Bayes Goes Big: Distributed MCMC and the Drivers of E-Commerce Conversion

Bastiaan C. Dunn[a], Flavius Frasincar[a,*], Vladyslav Matsiiako[a], David Boekestijn[a], Finn van der Knaap[a]

[a]*Econometric Institute, Erasmus University Rotterdam, P.O. Box 1738, NL-3000 DR Rotterdam, the Netherlands*

Abstract

This work researches the drivers of e-commerce conversion of one of the largest e-commerce companies in the Netherlands. We focus on product page conversion, i.e., the probability that a customer who visits a specific product page also buys the product. This probability differs between products, which we explain by a variety of factors like pricing, delivery times, reviews, seller type, and the quality of the content. Understanding the drivers of conversion is the first step in increasing it, and therefore an important step in increasing overall sales. We describe the process of transforming Big Data into valuable insights using Bayesian statistics and apply a Bayesian Binomial model to a dataset of 15 million records using a distributed MCMC algorithm. In addition, we apply a simple Binomial GLM on a small sample of these 15 million records for comparison. We find that using the full 15 million records results in a significant reduction in the variance of the estimated parameters. In terms of the actual drivers of e-commerce conversion, we observe that products with competitive pricing, short delivery times, and good content all achieve a higher conversion on average. Furthermore, reviews, average review score, and the number of reviews for a product have a large positive effect on conversion compared to other characteristics. This provides a unique insight into what makes people decide whether or not to purchase a product.

Keywords: MCMC, E-commerce, Conversion prediction, Bayesian statistics, Big Data

1. Introduction

Over the past decade, Big Data has become part of many companies' internal processes and has proven itself to be a valuable asset. Ways of storing and retrieving data are developing at an expo-

*Corresponding author. Tel. +31 10 40 81 340; fax: +31 10 40 89 162

Email addresses: b.c.dunn@student.eur.nl (Bastiaan C. Dunn), frasincar@ese.eur.nl (Flavius Frasincar), matsiiako@ese.eur.nl (Vladyslav Matsiiako), boekestijn@ese.eur.nl (David Boekestijn), 573834fk@student.eur.nl (Finn van der Knaap)

nential rate. In addition, tools for analyzing this data are being developed. As a result, Business Intelligence and Analytics have become increasingly important to modern companies (Chen et al., 2012; Ain et al., 2019). In fact, there is much literature about increasing e-commerce conversion, some of them providing proper actionable information (Saleem et al., 2019; Zumstein & Kotowski, 2020; Tong et al., 2022). In this development, the field of statistics, more specifically econometrics, has been awfully quiet. Few attempts have been made to bring econometric models and methods to the playing field of Big Data, despite several sources providing guidelines on the use of Big Data in econometrics (Varian, 2014; Marcellino et al., 2018). In this paper, one of the few methods that combines the two is applied to truly Big Data.

Although combining the seemingly complementary fields of econometrics and Big Data feels natural, there is little literature in this direction. An explanation could be the continuous tension between business, where Big Data exists, and the academic world, where most econometric methods are developed. Another reason could be that the expected returns of combining these two do not justify the investments needed to set up a Big Data environment that enables the use of econometric models. For example, it could be that the general performance of a statistical model does not increase much relative to the extra effort it takes to base the model on Big Data. Whatever the reason, there is very little work on the combination of these two areas.

This research is conducted with a Dutch company that is one of the largest e-commerce companies in the Netherlands. Each unique product they offer has its product page that shows information and images of the product. Once a customer browsing the website visits one of these product pages, it is called a visit. After this visit, the customer can either decide to buy the product or decide not to. The total number of orders of the product divided by the total number of visits is called the conversion.

The success of an e-commerce company is determined by many factors, of which conversion is one of the most important, as a change in its value directly affects sales. Conversion is partially driven by the content of the product page. Consequently, knowing what the effect of product page content is on conversion would be a great way to help grow overall sales. Thus, e-commerce conversion encompasses product page conversion, which is the focus of our paper. We formulate the following research question: *What are the drivers of product page conversion for e-commerce companies?*

Not only can the company sell its products online, but it can also act as a platform for third parties to offer their products. This sometimes leads to a product being sold by multiple sellers.

Customers get the option to decide from which seller they want to buy the product, but an algorithm does make a suggestion; we call it the Best Offer. Most of the customers buy the product from the seller suggested by the Best Offer algorithm as many people do not know they can buy it from other sellers. Consequently, 99% of total sales are made through the Best Offer.

As previously said, the seller who gets the Best Offer is determined by the Best Offer algorithm. This algorithm picks the offer that is "Best" for the customers and takes into account the price, delivery time, and the quality of the seller. Low price, short delivery time, and high quality of the seller are of course preferred. The sellers want to be the Best Offer on as many products as possible, meaning they regularly change their prices to compete with other sellers. So, many products' pages change regularly with respect to the seller, and thus often with respect to price, delivery time, and other details.

Without a doubt, it is essential to know what drives conversion. Why is the conversion higher for some products than for others? Educated guesses are being made on whether a characteristic has a positive or negative effect on conversion, but it is difficult to, for example, guess what people find more important between price and delivery time. In this paper, the effect of conversion of numerous characteristics is quantified. Companies can use these insights to increase the conversion. Moreover, it also has applications in determining the Best Offer because this algorithm needs to know what customers find important in order to pick the Best Offer for these customers. To model the conversion, we compare the performance of a Binomial Generalized Linear Model (GLM) (McCullagh, 1984), a Markov Chain Monte Carlo (MCMC) simulation (Kim & Kim, 2000), and the Consensus MCMC algorithm (Scott et al., 2016) in terms of computation time, out-of-sample performance, and the quality of parameter interpretation. We use a Bayesian approach as it is easily ammendable to parallelism, which is useful for large amounts of data like we have in this work.

This paper contributes to existing literature in the following ways. First, literature concerning the drivers of e-commerce conversion mainly focuses on customer behavior (Zerbini et al., 2022; Necula, 2023; Lin et al., 2023). This research explains the conversion based on characteristics that are independent of customers, which makes the gained insights into conversion much easier to implement in existing business processes. For example, from a technical standpoint, distinguishing customers is difficult to implement and consumes a lot of (IT) resources. Also, companies are not always allowed to distinguish based on customers; in most cases, they have to uphold the same price for a product for all customers. Second, we describe the complete process of transforming

data into business value. While most literature focuses on a single aspect of this process, without clearly linking it to other parts of the process, we describe the whole process and keep in mind the goal of adding value to both business and science. Third, we use the Consensus MCMC algorithm. The consensus MCMC algorithm is widely used in the literature (Ni et al., 2020; Buchholz et al., 2023) and most of the works that use this algorithm extend it (Wang et al., 2015; Rabinovich et al., 2015), while there are no papers that simply apply the algorithm and compare it to more simplistic methods. In addition, within the MCMC algorithm, we compare simple and information-based averaging, which, to the best of our knowledge, has not been done before.

The rest of this paper is structured as follows. We continue with an overview of related work in Section 2. We then describe the data used in our experiments in Section 3. Next, data preparation and the three considered models are presented in Section 4. An elaborate discussion and interpretation of the results follows in Section 5. Last, we provide a summary of our research and several suggestions for future work in Section 6.

2. Related Work

In this section, we present previous related literature. We first provide a general overview of basic methods for modeling conversion in Section 2.1. Then, in Section 2.2, we present a Bayesian approach for modeling conversion. Next, Section 2.3 describes a distributed solution for the Bayesian approach. Last, Section 2.4 discusses the three models used in this study.

2.1. General Overview

Conversion can be modeled in many ways, but three methods are commonly used:

- Ordinary Least Squares (OLS) with the dependent variable y/n;
- Binary response with logistic regression;
- Binomial Generalized Least Squares with logistic link function.

The first method models the measured conversion y/n to be linearly dependent on the covariates with symmetrically distributed errors; this is by far the most simplistic method. The second method splits all observations into a set of binary responses. This means that for every visit there is a row in the data with a binary variable indicating whether the product was bought. This results in a dataset with many observations as every visit has its row. The third method uses a Binomial GLM with a logistic link function (McCullagh, 1984).

The OLS approach is compared with the binary response with a logistic regression method by Zhao et al. (2001), who conclude that the logistic regression outperforms the OLS approach. A possible explanation is that the absolute size of y and n is neglected in the first method, as only the ratio y/n is considered. The third method is preferred over the second one because of the large dataset required by the second method. The equations of a Binomial GLM are quite trivial and are therefore not given (but the interested reader is referred to e.g. Dunn & Smyth (2018)).

2.2. Bayesian Conversion Model

Kim & Kim (2000) propose a Bayesian conversion model, which has approximately the same functional form as the Binomial GLM. The main difference is the addition of a random coefficient μ_i. The functional form of the Bayesian conversion model is as follows:

$$y_i \mid p_i \sim Bin(n_i, p_i), \tag{1}$$

$$p_i = \text{logit}(\theta_i) = F(\theta_i) = \frac{1}{1+\exp(-\theta_i)}, \tag{2}$$

$$\theta_i = \boldsymbol{x}_i^T \boldsymbol{\beta} + \mu_i, \tag{3}$$

$$\mu_i \sim \text{N}(0, \sigma_\mu^2). \tag{4}$$

Consider regression data $(n_i, y_i, \boldsymbol{x}_i),\ i = 1, \ldots, N$, where n_i is the total number of visits for observation i, y_i is the number of converted customers for observation i, and $\boldsymbol{x}_i$ is the vectors of known covariates for observation i. Then, the posterior specification is given as follows:

$$p(\boldsymbol{\theta}, \boldsymbol{\beta}, r_\mu \mid \boldsymbol{y}) \propto p(\boldsymbol{y} \mid \boldsymbol{\theta}) \cdot p(\boldsymbol{\theta} \mid \boldsymbol{\beta}, r_\mu) \cdot p(\boldsymbol{\beta}) \cdot p(r_\mu), \tag{5}$$

where $r_\mu = \sigma_\mu^{-2}$ for ease of notation. The distributions of each part of the above equation are given in Table 1.

Table 1: Distributions of each part of the posterior specification.

Part	Distribution	Density function
$p(\boldsymbol{y} \mid \boldsymbol{\theta})$	$\prod_{i=1}^N Bin(y_i;\, n_i,\, p_i)$	$\prod_{i=1}^N \binom{n_i}{y_i} F(\theta_i)^{y_i} (1-F(\theta_i))^{n_i - y_i}$
$p(\boldsymbol{\theta} \mid \boldsymbol{\beta},\, r_\mu)$	$\prod_{i=1}^N \phi(\theta_i;\, \boldsymbol{x}_i^T\boldsymbol{\beta},\, r_\mu^{-1})$	$\prod_{i=1}^N \left((2\pi)^{-1/2} r_\mu^{1/2} \exp\left[-\frac{r_\mu}{2} (\theta_i - \boldsymbol{x}_i^T\boldsymbol{\beta})^2 \right] \right)$
$p(\boldsymbol{\beta})$	$Flat$	$\propto 1$
$p(r_\mu)$	$Gamma(rate = a,\, shape = b)$	$\frac{a^b}{\Gamma(b)} \exp(-a\, r_\mu)\, r_\mu^{b-1}$

As shown in Table 1, we assume a flat prior for $\boldsymbol{\beta}$ ($p(\boldsymbol{\beta}) \propto 1$); this means we have no prior

belief on the distribution of $\boldsymbol{\beta}$. For the prior of r_μ, a $Gamma(a, b)$-distribution is used, where a is the rate parameter and b is the shape parameter. This is a so-called conjugate prior, which means that it is easily combined with the likelihood function such that it results in a known distribution.

Substituting the parts of Table 1 in the posterior specification, we get the following kernel for the posterior distribution:

$$\begin{aligned} p(\boldsymbol{\theta},\, \boldsymbol{\beta},\, r_\mu \mid \boldsymbol{y}) \propto & \prod_{i=1}^{N} \binom{n_i}{y_i} F(\theta_i)^{y_i} \left(1 - F(\theta_i)\right)^{n_i - y_i} \\ & \prod_{i=1}^{N} \left(r_\mu^{1/2} \exp\left[-\frac{r_\mu}{2} \left(\theta_i - \boldsymbol{x}_i^T \boldsymbol{\beta}\right)^2 \right] \right) \exp(-a\, r_\mu)\, r_\mu^{b-1}. \end{aligned} \tag{6}$$

The above equation gives the relation between the parameters and the data. Although this relation is known to us, it does not automatically mean it is analytically solvable. In this case, it is not, and therefore we need to use an MCMC simulation method.

2.3. Consensus Markov Chain Monte Carlo

MCMC methods construct a Markov chain where the equilibrium distribution of the chain is the target distribution. In Bayesian statistics, it is desired that the equilibrium distribution of the Markov chain is the posterior distribution of the parameters given the data. Although one can often define the functional form of this distribution, this form may not be (easily) derivable; this is frequently the case when the parameter space is highly multi-dimensional. The Gibbs sampling MCMC method (Hastings, 1970) solves this problem. It obtains a set, or chain, of samples in the parameter space that represents the true, full posterior distribution without sampling from it directly. Instead, it sequentially samples from the full conditional posterior distribution of each parameter separately, using the value of a sampled parameter in the subsequent draw. The Gibbs sampler thus benefits from the fact that the full conditional posterior of each parameter is (potentially) known and easier to sample from.

However, drawing from these full conditional posterior distributions may still be difficult. The Metropolis-Hastings algorithm (Hastings, 1970) overcomes this problem by sampling from candidate functions that approximate the density function (Chib & Greenberg, 1995). Of course, the candidate distribution must be easier to sample from. Then, given a draw of the candidate, it accepts this draw based on a calculated acceptance probability.

Even though these algorithms allow for relatively easy sampling from high-dimensional posteriors, an efficient distributed application requires processes to be distributed among workers and

communication between workers to be minimal. An iteration of the MCMC Gibbs sampler needs all previous results and is therefore not suitable in its normal form for this type of application. One of a handful of attempts to distributed MCMC simulation is that of Scott et al. (2016), who propose the Consensus MCMC algorithm. Consensus MCMC consists of two steps. The first step is to split the data (randomly) into groups (called shards), then distribute these among multiple workers and run an MCMC simulation on each shard. The second step is to combine all draws using smart weighting to create the full joint distribution of the parameters. This makes the Consensus MCMC algorithm "embarrassingly parallel" as defined by the MapReduce model (Dean & Ghemawat, 2008). This means that the application is easily distributed over multiple workers and communication between them is minimal. Consensus MCMC is thus computationally less heavy via distributed computing (Scott et al., 2016; Scott, 2017).

Algorithm 1 shows the general steps of the Consensus MCMC algorithm. For the weighting matrix $\boldsymbol{W}_s$, one can use the identity matrix. The draws of different shards are then simply averaged. Another choice for $\boldsymbol{W}_s$ is to use the inverse of the covariance matrix of $\boldsymbol{\theta}_s$. This is a way of information-based averaging, resulting in shards with a higher variance in the draws having a smaller weight. Scott et al. (2016) prove that information-based averaging is optimal for Gaussian posteriors, but they were not yet able to prove this for non-Gaussian posteriors.

Algorithm 1 The Consensus MCMC algorithm

1: Split the data $\boldsymbol{y}$ into shards $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_S$
2: Run an MCMC simulation on each shard, resulting in samples $\boldsymbol{\theta}_{sg} \sim p(\boldsymbol{\theta} \mid \boldsymbol{y}_s)$, where g is the index of the draws
3: Combine the draws using weighted averages: $\boldsymbol{\theta}_g = (\sum_s \boldsymbol{W}_s)^{-1}(\sum_s \boldsymbol{W}_s \boldsymbol{\theta}_{sg})$

2.4. The Considered Models

The first considered model is labeled model A: Small Sample Binomial GLM. We use a Binomial GLM with a logistic link function to link the covariates to the conversion. The data used to estimate the parameters of this model is a subsample of the full dataset. The second model is labeled model B: Small Sample Bayesian Model. This is a complete Bayesian specification for modeling binomial data. Model B uses the same subsample of the full dataset as model A. The third model is labeled model C: Distributed Bayesian Model. This model uses the same specifications as model B. The main difference lies in the number of observations used for parameter estimation; to fit this model, the full dataset is used. Model C is able to use the full dataset due to its distributed characteristic,

whereas models A and B only use a small subsample of the full dataset. Figure 1 shows a simple visual representation of the functional forms.

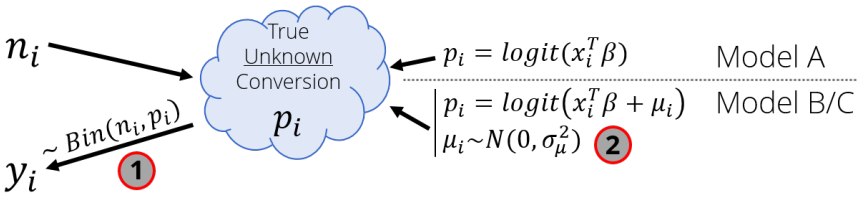

Figure 1: A graphical representation of the functional forms of model A, model B, and model C, with labels 1 and 2 indicating the locations that allow for variance.

We define the "true conversion" as the probability that a customer who visits a product page also buys that product. Focusing on the left side of Figure 1, we have a known number of visits n_i and sales y_i. The number of sales y_i is a random function of the number of visits n_i and the "true conversion" p_i. When n_i becomes extremely large, the value of y_i/n_i approaches p_i. However, when only two visits and one sale are observed, we are almost sure that the "true conversion" is different than $1/2 = 0.5$. This is a type of variation that is captured by all three models through the Binomial distribution (illustrated at label 1 in Figure 1). Effectively, using a Binomial distribution ensures that in the estimation process, observations with a large n_i are given a larger weight than observations with a small n_i.

The other side of Figure 1 shows the difference between model A and models B and C. Model A states that we can perfectly determine what the "true conversion" p_i is, and that all variance is caused by the Binomial distribution at label 1. However, models B and C have a random coefficient μ_i in the function to predict the value of p_i at label 2. Consequently, these models allow the "true conversion" p_i to differ from the predicted value. Adding the random coefficient to the equation is a subtle addition. Nevertheless, it can have a large influence on the final parameter estimates. For example, in the case of a temporarily popular product with a high n_i and y_i, this product is not behaving according to the true underlying model (this is often the case with temporary shocks). The variation we allow through the Binomial distribution at label 1 is small since we have such a high n_i and y_i. Consequently, model A will adjust the parameters in such a way that the outcome for these observations is close to the observed outcome. Models B and C on the other hand will probably not be influenced that much since they also allow variation within their estimation of p_i.

3. Data

In this section, we describe the data used in our experiments. First, in Section 3.1, we present the product page characteristics. Then, in Section 3.2, we discuss the data characteristics used in our models.

3.1. Product Page Characteristics and Hypothesis

Overall, three sets of information are available. These are product, offer, and seller characteristics. Table 2 gives a brief overview. We discuss the most important characteristics.

Table 2: An overview of product page characteristics with examples.

Characteristic	Type	Example
Average Product Review Score	Product	4 stars
Number of Product Reviews	Product	85 reviews
Content Enrichment Score	Product	0, 1, 2, or 3
Price	Offer	54.99 euro
Delivery Time	Offer	1 day
Ultimate Order Time	Offer	23:59
Last Mile Proposition	Offer	"SELECT"
Name	Seller	"famoshop.nl"
Average Seller Review Score	Seller	8.1
Number of Seller Reviews	Seller	366

Product characteristics do not change when another seller gets the Best Offer on a product. When someone buys the product, they are asked to rate their satisfaction with the product. They can give a score ranging from zero to five stars, with five stars being extremely satisfied and zero stars being not satisfied. Averaging all ratings for a product results in the Average Product Review Score, which is the first product characteristic.

The second product characteristic is the Number of Product Reviews. One might expect that a higher Average Product Review Score results in a higher conversion because people are more likely to buy a product that others are happy with. It could be expected that the Number of Product Reviews has a positive effect on the conversion since this number is often seen as an indication of the reliability of the Average Product Review Score (Chevalier & Mayzlin, 2006).

The third product characteristic is the Content Enrichment Score. This value is either 0, 1, 2, or 3. It is an indication of the quality of the product description on the product page. This score is not directly visible to the customers and is based on the number of product details filled in on the product page. For different sets of products, there exists a list of product details that are relevant

for that set of products. For example, for a router, a list of Internet speeds, number of ports, and other functionality is relevant. When the Content Enrichment Score is three, it means that for all of these specifications, values are shown on the product page. A lower score means that not all product specifications are known and shown. The hypothesis is that a higher score results in a higher conversion because when customers compare product pages of multiple websites, they will probably stick with the most informative one. It can also be the case that when the score is low and consequently not all product information is shown, customers are not sure whether it is the product they are looking for.

The next set of characteristics is based on the offer. First, the price relative to the market is important (here, we rank it by price stars; more is better). A five-star price means the price is the best in the market. A four-star price conforms to the market. A three-star price is around the average market price, and a two-star price is more expensive relative to the market. Finally, a one-star price is even worse and is not shown on the website since this harms the overall price perception of customers. The hypothesis is that a better price, relative to competitors, results in a higher conversion since people browsing the Internet and comparing prices of different e-commerce sites will probably buy the product at the website with the best price. This is also supported in literature (Greenberg, 2012; Maslowska et al., 2017), as high price sensitivity is often observed at online shops. Another price effect is the absolute size of the price. It could be argued that expensive products have a lower conversion, as impulse buying will probably happen more often on cheap products instead of expensive products. Also, when buying expensive products, customers will probably compare products more often.

The second offer characteristic is the Delivery Time, which indicates the number of days it takes to deliver the product. The hypothesis is that a lower delivery time results in higher conversion. When an offer has a long delivery time, people will probably delay their purchase because adding a day to an already long delivery time does not matter that much to them. This is in contrast with products that get delivered the next day. This hypothesis is supported by Lee (2002), who states that customers find fast delivery times important.

Adding to the Delivery Time is the Ultimate Order Time. When the delivery time is one day, it has an associated ultimate order time. The product has to be ordered before this time to be delivered the next day. The hypothesis is that a later ultimate order time results in a higher conversion for the same reasons as the characteristic Delivery Time.

An offer can have “SELECT” or “NONE” as Last Mile Proposition. When the Last Mile

Proposition is "SELECT", the customer gets to choose the delivery time and has additional options on where they want the product to be delivered. This gives the customer more freedom on how, where, and when the product is delivered. The hypothesis is that this results in a higher conversion since people, for example, do not have to stay at home to wait for their package.

The final set of characteristics is that of the seller. The first one is the Name. Although the specific name of a seller is not necessarily of interest, it matters whether the seller is a third party or the online shop itself. When the product is sold by the online shop itself, one might expect the conversion rate to be higher. From a previous analysis by the company, they know that conversion rates are higher when the seller's name is not shown, which could mean that customers have an aversion towards external sellers.

The last seller characteristics are the Average Seller Review Score and Number of Seller Reviews. Just like the Average Product Review Score and Number of Product Reviews, people can also rate their satisfaction with seller communication and product delivery. For the Average Seller Review Score, a customer rates the seller on a zero to ten scale, with ten meaning they are extremely satisfied with the seller. Again, it is hypothesized that a higher Average Seller Review Score and a higher Number of Seller Reviews result in a higher conversion for the same reason as the product scores.

3.2. Data Characteristics

The dataset consists of 210 days with an average of 70,587 unique products viewed per day. In the dataset, there exist 14,823,400 unique combinations of date and product ID. This means that on any given day, on average, 15,065,999/14,823,400 = 1.016 sellers have had the Best Offer for a product. Effectively, this means that in approximately 1.6% of the date and product ID combinations, multiple sellers had the Best Offer on the same day for a product. The average conversion over the whole dataset ($\Sigma_i y_i / \Sigma_i n_i$) is 6.22%. The average conversion when the online shop itself is the seller is 7.80%. In contrast, the average conversion for an external partner is only 4.42%. Table 3 shows an overview of the conversion dataset.

Table 3: An overview of the conversion dataset.

Variable name	Description	Set
y_sales	Number of sales	{0, 1, 2, ...}
n_views	Number of views	{1, 2, 3, ...}
category_10001	If product is in category_10001	{0, 1}
category_10002	If product is in category_10002	{0, 1}
category_10003	If product is in category_10003	{0, 1}
category_...	If product is in category_...	{0, 1}
average_product_review_score	average_product_review_score	[0, 5]
log_num_of_product_reviews	log(1 + num_of_product_reviews)	[0, ∞)
zero_product_reviews	If num_of_product_reviews is zero	{0, 1}
enrich23	If content_enrichment_score is 2 or 3	{0, 1}
classTraffic	If product_classification is Traffic	{0, 1}
classCore	If product_classification is Core	{0, 1}
logprice	log(price), minimum price is 1 cent	(-4.60517, ∞)
pricestar3	If price_star is 3	{0, 1}
pricestar4	If price_star is 4	{0, 1}
pricestar5	If price_star is 5	{0, 1}
dt1uot23t00	If delivery_time is 1 and ultimate_order_time is 23:00 or 00:00	{0, 1}
dt1uot20t22	If delivery_time is 1 and ultimate_order_time is 20:00 or 22:00	{0, 1}
dt1uot16t19	If delivery_time is 1 and ultimate_order_time is 16:00 or 19:00	{0, 1}
dt1uot12t15	If delivery_time is 1 and ultimate_order_time is 12:00 or 15:00	{0, 1}
dt23	If delivery_time is 2 or 3	{0, 1}
lastmileselect	If last_mile_proposition is "SELECT"	{0, 1}
seller_webshop	If seller is the online shop itself	{0, 1}
average_seller_review_score	average_seller_review_score	[0, 10]
log_num_of_seller_reviews	log(1 + num_of_seller_reviews)	[0, ∞)
zero_seller_reviews	If num_of_seller_reviews is 0	{0, 1}

The category_id row indicates to which group the product belongs. In total, we have 242 different categories in the conversion dataset. For each category, a variable is included which has a value of one if the product falls in the corresponding category, such that we can capture category-specific popularity (which cannot be captured by other covariates). We include both the Average Review Scores as is and take the logarithm of the Number of Seller Reviews plus one. An increase of 10 to 11 reviews is then valued the same as an increase of 1000 to 1100 and not valued 100 times as much. In addition, we add a variable that has the value of one if no reviews exist because we expect that this has an additional effect on the conversion rate.

Experts in the company indicated that the difference between a Content Enrichment Score of two and three is not always distinguishable. This has to do with the different criteria per category for obtaining a Content Enrichment Score of two or three. Therefore, we add the binary variable

enrich23, which has a value of one if the Content Enrichment Score is two or three, and zero otherwise. Effectively, a value of one for enrich23 means that the content on the product page is good.

We add the variables classTraffic and classCore to allow for variation within a category with respect to product popularity. The absolute size of the price is taken into account through the logarithm of the price. For the pricestar variable, we add a variable for three, four, and five-star prices. Note that we do not assume an order for price. Theoretically, a four-star price could therefore have a higher conversion than a five-star price.

The Delivery Time and Ultimate Order Time are combined into multiple dummy variables because there only exists an ultimate order time when the delivery time is one day. Just like the pricestar variables, we do not assume an order in the values of the variables. This could mean that a delivery time of four days has a higher conversion than a three-day delivery time. Lastly, there is a variable that has value one if the Last Mile Proposition is "SELECT" and a variable that has value one if the online shop itself is the seller.

In summary, we have 287 variables to represent categories and 20 characteristic variables, resulting in a total of $287 + 20 = 307$ variables in the dataset.

3.3. Sample Selection

We have data on the first seven months of 2017, which is shown in Figure 2. We use the first six months as our in-sample dataset for model C: Distributed Bayesian Model, resulting in 12.7 million observations for training purposes. In addition, we take a random sample of this in-sample dataset, which we call our small in-sample dataset, consisting of only 127,000 observations; it is used to fit models A and B. For models A and B, we need to work with a smaller sample as computations are done on one machine, which limits the size of the data that can be processed. Lastly, we use the out-of-sample data, the month of July, to assess the out-of-sample performance of all three models.

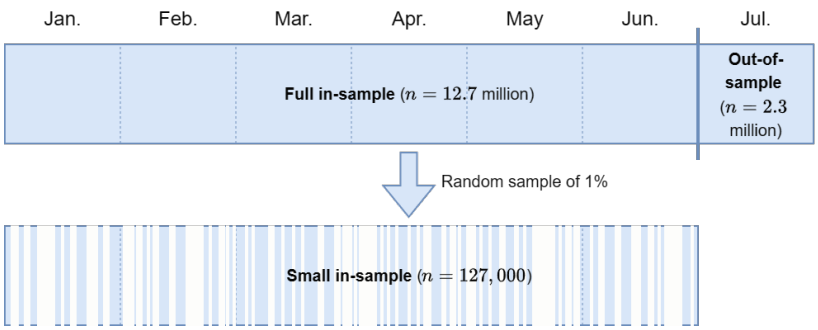

Figure 2: The data is split into an in-sample and out-of-sample dataset. Furthermore, a random sample is taken from the in-sample dataset for models A and B.

For model C, we need to split our in-sample dataset into multiple shards. The in-sample dataset is split into 250 shards, each containing approximately 12.7 million/250 = 50,800 observations. From experience, we know that using approximately 50,000 observations for MCMC simulations results in a smooth run.

4. Methodology

In this section, we describe the used methods. First, in Section 4.1, we present the data preparation. Then, Sections 4.2, 4.3, and 4.4 describe the three considered models. Next, in Section 4.5, we discuss the interpretation of the considered models, and follow with the evaluation measures in Section 4.6.

4.1. Data Preparation

For the remainder of this paper, the following variable specifications are used: $\boldsymbol{y}$ refers to the vector of sales, $\boldsymbol{n}$ to the vector of views, and $\boldsymbol{X}$ to the matrix of characteristics in numerical format. Note that the matrix $\boldsymbol{X}$ has no intercept because every category has its dummy variable, collectively making up the intercept. An element of $\boldsymbol{y}$ and $\boldsymbol{n}$, and a row of $\boldsymbol{X}$ are identified by the index $i = 1, \ldots, I$ such that we have values y_i, n_i, and a vector $\boldsymbol{x}_i$ corresponding to the ith observation.

4.2. Model A: Small Sample Binomial GLM

In model A, the relations stated in the following equations are modeled:

$$y_i \mid p_i \sim Bin(n_i, p_i), \tag{7}$$

$$p_i = \text{logit}(\theta_i) = F(\theta_i) = \frac{1}{1 + \exp(-\theta_i)}, \tag{8}$$

$$\theta_i = \boldsymbol{x}_i^T \boldsymbol{\beta}. \tag{9}$$

The model uses a GLM with a logistic link function and binomially distributed errors. The parameters are estimated using Iteratively Reweighted Least Squares (Holland & Welsch, 1977). The small in-sample dataset is used to fit the model, such that we can run it on a local machine using R.

4.3. Model B: Small Sample Bayesian Model

Next to model A, we consider a Bayesian approach as it is easily ammendable to parallelism, which is useful when working with large amounts of data like we do in this research. This model uses similar specifications to model A. Differently from model A, model B considers the random effects associated with observation i, which makes the model more flexible. The main difference is therefore the addition of a random coefficient μ_i, which results in the following equations:

$$y_i \mid p_i \sim Bin(n_i, p_i), \tag{10}$$

$$p_i = \text{logit}(\theta_i) = F(\theta_i) = \frac{1}{1 + \exp(-\theta_i)}, \tag{11}$$

$$\theta_i = \boldsymbol{x}_i^T \boldsymbol{\beta} + \mu_i, \tag{12}$$

$$\mu_i \sim \text{N}(0, r_\mu^{-1}). \tag{13}$$

In order to obtain parameter estimates, an MCMC simulation is used with a Gibbs sampler. Therefore, we need the conditional distributions $p(\theta_i \mid \boldsymbol{\beta}, r_\mu, \boldsymbol{y})$, $p(\boldsymbol{\beta} \mid \boldsymbol{\theta}, r_\mu, \boldsymbol{y})$, and $p(r_\mu \mid \boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{y})$. To get these conditional distributions, all elements can be removed which do not depend on the parameter of interest. The resulting equation is a kernel uniquely identifying a distribution, which can be a known or unknown distribution:

$$p(\theta_i \mid \boldsymbol{\beta}, r_\mu, \boldsymbol{y}) \propto F(\theta_i)^{y_i} \left(1 - F(\theta_i)\right)^{n_i - y_i} \exp\left[-\frac{r_\mu}{2} \left(\theta_i - \boldsymbol{x}_i^T \boldsymbol{\beta}\right)^2\right], \tag{14}$$

$$p(\boldsymbol{\beta} \mid \boldsymbol{\theta}, r_\mu, \boldsymbol{y}) = m.v.Normal\left(\boldsymbol{\beta};\, location = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\theta},\, covariance = r_\mu^{-1}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\right), \tag{15}$$

$$p(r_\mu \mid \boldsymbol{\beta},\, \boldsymbol{\theta},\, \boldsymbol{y}) = Gamma\left(r_\mu;\, rate = \frac{1}{2}\left[2a + \sum_{i=1}^{N} (\theta_i - \mathbf{x}_i^T\boldsymbol{\beta})^2\right],\, shape = \frac{1}{2}(2b + N)\right). \tag{16}$$

We can easily draw from the conditional distributions $p(\boldsymbol{\beta} \mid \boldsymbol{\theta}, r_\mu, \boldsymbol{y})$ and $p(r_\mu \mid \boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{y})$ since these are of a known form (Normal and Gamma). For the conditional distribution $p(\theta_i \mid \boldsymbol{\beta}, r_\mu, \boldsymbol{y})$, we

use the Metropolis-Hastings algorithm with the candidate function $g(\theta_i \mid \theta_i^{(m-1)}) = N(\theta_i^{(m-1)}, h^2)$. These conditional distributions are used to do an MCMC simulation. For each parameter, a draw from the conditional distribution of that parameter given the current values of the other parameters is saved. Combining all draws over all iterations yields the joint distribution of the parameters. This is an iterative process and each iteration depends on the previous iteration. Algorithm 2 shows an overview.

Algorithm 2 MCMC Simulation Schema

Set starting values for $\boldsymbol{\theta}^{(0)}$ and $r_\mu^{(0)}$
for m in $1, \ldots, M$ **do**
    Draw $\boldsymbol{\beta}^{(m)}$ from $p(\boldsymbol{\beta} \mid \boldsymbol{\theta}^{(m-1)}, r_\mu^{(m-1)}, \boldsymbol{y})$
    Draw $r_\mu^{(m)}$ from $p(r_\mu \mid \boldsymbol{\theta}^{(m-1)}, \boldsymbol{\beta}^{(m)}, \boldsymbol{y})$
    for i in $1, \ldots, N$ **do**
        Draw candidate θ_i^* from $g(\theta_i \mid \theta_i^{(m-1)}) = N(\theta_i^{(m-1)}, h^2)$
        Calculate $\alpha = \min(\frac{f(\theta_i^*)\, g(\theta_i^{(m-1)} \mid \theta_i^*)}{f(\theta_i^{(m-1)})\, g(\theta_i^* \mid \theta_i^{(m-1)})}, 1)$
        Set $\theta_i^{(m)} = \theta_i^*$ with probability α
        Set $\theta_i^{(m)} = \theta_i^{(m-1)}$ with probability $1 - \alpha$
    end for
end for

The total number of iterations M needed for convergence can differ but is often around $10,000$. By changing h^2 (variance), the (average) acceptance probability of the Metropolis-Hastings algorithm and consequently the convergence speed can be influenced. A high acceptance probability is as bad as a low acceptance probability; therefore, we aim for an acceptance probability of around 0.5. Through trial and error, we set $h^2 = 2$ such that the average acceptance probability α lies around 0.6. Also, the hyperparameters a and b have to be defined beforehand. Kim & Kim (2000) propose to use $a = 0.001$ and $b = 0.001$, which translates to having basically no prior belief on what the value of r_μ should be.

The starting values $\boldsymbol{\theta}^{(0)}$ and $r_\mu^{(0)}$ also have to be defined. In theory, we could pick random values. However, to increase convergence speed, we pick values we believe are close to the true values. We pick $r_\mu^{(0)} = 1$, which is the expected value of the prior of r_μ. For $\boldsymbol{\theta}^{(0)}$, we pick $\theta_i^{(0)} = -\log[1/((y_i + 0.1)/(n_i + 0.2)) - 1]$, which is the inverse logistic function for input y_i/n_i. We add 0.1 and 0.2 to y_i and n_i respectively, because the inverse logistic function maps to $-\infty$ or $+\infty$ with input 0 or 1, and the R programming language cannot handle this properly. The choice for 0.1 and 0.2 is arbitrary and any small value could be used as long as the numerator is larger than

the denominator.

The last parameter we have to define is the size of the burn-in sample. The burn-in sample comprises the first couple of draws in the MCMC simulation that are removed as these values have probably not converged yet. We pick a value of 0.25, meaning that the first 25% (e.g., 2,500 if M = 10,000) of the draws are removed.

4.4. Model C: Distributed Bayesian Model

The previously described process is also used to run the distributed MCMC simulation for model C. In addition, we use the same hyperparameters as described in the previous section. As explained by Scott et al. (2016), the Consensus MCMC algorithm distributes the data across multiple workers, runs the MCMC simulation on all shards, and then combines the results.

Each observation in the data is assigned a random number between 1 and 250. Then, the rows with the same random number represent a shard. The simulation in Algorithm 2 is executed on each part of the data (shard), which is done using Spark and the SparkR package. This allows us to group the data on the random number and then apply a custom function (Algorithm 2) to each group. Each simulation returns a list of draws for each parameter. These parameters are then combined according to step three in Algorithm 1 in Section 2.3. This can be done by simple averaging. However, one could also weight the draws of the parameters of each shard by the inverse of the covariance matrix of the draws from that shard. This is a way of information-based weighting, resulting in shards with a lot of uncertainty in the draws having a smaller weight than shards of which the draws have a low variance. We compare both methods of combining the draws, simple (equal-weighted) averaging and information-based averaging.

According to Scott et al. (2016), the prior has to be adjusted before running the simulation. Given the complete dataset, a prior has to be defined, one that is often dependent on the number of observations. Then, when one splits the data and runs separate MCMC simulations, this prior has to be adjusted to the new size of the dataset. In other words, the prior belief on the distribution of the parameters depends on the size of the dataset that is used. In this research, we do not change the priors for the parameters since they are uninformative. This implies that an adjustment of these priors will still result in uninformative priors.

4.5. Interpretation

For model A, we obtain parameter estimates $\widehat{\boldsymbol{\beta}}^{(A)}$ together with their corresponding p-values, which show the significance of the parameter estimates. For models B and C, we obtain a set of draws

from the posterior distribution. The combinations of the draws of $\boldsymbol{\beta}^{(m)}, r_\mu^{(m)}$, and $\theta_i^{(m)}$ make up the joint distribution $p(\boldsymbol{\theta}, \boldsymbol{\beta}, r_\mu \mid y)$. For both models, we summarize these draws to obtain the mean of the draws, from now on referred to as $\widehat{\boldsymbol{\beta}}^{(B/C)}$ and $\widehat{r_\mu}^{(B/C)}$. The most obvious way to summarize the draws is by taking the average. We use the 95% Highest Posterior Density (HPD) as a second measure to summarize the distribution of the parameters. The 95% HPD is the smallest possible interval containing 95% of the draws. Although we do not have a measure for significance for Bayesian models, we can check whether the value zero falls into the 95% HPD to get an idea of how reliable the estimated values $\widehat{\boldsymbol{\beta}}^{(B/C)}$ are.

With respect to the interpretation of the results, the three models are similar. All three of them link the probability of conversion p_i to the covariates $\boldsymbol{x}_i$ through a logistic link function. For model A, with estimated parameters $\widehat{\boldsymbol{\beta}}^{(A)}$, the expected value of p_i is:

$$E\left[p_i \mid \boldsymbol{x}_i, \widehat{\boldsymbol{\beta}}^{(A)}\right] = F\left(\boldsymbol{x}_i^T \widehat{\boldsymbol{\beta}}^{(A)}\right) = \frac{1}{1+\exp\left(-\boldsymbol{x}_i^T \widehat{\boldsymbol{\beta}}^{(A)}\right)}. \tag{17}$$

The only difference between model A and models B and C is that models B and C also include a random coefficient μ_i. Unfortunately, the expected value of p_i becomes more cumbersome:

$$\begin{aligned} E\left[p_i \mid \boldsymbol{x}_i, \widehat{\boldsymbol{\beta}}^{(B/C)}, \widehat{r_\mu}\right] &= E\left[F\left(\boldsymbol{x}_i^T \widehat{\boldsymbol{\beta}}^{(B/C)} + \mu_i\right) \mid \boldsymbol{x}_i, \widehat{\boldsymbol{\beta}}^{(B/C)}, \widehat{r_\mu}\right], \\ &= E\left[\frac{1}{1+\exp\left(-(\boldsymbol{x}_i^T \widehat{\boldsymbol{\beta}}^{(B/C)} + \mu_i)\right)} \mid \boldsymbol{x}_i, \widehat{\boldsymbol{\beta}}^{(B/C)}, \widehat{r_\mu}\right], \end{aligned} \tag{18}$$

where μ_i are i.i.d. $N(0, \widehat{r_\mu}^{-1})$. We can however do a simple simulation to obtain the expected value by simulating 1,000 draws for μ_i, calculating the corresponding value p_i for each draw of μ_i, and then averaging all values of p_i.

To give a meaningful interpretation to the estimated values of $\boldsymbol{\beta}$, the odds ratio is used. For model A, the odds ratio is defined as follows:

$$E\left[\frac{p_i}{1-p_i} \mid \boldsymbol{x}_i, \widehat{\boldsymbol{\beta}}^{(A)}\right] = \exp\left(\boldsymbol{x}_i^T \widehat{\boldsymbol{\beta}}^{(A)}\right). \tag{19}$$

The odds ratio for models B and C is defined as follows:

$$E\left[\frac{p_i}{1-p_i} \mid \boldsymbol{x}_i, \widehat{\boldsymbol{\beta}}^{(B/C)}, \widehat{r_\mu}\right] = E\left[\exp\left(\boldsymbol{x}_i^T \widehat{\boldsymbol{\beta}}^{(B/C)} + \mu_i\right) \mid \boldsymbol{x}_i, \widehat{\boldsymbol{\beta}}^{(B/C)}, \widehat{r_\mu}\right]. \tag{20}$$

In order to analyze the marginal effects of the variables $\boldsymbol{x}_i$ on the expected value of the odds ratio $p_i/(1-p_i)$, we take the derivative of the expected value of $p_i/(1-p_i)$ with respect to $\boldsymbol{x}_i$. This results in the following equation for model A:

$$\frac{\partial\, E\left[\frac{p_i}{1-p_i} \mid \boldsymbol{x}_i, \widehat{\boldsymbol{\beta}}^{(A)}\right]}{\partial\, \boldsymbol{x}_i} = E\left[\frac{p_i}{1-p_i} \mid \boldsymbol{x}_i, \widehat{\boldsymbol{\beta}}^{(A)}\right] \cdot \widehat{\boldsymbol{\beta}}^{(A)}, \tag{21}$$

and for models B and C:

$$\frac{\partial\, E\left[\frac{p_i}{1-p_i} \mid \boldsymbol{x}_i, \widehat{\boldsymbol{\beta}}^{(B/C)}, \widehat{r_\mu}\right]}{\partial\, \boldsymbol{x}_i} = E\left[\frac{p_i}{1-p_i} \mid \boldsymbol{x}_i, \widehat{\boldsymbol{\beta}}^{(B/C)}, \widehat{r_\mu}\right] \cdot \widehat{\boldsymbol{\beta}}^{(B/C)}. \tag{22}$$

Although the formulas for the marginal effects seem difficult, they have a nice interpretation. Simply put, if x_{i1} increases by one unit, the odds ratio increases by $\beta_1 \cdot 100\%$. Consequently, this means that the marginal effects are dependent on the current odds ratio. Thus, we are able to compare the parameter estimates between the three models by using the odds ratio.

Figure 3 shows the relation between the value p and the odds ratio $p/(1-p)$ for values of p between 0 and 0.2. For low values of p, the relation is almost linear. So, for low values of p, we can use the rule of thumb that the odds ratio is equal to the value p. For values of p higher than 0.2, this rule should not be used anymore. This relation makes the interpretation of the parameters somewhat easier. For example, when we have a value of $\beta_1 = 0.75$, this means that, on average, the odds ratio is 75% higher if x_1 increases by one unit. Assuming the current value of p is 0.05, then the odds ratio is $0.05/0.95 = 0.053$, which would increase by 75%. So, the new odds ratio would become $1.75 \cdot 0.53 = 0.092$, which corresponds to a value of $p = 0.092/(1 + 0.092) = 0.084$. This is $0.084/0.05 = 1.69$ times higher than before, so the value of p increased by 69%, which lies close to the 75% increase in odds ratio.

Note that due to the random coefficient μ_i in models B and C, the previous example does not hold exactly for these models. Therefore, we interpret the results only for the odds ratio as this is theoretically correct. The previous example can help the reader in translating these effects on the odds ratio to approximate effects on the conversion p.

4.6. Model Comparison

For model comparison, we take into account three aspects: (1) the calculation time and difficulty of obtaining parameter estimates, (2) the out-of-sample performance, and (3) the differences in

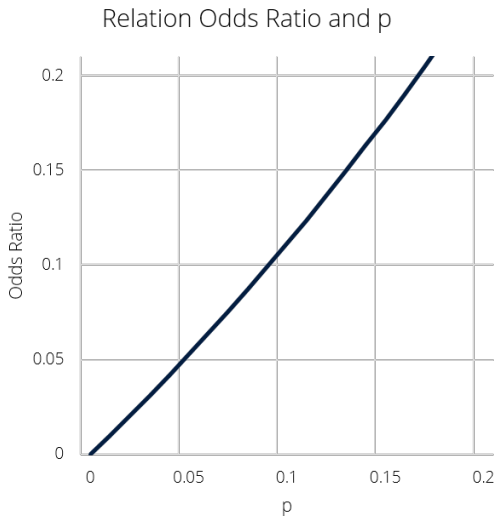

Figure 3: The relation between p and the odds ratio for p between 0 and 0.20.

qualitative interpretation. In this section, the method for obtaining the out-of-sample performance is discussed.

The out-of-sample dataset is used to make predictions. We predict the number of converted customers y for each observation in the out-of-sample dataset. Given the out-of-sample dataset $\{y_j, n_j, \boldsymbol{x}_j\}$ for $j = 1, \ldots, J$, the prediction for y_j is made as follows:

$$\widehat{y_j}^{(q)} = E\left[p_j \mid \boldsymbol{x}_j, \widehat{\boldsymbol{\beta}}^{(q)}\left(, \widehat{r_\mu}^{(q)}\right)\right] \cdot n_j, \tag{23}$$

where $\widehat{\boldsymbol{\beta}}^{(q)}$ indicates the parameter estimates of model $q = \{A, B, C\}$. $\boldsymbol{x}_j$ are the covariates for observation j in the out-of-sample dataset and n_j denotes the number of views of observation j.

We then compare these estimates $\widehat{y_j}^{(q)}$ with the true value y_j. Assuming that the out-of-sample dataset has J observations, a measure for the out-of-sample predictive performance is the Root Mean Squared Error (RMSE):

$$\text{RMSE}^{(q)} = \sqrt{\frac{1}{J}\sum_{j=1}^{J}\left(\widehat{y_j}^{(q)} - y_j\right)^2}. \tag{24}$$

We compare the performance between the three models but also compare it to a fourth estimation method, we call this the Control Model. This simple method multiplies the number of visits n with the average conversion over all products (6.22%). All three models should at least outperform

this method.

5. Results

In this section, we analyze the obtained results. For each model, we discuss the process of estimation, provide an overview of the parameter estimates, and go over the out-of-sample performance. We also provide a qualitative interpretation of said estimates.

5.1. Parameter Estimation and Predictions

For comparison, we have five models: model A, model B, model C with information-based averaging, model C with simple averaging, and our Control Model. The parameter estimates of the two models C come from the same set of draws but differ in the way these draws are combined. Our Control Model has no parameter estimates but does have an RMSE. Table 4 shows an overview of the parameter estimates and RMSE for each model. Each category has its intercept and we summarize all intercepts by taking the average. The RMSE is stated at the bottom of the table together with the RMSE relative to the Control Model. In addition, the average acceptance probability α for the random walk sampler (Metropolis-Hastings) is shown for the three Bayesian models.

For model A, using a dataset of 127,000 observations, the estimation took approximately 10 minutes. As for the p-values, two parameters stand out: zero_product_reviews and average_seller_review_score. Both have a p-value larger than 0.01, which means that these variables do not significantly differ from zero at a 1% significance level.

We used the same dataset as model A to estimate the parameters of model B. With the hyperparameters described in Section 4.3, the estimation using 10,000 iterations took 1.5 hours. This is fairly long compared to the estimation time of model A. A detailed overview of the parameter estimates is shown in Table A.6 in Appendix A. We observe that the variable zero_product_reviews stands out because the value zero lies inside the 95% HPD, together with zero_seller_reviews, lastmileselect, and pricestar3.

When using MCMC simulation, it is important to check whether the algorithm has converged. To be precise, we are looking for convergence in the distribution of the parameters. Methods for assessing convergence have been discussed a lot in the literature (Sinharay, 2003). We look at three criteria: the difference between the estimates of model A and model B, the correlation between the realized draws of the parameters, and the chain of realized draws of the parameters.

Table 4: Summary of parameter estimates and RMSE for the five models: model A, model B, model C with information-based averaging, model C with simple averaging, and the Control Model.

Variable	A	B	C (info.)	C (simple.)	Control
pricestar3	0.08	0.038	0.018	0.017	
pricestar4	0.112	0.085	0.065	0.065	
pricestar5	0.055	0.116	0.106	0.110	
logprice	-0.596	-0.695	-0.689	-0.688	
lastmileselect	0.152	0.070	0.032	0.033	
dt1uot12t15	0.438	0.409	0.375	0.375	
dt1uot16t19	0.378	0.418	0.398	0.397	
dt1uot20t22	0.359	0.400	0.415	0.416	
dt1uot23t00	0.284	0.351	0.363	0.363	
dt23	0.133	0.139	0.138	0.140	
enrich23	0.034	0.072	0.078	0.079	
average_product_review_score	0.236	0.240	0.242	0.242	
log_num_of_product_reviews	0.124	0.132	0.142	0.142	
zero_product_reviews	0.034	-0.044	-0.027	-0.028	
classCore	0.060	0.091	0.104	0.105	
classTraffic	0.221	0.348	0.335	0.336	
seller_webshop	1.235	0.933	0.893	0.994	
average_seller_review_score	0.014	0.092	0.086	0.087	
log_num_of_seller_reviews	0.021	0.022	0.026	0.026	
zero_seller_reviews	-0.808	-0.439	-0.373	-0.476	
Mean (category10xxx)	-2.920	-3.712	-3.708	-3.724	
r_μ		0.985	0.986	0.985	
Mean (acceptance probability α)		0.622	0.569	0.569	
RMSE	1.460	1.459	1.438	1.437	1.854
RMSE relative to Control Model	78.75%	78.69%	77.56%	77.51%	100.00%

We know that the parameter estimates of model A converged to the optimal values for that model since we used a GLM. Although we cannot directly compare model A to model B since the functional forms and estimation methods are different, we do expect similar parameter estimates since the functional forms do not differ significantly. We find that the signs of the parameter estimates for model B are the same as for model A, except for one parameter (zero_product_reviews). In addition, the relative sizes of the parameter estimates are similar.

Often, (highly) correlated draws of different parameters influence the speed of convergence. Figure 4 shows the correlation between the realized draws of the relevant parameters. We observe that most of the correlations occur between draws of parameters where variables are also correlated. For example, for the draws of the parameters for lastmileselect and dt1uot23t00, these variables are highly correlated since "SELECT" often implies a delivery time of one day with an ultimate order time of midnight. The correlation between these parameters is highly negative. We also observe a high correlation between parameters of dummy variables of the same variable. For example, the binary variables pricestar3, pricestar4, and pricestar5 are positively correlated because all three of

them have the same base case, namely a pricestar of one or two. This also applies to the delivery time. However, we do not observe large correlations between parameters for which we do not expect this based on the underlying variables, which indicates that the algorithm is more likely to converge.

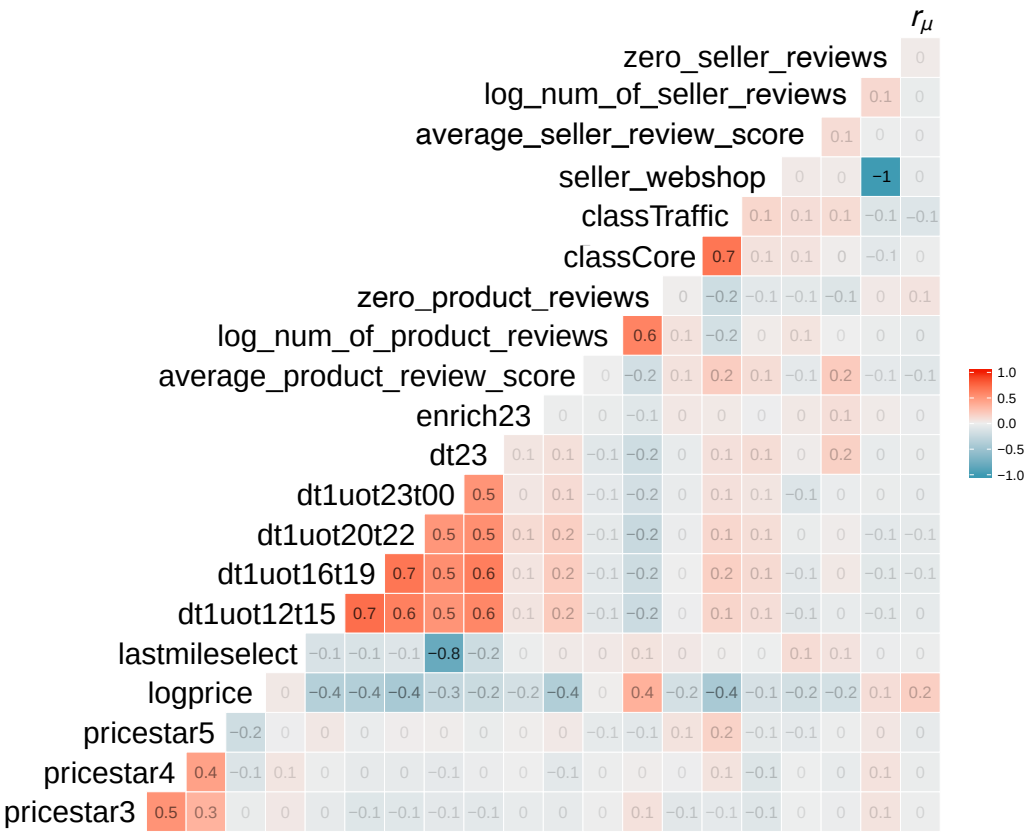

Figure 4: Correlation between realized draws of parameters for model B where red means a correlation of one and blue of minus one (dark grey in black and white printing).

Looking at the chain of realized draws of the parameters, the first few draws seem to move around a lot. However, often after 100 draws, it seems that the draws are more in line with the later draws, justifying the use of a burn-in sample. Taking all three criteria into account, we are fairly confident that model B has converged.

Model C uses the same functional form as model B but uses the full dataset and the Consensus MCMC algorithm with simple averaging and information-based averaging. As described in Section 3.3, each of the 250 shards contains 50,800 observations. Each run, with the same hyperparameters as model B, took approximately one hour. We used 16 workers, all running in parallel, so the estimation time was 250 shards$\cdot$1 hour/16 workers $\approx$ 16 hours. Comparing this to the other models

in Table 5, we observe that the estimation of model C is rather slow. However, a direct comparison is difficult, as model A and model B use much smaller datasets. In addition, as Table 5 shows, the estimation of model C is more difficult because the MCMC simulation has to be distributed over multiple workers and then combined into one set of parameter estimates, which is useful when working with such a large dataset. A detailed overview of the estimation result is shown in Table A.7 and Table A.8 in Appendix A for information-based and simple averaging, respectively. We find that none of the parameter estimates include the value zero in the 95% HPD.

Comparing the HPD intervals of model B and model C, we observe that the average width of the HPD intervals of model C is 8% of those of model B. This indicates that we are more certain about the estimation for model C, which could be expected as we use a larger training dataset for this model.

We use two methods of combining the draws of the 250 shards: information-based averaging and simple averaging. Table 4 shows that the parameter estimates for the two methods of combining are similar. This also applies to the HPDs in Tables A.7 and A.8 in Appendix A. Using the same criteria to assess convergence as for model B, we are fairly confident that both model C with information-based averaging and simple averaging have converged.

The RMSEs for the four (non-control) models are similar. All models outperform the Control Model by roughly 20%. Comparing the RMSEs for both models C, we find that the RMSE for information-based averaging (1.438) is slightly higher than that of simple averaging (1.437). The fact that information-based averaging is not better than simple averaging is at first sight counter-intuitive, but Scott et al. (2016) did warn that information-based averaging may not work optimally for non-Gaussian distributions (which we are working with). All Bayesian models beat model A. In addition, in line with our expectations, model C beats model B in terms of RMSE. This can be attributed to the larger training dataset.

Table 5: A summary of the estimation procedure for all considered models.

Model	Data size	Estimation time	Difficulty to estimate
A	127,000	10 minutes	easy
B	127,000	1.5 hours	moderate
C	12.7 million	16 hours	hard

5.2. Qualitative Interpretation

Next to out-of-sample performance, variable interpretation is important, as it can help companies increase their conversion rate. Figure 5 shows a visualization of the marginal effects of the variables on the odds ratio.

The first thing that stands out is that the sign of the effects conforms to expectations for most variables. Second, the difference between the three models is small. In the rest of this section, we use the parameter estimates of model C with information-based averaging in our qualitative interpretation, as model C has the best out-of-sample performance. We opt for information-based averaging because the values of the parameter estimates do not differ that much between information-based averaging and simple averaging. In addition, the difference between these models is minor and we choose model C with information averaging because it can be explained better in an intuitive way by weighting lower the parameters of shards with a lot of uncertainty instead of weighting all shards equally. When the value of a parameter does differ between models, we discuss this briefly.

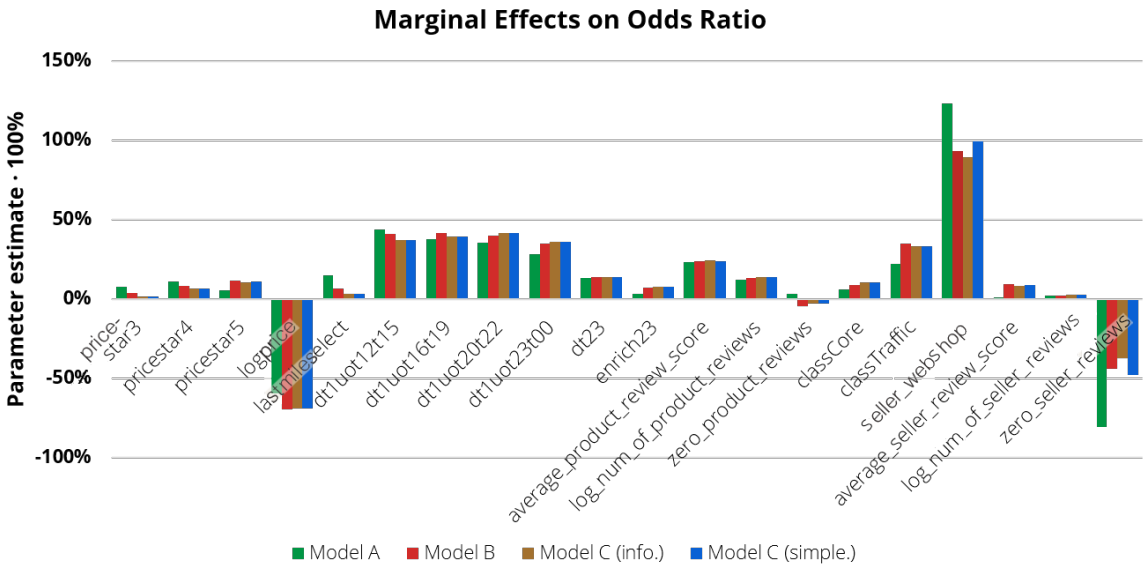

Figure 5: The marginal effects of the variables on the odds ratio for the four models.

When discussing "the effect" of a variable, we mean the average increase in odds ratio holding all other variables equal when we increase this variable by one unit. In the case of dummy variables, holding all other variables equal is not always possible, which is taken into account. An example of this is the pricestar variable, as it is impossible to have a pricestar of four and five at the same time.

Price. With respect to the pricestar variables, the effects are as expected. Having a pricestar of three, four, or five has a positive effect on the odds ratio compared to having a pricestar of one or two. In addition, the order is as expected, as having a pricestar of five is better than four, which is better than three. This is visualized in Figure 6. For example, the effect of increasing the pricestar from three to four is $0.065 - 0.018 = 0.047$ (using the parameter estimates of model C with information-based averaging in Table 4) since the value of the pricestar3 variable becomes zero and that of pricestar4 becomes one.

For model A, the effect of having a pricestar of three, four, or five is also positive but the order is not as we expected. Model A implies that, holding all other covariates equal, a pricestar of four has, on average, a higher odds ratio than a pricestar of five. This is unexpected, as a pricestar of five should indicate a more competitive price than a pricestar of four. This goes against the principles of traditional economics that say that a lower price results in higher sales.

Figure 6: Visualization of the effect of the variable pricestar on the odds ratio.

The absolute price has been taken into account through the variable logprice. This means that the marginal effect on the odds ratio is $\beta_{\text{logprice}}/\text{price}$ (using the chain rule for derivations). The effect is negative, meaning that higher absolute prices have, on average, a lower odds ratio. This is also in line with our expectations. For example, the difference between 10 euros and 100 euros is $-0.689 \cdot [\ln(100) - \ln(10)] = -1.59$, implying a 159% lower odds ratio for the 100 euro product. This is quite a lot, but this is also what is often observed in the industry; expensive products are compared much more than inexpensive products, resulting in a lower conversion.

Last Mile Proposition, Content Enrichment Score, and Product Classification. The effect of having the "SELECT" label (Last Mile Proposition) is that it increases the odds ratio by 3.2%. We expected this number to be higher, but the direction of the effect is as expected. By having a Content Enrichment Score of two or three relative to zero or one, the odds ratio is on average 7.8% higher. This is in line with our expectations. We observe that products with

Traffic classification relative to Tail classification have a 33.5% higher odds ratio. In addition, Core products have a 10.4% higher odds ratio relative to Tail products.

Delivery Time. The effects of delivery time and the ultimate order time have been visualized in Figure 7. We observe similar effects in all three models. A delivery time of two or three days has a 13.8% higher odds ratio than a delivery time of four days or more. The highest difference is found between two- or three-day delivery times and one-day delivery times with an increase of 23.7%. Once the delivery time is one day, the ultimate order time has not that much effect. Something that goes against our expectations is that an ultimate order time of midnight (00h) has, on average, a 5.2% lower odds ratio than that of a 22h ultimate order time. We also see this in the results of models A and B. A possible explanation for this is that because most people visit the website around 8 o'clock, an ultimate order time of 10 o'clock is more pressing than an ultimate order time of midnight, consequently resulting in a higher conversion. Another explanation is that it is caused by the fact that when there are too many orders on a day to handle, the ultimate order time is set back a couple of hours. This could mean that the ultimate order time is set back from midnight to 10 o'clock when there are a lot of orders, consequently resulting in a higher conversion.

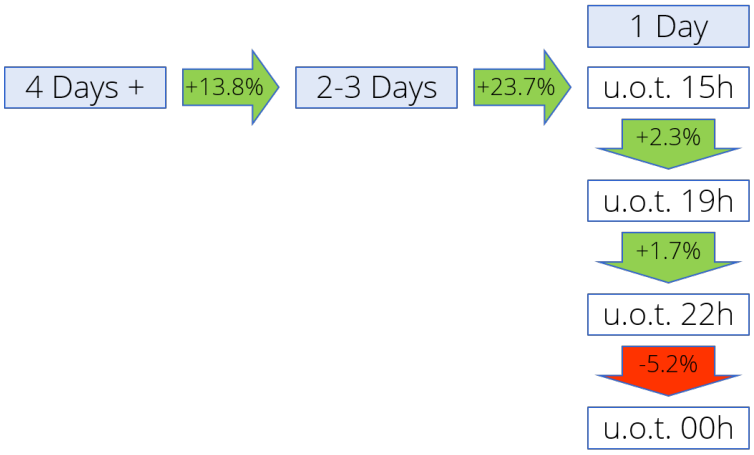

Figure 7: Visualization of the effect of delivery time on the odds ratio.

Product Reviews. With respect to the product reviews, we have three parameter estimates: average_product_review_score, log_num_of_product_reviews, and zero_product_reviews. We see that a higher number of product reviews and a higher average product review score have a positive effect on the odds ratio. Figure 8 shows the effect of different numbers of product reviews; note that a product can have an unlimited number of reviews. As the number of product reviews can be zero,

we decided to add one (1) to this number before taking the logarithm. For the average score of the product reviews, the effect is an increase in the odds ratio of 24.2% for an increase of one point on the one to five scale.

Figure 8: Visualization of the effect of number of product reviews on the odds ratio.

Seller and Seller Reviews. The odds ratio increases if the webshop itself is the seller, as we observe an increase of 89.3%. When the seller is one of the partners, the effects of the Seller Review Score and Number of Seller Reviews are positive. In Figure 9, the effect of the Number of Seller Reviews is visualized. We observe a large increase in the odds ratio when the Number of Seller Reviews increases from zero to one. This can be explained because the review score is only shown when a seller has at least one review score. The value of the Average Seller Review Score has an effect of 8.6% on the odds ratio. This means that, on average, the odds ratio is 8.6% higher when the Average Seller Review Score is one point higher on the one to ten scale.

Figure 9: Visualization of the effect of number of seller reviews on the odds ratio.

Temporary Shocks. We hypothesized that by including the random coefficient μ_i, models B and C would be able to capture temporary shocks relative to model A. Although it is difficult to prove this is happening, there are some indicators.

When a new product is launched, the conversion is generally higher than average for the first couple of days. Often these new products are offered by the webshop itself, they have the "SELECT" label, and they do not have any product reviews yet. We see in Table 4 that the estimates for these three variables (seller_webshop, lastmileselect, and zero_product_reviews) for model A are higher than for models B and C. Since model A is not able to properly capture these temporary shocks and when we assume that the temporary shocks often occur when these three variables apply, we expect that the parameter estimates of these three variables are higher than they actually are. This is exactly what we see in the parameter estimates. So, we suspect that models B and C capture

temporary shocks better than model A based on the difference in parameter estimates between model A and models B and C.

6. Conclusion

In this paper, we focused on the drivers of e-commerce conversion in a large e-commerce company in the Netherlands. We compared three models: a Small Sample Binomial GLM (model A), a Small Sample Bayesian Model (model B), and a Distributed Bayesian Model (model C). For model C, we used two different methods of summarizing the results, namely information-based averaging and simple averaging. The models differ with respect to the size of the used training data and the complexity of the functional form.

We found that Big Data adds value over using a small sample of this Big Data. In terms of predictive performance and qualitative interpretation, model C outperformed the other two considered models. In addition, we did find some indicators that the included random coefficient in models B and C captured temporary shocks. We formulated the following research question: *What are the drivers of product page conversion for e-commerce companies?* We observed that, amongst others, reviews (for sellers and products) are an important driver (both the average review score and the number of reviews). In addition, the conversion is higher when the webshop is the seller of a product compared to when a partner is the seller. All other effects were in general as expected.

For future work, we consider adding extra variables. In this case, we have included the most important variables in our opinion. A valuable addition could be to include interaction effects between the variables and the different categories, where the assumption would be that not all the effects are the same for all categories of products. In addition, adding time trend variables could be a valuable addition, possibly capturing seasonality. Last, future research could use the found driving factors of product page conversion for recommender systems, for example by using additional factors in a latent factor model approach (Wu et al., 2022; Luo et al., 2023).

References

Ain, N., Vaia, G., DeLone, W. H., & Waheed, M. (2019). Two decades of research on business intelligence system adoption, utilization and success – A systematic literature review. *Decision Support Systems*, *125*, 113113.

Buchholz, A., Ahfock, D., & Richardson, S. (2023). Distributed computation for marginal likelihood based model choice. *Bayesian Analysis*, *18*, 607–638.

Chen, H., Chiang, R. H., & Storey, V. C. (2012). Business intelligence and analytics: From big data to big impact. *MIS Quarterly*, *36*, 1165–1188.

Chevalier, J. A., & Mayzlin, D. (2006). The effect of word of mouth on sales: Online book reviews. *Journal of Marketing Research*, *43*, 345–354.

Chib, S., & Greenberg, E. (1995). Understanding the Metropolis-Hastings algorithm. *The American Statistician*, *49*, 327–335.

Dean, J., & Ghemawat, S. (2008). MapReduce: Simplified data processing on large clusters. *Communications of the ACM*, *51*, 107–113.

Dunn, P. K., & Smyth, G. K. (2018). *Generalized Linear Models with Examples in R* volume 53. Springer.

Greenberg, E. (2012). *Introduction to Bayesian Econometrics*. (2nd ed.). Cambridge University Press.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, *57*, 97–109.

Holland, P. W., & Welsch, R. E. (1977). Robust regression using iteratively reweighted least-squares. *Communications in Statistics - Theory and Methods*, *6*, 813–827.

Kim, D.-H., & Kim, E.-Y. (2000). Bayesian analysis for random effects binomial regression. *Communications for Statistical Applications and Methods*, *7*, 817–827.

Lee, P.-M. (2002). Behavioral model of online purchasers in e-commerce environment. *Electronic Commerce Research*, *2*, 75–85.

Lin, Q., Jia, N., Chen, L., Zhong, S., Yang, Y., & Gao, T. (2023). A two-stage prediction model based on behavior mining in livestream e-commerce. *Decision Support Systems*, *174*, 114013.

Luo, X., Zhou, Y., Liu, Z., & Zhou, M. (2023). Fast and accurate non-negative latent factor analysis of high-dimensional and sparse matrices in recommender systems. *IEEE Transactions on Knowledge and Data Engineering*, *35*, 3897–3911.

Marcellino, M. G., Papailias, F., Mazzi, G. L., Kapetanios, G., & Buono, D. (2018). Big data econometrics: Now casting and early estimates. *BAFFI CAREFIN Centre Research Paper*.

Maslowska, E., Malthouse, E. C., & Viswanathan, V. (2017). Do customer reviews drive purchase decisions? The moderating roles of review exposure and price. *Decision Support Systems*, *98*, 1–9.

McCullagh, P. (1984). Generalized linear models. *European Journal of Operational Research*, *16*, 285–292.

Necula, S.-C. (2023). Exploring the impact of time spent reading product information on e-commerce websites: A machine learning approach to analyze consumer behavior. *Behavioral Sciences*, *13*, 439.

Ni, Y., Jones, D., & Wang, Z. (2020). Consensus variational and Monte Carlo algorithms for Bayesian nonparametric clustering. In *2020 IEEE International Conference on Big Data* (pp. 204–209).

Rabinovich, M., Angelino, E., & Jordan, M. I. (2015). Variational consensus Monte Carlo. In *28th Annual Conference on Neural Information Processing Systems (NIPS 2015)* (pp. 1207–1215). Curran Associates.

Saleem, H., Uddin, M. K. S., Habib-ur Rehman, S., Saleem, S., & Aslam, A. M. (2019). Strategic data driven approach to improve conversion rates and sales performance of e-commerce websites. *International Journal of Scientific & Engineering Research*, *10*, 588–593.

Scott, S. L. (2017). Comparing consensus Monte Carlo strategies for distributed Bayesian computation. *Brazilian Journal of Probability and Statistics*, *31*, 668–685.

Scott, S. L., Blocker, A. W., Bonassi, F. V., Chipman, H. A., George, E. I., & McCulloch, R. E. (2016). Bayes and Big Data: The consensus Monte Carlo algorithm. *International Journal of Management Science and Engineering Management*, *11*, 78–88.

Sinharay, S. (2003). Assessing convergence of the markov chain monte carlo algorithms: A review. *ETS Research Report Series*, *2003*, i–52.

Tong, T., Xu, X., Yan, N., & Xu, J. (2022). Impact of different platform promotions on online

sales and conversion rate: The role of business model and product line length. *Decision Support Systems*, *156*, 113746.

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

Wang, X., Guo, F., Heller, K. A., & Dunson, D. B. (2015). Parallelizing MCMC with random partition trees. In *28th Annual Conference on Neural Information Processing Systems (NIPS 2015)* (pp. 451–459). Curran Associates.

Wu, D., Shang, M., Luo, X., & Wang, Z. (2022). An L1-and-L2-norm-oriented latent factor model for recommender systems. *IEEE Transactions on Neural Networks and Learning Systems*, *33*, 5775–5788.

Zerbini, C., Bijmolt, T. H., Maestripieri, S., & Luceri, B. (2022). Drivers of consumer adoption of e-commerce: A meta-analysis. *International Journal of Research in Marketing*, *39*, 1186–1208.

Zhao, L., Chen, Y., & Schaffner, D. W. (2001). Comparison of logistic regression and linear regression in modeling percentage data. *Applied and Environmental Microbiology*, *67*, 2129–2135.

Zumstein, D., & Kotowski, W. (2020). Success factors of e-commerce: Drivers of the conversion rate and basket value. In *18th International Conference e-Society* (pp. 43–50). IADIS Press.

A. Parameter Estimates of Bayesian Models

Tables A.6, A.7, and A.8 show the parameter estimates of model B, model C with information-based averaging, and model C with simple averaging. We report the mean, standard deviation, and HPD interval. The 95% HPD interval is the smallest possible interval containing 95% of the draws, such that we can assess how reliable the estimated parameters are.

Table A.6: Summary of realized draws for parameters of model B. We report the mean, standard deviation, upper and lower bound of the 95% HPD, and an indicator of whether zero lies in the HPD.

Variable	Mean	Std. dev.	HPD lower	HPD upper	Zero in HPD
pricestar3	0.038	0.021	-0.002	0.080	*
pricestar4	0.085	0.017	0.052	0.119	
pricestar5	0.116	0.023	0.069	0.159	
logprice	-0.695	0.009	-0.713	-0.677	
lastmileselect	0.070	0.043	-0.015	0.152	*
dt1uot12t15	0.409	0.037	0.339	0.481	
dt1uot16t19	0.418	0.032	0.353	0.479	
dt1uot20t22	0.400	0.035	0.331	0.468	
dt1uot23t00	0.351	0.045	0.263	0.439	
dt23	0.139	0.034	0.074	0.205	
enrich23	0.072	0.014	0.046	0.100	
average_product_review_score	0.240	0.012	0.217	0.263	
log_num_of_product_reviews	0.132	0.008	0.116	0.147	
zero_product_reviews	-0.044	0.019	-0.082	-0.006	
classCore	0.091	0.020	0.053	0.130	
classTraffic	0.348	0.020	0.310	0.388	
seller_webshop	0.933	0.229	0.479	1.394	
average_seller_review_score	0.092	0.019	0.055	0.131	
log_num_of_seller_reviews	0.022	0.005	0.013	0.032	
zero_seller_reviews	-0.439	0.233	-0.897	0.039	*
r_μ	0.985	0.002	0.982	0.990	

Table A.7: Summary of realized draws for parameters of model C with information-based averaging. We report the mean, standard deviation, upper and lower bound of the 95% HPD, and an indicator of whether zero lies in the HPD.

variable	Mean	Std. dev.	HPD lower	HPD upper	Zero in HPD
pricestar3	0.018	0.002	0.014	0.021	
pricestar4	0.065	0.001	0.062	0.067	
pricestar5	0.106	0.002	0.103	0.110	
logprice	-0.689	0.001	-0.691	-0.688	
lastmileselect	0.032	0.003	0.026	0.038	
dt1uot12t15	0.375	0.003	0.370	0.380	
dt1uot16t19	0.398	0.002	0.393	0.402	
dt1uot20t22	0.415	0.003	0.410	0.421	
dt1uot23t00	0.363	0.003	0.356	0.370	
dt23	0.138	0.002	0.134	0.143	
enrich23	0.078	0.001	0.076	0.081	
average_product_review_score	0.242	0.001	0.241	0.244	
log_num_of_product_reviews	0.142	0.001	0.140	0.143	
zero_product_reviews	-0.027	0.002	-0.030	-0.024	
classCore	0.104	0.001	0.102	0.107	
classTraffic	0.335	0.002	0.332	0.338	
seller_webshop	0.893	0.014	0.866	0.921	
average_seller_review_score	0.086	0.001	0.084	0.089	
log_num_of_seller_reviews	0.026	0.000	0.026	0.027	
zero_seller_reviews	-0.373	0.015	-0.400	-0.344	
r_μ	0.986	0.002	0.983	0.989	

Table A.8: Summary of realized draws for parameters of model C with simple averaging. We report the mean, standard deviation, upper and lower bound of the 95% HPD, and an indicator of whether zero lies in the HPD.

Variable	Mean	Std. dev.	HPD lower	HPD upper	Zero in HPD
pricestar3	0.017	0.002	0.013	0.021	
pricestar4	0.065	0.002	0.062	0.069	
pricestar5	0.110	0.002	0.106	0.114	
logprice	-0.688	0.001	-0.690	-0.686	
lastmileselect	0.033	0.004	0.025	0.041	
dt1uot12t15	0.375	0.004	0.367	0.382	
dt1uot16t19	0.397	0.003	0.391	0.403	
dt1uot20t22	0.416	0.004	0.409	0.424	
dt1uot23t00	0.363	0.004	0.354	0.372	
dt23	0.140	0.004	0.133	0.147	
enrich23	0.079	0.002	0.076	0.082	
average_product_review_score	0.242	0.001	0.240	0.244	
log_num_of_product_reviews	0.142	0.001	0.140	0.143	
zero_product_reviews	-0.028	0.002	-0.031	-0.024	
classCore	0.105	0.002	0.101	0.109	
classTraffic	0.336	0.002	0.333	0.341	
seller_webshop	0.994	0.025	0.946	1.041	
average_seller_review_score	0.087	0.002	0.083	0.090	
log_num_of_seller_reviews	0.026	0.001	0.025	0.027	
zero_seller_reviews	-0.476	0.025	-0.526	-0.430	
r_μ	0.985	0.002	0.981	0.988	