Extracting Characteristics from Product Images and its Application to Demand Estimation

Thomas W. Quan
Wayfair[*]

Kevin R. Williams
Yale School of Management
and NBER[†]

December 2021

Abstract

For many consumer products, demand has an inherent visual component. Consumers are concerned with, not only, the function of a product, but also the style and look of a product. However, our ability to convert these visual characteristics into measurables for analysis has, so far, been limited. By incorporating machine learning techniques and recent advances in econometrics, we are able to extract useful information from product images for use in the estimation of discrete-choice demand models. We find that including this information results in more sensible price elasticity estimates and improved out-of-sample prediction.

[*] Email correspondence: thomas.quan.1@gmail.com and kevin.williams@yale.edu.

[†] We thank online retailer for allowing us to collect the data used in this study.

1 Introduction

In many empirical demand models, products are modeled in characteristic space (Lancaster 1966, Berry 1994, Berry, Levinsohn, and Pakes 1995).[1] In such models, a product consists of a set of characteristics and consumers have preferences over these characteristics. In theory, every minute detail about a product can be captured and influences the utility a consumer derives from that product. However, in practice, accurately and concisely describing a product's characteristics can be quite difficult. As a result, data sets record only a subset of the universe of characteristics that compose a product. The characteristics captured in data will tend to be the characteristics that were the easiest to quantify. There are two reasons this may be problematic. First, the characteristics that are easily quantifiable may not be very important from the consumer's point of view, leaving many sources of product differentiation hidden from the standpoint of the practitioner. Second, the subset of characteristics recorded may still be quite large, sometimes outnumbering observations. The researcher is then required to further select among the observed characteristics for those variables deemed most important.[2]

Product images represent a rich and, largely, untapped source of product characteristic information. For many consumer products, demand has an inherent visual component. Consumers are concerned, not only, with the function of a product, but also the style and look of a product. For example, in art and fashion goods, the style and look of a product are among its primary attributes. Even for other goods, the product's image, on packaging or on the retailer's website, is often one of the primary sources of product information available to the consumer. A product's style and look can be easily conveyed

[1] Alternatively, demand can be modeled in product space. The constant elasticity of substitution (CES) model, a common product space model, has been shown to have a representation in characteristic space (Anderson, De Palma, and Thisse 1989, Hortaçsu and Joo 2018)

[2] Methods for model or variable selection have a long history in economics and statistics. Common approaches include information criterion methods, such as Akaike information criterion (AIC) and Bayesian information criterion (BIC), regularization methods, such as Lasso, and dimensionality reduction methods, such as principle component analysis (PCA). For an overview of these tools see Ng (2013).

to consumers through images, hearkening back to the well-known proverb "a picture is worth a thousand words." However, it is these combinations of characteristics that are often difficult to quantify. Take, for example, color. Suppose the main color of a product is blue. How would you further describe this blue? Light? Medium? Dark? How about the spectrum of shades in between? Since these kinds of distinctions matter to consumers, they should be quantified in the analysis.

Combining tools from machine learning and recent econometric advances by Chernozhukov, Chetverikov, Demirer, Duflo, Hansen, Newey, and Robins (2018), we show how product images can be incorporated into standard demand estimation techniques, such as Berry (1994). We find that product images contain information that is highly predictive of observed demand and including this information in the estimation results in significantly improved demand estimates. In particular, we find estimated price elasticities that are much more reasonable when product images are included compared to estimates using only traditional tabular data. We also find improved out-of-sample prediction when product images are included.

Product images can be thought of as an arrangement of pixels contained in a three dimensional matrix: height, width, and depth (color). Using the standard RBG color model, depth has three levels representing the amount of red, blue, and green contained in each pixel. One approach would be to simply flatten these matrices to create $H \times W \times D$ characteristic variables for each image. We could then include them as characteristics in our demand model and use traditional regression tools to obtain parameter estimates. However, this will perform poorly because (one-third of) an individual pixel in isolation contains very little information leading to tens of thousands of covariates with very little explanatory power. Additionally, the number of characteristics explodes exponentially with the dimensions of the image, leading to sample size concerns. Instead, we use convolutional neural networks (ConvNets or CNNs) to reduce each image's dimensionality and extract features important to consumers. An interesting advantage of this technique, particularly for products where demand is highly visual, is that the process of selecting which characteristics to include can largely be taken out of the practitioner's subjective hands.

We demonstrate the usefulness of images in demand estimation using point-of-sale transaction data containing the footwear sales of a major online retailer. This data first appeared in Quan and Williams (2018). Each transaction identifies the specific product purchased and the price paid. Each product can be matched with a set of pre-coded characteristics, such a review ratings, color, brand, and category. Each product can also be paired with a corresponding $102 \times 136 \times 3$ thumbnail image of the product.[3] All of the product images are of similar quality, with each image taken from the same angle, with the same lighting, and presented against the same background. This is important because, as highlighted by Zhang, Lee, Singh, and Srinivasan (2017), image quality may influence consumer demand with higher quality images resulting in higher demand, holding the actual product constant. The consistent quality of the images in my application should alleviate this concern.

As a first pass, we use the pre-coded characteristics and the product images, separately, to predict a measure of sales (logit mean utilities). Comparing the predictions, we find that using product images results in superior in-sample and out-of-sample fit, even when a flexible machine learning model (random forest) is used with the pre-coded data. That is, image features extracted using a ConvNet explains a larger portion of the variation in sales relative to easily quantifiable characteristics. This suggests that product images contain a great deal of information pertinent to consumer demand.

However, the parameters estimated by the naive application of machine learning

[3] The size and quality of images available online varies widely and has increased over time. Large, high quality images are relatively inexpensive to produce. For example, a typical smartphone has a 12+ megapixel (12 million pixels) camera. However, user download speeds constrain the practical sizes of images used in online retail. Retailers typically employ smaller thumbnail images to appear on search pages and larger images to appear on the main product page. In my application, we employ the smaller thumbnail images, which have dimensions $102 \times 136 = 13,872$ pixels, while the larger images that appear on this retailer's product pages are of dimensions $525 \times 700 = 367,500$ pixels.

should not be interpreted as measuring causal effects. Because the primary goal of machine learning is prediction, these tools face a trade-off between bias (regularization) and variance (overfitting). Both overfitting and the regularization intended to limit it generate bias in the estimates of our parameters of interest, such as the price coefficient. We address this by employing double/debiased machine learning (DML) (Chernozhukov, Chetverikov, Demirer, Duflo, Hansen, Newey, and Robins 2018). DML combines two techniques. First, the effects of a (potentially high-dimensional) vector of characteristics are partialled out, in the spirit of Frisch-Waugh-Lovell, using standard machine learning techniques. This isolates the parameters of interest from the other parameters and allows for the creation of Neyman-orthogonal moments to identify them. These moments are less sensitive to the estimates of the other parameters. Combined with a creative use of sample splitting, they show the bias induced by machine learning techniques can be removed from the estimates of the parameters of interest. Additionally, the moments can be formulated to allow for the inclusion of instrumental variables to control for endogeneity.

The inclusion of product images results in dramatically improved demand estimates. The mean estimated product-level price elasticity is -1.3 using easily quantifiable characteristics and -6.2 using the product images as characteristics. In my application, consumers face very large choice sets with most products having very close competitors, including products that are only slight variations of the same basic shoe model. Therefore, the higher price sensitivity seems to be more reasonable. Additionally, the movement in the price coefficient across specifications is consistent with omitted variable bias. In the presence of omitted variable bias, estimates of the price coefficient tend to be biased upwards, i.e. consumers are estimated to be too inelastic, because prices tend to be positively correlated with desirable omitted/unobserved characteristics. Theoretically, with a set of strong and valid instruments, the remaining omitted variables should not affect the estimates of the parameter of interest. However, in practice, finding such instruments is notoriously difficult and standard instruments used in demand estimation, such as those used in this paper, suffer from well known flaws. More specifically, the instruments commonly used in empirical applications may suffer from weak instrument problems or fail to satisfy the requisite exclusion restrictions, i.e. are not valid. These issues become particularly salient in my setting, where the pre-coded characteristics are likely to omit or inadequately represent many characteristics of first-order importance to consumer demand.

Machine learning techniques have been receiving increased attention in economics, but its implementation has been limited. One of the main drawbacks is that because the goal of machine learning is prediction, the estimated coefficients generally cannot be interpreted as estimates of the causal effects. One approach to identify the causal effect of an event, policy, or intervention is laid out in Varian (2014). First, identify an event of interest. Then with data from a period prior to the event, train a machine learning model and use it make out-of-sample predictions for a time period after the event. Assuming that the only difference between the pre and post-event time periods is due to the event, the causal effect can be measured as the difference between the machine learning prediction and the observed outcomes. An example of this approach in the demand estimation context can be found in Bajari, Nekipelov, Ryan, and Yang (2015). Their main focus is on the benefits of machine learning for variable selection in data with a very large number of possible explanatory variables, potentially greater than the number of observations, and for making out-of-sample predictions. They use the out-of-sample predictions to estimate the causal impact of promotion on sales. However, whereas their parameter estimates do not have a causal interpretation, we are able identify structural parameters of interest, namely the price coefficient.

The DML approach derives from a series of papers demonstrating the usefulness of using Lasso for variable selection and partialling out high-dimensional nuisance parameters (Belloni, Chen, Chernozhukov, and Hansen 2012, Belloni, Chernozhukov, et al. 2013, Belloni, Chernozhukov, and Hansen 2014). Sample splitting has also allowed for progress in the identification of causal effects using tree methods. For example, Athey and Imbens (2016) and Wager and Athey (2017) develop methods to identify heterogeneous causal treatment effects in experimental or observational data using regression trees and random forest algorithms, respectively.

Additionally, most of the literature has focused on the use variables that have already been quantified, but there are a couple of notable exceptions. Gentzkow, Kelly, and Taddy (2017) provide an overview of techniques to extract meaningful information from text. Zhang, Lee, Singh, and Srinivasan (2017), use image processing techniques to first classify images by measures of image quality. They then use these classifications to examine how image quality effects consumer demand. They find that better quality images result in higher demand, holding the actual product constant. My work differs in two respects. First, we allow the algorithm to determine the features that are useful for the prediction of demand, rather than classifying images according to a particular set of known attributes. Second, the features we extract are meant to capture actual product differentiation, rather than differences in the image's ability to convey that information to consumers.

The rest of the paper is organized as follows. Section 2 introduces the data. Section 3 outlines the demand model. Section 4 describes the DML procedure and the machine learning tools used to extract characteristics from images. Section 5 presents the estimation results and compares the results for specifications with and without product images and section 6 concludes the paper.

2 Data Overview

The data in this study comes from Quan and Williams (2018). It consists of transaction level observations of the shoe sales from a major online retailer. Each transaction consists of a timestamp, a 5-digit shipping zip code, price paid, a model ID (SKU), and a style ID (color). A product is defined at the model-style level. In this study, we will focus on sales of men's shoes from August 2012 to July 2013 aggregated to the national-month level.

Summary statistics can be found in Table 1. This subset of the data accounts for almost $300 million in revenue and consists of over 3.3 million pairs of shoes sold. These sales are across more than 12.5 thousand SKUs and 31 thousand total varieties (SKU-styles).[4] Style variants per SKU average two, but have a wide range, between 1 and 55.

Table 1: Summary Statistics (Pre-Coded)

	Mean	SD	Min	Max
Price	108.491	77.520	10.49	1,650
Comfort	4.436	0.617	1	5
Look	4.701	0.403	1	5
Overall	4.417	0.629	1	5
No Review	0.370	0.483	0	1
Total (Annual)				
- SKUs Stocked	12,519			
- SKU-Styles Stocked	31,775			
- Quantity Sold (1000's)	3,364.958			
- Revenue ($ mil)	293.546			
Monthly				
- SKUs Stocked	7,036.583	515.701	5,722	7,485
- SKU-Styles Stocked	14,547.417	847.305	12,708	15,445
- Styles per SKU	2.067	1.710	1	55
- Quantity Sold (1000's)	280.413	31.047	238.175	355.536
- Revenue ($ mil)	24.462	2.778	21.392	31.629

Monthly sales average around $25 million in revenue and 280 thousand pairs of shoes sold. As is usual in retail, sales are cyclical with quantity and revenue peaking in December, during the holiday season, and reaching its low in February. The data also suggest a great deal of turnover in products. Of the 31 thousand varieties observed over the course of the 12 month sample, an average of only 14.5 thousand products are in stock at any given point in time. Interestingly, the number of unique varieties in-stock does not follow the cyclical pattern of sales. Instead it is growing steadily over time. This is consistent with

[4] Among all shoes, men's shoes account for about one third of revenue, quantity sold, and varieties for this online retailer.

the trend of increasing variety in consumer goods, particularly among online retailers.

Each product has a number of pre-coded characteristics. These include sale prices, primary color, category,[5] brand, and review data on look, comfort, and overall appeal. The average listed price is \$108, but there is a great deal of variation in prices which range from \$10.49 to \$1, 650. Weighted by sales, the average price is \$87.23. Review ratings range from 1 (low) to 5 (high) and average 4.4, 4.7, and 4.4 for comfort, look, and overall appeal, respectively. However, these ratings are heavily skewed toward favorable ratings with the 15th percentile being 4 or above in all three categories. Roughly, 37% of observations contain no review data. Observations with no reviews tend to occur for newer products and products with fewer style variants.

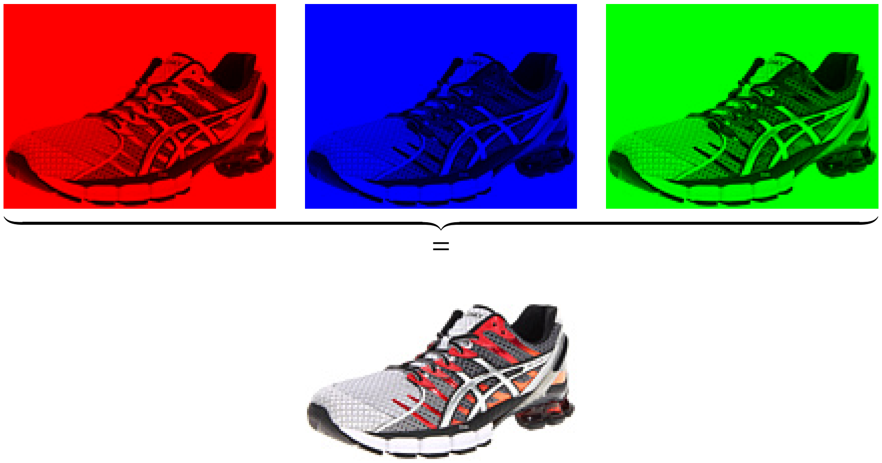


Figure 1: RBG Structure

Additionally, for each product stocked by the online retailer, a thumbnail image of the product was collected. Each image has a height of 102 pixels, a width of 136 pixels, and is color coded using the RBG model for a depth of 3. The RBG color model indicates the levels of red, blue, and green contained in each pixel. Figure 1 illustrates the RGB structure of the images. Each color is given an integer representing the level of that color's saturation between a low of 0 and a high of 255. For example, a black pixel is coded as

[5] Men's shoe categories include boat, boot, climbing, loafers, oxfords, sandals, slippers, and sneakers.

(0,0,0) to indicate the absence of color, while a white pixel is coded as (255,255,255) to indicate complete saturation. Thus, each image can be converted into a three dimensional matrix containing integers between 0-255 with dimensions $102 \times 136 \times 3$.

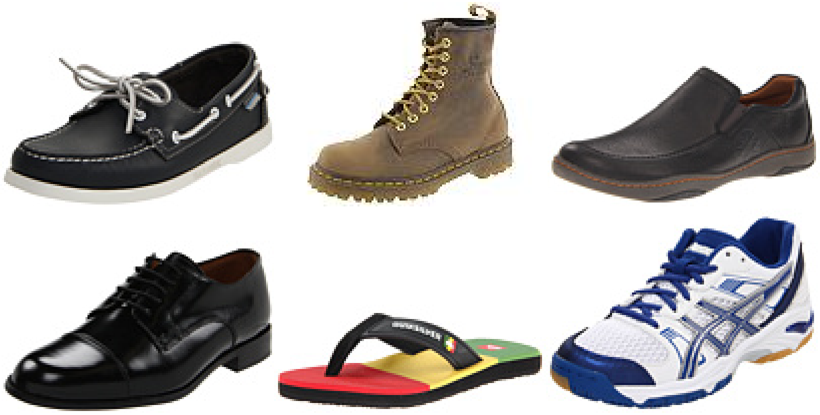


Figure 2: Examples of collected images

As shown in Figure 2, products in these images are displayed against a solid white background and are taken at similar angles with similar lighting. This is convenient for two reasons. First, this should make training the machine learning algorithm easier because the images are not distorted and the algorithm will not have to differentiate between the object of interest and other background noise. Second, as highlighted by Zhang, Lee, Singh, and Srinivasan (2017), image quality can influence how consumers perceive products. Since our online retailer presents consumers with similar high quality images for all products available on their website, this channel should not effect the analysis in my setting.

3 Consumer Demand

To highlight the impact of images on demand estimation and to ease the exposition of the DML procedure, we focus on the relatively simple and well known discrete choice logit framework.[6] A consumer i at time t chooses among $J_t + 1$ alternatives, where $j \in \{1, 2, ..., J_t\}$ is the set of available products at time t and the outside good is indexed as $j = 0$. Consumer i at time t chooses product j if and only if the utility derived from product j is greater than the utility derived from any other product, $u_{ijt} \geq u_{ij't}, \forall j' \in J_t \cup \{0\}$. For ease of notation, we suppress the t subscript in the remaining discussion of the model. Product j provides consumer i with utility equal to

$$u_{ij} = \delta_j + \varepsilon_{ij},$$

where δ_j is the mean utility of product j and ε_{ij} is drawn i.i.d. from a Type-1 extreme value distribution. The mean utility of product j is allowed to be partially linear in characteristics, which can be written as

$$\delta_j = g(x_j) + \alpha p_j + \xi_j,$$

where x_j is a, potentially high-dimensional, vector of product j's characteristics, p_j is product j's price, and ξ_j is the unobserved product quality of product j, which includes characteristics of product j that are unobservable to the econometrician.

Integrating over individuals' Type-1 extreme value error terms obtains the standard logit market share equation, which have following analytic form:

$$s_j = \frac{\exp\{\delta_j\}}{1 + \sum_{j' \in J} \exp\{\delta_{j'}\}}.$$

The market shares are a function of mean utilities, δ_j, where the outside good has utility normalized to zero, i.e. $\delta_0 = 0$. Market shares can then be inverted, as shown in Berry (1994), to yield a (partially) linear equation to estimate:

$$\delta_j = \log(s_j) - \log(s_0) = g(x_j) + \alpha p_j + \xi_j.$$

[6] It is relatively straightforward to extend the use of images to the nested logit framework and we discuss potential further extensions in the conclusion.

While ξ_j is unobserved by the econometrician, it is assumed that some or all of the characteristics in ξ_j are observed by consumers when they are making their purchasing decisions. Since market mechanisms will cause the price to be higher for products that have more desirable characteristics, the omitted characteristics are likely to create positive correlation between the price and the unobserved product quality. If left unaccounted for, this will result in upward bias in the estimated price coefficient. That is, consumers will be estimated to be too price insensitive. The Berry (1994) market share inversion allows for the use of linear instrumental variables techniques to be applied to control for this endogeneity.

In general, the primary parameter of interest to researchers is the coefficient on price, α, because it determines price elasticities and consumer welfare measures in the logit demand model.[7] On the other hand, parameters of the function $g(\cdot)$ are often considered "nuisance" parameters. While the marginal effects of some covariates may be of interest, this is often secondary as their precise values have no effect on price elasticities or measures of consumer welfare. However, including x_j in the estimation is still beneficial to avoid potential biases created by omitting them.[8]

In practice, the function $g(\cdot)$ is usually assumed to be a linear function of characteristics, x_j. However, in my application, the product characteristics are images. Since individual pixels have little meaning in isolation and are unlikely to have meaningful linear effects, it will be important to allow for flexible nonlinearities. For comparative purposes, we will use flexible machine learning estimators on both the pre-code data and the images. However, while techniques from machine learning allow for flexible estimation of the

[7] In a nested logit framework, the primary parameters of interest are α and the nesting parameter, λ. In a random coefficients framework, they are α and the random coefficients, σ.

[8] This has also motivated the inclusion of covariates and proxies that are correlated with demand, but may be difficult to interpret as structural parameters of the utility function. For example, some measure of advertising or promotion will often appear in empirical specifications, but we may not want to interpret this as advertising and promotion affecting the utility the consumer derives from consumption of the product. Similarly, in my application, we include review ratings as proxies for quality. However, we should not interpret this as a structural relationship where a consumer's utility is directly impacted by the experiences of other consumers.

function $g(\cdot)$, they introduce regularization bias into the estimate of the parameter of interest. To address this, we employ recent advances by Chernozhukov, Chetverikov, Demirer, Duflo, Hansen, Newey, and Robins (2018) to obtain a consistent estimate of α. We detail these techniques in the following section.

4 Estimation

In this section, we discuss the estimation of a discrete choice demand model using DML. We begin by discussing the DML technique of Chernozhukov, Chetverikov, Demirer, Duflo, Hansen, Newey, and Robins (2018) in a partially linear setting with instrumental variables (IV) and illustrate that the discrete choice framework naturally fits within their more general setting (Section 4.1). This allows for consistent estimation of the price coefficient in the presence of a high dimensional nuisance parameter and price endogeneity. We then discuss the particulars of estimating the nuisance parameters using ConvNets for image data (Section 4.2) and random forest for traditional pre-coded tabular data (Section 4.3).

4.1 Double/Debiased Machine Learning

Chernozhukov, Chetverikov, Demirer, Duflo, Hansen, Newey, and Robins (2018) derive results for obtaining root-N consistent estimates and making valid inference about a low-dimensional parameter of interest, α, in the presence of a high-dimensional nuisance parameter, $g(\cdot)$. To remove the regularization bias induced by machine learning techniques and obtain point estimates of α that are approximately unbiased and normally distributed, they combine two techniques in a process they call double or debiased machine learning (DML). First, the effects of the high-dimensional vector of characteristics (x) are partialled out, in the spirit of Frisch-Waugh-Lovell,[9] using standard machine learning techniques.

[9] The Frisch-Waugh-Lovell Theorem (Frisch and Waugh 1933, Lovell 1963) states that a subset of parameters in a multivariate regression can be obtained by first partialling out the effects of the other "nuisance" variables, then regressing the orthogonalized dependant variable on the orthogonalized variables of interest. This is implemented by first projecting the nuisance variables (X_2) onto the dependent variable (Y) and the variables

This allows for the construction of Neyman-orthogonal moments. These moments are less sensitive to the estimates of the nuisance parameter. Combined with creative use of a sample splitting, called cross-fitting, they show the regularization bias induced by machine learning techniques can be removed from the estimate of α.

After inverting market shares, the consumer demand model in Section 3 can be framed as a partially linear regression model. To illustrate key ideas, we first present the model where price is exogenous conditional on adequately controlling for x_j,

$$\begin{aligned} \delta_j &= g(x_j) + \alpha p_j + \xi_j, \quad E[\xi|x, p] = 0 \\ p_j &= m(x_j) + v_j, \quad E[v|x] = 0. \end{aligned} \tag{4.1}$$

Since the price endogeneity in discrete-choice demand models is driven by omitted variable bias, this simplified model assumes that all relevant characteristics are observed by the econometrician. Anything remaining in ξ_j is uncorrelated with price. This would hold, for example, if the elements in ξ_j are unobserved by firms when prices are set. The standard moment condition identifying α is

$$E[(\delta - g(x) - \alpha p)p] = 0.$$

Let us start by randomly splitting the data of size N in half. Designate one of these halves the main set, $\tilde{J}$, and the other the auxiliary set, $\tilde{J}^c$. The naive machine learning approach would be to construct a machine learning estimator $\hat{\alpha} p_j + \hat{g}(x_j)$. Suppose an estimate of $g(\cdot)$, $\hat{g}(\cdot)$, is obtained from the auxiliary set. Using the main set and replacing

of interest (X_1). Denote the projection matrix $M_{X_2} = X_2(X_2'X_2)^{-1}X_2'$. The parameters of interest can then be obtained by regressing the orthogonalized dependent variable $M_{X_2}Y$ on the orthogonalized variables of interest $M_{X_2}X_1$. Giles (1984) shows that it is straightforward to extend the Frisch-Waugh-Lovell Theorem to IV settings.

$g(\cdot)$ with $\hat{g}(\cdot)$, the naive estimator of α would be

$$\hat{\alpha} = \left(\frac{1}{n} \sum_{j \in \tilde{J}} p_j^2 \right)^{-1} \frac{1}{n} \sum_{j \in \tilde{J}} p_j(\delta_j - \hat{g}(x_j)).$$

However, Chernozhukov, Chetverikov, Demirer, Duflo, Hansen, Newey, and Robins (2018) show that the estimator $\hat{\alpha}$ does not converge in probability to the true value of α, i.e. $|\sqrt{n}(\hat{\alpha} - \alpha)| \xrightarrow{p} \infty$. To see this decompose the scaled estimation error as

$$\sqrt{n}(\hat{\alpha} - \alpha) = \left(\frac{1}{n} \sum_{j \in \tilde{J}} p_j^2 \right)^{-1} \frac{1}{\sqrt{n}} \sum_{j \in \tilde{J}} p_j \xi_j + \left(\frac{1}{n} \sum_{j \in \tilde{J}} p_j^2 \right)^{-1} \frac{1}{\sqrt{n}} \sum_{j \in \tilde{J}} p_j(g(x_j) - \hat{g}(x_j)).$$

The first term is well behaved in that it is approximately normally distributed and centered around zero, which follows from the central limit theorem. However, the second term is the regularization bias term, which is not centered and typically diverges. This is because the regularization in machine learning estimators induces biases in the estimate $\hat{g}(\cdot)$. For example, a Lasso estimator introduces a penalty for nonzero parameters.[10] This biases parameter estimates in $\hat{g}(\cdot)$ toward zero and the naive estimator of α is sensitive to these biases. In some instances, parameters with true values that are small will be set to exactly zero and, in this sense, regularization reintroduces omitted variable bias.

To remove the regularization bias, DML first partials out the effects of the high-dimensional vector of characteristics, x, from both δ and p, which can be viewed as performing a version of Frisch-Waugh-Lovell. The orthogonal quantities can then be

[10] Lasso is a penalized regression that minimized the following objective:

$$\frac{1}{N} \sum_{i=1}^{N} \left(y_i - \sum_{k=1}^{K} x_{ik} \beta_k \right)^2 + \lambda \sum_{k=1}^{K} |\beta_k|.$$

The first term is the usual mean squared error and the second term is the penalty term. The penalty is determined by a tuning parameter λ that controls the strength of the $\ell 1$ regularization or the amount of "shrinkage." When $\lambda = 0$, Lasso is equivalent to standard linear regression. When $\lambda = \infty$, all parameters are set to 0.

used to construct Neyman-orthogonal moments and an orthogonalized formulation of the estimator. The Neyman-orthogonal moment identifying α is

$$E[(\delta - E[\delta|x] - \alpha(p - E[p|x]))(p - E[p|x])] = 0.$$

Denote the unknown nuisance functions, $m(x) \equiv E[p|x]$ and $\ell(x) \equiv E[\delta|x]$. We can use the auxiliary set to obtain estimates of the nuisance parameters $\hat{\eta} = (\hat{m}, \hat{\ell})$ using machine learning. Denote the residuals $\hat{V}_j = p_j - \hat{m}(x_j)$ and $\hat{W}_j = \delta_j - \hat{\ell}(x_j)$. The empirical moment then uses the main set of observations and the estimators from the auxiliary sample, $\hat{\eta}$, in place of the unknown nuisance functions. The DML estimator of α is then

$$\check{\alpha} = \left(\frac{1}{n} \sum_{j \in \tilde{J}} \hat{V}_j^2 \right)^{-1} \frac{1}{n} \sum_{j \in \tilde{J}} \hat{V}_j \hat{W}_j.$$

This estimator is shown to be $\sqrt{n}$ consistent and approximately centered normal. To see this decompose the scaled estimation error as

$$\sqrt{n}(\check{\alpha} - \alpha) = \left(\frac{1}{n} \sum_{j \in \tilde{J}} V_j^2 \right)^{-1} \frac{1}{\sqrt{n}} \sum_{j \in \tilde{J}} V_j \xi_j + \left(\frac{1}{n} \sum_{j \in \tilde{J}} V_j^2 \right)^{-1} \frac{1}{\sqrt{n}} \sum_{j \in \tilde{J}} (m(x_j) - \hat{m}(x_j))(\ell(x_j) - \hat{\ell}(x_j)) + o_p(1).$$

As before, the first term is approximately normally distributed and centered around zero. The second term is now the product of two estimation errors, which is shown to vanish under a broad range of data generating processes. Finally, the final term, $o_p(1)$, is ensured by sample splitting. The remainder term contains terms like

$$\left(\frac{1}{n} \sum_{j \in \tilde{J}} V_j^2 \right)^{-1} \frac{1}{\sqrt{n}} \sum_{j \in \tilde{J}} \xi_j (m(x_j) - \hat{m}(x_j)).$$

Without sample splitting $\hat{m}$ may depend on ξ_j, which would require additional strong assumptions to ensure $o_p(1)$ (Belloni, Chernozhukov, Fernandez-Val, and Hansen 2017). However, with sample splitting, $\hat{m}$ is estimated independently of the ξ_j in the main set of data and this term vanishes at the appropriate rate.

Notice that convergence above is at rate root-n, which is $N/2$. Splitting the sample has led to a substantial loss in efficiency because only a subset of the data was used to estimate the parameter of interest. However, by reversing the roles of the main and auxiliary sets, referred to as cross-fitting, a second estimator of the parameter of interest can be obtained. Since the two estimators are approximately independent, averaging over them regains full root-N efficiency.

Let us now loosen the conditional exogeneity assumption by allowing p to be correlated with ξ, even after conditioning on x. Suppose we have an instrument z_j that is correlated with price, but uncorrelated with the unobserved quality, ξ_j. The partially linear instrumental variables (IV) specification is

$$\begin{aligned} \delta_j &= g(x_j) + \alpha p_j + \xi_j, \quad E[\xi|x, z] = 0 \\ z_j &= m(x_j) + v_j, \quad E[v|x] = 0. \end{aligned} \tag{4.2}$$

The DML procedure now partials out the effects of x from δ, p, and z. This leads to Neyman-orthogonal moment

$$E[(\delta - \ell(x) - \alpha(p - r(x)))(z - m(x))] = 0,$$

where $\ell(x) \equiv E[\delta|x]$, $r(x) \equiv E[p|x]$, and $m(x) \equiv E[z|x]$. This orthogonalized specification leads to consistent estimates of α using intuition similar to the above conditional exogeneity case. Chernozhukov, Chetverikov, Demirer, Duflo, Hansen, Newey, and Robins (2018) also prove that, under certain regularity conditions, estimators $\check{\alpha}$ constructed using this moment obeys $\sigma^{-1}\sqrt{N}(\check{\alpha} - \alpha) \rightsquigarrow N(0, 1)$, where $\sigma^2 = E[pv]^{-1}E[v^2\xi^2]E[pv]^{-1}$, allowing for inference.

DML Estimation Procedure

In practice, estimation using DML is quite straightforward and proceeds in the following steps

1. With the auxiliary set, $\tilde{J}^c$, use machine learning techniques to fit δ, p, and z on x. This creates the estimators $\hat{\ell}(x)$, $\hat{r}(x)$, and $\hat{m}(x)$, respectively;

2. With the main set, $\tilde{J}$, calculate the orthogonalized quantities

$$\begin{aligned}\check{\delta}_j &= \delta_j - \hat{\ell}(x)\\ \check{p}_j &= p_j - \hat{r}(x)\\ \check{z}_j &= z_j - \hat{m}(x);\end{aligned}$$

3. Using linear IV techniques, such as two-stage least squares (2SLS), regress $\check{\delta}$ on $\check{p}$ using $\check{z}$ as instruments to obtain and estimate of α, denoted $\hat{\alpha}_1$;

4. Reverse the roles of the main and auxiliary sets. Repeat steps (1)-(3) to obtain a second estimate of α, denoted $\hat{\alpha}_2$;

5. Take the average of the two estimates to obtain the final estimate,

$$\hat{\alpha} = \frac{1}{2}\left(\hat{\alpha}_1 + \hat{\alpha}_2\right).$$

4.2 ConvNet: Estimating nuisance parameters from product images

We now discuss the details behind estimating the nuisance parameters $\hat{\ell}(x)$, $\hat{r}(x)$, and $\hat{m}(x)$ with images in my application. Each of the images in my data have the same dimensions ($102 \times 136 \times 3$). Technically, one could flatten the three dimensional matrix representing each image to create a vector of $41,616$ covariates and include all of them as part of the vector of product characteristics, x_j. This would allow for the use of standard estimation techniques, such as OLS or 2SLS. However, obtaining a large enough sample size to include such a large number of covariates may be impractical in many circumstances and this challenge increases exponentially with the size of the image. Perhaps more importantly, an individual pixel contains very little information in isolation. This is because a pixel covers only a minuscule speck of the entire image. As a result, visible features that may be of interest to the consumer, such as brand logos, will require a combination of hundreds or thousands of pixels in a very particular arrangement. Thus, attempting to estimate linear marginal effects of individual pixels is unlikely to yield sensible results.

Instead, to perform dimensionality reduction and to extract features that are predictive of demand, we use a technique from machine learning called convolutional neural networks (ConvNet). ConvNets take images as inputs and output a prediction for some observable outcome chosen by the researcher. Weights/parameters within the ConvNet are chosen to minimize the prediction error for a training data set where outcomes are known.[11] The fitted model can then be used to make out-of-sample predictions. Using product images from the auxilary set, $\tilde{J}^c$, we train three models, predicting the outcomes δ, p, and z. This yields estimates $\hat{\ell}(x)$, $\hat{r}(x)$, and $\hat{m}(x)$ of the functions $\ell(x)$, $r(x)$, and $m(x)$, respectively.

Before getting into the specifics of ConvNets, let me first introduce more simple neural networks. Suppose we would like to predict an outcome y_j using a vector input x_j of dimension $1 \times K$. In standard regression analysis, x_j would be described as a set of regressors and y_j would be called the dependent variable for an observation j in the data set. A simple one-layer neural network would predict y by estimating a set of weights (or neurons) $\hat{\beta}$, i.e. $\hat{y} = \hat{\beta}x'$, where $\hat{\beta}$ has dimension $1 \times K$ for a continuous y.[12] In machine learning, this type of layer is referred to as a fully connected layer, i.e. each observation is connected to each neuron. The weights, $\hat{\beta}$, are chosen to minimize an

[11] We present a very simplified overview here. Interested readers can find a more comprehensive overview in the course notes for Stanford CS course CS231n: Convolutional Neural Networks for Visual Recognition (http://cs231n.stanford.edu/).

[12] If y is discrete, $\hat{\beta}$ has dimension [# classifications] $\times K$.

objective function, such as the mean square error, and this simple setup should give similar results to OLS. However, neural networks would generally include additional layers and an activation function for each layer. For example, consider a two-layer neural network, $\hat{y} = \hat{\beta}_2 \max\{0, \hat{\beta}_1 x'\}$, where $\max\{\cdot\}$ is an activation function that is applied element-wise and $\hat{\beta}_2$ is an additional layer of weights. Activation functions are so named because they determine when a "neuron" is "on" (activated) or "off" and is used in machine learning as a simple way to introduce nonlinearities. The $\max\{\cdot\}$ activation function is called a Rectified Linear Unit (ReLU). An example plot of a ReLU is presented in Table 3. All positive values of $\hat{\beta}x'$ remain unchanged, but any negative values are set to zero. While there are other activation functions, ReLUs have become the most commonly used and will be the activation function applied in this paper.

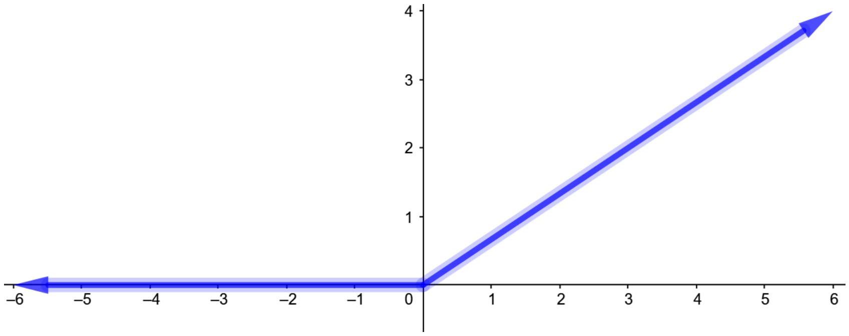
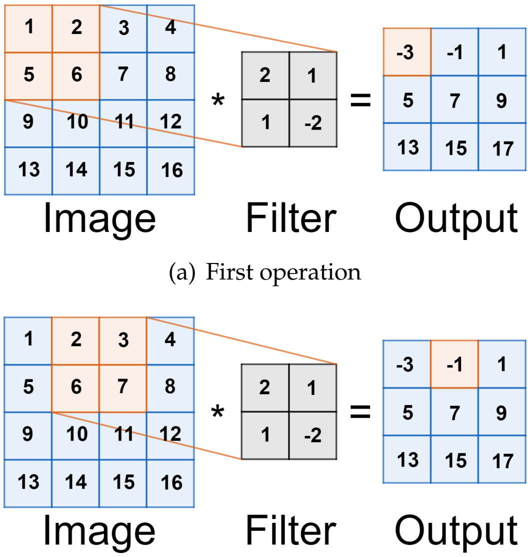


Figure 3: Illustration of a ReLU Activation Function

ConvNets are neural networks that take images as inputs and make use of layers that are more adapted to the structure of images (height, width, depth). ConvNets have four main types of layers: convolutional layers, pooling layers, activation layers, and fully connected layers. The last two types of layers are as above. The activation layers determines when a neuron should be activated and fully connected layers take vectors as inputs and outputs. A convolutional layer can be thought of a set of filters or kernels that are moved over the image. Each filter is assigned a height and width, called a kernel size, and has the same depth as the input. Each weight in the filter is multiplied element-by-element with the input and then summed to produce a number. As it moves over each section of the image, the convolutional layer creates a two dimensional matrix output and the output of different filters are stacked to create the depth of the output. An extremely simplified example of the operations performed in a convolutional layer is illustrated in Figure 4. On the left, is a 4×4 matrix. Suppose this represents an image and that we are applying a 2×2 filter to it. Begin with the 2×2 section in the top left corner of the image (sub-figure (a)). Perform an element-by-element multiplication with the filter then sum over these numbers to produce the first element of the output matrix.[13] Next, move the window to the right by one pixel and repeat, this produces the second element of the output matrix (sub-figure (b)).[14] Continue repeating these operations moving from left to right. At the end of the row, move the window down one pixel and repeat, again from left to right. This continues until the entire image has been covered by the filter.



(a) First operation

(b) Second operation

Figure 4: Illustration of Convolutional Layer Operation

The pooling layer is a dimensionality reduction tool that takes the maximum over a

[13] $1*2+2*1+5*1-6*2=-3$

[14] $2*2+2*1+6*1-7*2=-1$

region along the height and width dimensions. A simplified example of a 2 × 2 max pool operation is illustrated in Figure 5. The max pool operation takes the maximum over elements in its window reducing the dimensionality from 2 × 2 to 1 × 1. For example, in Figure 5, the first 2 × 2 window is highlighted in blue. The maximum over this window is 6 and this becomes the first element of the output matrix. A simple 2×2 max pool cuts the size of both the height and the width dimensions in half, but leaves the depth unchanged.

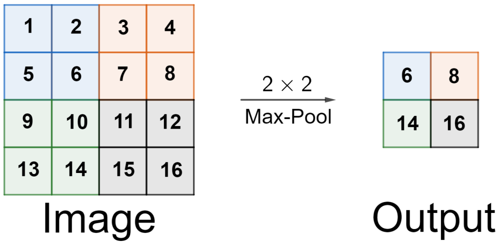


Figure 5: Illustration of a Max Pool Operation

The structure of ConvNets can be become very complex. "Deep" ConvNets can include hundreds or thousands of layers varying between types and sizes. For illustrative purposes, we employ a very simple ConvNet here. We begin with one convolutional layer of dimension 3×3×32, followed by a 2×2 max pool layer. The output of the max pool layer is then flattened to create a vector, which is then passed through two fully connected layers of dimension 1 × 32, and a final 1 × 1 layer that outputs the prediction. Despite this relatively simple structure, the model contains over 3.4 million parameters to “estimate,” which are regularized using an $\ell1$ penalty of 0.01. The ConvNet models are trained in Python using the Keras[15] package and TensorFlow[16] backend. In the next section, we show that the ConvNet model fits well and leads to much more sensible demand estimates compared to standard methods.

[15] https://keras.io/

[16] https://www.tensorflow.org/

4.3 Random Forest: Flexibly estimating nuisance parameters from pre-coded data

To flexibly estimate the nuisance parameters $\hat{\ell}(x)$, $\hat{r}(x)$, and $\hat{m}(x)$ with pre-coded data we use a non-parametric tree-based method called random forest. A basic regression tree can be thought of as a hierarchical series of nested if-else conditions or as recursive binary splitting. At each node, a characteristic is selected and a decision rule splits that characteristic. That is, for a characteristic k and split point s, the parameter space is partitioned into two regions

$$R_1(k,s) = \{X|X_k \leq s\} \quad \text{and} \quad R_2(j,s) = \{X|X_k > s\}.$$

The characteristic k and split point s are chosen to minimize

$$\min_{\hat{y}_{R_1}} \sum_{i:x_i \in R_1(k,s)} (y_i - \hat{y}_{R_1})^2 + \min_{\hat{y}_{R_2}} \sum_{i:x_i \in R_2(k,s)} (y_i - \hat{y}_{R_2})^2.$$

Each of these branches are then partitioned again in the same way and the process is continually repeated. A terminal node is called a leaf. In the limit, each observation of x would be assigned its own leaf and its own prediction of y. This, of course, would have perfect in-sample fit, but the out-of-sample fit would reveal severe overfitting. As a result, in practice, trees are expanded until a certain stopping criteria is met. Common examples include a specific number of splits, a minimum number of observations per leaf, or until the reduction in the objective falls below a certain threshold.

Random forest is an ensemble method that fits multiple regression trees, then averages over their predictions. Randomness is induced across trees by allowing only a random subset of characteristics to be considered at each node. The randomness and averaging across models has been shown to control overfitting and increase prediction accuracy. The random forest estimator is trained in Python using the scikit-learn[17] package.

5 Results

5.1 Feature Extraction

To illustrate that product images contain information that is useful for predicting consumer demand, we estimate the conditional expectation of the mean utilities given observable characteristics,

$$\begin{aligned}\delta_j &= E[\widehat{\delta_j | x_j}] + u_j \\ &= \hat{\ell}(x_j) + u_j\end{aligned}$$

That is, we fit the mean utilities, δ, using observable characteristics, excluding price. We then calculate the mean squared error of the associated predictions. Three specifications are considered. The first specification is a standard linear model estimated by ordinary least squares (OLS) using the pre-coded characteristics. The second specification also uses the pre-coded characteristics, but is fit with a random forest estimator, which captures potential nonlinearities and interactions between covariates. The final specification uses product images as characteristics and is fit using a ConvNet with the structure described in the Section 4.2.

Sales are aggregated to the national level and time horizons are defined to be at the monthly level. Included in the pre-coded characteristics are product ratings for comfort, look, and overall appeal and fixed effects for color and brand. The product ratings are time varying and reflect the scores consumers would observe at the time of purchase. To examine the out-of-sample fit of each specification, we split the sample into a training set and a test set and withhold the test set when fitting the model. The training set contains

[17] https://scikit-learn.org/

the observations from August 2012 to May 2013. The test set contains the observations from June and July 2013, which is 29,843 of 174,569 observations or roughly 17.1% of the sample.

Results from the fitting exercises are presented in Table 2. As a baseline, the first column contains a model fit with a constant only. Comparing the other specifications to the constant only model, it is clear that both sets of data contain information predictive of demand. In the specifications using pre-coded characteristics, the random forest estimator fits the data substantially better than OLS, suggesting important nonlinearities or interactions among the observed pre-coded characteristics. Comparing random forest to the ConvNet, we see that while the random forest specification has slightly better in-sample fit, the ConvNet performs better out-of-sample. The out-of-sample fit will be important because of DML's reliance on sample splitting.

Table 2: Summary of ConvNet Fit (MSE)

		Pre-Coded (Tabular)		Images
	Constant	OLS	Random Forest	ConvNet
Training Set (In-Sample)	1.643	1.207	0.568	0.578
Test Set (Out-of-Sample)	1.622	1.222	1.031	0.944

Across all specifications, we observe that the in-sample fit is better than the out-of-sample fit. While this is unsurprising, it suggests a degree of overfitting, which is particularly apparent for the machine learning estimators. However, both machine learning estimators produce superior out-of-sample performance compared to the standard linear specification. In order to use these machine learning techniques for demand estimation, we will need to address both overfitting and the regularization bias these techniques induce to reduce overfitting. DML allows us to address these concerns by first estimating the nuisance parameters on an auxiliary set of data. The estimated nuisance parameters from the auxiliary set are then used to predict values for the main set and construct the Neyman-orthogonal moments. Concern over overfitting is eliminated because the in-sample (auxiliary set) predictions are not used directly in the estimation of the parameters of interest, while the Neyman-orthogonal moments removes the influence of regularization bias on the estimates of the parameters of interest.

5.2 Logit Demand

We now discuss the main demand estimates. Price is instrumented for using typical BLP-style instruments. Included are the number of available styles for a particular shoe model and the number of within-category own and competitor products available for sale. The standard linear specification is estimated by 2SLS using the above instruments to instrument for price endogeneity. The partially linear IV specification (Equation 4.2) is estimated using DML to obtain consistent estimates of the price coefficient, α. We estimate the nuisance parameters in two ways. First, with the pre-coded characteristics we use a random forest estimator. Second, with the product images as characteristics we estimate a ConvNet model. To implement DML, the training set is split into an auxiliary set and a main set. Each split contains half of the unique SKUs in the training set. As discussed in Section 4.1, estimates of the nuisance parameters are obtained from the auxiliary set. The resulting estimators are then used to partial out the nuisance parameters in the main set, orthogonalizing the remaining variables. Finally, a point estimate of α, $\hat{\alpha}_1$, is obtained by 2SLS using the orthogonalized quantities. The role of the auxiliary and main sets are then reversed to obtain a second estimate, $\hat{\alpha}_2$, and the two estimates are averaged to obtain the final estimate, $\hat{\alpha}$.

Demand results are summarized in Table 3. The top panel presents estimates for the parameter of interest α. All of the estimated price coefficients have the correct sign and are highly statistically significant. The two sets of estimates using pre-coded characteristics, 2SLS and random forest, very similar, whereas using product images as characteristics results an estimate of α that is much larger in magnitude. That is, consumers are estimated to be much more price sensitive in the specification using product images.

Table 3: Demand Estimates

	Pre-Coded (Tabular)		Images
	2SLS	Random Forest	ConvNet
Price Coefficient (α)	-0.012***	-0.013***	-0.057***
	($4e^{-4}$)	(0.001)	(0.004)
Price Elasticity			
Mean	-1.317	-1.367	-6.153
Standard Deviation	0.941	0.977	4.396

Price elasticities implied by the logit model can be calculated as

$$\epsilon_j = \frac{p_j}{s_j}\frac{\partial s_j}{\partial p_j} = \alpha \cdot p_j \cdot (1 - s_j).$$

Using this equation, a product level elasticity is calculated for each product, in each of the specifications. The bottom panel of Table 3 presents the mean and standard deviation of the estimated product-level price elasticities. The average product-level price elasticity implied by the estimates using product images is roughly four times greater in magnitude than the estimates from the pre-coded characteristics, -6.2 vs. -1.3. The estimates stemming from the model using product images seems to be more much more reasonable than the estimates using pre-coded characteristics. In particular, the specification using the pre-coded characteristics imply consumers are much too price insensitive. Consumers in this market face very large choice sets with most products having very close competitors, including variants of the same model in a different style/color. This is further illustrated in the distribution of elasticities implied by the two sets of data. Figure 6 plots a historgram of implied elasticities from the pre-coded (random forest)[18] and image data estimates. A vertical black line is drawn at $\epsilon = -1$. As we can see, using the results from pre-coded data,

[18] Elasticities computed using the 2SLS estimates are slightly more inelastic, but are otherwise identical in shape.

a large fraction of products, over 40%, are estimated to have inelastic demand. Compared to the results using image data, which implies less than 0.07% of products have inelastic demand.

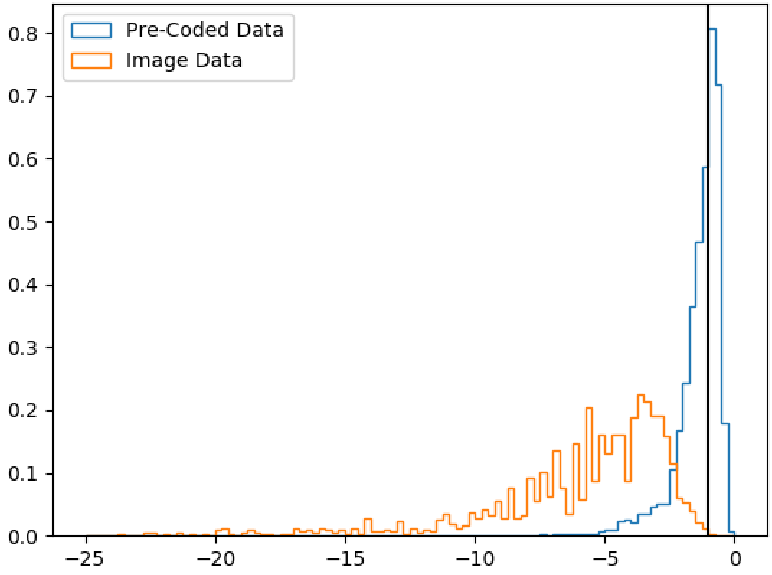


Figure 6: Histogram of Elasticities

Note that the same set of instruments are used with both sets of data. Theoretically, with a set of strong and valid instruments, the instruments should be identifying the parameter of interest and the remaining unobserved/omitted variables should not impact the resulting estimate. However, in practice, finding such instruments is notoriously difficult. The ideal set of instruments, information on firm-level cost shifters, is rarely available. As a result, empirical applications have settled for instruments that rely on fairly strong assumptions. Two commonly used instruments, BLP (used in this paper) and Hausman instruments, suffer from well known issues that are particularly salient in my application.

BLP-style instruments are constructed by taking the sum of characteristics across competing products (Berry, Levinsohn, and Pakes 1995). These instruments are intended to capture the density of products, i.e. the fierceness of competition, where this product is located in characteristic space. Competition affects the markups firms can charge, which implies the instrument will be correlated with price. But since the utility derived from a product is not influenced by the characteristics of other products, the instrument will be uncorrelated the unobserved quality, hence valid. However, when primary product attributes are not adequately captured in the data, competition in the observed characteristics will have only a weak relationship with price, leading to a weak instruments problem. Further, this may be exacerbated in settings with large choice sets, as the dependence of markups on the characteristics of other products decreases with the size of the choice set. Thus, BLP instruments may lose identifying power in settings with a large number of products, such as mine (Armstrong 2016).

Hausman instruments take the average prices of the same product in other markets as an instrument for price in the target market (Hausman 1996).[19] Prices will be correlated across markets through the costs of the firm. The validity argument then assumes that the remaining differences in the product's price across markets is driven local demand shocks that are independent across markets. This assumption is easily violated by, for example, national or regional advertising campaigns. Additionally, if important product attributes are omitted from the demand specification, the value of these omitted characteristics will be captured in the price of the product in all markets, leading to correlated demand shocks and invalidity of the instruments.

Because price tends to be positively correlated with the unobserved quality, we would expect omitted variable bias and weak instruments to lead to upward bias in the estimated price coefficient. Thus, the movement in the estimated price coefficient across specifications suggests the model using product images may be addressing omitted variable bias

[19] Hausman instruments cannot be used here because the online retailer charges uniform prices. In general, online retailers hesitate to (3rd degree) price discriminate for fear of public backlash.

by capturing additional characteristics not included in the pre-coded data.

6 Conclusion

Product images represent a rich and, largely, untapped source of product characteristic information. These images can quickly convey a tremendous amount of information to consumers and, in retail markets, are one of the primary sources of information available to consumers. However, our ability to convert these visual characteristics into measurables for analysis has been limited.

In this paper, we illustrated how product images can be included in structural demand estimation. Using tools from machine learning, features that are predictive of observed demand were extracted from product images. Combining this with recent advances in econometrics allow for the consistent estimation of a small set of demand parameters, in particular, the price coefficient, which allows for the consistent estimation of price elasticities.

Compared to traditional pre-coded tabular data, we find that including product image data results in more reasonable estimates of price elasticities and improved out-of-sample fit. Given the logit functional form assumption, estimates derived from a set of strong and valid instruments should result in similar estimates of the structural price coefficient. However, such instruments are notoriously difficult to find and standard instruments used in demand estimation suffer from well known issues. Because the same set of instruments is used in the estimation of both the pre-coded and image specifications, it is likely that including the image data addresses a problem with omitted variables present in the pre-coded data.

To simplify the exposition and highlight the power of using images as data in demand estimation, we perform the analysis in the context of a logit discrete-choice model. This places strong assumptions on the substitution patterns of consumer, such the well known independence of irrelevant alternatives (IIA) assumption. This can be loosened slightly by assuming a nested logit framework and it is straightforward to apply the DML procedure to the inverted nested logit market shares found in Berry (1994). This model would have two parameters of interest, the price coefficient and the nesting parameter, and one would simply need to orthogonalize the additional endogenous variable, the log of market shares conditional on nest, and include it in the IV regression.

However, extending DML to the random coefficients framework of Berry, Levinsohn, and Pakes (1995) may be more difficult. One of the main benefits of machine learning, the ability to flexibly approximate arbitrary functions, may actually make it difficult to separately identify random coefficient parameters on image features. This is because, as shown by Salanié and Wolak (2019), the random coefficients can be approximated in a linear estimator by including quadratic combinations of the regressors. If the machine learning estimator captures these quadratic relationships, it will soak up the variation in the data that identifies the random coefficients. Additionally for image data, while ConvNets can be described as feature extraction, it produces a prediction of the intended target rather than a set of traditional characteristics. One could pull out an intermediate layer and treat these as characteristics in another estimator, but it is unclear what the interpretation of these "characteristics" would be. In particular, it is unlikely that one vector would correspond with color and another to category. Instead each vector would be some indeterminate nonlinear function of all features in the image.

We leave it to future research to determine if a random coefficients model can be estimated while including image data. One possible solution may be to combine the two types of data. In the "outer loop," estimate key random coefficients from the pre-coded characteristics. Then in the "inner loop," use the combined pre-coded and image data to fit the mean utilities. However, one would have to address the stochastic nature of machine learning estimators to ensure convergence. Additionally, if it converges, it will likely to take a significant time investment because the machine learning models need to be re-estimated for each iteration of the loop.

References

Anderson, S. P., A. De Palma, and J.-F. Thisse (1989): "Demand for differentiated products, discrete choice models, and the characteristics approach," *The Review of Economic Studies*, 56(1), 21–35.

Armstrong, T. B. (2016): "Large market asymptotics for differentiated product demand estimators with economic models of supply," *Econometrica*, 84(5), 1961–1980.

Athey, S., and G. Imbens (2016): "Recursive partitioning for heterogeneous causal effects," *Proceedings of the National Academy of Sciences*, 113(27), 7353–7360.

Bajari, P., D. Nekipelov, S. P. Ryan, and M. Yang (2015): "Demand estimation with machine learning and model combination," *Working Paper*.

Belloni, A., D. Chen, V. Chernozhukov, and C. Hansen (2012): "Sparse Models and Methods for Optimal Instruments With an Application to Eminent Domain," *Econometrica*, 80(6), 2369–2429.

Belloni, A., V. Chernozhukov, et al. (2013): "Least squares after model selection in high-dimensional sparse models," *Bernoulli*, 19(2), 521–547.

Belloni, A., V. Chernozhukov, I. Fernandez-Val, and C. Hansen (2017): "Program Evaluation and Causal Inference With High-Dimensional Data," *Econometrica*, 85(1), 233–298.

Belloni, A., V. Chernozhukov, and C. Hansen (2014): "Inference on treatment effects after selection among high-dimensional controls," *The Review of Economic Studies*, 81(2), 608–650.

Berry, S., J. Levinsohn, and A. Pakes (1995): "Automobile Prices in Market Equilibrium," *Econometrica*, 63(4), 841–890.

Berry, S. T. (1994): "Estimating Discrete-Choice Models of Product Differentiation," *The RAND Journal of Economics*, 25(2), 242–262.

Chernozhukov, V., D. Chetverikov, M. Demirer, E. Duflo, C. Hansen, W. Newey, and J. Robins (2018): "Double/debiased machine learning for treatment and structural parameters," *The Econometrics Journal*, 21(1), C1–C68.

Frisch, R., and F. V. Waugh (1933): "Partial Time Regressions as Compared with Individual Trends," *Econometrica*, 1(4), 387–401.

Gentzkow, M., B. T. Kelly, and M. Taddy (2017): "Text as data," *Working Paper*.

Giles, D. E. (1984): "Instrumental variables regressions involving seasonal data," *Economics Letters*, 14(4), 339–343.

Hausman, J. A. (1996): "Valuation of new goods under perfect and imperfect competition," in *The economics of new goods*, pp. 207–248. University of Chicago Press.

Hortaçsu, A., and J. Joo (2018): "Semiparametric estimation of a CES demand system with observed and unobserved product characteristics," *Working Paper*.

Lancaster, K. J. (1966): "A new approach to consumer theory," *Journal of Political Economy*, 74(2), 132–157.

Lovell, M. C. (1963): "Seasonal Adjustment of Economic Time Series and Multiple Regression Analysis," *Journal of the American Statistical Association*, 58(304), 993–1010.

Ng, S. (2013): "Variable selection in predictive regressions," in *Handbook of economic forecasting*, vol. 2, pp. 752–789. Elsevier.

Quan, T. W., and K. R. Williams (2018): "Product variety, across-market demand heterogeneity, and the value of online retail," *The RAND Journal of Economics*, 49(4), 877–913.

Salanié, B., and F. A. Wolak (2019): "Fast," robust", and approximately correct: estimating mixed demand systems," *Working Paper*.

Varian, H. R. (2014): "Big data: New tricks for econometrics," *Journal of Economic Perspectives*, 28(2), 3–28.

Wager, S., and S. Athey (2017): "Estimation and inference of heterogeneous treatment effects using random forests," *Journal of the American Statistical Association*, (Forthcoming).

Zhang, S., D. Lee, P. V. Singh, and K. Srinivasan (2017): "How Much Is an Image Worth? Airbnb Property Demand Estimation Leveraging Large Scale Image Analytics," *Working Paper*.