Model-based Purchase Predictions for Large Assortments*

Bruno Jacobs
Department of Business Economics and Econometric Institute
Erasmus School of Economics
jacobs@ese.eur.nl

Bas Donkers
Department of Business Economics
Erasmus School of Economics
donkers@ese.eur.nl

Dennis Fok
Econometric Institute
Erasmus School of Economics
dfok@ese.eur.nl

February 18, 2016

*An abbreviated version of this paper is forthcoming in Marketing Science.

Abstract

Being able to accurately predict what a customer will purchase next is of paramount importance to successful online retailing. In practice, customer purchase history data is readily available to make such predictions, sometimes complemented with customer characteristics. Given the large assortments maintained by online retailers, scalability of the prediction method is just as important as its accuracy. We study two classes of models that use such data to predict what a customer will buy next: A novel approach that uses latent Dirichlet allocation (LDA), and mixtures of Dirichlet-Multinomials (MDM). A key benefit of a model-based approach is the potential to accommodate observed customer heterogeneity through the inclusion of predictor variables. We show that LDA can be extended in this direction while retaining its scalability. We apply the models to purchase data from an online retailer and contrast their predictive performance with that of a collaborative filter and a discrete choice model. Both LDA and MDM outperform the other methods. Moreover, LDA attains performance similar to that of MDM while being far more scalable, rendering it a promising approach to purchase prediction in large assortments.

1. Introduction

The ability to predict what a customer will purchase next is valuable in many marketing applications and this holds especially true for online retailing. Adequate predictions for the next products to be purchased enable online retailers to: implement a product recommendation system; determine the positions of products in the result of a customer's search query; optimize the collection of products to be displayed on a personalized landing page; or suggest products to complement the contents of a customer's shopping basket.

Examples of personalization in practice are Amazon's "Customers Who Bought This Item Also Bought" section, Apple's iTunes Genius and the Netflix recommendation system. There is also clear evidence that such personalized configurations influence behavior (Ghose et al., 2014; Pan et al., 2007; Salganik et al., 2006). All these applications have in common that they require a personalized selection of products out of a potentially large assortment. Ideally, the selection contains those products that are most likely to be of interest to the customer. Moreover, the selection should be relatively small as the available space to show products is often limited.

The effectiveness of personalization attempts crucially depends on the accuracy of the predictions. A complicating factor in purchase prediction is the fact that the typical online retailer sells items from a very broad assortment to an even larger customer base. Hence predictions should not only be accurate, but the prediction method should scale to large applications as well (Naik et al., 2008). Additionally, in order to be useful in an online setting the predictions should be available in near real-time. Obtaining predictions, and updating them as new information comes in, should therefore be fast.

The typical data available at an online retailer for purchase prediction are the customer purchase histories. In some cases additional customer characteristics (e.g. demographics) are also available. However, on the product level characteristics are often absent and if such product descriptions are available, it is not obvious how to extract useful predictors from this information. In this paper we therefore focus on predicting purchase behavior based on purchase history data, possibly complemented with customer characteristics.

Many online retailers predict a customer's next purchase using collaborative filtering algorithms, for example, by relying on counts of the co-occurrence of items in purchase history data (Jannach et al., 2011; Liu et al., 2009). In such a count-based approach a decision has to be made on how to measure the co-occurrence of items, as one can count pairs, triplets, or even higher-order product combinations. A choice for small sets of items results in information loss, i.e. purchase patterns that span many products might not be easily identified. On the other hand, for large combinations of products the matrix of co-occurrence counts becomes very sparse, resulting in predictions that are based on just a few matches in the customer base. Another challenge in collaborative filtering algorithms is incorporating customer characteristics. One possible approach is to partition the customer-base using such characteristics. However, this can only be done for a couple of variables with a limited number of levels, as otherwise sample sizes per subgroup become too small.

In contrast, model-based approaches to predict individuals' choices have a long history in marketing (Guadagni and Little, 1983; McFadden, 1986; Wagner and Taudes, 1986; Fader and Hardie, 1996) and such methods are well-suited to include customer characteristics. However, the usual implementations of these models tend to break down in the typical online retail setting, where a wide variety of products is sold to a large number of customers (Naik et al., 2008). One way to make methods more scalable is to consider only a subset of the data in terms of customers and/or products (Zanutto and Bradlow, 2006). Clearly, this is not a viable solution if the aim is to predict purchase behavior for each individual customer across the entire product assortment.

In this paper we try to bridge the gap between retail practice and marketing academia by discussing model-based prediction methods that do work in the context of large assortments. By developing such methods we open up an avenue for future research on marketing interventions in large-scale assortments, for example on the effectiveness of product recommendations, extending the work of Bodapati (2008). Note that this would not be feasible with the heuristic, count-based approaches currently used in practice. We consider two model-based approaches. In addition, we present (an implementation of) a count-based collaborative filter and a scalable version of a discrete choice model that will serve as benchmarks. We compare the methods on their (i) heterogeneity assumptions, (ii) estimation complexity, (iii) memory requirements for real-time online predictions, and (iv) predictive performance.

The first method we present is a novel approach inspired by topic models as used in the text modeling literature. Traditionally, a topic model describes a document by relating the words in the text to latent topics. We adapt this class of models to the purchase prediction context: Words become product purchases, a document is a customer's purchase history, and a topic represents a certain preference for products in the assortment. Given that the word "topic" does not make much sense in a retailing context, we refer to topics as motivations.[1] Naturally, customers can have more than one motivation, just like a document can cover multiple topics. This idea leads to a class of models that can describe and predict customer purchase behavior in large assortments.

The most frequently used topic model is latent Dirichlet allocation (LDA) by Blei et al. (2003). This model has been used to analyze very large text corpora (Ramage et al., 2010; Mimno et al., 2012), showing that LDA provides the necessary scalability. In contrast to the text modeling literature, where documents tend to contain many words, customers often have only a couple of purchases, or they might even be entirely new to the retailer. Given such limited information per customer, we need to formally estimate the population-level a-priori probabilities of having particular motivations. This extends the text modeling implementation of LDA, where these probabilities are typically considered to be known, or at best calibrated using heuristics (Wallach et al., 2009; Asuncion et al., 2009).

To account for observed heterogeneity, we extend LDA by relating customer characteristics to the a-priori motivation probabilities. This can capture heterogeneity that is related to variables such as referrer site, demographics, or other customer characteristics. Most likely this increases the predictive power of the model, in particular for the customers with few or no observed purchases. We refer to this model as LDA-X.

The next method we consider is a mixture of Dirichlet-Multinomials (MDM) (Jain et al., 1990). MDM specifies individual-specific probability vectors that contain a customer's purchase probabilities over all products in the assortment. In turn, these probability vectors follow a discrete mixture of Dirichlet distributions. MDM has previously been applied in marketing (Jain et al., 1990), but to the best of our knowledge never to a large product assortment. Although, in theory customer characteristics can also be included in MDM we will argue that the resulting model will no longer be feasible in terms of estimation complexity, given the setting of our application.

The predictive performance of LDA(-X) and MDM is compared to that of a count-based collaborative filter and a discrete choice model. We assess the predictive performance using data from an online retailer. For each method, we create customer-specific prediction sets that contain the products that are most likely to be purchased. These sets are next matched with hold-out purchase data. To gain more insight into the differences between the methods, we also consider the predictive performance for groups of customers that differ in the length of their observed purchase history. Furthermore, in a setting where repeat purchases are frequently made, e.g. fast moving consumer goods, performing well by correctly predicting frequently purchased products or repeat purchases might not be too difficult. Such recommendations might even be perceived as trivial or boring (Fleder and Hosanagar, 2009). We therefore also study the predictive performance for *unexpected* products, which we define as products that have not previously been purchased by the customer and are in the tail of the assortment.

The remainder of this article proceeds as follows: In Section 2 we present the methods used in this research and discuss their heterogeneity assumptions and scalability. Tech-

[1]While intuitively plausible, we do not claim that the actual decision process is driven by these motivations.

nical details are available in appendices. Subsequently, we apply the methods to data of an online retailer. An overview of this data is provided in Section 3 and the results are reported in Section 4. To conclude, we summarize our findings and provide directions for future research in Section 5.

2. Methods

In this section we present the prediction methods we consider in this paper. First, we introduce two model-based prediction methods, LDA(-X) and MDM, that infer latent customer traits from purchase data. We compare these methods on their heterogeneity assumptions and estimation complexity. Next, the two benchmark methods are introduced: a set of collaborative filters (CF) and a model built on discrete choice methodology (DCM) that captures customer heterogeneity through constructed, but *observed* predictor variables.

Subsequently, all methods are compared on their suitability to update predictions in a real-time setting. Finally, we discuss how we assess the quality of predictions.

All methods share the following notation: The products from the J-dimensional assortment are numbered $j = 1, \ldots, J$. For each customer $i = 1, \ldots, I$ we observe n_i product purchases from this assortment. The purchase history of customer i is denoted by the vector $\mathbf{y}_i = [y_{i1}, \ldots, y_{in_i}]$, where $y_{in} \in \{1, \ldots, J\}$ represents the n-th purchase of customer i. In addition we have customer-level characteristics coded in the K-dimensional vector $\mathbf{x}_i = [x_{i1}, \ldots, x_{iK}]'$. We combine the purchase histories in $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_I\}$ and the predictor variables in $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_I\}$.

2.1. Latent Dirichlet allocation

Our first model is inspired on topic models. The key idea underlying our application of these models to the context of purchase history data is that customer purchases are driven by a (small) set of latent motivations (the topics). Each motivation then drives preferences for a subset of products in the assortment, for example, a preference for eco-friendly products, for low-fat products, or for products for the sensitive skin.

In general, customers are likely to be driven by different motivations over time and even within a single purchase occasion. Additionally, the same product purchased by different customers may be driven by different underlying motivations: A movie can be purchased by a fan of the lead actor, or by a customer that is fond of the movie's genre. These features are embedded in topic models, in which customers may have multiple motivations and products may be associated with more than one motivation.

The basis for our method is latent Dirichlet allocation (LDA) introduced by Blei et al. (2003). LDA has been proven to scale to applications well beyond the dimensions of a typical online retailer. For example, it has been used to analyze over 8 million posts on Twitter that contain words from a vocabulary of more than 5 million entries (Ramage et al., 2010), or for the analysis of 1.2 million out-of-copyright books (Mimno et al., 2012). Below, we first present the details of our adaptation of LDA in the context of predicting customer purchase behavior. Next, we extend LDA by including customer-level predictor variables.

In LDA each latent motivation $m = 1, \ldots, M$ is represented by a probability vector $\boldsymbol{\phi}_m$ over the complete J-dimensional assortment. Given that a purchase is driven by motivation m, the probability of buying product j is simply ϕ_{mj}. The motivation-specific probability vectors are distributed as

$$\boldsymbol{\phi}_m | \boldsymbol{\beta} \sim \text{Dirichlet}_J(\boldsymbol{\beta}). \tag{1}$$

A priori there is no reason to favor one product over another in a motivation. This is reflected in the parameterization of $\boldsymbol{\beta}$, where we set each element equal to a common value β_0. This value determines whether the distribution in (1) tends to favor more narrow (β_0 close to zero) or more broad (large β_0) motivations (Wallach et al., 2009).

Even though each purchase is driven by a single motivation, a customer's entire purchase history may be driven by multiple motivations. This variation is described by an individual-specific discrete mixture $\boldsymbol{\theta}_i$ over the M motivations. The probability that a product purchase of customer i is driven by motivation m is then given by θ_{im}. These probabilities differ across customers and are modeled as

$$\boldsymbol{\theta}_i | \boldsymbol{\alpha} \sim \text{Dirichlet}_M(\boldsymbol{\alpha}). \tag{2}$$

Here, $\boldsymbol{\alpha}$ is an M-dimensional vector that captures the relevance of each motivation across the customer base. The expected value of the probability that motivation m drives a purchase equals

$$\text{E}\left[\theta_{im} | \boldsymbol{\alpha}\right] = \frac{\alpha_m}{\sum_{l=1}^{M} \alpha_l}. \tag{3}$$

Therefore, the larger the value of α_m, the more likely it is that a customer will make a purchase driven by motivation m.

The last step is to link motivations to actual purchases. We denote by $z_{in} \in \{1, \ldots, M\}$ the actual motivation that drives purchase y_{in}. As motivations are latent, we have to account for all possible motivations to obtain the marginal probability that customer i will purchase product j, resulting in

$$\begin{aligned} \Pr\Big[\, y_{in} = j | \left\{\boldsymbol{\phi}_l\right\}_{l=1}^{M}, \boldsymbol{\theta}_i \,\Big] &= \sum_{m=1}^{M} \Pr\Big[\, y_{in} = j | z_{in} = m, \left\{\boldsymbol{\phi}_l\right\}_{l=1}^{M} \Big] \Pr[\, z_{in} = m | \boldsymbol{\theta}_i \,] \\ &= \sum_{m=1}^{M} \phi_{mj} \theta_{im}. \end{aligned} \tag{4}$$

In the topic modeling literature it is common practice to determine the parameters of the Dirichlet distributions $\boldsymbol{\alpha}$ (for $\boldsymbol{\theta}_i$) and β_0 (for $\boldsymbol{\phi}_m$) by means of heuristics, rather than formally inferring their values from available data (Wallach et al., 2009), for example, imposing $\boldsymbol{\alpha} = 50/M$ (Griffiths and Steyvers, 2004) and $\beta_0 = 0.01$ (Steyvers and Griffiths, 2013), or by applying a grid search for $\boldsymbol{\alpha}$ and β_0 (Asuncion et al., 2009). These heuristics are not directly applicable in our setting as they have been designed for text modeling.

Given that purchase histories tend to be much shorter than documents, we expect the LDA predictions to be more sensitive to the values of $\boldsymbol{\alpha}$ and (to a lesser degree) of β_0. We therefore extend the common LDA model and place proper prior distributions on both parameters and formally estimate $\boldsymbol{\alpha}$ and β_0 in a Bayesian setting.

We specify a log-normal distribution for α_m, that is, we define

$$\log(\alpha_m) = \gamma_m, \tag{5}$$

and set a normal prior for γ_m. We set the mode of the log-normal distribution equal to M^{-1}, which is within the range of values frequently used in the literature on text modeling, and place 10% of its probability mass above 1.[2] This prior specification favors $\boldsymbol{\theta}_i$-vectors that allocate the majority of the probability mass to a small number of motivations, while it still allows for more uniformly distributed $\boldsymbol{\theta}_i$-vectors. Similarly, we place a log-normal distribution on β_0 with its mode equal to 0.01 and 10% of its probability mass above 1. This specification supports $\boldsymbol{\phi}_m$-vectors where only a few products from the assortment receive significant probability mass, representing fairly specific motivations. Still, this prior is rather uninformative and broader motivations that spread the probability mass more equally over the assortment remain quite likely.

These prior specifications also allow us to easily extend LDA by including customer characteristics, coded in $\mathbf{x}_i$. Such variables are likely to improve the predictive performance of the model. We extend the log-linear specification for α_m in (5) to α_{im} as follows:

$$\log(\alpha_{im}) = \gamma_m + \mathbf{x}_i'\boldsymbol{\delta}_m. \tag{6}$$

This links customer preferences – represented by the likelihood of each of the motivations – to the additional customer-level information, resulting in LDA-X. To illustrate the effect of this specification on the distribution of $\boldsymbol{\theta}_i$ consider the expected value of θ_{im}, which gives the probability that a typical customer with characteristics $\mathbf{x}_i$ makes a purchase driven by motivation m:

$$\mathrm{E}\left[\theta_{im}|\boldsymbol{\alpha}_i\right] = \frac{\alpha_{im}}{\sum_{l=1}^{M}\alpha_{il}} = \frac{\exp(\gamma_m + \mathbf{x}_i'\boldsymbol{\delta}_m)}{\sum_{l=1}^{M}\exp(\gamma_l + \mathbf{x}_i'\boldsymbol{\delta}_l)}. \tag{7}$$

The $\boldsymbol{\delta}_m$ parameters capture the dependence of the probability that motivation m is used, on the customer-specific variables $\mathbf{x}_i$. The prior distribution of γ_m and $\boldsymbol{\delta}_m$ can only be sensibly determined if the level and scale of the $\mathbf{x}_i$ variables are known. We therefore standardize the customer-level variables such that they have mean zero and unit variance. Given this scale, we assume that all elements in $\boldsymbol{\delta}_m$ are normally distributed with zero mean and variance equal to 0.04. This corresponds to a prior 95% confidence interval that is approximately equal to $[-0.4, +0.4]$. Note that this prior distribution is chosen to be relatively narrow on purpose, as the effect of $\boldsymbol{\delta}_m$ on α_{im} is exponential. As $\mathbf{x}_i$ is mean-centered, we use the same prior for γ_m as in LDA.

To obtain customer-specific predictive distributions, we condition on the model structure of LDA. In particular, given the model parameters $\boldsymbol{\alpha}$, β_0 and the latent purchase

[2] These two conditions implicitly identify the two parameters of the log-normal distribution.

assignments $\mathbf{Z}$, the predictive distribution for a new purchase $\tilde{y}_{in}$ can be shown to equal (Griffiths and Steyvers, 2004):

$$\begin{aligned}
\Pr[\,\tilde{y}_{in} = j|\mathbf{Z}, \boldsymbol{\alpha}, \beta_0, \mathbf{Y}\,] &= \sum_{m=1}^{M} \Pr[\,\tilde{y}_{in} = j|\tilde{z}_{in} = m, \mathbf{Z}, \beta_0, \mathbf{Y}\,] \Pr[\,\tilde{z}_{in} = m|\mathbf{z}_i, \boldsymbol{\alpha}\,] \\
&= \sum_{m=1}^{M} \mathbb{E}[\,\phi_{mj}|\mathbf{Z}, \beta_0, \mathbf{Y}\,] \, \mathbb{E}[\,\theta_{im}|\mathbf{z}_i, \boldsymbol{\alpha}\,] \\
&= \sum_{m=1}^{M} \left(\frac{\beta_0 + c_{mj}^{\mathrm{MJ}}}{J\beta_0 + \sum_{p=1}^{J} c_{mp}^{\mathrm{MJ}}} \right) \left(\frac{\alpha_m + c_{im}^{\mathrm{IM}}}{\sum_{l=1}^{M} \alpha_l + c_{il}^{\mathrm{IM}}} \right),
\end{aligned} \tag{8}$$

where c_{mj}^{MJ} is the number of times a purchase of product j is driven by motivation m and c_{im}^{IM} is the number of purchases made by customer i that are driven by motivation m. To obtain the predictive distribution for the LDA-X model one simply replaces $\boldsymbol{\alpha}$ with $\boldsymbol{\alpha}_i$ in (8).

2.2. Dirichlet-Multinomial models

The Dirichlet-Multinomial (DM) model (Jeuland et al., 1980; Goodhardt et al., 1984) is a known model-based approach to capture heterogeneity in purchase behavior. Applications of this model can be found in Grover and Srinivasan (1987); Fader (1993) and Fader and Schmittlein (1993). In this model, each customer is endowed with an individual-specific vector $\boldsymbol{\varphi}_i$ containing the purchase probabilities for each product in the J-dimensional assortment, where $\sum_{p=1}^{J} \varphi_{ip} = 1$. The probability that customer i purchases product j at a specific purchase occasion n is given by:

$$\Pr[\,y_{in} = j|\boldsymbol{\varphi}_i\,] = \varphi_{ij}. \tag{9}$$

Large values for the purchase probability φ_{ij} imply that customer i is likely to buy product j. In the DM model the customer-specific $\boldsymbol{\varphi}_i$-vectors are assumed to arise from a single Dirichlet distribution:

$$\boldsymbol{\varphi}_i|\boldsymbol{\beta} \sim \mathrm{Dirichlet}_J(\boldsymbol{\beta}). \tag{10}$$

The $\boldsymbol{\beta}$-vector describes the overall purchase behavior in the customer base: If product j is frequently purchased, β_j will have a large value relative to the other values in $\boldsymbol{\beta}$ and vice versa.

The original DM model has been extended such that the probability vectors $\boldsymbol{\varphi}_i$ originate from a finite mixture of Dirichlet distributions (Jain et al., 1990), not from a single Dirichlet distribution. This extension is known as a mixture of Dirichlet-Multinomials (MDM).

In MDM, each customer is assigned to one of M segments and each segment is characterized by its own Dirichlet distribution. Given that customer i is allocated to segment m, denoted by $s_i = m$, the customer's purchase probabilities $\boldsymbol{\varphi}_i$ are distributed as

$$\boldsymbol{\varphi}_i|s_i = m, \boldsymbol{\beta}_m \sim \mathrm{Dirichlet}_J(\boldsymbol{\beta}_m). \tag{11}$$

The $\boldsymbol{\beta}_m$-vectors are segment specific, describing the distribution of the purchase probability vectors for customers in segment m. Customers are hence expected to be similar, although not identical, within a segment, but rather different across segments.

Segment membership in MDM is described by an M-dimensional Categorical distribution with probability vector $\boldsymbol{\pi}$. The element π_m gives the a-priori probability that a customer is a member of segment m, that is,

$$\Pr[\, s_i = m|\boldsymbol{\pi}\,] = \pi_m. \tag{12}$$

As we consider MDM within the Bayesian paradigm we also specify prior distributions over $\boldsymbol{\pi}$ and the $\boldsymbol{\beta}_m$-vectors. For $\boldsymbol{\pi}$ it is natural to favor no segment over any other a priori, therefore we use a uniform distribution over the $(M-1)$-dimensional simplex. This corresponds to an M-dimensional Dirichlet distribution, parameterized by a vector of ones. For each β_{mj} we use a log-normal prior distribution with its mode located at 0.01 and 10% of the probability mass located above 1. This specification allows for $\boldsymbol{\varphi}_i$-vectors that allow many products to be purchased with a large probability, but it favors segments of customers who purchase from a more limited subset of the assortment.

Similar to the approach in LDA, we obtain customer-specific predictive distributions of a new purchase $\tilde{y}_{in}$ conditional on the data, parameters, and segment allocations. In MDM this requires a prediction of segment membership of the customer, combined with the purchase probabilities, conditional on segment membership:

$$\begin{aligned}
\Pr\Big[\, \tilde{y}_{in} = j|\mathbf{s}^{\setminus i}, \{\boldsymbol{\beta}_l\}_{l=1}^{M}, \mathbf{y}_i \,\Big] &= \sum_{m=1}^{M} \Pr[\, \tilde{y}_{in} = j|s_i = m, \boldsymbol{\beta}_m, \mathbf{y}_i\,] \Pr\Big[\, s_i = m|\mathbf{s}^{\setminus i}, \{\boldsymbol{\beta}_l\}_{l=1}^{M}, \mathbf{y}_i \,\Big] \\
&= \sum_{m=1}^{M} \mathbb{E}[\, \varphi_{ij}|s_i = m, \boldsymbol{\beta}_m, \mathbf{y}_i\,] \Pr\Big[\, s_i = m|\mathbf{s}^{\setminus i}, \{\boldsymbol{\beta}_l\}_{l=1}^{M}, \mathbf{y}_i \,\Big] \\
&= \sum_{m=1}^{M} \left(\frac{\beta_{mj} + c_{ij}^{\mathrm{IJ}}}{\sum_{p=1}^{J} \beta_{mp} + c_{ip}^{\mathrm{IJ}}} \right) \Pr\Big[\, s_i = m|\mathbf{s}^{\setminus i}, \{\boldsymbol{\beta}_l\}_{l=1}^{M}, \mathbf{y}_i \,\Big],
\end{aligned} \tag{13}$$

where $\Pr\Big[\, s_i = m|\mathbf{s}^{\setminus i}, \{\boldsymbol{\beta}_l\}_{l=1}^{M}, \mathbf{y}_i \,\Big]$ is specified in Appendix A (see equation (32)) and c_{ij}^{IJ} equals the number of times customer i has purchased product j. If i is a new customer $c_{ij}^{\mathrm{IJ}} = 0$ for all j by definition. Note that both components in (13) depend on the customer's purchase history, unlike LDA where only the motivation probabilities are customer specific.

2.3. Model inference

The predictive distributions specified above are conditional on the number of segments/motivations M, the model parameters, and segment/motivation allocations to customers/purchases. For a given number of M, we rely on Bayesian methodology to infer the model parameters and latent variables of the models. Direct inference on the posterior distribution is not tractable and therefore we derive Markov Chain Monte Carlo (MCMC) methods to generate samples from the posterior distribution. To be specific, we use a random walk Metropolis-Hastings within Gibbs sampler to draw samples from the target posterior distribution. The predictive distributions can then be obtained by averaging over these draws.

The full posterior of LDA(-X) is given by:

$$p(\mathbf{Z}, \{\boldsymbol{\phi}_l\}_{l=1}^{M}, \beta_0, \{\boldsymbol{\theta}_i\}_{i=1}^{I}, \boldsymbol{\gamma}, \{\boldsymbol{\delta}_l\}_{l=1}^{M} | \mathbf{Y}, \mathbf{X}), \tag{14}$$

where $\{\boldsymbol{\delta}_l\}_{l=1}^{M}$ is only relevant when customer characteristics $\mathbf{X}$ are included. Straightforward use of a Gibbs sampler for this posterior distribution is very inefficient. This is the result of a strong dependence between the latent motivation assignments $\mathbf{Z}$ on the one hand and the parameters $\boldsymbol{\phi}_m$ and $\boldsymbol{\theta}_i$ on the other hand. A Gibbs sampler would therefore require an excessive number of draws to properly explore this posterior. Instead, we take advantage of the fact that the Dirichlet distribution is the conjugate prior for a Categorical random variable. This allows us to marginalize over the $\boldsymbol{\phi}_m$ and $\boldsymbol{\theta}_i$ parameters, while retaining closed-form expressions for the conditional distributions of the other parameters in LDA. By doing so we substantially improve the mixing properties of the Gibbs sampler (Griffiths and Steyvers, 2004). Hence, we examine the so-called *collapsed* posterior distribution of LDA(-X), defined as:

$$p(\mathbf{Z}, \beta_0, \boldsymbol{\gamma}, \{\boldsymbol{\delta}_l\}_{l=1}^{M} | \mathbf{Y}, \mathbf{X}). \tag{15}$$

The elements of $\mathbf{Z}$ are sampled using a Gibbs sampler, while for the other parameters we implement a random walk Metropolis-Hastings sampler.

The set-up for inference in MDM is very similar to LDA(-X). The complete posterior distribution is given by:

$$p(\mathbf{s}, \{\boldsymbol{\varphi}_i\}_{i=1}^{I}, \{\boldsymbol{\beta}_l\}_{l=1}^{M}, \boldsymbol{\pi} | \mathbf{Y}). \tag{16}$$

Again, we marginalize over the discrete distributions $\boldsymbol{\varphi}_i$ and $\boldsymbol{\pi}$, resulting in a collapsed posterior distribution of MDM:

$$p(\mathbf{s}, \{\boldsymbol{\beta}_l\}_{l=1}^{M} | \mathbf{Y}). \tag{17}$$

Here the segment allocations $\mathbf{s}$ can be sampled in a Gibbs step, while the $\boldsymbol{\beta}_l$ parameters require a random walk Metropolis-Hastings sampler.

LDA(-X) and MDM are both members of the general class of mixture models. This class of models is well known to be susceptible to end up in an area around a local maximum of the posterior distribution. As is common in this literature, this risk is reduced by using multiple random starts (Wedel and Kamakura, 2000; Train, 2009). For each value of M, we consider 250 different random starts. We reduce the computational burden of this approach by evaluating each random start at several intermediate steps of the estimation routine. At each step, we continue only with the best performing candidates. The performance is measured by the likelihood that results from the model's predictive distributions, averaged over purchases in a model-selection data set. This measure is closely related to the goal of predicting a new purchase as accurately as possible.

The same performance measure is also used to determine the number of motivations (for LDA(-X)) or segments (for MDM). In particular, for each model we increase the value of M until we find a decrease in the average predictive likelihood of the model-selection data.[3] More details on the estimation routines are provided in Appendix A.

2.4. Comparison of LDA(-X) and MDM

Although the structures of LDA(-X) and MDM might appear quite similar at first sight, these models differ fundamentally on various grounds. In this subsection we first discuss this difference in terms of customer heterogeneity. Next, we consider the estimation complexity of the LDA(-X) and MDM models.

Heterogeneity assumption

MDM assumes that heterogeneity in purchase behavior can be described by segmenting the customer base in groups of customers. Customers across segments are expected to be dissimilar, while customers within a segment are expected to be rather similar. Hence, similarity between customers is mainly driven by segment membership. In LDA(-X) purchase behavior is described by motivations, where each motivation represents a preference for certain products in the assortment. Heterogeneity in purchase behavior is described by customer-specific probabilities for these motivations. This leads to a model where the purchases of a single customer are driven by *multiple* motivations. Here similarity between customers is motivation specific. Customers can have very similar purchase behavior for one set of products – corresponding to a shared motivation – and be very different for a set of products that belong to another motivation.

Which heterogeneity structure fits best depends on the specific situation. If customers typically have one or very few motivations, grouping customers in segments might be beneficial. If many combinations of motivations are present, the continuous mixture of motivations in LDA(-X) will be more parsimonious. Therefore, if a retailer has many different (latent) subcategories in its product assortment, and preferences across these subcategories vary rather independently across individuals, it is likely that the heterogeneity can be specified more parsimoniously by LDA(-X).

Although MDM assumes a hard clustering of customers into segments, one will use posterior segment probabilities to eventually make predictions. This will typically lead to a form of soft clustering, where a weighted combination of different segments is used. This brings the heterogeneity structure of MDM closer to that of LDA(-X). As we observe more purchases, the posterior segment probabilities in MDM will of course become more and more extreme, and in the end this converges to strictly assigning a customer to a single segment.

[3] In order to validate this approach we also consider the models for larger values of M. The predictive performance stabilized at the values obtained with the selected value of M.

Estimation complexity

The different heterogeneity assumptions underlying LDA(-X) and MDM have a large impact on estimation complexity through the number of customer-specific parameters. In MDM each customer is endowed with a distribution over the J-dimensional assortment, while in LDA(-X) a customer is described by a probability distribution over the M motivations, where M is generally much smaller than J. Even though we marginalize over these customer-specific distributions, this still affects the scalability of the models. Table 1 summarizes for each model the parameters that need to be sampled to infer the model structure after marginalization. We differentiate between the sampling technique required, as Gibbs steps tend to be much faster and have better mixing properties than Metropolis-Hastings steps (Damien et al., 1999).

Table 1: Parameters to sample in the MCMC estimation procedures across different models.

	Gibbs sampler		Metropolis-Hastings sampler	
Model	Parameters	No. parameters	Parameters	No. parameters
MDM	$\mathbf{s}$	I	$\{\boldsymbol{\beta}_l\}_{l=1}^M$	$M \times J$
LDA	$\mathbf{Z}$	N	β_0, $\boldsymbol{\gamma}$	$1 + M$
LDA-X	$\mathbf{Z}$	N	β_0, $\boldsymbol{\gamma}$, $\{\boldsymbol{\delta}_l\}_{l=1}^M$	$1 + M \times (1 + K)$

where
I: number of customers
J: assortment size
N: total number of purchases
M: number of segments/motivations
K: number of predictor variables in $\mathbf{x}_i$

In LDA(-X) we need as many motivation allocations as purchases (N in total), whereas for MDM we only need to sample one segment allocation per customer (I in total). Although the number of allocations is larger in LDA(-X), this does not imply that the total allocation in LDA(-X) is computationally more demanding. The sampling step for each motivation assignment in LDA(-X) involves only elementary arithmetic operations, while for each segment allocation in MDM we have to evaluate complex Gamma functions. It is difficult to exactly quantify the difference in computational complexity as it also depends on the (latent) structure in the data, but we anticipate that MDM will be slightly more complex for these Gibbs sampling steps.[4]

The remaining model parameters are sampled using Metropolis-Hastings steps and each of these steps is computationally demanding. For LDA we sample $1+M$ parameters and for LDA-X this increases to $1 + M \times (1 + K)$ parameters. These numbers are in sharp contrast to MDM in which $M \times J$ parameters are sampled. This renders MDM much more demanding in terms of estimation time, as the assortment size J is large. This is the price that has to be paid for the many degrees of freedom per customer. The number of Metropolis-Hastings steps in LDA(-X) is largely insensitive to the size of the assortment, the number of customers, and the number of purchases. In MDM,

[4]More details on the required sampling steps can be found in Appendix A.

on the other hand, the number of Metropolis-Hastings steps linearly increases with the assortment size. This limits the scalability of MDM, which is why we can only extend LDA by including observed heterogeneity through $\mathbf{x}_i$.

2.5. Benchmark methods

In this section we present the two benchmark methods to which we will compare the predictive performance of LDA(-X) and MDM. The first benchmark is a collaborative filter while the second is built on standard discrete choice modeling.

Collaborative filtering

A collaborative filter is a deterministic algorithm that predicts purchases by matching customers to each other based on purchase histories. There are many possible ways to implement a collaborative filter. Details of the actual implementations used in industry are not common knowledge. Therefore, below we develop our own implementation of a collaborative filter.

Ideally, a focal customer is matched to customers who purchased the focal customer's previously purchased products and at least one additional item. However, such a matching on the complete purchase history is in general not feasible due to the curse of dimensionality; the larger the purchase history, the less likely it matches with other customers' histories.

We alleviate this curse of dimensionality by instead matching on parts of the purchase history. First, for each customer i we replace the complete purchase history vector $\mathbf{y}_i$ by the set of unique sorted subvectors of length k that can be created from $\mathbf{y}_i$. We denote this set of vectors by H_i^k. For example, for $k = 2$ a customer's purchase history is replaced by all the unique sorted pairs that can be formed using the purchase history, so $\mathbf{y}_i = [1, 1, 1, 2, 3]$ would be reduced to the set H_i^2 containing the pairs $(1, 1)$, $(1, 2)$, $(1, 3)$, and $(2, 3)$.[5] Next, for each subvector in this set we match the focal customer against all customers. If k is relatively small, this will result in many more matches compared to a matching on the complete purchase history. This solves the curse of dimensionality problem at the cost of a loss of information.

We refer to a subvector of a customer's purchase history as a product combination, denoted by $\mathbf{h}$, and $c(\mathbf{h})$ gives the number of customers who purchased product combination $\mathbf{h}$, that is,

$$c(\mathbf{h}) = \sum_{i=1}^{I} \mathrm{I}\left[\mathbf{h} \in H_i^{\dim(\mathbf{h})}\right], \tag{18}$$

where $\dim(\mathbf{h})$ denotes the dimension of $\mathbf{h}$ and $\mathrm{I}[A]$ equals 1 if condition A is true and 0 otherwise. To obtain purchase predictions for customer i, using product combinations of size k, we score all products in the assortment based on their co-occurrence with each of the product combinations in H_i^k. For product combination $\mathbf{h} \in H_i^k$ the prediction

[5] The use of unique sorted pairs implies that $(1, 1)$ occurs in H_i^2 only once and that H_i^2 contains the pair $(1, 2)$ and not $(2, 1)$.

score for product j equals the number of customers who purchased j and the products in $\mathbf{h}$, normalized by the sum of the score for $\mathbf{h}$ and *any* product $p = 1, \ldots, J$. This normalization ensures that each product combination $\mathbf{h} \in H_i^k$ receives the same weight, independent of the prevalence of $\mathbf{h}$ in other customers' purchase histories. The final product score is the sum of the normalized scores across all $\mathbf{h} \in H_i^k$. Formally, for combination size k, the overall score of product j for customer i equals

$$s_{ij}^k = \sum_{\mathbf{h} \in H_i^k} \frac{c(\langle \mathbf{h}, j \rangle)}{\sum_{p=1}^J c(\langle \mathbf{h}, p \rangle)}, \tag{19}$$

where the arguments between angle brackets represent a single product combination of size $k+1$.[6] Hence, to obtain product scores s_{ij}^k, by matching customers based on purchase histories that are reduced to combinations of size k, we need the summary of all purchase histories reduced to product combinations of size $k+1$. So, matching customers on pairs of products requires counts over triplets of products as input for the purchase predictions.

The product ranking for each customer is constructed by sorting the products on the product score defined above.[7] This ranking obviously depends on k. In our application we consider collaborative filters with two combination sizes, $k = 1$ and $k = 2$, denoted by CF-1, and CF-2 respectively. Using $k = 1$, customers are matched on the presence of single products in their purchase history. For $k = 2$ customers are matched on the presence of pairs of products in their purchase history. Larger product combinations are not desirable in our application, both in terms of computational feasibility and the degree of sparseness in these larger combinations.

Discrete choice models

Random utility based multinomial choice models (Maddala, 1983; McFadden, 1986) have been extensively used in marketing to model discrete choices from a set of given alternatives. Implementing a traditional discrete choice model that directly uses purchase history data from a large assortment, however, is not feasible. Such a model would have to predict purchases for J products based on J predictor variables, where each predictor variable describes whether a product was purchased by the customer in the past, or not. This model specification would require the simultaneous estimation of $J(J-1)$ parameters, which is infeasible from a computational perspective and will also likely result in identification issues due to sparse data. Hence, traditional discrete choice models do not scale well when the number of products J becomes large.

The benchmark discrete choice model that we propose, resolves these problems by constructing the predictor variables in a smart way, enabling a huge reduction in the

[6]For $k > 0$ it is possible that a product combination $\mathbf{h}$ is never purchased with another product, i.e. for all p we have $\sum_{p=1}^J c(\langle \mathbf{h}, p \rangle) = 0$ in (19). If a customer's purchase history contains such a combination, we regress to a lower value of k for this customer.

[7]In the rare case that two or more products receive the same score, they are ranked according to their order in the data set, which is alphabetic.

number of parameters to be estimated. To get there, we first review the structure of the regular logit model.

In the binary logit model, the probability that customer i purchases product j is specified by:

$$\Pr[y_{in} = j] = \frac{\exp(\theta_{ij})}{1 + \exp(\theta_{ij})}.$$

Here, θ_{ij} represents the log odds of having purchased product j. Ignoring heterogeneity among customers for the moment, these odds will largely be driven by the log of the number of (unique) products purchased by customer i, denoted by u_i, and the relative attractiveness of product j. We capture the relative attractiveness of product j using the log odds of this product based on the *observed* product-purchase frequencies in the purchase data at the customer-base level. This leads to the following expression for the log odds of customer i buying product j:

$$\theta_{ij} = \alpha + \beta \log(u_i) + \gamma \log(\text{odds}_j). \tag{20}$$

The product ranking resulting from this specification will be the same for all customers as the product attractiveness is defined at the customer-base level, not the customer level. To obtain predictions that do differ across customers, we need to introduce heterogeneity in the model. To achieve this without resorting to a model with unobserved heterogeneity, as in LDA(-X) or in MDM, or requiring an excessive number of parameters, as in a regular choice model implementation, we construct variables at the customer-product level that characterize the attractiveness of product j for customer i using the available purchase history data.

The first step is to characterize customers based on their purchase history. We describe each customer's purchases by $\mathbf{v}_i$, a J-dimensional vector containing the proportions of each product in the customer's purchase history, with $\sum_{p=1}^{J} v_{ip} = 1$.[8] We then perform k-means clustering on these proportion vectors using M clusters. Customer heterogeneity can now be characterized by a customer's similarity with respect to each of the cluster means. We define the similarity of customer i with cluster m as:

$$w_{im} = \frac{1}{1 + ||\mathbf{v}_i - \bar{\mathbf{v}}^{(m)}||},$$

where $||\mathbf{v}_i - \bar{\mathbf{v}}^{(m)}||$ measures the Euclidean distance between customer i's proportion vector and the m-th cluster mean $\bar{\mathbf{v}}^{(m)}$.

We can now introduce customer-level heterogeneity in a parsimonious way by combining the cluster-level product attractiveness and the similarity measures w_{im}, that capture the relevance of each cluster for each customer. In particular, we can specify the log odds of customer i purchasing product j as:

$$\theta_{ij} = \alpha + \beta \log(u_i) + \sum_{m=1}^{M} \log(o_{mj})(\gamma_{1m} + \gamma_{2m} w_{im}), \tag{21}$$

[8]For smoothing purposes we add one pseudo observation to each customer's purchase history that is equal to the relative market shares of each product.

where o_{mj} denotes the odds for product j that corresponds to the purchase proportions in cluster mean $\bar{\mathbf{v}}^{(m)}$. Note that in this model specification, the parameters are not product specific, as the relative attractiveness of each product is captured through the summary of the purchase behavior of the various clusters.[9]

Maximum Likelihood estimation of this parsimonious discrete choice model (DCM) is relatively straightforward and including the other available predictor variables is therefore feasible. To do so, we extend the specification in (21) to include interactions with the customer-specific predictor variables in $\mathbf{x}_i$, resulting in:

$$\theta_{ij} = \alpha + \beta \log(u_i) + \sum_{m=1}^{M} \log(o_{mj}) \left(\gamma_{1m} + \gamma_{2m} w_{im} + \sum_{k=1}^{K} x_{ik} (\delta_{1km} + \delta_{2km} w_{im}) \right). \quad (22)$$

2.6. Real-time online predictions

For each of the prediction methods, it is straightforward to construct a product ranking over the assortment for each individual customer. In the context of online retailing it is important to continuously update this ranking based on the customer's new purchases. Re-estimating the (population-level) parameters can be done offline after a substantial amount of new data has been collected. However, updating the predictions for a specific customer should be feasible online. This allows the retailer to update predictions while customers select products during a shopping trip. For all methods, the real-time update step itself consists of simple arithmetic operations with the details provided in Appendix B. A possible bottleneck could be the amount of data that has to be available, retrieved and processed to enable the updates. In the top half of Table 2 we display the number of elements needed in order to update a single customer's product ranking in real-time, for each new product purchase that is observed. The bottom half of the table provides information on the amount of data that needs to be stored for the entire customer base to enable the aforementioned real-time update step.

The first row in Table 2 mimics the context of our application: A medium-sized online retailer with an assortment of 500 products, 10,000 customers, and on average 10 purchases per customer. The number of segments/motivations/clusters (M) is set to 20, which is slightly larger than our empirical findings in this paper, and we consider our implementation of a collaborative filter with combination size $k = 2$. In this context, the number of elements that have to be selected for the real-time update step is of the same order of magnitude across the prediction methods. The storage requirements, on the other hand, are of a different order of magnitude, i.e. millions for the collaborative filter versus thousands for the model-based approaches. However, for these settings all methods can easily be used in practice.

To illustrate the scalability of the various methods we increase the size of the assortment and customer base by a factor of ten and we double both the average purchase history size and M. Naturally, all memory requirements increase in this setting, but the

[9] Model specifications where the coefficients were allowed to be product specific suffered from severe identification problems in our application as the number of parameters is increased by a factor J.

Table 2: Comparison of memory requirements for real-time updating.

No. selected data elements for each real-time update step							
Retailer context							
I	J	n_i	M	LDA(-X)	MDM	CF-2	DCM
10,000	500	10	20	$1.00 \cdot 10^4$	$1.00 \cdot 10^4$	$5.51 \cdot 10^3$	$1.01 \cdot 10^4$
100,000	5,000	20	40	$2.00 \cdot 10^5$	$2.00 \cdot 10^5$	$1.05 \cdot 10^5$	$2.00 \cdot 10^5$
1,000,000	50,000	40	80	$4.00 \cdot 10^6$	$4.00 \cdot 10^6$	$2.05 \cdot 10^6$	$4.00 \cdot 10^6$
No. stored data elements for the real-time update step							
Retailer context							
I	J	N/I	M	LDA(-X)	MDM	CF-2	DCM
10,000	500	10	20	$2.10 \cdot 10^5$	$3.10 \cdot 10^5$	$6.77 \cdot 10^7$	$1.10 \cdot 10^5$
100,000	5,000	20	40	$4.20 \cdot 10^6$	$6.20 \cdot 10^6$	$6.30 \cdot 10^{10}$	$2.20 \cdot 10^6$
1,000,000	50,000	40	80	$8.40 \cdot 10^7$	$1.24 \cdot 10^8$	$6.26 \cdot 10^{13}$	$4.40 \cdot 10^7$

where
I: number of customers
J: assortment size
N: total number of purchases
M: number of segments/motivations/clusters
n_i: number of purchases made by customer i

rate of growth differs significantly. For the collaborative filter the storage requirements increase approximately by a factor of thousand, while the model-based approaches only increase by a factor of twenty. The same holds for the third context, in which we again increase the dimensions. This illustrates that the dimension reduction achieved by the model-based approaches ensures that they are suitable for real-time predictions in large scale applications, even if the number of underlying dimensions grows with the amount of available data. In addition, it is not feasible to use a combination size larger than $k = 2$ in our implementation of a collaborative filter, as in that case the storage requirements would increase even faster. For very large applications, one might even need to rely on the simpler CF-1, which only matches purchase histories on the presence of single products.[10]

2.7. Performance measures

To evaluate the methods for a range of different customization applications, we consider *prediction sets* of different sizes. A prediction set of size S contains the S highest ranked products for a customer. In case one is interested in recommending a single product, the prediction set of size 1 is most relevant. However, when customizing a page with search results the prediction set of size 10 may be more relevant. We assess the quality of a prediction set by matching its contents against hold-out purchase data. These purchases are denoted by $\mathbf{y}'_i$ for customer i and the number of unique purchased products in $\mathbf{y}'_i$ is

[10] In our application, this simpler collaborative filter performs systematically worse than CF-2.

given by u_i'.

We denote a complete ranking of all J products for customer i by the vector $\mathbf{r}_i$. The first element, r_{i1}, is the product that has the highest predicted purchase probability for the model-based rankings, the highest product score for the collaborative filters, and the highest odds for DCM. The quality of a prediction set of size S can be measured by the number of products in the prediction set that overlap with the hold-out purchases: $\sum_{s=1}^{S} \mathrm{I}[r_{is} \in \mathbf{y}_i']$. This number should be seen relative to the maximum number of hits possible in order to obtain a hit rate that may be compared across prediction sets of different sizes. This maximum is bounded by S, the size of the prediction set, and the number of unique hold-out purchases u_i'. Hence, the hit rate for customer i could be defined as: $\sum_{s=1}^{S} \mathrm{I}[r_{is} \in \mathbf{y}_i'] / \min(S, u_i')$.

If a prediction set is presented to a customer in an application, such as a recommendation list, the positions within the set are also of importance (Xu and Kim, 2008). We incorporate this notion in our hit rate by weighing the hits according to their ranks. For the s-th ranked product in a prediction set of size S this weight is specified as: $w(s, S) = 1 - \frac{s-1}{S}$. Combining the above, we obtain our final performance measure, the weighted hit rate:

$$h_i(\mathbf{r}_i, S) = \frac{\sum_{s=1}^{S} \mathrm{I}[r_{is} \in \mathbf{y}_i']\, w(s, S)}{\sum_{s=1}^{\min\left(S, u_i'\right)} w(s, S)}. \tag{23}$$

3. Data

We apply the prediction methods to purchase data from a medium-sized online retailer in the Netherlands.[11] The data starts at the launch of the retailing platform and it covers a period of approximately 67 weeks. The product assortment primarily consists of non-food fast-moving consumer goods, such as detergents, deodorants and shampoo. The assortment is complemented with a small selection of high turnover products for infants and toddlers, such as diapers and baby food. As a consequence, the data contains many repeat purchases.

Initially, the data contains 3,226 unique products IDs. These IDs correspond to a very fine-grained classification, e.g. different package sizes of the same product each receive a unique ID. We opt for a more coarse-grained classification and combine products on the category-brand level. For example, different fragrances of the same deodorant brand are aggregated to one category-brand combination. This approach results in a total of 440 unique category-brand combinations. Additionally, this aggregation step is applied to the customer orders: if an order contains multiple products from the same category-brand, we consider this as a single purchase from this category-brand. Finally, the category-brands that are purchased five times or fewer across all purchases are removed from the data. Below we will simply refer to the category-brand combinations as "products". After the aggregation steps the data contains 95,208 product purchases of 394 products made by 11,783 distinct customers.

[11]The authors wish to thank Christian van Someren, former Managing Director of Truus.nl, for kindly providing us this data.

We chronologically split the data in two parts: The first 80% of the purchases are used as in-sample data, while the hold-out data comprises the last 20% of the purchases. The hold-out data is used to assess the predictive performance of the methods. This division mimics the setting of predicting future purchase behavior. Subsequently, we split the in-sample data into an estimation and a model-selection subset. We randomly select half of the customers from the in-sample data and for each of these customers, a single product purchase is randomly selected as model-selection data. The remaining in-sample data is used to estimate LDA(-X), MDM, DCM, and to create the collaborative filters. The model-selection data is used to determine the number of motivations/segments/clusters (M) in LDA(-X), MDM and DCM respectively. Table 3 summarizes the three subsets of the data, in terms of number of customers, unique products, and number of product purchases.

Table 3: Characteristics of the three subsets of the purchase data.

Subset	Customers	Unique products	Purchases
Full data	11,783	394	95,208
Estimation data	8,831	393	71,346
Model-selection data	4,820	323	4,820
Hold-out data	3,745	369	19,042

It is quite likely that the type of customer acquired by the retailer changes over time, for example due to (a shift in) brand awareness or the mix of advertising channels that are used. Therefore, we investigate whether the customer's time of adoption at the retailer systematically shifts customer preferences. Model-free evidence for such a shift is provided in Table 4, which shows the purchase frequencies of the 10 most frequently purchased products in the estimation data for the first 25% of the estimation customer base, the early adopters, and similarly for the late adopters, the last 25% of the estimation customer base. The ordering of the 10 products for early versus late adopters is not only different, but the relative difference in purchase frequencies is quite substantial as well. For example, the product 'Baby/toddler nutrition – Olvarit' is purchased more than twice as often by early adopters relative to late adopters. In the tail of the assortment such relative shifts may even be larger.

This model-free evidence suggests that the predictive performance could be improved by including customers' time of adoption. We define the time of adoption as the number of days between a customer's first order, and the starting date of the retailing platform. We take the natural logarithm of this variable to allow for larger shifts in the preferences of customers acquired during the early stages of the retailing platform. Finally, this variable is standardized using the mean and variance in the in-sample data.

Table 4: Purchase frequencies of the 10 products that are most frequently purchased in the estimation data by the early and late adopters, respectively.

	Early adopters		Late adopters	
Rank	Products	%	Products	%
1	Diapers – Pampers	9.40	Diapers – Pampers	8.95
2	Baby/toddler nutrition – Nutrilon	5.20	Laundry – Ariel	4.73
3	Baby/toddler nutrition – Olvarit	4.65	Dishwashing – Dreft	3.70
4	Baby care – Zwitsal	3.65	Dental care – Oral-B	3.33
5	Laundry – Ariel	3.41	Baby care – Zwitsal	3.13
6	Paper towels – Page	3.01	Baby care – Pampers	3.00
7	Baby/toddler nutrition – Bambix	3.01	Baby/toddler nutrition – Nutrilon	2.79
8	Baby care – Pampers	2.14	Cleaning – Ambi Pur	2.04
9	Dishwashing – Dreft	2.07	Baby/toddler nutrition – Olvarit	2.02
10	Shaving – Gillette	1.95	Laundry – Lenor	1.98

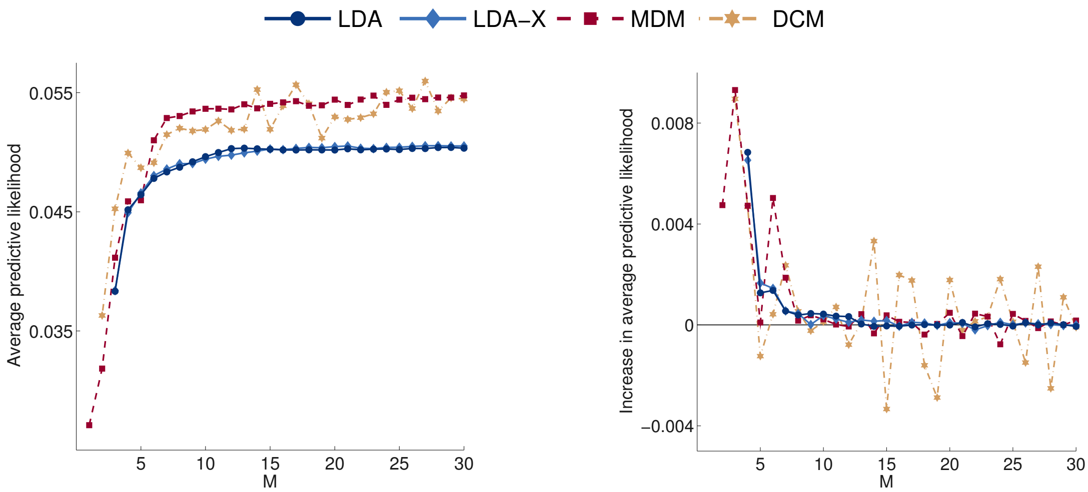
4. Results

In this section we present the results of the prediction methods considered in this paper. First, for LDA(-X), MDM, and DCM we determine M, the number of motivations, segments, and clusters respectively. Next, we focus on some details of the model results to highlight the concepts that underlie LDA(-X) and MDM. In this part we also illustrate how predictions are updated when a new purchase is observed for a customer. Finally, we compare the prediction methods by evaluating their predictive performance on the hold-out data, using the weighted hit rate.

4.1. Model selection

In all model-based approaches we have to determine M: the number of motivations, segments, and clusters. We evaluate LDA(-X) for $M = 3, \ldots, 30$ and MDM for $M = 1, \ldots, 30$, where MDM with $M = 1$ corresponds to the DM model. For each of these model configurations (choice of model plus a value of M) we use 250 different random starts to avoid local maxima. Throughout the estimation procedure the performance of each random start is measured by the average predictive likelihood for the model-selection data and, as discussed in Section 2.3, at several points during the procedure we drop the worst-performing starting values (see Appendix A). At the end of the estimation routine we use the parameter estimates that result from the random start that has the highest average predictive likelihood. We evaluate DCM for $M = 2, \ldots, 30$. To avoid local maxima in the k-means algorithm used in DCM, we use 1000 different random cluster initializations. For each value of M, the clustering that obtains the lowest within cluster sum-of-squares is selected.

The average predictive likelihoods for the model-based approaches are displayed in Figure 1a. We find that for each method the average predictive likelihood steeply in-



(a) (b)

Figure 1: Average predictive likelihood for the model-selection data as a function of M.

creases for the first few values of M and then levels off for larger values of M. This result indicates that choosing M too small likely impedes performance more than choosing M too large. The average predictive likelihood of LDA and LDA-X is similar, reaching a value of approximately 0.05 for the larger values of M. MDM performs slightly better, reaching a value close to 0.055. DCM performs similar and in between LDA(-X) and MDM, although its performance fluctuates across values of M. Note that the average predictive likelihood is merely an indicator for the actual predictive performance in our application, as we will consider the rank assigned to purchased products to evaluate the predictive performance and not the actual purchase likelihoods.

To determine the number of motivations and segments in LDA(-X) and MDM, we select the first value of M for which the average predictive likelihood decreases when M is increased by 1, i.e. we select the first local maximum. As the graphs in Figure 1a stabilize after their first local maximum, this approach results in a parsimonious, yet high performing model specification. Figure 1b shows the differences in performance between subsequent values of M. The first negative value – corresponding to a decrease in performance – is obtained at $M = 14$ for LDA, $M = 16$ for LDA-X, and $M = 12$ for MDM. Hence, we select $M = 13$ for LDA, $M = 15$ for LDA-X, and $M = 11$ for MDM. The average predictive likelihood is more volatile across values of M for DCM, resulting in the first local maximum for $M = 4$. In the spirit of our M selection criterion for LDA(-X) and MDM, we instead select the smallest value of M that corresponds to a local maximum in the range of the values of M where the predictive likelihood has leveled off. For DCM, this happens at $M = 14$.

4.2. Model results for LDA(-X) and MDM

Both LDA(-X) and MDM require quite a large number of parameters to capture the heterogeneity in preferences across the full assortment. For example, to characterize purchase behavior across the segments/motivations the models use $M \times J$ parameters. Clearly it does not make sense to display all these parameters. However, as LDA(-X) and MDM approach heterogeneity in a very different way, it is interesting to consider some differences in the estimation results across the models. In MDM heterogeneity is defined at the customer-segment level, while LDA(-X) models heterogeneity through motivations, i.e. preferences for a set of coherent products, with customers differing in the strength of these motivations. To illustrate this difference, we display the 10 most likely products in the two most likely segments/motivations for each model in Table 5.

Table 5: The 10 most likely products in the two most likely motivations (LDA and LDA-X) or segments (MDM).

LDA	Motivation 1 (Probability 0.21)		Motivation 2 (Probability 0.12)	
M=13	Product	%	Product	%
1	Diapers – Pampers	20.25	Shampoo – Andrelon	3.86
2	Baby/toddler nutrition – Nutrilon	19.13	Paper towels – Page	3.47
3	Baby/toddler nutrition – Olvarit	15.63	Laundry – Ariel	2.94
4	Baby/toddler nutrition – Bambix	9.87	Cleaning – Glorix	2.82
5	Baby care – Zwitsal	7.77	Laundry – Robijn	2.79
6	Baby care – Pampers	4.49	Conditioner – Andrelon	2.40
7	Pacifiers – Bibi	2.17	Shaving – Gillette	2.32
8	Bottle appliances – Philips AVENT	2.03	Deodorant – Dove	2.27
9	Diapers – Huggies	1.56	Baby care – Zwitsal	2.09
10	Bottle appliances – Nuby	1.21	Dishwashing – Dreft	2.05
LDA-X	Motivation 1 (Probability 0.21)		Motivation 2 (Probability 0.13)	
M=15	Product	%	Product	%
1	Diapers – Pampers	20.11	Cleaning – Glorix	5.79
2	Baby/toddler nutrition – Nutrilon	19.26	Paper towels – Page	5.37
3	Baby/toddler nutrition – Olvarit	16.04	Dishwashing – Dreft	3.78
4	Baby/toddler nutrition – Bambix	10.13	Laundry – Robijn	3.54
5	Baby care – Zwitsal	7.94	Cleaning – Ajax	3.50
6	Baby care – Pampers	4.10	Laundry – Ariel	3.27
7	Pacifiers – Bibi	2.13	Disposables – Komo	3.08
8	Bottle appliances – Philips AVENT	2.05	Paper towels – Edet	3.03
9	Diapers – Huggies	1.70	Cleaning – Sorbo	2.97
10	Bottle appliances – Nuby	1.25	Cleaning – Cif	2.29
MDM	Segment 1 (Probability 0.32)		Segment 2 (Probability 0.23)	
M=11	Product	%	Product	%
1	Diapers – Pampers	11.35	Diapers – Pampers	16.23
2	Laundry – Ariel	4.05	Baby/toddler nutrition – Nutrilon	14.34
3	Baby care – Zwitsal	4.03	Baby/toddler nutrition – Olvarit	11.76
4	Baby/toddler nutrition – Nutrilon	3.50	Baby/toddler nutrition – Bambix	7.08
5	Baby/toddler nutrition – Olvarit	3.37	Baby care – Zwitsal	6.50
6	Baby care – Pampers	3.14	Baby care – Pampers	4.05
7	Dishwashing – Dreft	3.12	Bottle appliances – Philips AVENT	1.82
8	Paper towels – Page	3.01	Laundry – Ariel	1.70
9	Dental care – Oral-B	2.60	Pacifiers – Bibi	1.64
10	Baby/toddler nutrition – Bambix	2.20	Diapers – Huggies	1.44

The top 10 most likely products are primarily baby related for the largest as well as the second largest segment in MDM. Additionally, there is much overlap at the product level: 7 products appear in both top 10 lists. For LDA and LDA-X the largest motivation relates to baby products and the order of the top 10 is the same, with only minor differences between the purchase probabilities. The second motivation for LDA-X is driven by cleaning products, while in LDA it is a mix of cleaning and personal care products (and one baby related product). So, for both LDA and LDA-X the second motivation is very different from the first, which contrasts with the results for MDM.

This difference can be explained by the distinction between a motivation and a segment. Motivations represent coherent sets of products, where customers can be interested in multiple of these sets. Segments capture the purchase behavior of groups of customers, and purchase behavior across groups likely overlaps. In other words, the motivations in LDA(-X) correspond to a clustering on the product level, whereas the segments in MDM represent a clustering on the customer level.

As the models differ substantially in terms of the underlying data structures that are captured, their predictions are also likely to be different. We investigate these differences in a hypothetical scenario. First, let us consider a customer with average customer characteristics who is new to the store, i.e. without previous observed purchases. Each model approximately yields the marginal distribution as predictive distribution for this customer. The top 5 products in the marginal distribution of the estimation data are displayed in Table 6.

Table 6: Purchase frequencies of the 5 products that are most frequently purchased in the estimation data.

Rank	Product	Frequency
1	Diapers – Pampers	9.20 %
2	Baby/toddler nutrition – Nutrilon	4.09 %
3	Laundry – Ariel	4.07 %
4	Dishwashing – Dreft	3.65 %
5	Baby/toddler nutrition – Olvarit	3.47 %

Next suppose that the customer purchases ‘Shampoo – Herbal Essences’. For each model the updated top 5, conditional on this purchase, is displayed in Table 7. Indeed, each model now provides a different ranking. It is interesting to focus on the new rank of the shampoo itself and the complementary conditioner of the ‘Herbal Essences’ brand. In the marginal distribution the shampoo and conditioner are ranked 113 and 119, respectively. Conditional on the purchase of the shampoo, these two products reach the top 5 in LDA (they get rank 3 and 2). For LDA-X the products do not occur in the top 5 but receive rank 17 and 16. Finally, in MDM the rank of the shampoo shifts to 26, while the rank of the conditioner barely changes and reaches only 117. This indicates that MDM fits the observed purchase well, but is hardly able to discover that the conditioner is a complement to the shampoo.

Table 7: Purchase probabilities of the 5 most likely product for each model, conditioned on the purchase of a single product.

	Purchased product: **Shampoo – Herbal Essences**	
LDA	Product	Probability
1	Diapers – Pampers	0.05
2	Conditioner – Herbal Essences	0.05
3	Shampoo – Herbal Essences	0.05
4	Baby/toddler nutrition – Nutrilon	0.02
5	Paper towels – Page	0.02
LDA-X	Product	Probability
1	Diapers – Pampers	0.05
2	Paper towels – Page	0.04
3	Laundry – Ariel	0.03
4	Dishwashing – Dreft	0.03
5	Baby care – Zwitsal	0.03
MDM	Product	Probability
1	Diapers – Pampers	0.09
2	Baby/toddler nutrition – Nutrilon	0.04
3	Baby care – Zwitsal	0.03
4	Laundry – Ariel	0.03
5	Paper towels – Page	0.03

4.3. Predictive performance

To assess a method's predictive performance we evaluate its weighted hit rate for the hold-out data, see (23). In the weighted hit rate, each hit receives a weight that depends on the rank assigned to the prediction. A better (numerical lower) rank receives a larger weight than a worse (numerical higher) rank. Figure 2 presents the predictive performance on the complete hold-out data for the model-based approaches, LDA(-X), MDM, DCM, and the two count-based collaborative filters, CF-1, and CF-2.

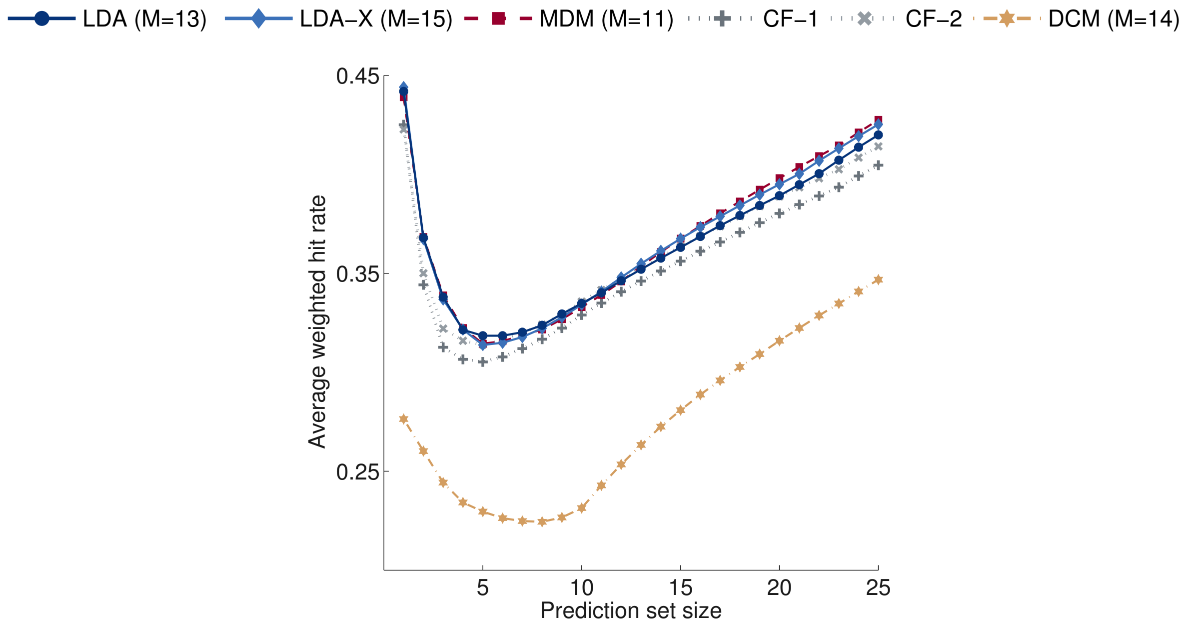


Figure 2: Predictive performance for the complete hold-out data, as a function of prediction set size.

In case we predict only a single product for each customer, i.e. a prediction set of size one, LDA-X has the best performance with a hit rate close to 0.45. For most prediction set sizes, LDA(-X) and MDM outperform the collaborative filters. The best performing collaborative filter is CF-2, which matches customers on the presence of pairs of products in their purchase history.

Given the decent predictive likelihoods generated by DCM (see Figure 1), it has an unexpected poor performance in terms of ranking the purchased products.

Note that the average hit rate declines for the first few prediction set sizes. This is a direct consequence of the denominator in the definition of the hit rate in (23), which divides the total number of hits by the maximum number of hits possible for a given customer and prediction set size. This number increases with the size of the prediction set until it reaches the number of unique products purchased by the customer. As the average number of unique purchases per customer in the hold-out data is almost 5, we indeed see the hit rates increase beyond that value for most methods.

We study the difference in performance for the prediction methods in more detail by separately considering specific groups of customers and products. In particular, we first divide the customers in the hold-out data into three groups based on the number of purchases in the estimation data: (i) 2185 customers with no prior observed purchases (Figure 3); (ii) 809 customers with a moderate amount (1-9) of purchases (Figure 4); and (iii) 751 customers with many (10 or more) purchases (Figure 5).

The most apparent difference in performance between these groups is visible in the range of the y-axis. If we observe many purchases for a customer the average hit rates are twice as large for the smaller prediction sets, compared to those for customers with no purchases in the estimation data. This is exactly according to our expectations, and provides empirical evidence that purchase history data is indeed very informative about a customer's future purchases.

By examining Figure 3 we see that for customers without previous purchases the collaborative filters perform very well (particularly for moderate-sized prediction sets). Note that for this specific group of customers the collaborative filters rank the products according to their market penetration in the customer base. Also for LDA and MDM there is no information that can be used to make a personalized prediction. LDA-X uses the time of adoption, although this does not seem to shift the baseline predictions a lot. Hence, the performance differences between LDA(-X) and MDM are small.

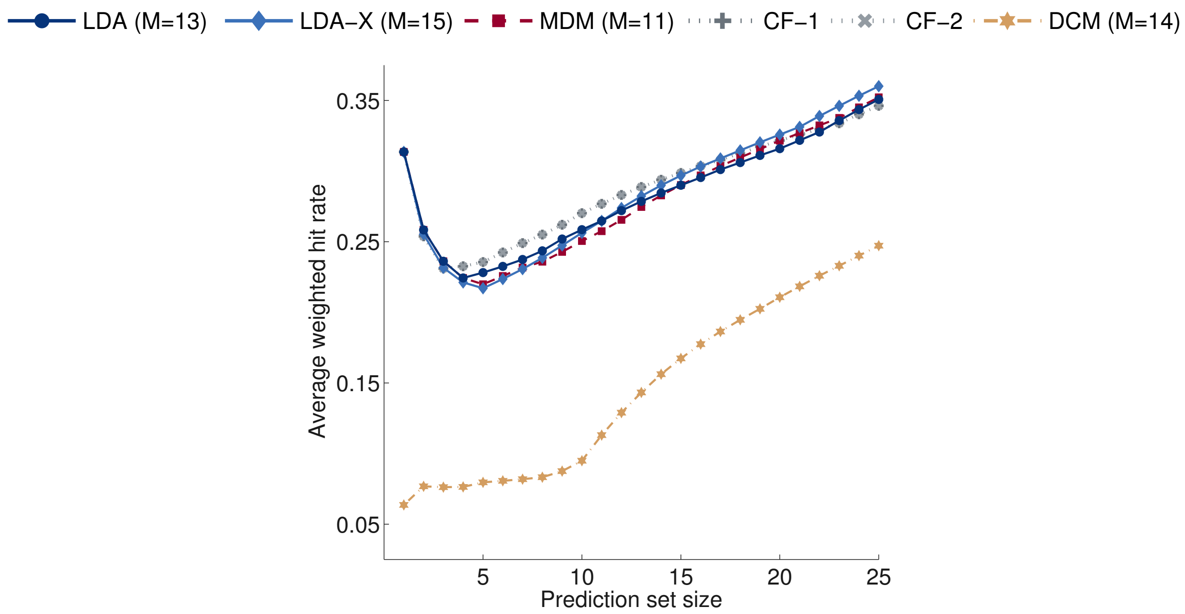


Figure 3: Predictive performance for the customers with no purchases in the estimation data.

In the absence of a purchase history, the similarity of a customer to each of the M clusters, used to create predictor variables in DCM, is rather meaningless. As a result, the DCM's predictive power is low for these customers. In fact, a large part of the performance gap on the complete hold-out data between DCM and the other prediction methods is driven by the poor performance for the group of customers without a purchase history.

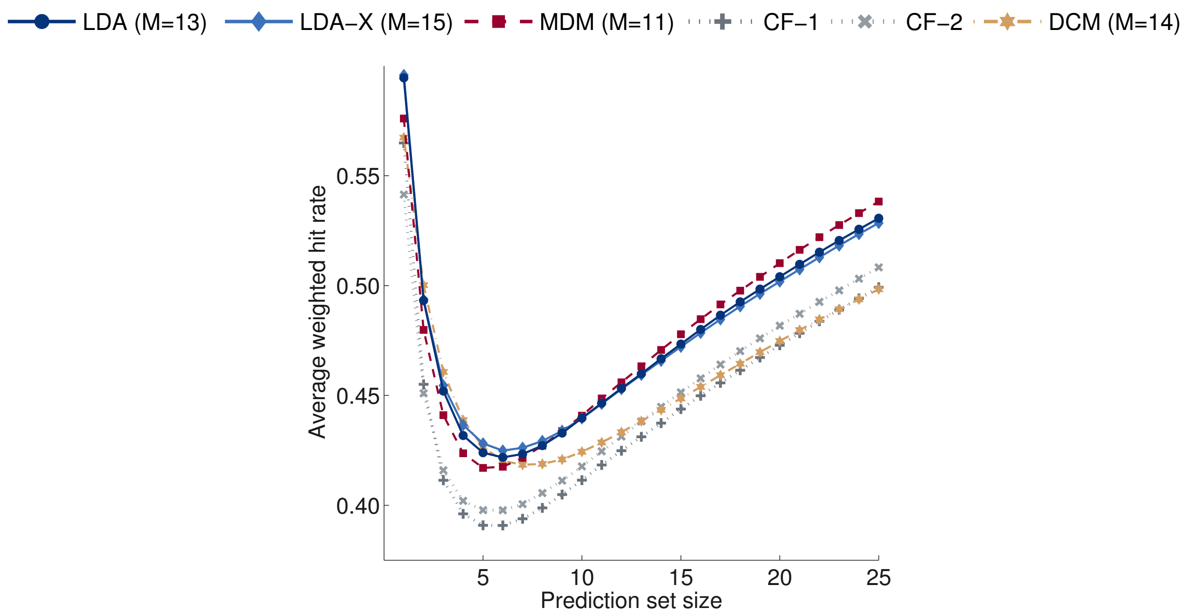


Figure 4: Predictive performance for the customers with a few purchases (1-9) in the estimation data.

We observe a different pattern for customers with a moderate number of past purchases in Figure 4, where LDA(-X) and MDM consistently outperform the collaborative filters. This indicates that these model-based methods are better able to learn from a customer's previous purchases than the collaborative filters. Comparing the methods, LDA(-X) attains the highest overall performance and performs best when we predict only a single product, while MDM performs better for larger prediction sets. The performance of DCM is competitive for the smaller prediction sets, although its relative performance drops substantially for larger prediction set sizes.

The final group of customers that we consider consists of those who made many purchases, displayed in Figure 5. The general conclusion is similar to that of the customers with a moderate number of purchases. However, in this case MDM obtains the highest performance for prediction sets that contain more than one product. This result, combined with the previous findings, may be explained by the flexibility of the customer-level heterogeneity structure. In MDM preferences are modeled by a customer-specific probability vector over the product assortment. On the other hand, in LDA(-X) a customer's individual preferences are described by a lower-dimensional probability vector over the M motivations. Both models learn from previous purchases, but in MDM this learning is directly incorporated in the preferences over the assortment, while in LDA(-X) it is done indirectly through the probabilities for the motivations. As a consequence, MDM

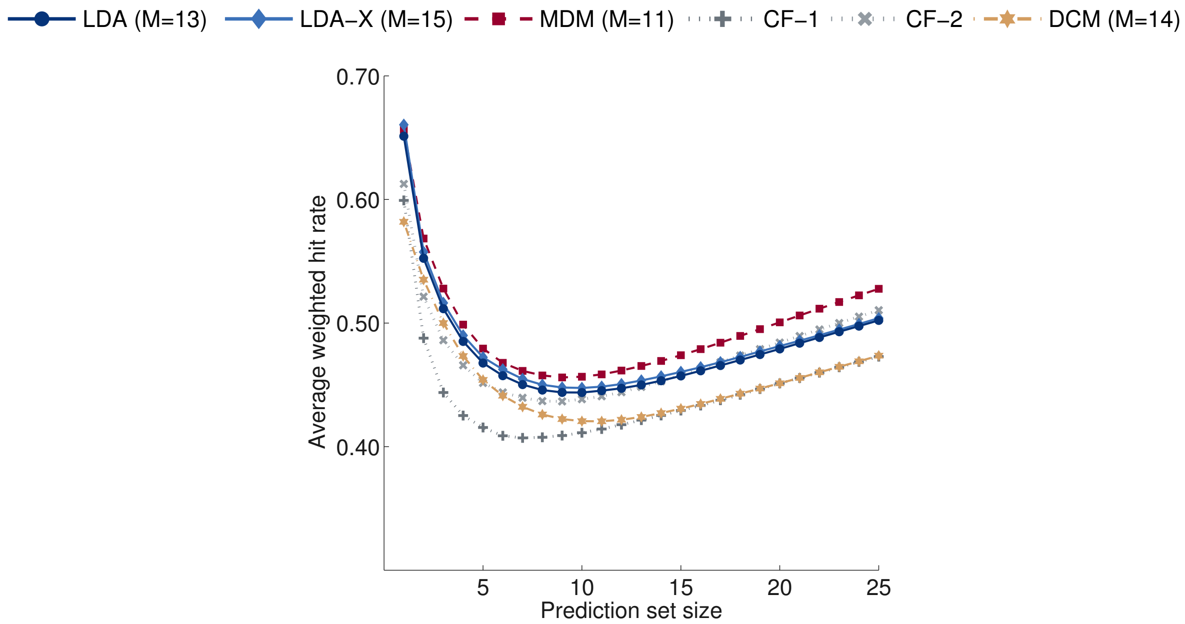


Figure 5: Predictive performance for the customers with many purchases (10 or more) in the estimation data.

has more degrees of freedom at the level of the individual customer as the assortment size J is much larger than the number of motivations M. This additional flexibility turns out to pay off when many purchases are observed for a customer.

The results above highlight the performance of the methods for the complete assortment. However, many of the highly-ranked products are products that are frequently purchased, or products that have been previously purchased by the focal customer. Customers can easily anticipate such recommendations and might even be bored by them (Fleder and Hosanagar, 2009). It is therefore interesting to evaluate the performance of the methods when predicting products that may be more *unexpected*.

To assess the performance of the methods for predicting such unexpected products, we evaluate the predictive performance for a restricted subset of the product assortment. This subset is constructed as follows: First, we remove 20% of the products in the assortment that are most frequently purchased in the estimation data. Second, we create a customer-specific restriction by removing the products that have previously been purchased by this customer. Subsequently, for each individual customer, we only consider the predictions and hold-out purchases for products that are contained in this restricted subset of the assortment. As customers are less likely to be aware of these products, performing well on this aspect could potentially increase the *cross-selling* performance of marketing actions that are based on such predictions.

The predictive performance for the restricted set of products is displayed in Figure 6. LDA and MDM perform better than the collaborative filters and DCM, but LDA-X

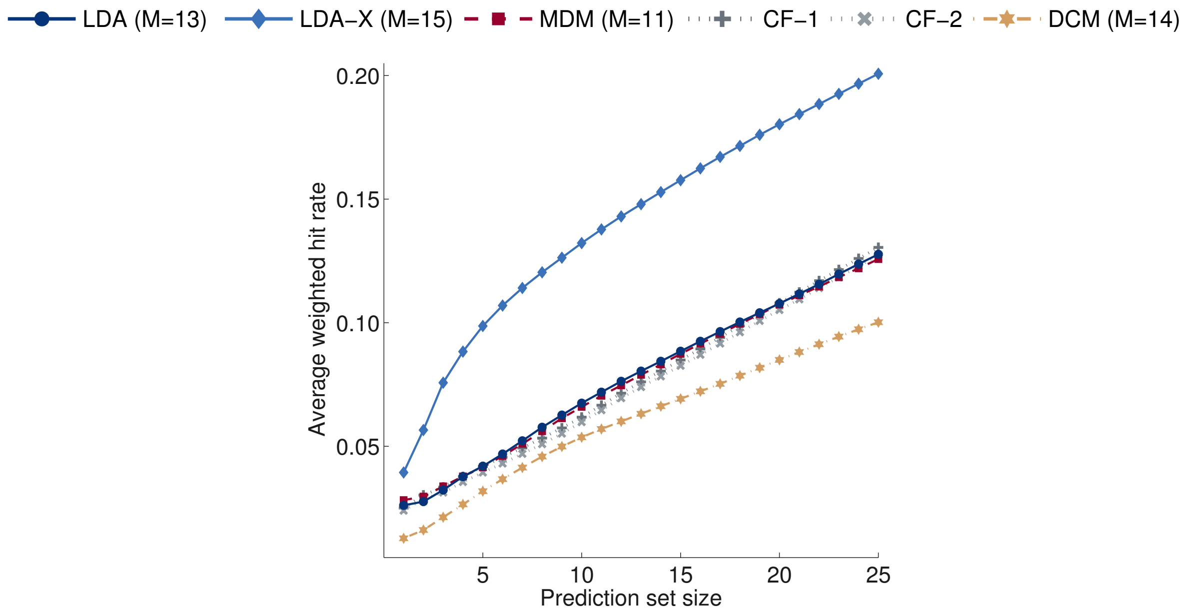


Figure 6: Predictive performance for the restricted subset of the hold-out data.

clearly outperforms all the other prediction methods. This remarkable performance difference primarily arises for the highly-ranked products. By examining these products, we find that the product 'Slimming nutrition - Weight Care' appears in the top of many of the LDA-X customer-level prediction sets. The prediction sets resulting from the other methods, however, do not contain this product. In fact, it turns out that 'Slimming nutrition - Weight Care' is the most frequently purchased product in the hold-out data. Its purchase frequency has shifted from 0.04% in the estimation data to 4.88% in the hold-out data. LDA-X is able to capture this shift through the time of adoption variable.[12] This shows that the inclusion of predictor variables has merit in the context of purchase prediction, even though the time of adoption variable in general does not add much explanatory power. The reason why we do not see a similar shift for DCM can be explained by the way the predictor variables enter the model. In LDA-X, it directly influences the likelihood of a certain motivation, in effect being able to boost a motivation that is relevant for customers who adopted later in time. In this case, it boosts the motivation that contains products that are purchased more frequently later in the observation period, including the period of the hold-out predictions. In contrast, in DCM the clusters are determined 'outside' the model, using the k-means algorithm. The performance of the clustering algorithm does not benefit from selecting a cluster that is linked to the other prediction variables, as the predictor variables are not included when constructing the clusters. In the absence of such clusters of customers, inclusion

[12]We acknowledge that there can be many external influences that drive this shift in purchase behavior. Our predictor variable (time of adoption) most likely serves as a proxy for the actual causes.

of the predictor variables cannot shift the importance of these products, as they are not contained in a separate cluster.

5. Conclusion

In this paper we have evaluated several methods for purchase prediction in large assortments using purchase history data. Inspired by the text modeling literature, we have introduced a novel model-based approach that uses latent Dirichlet allocation (LDA(-X)) to predict purchases. In addition, we have considered mixtures of Dirichlet-Multinomials (MDM), a framework well known in the brand-choice modeling literature. The performance of these model-based approaches has been contrasted against two benchmarks: a set of count-based collaborative filters, in which customers are matched on the contents of their purchase history, and a scalable implementation of a discrete choice model (DCM), that does not break down when used with a large product assortment. All methods are able to construct customer-specific product rankings over the assortment that can be used for purchase prediction.

Naturally, the prediction methods differ in their heterogeneity assumptions, estimation complexity, and memory requirements. In MDM purchase heterogeneity is specified at the customer level by segmenting the customer base. In LDA(-X), on the other hand, this heterogeneity is specified at the motivation level, which groups products, not customers. These heterogeneity assumptions also affect the estimation complexity of the models. MDM has more flexibility to model an individual customer's purchase behavior than LDA(-X), but this comes at the price of increased estimation complexity as more parameters have to be estimated. The estimation complexity of the logit part of the DCM is relatively low, but it does depend on customer clusters from an external method (i.e. the k-means algorithm). The collaborative filter has as advantage that no (latent) model structure has to be estimated, but its storage requirements for generating real-time online predictions rapidly increase for large applications. In contrast, the model-based approaches require less storage and additionally this grows much slower with the size of the application.

The performance of the methods was assessed based on purchase prediction sets derived from the product rankings, and comparing these sets to actual hold-out purchases. In general, LDA(-X) and MDM perform best and, even though these two models are conceptually rather different, their predictive performance is comparable. In addition, we have considered the setting where we focus on the predictive performance for products in the tail of the assortment that have not been purchased yet by the customer. In this case LDA-X clearly outperforms the other methods, which can be attributed to the time of adoption variable that is included in LDA-X. Although DCM also includes this predictor variable, its dependence on the k-means algorithm prevents it from effectively using the additional information to generate better predictions.

In summary the LDA(-X) prediction method that we have introduced in this paper is the most promising approach to purchase prediction, particularly in the context of large online retailers. Its predictive performance is very competitive compared to the other methods and it scales well with the size of the application. Finally, it is a self-contained prediction method that can readily accommodate additional information available to the retailer. In our application we only had access to a fairly weak predictor, but the potential benefits of including stronger predictors of customer preferences into the model could be large.

To conclude, LDA(-X) can be readily used as a stepping stone for further model-based research that quantifies and optimizes the impact of marketing interventions in large-scale retailing environments. For example, one could optimize a recommendation system that differentiates between the likelihood of purchasing a product and the added benefits from recommending that product (Bodapati, 2008; Wagner and Taudes, 1986); something that is difficult to implement in a count-based method such as a collaborative filter. We obviously consider such extensions an interesting avenue for further research.

A. Estimation details for LDA(-X) and MDM

In this appendix we present the estimation details for LDA(-X) and MDM. First, we discuss our random start routine, aimed at minimizing the risk of ending up in locally optimal solutions. Second, we present the conditional posterior distributions that are used in the MCMC samplers. Finally, at the end of this appendix we provide a high-level description of the inference algorithm for LDA(-X) in pseudocode.

A.1. Random start routine

LDA(-X) and MDM are both members of the general class of mixture models. This class of models is well known to be susceptible to end up in an area around a local maximum of the posterior distribution. We reduce this risk by considering multiple random starts. For MDM a random start is an initialization of the segment assignments $\mathbf{s}$, while in LDA(-X) it is an initialization of the motivation assignments $\mathbf{Z}$.

For each model, we initially consider 250 different random starts. For each of these starts we draw 1,000 samples using our MCMC methodology. These samples are used to infer each customer's posterior predictive distribution and to calculate the average predictive likelihood of the model-selection data. The 50 starts that obtain the highest average predictive likelihood are selected. For these starts, we repeat the above procedure and next select the 15 best performing starts. Again, we repeat the procedure but this time draw 20,000 samples. Finally, we continue with the random start that obtains the highest average predictive likelihood.

The 22,000 draws that are generated within the random start routine for the single remaining model are considered as the burn-in period of the chain. For this selected random start we finally draw another 10,000 samples. We thin this chain by selecting every tenth draw, resulting in 1,000 posterior samples.

A.2. Conditional posterior distributions

In this section we present the details of our MCMC sampler. For each sampling step in each model, we present the corresponding conditional posterior distribution. In the presentation below we use the notation superscript $^{\setminus n}$ to indicate that the n-th element is excluded from a vector, matrix, or set. A general density function is denoted by $p()$, while we use $\pi()$ in case the density corresponds to a prior distribution in which the parameters are fixed. The probability density function of the standard normal distribution is denoted by $\phi()$ and the Gamma function is denoted by $\Gamma()$. Finally, in LDA-X we replace $\boldsymbol{\gamma}$ and $\{\boldsymbol{\delta}_l\}_{l=1}^{M}$ by $\{\boldsymbol{\alpha}_i\}_{i=1}^{I}$, whenever this simplifies notation.

As the derivations in this appendix rely heavily on the Dirichlet-Multinomial distribution, we first provide its density in terms of Gamma functions. The Dirichlet-Multinomial distribution corresponds to a data generating process where first a probability vector $\boldsymbol{\theta} \sim \text{Dirichlet}(\boldsymbol{\alpha})$ is generated and subsequently, this vector is used to generate a set of Categorical random variables, denoted by $\mathbf{z}$. The marginal density of $\mathbf{z}$

in terms of $\boldsymbol{\alpha}$ is called the Dirichlet-Multinomial distribution. This density is given by:

$$\begin{aligned} p(\mathbf{z}|\boldsymbol{\alpha}) &= \int_{\boldsymbol{\theta}} p(\mathbf{z}|\boldsymbol{\theta}) p(\boldsymbol{\theta}|\boldsymbol{\alpha}) d\boldsymbol{\theta} \\ &= \frac{\Gamma\left(\sum_{l=1}^{M} \alpha_l\right)}{\Gamma\left(\sum_{l=1}^{M} \alpha_l + c_l^{\mathrm{M}}\right)} \prod_{m=1}^{M} \frac{\Gamma\left(\alpha_m + c_m^{\mathrm{M}}\right)}{\Gamma\left(\alpha_m\right)}, \end{aligned} \tag{24}$$

where c_m^{M} is the number of elements in $\mathbf{z}$ that are equal to m and M gives the number of categories.

A.2.1. LDA

The joint density for the collapsed LDA model can be written as

$$p(\mathbf{Y}, \mathbf{Z}, \beta_0, \boldsymbol{\alpha}) \propto p(\mathbf{Y}|\mathbf{Z}, \beta_0) p(\mathbf{Z}|\boldsymbol{\alpha}) \pi(\beta_0, \boldsymbol{\alpha}). \tag{25}$$

In our implementation of LDA we impose $\beta_0 \sim \mathrm{logN}(\mu_{\beta_0}, \sigma^2_{\beta_0})$ and $\alpha_m \sim \mathrm{logN}(\mu_\alpha, \sigma^2_\alpha)$. The prior distributions, combined with the LDA model specification, define the complete joint distribution in (25). The MCMC sampler for this model contains Gibbs steps for all the separate elements of $\mathbf{Z}$ and Metropolis-Hastings steps for β_0 and the elements of $\boldsymbol{\alpha}$.

The conditional posterior probability that $z_{in} = m$, i.e. that the n-th purchase of customer i is driven by motivation m, is proportional to:

$$\begin{aligned} &\Pr\left[\, z_{in} = m | y_{in} = j, \mathbf{Z}^{\setminus in}, \beta_0, \boldsymbol{\alpha}, \mathbf{Y}^{\setminus in} \,\right] \\ &\quad \propto \Pr\left[\, z_{in} = m | \mathbf{Z}^{\setminus in}, \boldsymbol{\alpha} \,\right] \Pr\left[\, y_{in} = j | z_{in} = m, \mathbf{Z}^{\setminus in}, \beta_0, \mathbf{Y}^{\setminus in} \,\right] \\ &\quad \propto \left(\alpha_m + c_{im}^{\mathrm{IM}\setminus in}\right) \frac{\beta_0 + c_{mj}^{\mathrm{MJ}\setminus in}}{J\beta_0 + \sum_{p=1}^{J} c_{mp}^{\mathrm{MJ}\setminus in}}, \end{aligned} \tag{26}$$

where $c_{mj}^{\mathrm{MJ}\setminus in}$ is the number of times a purchase of product j is driven by motivation m and $c_{im}^{\mathrm{IM}\setminus in}$ is the number of purchases made by customer i that are driven by motivation m, excluding z_{in} and y_{in}. This result can straightforwardly be used to obtain samples for $\mathbf{Z}$.

The conditional posterior density of β_0 is given by

$$\begin{aligned} p(\beta_0|\mathbf{Z}, \boldsymbol{\alpha}, \mathbf{Y}) &\propto \pi(\beta_0) p(\mathbf{Y}|\mathbf{Z}, \beta_0) \\ &\propto \pi(\beta_0) \prod_{l=1}^{M} \frac{\Gamma\left(J\beta_0\right)}{\Gamma\left(J\beta_0 + \sum_{p=1}^{J} c_{lp}^{\mathrm{MJ}}\right)} \prod_{p=1}^{J} \frac{\Gamma\left(\beta_0 + c_{lp}^{\mathrm{MJ}}\right)}{\Gamma\left(\beta_0\right)}. \end{aligned} \tag{27}$$

As (27) results in a non-standard density, we use a random walk Metropolis-Hastings step to obtain samples for β_0. Candidate values are generated from $\mathrm{logN}(\beta_0, s^2_{\beta_0})$, where

β_0 denotes the current value of the parameter and the variance $s^2_{\beta_0}$ is calibrated during the start value selection procedure such that we obtain an acceptance rate of about 50%.

The conditional posterior density of α_m is

$$\begin{aligned} p(\alpha_m|\boldsymbol{\alpha}^{\setminus m}, \mathbf{Z}, \beta_0, \mathbf{Y}) &\propto \pi(\alpha_m) p(\mathbf{Z}|\boldsymbol{\alpha}) \\ &\propto \pi(\alpha_m) \prod_{i=1}^{I} \frac{\Gamma\left(\sum_{l=1}^{M} \alpha_l\right)}{\Gamma\left(\sum_{l=1}^{M} \alpha_l + c_{il}^{\mathrm{IM}}\right)} \left(\frac{\Gamma\left(\alpha_m + c_{im}^{\mathrm{IM}}\right)}{\Gamma\left(\alpha_m\right)}\right). \end{aligned} \tag{28}$$

Again this is a non-standard density and the same type of random walk Metropolis-Hastings step as before is used to obtain samples for α_m.

A.2.2. LDA-X

LDA-X extends LDA by allowing customer-specific predictor variables $\mathbf{X}$ to affect the motivation probabilities. The collapsed joint density for the LDA-X model can be rewritten as

$$p(\mathbf{Y}, \mathbf{Z}, \beta_0, \boldsymbol{\gamma}, \{\boldsymbol{\delta}_l\}_{l=1}^{M}) \propto p(\mathbf{Y}|\mathbf{Z}, \beta_0) p(\mathbf{Z}|\{\boldsymbol{\alpha}_i\}_{i=1}^{I}) \pi(\beta_0, \boldsymbol{\gamma}, \{\boldsymbol{\delta}_l\}_{l=1}^{M}), \tag{29}$$

where $\alpha_{im} = \exp(\gamma_m + \mathbf{x}_i' \boldsymbol{\delta}_m)$, $\gamma_m \sim \mathrm{N}(\mu_\gamma, \sigma^2_\gamma)$, and $\delta_{mk} \sim \mathrm{N}(\mu_\delta, \sigma^2_\delta)$.

The MCMC sampler for LDA-X includes a Gibbs step for every element of $\mathbf{Z}$ and random walk Metropolis-Hastings steps for β_0 and all elements of $\boldsymbol{\gamma}$ and $\{\boldsymbol{\delta}_l\}_{l=1}^{M}$. Considering the relation $\alpha_{im} = \exp(\gamma_m + \mathbf{x}_i' \boldsymbol{\delta}_m)$, it is easy to see that we obtain the conditional posterior distributions for the elements of $\mathbf{Z}$ by writing α_{im} instead of α_m in (26). The conditional posterior for β_0 is exactly the same as in (27).

The conditional posterior density of δ_{mk} equals

$$\begin{aligned} p(\delta_{mk}|\boldsymbol{\delta}_m^{\setminus k}, \mathbf{Z}, \beta_0, \boldsymbol{\gamma}, \{\boldsymbol{\delta}_l\}_{l \neq m}, \mathbf{Y}, \mathbf{X}) &\propto \pi(\delta_{mk}) \prod_{i=1}^{I} p(\mathbf{z}_i|\boldsymbol{\alpha}_i) \\ &\propto \pi(\delta_{mk}) \prod_{i=1}^{I} \frac{\Gamma\left(\sum_{l=1}^{M} \alpha_{il}\right)}{\Gamma\left(\sum_{l=1}^{M} \alpha_{il} + c_{il}^{\mathrm{IM}}\right)} \frac{\Gamma\left(\alpha_{im} + c_{im}^{\mathrm{IM}}\right)}{\Gamma\left(\alpha_{im}\right)}, \end{aligned} \tag{30}$$

where δ_{mk} influences the likelihood through α_{im}. A random walk Metropolis-Hastings step in the MCMC sampler is used to obtain samples for $\{\boldsymbol{\delta}_l\}_{l=1}^{M}$. Candidate values are obtained from $\mathrm{N}(\delta_{mk}, s^2_{\delta_{mk}})$, where δ_{mk} denotes the current value of the parameter and the variance $s^2_{\delta_{mk}}$ is calibrated during the start value selection procedure such that we obtain an acceptance rate of about 50%. The Metropolis-Hastings sampler for γ_m can be derived in an analogous way.

A.2.3. MDM

The joint collapsed density for the MDM model may be rewritten as

$$p(\mathbf{Y}, \mathbf{s}, \{\boldsymbol{\beta}_l\}_{l=1}^{M}) \propto p(\mathbf{Y}|\mathbf{s}, \{\boldsymbol{\beta}_l\}_{l=1}^{M}) \pi(\{\boldsymbol{\beta}_l\}_{l=1}^{M}) \int_{\boldsymbol{\pi}} p(\mathbf{s}|\boldsymbol{\pi}) \pi(\boldsymbol{\pi}) d\boldsymbol{\pi}, \tag{31}$$

where the priors are given by $\boldsymbol{\pi} \sim \text{Dirichlet}(1,\ldots,1)$ and $\beta_{mj} \sim \text{logN}(\mu_\beta, \sigma_\beta^2)$. As is clear from the notation we integrate over the prior of distribution $\boldsymbol{\pi}$. The prior distributions, combined with the MDM model specification, define the complete joint distribution in (31). In our MCMC sampler we use separate Gibbs sampling steps for all segment assignments in $\mathbf{s}$ and Metropolis-Hastings sampling steps for the elements of $\{\boldsymbol{\beta}_l\}_{l=1}^M$.

The conditional posterior probability that $s_i = m$, i.e. that customer i is allocated to segment m, is

$$\begin{aligned}\Pr\Big[\, s_i = m|\mathbf{s}^{\setminus i}, \{\boldsymbol{\beta}_l\}_{l=1}^M, \mathbf{Y} \Big] \\ &\propto \Pr\big[\, s_i = m|\mathbf{s}^{\setminus i}\big]\, p(\mathbf{y}_i|s_i = m, \boldsymbol{\beta}_m) \\ &\propto \left(1 + c_m^{\mathrm{M}\setminus i}\right) \frac{\Gamma\left(\sum_{p=1}^J \beta_{mp}\right)}{\Gamma\left(\sum_{p=1}^J \beta_{mp} + c_{ip}^{\mathrm{IJ}}\right)} \prod_{p=1}^J \frac{\Gamma\left(\beta_{mp} + c_{ip}^{\mathrm{IJ}}\right)}{\Gamma\left(\beta_{mp}\right)},\end{aligned} \tag{32}$$

where c_{ip}^{IJ} is the number of times customer i purchased product p and $c_m^{\mathrm{M}\setminus i}$ denotes the number of customers allocated to segment m, excluding customer i. Equation (32) implies probabilities that can straightforwardly be used to obtain samples for s_i.

The conditional posterior density of β_{mj} is given by

$$\begin{aligned}p(\beta_{mj}|\boldsymbol{\beta}_m^{\setminus j}, \mathbf{s}, \{\boldsymbol{\beta}_l\}_{l\neq m}, \mathbf{Y}) \\ &\propto \pi(\beta_{mj}) \prod_{i=1}^I p(\mathbf{y}_i|s_i = m, \boldsymbol{\beta}_m)^{\mathbb{I}[s_i=m]} \\ &\propto \pi(\beta_{mj}) \prod_{i=1}^I \left(\frac{\Gamma\left(\sum_{p=1}^J \beta_{mp}\right)}{\Gamma\left(\sum_{p=1}^J \beta_{mp} + c_{ip}^{\mathrm{IJ}}\right)} \frac{\Gamma\left(\beta_{mj} + c_{ij}^{\mathrm{IJ}}\right)}{\Gamma\left(\beta_{mj}\right)} \right)^{\mathbb{I}[s_i=m]}.\end{aligned} \tag{33}$$

As (33) clearly results in a non-standard density we use a random walk Metropolis-Hastings step in the MCMC sampler to obtain samples for $\{\boldsymbol{\beta}_l\}_{l=1}^M$. Candidate values are obtained from $\text{logN}(\beta_{mj}, s_{\beta_{mj}}^2)$, where β_{mj} denotes the current value of the parameter and the variance $s_{\beta_{mj}}^2$ is calibrated during the start value selection procedure such that we obtain an acceptance rate of about 50%.

A.3. Pseudocode for LDA(-X)

In this section we provide pseudocode for the inference algorithm for LDA(-X). Algorithm 1 contains a high-level description of the inference algorithm for LDA(-X). More detailed pseudocode for our initialization procedure, sampling, and calibration of the Metropolis-Hastings proposal variances can respectively be found in Algorithms 2, 3, and 4. The pseudocode depends on the implementation details of our random start routine, discussed in Appendix A.1, as well as the conditional posterior distributions for LDA(-X) presented in Appendix A.2. The target acceptance rate for all univariate Metropolis-Hastings samplers is set to 50%.

Algorithm 1 Pseudocode for LDA(-X)

Q: number of estimation rounds
$K(q)$: set of random starts in round q
$T(q)$: number of samples to be drawn in round q
for each estimation round $q = 1, \ldots, Q$ **do**
    for each random start k in $K(q)$ **do**
        if q is the first round **then** initialize the k-th random start
            INITIALIZATION (Algorithm 2)
        end if
        for $t = 1, \ldots, T(q)$ **do**
            // Sample a new state from the MCMC chain
            SAMPLING (Algorithm 3)
            for each parameter sampled with a Metropolis-Hastings step **do**
                CALIBRATION (Algorithm 4)
            end for
        end for
        Calculate the average predictive likelihood of the model-selection data over the last $T(q)$ states
    end for
    if q is not the last round **then**
        Select the random starts with the highest average predictive likelihood in this round for $K(q+1)$
    end if
end for

Algorithm 2 Initialization of a random start in LDA(-X)

N: number of purchases in the estimation data
procedure INITIALIZATION
  // Set the initial model parameters
  $\beta_0 = 0.01$
  if the model is LDA **then**
    $\alpha_m = \frac{1}{M}$ for $m = 1, \ldots, M$
  else if the model is LDA-X **then**
    $\gamma_m = \log \frac{1}{M}$ for $m = 1, \ldots, M$
    $\delta_{mk} = 0$ for $m = 1, \ldots, M$, $k = 1, \ldots, K$
  end if
  Set all counts c_{im}^{IM} and c_{mj}^{MJ} to zero
  // Initialize the motivation assignments $\mathbf{Z}$ in random order
  for each n in random permutation of 1 to N **do**
    Sample z_{in} with a Gibbs step, using the distribution in Equation (26)
    Increase the corresponding elements c_{im}^{IM} and c_{mj}^{MJ} using sampled z_{in} and y_{in}
  end for
  // Set the initial Metropolis-Hasting variances and calibration window sizes
  $s_{\beta_0}^2 = 0.1, w_{\beta_0} = 10$
  if the model is LDA **then**
    $s_{\alpha_m}^2 = 0.01, w_{\alpha_m} = 10$, for $m = 1, \ldots, M$
  else if the model is LDA-X **then**
    $s_{\gamma_m}^2 = 0.01, w_{\gamma_m} = 10$ for $m = 1, \ldots, M$
    $s_{\delta_{mk}}^2 = 0.01, w_{\delta_{mk}} = 10$ for $m = 1, \ldots, M$, $k = 1, \ldots, K$
  end if
end procedure

Algorithm 3 Sampling a new state for LDA(-X)

I: number of customers
n_i: number of purchases by the i-th customer
M: number of motivations
K: number of predictor variables in $\mathbf{x}_i$
procedure SAMPLING
  for each customer $i = 1, \ldots, I$ **do**
    for each datapoint $n = 1, \ldots, n_i$ **do**
      Decrease the corresponding elements c_{im}^{IM} and c_{mj}^{MJ} using current z_{in} and y_{in}
      Sample z_{in} with a Gibbs step, using the FCD in Equation (26)
      Increase the corresponding elements c_{im}^{IM} and c_{mj}^{MJ} using new z_{in} and y_{in}
    end for
  end for
  Sample β_0 with a Metropolis-Hastings step, using the distribution in Equation (27)
  if the model is LDA **then**
    for each motivation $m = 1, \ldots, M$ **do**
      Sample α_m with a Metropolis-Hastings step, using the distribution in Equation (28)
    end for
  else if the model is LDA(-X) **then**
    for each motivation $m = 1, \ldots, M$ **do**
      Sample γ_m with a Metropolis-Hastings step, using the distribution similar to Equation (30)
      for each predictor variable $k = 1, \ldots, K$ **do**
        Sample δ_{mk} with a Metropolis-Hastings step, using the distribution in Equation (30)
      end for
    end for
  end if
end procedure

Algorithm 4 Calibration of the Metropolis-Hastings proposal variance s^2

n: number of samples drawn in this calibration window
n_A: number of accepted samples in this calibration window
w: size of the calibration window
s^2: current proposal variance
AR: target acceptance rate
procedure CALIBRATION
  if n is equal to w **then**
    // Calculate the 95% confidence bounds of the Binomial(n, $w \times AR$) distribution
    bounds = quantile function for Binomial(n, $w \times AR$) evaluated at 0.025 and 0.975
    if n_A is outside these bounds **then** calibrate the proposal variance
      if $n_A > w \times AR$ **then** the variance is increased
        $s = s \times \min\left(\sqrt{\frac{n_A}{w \times AR}}, 4\right)$
      else if $n_A < w \times AR$ **then** the variance is decreased
        $s = s \times \max\left(\sqrt{\frac{n_A}{w \times AR}}, \frac{1}{4}\right)$
      end if
    end if
    // Reset n and n_A for new calibration window and increase w
    $n = 0$, $n_A = 0$
    if $w < 500$ **then**
      $w = w + 10$
    end if
  end if
end procedure

B. Real-time online predictions

Once we observe a new purchase for a customer we naturally want to update the predictions based on this new information. However, in an online setting it is not feasible to re-estimate a complete model in real-time. Instead, we update the customer-specific elements in real-time based on the new information, and fix the parameters that are specified at the customer-base level to their posterior means. Naturally, after observing new purchases for many customers, it makes sense to re-estimate the model structure including this new data. In this appendix we discuss for each of the prediction methods how the predictions may be updated in real-time and what the corresponding memory requirements are.

B.1. LDA(-X)

The predictive distribution for customer i in LDA(-X) is given in (8). To calculate the predictive distribution we need to evaluate two expectations: $\mathbb{E}[\,\phi_{mj}|\mathbf{Z}, \beta_0, \mathbf{Y}\,]$ and $\mathbb{E}[\,\theta_{im}|\mathbf{z}_i, \boldsymbol{\alpha}\,]$. The first expectation is the expected value of the purchase probability for product j under motivation m. As this expectation is specified at the customer-base level, we fix it to its posterior mean. The second expectation is the expected value of the individual-specific discrete mixture over the M motivations. This expectation is customer-specific and hence we update it after observing a new purchase.

For this update of $\mathbb{E}[\,\theta_{im}|\mathbf{z}_i, \boldsymbol{\alpha}\,]$ we use an approximation step. First we define $\eta_{im} = \alpha_m + c_{im}^{\mathrm{IM}}$ and use the property of the Dirichlet distribution that $\mathbb{E}[\,\theta_{im}|\mathbf{z}_i, \boldsymbol{\alpha}\,]$ is proportional to η_{im}. To update η_{im}, we add the expected value of the motivation allocation of the new purchase (denoted by $\tilde{y}_{in}$) to its previous value. To be more precise, after each new purchase $\tilde{y}_{in}$ we increase η_{im} by:

$$\begin{aligned}
\Delta\eta_{im} &= \Pr\Big[\,\tilde{z}_{in} = m|\tilde{y}_{in} = j, \{\boldsymbol{\phi}_l\}_{l=1}^{M}, \boldsymbol{\eta}_i\,\Big] \\
&= \frac{\Pr[\,\tilde{y}_{in} = j|\tilde{z}_{in} = m, \boldsymbol{\phi}_m\,]\Pr[\,\tilde{z}_{in} = m|\boldsymbol{\eta}_i\,]}{\sum_{l=1}^{M}\Pr[\,\tilde{y}_{in} = j|\tilde{z}_{in} = l, \boldsymbol{\phi}_l\,]\Pr[\,\tilde{z}_{in} = l|\boldsymbol{\eta}_i\,]} \\
&= \frac{\phi_{mj}\eta_{im}}{\sum_{l=1}^{M}\phi_{lj}\eta_{il}},
\end{aligned} \tag{34}$$

for $m = 1, \ldots, M$. Subsequent updates of the posterior mean of $\boldsymbol{\theta}_i$ can be obtained by sequentially updating the value of $\boldsymbol{\eta}_i$. This approximating update procedure provides an effective and efficient way to incorporate new information from purchases in LDA(-X).

The number of elements that have to be retrieved for an individual update step is equal to $(M \times J) + M$, namely the $\{\boldsymbol{\phi}_l\}_{l=1}^{M}$ vectors and the individual-specific $\boldsymbol{\eta}_i$ vector. To be able to perform this step for each customer, $(M \times J) + (I \times M)$ elements have to be stored in total.

B.2. MDM

The predictive distribution for customer i in MDM is given in (13). The $\{\boldsymbol{\beta}_l\}_{l=1}^M$ vectors describe the probability distributions that correspond to the purchase behavior of the customer segments and hence, are not individual-specific. As a consequence we fix them to their posterior mean. The customer-specific purchase counts c_{ij}^{IJ} are updated straightforwardly according to the new purchase, while the current segment probabilities $\Pr\Big[\, s_i = m|\mathbf{s}^{\setminus i}, \{\boldsymbol{\beta}_l\}_{l=1}^M, \mathbf{y}_i \Big]$ can be updated using the recursive property of the Gamma function, i.e. $\Gamma(n+1) = \Gamma(n)n$, see equation (32) in Appendix A.

The number of elements that have to be retrieved for an individual update step is equal to $(M \times J) + M + n_i$, namely the $\{\boldsymbol{\phi}_l\}_{l=1}^M$ vectors, the customer-specific segment probabilities $\Pr\Big[\, s_i = m|\mathbf{s}^{\setminus i}, \{\boldsymbol{\beta}_l\}_{l=1}^M, \mathbf{y}_i \Big]$, and $\mathbf{y}_i$ the purchase history of customer i. To be able to perform this step for each customer, $(M \times J) + (I \times M) + N$ elements have to be stored in total.

B.3. Collaborative filters

Suppose that a customer has n_i previously observed purchases. A new purchase made by this customer adds a maximum of $\binom{n_i}{k-1}$ product combinations to H_i^k. In order to incorporate this new information in the product ranking of customer i, we need to add for every new product combination the corresponding normalized score to s_{ij}^k (see (19)). Hence, this update step requires the retrieval of $\binom{n_i}{k-1}$ rows with J counts, the J current scores s_{ij}^k, and the purchase history $\mathbf{y}_i$ containing n_i purchases. This results in $\left(\binom{n_i}{k-1} \times J\right) + J + n_i$ elements to be retrieved when making an individual update.

To enable real-time updates for all customers, we have to combine each of the J products with each of the $\binom{J+k-1}{k}$ possible product combinations of size k and store the count for this combination of size $k+1$. In addition, we have to store the current scores and purchase history of each customer. In total this requires $\binom{J+k-1}{k} \times J + (I \times J) + N$ elements to be stored. Dependent on the combination of k and the dimensions of the application, storage of this information and real-time updating of predictions may or may not be feasible.

B.4. Discrete choice model

The predictive distribution for customer i in the DCM is obtained by calculating the log odds for all J products, as specified in (22). To calculate these log odds we need the model parameters and the cluster centroids, which are both specified at the customer-base level. Updating the customer-specific purchase history $\mathbf{y}_i$ according to the new purchase and calculating the new weights for the M customer clusters is straightforward.

The number of elements that have to be retrieved for an individual update step is equal to $(M \times J) + (2 + 2M(1+K)) + n_i$, namely the cluster means $\left\{\bar{\mathbf{v}}^{(l)}\right\}_{l=1}^M$, the logit parameters, and $\mathbf{y}_i$ the purchase history of customer i. To be able to perform this step for each customer, $(M \times J) + (2 + 2M(1+K)) + N$ elements have to be stored in total.

References

Asuncion, Arthur, Max Welling, Padhraic Smyth, Yee Whye Teh. 2009. On smoothing and inference for topic models. *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. 27–34.

Blei, David M., Andrew Y. Ng, Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research* **3** 993–1022.

Bodapati, Anand V. 2008. Recommendation systems with purchase data. *Journal of Marketing Research* **45**(1) 77–93.

Damien, Paul, Jon Wakefield, Stephen Walker. 1999. Gibbs sampling for bayesian non-conjugate and hierarchical models by using auxiliary variables. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **61**(2) 331–344.

Fader, Peter S. 1993. Integrating the dirichlet-multinomial and multinomial logit models of brand choice. *Marketing Letters* **4**(2) 99–112.

Fader, Peter S., Bruce G. S. Hardie. 1996. Modeling consumer choice among SKUs. *Journal of Marketing Research* **33**(4) 442–452.

Fader, Peter S., David C. Schmittlein. 1993. Excess behavioral loyalty for high-share brands: Deviations from the dirichlet model for repeat purchasing. *Journal of Marketing Research* **30**(4) 478–493.

Fleder, Daniel, Kartik Hosanagar. 2009. Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity. *Management Science* **55**(5) 697–712.

Ghose, Anindya, Panagiotis G. Ipeirotis, Beibei Li. 2014. Examining the impact of ranking on consumer behavior and search engine revenue. *Management Science* **60**(7) 1632–1654.

Goodhardt, G. J., A. S. C. Ehrenberg, C. Chatfield. 1984. The dirichlet: A comprehensive model of buying behaviour. *Journal of the Royal Statistical Society. Series A (General)* **147**(5) 621–655.

Griffiths, Thomas L., Mark Steyvers. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences*, vol. 101. 5228–5235.

Grover, Rajiv, V. Srinivasan. 1987. A simultaneous approach to market segmentation and market structuring. *Journal of Marketing Research* **24**(2) 139–153.

Guadagni, Peter M., John D. C. Little. 1983. A logit model of brand choice calibrated on scanner data. *Marketing Science* **2**(3) 203–238.

Jain, Dipak, Frank M. Bass, Yu-Min Chen. 1990. Estimation of latent class models with heterogeneous choice probabilities: An application to market structuring. *Journal of Marketing Research* **27**(1) 94–101.

Jannach, Dietmar, Markus Zanker, Alexander Felfernig, Gerhard Friedrich. 2011. *Recommender systems: an introduction*. Cambridge University Press.

Jeuland, Abel P., Frank M. Bass, Gordon P. Wright. 1980. A multibrand stochastic model compounding heterogeneous erlang timing and multinomial choice processes. *Operations Research* **28**(2) 255–277.

Liu, Duen-Ren, Chin-Hui Lai, Wang-Jung. 2009. A hybrid of sequential rules and collaborative filtering for product recommendation. *Information Sciences* **179**(20) 3505–3519.

Maddala, G.S. 1983. *Limited-Dependent and Qualitative Variables in Econometrics*. Cambridge University Press.

McFadden, Daniel. 1986. The choice theory approach to market research. *Marketing Science* **5**(4) 275–297.

Mimno, David, Matthew D. Hoffman, David M. Blei. 2012. Sparse stochastic inference for latent dirichlet allocation. *Proceedings of the Twenty-Ninth International Conference on Machine Learning*. 1599–1606.

Naik, Prasad, Michel Wedel, Lynd Bacon, Anand Bodapati, Eric Bradlow, Wagner Kamakura, Jeffrey Kreulen, Peter Lenk, David M. Madigan, Alan Montgomery. 2008. Challenges and opportunities in high-dimensional choice data analyses. *Marketing Letters* **19**(3-4) 201–213.

Pan, Bing, Helene Hembrooke, Thorsten Joachims, Lori Lorigo, Geri Gay, Laura Granka. 2007. n google we trust: Users decisions on rank, position, and relevance. *Journal of Computer-Mediated Communication* **12**(3) 801–823.

Ramage, Daniel, Susan Dumais, Dan Liebling. 2010. Characterizing microblogs with topic models. *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*. 130–137.

Salganik, Matthew J., Peter Sheridan Dodds, Duncan J. Watts. 2006. Experimental study of inequality and unpredictability in an artificial cultural market. *Science* **311**(5762) 854–856.

Steyvers, Mark, Tom Griffiths. 2013. *Handbook of Latent Semantic Analysis*, chap. Probabilistic Topic Models. Psychology Press, 424–440.

Train, Kenneth E. 2009. *Discrete choice methods with simulation*. 2nd ed. Cambridge University Press.

Wagner, Udo, Alfred Taudes. 1986. A multivariate polya model of brand choice and purchase incidence. *Marketing Science* **5**(3) 219–244.

Wallach, Hanna M., David Mimno, Andrew McCallum. 2009. Rethinking LDA: Why priors matter. *Advances in Neural Information Processing Systems 22*. 1973–1981.

Wedel, Michel, Wagner A. Kamakura. 2000. *Market segmentation: conceptual and methodological foundations*. 2nd ed. Kluwer Academic Publishers.

Xu, Yunjie (Calvin), Hee-Woong Kim. 2008. Order effect and vendor inspection in online comparison shopping. *Journal of Retailing* **84**(4) 477–486.

Zanutto, Elaine L., Eric T. Bradlow. 2006. Data pruning in consumer choice models. *Quantitative Marketing and Economics* **4**(3) 267–287.