predictive performance among the resulting utility models.

Non-neural Embedding

[3] The embedding algorithms we describe and employ are sufficiently efficient that a vector space with as few as 5 dimensions suffices to describe products in our empirical study.

Non-neural embedding models employ a bag-of-words approach and depict the generation of the words as a function of chance and the relevance of a word (e.g., the word “tart”) to the document (e.g., the taste of the wine in the tasting note). We adapt to context a state-of-the-art unsupervised model by Arora et al. (2017) with a two-component generative process. For product *g*, word location *l*, the first component of the model occurs with probability α and the second component of the model occurs with probability $(1-\alpha)$. If the first component occurs, word *w* is generated independent of the attributes of product *g* and the words that are generated before or after location *l*. If the second component occurs, word *w* is generated at location *l* with multinomial logit probability determined by the cosine distance of the vector representation of a word in the vocabulary (v_w) from the vector representation v_g of product *g*. Note that consistent with the bag-of-words assumption, in both components the probability that word *w* is generated is independent of location *l*.

(1) $$Pr(w|g,l) = \alpha p(w) + (1-\alpha)\frac{\exp(\langle v_w, v_g \rangle)}{\sum_{z \in V} \exp(\langle v_Z, v_g \rangle)}.$$

To derive a tractable maximum likelihood estimator, we assume that the vocabulary is distributed uniformly on the unit sphere and hence that a small change in v_g has a negligible effect on $\sum_{z \in V} \exp(\langle v_z, v_g \rangle)$. Replacing $\sum_{z \in V} \exp(\langle v_z, v_g \rangle)$ with *Z*, a normalizing constant, we get:

(2) $$l(v_g|w) = \log\left[\alpha p(w) + (1-\alpha)\frac{\exp(\langle v_w, v_g \rangle)}{Z}\right].$$

The first-order approximation to the maximum likelihood estimator of the model is the (weighted) mean of the (normalized) vector representations of the words used to describe product *g:*

(3) $$argmax_{v_g : \|v_g\|=1} l(v_g|w) \approx \frac{(1-\alpha)/\alpha Z}{p(w)+(1-\alpha)/\alpha Z} v_w.$$

Intuitively, the method places less weight on frequently occurring words (e.g., prepositions) that are less informative and greater weight on infrequently occurring words (e.g., "merlot") that are more informative.

Importantly, this model does not require any training data. Therefore, by using this model to form the product embedding (and the utility model), our proposed method can be deployed in data-scarce contexts such as when a product is very novel or very niche, and therefore, when product descriptions are hard to obtain and analyze.

Feedforward Neural Embedding

Feedforward neural embedding models generalize the generative process to include the influence of neighboring words (to a word location). For example, the word "tart" in wines relates to acidity. In a feedforward neural model, the generative probability of the word "tart" in a word location differs by whether the word "acidity" is present or absent in the neighborhood of "tart" in a wine description.

We adapt to context a model proposed by Le and Mikolov (2014) that employs a feedforward neural model to treat the emanation of words as a non-linear function of the vector representation of a document and the neighboring words. The architecture of the model is described in detail in Web Appendix A1.

The feedforward model adds local information describing neighboring words and relaxes the parametric restrictions imposed by the cosine distance function in the non-neural model. Specifically, the probability that word w is emanated in location l of the product description of product g, $Pr(w|g,l)$, is:

$$(4) \qquad Pr(w|g,l) = \mu_{ff}\left(v_{fg}, w_{g(l-2)}, w_{g(l-1)}, w_{g(l+1)}, w_{g(l+2)}\right),$$

where v_{fg} is the representation of wine g in the feedforward embedding, μ_{ff} is the transfer function of the model, and $w_{g(l-2)}, w_{g(l-1)}, w_{g(l+1)}, w_{g(l+2)}$ are the neighboring words in the quintagram context window.

v_{fg} and μ_{ff} are learned from the data. In particular, the feedforward neural model is a universal function approximator—it has the capacity to learn any (Borel-measurable) generative process of product descriptions to any desired degree of accuracy (Hornik et al. 1989). As we apply equation (4) to the verbal description of products, consequently the representation (v_{fg}) and the transfer function (μ_{ff}) learn to express the extent to which product descriptors (such as "tart") apply to product g. Thus, the collection of all representations ($v_{fg}, \forall g$) is the feedforward product embedding, as it captures all information pertaining to the prose description of the products.

Recurrent Neural Embedding

Recurrent neural embedding models generalize the generative process of feedforward models to encompass the influence of all words in the sequence, including those outside the local neighborhood of a word location. For example, in a recurrent neural embedding model, the generative probability of the descriptor "tart" in a word location differs by whether the descriptor "acidity" is present or absent in the remainder of the wine description.

To design a recurrent neural embedding model, we employ the encoder-decoder architecture proposed by Sutskever et al. (2014) for sequence-to-sequence learning: (1) the recurrent encoder layer takes as input a sequence of words and outputs a state vector; and (2) the recurrent decoder layer takes the state vector from the encoder layer and outputs a sequence of words. The state vector is the only link between the encoder and decoder layers. Therefore, for any input sequence of words, the corresponding encoded state vector is its vector-space representation. Due to the flexibility and expressiveness of recurrent neural networks, similar recurrent neural network embedding algorithms have been adopted widely in natural language processing. For example, Palangi et al. (2016) propose an algorithm for document retrieval that is based on the similarity of the vector representation of a query in a query embedding (formed by mapping queries to a vector space) to the vector-space representation of documents in a document embedding (formed by mapping documents to a vector space).

We use the model in an autoencoder configuration (where a model is trained to reconstruct an input sequence), thereby ensuring that the model learns to encode the information contained in the input sequence in the state vector. We update Sutskever et al.'s (2014) model in two ways to improve its performance in the preference-measurement task. First, we use Gated Recurrent Units (GRU), a type of recurrent neurons developed to address the vanishing-gradient problem in textual data. Second, we use a bidirectional-layer architecture to capture bidirectional ordering dependencies in textual data. The architecture of our model is described in detail in Web Appendix A2.

For any sequence of words $\{w_{g1}, \dots, w_{gl}, \dots, w_{gK}\}$ describing product g, the model describes:

$$Pr(w|g,l) = \mu_r\left(v_r\left(w_{g1}, \dots, w_{gl}, \dots, w_{gK_g}\right), w_{g(l-1)}\right), \tag{5}$$

where $v_r\left(w_{g1}, \dots, w_{gl}, \dots, w_{gK_g}\right)$ is the state vector from the encoder, μ_r is a transfer function, and K_g is the number of words in the description of product g.

Similar to the feedforward model, v_r and μ_r are learned from the data. In particular, the recurrent neural model is a universal function approximator of sequential data (Schäfer and Zimmermann 2006). As we apply equation (4) to the verbal description of products in a study, consequently the product representation function (v_r) and the transfer function (μ_r) learn to express the extent to which each descriptor (e.g., tart) applies to product g. Similar to the feedforward embedding, the collection of representations of all products in the data ($v_r(w_g), \forall g$) is the recurrent product embedding.

In equation (5), the emanation probability varies with all words in a product description ($w_{g1}, \dots, w_{gl}, \dots, w_{gK}$) and not just with the neighboring words in a local context window ($w_{g(l-2)}, w_{g(l-1)}, w_{g(l+1)}, w_{g(l+2)}$). Hence, the product vector in the recurrent neural embedding model reflects both global and local information, whereas the product vector in the feedforward model reflects only local information.

Embedding-based Utility Model

Product attributes can be classified into numerical product attributes (e.g., product size) and non-numerical product attributes (e.g., wine taste; Chung and Rao 2012). Numerical product attributes can be directly incorporated into the utility function and are typically not described in prose. Non-numerical product attributes are described in prose and are captured through the product embedding. Therefore, we consider the representation of the numerical and non-numerical product attributes separately. We specify the following model:

(6) $$u_{ij} = f\left(\beta_i, x_j^{vec}, x_j^{num}\right) + \varepsilon_j,$$

where u_{ij} is the utility to consumer *i* from choosing product *j*, β_i is a preference vector that describes the preferences of consumer *i*, x_j^{vec} is the vector-space representation of the non-numerical attributes of product *j*, x_j^{num} is a vector of the numerical attributes of product *j*, and ε_j is the error term. *f* is a mathematical function; β_i is distributed in accordance with a specified mixing distribution.

There are no restrictions on the utility specification (*f*) that are specific to our model and application; many econometric models conform to equation (6). As such, a large body of literature in marketing and economics discusses utility specification in structural models (Chintagunta et al. 2006). Therefore, in this research, for brevity and expositional clarity, we assume *f* is additive and separable, and abstract from numerical product attributes. In particular, we specify $f\left(\beta_i, x_j^{vec}, x_j^{num}\right) = x_j^{vec}\beta_i'$ in our empirical study. This functional form is not integral to our method and model, which can be used to specify many functional forms.

Two axioms govern the inclusion of numerical representations of attributes in random utility models (see PCS 5.1 and PCS 5.2 in McFadden 1981): (1) existence—the numerical representations exist for all products; and (2) uniqueness—the numerical representations are unique for products with unique attributes. The vector representations from a product embedding satisfy both axioms—all prose descriptions have a numerical representation, and the representations are unique to unique prose descriptions. Therefore, equation (6) describes a model that is consistent with random utility maximization.

Similar to conjoint analysis, we develop our method for marketing managers to identify consumer trade-offs among a set of categorical product attributes that describe the product. For example, in global wine trade, wines are typically described by three categorical product attributes—region, country, and varietal. Therefore, in our empirical study, we measure the extent to which consumer valuations depend on these attributes, as this information is likely to be useful to wine brands, retailers, and distributors for effective marketing decision-making.

To infer how each attribute contributes to consumers' utility, and hence how consumers trade off between attributes, we use the vector representations of products to infer the locations of the attribute-levels on the normed vector space. Specifically, given *K* attributes with L_k levels each, we model the vector representation of product *j* as:

$$x_j = x_0 + \sum_{k=1}^{K}\left(\sum_{l=1}^{L_k} x^{kl}\gamma_j^{kl}\right) + \zeta_j, \tag{7}$$

where x_j is the vector representation of product *j*, x^{kl} is the vector representation of level *l* of attribute *k*, γ_j^{kl} is a dummy variable that indicates if level *l* of attribute *k* is present in product *j*, and ζ_j is the error term. Note that for brevity, we specify a linear functional form in equation (7). Our model and method do not require this specific functional form; many functional forms relating the representation of a product to the representations of attribute-levels are admissible in our framework.

It follows that the contribution to the utility of consumer i from attribute k with level l (u_{ikl}) is:

$$u_{ikl} = f\left(\beta_i, x^{kl}, x_j^{num}\right) = x^{kl}\beta_i{}'. \tag{8}$$

In equations (6) and (8), the number of parameters in the utility model (i.e., the cardinality of β_i) varies with the dimensionality of the embedding but does not vary with the number of attribute-levels included in the study. Instead, partworths are inferred from the locations of attribute-levels that are learned from the product descriptions, and from a consumer's utility parameters on the vector space. In contrast, in the canonical categorical-attribute-based utility model, the number of parameters increases linearly with the number of attribute-levels, as a set of parameters is learned from participant responses for every attribute-level included in the model. Therefore, in products with more attribute-levels, as is typical in products with complex attributes, our proposed embedding-based model is more parsimonious than extant categorical-attribute-based models. This, in turn, reduces the data requirements of the study, thereby facilitating analyses in situations and contexts (such as our study on wines) where extant models and methods are cost prohibitive.

While the product embeddings constructed using the described embedding models are mathematically appropriate for our utility model, we further map them to a vector space with an orthonormal basis for three reasons. First, the transformation helps ensure that groups of similar products load strongly on one axis and weakly on others (p. 59, Stewart 1981; p. 134, Wedel and Kamakura 2012). Second, the transformation maximizes the angle between the basis vectors and therefore maximizes the dispersion of products in the vector space, aiding estimation. Third, as the basis vectors in the new vector space are of unit length and orthogonal, it makes the coefficients of our utility model comparable, which aids in interpretability.

Specifically, we compute the (compact) Singular Value Decomposition of the stacked product representations and use the left-singular vectors as product representations in a utility model. Web Appendix B shows that this transformation ensures that the new product representations form a vector space with an orthonormal basis, the mapping is lossless, and partworth definitions on the new vector space are mathematically equivalent to partworth definitions on the original vector space.

EMPIRICAL APPLICATION: A STUDY OF WINE PREFERENCES

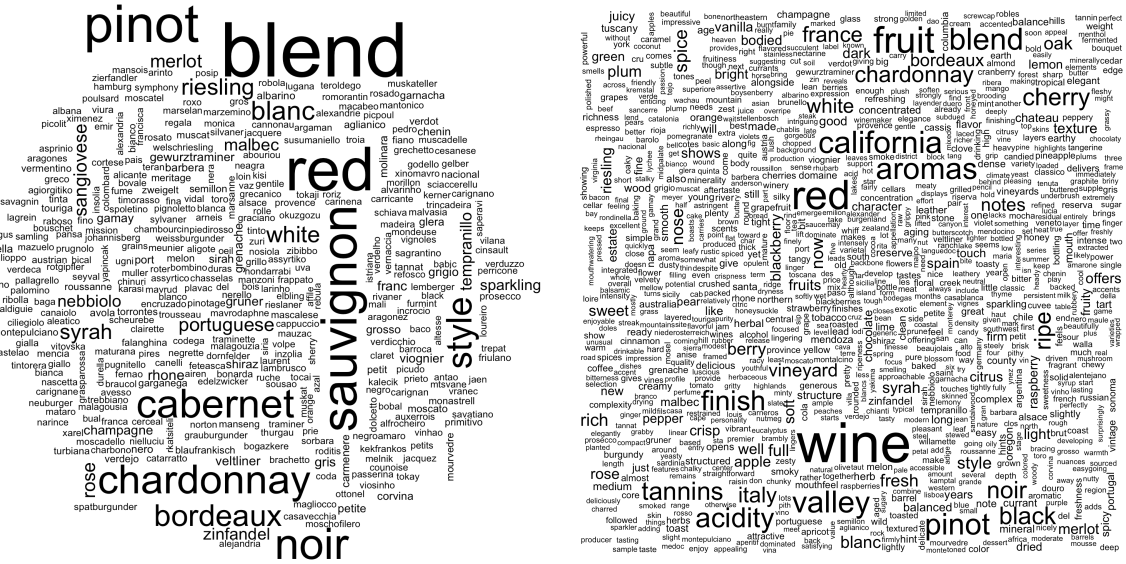
We situate the empirical application of our proposed method in the context of wine for three key reasons. First, wine is typically used to exemplify products whose primary value lies in the sensory and affective experiences it provides (e.g., Cooper-Martin 1991, Hadj Ali and Nauges 2007; Gilovich and Gallo 2020). Second, wines are highly differentiated (Lynch and Ariely 2000), with distinctive, complex, and nuanced product attributes (Jaeger et al. 2009; MacNeil 2015). As such, it is difficult to use categorical variables to portray wines with adequate richness and granularity. Third, wines are often described in the marketplace in prose to express their idiosyncratic aroma and flavor profiles. In addition, the wine industry is important to the global economy, spanning agriculture, manufacturing, and trade sectors. Global wine sales amounted to US$370 billion in revenue and 27.3 billion liters in volume in 2019 and are estimated to reach US$428 billion and 27.9 billion liters in 2023 (Statista 2020).

We conduct a preference-measurement study from the perspective of alcohol companies (e.g., Pernod Ricard, E & J Gallo, Diageo) that need to know consumers' valuations of wines made in different regions and countries, from different varietals, in different markets. This information is critical in the wine industry, as these attributes determine how the wine is valued (MacNeil 2015). For example, wines from some regions (such as Champagne in France) are valued more highly than similar wines from other regions (such as Asti in Italy). As such, it is typical for wines in the global market to be traded, purchased, and consumed on the basis of these three attributes (MacNeil 2015).

Product Description Data

We obtain a large-scale dataset of wine descriptions from Kaggle (a repository of datasets) with the name, region, country, varietal (or blend), and tasting notes of 119,955 wines from 44 countries. The tasting notes of a wine are derived from a blind taste test (i.e., without knowledge of pricing, production size, label, provenance, and varietal) of each wine by experts, who describe the sensory experience of consuming the wine. These are the tasting notes by real wine experts as originally published in *Wine Enthusiast* magazine (one of the largest global wine magazines), which are commonly used by retail outlets to describe wines to consumers. In Figure 1, the left panel is a word cloud of the 708 labels of varietals and blends, and the right panel is a word cloud of the top 1,000 taste descriptors. Figure 1 exemplifies the expansive and elaborate lexicon used to describe wines.

Figure 1: Word Cloud of Varietals



Note: The left panel is a word cloud of all (706) labels of varietals. The right panel is a word cloud of the 1,000 most common taste descriptors.

The product description data requirements of our method comport with data availability in industries such as restaurants, hotels, and tourism where product descriptions are available on websites (such as TripAdvisor) that are used by consumers globally. Another important source of large-scale product description data is crowdfunding (e.g., Kickstarter), where entrepreneurs describe new products in order to attract consumers and investors (Mukherjee et al. 2019). Moreover, a product embedding can be formed without using any product description data by using pretrained word embeddings and the non-neural embedding model (which we describe and empirically assess in a later section). In this case, the method requires only the product descriptions presented to participants in the study. In sum, while our method benefits from the use of more context-specific (product description) data, it can be easily adapted and used in data-scarce environments (such as novel products in nascent industries).

We pre-process the product descriptions according to standard practice as follows. To remove special characters, we transcode all strings to ASCII (Latin-1). Next, we convert all characters to lowercase, remove punctuation, expand contractions (e.g., *won't* to *will not*), remove numbers, and remove stop words (i.e., frequently occurring words that are relatively uninformative, e.g., *the*). We also remove words that occur infrequently (less than 50 times in our entire data).

Trained Embeddings

Word Embedding

We train a 10-dimensional[4] GloVe word embedding on the words used to describe wine. During training, we utilize the functional form and hyperparameter values recommended by Pennington et al. (2014). We use Adam, an extension of two classic

[4] We trained several embeddings where we varied the number of dimensions. We evaluated the embeddings qualitatively on accuracy and quantitatively on the fit of the utility model. We found that our data favor a 10-dimensional embedding. In our paper, we compare a 10-dimensional embedding with a 5-dimensional and a 15-dimensional embedding. Other results are available from the authors.

stochastic gradient-descent algorithms (AdaGrad and RMSprop), to train the embedding models (Kingma and Ba 2014). We use cosine similarity to measure the similarity of vector representations (Wilson and Schakel 2015).

To establish that the GloVe word embedding captures meaning specific to word usage in the wine-tasting notes, we conduct the following qualitative assessment. Red wines are commonly described using descriptors reminiscent of black fruit and blue fruit (see the deductive tasting grid of the Court of Master Sommeliers[5]). We chose blackberry from black fruit and blueberry from blue fruit as focal fruits. Blackberry and blueberry are predominantly used to describe red wines. Table 1 lists the words that are the most similar in vector representation to blackberry and blueberry (Columns 1 and 2) and to both fruits jointly (Column 3). Table 1 lists other fruits (e.g., cassis, currant, and plum) and other closely related tastes (e.g., chocolate, mocha, and licorice) to black and blue fruit, suggesting that the model achieves a reasonable encoding of wine taste descriptors in a relatively low-dimensional (10-dimensional) vector space.[6] Furthermore, the addition of word vectors is sensible, as the tastes in Column 3 reflect what is common between the two focal tastes.

[5] The Court of Master Sommeliers is the premier examination body for sommeliers. The deductive tasting grid used for sommelier examination is available at: https://www.mastersommeliers.org/ [accessed April 23, 2021].

[6] The word "milk" appears in the list because "milk chocolate" is a common flavor and aroma descriptor in red wines (such as Cabernet Sauvignon and Pinot Noir). For example, the famed wine critic James Suckling described the taste of Petrolo Toscana Galatrona 2011 as "…a phenomenal pure merlot with blueberries, raspberries and hints of milk chocolate" (https://www.jamessuckling.com/tasting-notes/24836/petrolo-toscana-galatrona-2011).

Table 1: Taste Descriptors in the GloVe Embedding

Focal descriptor		*blackberry*	*blueberry*	*blackberry* and *blueberry*
Order of similarity of taste descriptor	*1st*	blackberry	blueberry	blackberry
	2nd	chocolate	mocha	blueberry
	3rd	cassis	blackberry	chocolate
	4th	mocha	chocolate	mocha
	5th	black	spices	cassis
	6th	blueberry	cola	black
	7th	dark	milk	cherry
	8th	meaty	cherry	licorice
	9th	licorice	jam	spices
	10th	cherry	cinnamon	milk

Note: List of the 10 most similar taste descriptors by the cosine similarity of GloVe representations to the focal descriptors in the column headings. Note that "milk" relates almost entirely to the use of "milk chocolate" as a descriptor in red wines.

Product Embedding

We use our GloVe word embedding and the value of the hyperparameter ($a = 0.0001$) recommended by Arora et al. (2017) to construct the non-neural embedding. We use TensorFlow, a machine learning library designed for neural networks, to train both the feedforward embedding and the recurrent embedding. Web Appendix C provides details on how we implement these models in TensorFlow.

To establish that the product embeddings capture the information in the wine-tasting notes, we conduct the following qualitative assessment. Table 2 lists the three wines that are the most similar in vector representation (in cosine distance) to the wine described in the quote at the beginning of our paper (Nino Franco NV Rustico Brut). The first row of Table 2 presents the tasting notes of the focal wine. The second, third, and fourth rows present the tasting notes of the three wines with the closest (most similar) numerical representations to the focal wine in the non-neural, feedforward (neural), and recurrent (neural) embeddings.

Table 2: Wines in the Product Embedding

	Non-Neural Embedding	Feedforward Neural Embedding	Recurrent Neural Embedding
Focal wine	Aromas of white spring flower, Bartlett pear and citrus waft out of the glass. The racy, refreshing palate is full of energy, offering crisp yellow-apple, lemon drop and orange zest flavors balanced by vibrant acidity. A perlage of small, refined and continuous bubbles provides the silky backdrop.		
Most similar wine	This offers aromas of jasmine, hawthorn and ripe pear. The round palate delivers creamy green apple, white peach, tangerine zest and a note of honeyed almond accompanied by racy acidity and a foaming mousse.	Here's a refreshing and savory blend of Catarratto, Pinot Bianco, Sauvignon and Traminer. It's loaded with succulent white peach, juicy pineapple and citrus zest. Crisp acidity gives this a clean, quenching finish.	Aromas of ripe pear and green apple follow over to the rich creamy palate along with notes of nectarine and glazed lemon drop. Bright acidity provides freshness while a soft mousse lends finesse.
2nd most similar wine	Honeysuckle and green apple aromas follow over to the foaming palate along with white peach and grapefruit. A candied lemon drop note caps off the finish while bright acidity lifts the rich flavors.	Refreshing, fun and refined, this crowd-pleasing sparkler offers ripe Bartlett pear, green apple and a hint of nectarine drop. Crisp acidity and a lively perlage give it a vibrant edge.	Crisp and refreshing, this lovely sparkler offers sensations of white wild flowers, green apple, citrus and Bartlett pear. Vibrant acidity balances the creamy, elegantly foaming palate and gives it a dry, invigorating finish.
3rd most similar wine	Crisp and refreshing, this lovely sparkler offers sensations of white wild flowers, green apple, citrus and Bartlett pear. Vibrant acidity balances the creamy, elegantly foaming palate and gives it a dry, invigorating finish.	Creamy and delicious, this elegant off-dry sparkler doles out layers of sweet white peach, yellow apple, pear and a tangy note of candied lemon drop. A silky perlage gives it a smooth polished texture while bright acidity lifts the rich flavors.	This bubbly Prosecco Superiore offers vibrant tones of white flower and cut grass followed by light touches of peach and honeydew melon. The mouthfeel is creamy and soft with a subtle touch of sweetness.

Note: Tasting notes of wines whose numerical representations are the closest (in cosine distance) to the numerical representation of the focal wine in each product embedding.

Table 2 shows that the product embeddings encode wine tastes in the vector space. In particular, the focal wine is a light, refreshing, fruit-forward white wine from Veneto in Italy. The wines with the most similar representation to this wine are all light, refreshing, fruit-forward white wines with similar taste characteristics (e.g., descriptors such as spring flowers, peach, citrus, apples, lemon drop, and orange zest) from Italy. Eight of the nine wines are from the same region (Veneto) and are made using the same white wine grape varietal (Glera) as the focal wine, whereas the last wine is from a nearby region in Italy

(Sardinia) and is made using a blend of white wine grape varietals (Catarratto, Pinot Bianco, Sauvignon, and Traminer). Note that these wines were selected solely on the basis of the similarity of product representations derived from the tasting notes and not on the basis of the wines' region, country, or varietal. Therefore, Table 2 shows that the embedding algorithms can infer and encode the underlying attributes of wines in the vector space from the wines' tasting notes.

Research Design

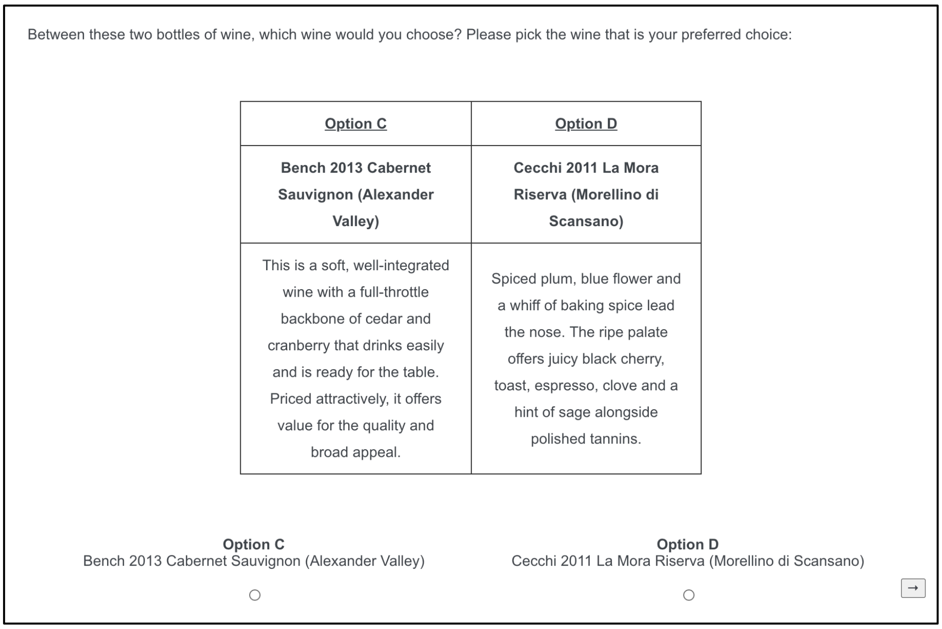
We conduct a preference-measurement study to demonstrate our proposed method. We partnered with Qualtrics, a reputed international service provider for market research, to collect data from 1,000 panelists (50.5% women; age = [25, 89]) from Australia (N = 250), New Zealand (N = 200), and the US (N = 550).[7] The panelists were at least 25 years old,[8] drank wine regularly (indicated drinking at least one glass of wine in the last 28 days),[9] and were employed. In addition, we tasked the service provider with ensuring that the panelists were demographically representative of the market of wine drinkers in the respective countries. Thus, our study setup is typical of online market research, and our substantive findings should apply to the overall market for wines in these three countries and should be of interest and relevance to a broad range of wineries, wine distributors, and wine retailers.

In the study, participants sequentially evaluate 32 pairs of randomly selected wines from our product-descriptions dataset (the complete set of 119,955 wines from 44 countries) and choose their preferred wine in each pair (see Figure 2 as an example of the choice task). We chose 32 different random pairs for each participant in order to tesselate the vector space, to ensure that our estimates generalize to all wines in our data. Each wine is described to

[7] We had originally requested Qualtrics to obtain 25% of the data from Australia, 25% from New Zealand, and 50% from the US. As we required participants to be regular wine drinkers, the pool of eligible participants in New Zealand was found to be limited. Therefore, Qualtrics requested, and we agreed, to rebalance these proportions to 25%, 20%, and 55% from Australia, New Zealand, and the US, respectively.
[8] The age requirement was added for ethical considerations.
[9] A majority of participants (86.5%) reported that they consume at least one glass of wine per week.

participants by its name and tasting notes in prose, akin to what consumers see when they evaluate wines in real life, online and in brick-and-mortar stores. Our method admits the inclusion of numerical product attributes (e.g., price) in the stimuli presented to participants and in the indirect utility specification (see equation 8). Our primary aim in conducting the study, however, was to investigate the properties of our proposed method. Therefore, to ensure that our empirical analysis relates only to the product attributes communicated in the prose product descriptions, we chose a more parsimonious research design whereby we presented participants with only the prose product descriptions.

Figure 2: Sample Screenshot from the Main Task



Between these two bottles of wine, which wine would you choose? Please pick the wine that is your preferred choice:

Option C	Option D
Bench 2013 Cabernet Sauvignon (Alexander Valley)	**Cecchi 2011 La Mora Riserva (Morellino di Scansano)**
This is a soft, well-integrated wine with a full-throttle backbone of cedar and cranberry that drinks easily and is ready for the table. Priced attractively, it offers value for the quality and broad appeal.	Spiced plum, blue flower and a whiff of baking spice lead the nose. The ripe palate offers juicy black cherry, toast, espresso, clove and a hint of sage alongside polished tannins.

Option C
Bench 2013 Cabernet Sauvignon (Alexander Valley)
○

Option D
Cecchi 2011 La Mora Riserva (Morellino di Scansano)
○

The preference-measurement data describe 32,000 wine choice tasks that were completed by 1,000 participants. The choice tasks correspond to 49,548 randomly selected wines, representing 41.3% of all wines in our data.[10] The evaluated wines encompass a diverse and comprehensive range, spanning 367 wine-growing regions, 40 wine-growing countries, and 566 wine-grape varietals, which is far more attribute-levels than is

[10] In 14,452 cases, a wine was selected at random more than once.

recommended by our data provider (Qualtrics) for a typical conjoint study with 1,000 participants.

Participants indicated that they are generally interested in wine and like wine, reporting an average of 5.59 and 6.22 on the respective 7-point scales (“not at all interested/very interested”; “don’t like it at all/like it very much”). As checks, participants’ task involvement was assessed on three Likert-scale questions (e.g., “I could relate to the overall situation of evaluating wines”; 1 = strongly disagree, 7 = strongly agree) (α = .74), and participants’ comprehension of wine descriptions was assessed on one item (1 = very difficult, 7 = very easy). Participants were fairly involved (*M* = 5.73) and found the wine descriptions to be fairly easy to understand (*M* = 5.43).

To measure individual consumer preferences, we model the observed choice r_{cw} from consumer *c* for wine *w* as depending on the latent variable r^*_{cw}:

(9) $$r^*_{cw} = \delta_c x_w + \varepsilon_{cw},$$

where δ_c is a consumer-specific vector of coefficients that captures consumer-specific heterogeneity, and x_w is a *K*-dimensional product embedding of wine *w*. We assume ε_{cw} is distributed i.i.d. Gumbel, and we observe that the participant chooses the wine on the right if $r^*_{cw_l} < r^*_{cw_r}$, where w_l and w_r are the wines on the left and the right, respectively, thereby yielding a logit model. We assume (uninformative) flat priors for all population coefficients and half student-t priors with 3 degrees of freedom and a scale parameter based on the standard deviation of the response for all group-level coefficients (Gelman 2006). We use the No-U-Turn Sampler to estimate the model.

To test and validate the model, we split the sample as follows. For each participant, we randomly select 28 choice tasks for the estimation sample and 4 choice tasks for the holdout sample. In the estimation sample, we use Leave-One-Out Cross-validation (LOOCV) to compare utility models and to conduct inference (Kim et al. 2020). LOOCV is a procedure whereby statistical models are estimated and tested on datasets formed by leaving out one observation from each dataset, estimating the model on this dataset, and then using the model to predict the pointwise posterior marginal density of the left-out observation. LOOCV has high statistical power in both testing (as all the data are used to test the model) and estimation (as almost all the data, except for a focal observation in each fold, are used to train the model). We use the Pareto smoothed importance sampling algorithm of Vehtari et al. (2017) to compute LOOCV probabilities and the LOOCV Information Criterion (LOOIC). Web Appendix D provides further details.

In addition, we use estimates from the estimation sample (composed of 28,000 observations) to predict pointwise purchase probabilities in the holdout sample (composed of 4,000 observations). We report several metrics that we derive from the probabilities in the holdout sample that are used to compare and characterize our utility model. These include the hit rate, the Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve, and sensitivity, specificity, and balanced accuracy from confusion matrices that characterize the predictive accuracy of the models.

Results

We estimate three utility models that incorporate product embeddings from the three aforementioned methods: M1, a model with the non-neural embedding; M2, a model with the feedforward embedding; and M3, a model with the recurrent embedding. In addition, we estimate three benchmark models. We estimate a benchmark model (BM1) that includes region, country, and grape varietal fixed effects. To allow for individual-level parameters in BM1, we aggregate attribute-levels that appear less than 10 times in our estimation sample, which is the recommended minimum number of observations in a regression to ensure robustness. In addition, we estimate two benchmarks from the Natural Language Processing literature: BM2 is a benchmark model with Latent Semantic Analysis (LSA) feature loadings

(Eliashberg et al. 2007); BM3 is a benchmark model with Latent Dirichlet Allocation (LDA) topic intensities (Büschken and Allenby 2016, Toubia et al. 2019). Web Appendix E describes how we construct BM2 and BM3.

Model Fit

Table 3 reports the LOOIC of each model. The LOOIC is the expected log posterior density of the model reported in the format of an information criterion metric and is a summary measure of overall in-sample fit; a lower LOOIC is preferred (Vehtari et al. 2017, Kim et al. 2020). Our analyses show that all three embedding-based models (M1, M2, and M3) fit the data better than all three benchmark models (BM1, BM2, and BM3). Among the embedding models, M3 (i.e., recurrent neural embedding) fits the data best.

Table 3: In-Sample Model Comparison

	BM1	BM2	BM3	M1	M2	M3
LOOIC	38706.3	38247.7	38008.1	37876.4	37601.9	37380.6

Note:
1. BM1 = fixed effects; BM2 = LSA feature loadings; BM3 = LDA topic intensities; M1 = non-neural embedding; M2 = feedforward neural embedding; and M3 = recurrent neural embedding.
2. LOOIC = Leave-One-Out Information Criterion. The model with the lowest LOOIC is preferred.

Table 4 reports the performance of the six models in the holdout sample. The first row in Table 4 reports models' AUC of the ROC curve in the holdout sample. The AUC corresponds to the following. Suppose we select an observation at random from the holdout sample where the participant chose a wine on the left (we term this a negative-class instance) and an observation at random where the participant chose the wine on the right (we term this a positive-class instance). The AUC is the probability that the model will correctly assign a higher probability to the positive-class instance of being positive than to the negative-class instance of being positive. Thus, a higher AUC is indicative of a model with higher predictive validity. Consistent with our earlier results, we find that M3 has the highest AUC among all six models, indicating that the model performs better at describing participants' choices. In addition, all product-embedding-based utility models (M1, M2, and M3) have a higher AUC than all benchmark models (BM1, BM2, and BM3).

Table 4: Out-of-Sample Model Comparison

	BM1	BM2	BM3	M1	M2	M3
AUC	0.538	0.564	0.586	0.601	0.607	0.619
Hit Rate	0.527	0.541	0.557	0.564	0.571	0.575
Sensitivity	0.609	0.530	0.548	0.555	0.558	0.557
Specificity	0.440	0.554	0.567	0.574	0.585	0.594
Balanced Accuracy	0.525	0.542	0.557	0.565	0.572	0.576

Notes:
1. BM1 = fixed effects; BM2 = LSA feature loadings; BM3 = LDA topic intensities; M1 = non-neural embedding; M2 = feedforward neural embedding; and M3 = recurrent neural embedding.
AUC = AUC of the ROC curve.
2. Hit Rate = Fraction of (out-of-sample) observations correctly predicted by the model.
3. Sensitivity = (True Positive) / (Number of Positive).
4. Specificity = (True Negative) / (Number of Negative).
5. Balanced Accuracy = (Sensitivity + Specificity) / 2.

The second row in Table 4 reports models' hit rate in the holdout sample. The hit rate is the percentage of observations in the holdout sample that are correctly classified by the model. A higher hit rate is desirable as it also indicates better predictive ability. The results are again consistent with our earlier findings, and M3 has the highest hit rate. Furthermore, all embedding models perform better than all benchmark models.

The remaining rows of Table 4 present metrics derived off the confusion matrices. Specifically, sensitivity is the percentage of positive-class observations in the holdout sample that are correctly classified by the model; specificity is the percentage of negative-class observations in the holdout sample that are correctly classified by the model. The balanced accuracy is the mean of the sensitivity and the specificity.

BM1 overpredicts the positive class. Therefore, it has a high sensitivity but low specificity, which indicates that it is unable to provide balanced and accurate classification. Consistent with our prior results, M3 has the highest sensitivity among all models except BM1, and the highest specificity among all models. Furthermore, M3 has the highest balanced accuracy—a measure that jointly accounts for both negative- and positive-class accuracy—across all models, again indicating that M3 performs better than any of the other models in measuring consumer preferences. Moreover, all three product-embedding-based utility models (M1, M2, and M3) have a higher balanced accuracy than the three benchmark models (BM1, BM2, and BM3).

In sum, all embedding-based models (M1, M2, and M3) outperform all three benchmark models (BM1, BM2, and BM3) on a wide variety of metrics derived from both cross-validation and testing in the holdout sample. Across the models, participants' preferences are best described by M3 (a utility model with the bGRU–AE derived product representations). This is in line with the notion that a more flexible generative model, such as bGRU–AE, encodes more information in its product representation. Therefore, despite its larger data requirement, the bGRU–AE product embedding leads to a utility model that more precisely captures participants' preferences for wines. Hence, we use the best-fitting model (M3) to characterize preferences and to conduct further analyses.

Partworths

An important benefit of our method is that it can be used to compute individual-specific partworths for all participants in a study and for attribute-levels in the data (i.e., for all 427 regions, 44 countries, and 708 grape varietals). This is far more than would be feasible using extant methods and participant choice data of equivalent length. Reporting 1,179 individual-specific partworths in a tabular format, however, would unduly lengthen our paper. Therefore, for brevity and as exemplars, in Table 5 we report the partworths of a consumer whose preference parameter vector is the average of the preference parameter vector of all participants in the study. We report partworths for the four most common wine regions, the four most common wine countries, and the four most common wine varietals in our data.

Table 5: Partworths of Regions, Countries, and Varietals

	Estimate	Standard Error
Regions:		
Bordeaux	-0.090 ***	0.013
California	-0.017	0.017
Tuscany	0.013	0.011
Washington	-0.173 ***	0.026
Countries:		
France	0.101 **	0.041
Italy	0.035	0.033
Spain	-0.170 ***	0.030
US	0.166 ***	0.041
Varietals:		
Cabernet Sauvignon	-0.119 ***	0.028
Chardonnay	0.006	0.037
Pinot Noir	0.031	0.027
Red Blend	-0.070 ***	0.022

Notes:
1. Estimate = partworth of the representative consumer.
2. $^{***} = p < 0.001$; $^{**} = p < 0.01$; $^{*} = p < 0.05$.
3. Standard errors computed using the delta method.

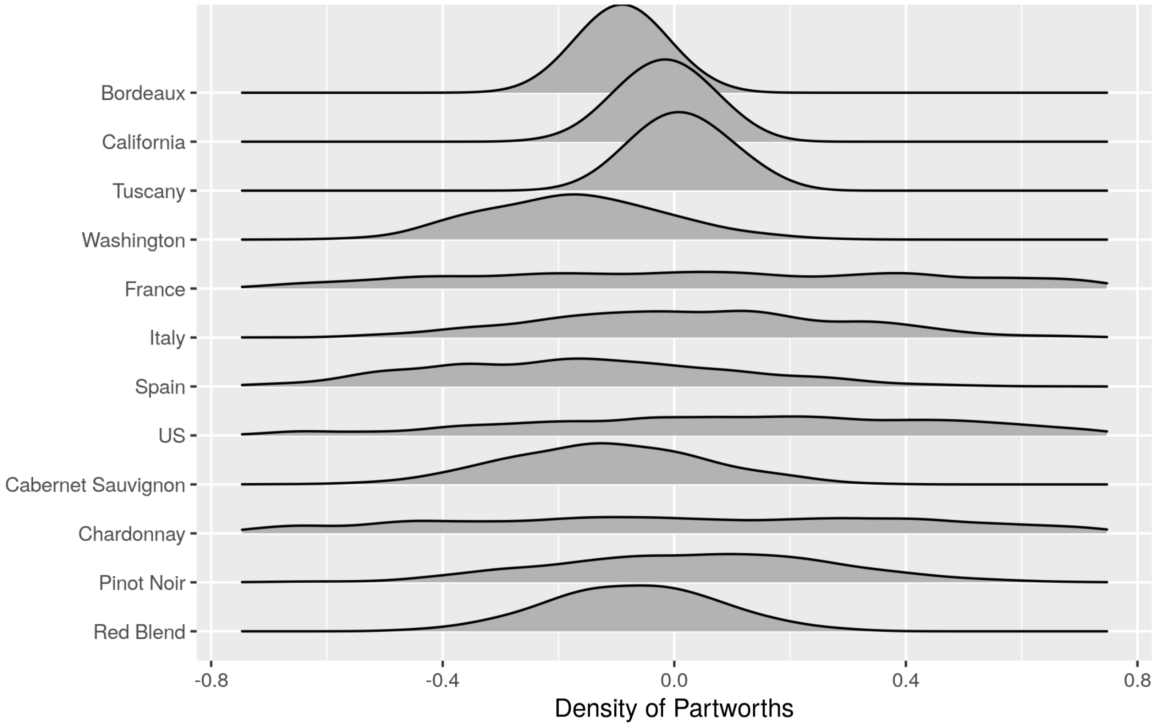
We find that the consumer prefers wines from France and the US over wines from Spain. The consumer also prefers wines from smaller wine regions such as Tuscany to wines from larger wine-growing regions such as Bordeaux and California.[11] Finally, the consumer prefers wines made from certain grape varietals (such as Pinot Noir) over other grape varietals (such as Merlot). Note that as the partworths are computed using estimated quantities, we use the delta method to compute standard errors.

We now visually summarize the preferences of all participants in our study for these attributes and attribute-levels. Specifically, Figure 3 presents the density of (individual-specific) estimates, for all 1000 participants in our study, of the 12 partworths in Table 5. Figure 3 shows that there is considerable individual-level heterogeneity in partworths.

[11] Traditionally, smaller wine-growing regions are regarded as better than larger wine-growing regions (Puckette 2020) because the former typically have stricter viticultural and winemaking regulations (Thompson 1987). Therefore, wines from smaller regions are more desirable as they are associated with higher quality and limited availability.

Moreover, the extent of individual-level heterogeneity itself varies across attributes and attribute-levels. In particular, the participants are more heterogeneous in partworths for countries and varietals than for regions. Furthermore, wines from some countries (e.g., Spain) are viewed very unfavorably by most participants, while there is considerable heterogeneity in how the participants value wines from other countries (e.g., US). Similarly, participant preferences for wines made from chardonnay show the most dissimilarity, whereas participants relatively consistently prefer wines made using pinot noir to wines made using cabernet sauvignon. Taken together, Figure 3 underscores the importance of measuring individual-level partworths and therefore highlights the managerial benefit of the cost advantages of our method.

Figure 3: Density of Participants' Partworths

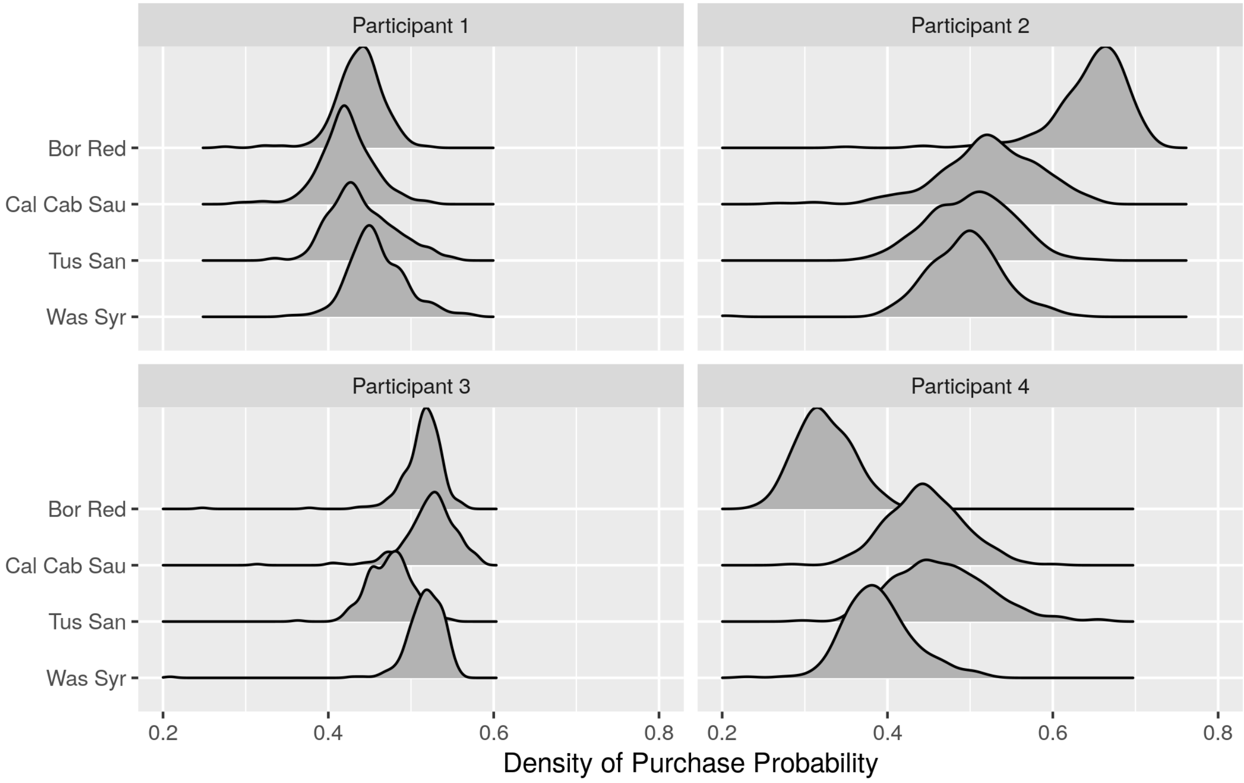


Note: Composite of the density plots of participant-specific partworths for each attribute-level. From top to bottom, the first four plots correspond to the four largest wine-growing regions (out of 427 regions), the next four plots correspond to the four largest wine-growing countries (out of 44 countries), and the last four plots correspond to the four most common grape varietals (out of 708 varietals) in our data. The value of the partworth is on the x-axis, and its density is on the y-axis.

Product-specific Valuations

Communicating consumer preferences as partworths of categorical attributes enables a parsimonious yet rigorous description of preferences that can be communicated succinctly to stakeholders. In addition, if products are traded on the basis of the categorical attributes, as is the case for wines, then the partworths are directly actionable for brands, manufacturers, retailers, and distributors. However, a downside to the use of categorical attributes is that in many categories, products that are observationally equivalent in categorical descriptors differ in other respects that are not captured by the categorical attributes. Specifically, wines that are observationally equivalent in categorical descriptors (i.e., wines from the same region, country, and made using the same grape varietal) often vary in taste and therefore in consumer preferences.

Figure 4: Density of Participants' Purchase Probabilities



Note: Composite of the density of participants' purchase probabilities of red wines from Bordeaux ("Bor Red"), cabernet sauvignon from California ("Cal Cab Sau"), sangiovese from Tuscany ("Tus San"), and syrah from Washington ("Was Syr").

To illustrate this phenomenon, we randomly select four participants from the US, and 1,000 wines of which 250 are from each of four iconic types: red blends[12] from Bordeaux, France; cabernet sauvignon from California, US; sangiovese from Tuscany, Italy; and syrah from Washington state, US. Figure 4 presents the probability that each participant will purchase (select) wine of each of the four types, if the wine was paired with a baseline alternative[13] in a choice task.

Figure 4 establishes that there is considerable dissimilarity in participants' purchase probabilities of red wines of the same type and red wines of different types. In particular, participant 1 dislikes red wines but differentially values various wines. Participant 2 strongly prefers red wines from Bordeaux. Participant 2's valuations, however, also vary greatly among different wines. In contrast, participant 3 is relatively indifferent among different wines of a type, while disliking sangiovese wines from Tuscany. Participant 4 strongly dislikes red wines from Bordeaux but likes some cabernet sauvignon wines from California and some sangiovese wines from Tuscany. Participant 4's valuations are much more heterogeneous than participant 3's valuations.

Importantly, while the addition of geographical attributes may improve the specificity of the categorical-attribute-based model and enable it to distinguish between wines from the same region, it also increases data requirements and therefore study costs. For example, there are 57 wine sub-regions in Bordeaux (France), 107 wine sub-regions in California (US), 41 wine sub-regions in Tuscany (Italy), and 16 wine sub-regions in Washington state (US).[14] Therefore, the addition of only the wine sub-regions corresponding to these four types of

[12] Red wines from Bordeaux are made from a blend of the following varietals: Cabernet Franc, Cabernet Sauvignon, Carmenère, Malbec, Merlot, and Petit Verdot.

[13] For expositional clarity, we set the utility of the baseline alternative to 0.

[14] Different countries use different nomenclature for geographical indications in wine. For consistency, we use the term "wine sub-regions" to refer to appellations in France, American viticultural areas in the US, and Denominazione di Origine Controllata wines in Italy. Each of these is a legally defined and protected sign guaranteeing the origins and manufacture of the wine (MacNeil 2015).

wines (corresponding to four wine regions of 708 wine regions in our dataset) would require adding 221 attribute-levels to the preference-measurement study, which would increase the data required for the study. Furthermore, the phenomenon that products that are observationally equivalent in categorical descriptors differ dramatically holds even after the inclusion of more categorical attributes, and this is a fundamental feature of products such as movies and wines (Chung and Rao 2012).

Decision Support System: Product Assortment

As we specify and estimate a utility model, our method can be used to forecast consumer purchases, and therefore used to support a variety of marketing decisions including pricing, product development, and product distribution. For brevity and as such analyses are standard, we focus on a use-case that builds on our earlier results and demonstrates and establishes the benefits of the enhanced specificity of our proposed embedding-based method and model.

In particular, we take the perspective of a brick-and-mortar wine retailer in the US that needs to decide on its product assortment. Globally, wine is predominantly sold in brick-and-mortar stores. For example, the US is at the vanguard of online wine sales. Yet, in 2019, online wine sales accounted for only 10.8% of total retail wine sales in the US (Briscoe 2020). Importantly, unlike online retailers (such as wine.com), brick-and-mortar retailers are constrained by shelf space and therefore can carry only a limited assortment. Furthermore, prior research has shown that a retailer's product assortment is a key driver of consumer store choice and purchase decisions (Briesch et al. 2009). The role of the product assortment is magnified in categories that are highly differentiated, such as wines (Lynch and Ariely 2000). Therefore, for brick-and-mortar retailers, choosing the right wine assortment is vital to profitability.

Importantly, the categorical-attribute-based model provides limited guidance in this managerial decision, as the categorical model assumes that a consumer would be indifferent between different wines of the same type. In particular, of the 550 participants from the US, the categorical-attribute-based model predicts that 223 participants (40.5%) would most prefer red wines from Bordeaux, 194 participants (35.3%) would most prefer Cabernet Sauvignon from California, 102 participants (18.5%) would most prefer Sangiovese from Tuscany, and 31 participants (5.6%) would most prefer Syrah from Washington, while being completely indifferent among different wines of the same type. Thus, the categorical-attribute-based model would suggest carrying any assortment of the four types of wines.

To determine which wines the retailer should carry, we conduct the following analysis. To simplify the exposition, we limit our attention to the 1,000 wines of the four types described earlier and in Figure 4. For each wine, we determine its valuation by each US participant. We choose wines that were valued the most by each participant as wines that are likely to best perform at retail.

The model recommends 59 wines, comprising 18 reds from Bordeaux, 23 Cabernet Sauvignon from California, 12 Sangiovese from Tuscany, and 6 Syrah from Washington, from the set of 1,000 wines. The model predicts that US participants would purchase a wine from this selection in 64% of choice tasks, where the wine was paired against a baseline alternative. In comparison, our model predicts that US participants would purchase a randomly chosen wine (from the set of 1,000 wines) in 48% of similar choice tasks. This improvement in performance is a direct result of the enhanced specificity of our method and model.

Dimensionality of the Vector Space

Increasing the dimensionality of the vector space has two opposing effects. On the one hand, a larger vector space is more expressive and allows for a more granular description of the consumer experience and a more detailed capture of consumer preferences. Thus, using a larger vector space in consumer-preference measurement can improve the performance of the measurement method. On the other hand, a larger vector space also increases the data requirements, both for training a product embedding and for estimating the utility model on participant choices. Thus, increasing the dimensionality of the vector space may backfire and impair the performance of the measurement method.

To examine this issue, we construct 5-dimensional and 15-dimensional product embeddings of M3, which is the best-performing model in 10 dimensions. Our analyses show that the 5-dimensional model and the 15-dimensional model fit worse than the 10-dimensional model. Specifically, for the 5-dimensional model relative to the 10-dimensional model, ΔLOOIC = 37609.3 – 37380.6 = 228.87; for the 15-dimensional model relative to the 10-dimensional model, ΔLOOIC = 37559.2 – 37380.6 = 178.6. Therefore, in our study, we find that a 10-dimensional vector space represents a good compromise, as it provides an accurate description of product differentiation while being more parsimonious than the 15-dimensional model.

Transfer Learning

We train a word embedding that is specific to our context of wines. An alternative is to employ transfer learning—using a word embedding pretrained on a large and broad cross-context dataset (e.g., articles from Wikipedia). Transfer learning is particularly crucial because by pairing it with product-embedding algorithms (such as the non-neural embedding model we describe in this paper), our method can be used to conduct a preference-measurement study even in contexts where product description data are unavailable.

The consequences of using pretrained word embeddings are a priori unclear. On the one hand, a word embedding that was trained on a cross-context corpus is likely to be less accurate in a specific context than a word embedding trained on data from that specific context, given that word usage is often context-specific. For example, “red” in the context of wines implies a flavor profile[15] (as it relates to wine production) whereas in other contexts it typically refers to a color. On the other hand, a larger dataset may provide more information to the word-embedding model, which may allow the model to discover structure that is missed in an embedding trained on a smaller context-specific dataset. In this regard, the pretrained embedding may lead to more accurate product embeddings.

To examine this issue, we download the Word2Vec embedding (trained on a Google dataset with one billion words) from Mikolov et al.’s (2013) repository. Table 6, analogous to Table 1, lists the words that have the most similar Word2Vec representation to the representations of blackberry and blueberry (Columns 1 and 2) and to both fruits jointly (Column 3). Table 6 shows that the pretrained embedding is noisier and less accurate than the wine-specific embedding. For example, blackberry—as a taste or fruit—is spelled the same as a brand of mobile phones. Consequently, many words listed in Table 6 relate to the mobile phone brand rather than to the taste (e.g., Curve_3g, which is a BlackBerry smartphone). Furthermore, the downloaded representation is noisy, as it was trained on crawled data. Thus, the fourth most similar word to blackberry is banana_gelato, while the seventh most similar word to blackberry is the URL of a mobile phone shop.

[15] Red wine and white wine differ in fermentation processes, and therefore in flavor profile. In red wine, the grape juice is fermented with the grape skins. Heat and alcohol generated in the fermentation process extract flavonoids and other phenolic compounds from the grape skins (e.g., adding tannin), thereby affecting its sensory properties. This process also tints the wine. In contrast, in white wines, the grape juice is fermented without the grape skins. Hence, the wine is not tinted (see Bakker and Clarke 2011, MacNeil 2015).

Table 6: Similar Taste Descriptors in the Word2Vec Embedding

Focal descriptor		*blackberry*	*blueberry*	*blackberry* and *blueberry*
Order of similarity of taste descriptor	*1st*	blackberry	blueberry	blueberry
	2nd	blackberries	blueberries	blackberry
	3rd	BlackBerry	strawberry	berry
	4th	banana_gelato	berry	blackberries
	5th	berry	cranberry	strawberry
	6th	ripe_blackberries	berries	blueberries
	7th	deals_http://www.directphoneshop.co.uk/dealset.asp?id=####	Blueberry	raspberries
	8th	Curve_3g	strawberries	berries
	9th	blueberry_blackberry	raspberries	Blueberry
	10th	raspberry	cherries	raspberry

Note: List of the 10 most similar taste descriptors by the cosine similarity of pretrained Word2Vec representations to the focal descriptors in the column headings.

Next, we examine the effect of using a pretrained embedding to construct product embeddings in wines. To ensure the comparison is fair, we use the first 10 left-singular vectors to construct a recurrent product embedding, which is the best-performing product embedding using a context-specific word embedding. We re-estimate M3 using the product embedding formed from the pretrained word embedding. We find that M3 with the pretrained word embedding has a worse fit compared to its counterpart with the context-specific word embedding (ΔLOOIC = 38480.3 – 37380.6 = 1099.7). Importantly, however, M3 using the pretrained word embedding has a better model fit than the categorical-variable-based model that is typical and traditional in extant quantitative methods for preference measurement, such as conjoint analysis (ΔLOOIC = 38706.3 – 38480.3 = 226.0). Therefore, in sum, while a context-specific word embedding leads to a better fitting utility model, there is still value in using transfer learning to reduce the costs of constructing a product embedding.

CONCLUSION

Over the last five decades, considerable progress has been made in the measurement of consumer preferences, including both stated preferences (Hauser et al. 2006, Netzer et al. 2008) and revealed preferences (Blundell et al. 1993, Erdem et al. 2005, Chintagunta et al. 2006). Despite these advances, the application of quantitative methods to preference measurement has been constrained by the extent to which non-numerical product attributes can be described by categorical variable(s).

Our paper addresses this limitation. We propose a novel embedding-based method with a simple and realistic study design that is straightforward to administer. We demonstrate the value of our proposed method in a study of wines, an experiential product category characterized by rich and complex qualitative attributes. We establish that our embedding-based method is better at predicting (in-sample and out-of-sample) consumer choices than are extant approaches. We detail individual-specific partworths for wine attributes, and also use our estimates to conduct individual-level analyses to support substantively critical managerial decisions. In sum, our results show that the use of our proposed embedding-based utility model and marketing research method enables the measurement of detailed preferences of consumers for products with complex product attributes that have many attribute-levels and are hence best described in prose.

Our method is more cost-effective in data collection than extant methods. For example, we study consumer preferences in three countries (Australia, New Zealand, and the US) for wines from 427 wine-growing regions in 44 wine-growing countries made from 708 wine-grape varietals. If we conduct a conjoint study with these three attributes across the three countries, Qualtrics and Sawtooth software (both leading global conjoint analytics companies) recommend a minimum sample size[16] of about 25,000 participants. In contrast, our method yields robust and reliable estimates on data from 1,000 participants, which is 4% of the sample size required for a traditional conjoint study.

Our methodological improvements are particularly relevant and important in contexts where the product is hard to describe adequately as a list of categorical variables. In addition, our method is likely to benefit the measurement of preferences for infrequently occurring attributes that are difficult to quantify and estimate precisely in traditional conjoint analysis (Chung and Rao 2012). Importantly, the frequency with which an attribute occurs in the data is a poor correlate of the economic and managerial significance of the attribute. For example, as with any limited edition or collectable item, descriptors of rare wines are of great economic significance even though such wines are observed relatively infrequently. In our method, information on partworths accrues through the entire vector space, which enhances the efficiency of the research design.

Our approach is extensible in several ways. While we elaborate the application of our proposed approach to primary data, it can also be applied to secondary data without major modification. Importantly, such data are now available in many categories including hospitality, travel, and entertainment. Moreover, when investigating consumer preferences, we ask participants to evaluate randomly selected wines. In the spirit of Toubia et al. (2004), we could instead use the vector space to develop a more efficient experimental design. In addition, it would be interesting to investigate behavioral phenomena such as the attraction effect in our model (Lee and Feinberg 2021). With the field of machine learning in the midst of a renaissance, we hope our research spurs interest in the use of embedding methods for preference measurement.

[16] https://www.qualtrics.com/support/conjoint-project/getting-started-conjoints/getting-started-choice-based/conjoint-analysis-white-paper/ [accessed April 23, 2021].

REFERENCES

Arias-Bolzmann L, Sak O, Musalem A, Lodish L, Báez R, De Sousa LJ (2003) Wine pricing: The influence of country of origin, variety, and wine magazine ratings. *International Journal of Wine Marketing*, *15*(2):47–57.

Arora S, Liang Y, Ma T (2017) A simple but tough-to-beat baseline for sentence embeddings. In *Proceedings of the International Conference on Learning Representations (ICLR) 2017*.

Ashenfelter O (2010) Predicting the quality and prices of Bordeaux wines. *Journal of Wine Economics*, *5*(1):40–52.

Athey S, Blei D, Donnelly R, Ruiz F, Schmidt T (2018) Estimating heterogeneous consumer preferences for restaurants and travel time using mobile location data. *AEA Papers and Proceedings*, *108*:64–67.

Bakker J, Clarke RJ (2011) *Wine Flavour Chemistry*. (Wiley, Oxford, UK).

Blei DM (2012) Probabilistic topic models. *Communications of the ACM*, *55*(4):77–84.

Blundell R, Pashardes P, Weber G (1993) What do we learn about consumer demand patterns from micro data? *The American Economic Review, 83*:570–597.

Briesch RA, Chintagunta PK, Fox EJ (2009) How does assortment affect grocery store choice? *Journal of Marketing Research*, *46*(2):176–189.

Briscoe K (2020) Digital wine sales are booming, and some wonder if they'll last. *Wine Magazine*, at: https://www.winemag.com/2020/05/07/online-wine-sales-last/ [accessed April 23, 2021].

Büschken J, Allenby GM (2016) Sentence-based text analysis for customer reviews. *Marketing Science*, *35*(6):953–975.

Chintagunta P, Erdem T, Rossi PE, Wedel M (2006) Structural modeling in marketing: Review and assessment. *Marketing Science*, *25*(6):604–616.

Chung J, Rao VR (2012) A general consumer preference model for experience products: Application to internet recommendation services. *Journal of Marketing Research, 49*(3):289–305.

Cooper-Martin E (1991) Consumers and movies: Some findings on experiential products. *ACR North American Advances*.

Eliashberg J, Hui SK, Zhang ZJ (2007) From story line to box office: A new approach for green-lighting movie scripts. *Management Science*, *53*(6):881–893.

Erdem T, Srinivasan K, Amaldoss W, Bajari P, Che H, Ho T, Hutchinson W, Katz M, Keane M, Meyer R, Reiss P (2005) Theory-driven choice models. *Marketing Letters*, *16*(3/4):225–237.

Frederick S, Lee L, Baskin E (2014) The limits of attraction. *Journal of Marketing Research, 51*(4):487–507.

Gelfand AE, Dey DK, Chang H (1992) Model determination using predictive distributions with implementation via sampling-based methods. Stanford University, CA, USA.

Gelman A (2006) Prior distributions for variance parameters in hierarchical models (comment on article by Browne and Draper). *Bayesian Analysis*, *1*(3):515–534.

Gilovich T, Gallo I (2020) Consumers' pursuit of material and experiential purchases: A review. *Consumer Psychology Review*, *3*(1):20–33.

Green PE, Krieger AM, Wind Y (2001) Thirty years of conjoint analysis: Reflections and prospects. *Interfaces*, *31*(3_supplement), S56–S73.

Hadj Ali H, Nauges C (2007) The pricing of experience goods: The example of en primeur wine. *American Journal of Agricultural Economics*, *89*(1):91–103.

Hauser J, Eggers F, Selove M (2019) The strategic implications of scale in choice-based conjoint analysis. *Marketing Science*, *38*(6): 1059-1081.

Hauser J, Tellis GJ, Griffin A (2006) Research on innovation: A review and agenda for marketing. *Marketing Science, 25*(6):687–717.

Holbrook MB, Hirschman EC (1982) The experiential aspects of consumption: Consumer fantasies, feelings, and fun. *Journal of Consumer Research*, *9*(2):132–140.

Hornik K, Stinchcombe M, White H (1989) Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5):359–366.

Huber O (1980) The influence of some task variables on cognitive operations in an information-processing decision model. *Acta Psychologica, 45*(1/3):187–196.

Jaeger SR, Danaher PJ, Brodie RJ (2009) Wine purchase decisions and consumption behaviours: Insights from a probability sample drawn in Auckland, New Zealand. *Food Quality and Preference*, 20(4): 312-319.

Kim S, Lee C, Gupta S (2020) Bayesian synthetic control methods. *Journal of Marketing Research*, *57*(5):831–852.

Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kleinmuntz DN, Schkade DA (1993) Information displays and decision processes. *Psychological Science*, *4*(4):221–227.

Le QV, Mikolov T (2014) Distributed representations of sentences and documents. *International Conference on Machine Learning*, 1188–1196.

Lee KY, Feinberg FM (2021) Modeling and measuring scale attraction effects: An application to charitable donations. *Journal of Marketing Research,* forthcoming.

Lynch Jr JG, Ariely D (2000) Wine online: Search costs affect competition on price, quality, and distribution. *Marketing Science*, *19*(1):83–103.

MacNeil K (2015) *The Wine Bible* (Workman Publishing, New York).

Malhotra NK (1984) Reflections on the information overload paradigm in consumer decision making. *Journal of Consumer Research*, *10*(4):436–440.

McFadden D (1981) Econometric models of probabilistic choice. In Manski C and McFadden D, eds. *Structural Analysis of Discrete Data with Econometric Applications* (MIT Press, MA, USA).

Mikolov T, Sutskever I, Chen K, Corrado G, Dean J (2013) Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems, 26*:3111–3119.

Morales A, Amir O, Lee L (2017) Keeping it real in experimental research: Understanding when, where, and how to enhance realism and measure consumer behaviour. *Journal of Consumer Research, 44*(2):465–476.

Mukherjee A, Chang HH, Chattopadhyay A (2019) Crowdfunding: Sharing the entrepreneurial journey. In *Handbook of the Sharing Economy* (Edward Elgar Publishing, UK).

Netzer O, Toubia O, Bradlow ET, Dahan E, Evgeniou T, Feinberg FM, Feit EM, Hui SK, Johnson J, Liechty JC, Orlin JB (2008) Beyond conjoint analysis: Advances in preference measurement. *Marketing Letters, 19*(3/4):337–354.

Palangi H, Deng L, Shen Y, Gao J, He X, Chen J, Song X, Ward R (2016) Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *24*(4):694–707.

Park YH, Ding M, Rao VR (2008) Eliciting preference for complex products: A web-based upgrading method. *Journal of Marketing Research*, *45*(5):562–574.

Payne JW, Bettman JR, Johnson EJ (1993) *The Adaptive Decision Maker* (Cambridge University Press, Cambridge, UK).

Pennington J, Socher R, Manning C (2014) GloVe: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

Puckette M (2020) *Looking for good wine? Start with the appellation.* At: https://winefolly.com/deep-dive/looking-for-good-wine-start-with-the-appellation/ [accessed April 23, 2021].

Schäfer AM, Zimmermann HG (2006) Recurrent neural networks are universal approximators. *International Conference on Artificial Neural Networks*, 632–640.

Scholz SW, Meissner M, Decker R (2010) Measuring consumer preferences for complex products: A compositional approach based on paired comparisons. *Journal of Marketing Research*, *47*(4):685–698.

Statista (2020) *Wine Worldwide*. Available at: https://www.statista.com/outlook/10030000/100/wine/worldwide [accessed April 23, 2021].

Stewart DW (1981) The application and misapplication of factor analysis in marketing research. *Journal of Marketing Research*, *18*(1):51–62.

Stone DN, Schkade DA (1991) Numeric and linguistic information representation in multiattribute choice. *Organizational Behavior and Human Decision Processes*, *49*(1):42–59.

Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, 3104–3112.

Thompson CC (1987) Alcoholic beverages and vinegars. *Quality Control in the Food Industry*, *4:*57–64.

Toubia O, Hauser J, Simester DI (2004) Polyhedral methods for adaptive choice-based conjoint analysis. *Journal of Marketing Research*, *41*(1):116–131.

Toubia O, Evgeniou T, Hauser J (2007a) Optimization-based and machine-learning methods for conjoint analysis: Estimation and question design. *Conjoint Measurement: Methods and Applications*, *12*:231–258.

Toubia O, Hauser J, Garcia R (2007b) Probabilistic polyhedral methods for adaptive choice-based conjoint analysis: Theory and application. *Marketing Science*, *26*(5):596–610.

Toubia O, Iyengar G, Bunnell R, Lemaire A (2019) Extracting features of entertainment products: A guided Latent Dirichlet Allocation approach informed by the psychology of media consumption. *Journal of Marketing Research*, *56*(1):18–36.

Vehtari A, Gelman A, Gabry J (2017) Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and Computing, 27*(5):1413–1432.

Wedel M, Kamakura WA (2012) *Market Segmentation: Conceptual and Methodological Foundations*, Vol. 8 (Springer Science & Business Media, New York City, USA).

Wilson BJ, Schakel AM (2015) Controlled experiments for word embeddings. *arXiv preprint arXiv:1510.02675*.

WEB APPENDIX A: NEURAL PRODUCT EMBEDDINGS

In this section, we describe the architecture of our neural embedding models. We begin by describing our notation and defining the model primitives.

Neural networks are mathematical models that are constructed by combining simpler mathematical models, termed neurons, in a directed, weighted graph. The neurons are arranged in layers, which are sets of neurons that are connected to the same input vector, share intermediate variables, and jointly produce an output vector. We use five types of neurons in the neural models:

1. Input neurons: Input neurons connect the data to the neural network. They take the data as input and pass it unchanged to the next layer of the neural network.
2. Embedding neurons: A layer of embedding neurons maps a (bounded) integer variable identifying an object to a vector representation of the object. Specifically, the embedding layer consists of a parameter matrix and an operator whose output, when the integer variable is i, is the ith row of the parameter matrix. Thus, this matrix has dimensions $N \times D$, where N is the cardinality of the set of unique objects (e.g., words in the vocabulary; documents in the collection) and D is the dimensionality of the embedding.
3. Perceptrons: Sigmoid perceptrons implement a system of non-linear regressions—the output vector of the layer of sigmoid perceptrons is the composition of a sigmoid function and an affine function applied to the inputs. A layer of sigmoid perceptrons is described by the equation:

(A1)
$$y = tr(x) = \sigma\left(B_y + W_y x\right),$$

where y is the output vector, x is the input vector, B_y and W_y are a vector and a matrix of parameters, respectively, and σ is the logistic function. tr is the transfer function of the layer.

4. Gated Recurrent Unit (GRU) neurons: Recurrent neurons are used to express Markov dependencies along a sequence. A layer of GRU neurons is described by four equations:

(A2) $$z_t = \sigma(B_z + W_z x_t + U_z h_{t-1}),$$

(A3) $$r_t = \sigma(B_r + W_r x_t + U_r h_{t-1}),$$

(A4) $$\widehat{h_t} = \varphi\big(B_h + W_h x_t + U_h(r_t \circ h_{t-1})\big),$$

(A5) $$h_t = (1 - z_t) \circ h_{t-1} + z_t \circ \widehat{h_t},$$

where *t* is the longitudinal element of the data, x_t is the input vector, z_t is the output vector, h_t is the state vector, and $\{r_t, \widehat{h_t}\}$ are intermediate variables. $\{B_z, W_z, U_z\}$, $\{B_r, W_r, U_r\}$, and $\{B_h, W_h, U_h\}$ are parameter vectors and matrices. φ is the hyperbolic tangent. $\circ$ is the Hadamard (element-by-element) product.

Equations (A2 – A4) play the following role in the model. Equation (A2) implements a regression model such that its output is a non-linear function of an affine transformation of both the inputs and the current state vector. Equation (A3) governs the extent to which the Markov process that the model represents retains prior information. Equation (A4) proposes a new state vector as an affine transformation of the inputs and the current state vector. Finally, equation (A5) is a composite of the outputs, the current state vector, and the proposed state vector, such that the model implements a first-order Markov process.

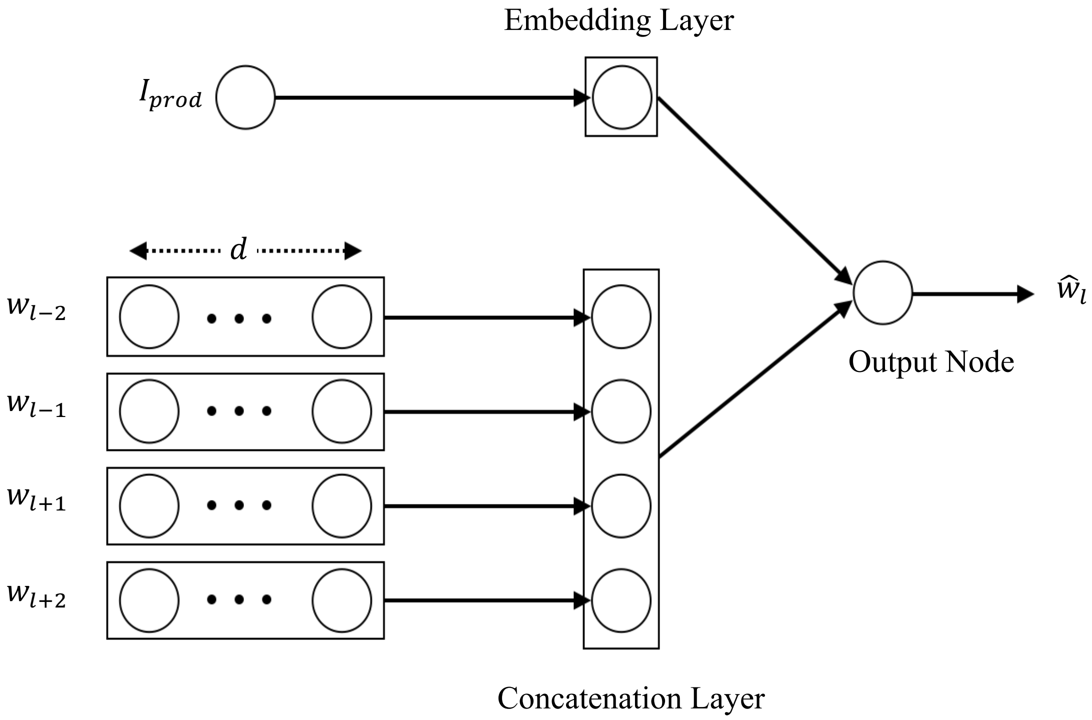
5. Output sigmoid neurons: Output neurons implement a logistic regression of the outputs of the prior layer of the network.

The neural embedding models contain a sequential composition of neural layers with an input layer that receives the independent variables (termed the model's "features"), hidden layer(s) that compute model intermediates, and an output layer that yields the model's estimate of the dependent variables. We next describe the architecture of the neural embedding models in separate sub-appendices.

Web Appendix A1: Feedforward Embedding Model

The feedforward embedding is constructed by developing a model to predict the focal word (w_l) given the vector representation of the product and the concatenated representation of the two left-neighboring and two right-neighboring words. The model features two branches that are connected to a single output node (Figure A1 illustrates the architecture of the feedforward embedding model). The two branches serve the following functions.

Figure A1: Feedforward Embedding Model



Notes: Neurons are depicted by a circle. Layers of neurons are delineated by rectangular boxes. Solid arrows depict the movement of data.

The first branch uses an embedding layer to construct the d-dimensional representation of the product. The parameters of the embedding layer are a $\text{Num}_{prod} \times D$ matrix, where Num_{prod} is the number of products and D is the dimensionality of the embedding layer. Let I_{prod} denote the position of the product in any random ordering of the products. Then the output of the embedding layer is the I_{prod} row of this matrix.

The second branch concatenates the d-dimensional representations (from a word embedding) of the two words to the left of each focal word and the two words to the right of the focal word. The two branches are connected to a neuron with softmax activation over the vocabulary. Thus, the output layer takes as input the product embedding of the product description and the word embeddings of the neighboring words and uses these to predict the focal word. The output of the output node ($\widehat{w}_l$) is a probability distribution over the vocabulary. The model is trained to predict all words in each product description, excluding the first two and the last two words of the description, which do not have a sufficient number of neighboring words.

Importantly, any continuous function $f(x)$ between measurable spaces can be arbitrarily closely approximated as the weighted sum of the outputs of a (finite) layer of sigmoid perceptrons (Hornik et al. 1989). Therefore, the feedforward embedding model described has the capacity to arbitrarily closely approximate any distribution function resulting from any generative process $\mu: \{X, Y\} \longrightarrow \{0,1\}$, where X is the subspace of the vector representations of the product and the words in the quintagram context window, and Y is the subspace of the outputs, if and only if the product embedding accurately summarizes all relevant information about the product in its vector representation.[17] It follows that the vector representation of a product, the collection of which is the embedding, thus serves to incorporate the information in the prose descriptions in the utility model.

Web Appendix A2: Recurrent Embedding Model

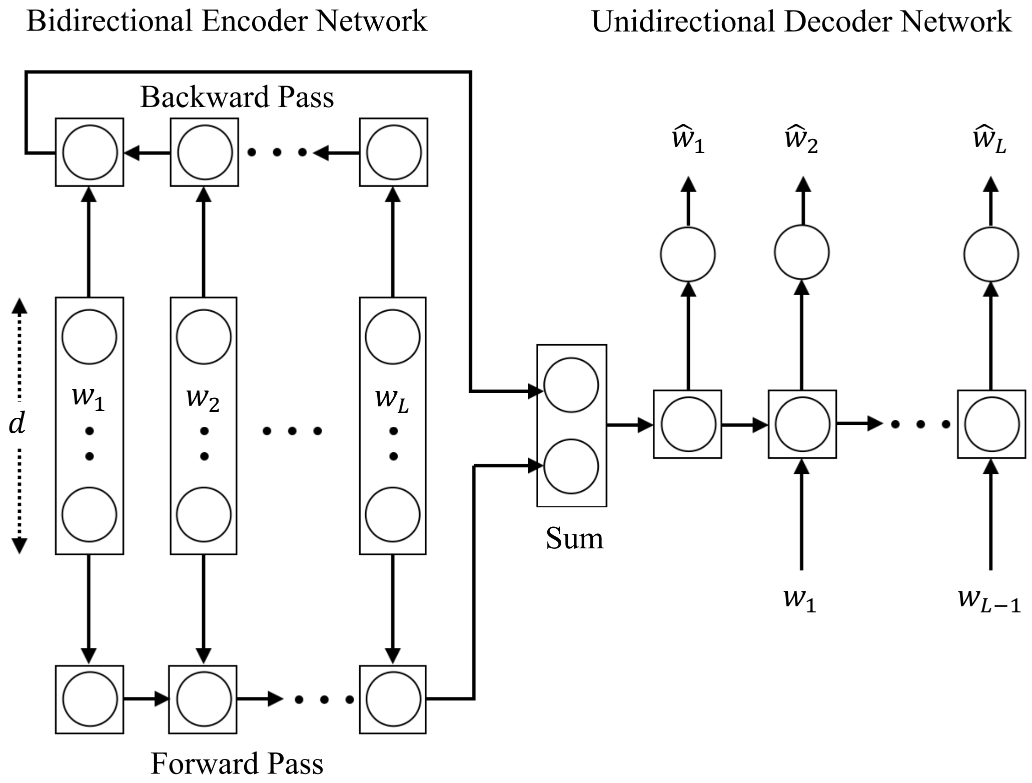
The recurrent embedding model is composed of two sub-models: (1) a bidirectional encoder model and (2) a unidirectional decoder model. Figure A2 depicts the model visually.

As can be seen in Figure A2, the bidirectional encoder model takes as input the *d*-dimensional vector representation of the sequence of words $\{w_1, \dots w_L\}$ in each product description and produces a state vector that summarizes the input description. The encoder

[17] We tested alternative model architectures where we included additional hidden layers to enhance the capacity of the model to approximate more complex functions. We found that a shallow architecture (i.e., one with fewer hidden layers) suffices in our application.

model has two layers. The first layer, termed the forward pass, takes as input the state vector up to the prior word ($s_{f(l-1)}$) and the focal word (w_l) to construct the state vector up to the current word (s_{fl}). Thus, this layer proceeds "forward" (i.e., from left to right, in the direction that the prose would be read). The second layer, termed the backward pass, proceeds "backward" (i.e., from right to left, in the reverse direction) and constructs the state vector for the focal word (s_{bl}) from the state vector beyond the current word ($s_{b(l+1)}$) and the focal word (w_l). The state vectors from the forward pass (s_{fL}) and from the backward pass (s_{b1}) are summed to create the final state vector of the encoder model.

Figure A2: Recurrent Embedding Model



Notes: Neurons are depicted by a circle. Layers of neurons are delineated by rectangular boxes. Solid arrows depict the movement of data.

The decoder model uses the sequence of words offset by one word and the state vector from the encoder model to predict the sequence of words. Unlike the bidirectional encoder model, the decoder model is unidirectional to prevent the decoder from cheating by encoding the next word(s) in a backward pass. Finally, the output from the decoder is passed to a dense layer with softmax activation, whose output is a probability distribution over the vocabulary.

As a recurrent neuron is a feedforward perceptron augmented by a state vector, similar to feedforward networks that are universal approximators in cross-sectional data, recurrent neural networks are universal approximators in sequential data. Specifically, a recurrent network has the capacity to arbitrarily closely approximate any state space model between measurable spaces (Schäfer and Zimmermann 2006). This class of models is very broad and includes all typical Markov models in marketing. In the state space model, information on “prior” observations (in the natural direction of the sequential data, which is left to right in English) is completely summarized in the state vector. As the bidirectional encoder model proceeds in both directions (the forward pass encodes from left to right, and the backward pass encodes from right to left), the final state vector encodes the entire verbal description. Furthermore, as the embedding model is trained in an autoencoder configuration and the state vector is used by the decoder model to recreate the input sequence, the model learns to encode all relevant information about the product in the state vector. It follows that the state vector can then be used as the vector representation of the product, the collection of which is the recurrent embedding, in the utility model.

WEB APPENDIX B: ORTHONORMAL BASIS

We seek to construct a product embedding in which the normed vector space has an orthonormal basis. We proceed as follows. Given a product embedding, we stack the *d*-dimensional representations of *N* products vertically to form *X*, a *N* × *d* matrix. β_i is a *1* × *d* coefficient vector describing the preferences of a consumer over the *d*-dimensional vector space. Then:

(B1) $$\overline{\text{Utility}_\imath}\,' = X\beta_i{}',$$

where $\overline{\text{Utility}_\imath} = \{\overline{u_{\imath 1}}, \dots, \overline{u_{\imath N}}\}$ is a *1* × *N* row vector that describes the deterministic component of consumer *i*'s utility—for example, $\overline{u_{\imath 1}} = x_1\beta_i{}'$, where x_1 is the first row of X. Consider the (compact) Singular Value Decomposition of *X:*

(B2) $$X = U\Sigma V'.$$

The number of products is greater than the number of dimensions of the vector space (*N* > *d*). Hence, *U* is a *N* × *d* orthonormal matrix, Σ is a *d* × *d* diagonal matrix, and *V* is a *d* × *d* orthonormal matrix. If we use *U* as the product embedding instead of *X* in the utility model, then we estimate:

(B3) $$\overline{\text{Utility}_\imath} = X\beta_i{}' = U\Sigma V'\beta_i{}' = U\left(\Sigma V'\beta_i{}'\right) = U(\beta V\Sigma)' = U\beta_{iSDV}{}',$$

where $\beta_{iSDV} = \beta_i \mathrm{V}\Sigma$.

U (the *d* left-singular vectors of *X*) form an orthonormal basis over the column space of *X*. Therefore, the transformation is lossless. Furthermore, *V* is a rotation matrix and Σ is a scaling matrix. Hence, estimating a utility model formed using the transformed embedding has the effect of rotating the vector space coefficients by *V* and scaling them by Σ. The transformation, however, leaves the partworths unchanged:

(B4) $$x^{kl}\beta_i{}' = u^{kl}\left(\Sigma V'\beta_i{}'\right) = u^{kl}\beta_{iSDV}{}',$$

where u^{kl} is the location of the *l*th level of the *k*th attribute in the transformed vector space.

WEB APPENDIX C: NEURAL PRODUCT EMBEDDING MODEL IMPLEMENTATION

We use the Keras Advanced Programming Interface (API) to implement the neural product embedding models in TensorFlow. The Keras API includes functions that allow us to define and connect the neural layers that compose the mathematical model. These model definitions are then compiled using the Keras compiler to form a computational graph in TensorFlow—a directed graph in which nodes describe mathematical operations, and directed edges describe the flow of data in the form of tensors (which are high-dimensional mathematical arrays) between the nodes. The computational graph is then used to train the network parameters towards an objective.

TensorFlow implements reverse accumulation automatic differentiation, which facilitates optimization. Neural networks are compositions of simple mathematical functions. Therefore, the gradient of a model with respect to any parameter or variable is composable by applying the chain rule to the gradients of its constituent functions (the nodes of the computational graph). TensorFlow implements reverse accumulation automatic differentiation as follows. In the forward pass, the TensorFlow kernel derives the value at each node of the computational graph for each observation. In the reverse pass, the TensorFlow kernel derives the gradient at each node of the computational graph for each observation. The composition of these values enables optimization.

In both neural product embedding models, the output is a probability distribution over the vocabulary. Therefore, following common practice, we train the network to minimize the categorical cross-entropy loss function. This loss function is equivalent to maximizing the likelihood of a multinomial logit model applied to the output of the layer preceding the output layer of the network.

We use the Adam optimizer to train the models. Our empirical investigations show that both neural embedding models are robust to the choice of stochastic optimizers (e.g., AdaGrad and RMSprop). Stochastic optimizers divide the data into randomly selected batches (of a fixed batch size), which are sequentially processed until the model has been trained on the entire set of data. Each pass over the data is termed an epoch. We train each model for up to 500 epochs. In each epoch, we randomly select 10% of the data as validation data. If the training is not able to decrease the loss (improve model fit) on the validation data over 3 successive epochs, then we terminate model training.

We trained the models on Tensor Processing Unit (TPU) instances on the Google Cloud. TPUs are custom-designed Application-Specific Integrated Circuits processors that are better suited to the training and use of neural networks than Central Processing Units and Graphical Processing Units. In our testing, we found that TPU instances dramatically reduced training times for both neural embedding models.

WEB APPENDIX D: LEAVE-ONE-OUT CROSS-VALIDATION

Cross-validation is an out-of-sample testing procedure for assessing the performance of a statistical model. Leave-One-Out Cross-validation extends n-fold cross-validation by estimating the posterior probability of each observation (the validation sample in the fold) using a model estimated on all other observations in the data. LOOCV is more efficient on two fronts—(1) all the data, except for a focal observation, are used to estimate the model; and (2) all the data (across the folds of the validation exercise) are used to test the model. In contrast, in typical holdout sample validation, data are sacrificed in both estimation and testing, which reduces the power of both estimation and testing.

In a Bayesian context, Gelfland et al. (1992) propose an information-sampling recipe for the computation of the pointwise posterior probabilities given the prior (Θ) and the data:

(D1) $$p(y_i|y_{-i}) = \int P(y_i|x_i,)P(\theta|y_{-i}, x_{-i}) \,, i = 1, \dots N,$$

where y_i is the *i*th observation of the dependent variable, x_i is the *i*th observation of the independent variables, y_{-i} is the vector of dependent variables except the *i*th observation, and x_{-i} is the matrix of independent variables excluding the *i*th observation. Gelfland et al.'s algorithm was refined by Vehtari et al. (2017) to incorporate Pareto smoothing to reduce the influence of outliers.

The expected log posterior density (elpd) of a model is a measure of its predictive accuracy. We use Vehtari et al.'s (2017) method to compute the LOOCV pointwise posterior probabilities, which we use to compute the LOOCV estimate of the elpd of the model:

(D2) $$\text{elpd}_{loo} = \sum_{i=1}^{N} P(y_i|y_{-i}).$$

As is traditional and conventional, we report this statistic in the format of an information criterion by multiplying it by -2 (see Kim et al. 2020 for similar practice).

WEB APPENDIX E: BENCHMARK MODELS

We specify two benchmark models from the Natural Language Processing literature that cannot be used to infer partworths in a utility model but provide a baseline for how much information is captured by the product embedding models: Eliashberg et al. (2007), who propose using Latent Semantic Analysis factors, and Toubia et al. (2019), who propose using Latent Dirichlet Allocation topic intensities. These are labelled BM2 and BM3 respectively in the main text of the paper.

The Latent Semantic Analysis model (a factor analysis of the tf-idf matrix; henceforth LSA) decomposes documents into latent factors and infers the extent to which the factors feature in a document (i.e., the factor intensity of the document). Topic models such as the Latent Dirichlet Allocation (LDA) model are an alternative to LSA. Topic models summarize the topics (issues) discussed in a document and infer the extent to which the topics feature in a document (i.e., the topic intensity of the document) (Blei 2012).

We constructed LSA feature loadings and LDA topic intensities for inclusion in the benchmark models as follows. We conducted a principal components analysis of the document-term matrix. We used the elbow point of the scree plot of the proportion of the variance explained by the left-singular vectors. As the elbow point of the scree plot was at 4 vectors, we used the 4 largest left-singular vectors of the document-term matrix (the LSA feature loadings). To decide on the number of topics in the LDA model, we selected 10,000 random wine descriptions (about 8.34% of the 119,955 wine descriptions) as a holdout sample. We estimated 10 different LDA models where we varied the number of topics in multiples of 4, from 4 to 40. We constructed a scree plot of the perplexity of the LDA models on the holdout sample and chose the elbow point of the scree plot as the optimal number of topics. We found 8 topics to be optimal.