Multiscale and Dirichlet Methods for Supply Chain Order Simulation

R. Paul T. Sabin

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Statistics

David M. Higdon, Chair
Xinwei Deng
Kimberly P. Ellis
Leanna L. House

February 22, 2019
Blacksburg, Virginia

Keywords: Multiscale, Dirichlet, Bayesian, Supply Chain

Multiscale and Dirichlet Methods for Supply Chain Order Simulation

R. Paul T. Sabin

Supply chains are complex systems. Researchers in the Social and Decision Analytics Laboratory (SDAL) at Virginia Tech worked with a major global supply chain company to simulate an end-to-end supply chain. The supply chain data includes raw materials, production lines, inventory, customer orders, and shipments. Including contributions of this author, Pires et al. (2017) developed simulations for the production, customer orders, and shipments. Customer orders are at the center of understanding behavior in a supply chain. This dissertation continues the supply chain simulation work by improving the order simulation. Orders come from a diverse set of customers with different habits. These habits can differ when it comes to which products they order, how often they order, how spaced out those orders times are, and how much of each of those products are ordered. This dissertation is unique in that it relies extensively on Dirichlet and multiscale methods to tackle supply-chain order simulation. Multiscale model methodology is furthered to include Dirichlet models which are used to simulate order times for each customer and the collective system on different scales.

Multiscale and Dirichlet Methods for Supply Chain Order Simulation

R. Paul T. Sabin

This dissertation continues the supply chain simulation work of researchers (Pires et al. (2017)) in the Social and Decision Analytics Laboratory (SDAL) at Virginia Tech by improving the order simulation. Orders come from a diverse set of customers with different habits. These habits can differ when it comes to which products they order, how often they order, how spaced out those orders times are, and how much of each of those products are ordered. This dissertation is unique from the previous work at SDAL which considered few of these factors in order simulation and introduces statistical methodologies to deal with the complex nature of simulating an entire supply chain's orders.

This dissertation is dedicated to my wife Helen Ann Sabin, who put more hours towards this doctorate than I did. She has done a wonderful job raising 4 kids in circumstances much more difficult than most including having a spouse in graduate school on top of working a full-time job. This dissertation would have never been completed if it weren't for her picking up all that I neglected during the early mornings, late nights, and countless hours in between while I was working on my coursework, research, and dissertation.

I would like to acknowledge first my father and mother, Terry and John Sabin. They taught me the value of education at a young age and always encouraged me to further my learning. Next, I would like to thank my advisor Dr. David Higdon who spent many hours with me video-conferencing while I completed this dissertation while living remotely. He was always involved and very flexible despite being busy with other projects that I'm sure were more pressing. I want to thank Ken Hamall and his team at Procter & Gamble for their assitance on this work. I would also like to thank Professor Emeritus Dr. Jeffrey Birch who recruited me to come to Virginia Tech and further encouraged me to finish my degree when I received a full-time job offer at ESPN. I would also like to thank current graduate chair Dr. Marco Ferreira for working with me in finding a path to fulfill all the requirements for a Ph.D. in statistics. Lastly I would like to thank Dr.'s William Christensen, Scott Grimshaw, and Shane Reese of Brigham Young University for first pushing me to pursue a Masters Degree and then a Ph.D. I had many moments where I doubted I could even finish the curriculum of a Bachelor's degree in statistics and each of them showed a belief in me when I didn't have much of a belief in myself.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

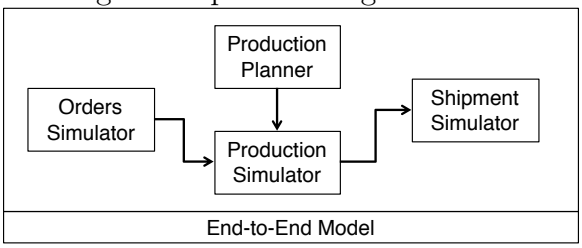
Supply chains are complex systems. While simulation has been widely used to model supply chain processes (Jahangirian et al. (2010)), informing these models through the use of transactional data has not been extensively explored. This could in part be due to the difficulties around stakeholder engagement because simulations often require extensive data gathering and processing (McNaught and Chan (2011)). Recent advances in data collection and storage, however, facilitate the integration of these new data sources with simulation models. Bayesian statistical methods have proven to be a useful framework for combining observational data with simulation-based models while accounting for the various sources of uncertainty (Reese et al. (2004); Bayarri et al. (2007)). Such uncertainty is a key characteristic throughout many parts of the supply chain (McNaught and Chan (2011)).

Bayesian approaches have been applied to various areas of manufacturing, such as fault diagnosis (e.g., Garcia et al. (2008); Jeong et al. (2006); Jin et al. (2012)), quality control (e.g., Correa et al. (2009); Pradhan et al. (2007)), reliability (e.g., Ok et al. (2008); Langseth and Portinale (2007); Li and Meeker (2014); Celik and Son (2010)), and supply chain disruptions (e.g., Soberanis (2010)). Moreover, hybrid approaches that combine two or more simulation techniques (e.g., discrete-event simulation, system dynamics) have seen a rise in popularity due to the increased trend in providing "enterprise wide solutions" that take into account the impact that different parts of an organization have on one another (Jahangirian et al. (2010)). Other hybrid approaches have explored the use of these simulation techniques with Bayesian modeling (e.g., Xu and Son (2013)). Chick (2004), in particular, stresses the benefits of combining Bayesian methods with discrete-event simulation, including for input modeling, response surface modeling, and uncertainty analysis.

While such previous models explored hybrid methods, an effort to integrate Bayesian modeling and discrete-event simulation with disparate big data streams across a supply chain system (from customer orders to deliveries) was conducted by the Social and Decision Analytics

Laboratory at Virginia Tech (Pires et al. (2017)). The motivation behind this dissertation is the work of Pires et al. (2017) of which I was a co-author. Our work allowed a better understanding of how to manage the uncertainty, variation, and dependencies in the supply chain and to make predictions of behaviors under new conditions. This required the development of a data informatics model that could be used to realize a digital synchronized supply chain model. To realize this model, we took a hybrid approach that combines Bayesian modeling with discrete-event simulation and applied it to the supply chain process of an organization that carries out both manufacturing operations and distribution activities. We informed the model using approximately one year of transactional data. A driving force for creating this model was to better understand the balance between inventory, profit, and service. It should be noted that while we simulated the main processes between customer orders and customer shipments, a complete supply chain model would also need to include data on consumers and raw material suppliers.

Figure 1.1: High-level process diagram of the simulators.



1.1 Supply Chain Orders

A supply chain for an international company can be a very complex system. Perhaps the goal of the end-to-end supply chain simulation is better understood through a simpler baseball analogy given by Stern (2005). Suppose that you are the manager of the baseball team who is at-bat while down by one run in the bottom of the eighth inning. There is a base runner on first base. You have a decision to make, should you bunt to send the runner to second, but give up an out in the process, or should you let the next batter swing away? Bunting is a stochastic process where the batter bunting has a probability $p_{bunt} \in (0, 1)$ of successfully executing the bunt. Swinging away in an effort to get a base hit is also a stochastic process with a success probability $p_{swing} \in (0, 1)$. In this situation there is a decision that has to be made (bunt or swing away) with a resulting stochastic process related to each possible decision. The decision made should be the one that maximizes the probability of winning the game. Winning the game in this situation requires scoring at least 2 runs before the end of the ninth inning.

Stern (2005) solves the question of whether or not to bunt by estimating the transition probabilities between the 24 out and base state combinations in a Markov Chain. Due to the simplicity of the game of baseball, an exact solution can be solved analytically. The end-to-end supply chain is much more complex but the set-up of the problem is very similar. In the baseball example a decision had to be made: Bunt or swing away. In the supply chain system there are two main areas where decisions have to be made:

- Production schedules and
- Shipping service: The trade-off in between shipping an incomplete order sooner or a complete order later.

In both systems there are stochastic processes that interact with these decisions. In baseball the stochastic processes involve the uncertainty around the probability of a successful bunt or a successful at-bat when swinging away. In the supply chain system, the main stochastic system is that of the customer orders. (A diagram of the complete end-to-end model can be seen in Figure 1.1.) The production simulator is a function of the production planner and the orders coming in. The shipment simulator is deterministic based on the shipping service rules input by the user. The order simulator is the most important aspect of this end-to-end model as customer demand is the stochastic process of a supply chain with the largest down-stream impact. The order simulator used in Pires et al. (2017) was useful and helped provide concrete answers on which levers to pull for the production planner and the quality of the shipping service. This dissertation will devote its time to improving the order model simulator used in Pires et al. (2017) (explained in Section 1.2).

1.1.1 Bull-whip Effect

Before getting into the specific mechanisms of the order simulator used in Pires et al. (2017) and how it will be improved, it is important to understand the effects a poor order (demand) simulator can have on an end-to-end supply chain simulation. Customer satisfaction, the primary goal of supply chains, often falls short due to random fluctuations in the demand pattern (Min and Zhou (2002)). As customer demand changes, there is increased variability upstream on the supply chain due to shifts in inventory levels (Wright (1961), Lee et al. (1997), Blanchard (1983), Blinder (1982), and Kahn (1987)), a phenomenon known as the "Bull-whip Effect" that has been extensively studied. The Bull-whip Effect causes inefficient management of production and inventory levels (Blinder (1982) and Kahn (1987)) which, of course, hurt the profits of each upstream in a supply chain. Inventories are insurance against uncertainty in a supply chain (Davis (1993)). As insurance, storing inventory is costly, as is a lack of sufficient inventory when demand increases and customers do not receive their products in a timely manner. As inventory levels decrease the production lines need to be

kicked to gear replenish the inventory levels of that product.

Several have studied the causes of the Bull-whip effect and suggested ways to diminish its influence. Lee et al. (1997) claim that a cause for the Bull-whip effect is that the information transferred from orders as it moves upstream often distorts and misleads those in charge of inventory and production decisions. In this they claim that demand forecasting is one of five main causes for the Bull-whip effect. Chen et al. (2000)a showed that updating the mean and variance of demand forecasts based on observed data too frequently can cause an overestimation of actual customer demand. Chen et al. (2000)a also showed that one way to decrease variation in the supply chain is to make sure each stage of the chain has access to observed customer demand information.

The most direct way to decrease the Bull-whip effect would be to address the problem at its source, variance in customer orders. Chen et al. (2000)a and Lee et al. (1997) attempt to quantify the Bull-whip effect for a single retailer and a single manufacturer. Assuming the retailer employs a moving average, Chen et al. (2000)a used an auto-regressive (AR-1) process to forecast demand and create a lower-bound on the variance of the orders from the retailer. Chen et al. (2000)b followed that up and assumed that retailers use an exponential smoothing model, which is described in most operations management textbooks, to forecast lead-time and calculate an order variance lower-bound. Carbonneau et al. (2008) compares machine learning techniques, including neural-networks and support vector machines, in demand forecasts to linear regression, moving average, and last observed forecasting techniques. They find that the machine learning techniques predict order demand much better than smoothing or moving average techniques but that they are statistically insignificant from an autoregressive multiple linear regression model.

Hundreds of Customers

This research differs from these other studies of customer demand in supply chains in various fashions. First, it differs from Lee et al. (1997), Chen et al. (2000), and Chen et al. (2000)b in that it does not assume that the retailer uses a model to place orders. Instead, the order data simulation is based on observed order data by hundreds of retailers across two years. Different models are built based on this data to probabilistically recreate customer orders. Through realistic order simulation in a larger supply chain simulation, the consequences of the Bull-whip effect can be appropriately anticipated and dealt with in a more optimal manner.

Secondly, Lee et al. (1997), Chen et al. (2000)a, and Chen et al. (2000)b all assume a single retailer while Carbonneau et al. (2008) looks at simulated and observed order data from a

collective group of consumers. The methods of forecasting demand evaluated by Carbonneau et al. (2008) may work well for an entire class of customers; they do not, however, provide the level of granularity needed to simulate an entire supply chain in the manner of Pires et al. (2017). The supply chain simulation in both Pires et al. (2017) and this dissertation look at the collective group of customers **and** each individual customer. The approach is designed to handle any number of customers that may behave very differently from one another.

If the goal of a supply chain is to please the consumer, and on-time deliveries are an essential part to that, order times also need to be modeled. It is not enough to forecast total demand in a particular day, week, or month. Each customer will react to whether or not delivery and service is on-time. For that reason this paper will model order times not by pre-defined "buckets" of time such as days, weeks, or months, but as a continuous variable.

In this fashion, the model derived in this paper for order times is multiscale, meaning that it is capable of simulating and describing behavior for all collective customers and for each individual customer simultaneously. I show in Section 4.3.2 an example of how the behavior of the order times for the entire group of customers is not harmonious with the implied behavior in aggregating the model for all customers. In other words, the set of all customers behave differently than each individual customer independently.

1.2 Need for Better Order Simulator

In order to understand why this dissertation's purpose is to improve on the order simulator used in Pires et al. (2017), it is first important to understand the order simulator used in that paper. The method, presented below with some light editing from the original paper, relies heavily on resampling from existing data and basic assumptions, many of which will be shown to be generous or limiting. This paper will almost extensively use Bayesian statistical models to replace most of the empirical estimation described below. Statistical methods introduce the ability to make inference on the similarities and differences between different aspects of customer behavior, important in the business of supply chains. The statistical models, presented from a Bayesian perspective, will also allow for the simulation for new customers. With no pre-existing orders, empirical methods do not know how to handle new customers.

From Pires et al. (2017):

The orders are simulated using a combination of Bayesian forecasting at an aggregated level and empirically-based estimation at the detailed customer × product × order level. The approach used for simulating a month of orders is outlined below.

1. A time-series model Harrison and West (1999) is used to forecast the total volume of orders – aggregated over all customers and product groups – expected for the distribution center for the month.

2. This total volume is distributed over customers and product groupings based on the historical distribution from the previous month. This distribution could be estimated from an alternative time window or could account for known changes that will be present for the month being forecasted.

3. For each forecasted volume by customer, a set of orders that match previous customer order behavior are constructed that sums to the forecasted volume by customer and product grouping. Below are the details for this step:

   (a) For each customer, estimate the monthly order rate μ using historical data (typically the previous month). The simulated total order volume for the month is drawn from the resulting predictive distribution for the model.

   (b) Sample the estimated number of customer orders N from a Poisson distribution with mean μ.

   (c) Alter the ordered amounts in the sampled orders so that the volume aggregates appropriately over each of the product groups. This is done by treating the collection of orders as a vector random variable $\{v_{po}\}$ where $p = 1, \ldots, P$ indexes product, and $o = 1, \ldots, N$ indexes order. We construct a distribution for this vector by combining distributions that characterize different features of this collection of customer orders. A description of these distributions is shown in Table 1.1. These densities are combined using Bayes rule Dawid et al. (1995), and then sampled using Markov chain Monte Carlo Robert and Casella (1999).

   (d) Change the order dates of the sampled orders to reflect dates compatible with the month being simulated.

Table 1.1: Distributions used to characterize the different features of customer orders.

density	description
$p(v_{po})$	Independent normal distributions for each v_{po} that is centered at the actual order amount, with standard deviation of 15%.
$f(\sum_{p \in G_k, o} v_{po})$	For each product group G_k, $f(\cdot)$ is a normal density with mean equal to the total volume determined in step 2, and a standard deviation of 10%.
$I[v_{po} \geq 0]$	Constraints ensuring that each order volume is either 0 or positive.

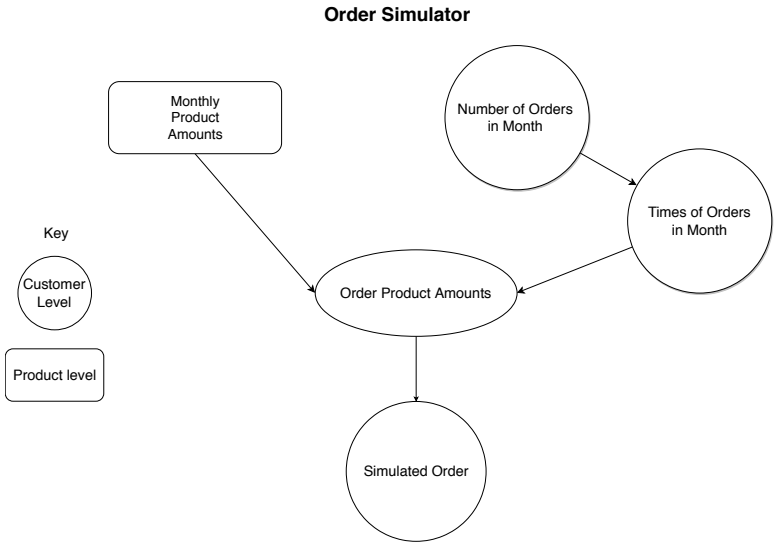
This algorithm results in a collection of orders that (1) reflects previous order specific product grouping behavior for each customer, (2) matches specified totals by product grouping and customer for that month, and (3) matches the forecasted grand total volume for the month. Here the forecast is based on data up to the month being forecasted. It is important that the simulated orders capture the monthly and daily variations seen in the actual orders so that the optimization of production planning and inventory levels will be robust to variation in the ordering. Using this holdout approach, we find that the simulated order timings and volume match to within about 5% for very regular products. Less regular products can show substantial variation in order timing and volume over the course of a month. For example, one product sees more than half its monthly volume arrive on a single day. The fitted order simulator generally recognizes these high variation products, producing substantial variation in the simulated order histories.

The preceding approach describes in detail the following pieces that are needed to have a complete order simulator,

1. Monthly order volume,
2. Number of orders per month for each customer,
3. Order time for each order, and
4. Allocating volume to each product in each order .

In short, this dissertation will also have each of those four parts. Each part develops models that get to the customer or product level as well as the aggregate level for all products/customers when necessary. The time-series for monthly order volume (1) will be expounded to include a multiscale model that models the monthly volume for all of the product groups. This step alone increases the accuracy and statistical prediction properties of the simulator by considering the relationships between the monthly order volumes of products. The Poisson model described in (2) above for the number of orders in a month for a customer is further enhanced by introducing a time-series component, when upon further inspection, it is apparent that there is a time-dependency in the number of orders in a month for many customeres. The order times (3) will not be based on sampling historical order times for each customer, but rather an innovative approach using the properties of Dirichlet random variables combined with multiscale models to probabilistically model order times in the future months. Lastly, instead of volume allocation (4) to customers and products by historical rates, a method used in topic-models, Latent Dirichlet Allocation (LDA) will be introduced as a means to cluster and predict order volume allocation.

Figure 1.2: Diagram of the four models that combine to create the order simulator.



1.3 Review of Dirichlet Models

Models that make heavy use of the Dirichlet likelihood and Dirichlet random vectors are used heavily throughout this paper. These models are all presented from a Bayesian perspective. This section will review some of the properties of Dirichlet random variables so that the discussion later will be more clear. While multiscale methods are also used several different places in this work, the foundations of those will be visited in Chapter 2.

There are many well-known ways to think about the Dirichlet distribution, one interpretation is that it can be thought of as the multivariate extension of the beta distribution. Recall, that if $x \sim Beta(\alpha, \beta)$ then $x \in (0, 1)$. Similarly if $\mathbf{x} = (x_1, x_2, \ldots, x_k)$, $(x_1, x_2, \ldots, x_K) \in (0, 1)^K$, and $\sum_{i=1}^{K} x_i = 1$ then $\mathbf{x} \sim Dir(\boldsymbol{\alpha})$. The form of the probability density function (PDF) is similar to that of the beta distribution as,

$$f(\mathbf{x}) = \frac{\Gamma(\sum_{i=1}^{K} \alpha_i)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \prod_{i=1}^{K} x_i^{\alpha_i - 1} \ . \tag{1.1}$$

The expected value and variance for any element of Dirichlet random vector is shown in Equations 1.2 and 1.3. The expectation is the percentage of the element of the parameter $\boldsymbol{\alpha}$ compared to the sum of $\boldsymbol{\alpha}$. The variance takes into account not only element i's fraction of the sum of the parameter vector $\boldsymbol{\alpha}$ but also the total sum of $\boldsymbol{\alpha}$. Thus, larger values of

$\sum_{i=1}^{K} \alpha_i$ result in a smaller variance for the Dirichlet random vector.

$$\mathrm{E}(x_i) = \frac{\alpha_i}{\sum_{i=1}^{K} \alpha_i} \tag{1.2}$$

$$\mathrm{Var}(x_i) = \frac{\alpha_i((\sum_{i=1}^{K} \alpha_i) - \alpha_i)}{(\sum_{i=1}^{K} \alpha_i)^2(\sum_{i=1}^{K} \alpha_i + 1)} \tag{1.3}$$

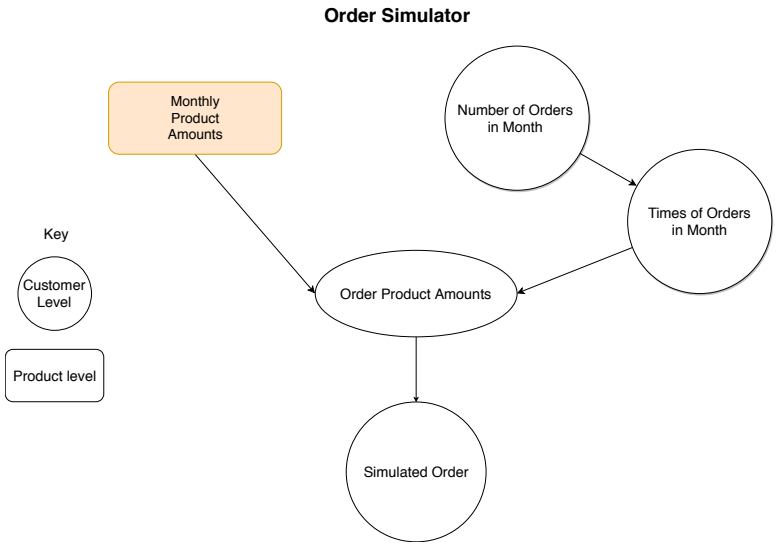
There are several different ways to simulate Dirichlet random vectors. The most common comes from Ferguson (1973) who showed that for a set of independent random variables, $Z_j \sim Gamma(\alpha_j, 1)$ where $\alpha \geq 0 \; \forall j$ and $\alpha > 0$ for some j, if we set $Y_j = \frac{Z_j}{\sum_{i=1}^{K} Z_j}$ then $\mathbf{Y} \sim Dirichlet(\boldsymbol{\alpha})$ where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_K)$. This interpretation is often used computationally to simulate a Dirichlet random variable.

In Bayesian analysis the Dirichlet distribution comes in very handy. Since the Dirichlet is a multivariate generalization of the beta distribution, it is natural that it is a conjugate prior to the multivariate extension of the binomial distribution, the multinomial distribution. Recall, that the multinomial distribution is the result of n independent trials of k possible outcomes. The Dirichlet distribution is central to several advances in recent decades in methods of nonparametric Bayesian analysis and topic models (Ferguson (1983) and Blei et al. (2003)). This dissertation will introduce the Dirichlet distribution and random variable as a key to supply chain order simulation. Over the following chapters the methodology behind the models in each part of the enhanced order simulator will be established. In each chapter the methodology will be applied to the actual supply chain order data over the time period of January 2013 to December 2014.

Chapter 2

All Product's Monthly Volume

Figure 2.1: Diagram of the four models that combine to create the order simulator. This chapter covers highlighted portion.



Perhaps the most defining feature of trying to build a system for order simulation is all the moving parts that need to accurately reflect the dynamic order process. First, there's the fact that attached to each order is *how much* is ordered of *which* products and *when* they are ordered. Each order can consist of any combination of the products available. Second, this simulator is concerned with *all* customers, each with different habits in timing and volume. Creating a model for the order volume for one product regardless of the customer is a much more straight-forward problem and there are several existing methods across economics, operations research, and statistics which may suffice (e.g., Cragg (1971)). Adding to the complexity is that an order simulator that is worried about production/inventory

and shipping is much more difficult because it is concerned with individual customers for shipments *and* the aggregate behavior for each products that affect production schedules and inventory levels. In this chapter I start with the aggregate behavior of each product. Personalizing the simulator for specific customers will be addressed in later chapters.

Order quantities are usually thought of as discrete, typically Poisson random variables such as 125 bars of soap or 12 packs of diapers. Additionally, each product has many different packages. For example, the same brand and size of diapers is sold in various quantities, each quantity with a unique stock keeping unit (SKU). The sizes of packages change constantly with new SKU's being introduced all the time and others being discontinued fairly regularly. This makes it difficult to build a statistical model on the SKU level. Instead, sets of SKU's are grouped together for a given product in what will be referred to hereafter as a product grouping. A brand of shampoo or soap is treated as one product grouping, while diapers are grouped into the brand and size. Sometimes, I will refer to a size or similar sizes as a tier. The data available is the most rich for the diaper products as those products are the highest produced at this production facility.

If I want to construct models with total amounts ordered across products, the units have to be standardized. Each SKU has a conversion from any unit of measurement to what is referred to as "standardized units." These will be used throughout this paper because the values in common units of measurement is proprietary. These standardized units (su for short) are not integer values but are continuous values. Thus, from now on order quantities for products discussed won't take on whole numbers and will be treated as approximately continuous random variables.

2.0.1 Tree Structure of Products

The products which are produced at the facility range from diapers and wipes to shampoo to feminine hygiene. Certainly there are trends relating to the overall demand of customers and specific trends to a class of products, such as shampoo. While there are most likely cyclical trends over the course of the year, with only two years of data, estimation of such trends could introduce over-fitting. As such, I aim to capture any general trends both of overall demand as well as demand pertaining to a class of products or to a specific product. Naturally the overall demand in a month is defined as the sum of the demand of all individual products plus small random noise.

The structure of the products will be as follows:

- All Products

- Diapers & Wipes
  - Diaper Brands
    - Sizes within Diaper Brands
  - Wipes Brands
- Other Non-Diaper Products .

Some diaper sizes are not ordered regularly from this location. In Figure 2.2 one can see certain diaper sizes that are ordered in high quantities every month and others that are not ordered in some months. The sizes that are not ordered every month are either the smallest or largest sizes. The sizes with zero monthly orders are combined with the next size larger or smaller to ensure all product groups are nonzero.

Figure 2.2: Monthly order amounts for diaper brands. Each row is a different size. Values are converted to standardized units (SU).

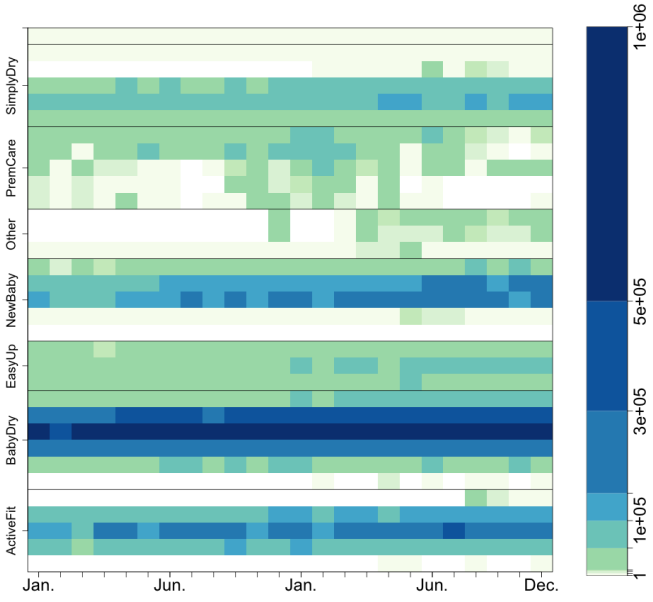
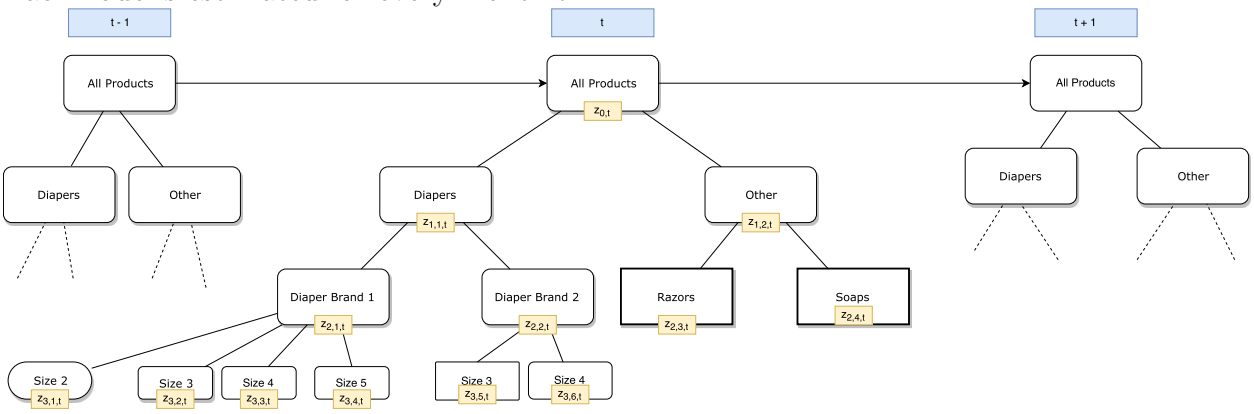


Figure 2.3 shows the overall tree structure in graphical format. The method used to model the demand in each of the nodes will be a multiscale model with time dependencies across months. In multiscale models, the node from which all others are descended is referred to as the most coarse node or the most coarse level. The nodes that have no descendants are referred to as being on the most fine level. The model is multiscale because it is fit both on the most fine and the most coarse levels.

Figure 2.3: Tree diagram of the multiscale structure of products
Each node is estimated for every month t.



The most coarse level of products has a fairly consistent order pattern from month to month (see Figure 2.4). Beyond the most coarse level, the amounts for less coarse product nodes are not as consistent but still exhibit much time dependency. The amount ordered at the most coarse level filters down to each of the descendant nodes. At the same time, each node does not always have the same fraction of the demand of the parent node. The fractions of the descendants also vary based on each node's previous values in the multiscale time series. In Pires et al. (2017), a Dynamic Linear Model (Harrison and West (1999)) was fit on the most coarse level, but no model was used to estimate the trend in order volume in the descendant nodes. Instead, historical trends were used, which omits the ability to estimate the time series variance. This model will not only estimate the expected order volume for each product node, but also the true demand (i.e. "signal") for each product node. This true demand will be able to draw strength by accounting for what demand trends are occurring in other various product nodes, whether on the fine level or the most coarse.

2.1 Coarse Model

The most coarse node, i.e. the total amount ordered (su) for all the products in this production/distribution region will be treated as a Dynamic Linear Model (DLM). A similar DLM is also used in Pires et al. (2017) to model the amount ordered across all products in each month. Dynamic Linear Models (also called state-space models) were introduced by West and Harrison (1997), and are a generalization of nearly every linear model, including ones with time dependencies. The coarse model as described in Equations 2.1 to 2.4 is similar to an auto-regressive time series model. One major difference is that the time dependency is on the latent demand (evolution equation) at time (month) t, denoted by $z_{0,t}$. In traditional frequentist time-series models, the time dependency is on the observed quantity y_t. The ob-

servation error is v_t and the evolution error is w_t. Both errors are normally distributed with ν as the variance of the observation error and $\phi\nu$ as the variance for the evolution error. In this parameterization, ϕ is referred to as the signal to noise parameter, meaning that $\phi > 1$ denotes that the uncertainty around the signal is a bigger contributor than the uncertainty around the observed values of the volume. Similarly $\phi < 1$ would have the inverse interpretation.

$$y_t = z_{0,t} + v_t \tag{2.1}$$
$$z_{0,t} = z_{0,t-1} + w_t \tag{2.2}$$
$$v_t|\nu \sim N(0, \nu) \tag{2.3}$$
$$w_t|\phi, \nu \sim N(0, \phi\nu) \tag{2.4}$$

2.1.1 Coarse Priors

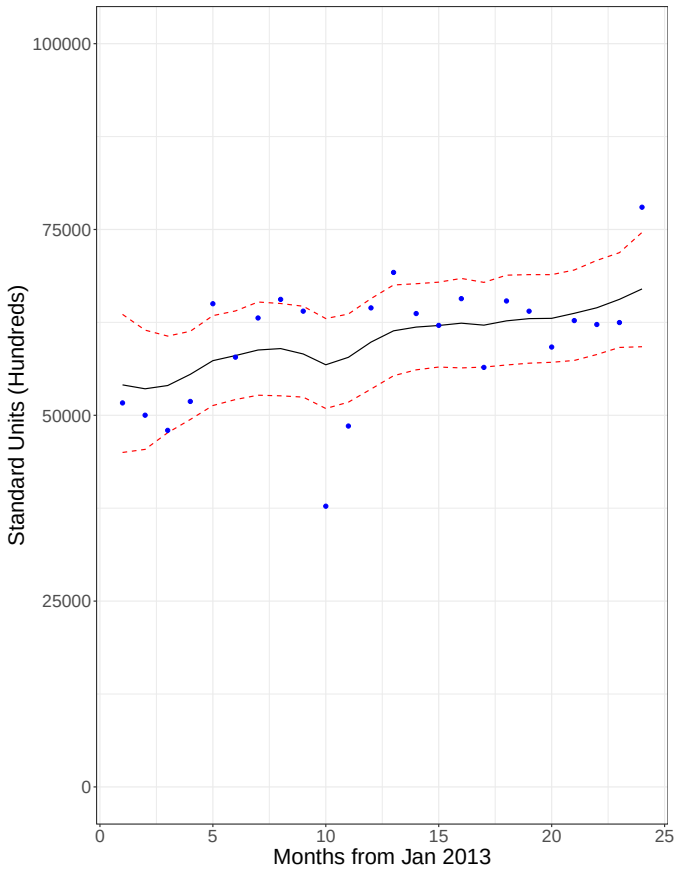
The prior specification for the observation and evolution variance terms need to be specified. The observed values for the coarse order volume in the months are all between 30,000 and 80,000 standardized units (su). This means that the variance terms will be quite large since the scale is quite large. In defining prior values, it is desired that the observed data dominant the posterior. For that purpose there are not overly restrictive prior specification on the evolution and observation variances. Most importantly with talking to the industry experts, the evolution variance should be smaller than the observation variance. This is reflected in the prior specifications in Equations 2.5 to 2.8. In defining *a priori* that ϕ is expected to be less than 1, I am signaling that it is much more important to capture the general demand trend, which will be more predictive than trying to have a good model fit on the observed values.

$$\frac{1}{\nu} \sim Gamma(a_\nu, b_\nu) \tag{2.5}$$
$$\frac{1}{\phi} = \tau \sim Gamma(a_\tau, b_\tau) \tag{2.6}$$
$$a_\nu = 2, b_\nu = 0.5 \tag{2.7}$$
$$a_\tau = 100, b_\tau = 10 \tag{2.8}$$

In addition to the prior specification of the observation and evolution variances, the state space model requires a specified starting distribution at time $t = 0$ for $z_{0,t}$. This will be set as $z_{0,0} \sim N(50,000, \phi\nu)$. The model will be fit via the Forward Filtering Backward Sampling (FFBS) Markov chain Monte Carlo (MCMC) algorithm (Frühwirth-Schnatter (1994), De Jong and Shephard (1995), & Doucet et al. (2000)). The FFBS algorithm first performs

a Kalman filter (Kalman (1960)) in estimating $z_{0,t+1}|z_{0,t}$ for each time $t = 1, \ldots, T-1$. Then the backward sampler starts at $t = T$ and works back by sampling from the distribution of $z_{0,t}|z_{0,t+1}$. The 95% credible posterior interval and the posterior mean of $z_{0,t}$ is seen in Figure 2.4. The posterior mean of $z_{0,t}$ can also be thought of as the most likely estimate of the total demand volume for each month. It is apparent that demand is not constant and generally increased over this two year time period.

Figure 2.4: All product volume ordered by month with posterior mean and 95% credible interval.



In Table 2.1 a posterior summary of the observation and evolution variance is given. As was expected *a priori*, there is more observation variance than evolution variance. In common vernacular there is a lot of noise compared to the signal.

Table 2.1: Posterior summary for observation & evolution variance

	Mean	95% Cred. Int.
ν	52,981,994	(26,571,471 - 102,850,239)
ϕ	0.1283	(0.0975 - 0.1879)
$\phi\nu$	6,640,740	(3,469,546 - 12,232,239)

2.2 Multiresolution Model

In order to establish a multiscale or multiresolution model for the order volumes for each of the nodes in the tree diagram described in Section 2.0.1, some assumptions about the coarse to fine relationship need to be made. For these purposes $z_{0,t}$ doesn't necessarily refer to the **most** coarse node. It will refer to any parent node, i.e. the order volume demand at time t for a node with a vector of descendents with order volume demand $\mathbf{z}_{1,t}$ (i.e. children nodes). I will assume the following base model at time t for the order volume in a month for any node and its direct descendents,

$$q(z_{0,t}) = N(\mu_{0,t}, v_0) \ , \tag{2.9}$$

$$p(\mathbf{z}_{1,t}) = N(\boldsymbol{\mu}_{1,t}, \langle \mathbf{v}_1 \rangle) \ , \tag{2.10}$$

$$p(z_{0,t}|\mathbf{z}_{1,t}) = N(z_{0,t}; \mathbb{1}^\intercal z_{1,t}, \delta) \ . \tag{2.11}$$

Here $\langle \mathbf{v}_1 \rangle$ is a diagonal matrix of a variance vector $\mathbf{v}_1$. In other words, each descendent has its own independent base model. The conditional distribution $p(z_{0,t}|\mathbf{z}_{1,t})$ is one that aggregates the fine data to its parent node. The aggregation in this case is the sum of the fine volume within some normal variance δ. This variance δ acts as a measure of tolerance for how close the value of the parent node needs to be to the sum of its descendents. The measure q and the measure p may not agree with each other. For example, imagine a process where the model for the fine volume $p(\mathbf{z}_{1,t})$ is believed along with the conditional aggregation model $p(z_{0,t}|\mathbf{z}_{1,t})$. This implies,

$$p(z_{0,t}) = \int p(z_{0,t}, \mathbf{z}_{1,t}) d\mathbf{z}_{1,t} \tag{2.12}$$

$$\implies p(z_{0,t}) = \int p(z_{0,t}|\mathbf{z}_{1,t}) p(\mathbf{z}_{1,t}) d\mathbf{z}_{1,t} \tag{2.13}$$

$$\implies p(z_{0,t}) = N(z_{0,t}; \mathbb{1}^\intercal \boldsymbol{\mu}_{1,t}, \mathbb{1}^\intercal v_1 \mathbb{1} + \delta) \ . \tag{2.14}$$

Thus, the density $p(z_{0,t})$ is implied by the densities $p(\mathbf{z}_{1,t})$ and $p(z_{0,t}|\mathbf{z}_{1,t})$. Unfortunately $p(z_{0,t})$ and $q(z_{0,t})$ may not agree with each other. This will be the case as long as $\mu_{0,t} \neq \mathbb{1}^\intercal \boldsymbol{\mu}_{1,t}$ or $v_o \neq \mathbb{1}^\intercal v_1 \mathbb{1} + \delta$. For these purposes it is assumed the model for the coarse volume as described in Section 2.1 is not necessarily equal to $p(z_{0,t})$.

This is analogous to updating what is believed about $z_{0,t}$ after the values of $\mathbf{z}_{1,t}$ infer a different probability model about $z_{0,t}$. In other words, based on the fine models, the coarse model is $p(z_{0,t}) = N(z_{0,t}; \mathbb{1}^\intercal \boldsymbol{\mu}_{1,t}, \mathbb{1}^\intercal v_1 \mathbb{1} + \delta)$. Then, new information comes in and suggests $q(z_{0,t}) = N(\mu_{0,t}, v_0)$. In turn this new belief of the coarse model $q(z_{0,t})$ implies a different belief about $\mathbf{z}_{1,t}$. It is desired to obtain the joint probability of $z_{0,t}$ and $\mathbf{z}_{1,t}$ but it is needed to reconcile $p(\mathbf{z}_{1,t})$ with $q(\mathbf{z}_{1,t})$. With the new information, $q(z_{0,t})$, Jeffrey's rule of conditioning needs to be used to update the model of all the fine nodes, $\mathbf{z}_{1,t}$.

Jeffrey's Rule of Conditioning

We draw on the construction of a multiscale Markov Random Field model as suggested by Ferreira et al. (2005). In this approach to building a multiscale model one must apply Jeffrey's rule of conditioning (see Jeffrey (1988)). Jeffrey's rule is a method in which the probability measure is revised for a subset of unknowns when new information is received.

> From Ferreira and Lee (2007): "[Jeffrey's] rule explains how to revise an old joint probability model for unknowns when new information completely revises the marginal probability distribution for a subset of these unknowns ..., we build an initial joint probability model for the coarse and fine levels by assuming that the fine level follows a Markov random field process and by assuming that the coarse level is a linear function of the fine level plus noise."

There is a distinction between Jeffrey's rule of conditioning and the classical use of Bayes rule. One way to conceptualize this difference comes from Dubois et al. (1991):

Case 1 A die has been tossed. You assess the probability that the outcome is 'Six'. Then a reliable witness says that the outcome is an even number. How do you update the probability that the outcome is 'six' taking in due consideration the new piece of information.

Case 2 One hundred dice have been tossed. You assess the proportion of 'six'. Then you decide to focus your interest on the dice with an even outcome. How do you compute the proportion of 'six' among the dice with an even outcome.

The first case is a revision of the probability measure due to new information. Case 2 is a refocusing, no new information is available, but the focus is now on a given subset of the original set. Bayes rule can be applied in both cases but Jeffrey's rule is most applicable to case 2.

Jeffrey's rule of conditioning is necessary to ensure that a multiscale model is consistent at each level. How do we reconcile the measures $q(\mathbf{z}_{1,t})$ and $p(\mathbf{z}_{1,t})$? Using Jeffrey's rule of conditioning we assume $q(z_{1,t,c}|z_{0,t}) = p(z_{1,t,c}|z_{0,t})$ for each customer (c) partition of $\mathbf{z}_{1,t}$. Then,

because each $z_{1,t,c}$ is conditionally independent, by Jeffrey's Rule $p(\mathbf{z}_{1,t}|z_{0,t}) = q(\mathbf{z}_{1,t}|z_{0,t})$. The multiscale model will estimate the joint distribution of each and every node. Following the framework Ferreira et al. (2005), we have the following series of equations,

$$q(z_{0,t}, \mathbf{z}_{1,t}) = q(\mathbf{z}_{1,t}|z_{0,t})q(z_{0,t}) \tag{2.15}$$
$$(\textit{Jeffrey's Rule}) \implies = p(\mathbf{z}_{1,t}|z_{0,t})q(z_{0,t}) \tag{2.16}$$
$$(\textit{Bayes Theorem}) \implies \propto p(z_{0,t}|\mathbf{z}_{1,t})p(\mathbf{z}_{1,t})q(z_{0,t}) \ . \tag{2.17}$$

Thus, we first need to derive $p(\mathbf{z}_{1,t}|z_{0,t})$ which will allow not only the derivation of the joint multiscale distribution across all nodes, but also sample from the descendants given only the parent node. The indicator for each month t is ignored here for simplicity's sake.

$$p(\mathbf{z}_1|z_0) \propto p(z_{0,t}|\mathbf{z}_{1,t})p(\mathbf{z}_1) \tag{2.18}$$
$$\propto exp\left\{-\frac{1}{2}(\mathbf{z}_1 - \boldsymbol{\mu}_1)^\intercal v_1^{-1}(\mathbf{z}_1 - \boldsymbol{\mu}_1) - \frac{1}{2}(\mathbb{1}^\intercal \mathbf{z}_1 - z_0)^\intercal \delta_1^{-1}(\mathbb{1}^\intercal \mathbf{z}_1 - z_0)\right\} \tag{2.19}$$
$$\propto exp\left\{-\frac{1}{2}(\mathbf{z}_1 - \mathbf{m}_1)^\intercal \Lambda(\mathbf{z}_1 - \mathbf{m}_1)\right\} \tag{2.20}$$

where $\Lambda = (\mathbb{1}\delta_1^{-1}\mathbb{1}^\intercal + v_1^{-1})$ and $m_1 = \Lambda^{-1}(\mathbb{1}\delta_1^{-1}z_0 + v_1^{-1}\boldsymbol{\mu}_1)$. This result can then be used to recursively sample from each and every set of descendants. If one is interested in the closed form of the joint distribution of a parent node with its descendants, it can also be derived simply using Equations 2.17 and 2.20. This is demonstrated in Appendix C. For the purposes of the order simulator, being able to sample the descendents of each node given the model for most coarse node (described in Section 2.1) will be sufficient.

After obtaining $p(\mathbf{z}_1|z_0)$, the updated marginal model on the fine data is then obtained by integrating out the coarse random variable $z_{0,t}$,

$$q(\mathbf{z}_{1,t}) \propto \int p(z_{0,t}|\mathbf{z}_{1,t})p(\mathbf{z}_{1,t})q(z_{0,t})dz_{0,t} \tag{2.21}$$
$$\implies q(\mathbf{z}_{1,t}) = N(A\mu_{0,t} + \Lambda^{-1}v_1^{-1}\boldsymbol{\mu}_{1,t}, Av_0^{-1}A^\intercal + \Lambda^{-1}) \ , \tag{2.22}$$

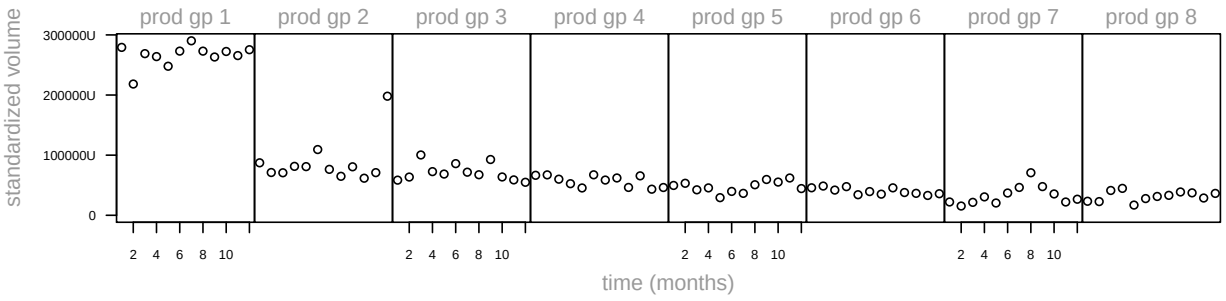
where $\Lambda = \mathbb{1}\delta^{-1}\mathbb{1}^\intercal + v_1^{-1}$ and $A = \Lambda^{-1}\mathbb{1}\delta^{-1}$.

2.2.1 Applied Multiresolution Model

The model for the coarse order volume per month presented in Section 2.1 will serve as a springboard to estimating the volume for each descendant node. Some exploratory analysis

is needed to determine which mean and variance specification from Equation 2.10 is best for some of the descendant nodes.

Figure 2.5: Monthly order volumes aggregated for each of 8 product groupings in 2013.



Each node of product groups tends to have similar auto-regressive features as observed in the overall aggregated volume (Figure 2.2). This can be seen in Figure 2.5 for 8 different product groupings over the course of 2013. With each descendent level of the tree, one would expect a larger variance in relationship to the volume being ordered.

The behavior for the descendant nodes is similar to that of the aggregated amounts. Thus, I use a model similar to Equation 2.2 for each of the product groups. Prior specifications similar to Equations 2.5 to 2.8 provide a decent model fit. One important feature in the multiresolution model is the variance δ in the linking distribution (Equation 2.11). Should the delta in the linking distribution be too large, then the variance for each node will shrink to 0. Thus the prior specification for δ is important. The prior distribution chosen is $\delta_i \sim Gamma(a_{\delta,i}, b_{\delta_i})$ for node i. Since the scale for each node is different, values for $a_{\delta,i}$ and b_{δ_i} were such that $E(\delta_i)$ is equal to 0.03% of the average volume for the node and $Var(\delta_i)$ is 0.06% of the average volume.

An example of the posterior mean and 95% credible intervals for BabyDry Tier 1 and each of its parent nodes across 24 months in the standardized units is seen in Figure 2.6. The multiscale model appears to fit well and provides estimates for the true demand of each product group and any subsequent aggregation of them according to the product tree described in Section 2.0.1.

Figure 2.6: Dynamic Linear Model across 24 months from most coarse to most fine level on a branch of the product tree.

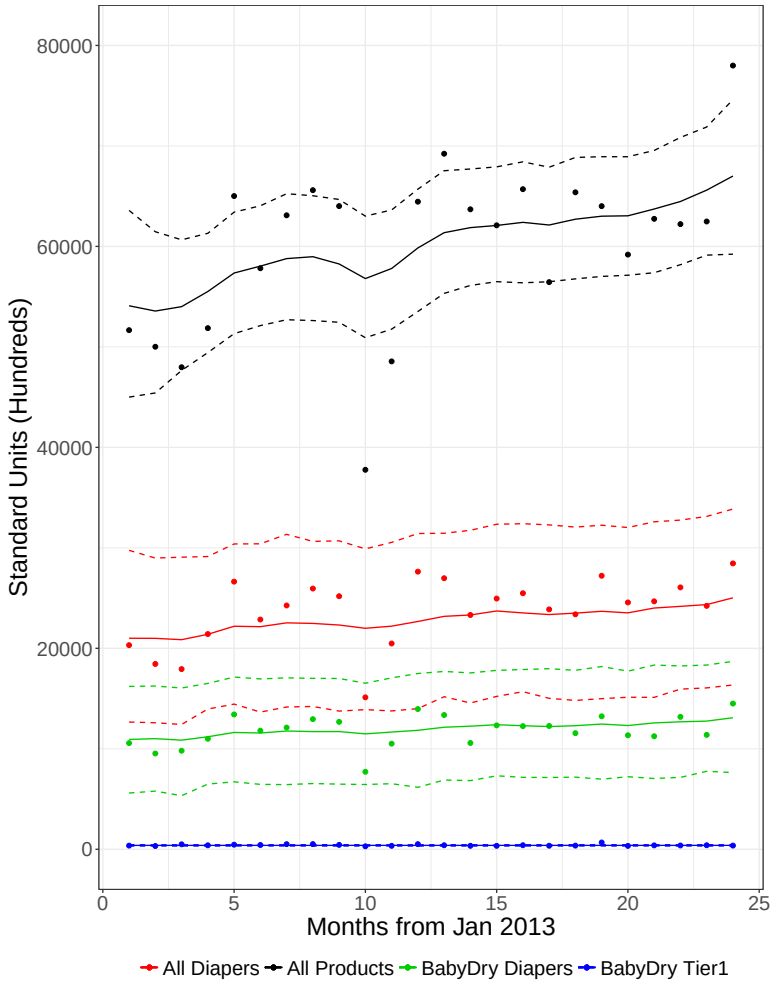
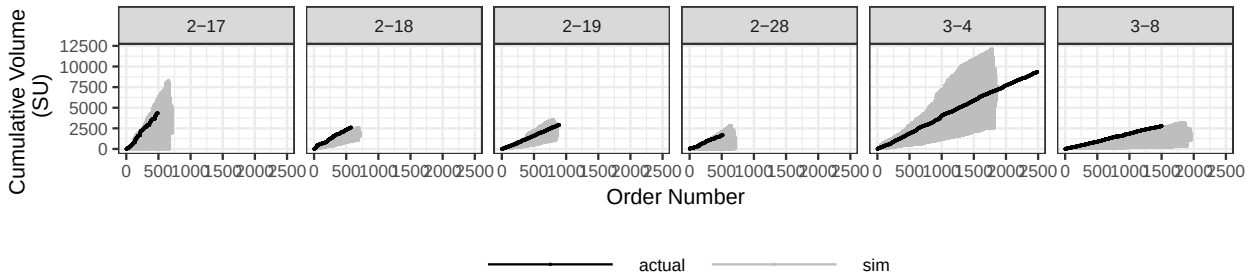


Figure 2.7 shows 50 simulations of cumulative order amounts across all customers for six different products compared to actual order amounts in December 2014. The amounts for the actual month appear to fit in well compared to the results of the 50 simulations. Products are referred to by the level on the product tree (2 or 3) and which node in that level (17, 18 ...). The one product with id "3-4" appears to have fewer number of orders simulated (x-axis) for the product than observed. Regardless of having fewer orders, how much of this product (y-axis) was ordered in aggregate seems to be near the median per order of the simulations.

Figure 2.7: 50 simulated order amounts (gray) for selected products compared to observed orders amounts (black) for December 2014.

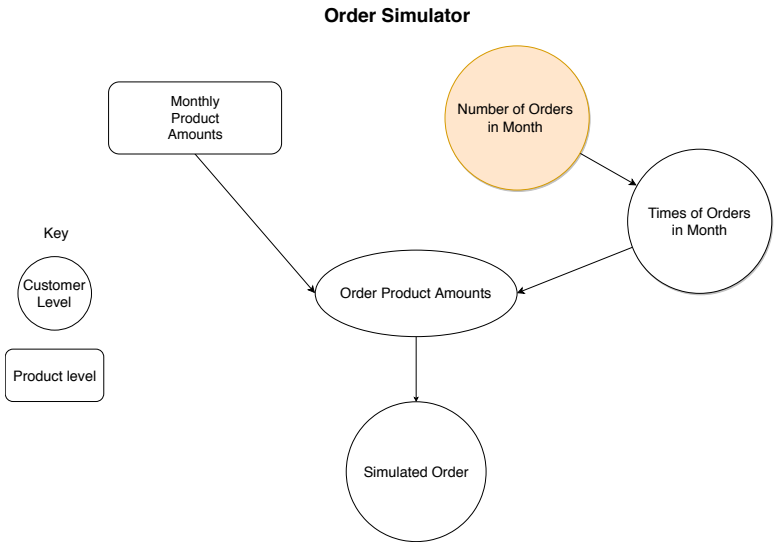


The model described in this chapter controls how high up the y-axis in cumulative order volume each product gets for the month. The number of times each product is ordered in a month (x-axis) is the outcome of the model for number of orders for a customer in Chapter 3. This jumps in order volume between orders draws on the model for amounts of each product in each order, which is addressed in Chapter 5.

Chapter 3

Number of Orders Each Month

Figure 3.1: Diagram of the four models that combine to create the order simulator. This chapter covers highlighted portion.



A useful end-to-end order simulator needs to take into account differences in each customer. In the previous chapter the monthly amounts for each product were simulated. Those were aggregate amounts across all customers, but specific to each product. This chapter will focus entirely on the customers and ignore the products. A customer may differ from one to another in its order behavior by

1. how many times does it order,
2. how often it orders,

3. which products it orders,
4. how much it orders for each product in each order, and
5. how consistent the customer is to the previous items?

This chapter will attempt to answer item 1, "How many times a customer orders?" and related item 5, "How consistent each customer is with respect to how many times it orders?" There is a difference between how many times a customer orders and how often it does. The former will be addressed in this chapter by addressing how many orders to expect in a month. The latter will be addressed in Chapter 4 which looks at the spacings between those orders in a month for a given customer.

3.1 Data Exploration

The number of times in a month a customer orders varies widely and is heavily right skewed. In over fifty percent of all possible customer and month combinations there are no orders placed. Of the 941 customers in the dataset, about half are customers that order on average less than once per month. On the contrary, as seen in Table 3.1, 10.4% of all unique customer by month combinations consist of 11 or more orders in a month with one the maximum being 598 orders in one month. This is largely due to different ordering systems from customer to customer. Some customers are large and have automated systems to order when their own inventory gets low, while others place orders as needed with sales representatives.

Table 3.1: Frequency of the number of orders in a month by customer (24 months x 941 customers).

0	1	2	3	4	5	6	7	8	9	10	11+
11,937	2,672	1,433	984	816	590	508	415	390	311	195	2,333

Max Orders in Month by a Customer: 589

The differences between the customers necessitate that the order simulator be able to simulate different behavior depending on the customer. When it comes to the number of orders in a month, each customer will have a model independent that of other customers.

3.2 Simple Model

An initial approach involves a Poisson process where each customer c has a true number of orders n in month t such that

$$n_{t,c}|\lambda_c \sim Poisson(\lambda_c) \ . \tag{3.1}$$

The likelihood laid out in Equation 3.1 assumes each customer is independent of one another and each month is independent of one another. If these principles hold true, Occam's Razor would suggest to keep the model as is. In order to evaluate whether or not that is true, a conjugate prior analysis is performed. Assume each λ_c has a prior distribution

$$\lambda_c \sim Gamma(a_\lambda, b_\lambda) \ . \tag{3.2}$$

In assigning values to the prior distribution, the goal was to keep the prior mean approximately equal to that of the average number of orders in a month for all customers. This is about 4.81 orders per month. This is preferred to a Jeffrey's prior or another completely non-informative prior. A completely non-informative prior will likely be less accurate in simulating future orders because there are only 24 months in the dataset. That is not a lot of data points for each individual customer. By relying on this overall mean there is some information shared across all customers, in form of the prior distribution. In order to set the prior parameters a_λ and b_λ, if the desired prior mean is known, then a desired prior variance will define the parameters. The prior variance should be somewhat large in order to not be too restrictive for the few customers with very large number of orders. For this reason the prior variance is chosen to be 400, which is close to the observed variance of 403. This leaves prior parameters of $a_\lambda = 0.057$ and $b_\lambda = 0.012$.

Figure 3.2: Posterior predictive 90% interval on number of customer orders.

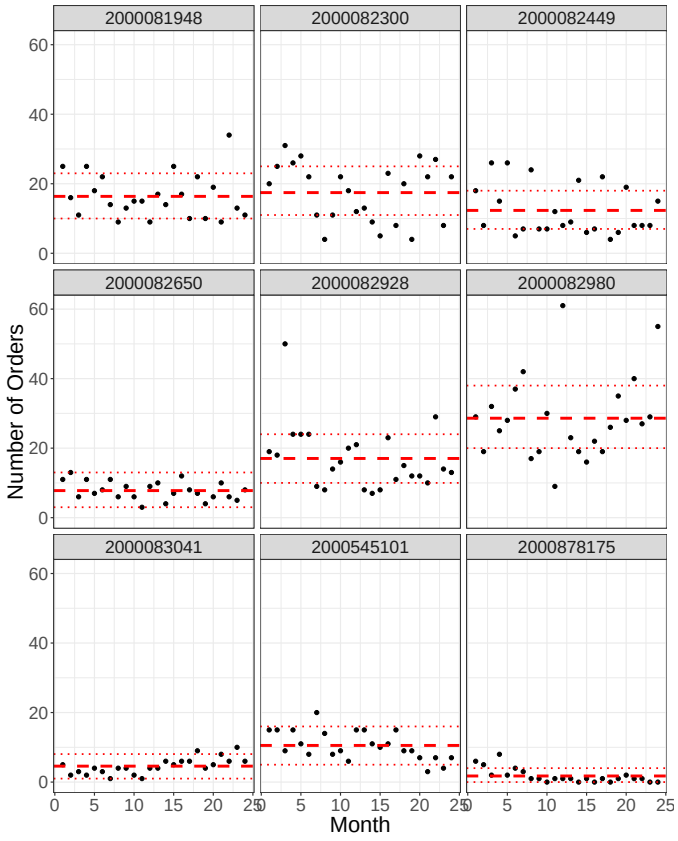


Figure 3.2 shows the posterior predictive mean and 90% intervals for nine different customers. The likelihood is a Poisson distribution, thus the upper and lower bounds of the posterior predictive distribution interval are inclusive non-negative integer values. In the plot there is a variety of number of orders in a month. There are depicted customers that order about thirty times a month, some about twenty, others about ten, and some about once a month. For the customers with about 0-4 orders each month, the posterior predictive intervals have very good coverage. For those who order more, there are appears to be worse coverage. This could be a function of the prior distribution being too informative and narrow, but since the variance on the prior distribution of λ_c was large, this is likely due to other factors such as variance due to time dependency. For example in the 2nd row and 3rd column of Figure 3.2, the customer has some months of low order numbers and gradually orders more until ordering a lot in some months. This model only focuses on the mean, leaving several months more than expected outside the predictive interval. Recall that over half of all order numbers in a month is 0, thus the intervals with low coverage is not a representative sample of all customers, since most will be similar to the customers that order 0-4 times a month. Despite most customer's having a model with good coverage, there are some that show a real time series element to their ordering, as recently discussed. In order to make the order

simulator better, this will need to be taken into account.

3.3 Poisson Time Series Model

Using Bayesian hierarchical models, adding a time dependency to the previous model is not too difficult. The likelihood will be similar to Equation 3.1, with an added index t on the Poisson parameter λ for the month of reference,

$$n_{t,c}|\lambda_{t,c} \sim Poisson(\lambda_{t,c}) \ . \tag{3.3}$$

The prior distribution will then involve the time dependency and add a level of hierarchy as follows,

$$\lambda_{t,c}|\gamma \sim Gamma(\frac{1}{\gamma}\lambda_{t-1,c}, \frac{1}{\gamma}) \tag{3.4}$$

$$\gamma \sim Gamma(a_\gamma, b_\gamma) \ . \tag{3.5}$$

The prior on $\lambda_{t,c}$ (Equation 3.4) is an autoregressive or AR-1 prior. This follows the Markov property, meaning that the distribution for the next value is dependent only on the previous value. This is seen in the 1st and 2nd moments of the prior distribution. The prior expectation is $E(\lambda_{t,c}) = \lambda_{t-1,c}$ and $Var(\lambda_{t,c}) = \gamma\lambda_{t-1,c}$. Thus the best guess for the next month's value of the parameter is the previous month's value. This parameterization also gives two unique interpretations of γ. The first is that γ may be used as a dispersion parameter. Using a dispersion parameter allows the Poisson model to have a different mean than the variance. The second interpretation is one of a correlation type parameter. It does not have the same interpretation as Pearson's correlation coefficient (Galton (1877) and Pearson (1895)), but has a similar interpretation to the coefficients in an autoregressive model. For example, the smaller the value of γ then $\lambda_{t,c}$ is more likely to be close to $\lambda_{t-1,c}$, and in turn $n_{t-1,c}$. The larger value of γ suggests that $\lambda_{t,c}$ is not as likely to be as influenced by $\lambda_{t-1,c}$.

3.3.1 Estimation

The model is fit via Gibbs sampling. The Gibbs sampler is straightforward as the complete conditional posterior distribution of $\lambda_{t,c}$ is conjugate and thus is available via closed form solution,

$$\lambda_{t,c}|\lambda_{t-1,c},\gamma \sim Gamma(n_{t,c} + \frac{1}{\gamma}\lambda_{t-1,c}, 1 + \frac{1}{\gamma}) \ , \tag{3.6}$$

where $n_{t,c}$ is the number of orders made in month t for the customer c. There also has to be an initial starting value of $\lambda_{0,c}$. I set $\lambda_{0,c} = 4.81$ because that is the overall mean for all customers as discussed in Section 3.2. For a customer with no previous order amounts, the prior value of $\lambda_{0,c}$ allows for a starting point in simulations.

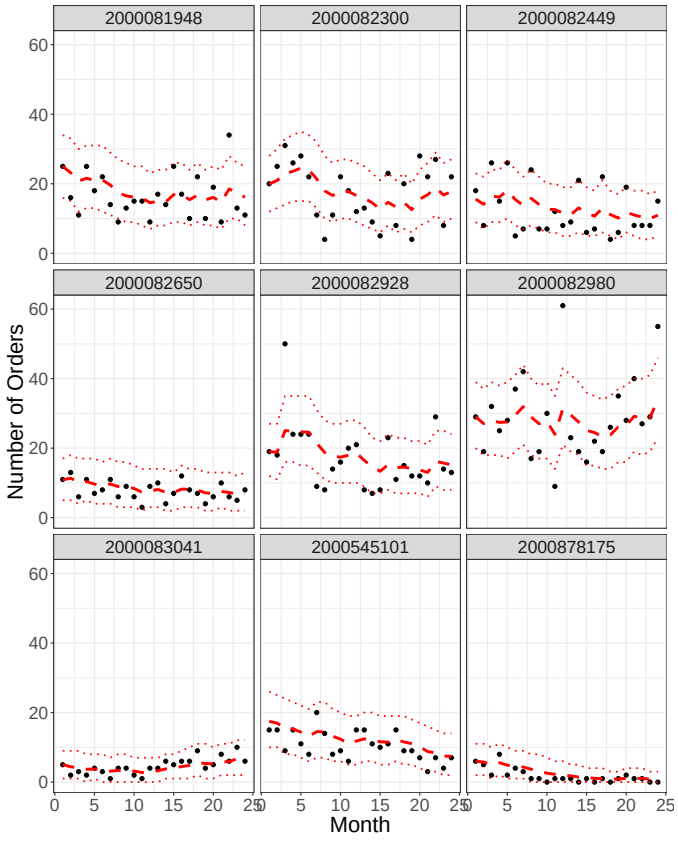
Figure 3.3.1 shows the posterior predictive distributions for those same nine customers as seen in Figure 3.2. Those same customers see many of the observed number of order values fall within that 90% interval that were previously outside. There are clear trends upwards and downwards for several of the customers, while others appear fairly constant. This more flexible model does a better job capturing the values for each customer. There are likely seasonal effects for many customers, but with only 24 months, that is hard to capture. With more data those season effects will need to be incorporated in the prior parameters of $\lambda_{t,c}$. This could be done via another level of hierarchy such that

$$\lambda_{t,c}|\lambda_{t-1,c},\gamma \sim Gamma(\frac{1}{\gamma}a^{\star}, \frac{1}{\gamma}) \tag{3.7}$$

$$a^{\star} = \lambda_{t-1,c} + G\boldsymbol{\theta} \ . \tag{3.8}$$

Here G is a matrix of potential seasonal or other external predictors that could impact one's expectation for the number of orders for customer c. $\boldsymbol{\theta}$ is the vector of estimable parameters associated with G.

Figure 3.3: Time-varying posterior predictive 90% interval on number of customer orders.



The hierarchical structure of the this time-dependent model allows the direct calculation of the posterior predictive mean and variance for the number of orders next month $n_{t+1,c}$ given the previous month's value. These are derived via the conditional expectation and variance formulas. The posterior predictive mean,

$$\begin{aligned} E(n_{n+1,c}) &= E(E(n_{t+1,c}|\lambda_{t+1,c})) && (3.9)\\ &= E(\lambda_{t+1,c}) && (3.10)\\ &= \frac{\gamma\lambda_{t,c}}{\gamma} && (3.11)\\ &= \lambda_{t,c}\ , && (3.12) \end{aligned}$$

and the posterior predictive variance,

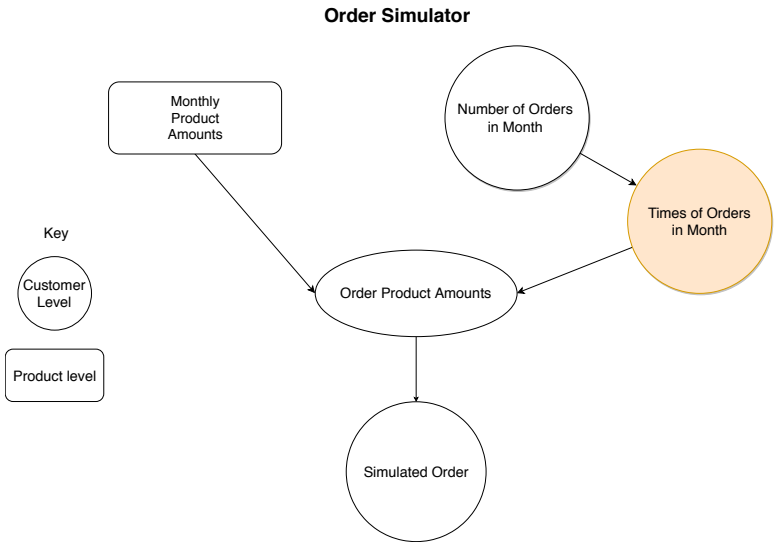
$$\begin{aligned} Var(n_{n+1,c}) &= E(Var(n_{t+1,c}|\lambda_{t+1,c})) + Var(E(n_{t+1,c}|\lambda_{t+1,c})) && (3.13)\\ &= E(\lambda_{t+1,c}) + Var(\lambda_{t+1,c}) && (3.14)\\ &= \lambda_{t,c} + \lambda_{t,c}\gamma && (3.15)\\ &= \lambda_{t,c}(1+\gamma)\ . && (3.16) \end{aligned}$$

Thus the posterior predictive distribution for next month's number of orders $n_{t+1,c}$ is a quasi-Poisson likelihood with mean and variance as shown in Equations 3.12 and 3.16. Using these equations the order simulator is capable of simulating the number of orders in the next month for each existing or new customer.

Chapter 4

Order Time Model

Figure 4.1: Diagram of the four models that combine to create the order simulator. This chapter covers highlighted portion.



4.1 Introduction

We appeal to Bayesian multilevel time-series statistical models, which allow predictive distributions of both order times and amounts from all downstream customers to be estimated. It is from these predictive distributions we simulate the orders for all of products from a large manufacturer with many customers relying on real-time data as it comes in, not solely on the typical assumptions of an auto-regressive process, historical averages, or exponential smoothing in order lead time that as previously mentioned in Chapter 1 that are typical in

supply chain demand forecasting.

Our dataset has orders from companies in several different countries in Western Europe comprising of small corner markets to large international retailers. The modern economy consists of global supply chains which are more difficult to manage than domestic ones due to various factors such as differing cultures to less effective demand forecasting (Dornier et al. (1998), MacCarthy and Atthirawong (2003), and Meixell and Gargeya (2005)). This research improves demand forecasting based on actual observed data in a complex supply chain. The structure of our model allows integration of a more complex demand forecasting model to completely simulated supply chains such as in Pires et al. (2017), who used real data to build a hybrid system that combined Bayesian modeling with discrete event simulation. While Pires et al. (2017) use empirical data to simulate orders as part of a complete supply chain simulation to better understand the dynamics of inventory, demand, and service has many nice features, the order model is, like previous ones, simplistic. In the order model of Pires et al. (2017), assumptions about order times following a uniform distribution are made. We relax such assumptions and propose a more flexible approach that allows for uniform behavior, should the data support it, and non-uniform behavior should the observed orders show sufficient evidence. More accurate simulation and modeling of all customer's orders will allow better planning of production runs, better manage inventory levels and ensure timely delivery to customers.

4.2 Order Overview

Our order data consists of hundreds of customers who order many different products manufactured at a particular plant. I will focus on mainly on diapers here, although the methods in practice are applied to orders containing any combination of the products offered. Diapers are the most commonly ordered product in this dataset, which is why they were chosen as the example for modeling the order times. Each order is a part of sale to a customer and includes each product, its tier and size, as well as the day and time it was ordered and the amount of the order. The order amounts are scaled to a standardized size to compare order sizes across products and tiers. For many customers, the orders are auto-generated when their inventory runs low, meaning that there is wide variety of behaviors of the timings of the orders. Wholesale chains may need to order just seconds apart while small retailers may not make more than one or two orders per month, spaced weeks apart.

Table 4.1: Order data

Customer	Group	Tier	Size	Order Time	Order Amount
081948	Diapers	A	4	2013-01-02 07:01:34	29.55
082928	Diapers	B	4	2013-01-02 07:01:34	55.99
081948	Diapers	A	5	2013-01-02 07:01:40	27.22
545101	Diapers	C	1	2013-01-02 07:01:44	93.34
545101	Diapers	D	4	2013-01-02 07:01:48	23.95
081948	Diapers	B	2	2013-01-02 07:01:49	153.62

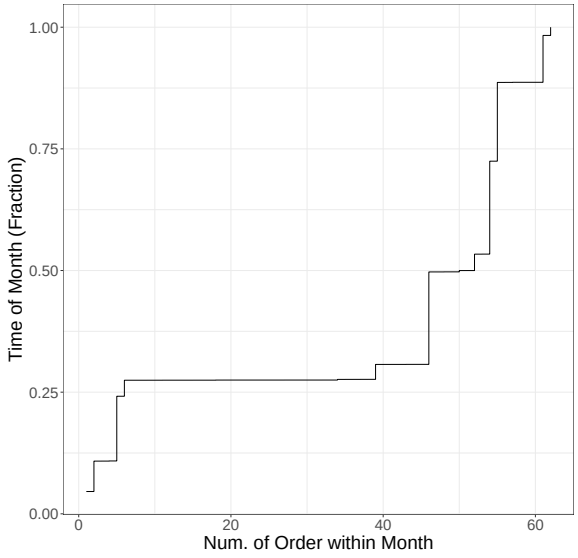
Most analysis previously mentioned focuses on modeling the amount of an order in a given time frame, months, weeks, days, or hours. Such an approach is not conducive when trying to model when an individual customer will order but is conducive when solely focused on aggregate amounts of a product in a given time frame regardless of the customer ordering. Even still, such time periods are a discretization of a continuous variable, time. Information is lost when discretization occurs, thus our model will focus on order times as a continuous process with an amount attached to each order coming in at a given time.

The times during a month when a customer orders a subset of products can be thought of as fractions of the month as a whole. This interpretation makes observed months of varying lengths – January with 31 days and February with 28 days– on the same scale and can be represented by distributions with positive support on the $[0, 1]$ interval. When thinking about random variables that take on values between $[0, 1]$, Beta distributed random variables come to mind. Instead of dealing with the raw order times we model the spacings between the order times. If the raw order time fractions follow a Beta distribution, the collective spacings follow a multivariate Dirichlet distribution. The Dirichlet distribution has properties that link it not only to Beta random variables but also to order statistics of a random variable that follows a uniform distribution. Since this analysis is an expansion of the work done by Pires et al. (2017), using uniform order times, this approach can be thought of as a relaxation of the Pires et al. (2017) approach. Additionally, Dirichlet random variables are a finite realization of the Dirichlet process, allowing for more flexible distribution functions.

Similar to how a Dirichlet process is a distribution across many distributions, the cumulative values of a multivariate discrete Dirichlet random variable can be thought of as a discrete cumulative distribution function. In figure 4.2 we show an empirical CDF of order times for a typical customer's orders in a given month. While some orders are spaced far apart, there are about 30 orders within minutes of each other about a quarter of the way through the month. This may be caused by many customers having automatic systems to command orders. This behavior is common and even reflects retailers patterns of periods of sales in which inventory is dumped quickly and needs to be replenished while at other times sales are slow and thus no orders are needed to the manufacturer for days or even weeks on end. While

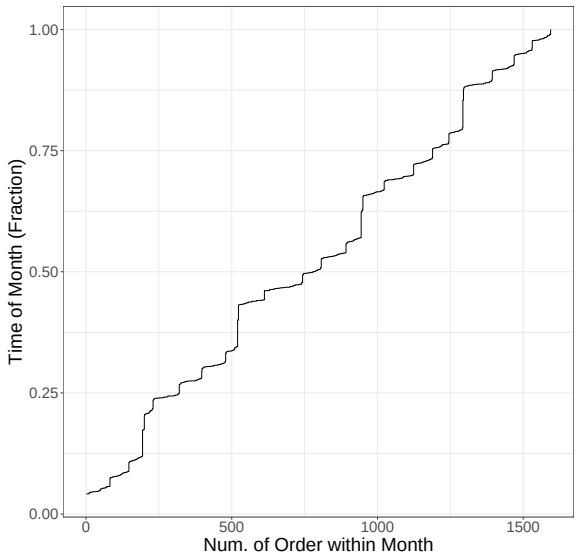
a Beta distribution may be able to have high density in a certain area, it would be unable to fit well multiple high number of order times in a month since it is a unimodal distribution.

Figure 4.2: Size 4 diaper orders for customer# 2002315820 - January 2013



While one customer's order behavior may not be very regular, collectively all of the customers exhibit very regular behavior as can be seen in Figure 4.3. Each vertical line is a period of time when there are no products ordered for an extended period of time. One will notice in this month 4 large time-lengths with little to no orders, each of those separated by 4 smaller time periods with sparse orders. As the reader likely presumed, these longer order-free times are weekends and the shorter ones are weeknights. Properly adjusting these differences between non-business and business hours in practical settings will be discussed later. For argument's purposes we will assume, for the time being, there are no non-business hours that would inherently make a certain time period any more or less likely to observe orders from any customer.

Figure 4.3: Order times for highest demand diaper brand x size (all customers) - January 2013



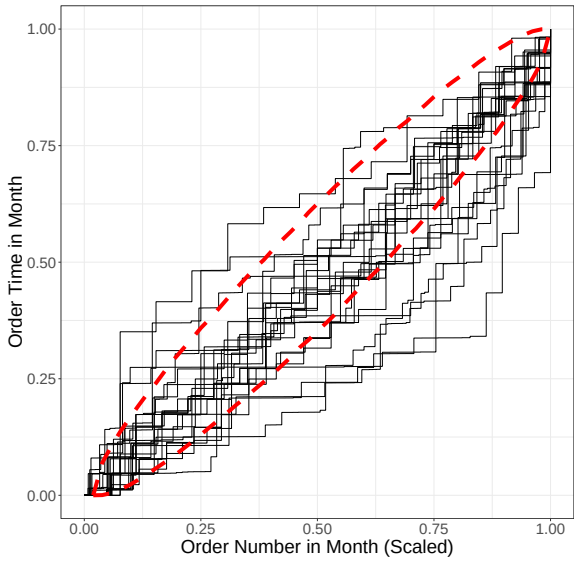
We desire to construct a model for each product that has the following properties. First, it should quantify the order behavior for each individual customer as well as the product as a whole. Second, the model should allow straightforward prediction and simulation of the process that will allow optimization routines to be performed similar to that of Pires et al. (2017). Third, the model should be able to draw information from the fine (customer) level to inform the coarse (product) level as well as from the coarse to the fine level. The inclusion of both of these levels will make the model robust for differing behavior than would be implied with a model solely on the coarse level or solely on the fine level.

4.2.1 Uniform Order Timings Model

The simple method used in Pires et al. (2017) assumes the order times follow a uniform distribution on the unit line. We hypothesize that order times do not follow a uniform distribution and that a uniform distribution does not adequately account for the variance in order times. Suppose we simulate 10,000 months of orders following the uniform distribution with each month containing 50 orders. In each month we simulate, we take the order statistics. There is evidence that the simulated 95% confidence interval (actually looks like a "football") of

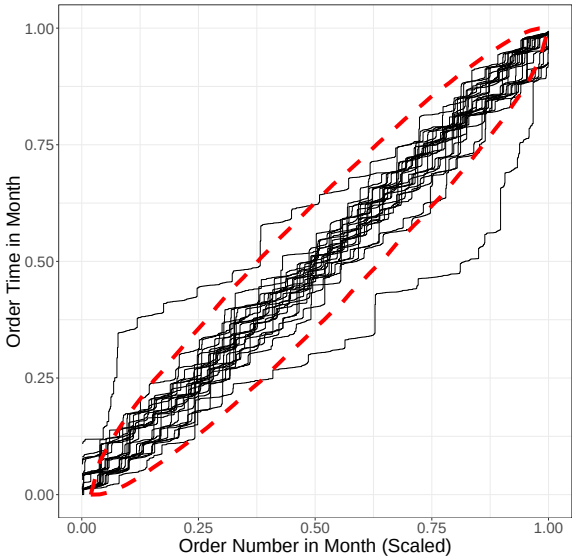
order times results in under-coverage of our actual order data. An example of one customer is seed in Figure 4.4 where about 25% of all order times fall outside the 95% simulated interval.

Figure 4.4: Simulated cumulative uniform random variable and 95% intervals compared to an observed customer's order times.



When combining all the customer times together for each month, the uniform model has better coverage. Figure 4.5 shows all orders in each month for the most ordered product. Holistically the uniform model seems a good fit, perhaps maybe even the 95% simulated confidence interval is too wide for the order times of this product when ignoring the customer labels. We seek a model that allows either the uniform model or something with the flexibility to have greater or smaller variance should the individual customer or the customers collectively as a whole, demonstrate such behavior.

Figure 4.5: Simulated cumulative uniform random variable and 95% intervals compared to an all observed customers' order times.



4.3 Dirichlet Process Model

Two well-known distributions on a random vector x where $\sum_{i=1}^{K}(x_i) = 1$ and $x_i \in (0,1) \forall i$ are the multivariate uniform and the Dirichlet distribution. A random vector, $\mathbf{x}$ is distributed as a multivariate uniform with the probability density function,

$$f(\mathbf{x}) = \prod_{i=1}^{K} f(x_i) \text{ where}$$
$$f(x_i) = \begin{cases} 1 & \text{if } x_i \in [0,1] \\ 0 & \text{otherwise.} \end{cases} \tag{4.1}$$

This is a simplification of the uniform distribution where the support only goes from 0 to 1. A random vector x where $\sum_{i=1}^{K}(x_i) = 1$ and $x_i \in (0,1) \forall i$ follows a *symmetric* Dirichlet distribution with a probability density function,

$$f(x_1, x_2, \ldots, x_{K-1}) = \frac{\Gamma(\alpha(K-1))}{\Gamma(\alpha)^{(K-1)}}(\prod_{i=1}^{K-1} x_i)^{\alpha-1} \tag{4.2}$$
$$x_K = 1 - \sum_{i=1}^{K-1}(x_i)$$

where α is the singular concentration parameter for all x_i. This singular concentration parameter simplifies the Dirichlet distribution and ties in nicely to the the Dirichlet Process. Antoniak (1974) and Ferguson (1983) expounded upon the previous Dirichlet work of Ferguson (1973) to develop the Dirichlet Process and Dirichlet Process mixture models. The Dirichlet Process is a generalization of the Dirichlet distribution. Conversely the Dirichlet distribution can be thought of as the distribution for which a finite realization of the Dirichlet Process follows. As mentioned in Section 4.2, the order times themselves can be thought of as its own discrete distribution. An additional interpretation of the Dirichlet process is that of a distribution across distributions. Thus, the set of sequential order times in a given month can be thought of as a finite realization of a Dirichlet Process.

There are several ways to construct the Dirichlet distribution, giving it many interpretations. Due to usefulness of the Dirichlet Process in the clustering of "big data", the Chinese Restaurant Process and Stick Breaking Process are often used to set-up the Dirichlet Process. The finite Dirichlet Process, or Dirichlet distribution can also be constructed using the process of order statistics of uniformly distributed random variables.

Suppose ,

$$u_i \sim U(0,1) \text{ for } i = 1, 2, \ldots, n. \tag{4.3}$$

Then the order statistics are defined as

$$u_{(1)}, u_{(2)}, \ldots, u_{(n)} \ . \tag{4.4}$$

We then define the spacings, w_i between the order statistics,

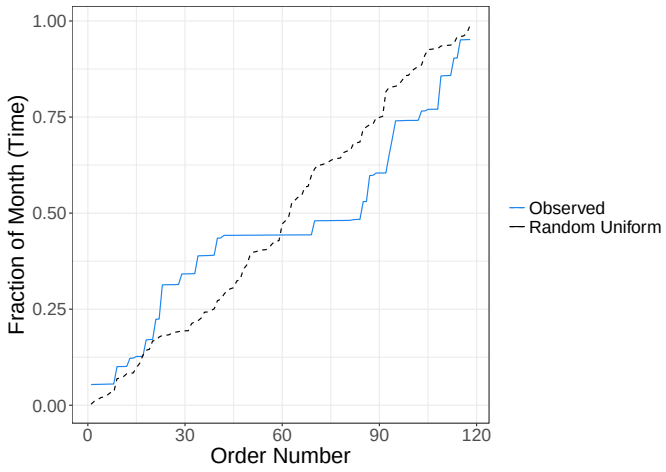
$$\begin{aligned} w_1 &= u_1 \\ w_i &= u_i - u_{i-1} \text{ for } i = 2, 3, \ldots, n \\ w_{n+1} &= 1 - u_n \end{aligned} \tag{4.5}$$

The joint distribution of all of the uniform order statistic spacings can be rewritten as

$$\begin{aligned} f(\mathbf{w}) &= n! f(w_1) f(w_2) \ldots f(w_{n+1}) \mathbb{1}(0 < w_1 < w_2 < \ldots < w_{n+1}) \\ &= n! \mathbb{1}(0 < w_1 < w_2 < \ldots < w_{n+1}) \end{aligned} \tag{4.6}$$

which is equivalent to the probability density function of a random vector that follows a Dirichlet distribution with parameter vector $\boldsymbol{\alpha} = \mathbf{1}_{n+1}$, where $\mathbf{1}$ is a vector of 1's of length $n+1$.

Figure 4.6: Order time fractions for customer in February 2013 and random uniform times



Thus, $\mathbf{w}$ follows a *symmetric* Dirichlet distribution, with $E(w_i) = i/n$ and $Var(w_i) = \frac{n}{(n+1)^2(n+2)}$. Since in the order data for the number of orders in a month (n_t) varies both within customer and between, we propose a more flexible specification of the symmetric Dirichlet Distribution (Equation 4.2) where

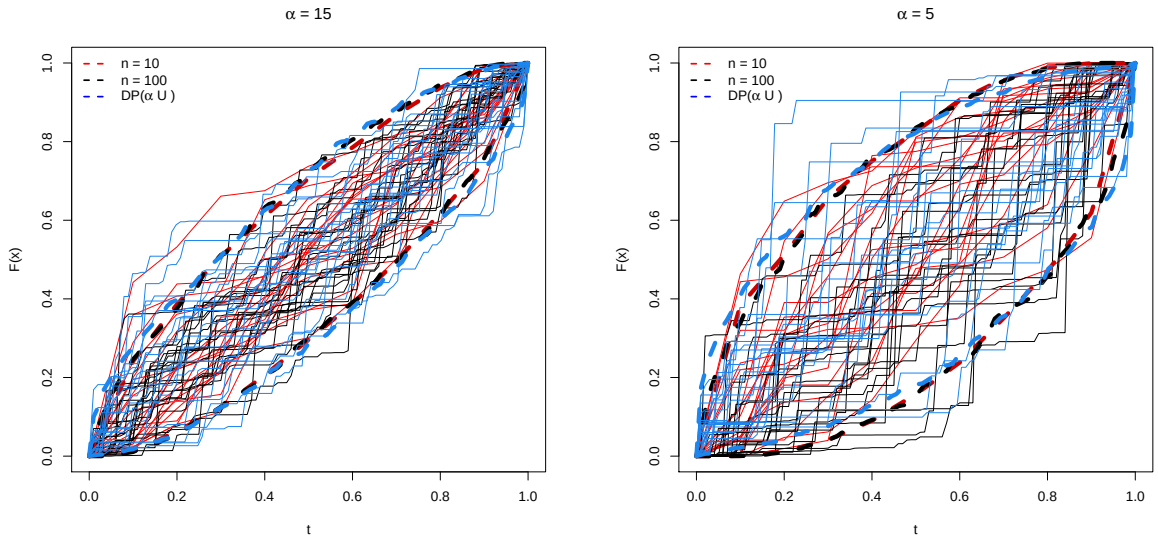
$$\mathbf{w} \sim Dir(\frac{\alpha}{n_t + 1}\mathbf{1}_{n+1}) \ . \tag{4.7}$$

Under this parameterization, α determines how concentrated the data are given the dimension of the vector $\mathbf{w}$, while still having a uniform base distribution. Should $\alpha = n_t + 1$, then we are left with uniformly distributed order times. Should $\alpha > n_t + 1$, then the variance on the observed realizations of the Dirichlet Process will be smaller than that of a uniform distribution. Another way to think of this is that the values generated from this process are

more regular, or evenly spaced. Conversely, should $\alpha < n_t + 1$, then the variance on the order timings will be larger (less regular) than that of the uniform distribution.

This parameterization keeps the symmetric Dirichlet Distribution's feature of a single concentration parameter regardless of the dimension of finite realization of the Dirichlet Process. In Figure 4.7 we show comparisons of draws of a random vector $\mathbf{w} \sim Dir(\frac{\alpha}{n+1}\mathbf{1}_{n+1})$ with different dimensions $n = 10$ and $n = 100$. This is compared to draws of $\mathbf{w}$ from a Dirichlet Process ($DP(\alpha G_0)$), constructed through the stick-breaking process, with the base distribution equal to a standard uniform ($G_0 = U(0, 1)$). This visually shows how with this symmetric parameterization – $Dir(\frac{\alpha}{n+1}\mathbf{1}_{n_1})$, α, regardless of $n = 10$, $n = 100$, or $n = \infty$ (the Dirichlet Process), has the same interpretation as a concentration parameter.

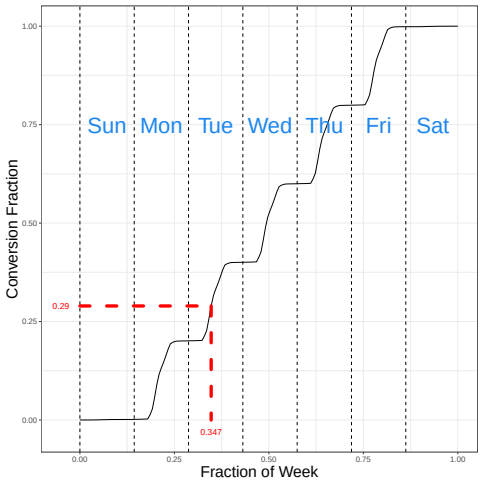
Figure 4.7: Simulated Dirichlet random vectors (and 90% intervals) in proposed parameterization compared to simulated Dirichlet Processes with a uniform base distribution.



4.3.1 Base Distribution & Non-business Hours

In the previous section, the connection between our symmetric parameterization of the finite Dirichlet distribution and the Dirichlet Process with a uniform base distribution was shown. Up to this point the issue of non-business hours was temporarily ignored. I further propose a different base distribution to account for the weekend and night effects. This base distribution can also be viewed as a transformation of our order times to order times following a uniform base distribution. As seen in Figure 4.3 it is apparent that the rate of orders decreases during non-business hours (nights, weekends, and holidays).

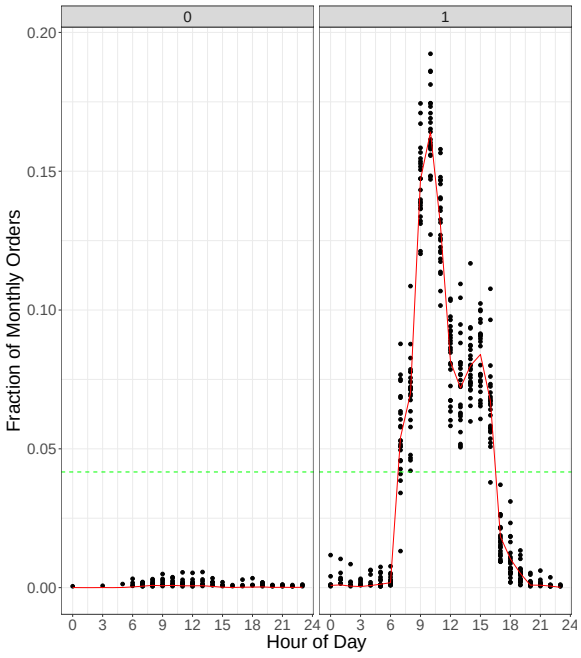
Figure 4.8: Conceptually adjusting for nights and weekends.



An example of how this adjustment for business and non-business hours could be done is shown in Figure 4.8. This method to account for non-business hours essentially will condense periods of non-business hours on the 0 to 1 line and expand the business hours to make up for it. In all simplicity the goal is to create a mapping function unique to each month to evaluate order times on the same 0 to 1 time scale.

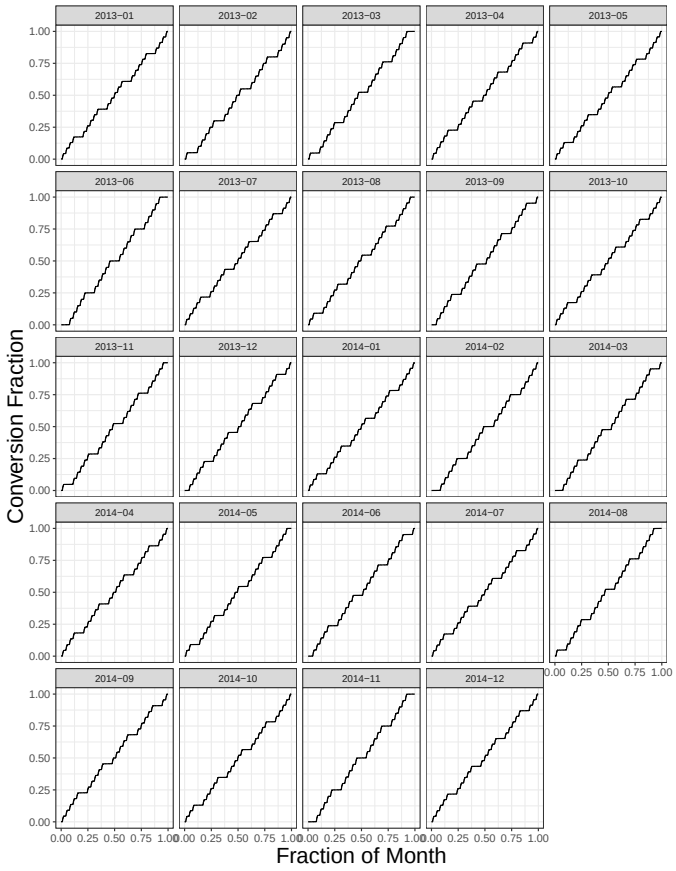
Let $r_{i,j}$ be the fraction of the monthly orders made of a certain product during hour i of day-type j, where $j = 0$ for a non-business day and $j = 1$ for a business day. In Figure 4.9 we see the varying hourly rates during both business days and non-business days.

Figure 4.9: Hourly order rates (business and non-business days).



Suppose we have a prior belief about the fraction of orders coming in each hour of the day informed by experts most familiar with order patterns. A conjugate analysis of these hourly fractions involves the prior distribution of fractions, $\mathbf{r} \sim Dir(\alpha_r)$ and a likelihood $\mathbf{x}|\mathbf{r} \sim Multinomial(\mathbf{r})$ where $\mathbf{x}$ is a multivariate vector denoting which hour in which day-type the order took place. The selection of α_r is intuitive if using expert opinion, if none exists, then thought will need to be taken to choose a hyper-prior on α_r, knowing that orders come in less frequently during non-business hours. The posterior distribution for the fractional rates is thus $\mathbf{r}|\mathbf{x} \sim Dir(\boldsymbol{\alpha}_r^\star)$ where $\alpha_{r_{i,j}}^\star = \alpha_{r_{i,j}} + n_{i,j}$. Using the posterior mean for each hourly rate, we construct base distributions for each month (see Figure 4.10) that accounts for the inherent differences in the chance that an order comes in during business hours versus non-business hours.

Figure 4.10: Base distributions for each month in dataset.



The set of base distributions observed in Figure 4.10 nicely accounts for the long stretches of time with no orders during non-business hours as seen in Figure 4.3. In Figure 4.12 we see the distribution of order times after accounting for nights and weekends, a distribution that for all customers resembles a uniform cumulative distribution function.

Figure 4.11: Comparing the raw order times for all customers (black) to the adjusted times (blue) for four months. The adjusted times no longer show obvious jumps during non-business hours.

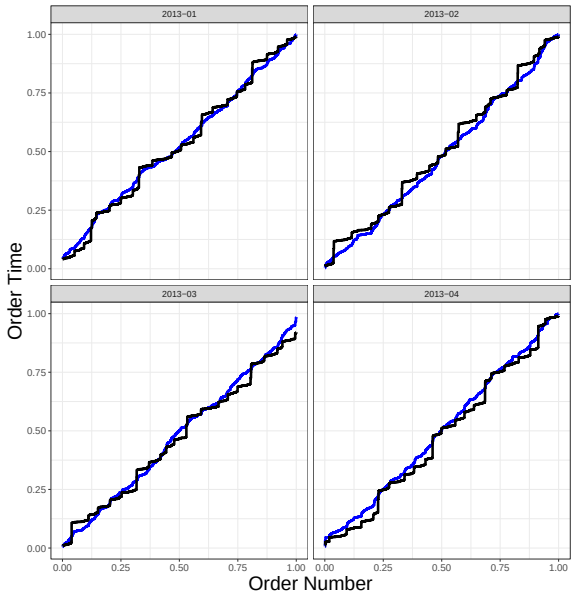
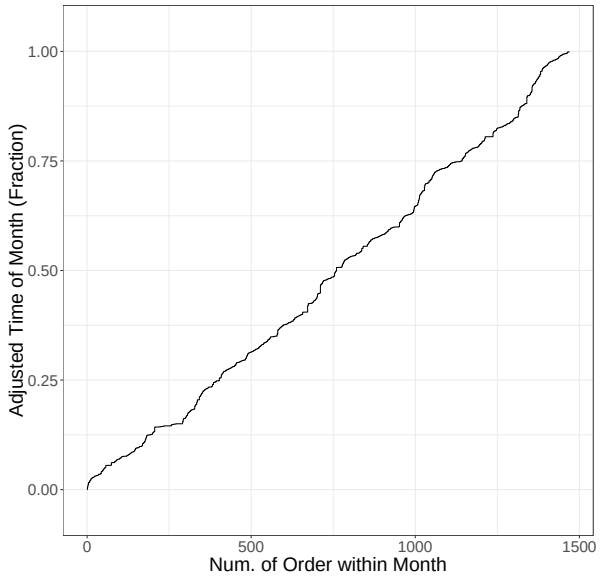


Figure 4.12: Order times for all customers of highest demand diaper brand x size (January 2013).



4.3.2 Posterior Estimation of Concentration Parameter

After adjusting for the hour in which an order comes in, we now return to posterior estimation of concentration parameter as outlined in Equation 4.7. For a given customer c, there are up to T months of orders. Each month has n_t orders and n_t times at which those orders were placed.

Choice of Prior Distribution

Using the $\frac{\alpha}{n_t+1}\mathbf{1}_{n_t+1}$ parameterization of the Dirichlet concentration parameter, each customer's single parameter determines how regular its order times are. A well-defined prior distribution with positive support ($\alpha > 0$) should result in a proper posterior. Some sensible choices of prior distribution would be the Gamma distribution, Jeffreys prior, or a uniform prior with a large enough support range.

One purpose of our model is to allow for the flexibility that the adjusted order times not be

restricted to follow uniform order statistics. The choice of a prior distribution can try to be non-informative or informative. Should the informative prior be desired, one approach is to put most of the mass on the null model, which is the uniform model where the concentration parameter α is close to the average number of order time spacings ($n_t + 1$) in each month. The Gamma prior, with mean near our prior belief about α and variance reflecting how confident *a priori* we are in that prior belief, fits these requirements.

On the other hand, should the desire be to have a non-informative prior on α, the Jeffrey's prior $\pi_J(\alpha)$ or another flat prior such as a uniform would allow the data to dominate the posterior analysis. The Jeffreys prior, which exists, is proper in this univariate case. The Jeffreys prior is defined as

$$\pi_J(\alpha) \propto I(\alpha)^{\frac{1}{2}} \ .$$

$$\pi_J(\alpha) \propto I(\alpha)^{\frac{1}{2}} = \left[\sum_{t=1}^{T} \frac{1}{n_t} \frac{d^2}{d\alpha^2} \log\left(\Gamma\left(\frac{\alpha}{n_t}\right)\right) - T\frac{d^2}{d\alpha^2}\log(\Gamma(\alpha))\right]^{\frac{1}{2}} \tag{4.8}$$

The derivation of Equation 4.8 prior can be found in the Appendix (Section A). Posterior estimation can be done by various methods including Gibbs sampling, or even direct evaluation of the proportional posterior (Equation 4.10).

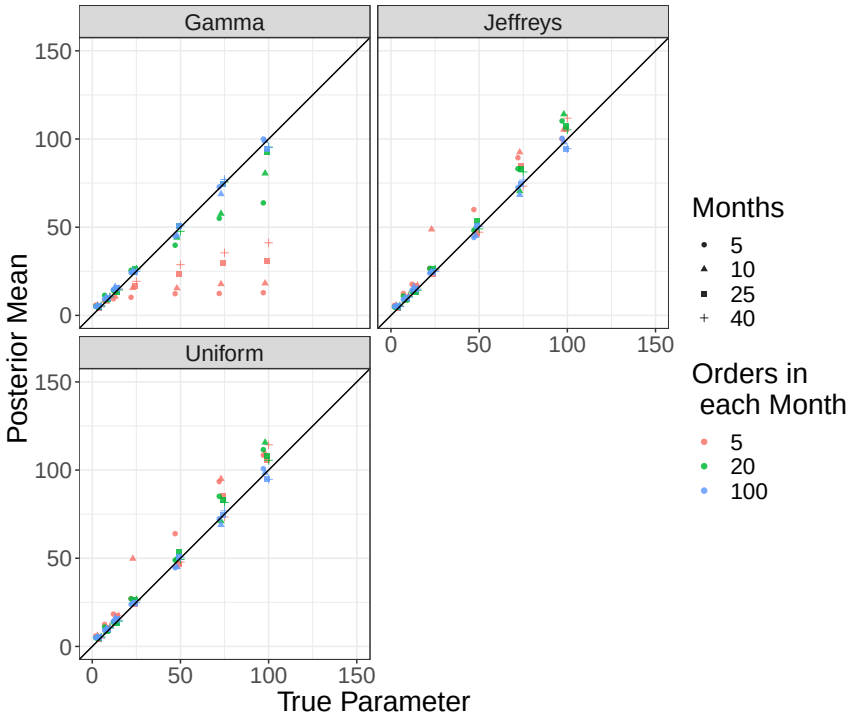
$$\pi(\alpha|\mathbf{x}) \propto f(x_1, x_2, \ldots, x_{K-1})\pi_J(\alpha) \tag{4.9}$$

$$\propto \frac{\Gamma(\alpha)}{\Gamma(\frac{\alpha}{n+1})^{(n+1)}}(\prod_{i=1}^{n+1} x_i)^{\frac{\alpha}{n+1}-1}\pi_J(\alpha) \tag{4.10}$$

.

To demonstrate the differences in prior choices, a simulation study was performed with Gamma, Jeffreys, and Uniform priors across different combinations of months ($t \in (5, 10, 25, 30)$) and orders within each month ($n \in (5, 20, 100)$). True values of $\alpha \in (5, 10, 15, 20, 25, 50, 75, 100)$ were used for each combination of the number of months (t) and the number of orders in each month (n_t). These values were chosen to represent the variety in the number of orders observed in our data and differing levels of replication across months. The parameters of the Gamma prior were chosen via method matching, where the mean was equal to the number of orders plus one (n_t+1), while the standard deviation was equal to 30% of the true value of α. The other two priors are flat or non-informative priors, meaning little to no hyper-parameter selection is needed.

Figure 4.13: A simulation study analyzing the effect of the number of months, the number of orders in each month, and the prior distribution for α on the relationship between the true value of α and the posterior mean $\hat{\alpha}$.



In Figure 4.13 we see that when the number of months is small and the number of orders each month (n_t) is much different than the true α parameter value (e.g., $\alpha = 100$ and $n = 5$), the posterior mean is very different than the true value of α. The Gamma prior shrinks the estimate ($\hat{\alpha}$) towards the value that would satisfy uniform regularity of order times (e.g., $n + 1 = 6$). While the posterior mean with a non-informative prior is closer to the true value of α for small samples, the posterior mean is still not biased to the true value in such situations and has little interpretability. In this sense, the Gamma prior allows the posterior to provide firm evidence as to whether or not α follows a uniform order time spacing a la

Pires et al. (2017).

All priors result in close estimates and good 95% posterior coverage for large samples. The data we are concerned about has customers with large number of orders, and some with very small numbers of orders. In small samples, if better 95% posterior coverage is the desired criterion in prior selection, then a flat or non-informative prior should be chosen (see Table 4.3.2). On the other hand, if the purpose of the model is to evaluate whether or not the uniform order time model is appropriate, a prior such as the gamma, which can shrink the posterior towards the uniform order time model, should be selected. Since we desire to use the uniform model as a base, deviating only should the data dictate, we choose gamma priors that will shrink the estimates towards the uniform model.

Table 4.2: 95% Posterior coverage per prior

	Gamma	n		Jeffreys	n		Uniform	n	
T	5	20	100	5	20	100	5	20	100
5	0.25	0.57	0.84	0.93	0.90	0.91	0.92	0.89	0.89
10	0.27	0.64	0.87	0.92	0.92	0.90	0.92	0.91	0.89
25	0.32	0.74	0.91	0.91	0.92	0.91	0.90	0.92	0.91
40	0.34	0.81	0.91	0.93	0.95	0.92	0.93	0.95	0.92

4.4 Multi-level Order Model

Data is considered to be of multiple scales when there are varying levels of granularity associated with the observed data. The most granular level is referred to as the "fine" level while the least granular is referred to as the "coarse" level. Typically multiscale data may be aggregated from the fine level to coarse and reciprocally appropriated from the coarse level to the fine through a pre-defined link such as the sum or mean of the fine levels being the link to the more coarse level. Our coarse scale is the order time spacings of all customers together and the fine scale is the order time spacings of each customer.

4.4.1 Data Notation

For our data, we are interested in customers that each have an observed vector of order times, $\mathcal{X}_{t,c}$ at time (month) t and customer group c. Each element of vector $\mathcal{X}_{t,c}$ lives in the unit line. That is, $\mathcal{X}_{t,c,i} \in (0,1)$ for all t, c, and i where i is the i^{th} order of month t for customer c. Each order time is the fraction of the month (after adjusting for non-business hours) at which

the order took place. Additionally we assume that the elements of $\mathcal{X}_{t,c}$ are in ascending order.

Now we will define the vector $\mathcal{Z}_t$ to be the concatenated vector $(\mathcal{X}_{t1}, \mathcal{X}_{t2}, \ldots, \mathcal{X}_{t,c})$ in month t. This combined vector is also such that all of its elements are arranged in ascending order. We will also define $n_{t,c}$ to be the number of elements (orders) in $\mathcal{X}_{t,c}$ and n_t to be the number of elements (orders) in $\mathcal{Z}_t$. Using this notation, the observed order times at both the fine and coarse levels are thought to follow the following processes similar to Equation 4.2,

$$\mathbf{z}_t = A_t \begin{pmatrix} \mathcal{Z}_t \\ 1 \end{pmatrix} \sim Dir\left(\frac{\alpha}{n_t + 1}\mathbf{1}_{n_t+1}\right) \tag{4.11}$$

$$\mathbf{x}_{t,c} = A_{t,c} \begin{pmatrix} \mathcal{X}_{t,c} \\ 1 \end{pmatrix} \sim Dir\left(\frac{a_c}{n_{t,c} + 1}\mathbf{1}_{n_{t,c}+1}\right) \ . \tag{4.12}$$

The matrix A_t is a square matrix of dimension $n_t \times n_t$ that when multiplied by a vector of ordered values on the unit line (as in 4.11) returns the spacings between those ordered values on the unit line (see Appendix B).

4.4.2 Multiscale Model for Order Times

Why Multiscale? We want to define a joint model for the data observed on the coarse level and the data observed the fine level. When simulating the orders for this supply chain it is important to make sure the order times for each individual customer are well-estimated so the shipments to each customer can be part of the optimization. Conversely, when the orders come in for a given product, regardless of customer it is important to plan production schedules and inventory levels for that product.

Should the estimation of the concentration parameter (a_c in 4.12) of our Dirichlet processes for each customer imply a model on the coarse scale that is consistent with the data on the coarse scale then there is no need for a multiscale model. Using a simulation study we attempt to see if there is a need for this multiscale model using the following steps:

1. estimate $\hat{a}_c$ for each customer based on the observed customer order times,
2. simulate order times for each customer with $\hat{a}_c$,
3. group all simulated customers' order times from (2) and estimate $\hat{\alpha}_{sim}$ - the parameter governing the coarse customer order times implied from the simulations of the fine order times, and
4. estimate $\hat{\alpha}_{coarse}$ from the observed coarse order times and compare to $\hat{\alpha}_{sim}$.

If there is evidence that $\hat{\alpha}_{sim}$ could be equal to $\hat{\alpha}_{coarse}$, then there is no need for a multiscale model. If that were the case, the estimated parameters of the fine customer order times would be sufficient to describe the process at the coarse level. From the simulation study it is evident that is not the case. The simulation study done for the most ordered diaper product resulted in $\hat{\alpha}_{sim} = 251.1$ and $\hat{\alpha}_{coarse} = 639.4$. Similar results hold true for the other products.

Recall in Chapter 2 I was trying estimate the order volume for a product during month t using Jeffrey's rule of conditioning. A brief reminder is explained below:

Jeffrey's Rule of Conditioning

Assume for any node and descendants that $z_{0,t}$ is the volume of the more coarse level and $\mathbf{z}_{1,t}$ is the vector of its descendants at the finer level. The researcher believes the likelihood of the more fine (descendent) level is $p(\mathbf{z}_{1,t}) = N(\boldsymbol{\mu}_{1,t}, v_1)$ with a link to the more coarse (parent) level of $p(z_{0,t}|\mathbf{z}_{1,t}) = N(z_{0,t}; \mathbb{1}^\intercal z_{1,t}, \delta)$. These distributions imply the measure of the coarse level is $p(z_{0,t}) = N(z_{0,t}; \mathbb{1}^\intercal \boldsymbol{\mu}_{1,t}, \mathbb{1}^\intercal v_1 \mathbb{1} + \delta)$. Suppose we receive new information and believe the true model of the coarse actually has measure,

$$q(z_{0,t}) = N(\mu_{0,t}, v_0) \ , \tag{4.13}$$

where either $\mu_{0,t} \neq \mathbb{1}^\intercal \boldsymbol{\mu}_{1,t}$ or $v_0 \neq \mathbb{1}^\intercal v_1 \mathbb{1} + \delta$.

Jeffrey's rule of conditioning is necessary to ensure that a multiscale model is consistent at each level. How do we reconcile the measures $q(\mathbf{z}_{1,t})$ and $p(\mathbf{z}_{1,t})$? Using Jeffrey's rule of conditioning we assume $q(z_{1,t,c}|z_{0,t}) = p(z_{1,t,c}|z_{0,t})$ for each customer (c) partition of $\mathbf{z}_{1,t}$. Then, since each $z_{1,t,c}$ are conditionally independent, by Jeffrey's Rule $p(\mathbf{z}_{1,t}|z_{0,t}) = q(\mathbf{z}_{1,t}|z_{0,t})$. The multiscale model will estimate the joint distribution of each and every node. Following the framework Ferreira et al. (2005) we have the following series of equations,

$$\begin{aligned} q(z_{0,t}, \mathbf{z}_{1,t}) &= q(\mathbf{z}_{1,t}|z_{0,t}) q(z_{0,t}) && (4.14) \\ (\textit{Jeffrey's Rule}) \implies &= p(\mathbf{z}_{1,t}|z_{0,t}) q(z_{0,t}) && (4.15) \\ (\textit{Bayes Theorem}) \implies &\propto p(z_{0,t}|\mathbf{z}_{1,t}) p(\mathbf{z}_{1,t}) q(z_{0,t}) \ . && (4.16) \end{aligned}$$

The updated model on the fine data is then obtained by integrating out the coarse random variable,

$$\begin{aligned} q(\mathbf{z}_{1,t}) &\propto \int p(z_{0,t}|\mathbf{z}_{1,t}) p(\mathbf{z}_{1,t}) q(z_{0,t}) dz_{0,t} && (4.17) \\ \implies q(\mathbf{z}_{1,t}) &= N(A\mu_{0,t} + \Lambda^{-1} v_1^{-1} \boldsymbol{\mu}_{1,t}, A v_0^{-1} A^\intercal + \Lambda^{-1}) \ , && (4.18) \end{aligned}$$

where $\Lambda = \mathbb{1}\delta^{-1}\mathbb{1}^\intercal + v_1^{-1}$ and $A = \Lambda^{-1}\mathbb{1}\delta^{-1}$.

Dirichlet Aggregation

Taking the Normal likelihood with a normal link between the fine and the coarse level works nicely as just demonstrated. When dealing with Dirichlet likelihoods as in Equations 4.11 and 4.12, the link between the coarse and fine levels is not as obvious. Suppose that in a given month we observe the customer order times (as fractions in month) and order time spacings for customer (1) and (2) as follows:

$$\text{Raw Times} \text{ — } \text{Time Spacings} \tag{4.19}$$

$$\mathcal{X}_1 = \begin{pmatrix} 0.10 \\ 0.50 \\ 0.70 \end{pmatrix} \mathbf{x}_1 = \begin{pmatrix} 0.10 \\ 0.40 \\ 0.20 \\ 0.30 \end{pmatrix} \tag{4.20}$$

$$\mathcal{X}_2 = \begin{pmatrix} 0.05 \\ 0.45 \\ 0.75 \\ 0.95 \end{pmatrix} \mathbf{x}_2 = \begin{pmatrix} 0.05 \\ 0.40 \\ 0.30 \\ 0.20 \\ 0.05 \end{pmatrix} \tag{4.21}$$

$$\mathcal{Z} = \begin{pmatrix} 0.05(1) \\ 0.10(1) \\ 0.45(2) \\ 0.50(2) \\ 0.70(3) \\ 0.75(3) \\ 0.95(4) \end{pmatrix} \mathbf{z} = \begin{pmatrix} 0.05(1) \\ 0.05(1) \\ 0.35(2) \\ 0.05(2) \\ 0.20(3) \\ 0.05(3) \\ 0.20(4) \\ 0.05 \end{pmatrix} . \tag{4.22}$$

As shown in Equation 4.22, the coarse vector ($\mathcal{Z}$) for the raw order time fractions is simply values of $\mathcal{X}_1$ and $\mathcal{X}_2$ combined and in order from least to greatest. Since our likelihoods deal with the order spacings and not the raw times, the link needs to connect the spacings (vectors on the right in the previous example) of the fine scale and the coarse scale. While the link between the two may not be obvious, given that we know which orders come from which customers, we can start at the coarse level and use the Dirichlet Aggregation Property to imply fine distributions.

For example, the Dirichlet aggregation property states that if we know $\mathbf{z} \sim Dir((5,5,5,5)^\intercal)$ then if we add any number of elements in $\mathbf{z}$ together, the corresponding values in the Dirichlet

parameter are added together such that $(z_1, z_2 + z_3, z_4)^\intercal \sim Dir((5, 10, 5)^\intercal)$. Conditional on knowing the classification of the customers of each order, the fine level order time spacings are the aggregation of the coarse order time spacings. In 4.23 we see the customer labels in parentheses next to the order time spacing at the coarse level. In order to get the spacings for the fine level of that customer, you add each of the previous spacings since the last order of that customer until the current order's spacing. Using the Dirichlet aggregation property to link from the coarse spacings ($\mathbf{z}$) to $\mathbf{x}_1$ conditional on the customer labels is done as follows:

$$\begin{aligned}
\mathbf{z} &= \begin{pmatrix} 0.05 & (1) \\ 0.05 & (1) \\ 0.35 & (2) \\ 0.05 & (2) \\ 0.20 & (3) \\ 0.05 & (3) \\ 0.20 & (4) \\ 0.05 & \end{pmatrix} \\
\mathbf{x}_1 &= \begin{pmatrix} 0.10 & = (1) & +(1) & \\ 0.40 & = (2) & +(2) & \\ 0.20 & = (3) & & \\ 0.30 & = (3) & +(4) & +0.05 \end{pmatrix} \\
\mathbf{x}_2 &= \begin{pmatrix} 0.05 & = (1) & & \\ 0.40 & = (1) & +(2) & \\ 0.30 & = (2) & +(3) & +(3) \\ 0.20 & = (4) & & \\ 0.05 & & & \end{pmatrix} .
\end{aligned} \tag{4.23}$$

Following 4.11 and the Dirichlet aggregation property (shown in Equation 4.23) implies

$$\mathbf{x}_1 \sim Dir\left(\left[2\frac{\alpha}{n+1}, 2\frac{\alpha}{n+1}, \frac{\alpha}{n+1}, 3\frac{\alpha}{n+1}\right]^\intercal\right) . \tag{4.24}$$

While this implied distribution may be useful this does not follow the singular Dirichlet model/Dirichlet Process that is believed to be the true model for $\mathbf{x}_1$ (Equation 4.12).

Multiscale Dirichlet

From the previous section we have figured out the coarse likelihood, the implied fine likelihood and the belief of true model of the fine order time spacings. From equations 4.11 and 4.12 we have (ignoring month/time labels),

$$
\begin{aligned}
(Coarse)\ p(\mathbf{z}) &= Dir\left(\frac{\alpha}{n+1}\mathbb{1}_{n+1}\right) \\
(Link)\ p(\mathbf{x}_c|\mathbf{z},\mathbf{s}) &= ? \\
(Fine)\ q(\mathbf{x}_c) &= Dir\left(\frac{a_c}{n_c+1}\mathbb{1}_{n_c+1}\right)\ ,
\end{aligned}
$$

where c denotes the customer and $n = \sum_{c=1}^{C} n_c$ is the number of orders in a given month. The Dirichlet aggregation property lends itself to be a link from the coarse scale to the fine scale as shown in Equation 4.23. This link is expressed by

$$p(\mathbf{x}_c|\mathbf{z},\mathbf{s}) = Dir\left(\frac{\alpha}{n_z+1}\mathbf{u}\right) \tag{4.25}$$

$$\mathbf{u} = \left(\sum_{1}\mathbb{1}(s_i \neq c)+1, \sum_{2}\mathbb{1}(s_i \neq c)+1, \ldots, \sum_{n_c}\mathbb{1}(s_i \neq c)+1\right) \tag{4.26}$$

where $\sum_1 \mathbb{1}(s_i \neq c)$ denotes the number of values in the vector of customer labels, $\mathbf{s}$ before the first order from customer c. This implied model of the descendent order time spacings isn't harmonious with the assumed descendent model which follows a singular Dirichlet, $q(x_c) = Dir\left(\frac{a_c}{n_c+1}\mathbb{1}_{n_t+1}\right)$.

The multiscale Dirichlet model differs slightly from that of the Normal-Normal (N-N) multiscale model presented in Chapter 2 and earlier on in this chapter. First, the N-N example assumed the fine model was known as well as the fine to coarse link and those two combined to imply a coarse distribution. Upon receiving new information about the coarse model, the fine model was revised to make it consistent with that new information. Contrary to the N-N model, the Dirichlet-Dirichlet multiscale model starts with the coarse model being known along with the coarse to fine link and the implied fine model. New information is then received about the *fine* model (new information is received about the coarse model in the N-N setup). With that new information about the fine model, the coarse model is revised to make it consistent.

Using the principles of Jeffrey's rule of conditioning, we update the joint distribution $q(\mathbf{z},\mathbf{x})$ of the coarse and fine scales. First, we update the conditional distribution for a customer's

order times given the coarse order times ($q(\mathbf{x}_c|\mathbf{z})$) as follows,

$$q(\mathbf{x}_c|\mathbf{z}) \propto q(\mathbf{z}|\mathbf{x}_c)q(\mathbf{x}_c) \text{ Bayes Rule} \tag{4.27}$$

$$\implies \propto p(\mathbf{z}|\mathbf{x}_c)q(\mathbf{x}_c) \text{ Jeffreys Rule} \tag{4.28}$$

$$\implies q(\mathbf{x}_c|\mathbf{z}) \propto p(\mathbf{x}_c|\mathbf{z},\mathbf{s})p(\mathbf{s}|\mathbf{z})q(\mathbf{x}_c) \tag{4.29}$$

$$\implies q(\mathbf{x}_c|\mathbf{z}) \propto \prod_{j=1}^{n_c} x_{c_j}^{\frac{\alpha}{n_z+1}u_j-1} \prod_{j=1}^{n_c} x_{c_j}^{\frac{a_c}{n_c+1}-1} \tag{4.30}$$

$$\implies \propto \prod_{j=1}^{n_c} x_{c_j}^{a^\star_{c_j}-1} p(\mathbf{s}|\mathbf{z}) \tag{4.31}$$

$$\implies a^\star_{c_j} = \frac{(n_c+1)\alpha+(n+1)a_c}{(n+1)(n_c+1)} + \frac{(n_c+1)\alpha\sum_j \mathbb{1}(s_i \neq c) - (n+1)(n_c+1)}{(n+1)(n_c+1)} \,. \tag{4.32}$$

The conditional distribution of the fine order times given the coarse will be useful for simulation of future data. Once the coarse data is simulated, each customer will be able to be simulated very efficiently using this conditional distribution. With this conditional distribution we are able to derive the revised joint distribution of the fine and coarse data.

The revised conditional distribution of the fine order times for a customer given the coarse order times has a several qualities that lead to a better understanding of how the multiscale model works. The parameter vector $\mathbf{a}^\star_c$ for this conditional distribution itself is a combination of a weighted average between the coarse and fine parameters $\left(\frac{(n_c+1)\alpha+(n+1)a_c}{(n+1)(n_c+1)}\right)$ and an adjustment for how regular that particular customer's orders are amongst all customer's orders $\left(\frac{(n_c+1)\alpha\sum_j \mathbb{1}(s_i\neq c)-(n+1)(n_c+1)}{(n+1)(n_c+1)}\right)$. The weighted average portion is weighted based on how many orders are observed from that customer (n_c) versus all the orders of that product in total (n). Should one customer account for a large percentage of all orders this revised conditional distribution will gravitate towards the concentration parameter for that particular customer (a_c). Should a customer have a small percentage of the orders, the concentration parameter will assume behavior closer to that of all the customer's collectively (α).

A special property occurs if the orders for a customer are regularly spaced apart in the sequence of all the orders. Such an occurrence would mean that in the multiscale concentration parameter ($a^\star_{c_j}$), the part $\sum_j \mathbb{1}(s_i \neq c+1)$ which denotes how many orders have passed since the last order from customer c would be known. This value is derived from knowing that for customer c there are n_c+1 order spacings and $n+1$ order spacings for all the customers, so customer c's orders would need to happen every $\frac{n+1}{n_c+1}$ orders. This leads us to the following theorem:

Theorem 1. *If on the coarse scale,* $\mathbf{z} \sim Dir\left(\frac{\alpha}{n+1}\mathbb{1}_{n+1}\right)$ *where* $\alpha = n+1$ *and on the fine scale* $\mathbf{x}_c \sim Dir\left(\frac{a_c}{n_c+1}\mathbb{1}_{n_c+1}\right)$*, and the values from which the spacings* $\mathbf{z}$ *come from are ordered sequentially with descendant c's values spaced at equal integers from other customer's orders, then the conditional distribution* $\mathbf{x}_c|\mathbf{z} \sim Dir(\mathbf{a}_c^\star)$ *where* $a_{c_j}^\star = \frac{a_c}{n_c+1}$*.*

This is equivalent to saying that if the orders for customer c are equally spaced between other orders and the coarse data follows uniform behavior then the conditional multiscale distribution of $\mathbf{x}_c|\mathbf{z}$ is equal to the believed distribution of the fine data. In other words, given these conditions the coarse scale provides no new information to the fine scale. To prove this theorem we start with an alternate way to write $a_{c_j}^\star$:

Proof.

$$a_{c_j}^\star = \frac{\alpha}{n+1}\sum_j \mathbb{1}(s_i \neq c+1) + \frac{a_c}{n_c+1} - 1 \tag{4.33}$$

$$\implies = \frac{\alpha}{n+1}\frac{n+1}{n_c+1} + \frac{a_c}{n_c+1} - 1 \tag{4.34}$$

$$\implies = \frac{a_c(n+1)+\alpha(n_c+1)}{(n+1)(n_c+1)} - 1 \tag{4.35}$$

$$\implies = \frac{a_c(n+1)+(n+1)(n_c+1)}{(n+1)(n_c+1)} - 1 \tag{4.36}$$

$$\implies = \frac{a_c}{n_c+1} \tag{4.37}$$

□

Similar to Theorem 1, we get another useful result of the data is uniform on the fine scale for a particular customer:

Theorem 2. *If on the coarse scale,* $\mathbf{z} \sim Dir\left(\frac{\alpha}{n+1}\mathbb{1}_{n+1}\right)$ *and on the fine scale* $\mathbf{x}_c \sim Dir\left(\frac{a_c}{n_c+1}\mathbb{1}_{n_c+1}\right)$ *where* $a_c = n_c + 1$*, and the values from which the spacings* $\mathbf{z}$ *come from are ordered sequentially with descendant c's values spaced at equal integers from other customer's orders, then the conditional distribution* $\mathbf{x}_c|\mathbf{z} \sim Dir(\mathbf{a}_c^\star)$ *where* $a_{c_j}^\star = \frac{\alpha}{n+1}$*.*

Opposite of Theorem 1, Theorem 2 provides a situation where the customer's orders follow the exact distribution of that of the coarse order times, with the only difference being the number of orders observed.

Proof.

$$a^{\star}_{c_j} = \frac{\alpha}{n+1}\sum_j \mathbb{1}(s_i \neq c+1) + \frac{a_c}{n_c+1} - 1 \tag{4.38}$$

$$\implies = \frac{\alpha}{n+1}\frac{n+1}{n_c+1} + \frac{a_c}{n_c+1} - 1 \tag{4.39}$$

$$\implies = \frac{a_c(n+1)+\alpha(n_c+1)}{(n+1)(n_c+1)} - 1 \tag{4.40}$$

$$\implies = \frac{(n_c+1)(n+1)+\alpha(n_c+1)}{(n+1)(n_c+1)} - 1 \tag{4.41}$$

$$\implies = \frac{\alpha}{n+1} \tag{4.42}$$

□

Now with the conditional distribution of $\mathbf{x}_c|\mathbf{z}$ known we can derive the proportional joint multiscale distribution of the fine ($\mathbf{x}$) and the coarse ($\mathbf{z}$) data.

$$q(\mathbf{x},\mathbf{z}) = q(\mathbf{z}|\mathbf{x})q(\mathbf{x}) \text{ Definition} \tag{4.43}$$

$$\propto \prod_{c=1}^{C} p(\mathbf{z}|\mathbf{x}_c)q(\mathbf{x}_c) \text{ Jeffreys Rule} \tag{4.44}$$

$$\propto p(\mathbf{z})\prod_{c=1}^{C} p(\mathbf{x}_c|\mathbf{z})q(\mathbf{x}_c) \text{ Bayes Rule} \tag{4.45}$$

$$\implies q(\mathbf{x},\mathbf{z}) \propto p(\mathbf{z})\prod_{c=1}^{C} q(\mathbf{x}_c|\mathbf{z}) \tag{4.46}$$

This joint distribution of the fine and coarse data is a combination of the revised conditional distribution of the fine data given the coarse data and the original distribution of the coarse data. The joint multiscale distribution is the likelihood of our data and can be used to estimate a_c for each customer as well as α for all customers combined. All of these parameters appear in the multiscale parameter $\mathbf{a}^{\star}_c$. If estimating using a technique like MCMC, where the practitioner is mostly concerned with a proportional posterior distribution, a_c separates out from the rest of the terms in $\mathbf{a}^{\star}_c$ and its proportional conditional posterior is easily sampled from. On the other hand, the coarse parameter is more computationally intensive to estimate, but there is only one such parameter while the parameters for the individual customers reduce to the same estimation of the fine-only model $\mathbf{x}_c|a_c \sim Dir\left(\frac{a_c}{n_c+1}\mathbb{1}_{n_c+1}\right)$.

Simulation Study

With this new approach to handling multiscale Dirichlet data, it is of interest to know whether the multiscale approach is of practical significance as opposed to simply a tool that doesn't impact the estimation of our coarse parameter in practice. Recall that for the parameters on the fine scale (a_c), the estimation after the multiscale adjustment is the same as before. For the coarse parameter (α) that is not the case.

The simulation study is done as by simulating individual customers with assigned concentration parameters. Recall the model for the fine data from Equation 4.12,

$$(Fine)\ q(\mathbf{x}_c|a_c) = Dir\left(\frac{a_c}{n_c+1}\mathbf{1}_{n_c+1}\right)\ . \tag{4.47}$$

Simulations were done with 10 replications of every distinct combination of number of orders in a month for a customer ($n_c \in (2, 5, 10, 50)$), total number of months ($T \in (2, 5, 10, 30)$), and total number of customers ($C \in (2, 6, 15, 25, 100)$). The values of a_c were randomly selected with equal probability of selection from elements of the vector $\mathbf{a} = (2, 5, 15, 25, 50, 100)$. In each simulation the fine parameters are estimated, the simulated fine data is aggregated at the coarse scale where both the multiscale coarse parameter and non-multiscale coarse parameter (assumes only the model in Equation 4.11) are estimated. For posterior estimation in each simulation the Jeffrey's prior (Equation 4.8) was used on both the fine and coarse scales to prevent shrinking the parameter estimates in simulations with small numbers of customers, months, or orders.

Table 4.3: Percentage of simulations where the revised multiscale parameter for the coarse data is less than the estimated parameter for the coarse data without a multiscale adjustment.

	Months			
C	2	5	10	30
2	0.42	0.46	0.42	0.39
6	0.45	0.43	0.40	0.38
15	0.44	0.37	0.41	0.30
25	0.40	0.49	0.45	0.43
100	0.96	0.91	0.93	0.96

As shown in Table 4.4.2, there are some observable trends. For the post part if there are 30 or less months observed with 25 or less customers, it appears that the multiscale parameter is actually greater than the concentration parameter for the non-multiscale model more than half the time. In our simulation, at the 100 customer level the multiscale concentration parameter was less than the non-multiscale concentration parameter more than 90% of the time across all observed number of months. This is evidence that the multiscale model is working to adjust for the structure of the data. When we introduced our motivation to explore this multiscale model (Section 4.4.2) we observed how when we estimated parameters

for each customer at the fine scale and then simulated coarse data from those fine parameters we estimated a much smaller coarse parameter ($\hat{\alpha}_{sim} = 251.1$) than we estimated when estimating coarse parameter directly from the coarse data ($\hat{\alpha}_{coarse} = 639.4$).

Multiscale Order Times: Diapers

In our customer order data we can estimate the parameters for the multiscale model from Equation 4.46 for the most popular size of the most popular diaper. This is the same product that is used in experiment done in Section 4.4.2. We sample 2,000 posterior draws of the coarse order time concentration parameter from the multiscale model as well as 2,000 posterior draws of the coarse concentration parameter ignoring the multiscale adjustment. We use the Jeffrey's prior for the coarse data since our prior analysis showed little impact on the choice of prior with large amounts of data, which is the case for the coarse scale. For the parameters for the individual customers the priors were of the class $a_c \sim Gamma(10\bar{n}_{t\cdot}, 10)$ where $\bar{n}_{t\cdot}$ is the average number of orders in a month for a given customer. This prior has a mean equal to the observed average number of orders in a month and a variance of $\frac{1}{10}\bar{n}_{t\cdot}$.

Comparing the estimated coarse parameter from the multiscale model to the estimated parameter from the coarse likelihood (Equation 4.11) ignoring the multiscale element, we see that the multiscale parameter is clearly smaller than that of the non-multiscale coarse parameter. This is the same behavior that was observed in Section 4.4.2 when the coarse data was simulated based on the fine parameters. While the posterior distribution places the multiscale parameter above the parameter from the simulated data ($\hat{\alpha}_{sim} = 251.1$) with probability close to 1, the multiscale parameter was much more in line with what we observed in that simulation versus what is observed when the multiscale structure of the order times are ignored ($\hat{\alpha}_{coarse} = 639.4$). The model now is harmonious on the coarse scale with what is implied based on the model on the fine scale.

Table 4.4: Comparing the multiscale and the non-multiscale posterior estimation of concentration parameter for the coarse order times of BabyDry Size 4.

	mean	2.5%	97.5%
$\hat{\alpha}\|\cdot$	639.43	631.12	647.87
$\hat{\alpha}_{multiscale}\|\cdot$	262.92	260.55	265.36

Figure 4.14: Observed relationship between orders per month and estimated concentration parameters.

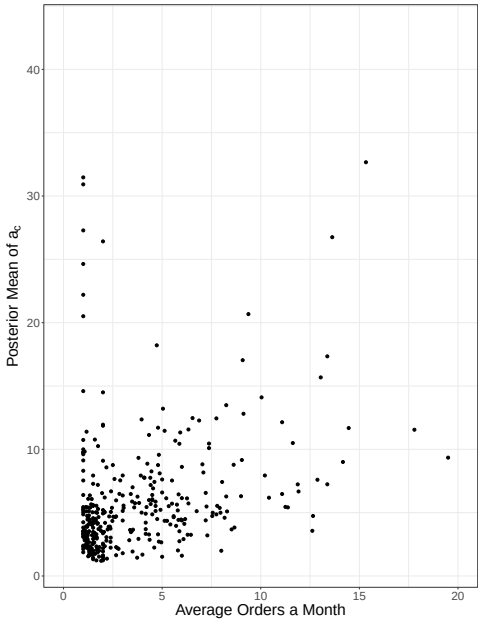
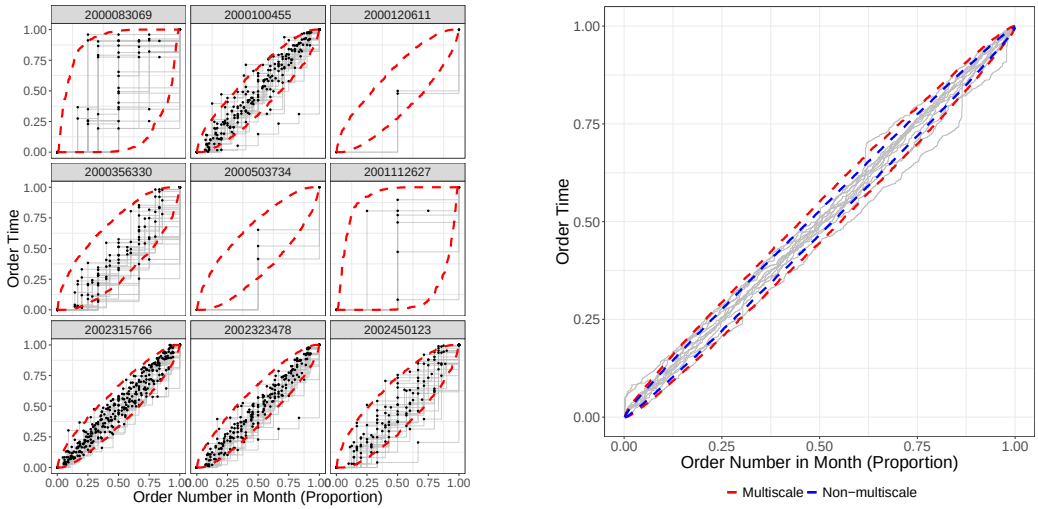


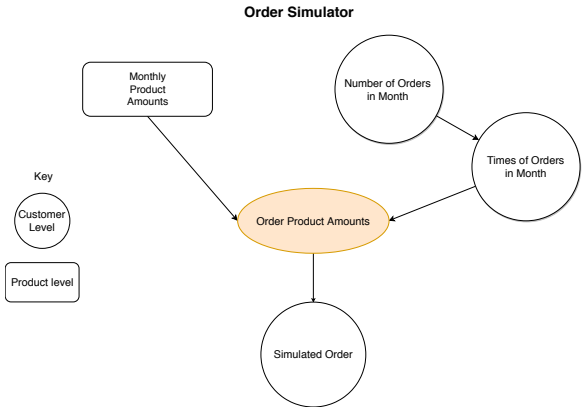
Figure 4.15: 90% posterior intervals of order times on the fine scale (left) and the aggregated coarse scale (right) for nine different customers. The multiscale α on the coarse data is wider than the non-multiscale, giving the model better posterior coverage due to the variance introduced by the fine customer order times.



Chapter 5

Product Amounts in Each Order

Figure 5.1: Diagram of the four models that combine to create the order simulator. This chapter covers highlighted portion.



I previously established a model for the number of orders in a month, the order times for each customer's orders, and the aggregate amounts of each product ordered in a month. A full order simulator needs to be able to simulate the amounts of each product for each order for each customer (customer x order x product). Recall that in Chapter 2 the aggregate monthly amounts were linked via a multiscale process. These monthly amounts ordered for a product are fairly consistent. Contrarily, the amounts of each product for a particular order are not very consistent. For this part of the order simulator the following questions need to be answered:

- Which products are included in this order?
- How much of each product is ordered ?

Table 5.1: Sample of order product amounts.

			Products						
Customer	Sale	Month	A Tier 2	B Tier 1	B Tier 2	B Tier 3	C Tier 2	C Tier 3	...
2000081948	48185962	1	119.46	0.00	0.00	0.00	0.00	0.00	...
2000082928	48217060	1	40.14	0.00	0.00	0.00	55.99	29.55	...
2000081948	48185963	1	0.00	0.00	0.00	0.00	234.72	171.57	...
2000082980	48196406	1	80.73	0.00	0.00	0.00	299.84	136.26	...
2000878175	48195556	1	0.00	0.00	0.00	0.00	0.00	0.00	...
2000082449	48235266	1	0.00	0.00	0.00	0.00	85.34	23.80	...

5.1 Constraints to Modeling Product Amounts in Each Order

Beyond answering which products and how much is ordered for each order it is also necessary to have a model that is consistent with the monthly amounts as specified in Chapter 2. The simulated amount coming from the monthly model captures the aggregate trends of each product from month-to-month. The product amounts in each order has no need to be concerned with such trends as long as the sum within each product across all orders is nearly equal to the monthly amount for that product. This can be viewed as a constraint such that

$$\sum_{c=1}^{C} \sum_{j=1}^{n_{t,c}} z_{p,t,c,j} = z_{p,t} \ . \tag{5.1}$$

Here $z_{p,t}$ is the true demand for for product p in month t, as in Chapter 2. Each customer c has a different number of orders in the month, denoted by $n_{t,c}$. While it may be possible to build this constraint into a model, it would further complicate an already complex system. Another way to build in the constraint is to condition the data on the monthly amount for each product. As this part is the last piece of the simulator, the monthly product amounts are always available. Conditional on the monthly amount, the raw volume ordered for each product is thus divided by total amount ordered for that product in a month as seen in Equation 5.2. This eliminates some complexity to the model by dealing directly with the fractions as opposed to the raw amounts. Inference will be made on these fractions which can then be transformed to the raw amounts.

$$z^{f}_{p,t,c,j} = \frac{z_{p,t,c,j}}{z_{p,t}} \tag{5.2}$$

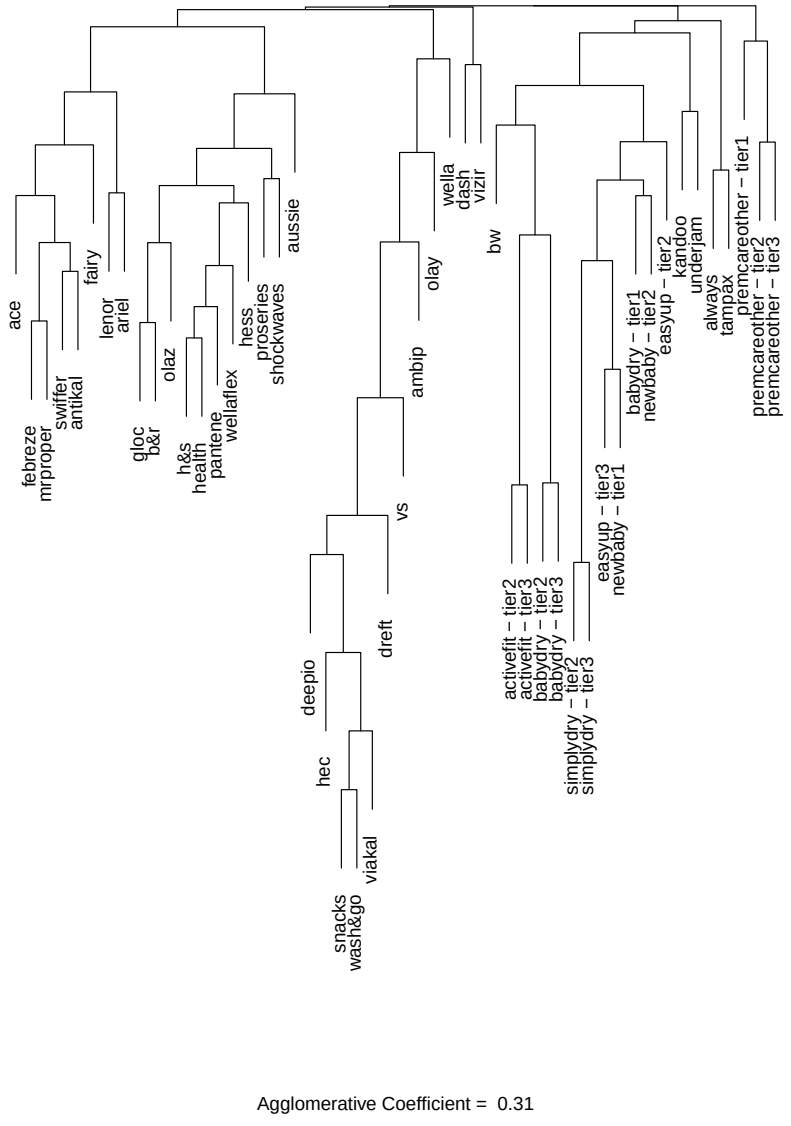
Each customer may have different ordering habits. A model that treats customers differently is necessary. In any simulator it is important to be able to capture that certain products are ordered together for some customers, while other customers may order different products together. For example, a grocery store may tend to order diapers, wipes, toothbrushes, and hairbrushes together while a department store is likely to order hairbrushes and makeup together. In both situations hairbrushes are ordered but have different correlated products because the stores are inherently different. With 50 unique product groupings it would take a very large amount of orders within each customer to be able to properly capture the covariance structure between the products. Only a few customers order enough to make such an approach feasible. As such, it will be necessary to group customers together in some way to make shared inference across some number of clusters.

5.2 Clustering Orders

The complete order x product data available across the two years of interest contain over 108,684 different unique orders with the 50 products I have grouped all the SKU's into. There are 941 customers that account for those 108,684 orders. The most frequent customer has 9,323 orders in that two-year span, while 90 different customers have only 1 order placed in those two years. The desired method is to predict the product amounts for a given order the researcher is interested in. While clustering customers together may make sense, some customer's have large number of orders that potentially come from different clusters. Take for example a large on-line retailer such as Amazon which has several different product areas. Some orders may come from a cluster that loosely represents baby products while another order comes from a cluster that is similar to feminine hygiene.

Hierarchical clustering is a classical method that uses measures of similarity or distance between multivariate vectors (Ward Jr (1963)). The problem with using hierarchical clustering in this problem is due to computational limits. Hierarchical clustering needs to explore the partitions of $n(n-1)/2$ similarity scores. Calculating those similarity scores for a given distance measure (often Euclidean or Manhattan) is computationally taxing when $n = 108,684$. Plus once those are calculated the clustering algorithm needs to sufficiently explore the partition space. Principal component analysis (Pearson (1901) & Hotelling (1933)), factor analysis (Cattell (1952)), or other methods that require a decomposition of an n x n matrix with such a large n are similarly computationally cumbersome.

Figure 5.2: Hierarchical clustering across product amounts ordered



Hierarchical clustering on the products themselves is not computationally intensive. While this does not provide a good avenue to the correlations of products for specific orders and customers, it does provide a good exploratory view on which products have the most similar amounts ordered across all customers. The dendrogram in Figure 5.2 shows four main clusters that qualitatively are (from left to right):

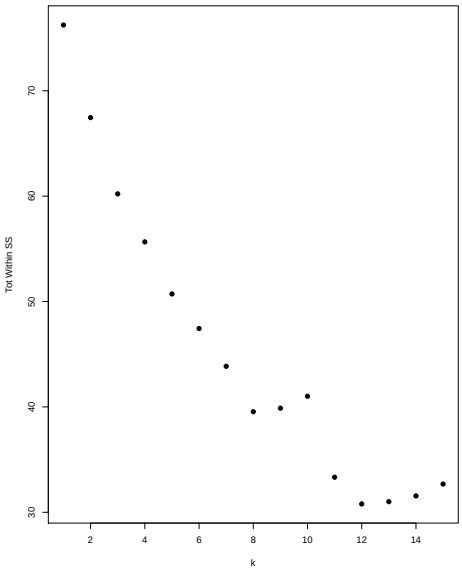
1. Cleaning Products,

2. Beauty Products,
3. Detergents, and
4. Diapers, wipes, and tampons.

The dendrogram is a good reference for which to compare the final clustering system to see if these groups of products are apparent in different clusters.

Methods which are not as computationally difficult with such large row rank typically have a predetermined number of clusters against which an algorithm is fit. K-means is one such algorithm that finds k centroids which are chosen to minimize the multivariate within-cluster sum of squares (Steinhaus (1956), Lloyd (1957), Forgy (1965), & MacQueen et al. (1967)). One could also minimize the Manhattan distance from the cluster centroid in a k-medians algorithm. In choosing the number of k, one exploratory approach is to choose a value when the decrease in total within-cluster sums of squares for an addition k falls below a predetermined threshold. I conducted such a study and $k = 8$ or $k = 12$ were sensible values for k as seen in Figure 5.3.

Figure 5.3: Total within-cluster sums of squares for k-means



One drawback with the k-means algorithm is in its framework of minimizing the within-cluster sum of squares (Steinhaus (1956)). The algorithm does not distinguish clusters well if the clusters are not of similar size or of a spherical shape. In our data the k-means algorithm tends to put almost all of the observations into one cluster with the centroid being

close to the overall mean.

5.2.1 Latent Dirichlet Allocation

If the purpose of our work here is to create a better simulator than the effort to cluster the orders should make sense from a simulation perspective. For a given order i I propose that it will come from one of K clusters with the probability of of each cluster being selected following the vector $\boldsymbol{\theta}_i$. This could also be expressed as,

$$s_i \sim Multinomial(1, \boldsymbol{\theta}_i) \tag{5.3}$$

where s_i is the indicator of the cluster for which the i^{th} vector of orders (ignoring its customer label) belongs. $\boldsymbol{\theta}_i$ is an K-dimensional vector where each value is the probability that the order is from the i^{th} cluster. A multinomial likelihood with $n = 1$ is sometimes referred to as the *categorical* likelihood. The Dirichlet distribution of course is the conjugate prior to $\boldsymbol{\theta}_i$. This the same setup as is used in what is commonly referred to as "Topic Models."

The structure of perhaps the most prevalent topic model, Latent Dirichlet Allocation (LDA) (Blei et al. (2003)), has the properties for which I seek. LDA tries to classify documents into a combination of topics. Subsequently, in each of these topics there are distributions across all the different words that are in the population of words in corpus of documents. I desire to have clusters (topics) of orders (documents), where each cluster (topic) has a distribution across products (words). One drawback to LDA is that it deals with discrete counts of data whereas I have continuous fractions of order amounts for each product.

Recall the structure of the product amounts from Table 5.1 and recall that each product amount has been scaled to be a fraction of the monthly amount ordered for that product. Thus the fractional product amounts ($z^f_{p,t,c,j}$) are contained in the unit line $(0, 1)$. Suppose we discretize our product amount fractions into integers using a ceiling of a multiplication such that

$$z^i_{p,t,c,j} = \left\lceil z^f_{p,t,c,j} * 10,000 \right\rceil . \tag{5.4}$$

In essence no information is lost until after the fifth significant digit of the fractional amounts. Several integer values were tested before settling on 10,000. The number 10,000 was chosen to be a medium between not losing much information in the data and keep the integers to be values in ranges common to Topic Models. Table 5.2 contains the integer values which can be compared to the original raw volume of each product for the same orders in Table 5.1. Within each product the proportional relationship across the orders is very similar. In Topic Model terminology, each row of orders with integer amounts (such as the ones seen in

Table 5.2) is the aggregated number of times each word appears in each document.

Table 5.2: Sample of discretized order product amounts

			Products						
Customer	Sale	Month	A Tier 2	B Tier 1	B Tier 2	B Tier 3	C Tier 2	C Tier 3	...
2000081948	48185962	1	6	0	0	0	0	0	...
2000082928	48217060	1	2	0	0	0	6	7	...
2000081948	48185963	1	0	0	0	0	22	40	...
2000082980	48196406	1	0	0	0	0	28	32	...
2000878175	48195556	1	0	0	0	0	8	0	...
2000082449	48235266	1	0	0	0	0	8	6	...

The topic model inspired clustering process can be defined by making the following distributional assumptions:

$$\begin{aligned} z_{d=1...M,w=1...N_d}|\boldsymbol{\theta}_d &\sim Multinomial_K(1,\boldsymbol{\theta}_d) \\ w_{d=1...M,w=1...N_d}|\boldsymbol{\varphi}_{z_{dw}} &\sim Multinomial_V(1,\boldsymbol{\varphi}_{z_{dw}}) \\ \boldsymbol{\theta}_{d=1...M}|\boldsymbol{\alpha} &\sim Dirichlet_K(\boldsymbol{\alpha}) \\ \boldsymbol{\varphi}_{k=1...K}|\boldsymbol{\beta} &\sim Dirichlet_V(\boldsymbol{\beta}) \ . \end{aligned} \tag{5.5}$$

Here $z_{d,w}$ represents an indicator that integer (word) w of order (document) d comes from each of the $K = 9$ clusters (topics). Prior values for $\boldsymbol{\alpha} = \boldsymbol{\beta} = \mathbb{1}$ in order to have a non-informative prior. Several values of K were considered, and while there is not a definite "correct" value, that value seemed to have the most distinct clusters without redundancy. There are $M = 108,684$ total orders and N_d equals the row sum of the integer values in each order. The indicator $w_{d,w}$ then signifies the distribution across words in the topic of $z_{d,w}$ for that same integer.

Variational Bayes Justification

Topic models can be fit using Gibbs sampling but approximate posterior distributions can be estimated using a variational Bayes approach. Even with over 100,000 orders the variational Bayes approach takes only a few minutes. Variational Bayes methodology is used to provide approximate posterior probabilities. Since only approximate posterior inference is available via the variational approach, one should make sure the approximate posterior probabilities are consistent. I performed a five-fold cross validation, where the data was fit on 80 % of the data and then the posterior predictive distributions for the clusters of 20% test data were calculated. In Table 5.3 the percentage of orders with the most likely cluster a posteriori are

shown for each of the five cross-validated folds. The first cross-validation fold was the most different, but in all, the variational Bayes returns similar out of sample results.

Table 5.3: Most likely clusters a posteriori across five-fold cross validation.

	Cluster								
CV	1	2	3	4	5	6	7	8	9
1	0.02	0.05	0.46	0.02	0.08	0.07	0.13	0.15	0.02
2	0.45	0.05	0.01	0.07	0.15	0.03	0.14	0.01	0.09
3	0.46	0.04	0.01	0.07	0.15	0.02	0.15	0.01	0.10
4	0.46	0.04	0.01	0.07	0.14	0.02	0.16	0.02	0.09
5	0.46	0.04	0.02	0.07	0.13	0.02	0.16	0.01	0.09

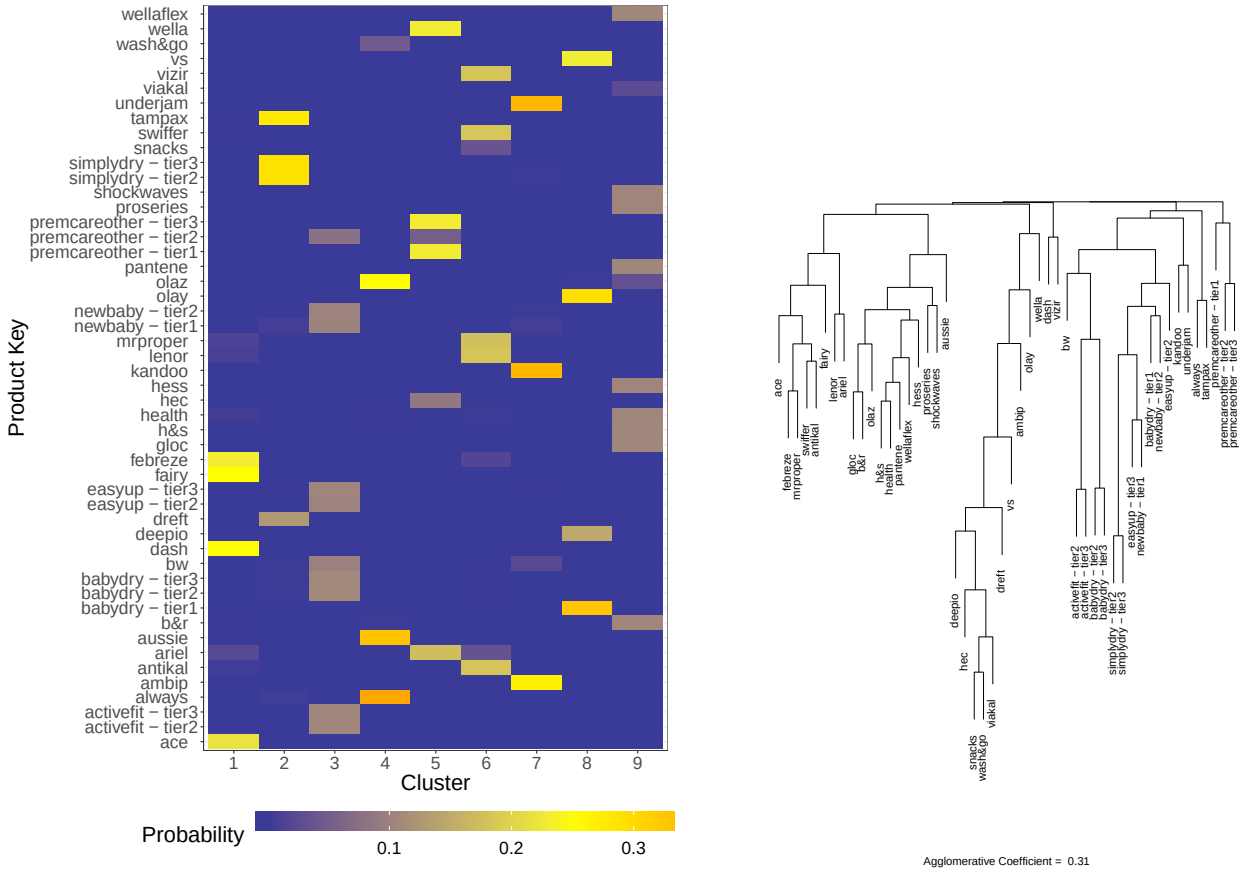
In Table 5.4 the perplexity for each out of sample data is shown to be consistent. The perplexity is defined as $exp\left(-\frac{L(\boldsymbol{w})}{n_{products}}\right)$ where $L(\boldsymbol{w})$ is the log-likelihood. The variational Bayes approach appears to give stable results.

Table 5.4: Perplexity values across five-fold cross validation.

CV	Perplexity
1	10.63
2	10.49
3	10.50
4	10.73
5	11.70

5.2.2 Results

Figure 5.4: Product (word) probability by cluster



While it may not be intuitive converting the fractional amounts to integers, the result of the LDA fits the data very well. For example we can see in Figure 5.4 the posterior probability within each cluster that a given product would be ordered. These are the approximate posterior means of $\varphi_{k=1\ldots K}$ from Equation 5.5. There are very clearly groupings in these clusters that follow similar groupings observable in the hierarchical clustering on product amounts.

Table 5.5: Count of orders by most likely cluster *a posteriori*

Cluster	1	2	3	4	5	6	7	8	9
Count	6980	5496	46335	3993	6883	12765	3733	1650	20849

There are several validating features for using LDA and the number of clusters ($K = 9$). First, each of the clusters has a sizable number of orders where that cluster was the most

likely *a posteriori* (see Table 5.5). Second, for $\boldsymbol{\varphi}_{k=1\ldots K}$ (the product probabilities given the cluster), every product has a cluster for which it has at least a 5% chance of being selected. At the same time there are only a couple products that have more than a 5% posterior chance of being selected in more than one cluster. This is evidence that the clusters are distinguishing different order patterns while not dissecting the data too thin. In Table 5.6 we see the approximate mean posterior probabilities ($\boldsymbol{\theta}_{1,\ldots,6}$) of the first six orders in the dataset.

5.3 Simulating Future Order Product Amounts

The LDA model was built on the orders by product amounts. In simulating future orders using the results from the LDA I only have the customer for which the order belongs. If a given customer orders similarly each time then $\boldsymbol{\theta}_d$ should be similar for all orders $d \in S_c$ where S_c is a vector of length M denoting which orders belong to customer c. To denote whether or not $\boldsymbol{\theta}_d$ is similar across all orders for given customer a measure of multivariate distance, such as Euclidean, can be used. In doing such an analysis it is apparent that many customers have many different order types. Take the example in Table 5.6 where two entries for customer `2000081948` are seen. In the first order cluster 6 is the most likely with a plurality of the probability at 0.339. This order seems to be pulling from many different clusters while the second order from that customer has cluster 1 with a probability of 0.613 and cluster 6 with 0.333. The first order has large variance ($\theta = \frac{1}{9}\mathbb{1}$ would have largest variance) while the second has about 0.95 probability between two clusters alone.

Table 5.6: Sample of cluster (topic) probabilities

			Clusters								
Customer	Sales Doc	Month	1	2	3	4	5	6	7	8	9
2000081948	48185962	1	0.072	0.007	0.172	0.008	0.000	0.339	0.117	0.024	0.262
2000082928	48217060	1	0.343	0.000	0.000	0.071	0.038	0.229	0.000	0.031	0.288
2000081948	48185963	1	0.613	0.000	0.000	0.004	0.000	0.333	0.000	0.000	0.049
2000082980	48196406	1	0.622	0.002	0.002	0.002	0.002	0.365	0.002	0.002	0.002
2000878175	48195556	1	0.793	0.000	0.000	0.041	0.000	0.064	0.000	0.000	0.101
2000082449	48235266	1	0.219	0.001	0.026	0.001	0.001	0.713	0.001	0.001	0.037

The simulation method will simulate each customer and each order sequentially. For a given simulation, it will sample from the existing orders in the dataset for customer c such that,

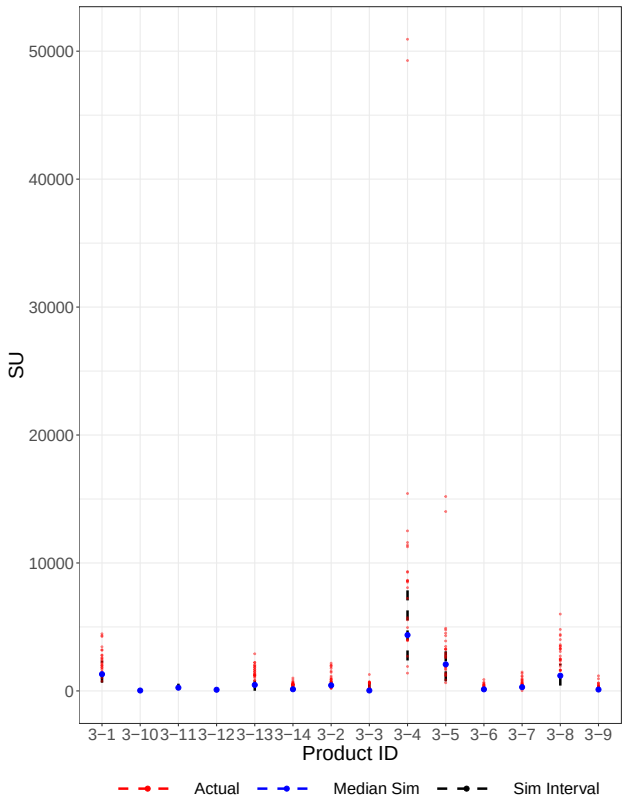
$$\begin{aligned} o_{c,i}|\gamma_c &\sim Multinomial_{n_c}(1, \gamma_c) \\ \gamma_c &\sim Dirichlet(\psi) \ , \end{aligned} \tag{5.6}$$

where n_c is the number of orders for customer c. The prior for ψ will be equal to $\mathbb{1}_{n_c}$. This is equivalent to a bootstrap implementation of selecting from which order to pull. The value of N_d for $o_{c,i}$ contains the sum of the integer values from the selected order that will then be used to sample from $\boldsymbol{\theta}_{o_{c,i}}$ and $\boldsymbol{\varphi}_{o_{c,i}}$ N_d times. This process adds another multinomial-Dirichlet layer on top of the generative process that defines the Latent Dirichlet Allocation (Equation 5.5).

An example of this simulating process can be seen for a certain customer in Figure 5.5. In this example the simulations are for one specific month, whereas the actual results are 24 sequential months. It is apparent that for these products the simulations mostly fall within the 80% simulation band. The simulating behavior for each product amount is right-skewed, which is to be expected.

For many customers there are one or two products for which the simulation expects the customer to order that are not observed in the 24 month sample. This is due to the clustering mechanism used. Two customers may have similar order behavior for almost all 50 products except one or two. Most likely the orders from these customers will be given a high probability of being from the same cluster. While some may view this as a negative feature of the clustering method, I view it as a positive. This is evidence that the model is more anticipatory than solely sampling from the existing orders from each customer (bootstrap). That bootstrap method was used by Pires et al. (2017) and limits the simulator from suggesting products that aren't in a customer's order history despite if there is evidence to suggest the product is likely to be in the customers ordering future.

Figure 5.5: Sample of simulated product amounts compared to actual (24 months)



Chapter 6

Conclusion

In each of the preceding chapters, a methodology was presented and expanded for each of the aspects that is required to simulate orders in an end-to-end supply chain. This order simulation is just one piece that is needed for a complete simulation of a supply chain. While there have been many attempts to simulate orders in the past, this methodology contains several unique features, such as:

- Models were based entirely on observed supply chain data.
- A multiscale model of monthly amounts across tree structure of products. This captures the correlations between different products within the same general group.
- Auto-regressive behavior for the number of orders in a month for each customer.
- Order times treated as a continuous variable within a month. Most order simulations focus on per day or per week orders, without any effort to view orders as continuous in the modern 24 hour globalized economy.
- Introduced multiscale Dirichlet models for order time simulation.
- Applied topic model methodology to supply chains.

Not everyone of these features may be applicable or optimal in each supply-chain setting. For our data, these features are descriptive, emulate the true data, and split behavior up into either individual customers or the collective group of customers.

With this methodology one feature that could be included is creating a model that estimates the chances of a new customer emerging, and what attributes that customer may have. Methods in this paper already described in each chapter how, using prior distributions, a simulation could occur for a new customer given the new customer was known.

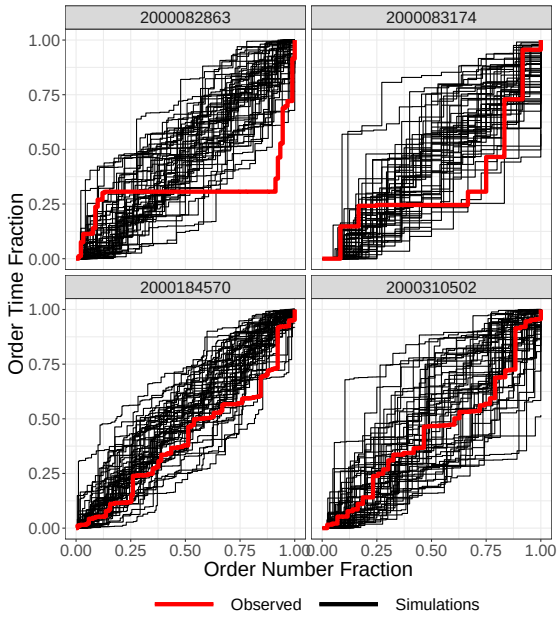
Not only does a supplier have the potential to gain new customers, but also to lose existing ones. The time-series nature of the model for the number of orders per month already handles the chance of a customer no longer doing business with the supplier. This is accomplished due to the increasing probability as time goes on with no orders that the customer will have no orders in the future. This is a useful feature of that model and could be used to flag customers to follow-up on to entice the customer to come back and start doing business again with the production company.

To wrap up this paper, in Section 6.1, the complete order simulator performs 50 simulations of orders for the last month of order data available. These simulations are then compared to the observed orders for that month for four selected customers and where applicable, the entire population of customers. As each aspect of the simulation is demonstrated the strength and some potential areas of improvement for the order simulator will be discussed.

6.1 Simulation Example and Review

The order time simulations for each customer as seen in Figure 6.1 fit the observed data (after business-hour adjustments) fairly well. The customer in the upper-left plot has a long period of low-activity, which makes the observed times seem a little out of line with the simulations. In reality, that month was a little out of character with the customer itself in other months.

Figure 6.1: Simulated order times for four selected customers compared to final observed month for those customers.



In Figure 6.2 the simulations of each customer's order times are combined to show what the aggregate (coarse) simulated order times would look like against the observed order times for that month. It is clear that with the larger sample that comes from combining all the customers, the coarse order times don't fit the observed times that well. That is why in Chapter 4 a multiscale model was introduced that accounted for this behavior in the order times across both the coarse and the fine scale. Using the updated belief about the multiscale order times as discussed in Chapter 4, the coarse order times fit the model much better as seen in Figure 6.3. This discrepancy in the fit for the coarse order times is exactly why the development and use of the multiscale Dirichlet model is important to the simulation of the customer order times.

Figure 6.2: The aggregation of the 50 customer simulated order times without the multiscale adjustment compared to the observed coarse order times.

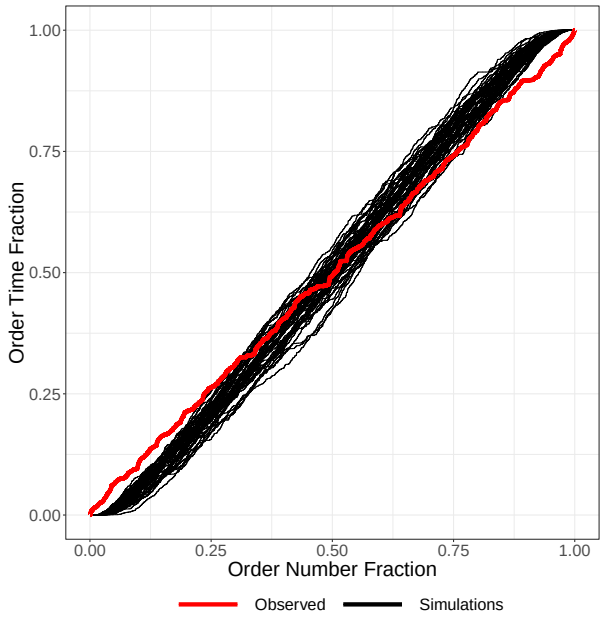
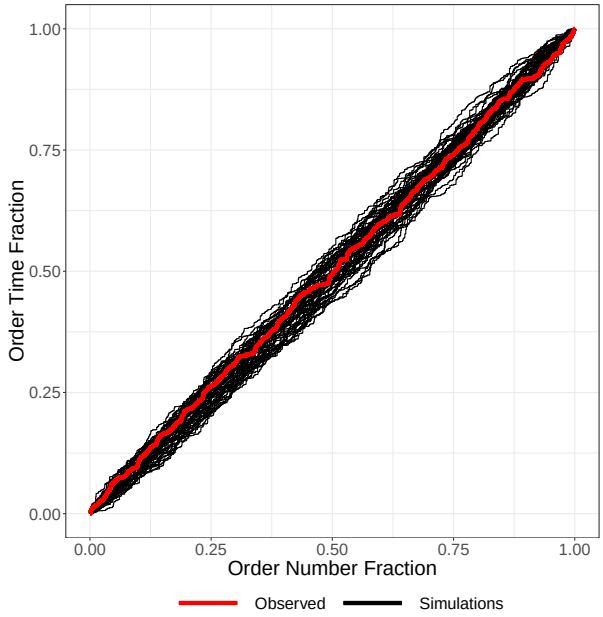


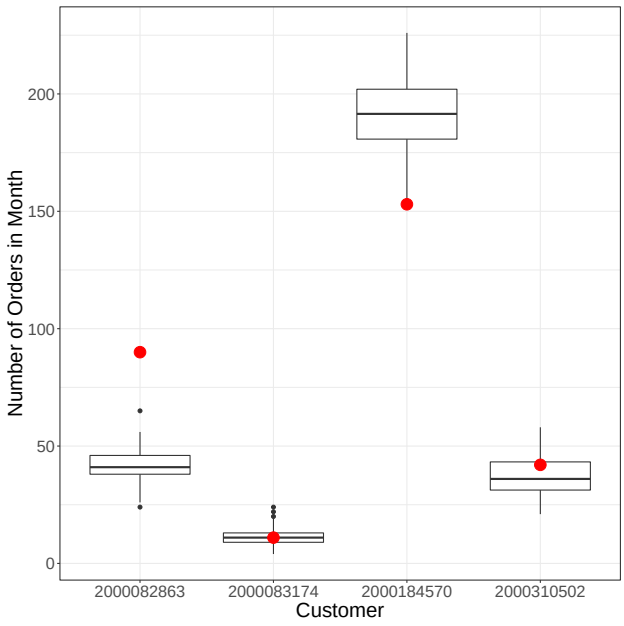
Figure 6.3: The 50 simulated coarse order times *with* the multiscale adjustment compared to the observed coarse order times.



Next in Figure 6.4, the number of orders in the month for the four customers selected are compared to the 50 simulated number of orders. For two of the customers the number of

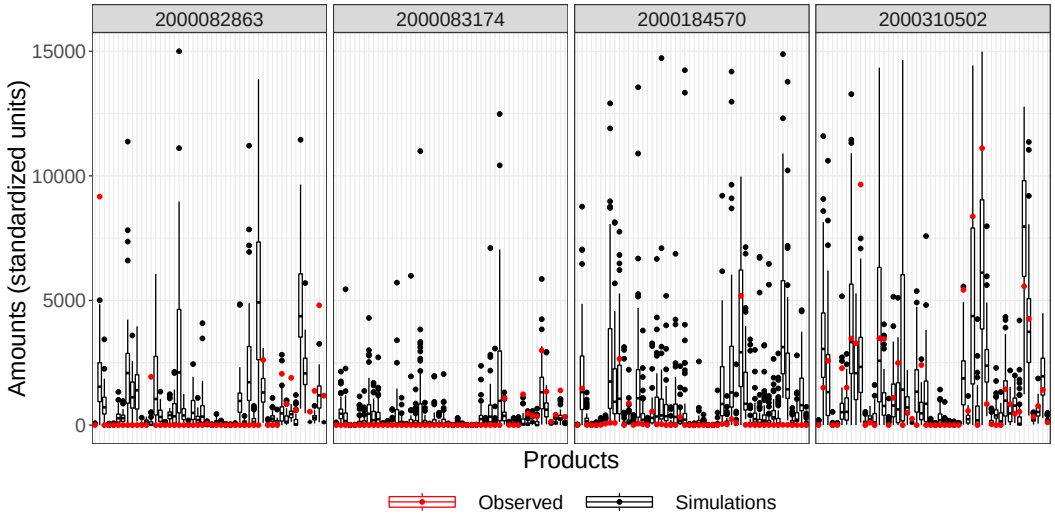
orders is close to the posterior predictive median while for two others the number of orders is at one extreme or another of the simulations. 50 simulations is not that much and with the time dependency of this analysis it could be that the trends for those customers are changing. Chapter 3 shows how the time dependency is an important aspect of the model for the number of orders. Figure 3.3.1 in that chapter shows a general picture of how that time dependency affects the predictive distribution of the number of orders.

Figure 6.4: The 50 simulations of the number of orders for the four customers selected compared to the actual number in the last month of the dataset.



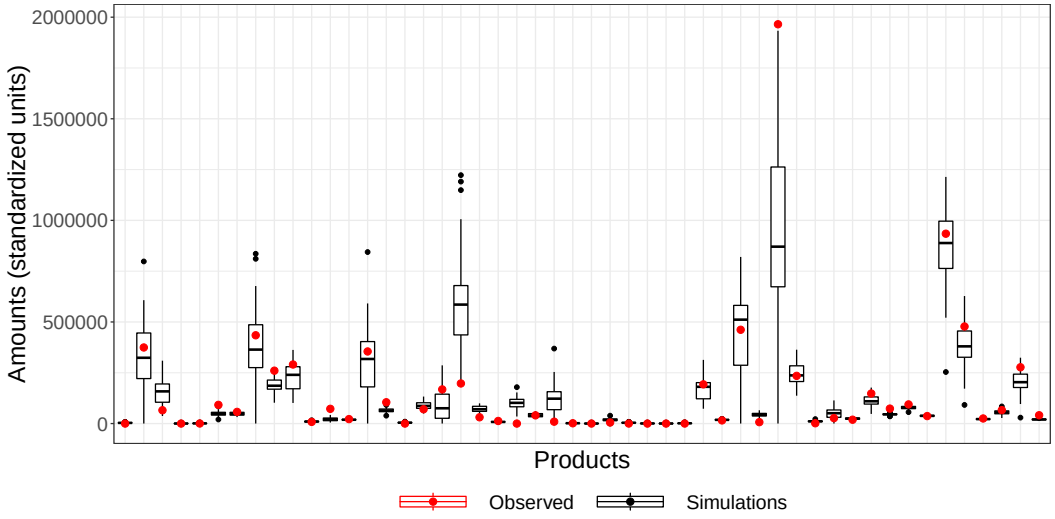
The final two plots show the simulated amounts for each of the products versus the actual amount ordered for that product. Figure 6.5 shows this broken up by customer and Figure 6.6 shows the aggregate amounts for each of the products across all customers. Due to the wide discrepancy in how much is ordered for different products, it appears that many products were ordered at 0 volume. While some are, a lot of those are just small positive amounts and the simulations also show small positive amounts. There are some products where the observed amounts for a customer are outside the maximum of the simulations for that amount, but as you can see from the plot, several of these products have a very wide range in the simulations because the customer is not very consistent with how much they order for that product from month to month.

Figure 6.5: Simulated amounts for each product across the four customers of interest.



For all the customers combined, the simulated amounts are almost all in line with the observed amounts. This speaks to not only to the strength of the model but also the smaller predictive variance that comes in estimating the collective group of customers as opposed to the larger variance for an individual customer.

Figure 6.6: Simulated amounts for each product across all customers compared to observed amounts.



There are several potential next steps in furthering this work. With more data, seasonal effects should be able to be estimable in several areas. Additionally, experts in the field

suggest that customers will place an order up to the point of filling a truck but do not want to go over and be charged for another truck shipment in order to be more efficient with shipping costs. In future work, some adjustment for the caps in amounts that a customer may place should be considered. Even without those features, this work has shown to be a vast improvement on the previous order simulator used in Pires et al. (2017). Lastly, this order simulator opens the door to further advancements in supply-chain simulation by being able to relax assumptions around various order time constraints and the complex relationships between products and customers in orders.

Appendix A

The derivation of the Jeffrey's prior distribution of α in a symmetric Dirichlet Distribution with random vector $\mathbf{w}_{tc} \sim Dir(\frac{\alpha}{n_t}\mathbf{1}_{n_t})$ for a given customer c with n_t orders in each of $t = 1, 2, \ldots, T$ months is as follows:

$$
\begin{aligned}
I(\theta) &= -E\left[\frac{d^2}{d\theta^2} log(f(\mathbf{x}|\theta))\right] \qquad \text{(A.1)}\\
\vdots &= \vdots \\
f(\mathbf{x}|\theta) &= \prod_{t=1}^{T} \frac{\Gamma(x)}{\Gamma(\frac{\alpha}{n_t})} \prod_{i=1}^{n_t} x_i^{\frac{\alpha}{n_t - 1}} \\
&= \frac{\Gamma(x)^T}{\Gamma(\frac{\alpha}{n_1})\Gamma(\frac{\alpha}{n_2}) \ldots ...\Gamma(\frac{\alpha}{n_T})} \prod_{t=1}^{T}\prod_{i=1}^{n_t} x_{ti}^{\frac{\alpha}{n_t - 1}} \\
\ell(\mathbf{x}|\theta) = \log(f(\mathbf{x}|\theta)) &= T\log(\Gamma(\alpha)) - \sum_{t=1}^{T} n_t \log\left(\Gamma\left(\frac{\alpha}{n_t}\right)\right) + \sum_{t=1}^{T}\sum_{i=1}^{n_t}(\frac{\alpha}{n_t} - 1)\log(x_{ti}) \\
\frac{d\ell}{d\alpha} &= T\frac{d}{d\alpha}\log(\Gamma(\alpha)) - \sum_{t=1}^{T}\frac{d}{d\alpha}\log\left(\Gamma\left(\frac{\alpha}{n_t}\right)\right) + \sum_{t=1}^{T}\sum_{i=1}^{n_t}\frac{\log(x_{ti})}{n_t} \\
\frac{d^2\ell}{d\alpha^2} &= T\frac{d^2}{d\alpha^2}\log(\Gamma(\alpha)) - \sum_{t=1}^{T}\frac{1}{n_t}\frac{d^2}{d\alpha^2}\log\left(\Gamma\left(\frac{\alpha}{n_t}\right)\right) \\
I(\theta) = -E\left[\frac{d^2}{d\theta^2} log(f(\mathbf{x}|\theta))\right] &= \sum_{t=1}^{T}\frac{1}{n_t}\frac{d^2}{d\alpha^2}\log\left(\Gamma\left(\frac{\alpha}{n_t}\right)\right) - T\frac{d^2}{d\alpha^2}\log(\Gamma(\alpha)) \\
\pi_J(\theta) \propto I(\theta)^{\frac{1}{2}} &= \left[\sum_{t=1}^{T}\frac{1}{n_t}\frac{d^2}{d\alpha^2}\log\left(\Gamma\left(\frac{\alpha}{n_t}\right)\right) - T\frac{d^2}{d\alpha^2}\log(\Gamma(\alpha))\right]^{\frac{1}{2}} \qquad \text{(A.2)}
\end{aligned}
$$

The equation A.1 could be simplified further, albeit it isn't necessary since the calculation of first and second order derivatives of the $\log(\Gamma(x))$ function is very straightforward com-

putationally.

Appendix B

More explanation of the notation surrounding the notation around the observed order time fractions and the spacings between each order time.

We define $\mathbf{x}_{t,c}$ as follows,

$$\mathbf{x}_{t,c} = A \begin{pmatrix} \mathcal{X}_{t,c} \\ 1 \end{pmatrix} \ , \tag{B.1}$$

and similarly

$$\mathbf{y}_t = A \begin{pmatrix} \mathcal{Y}_t \\ 1 \end{pmatrix} \ , \tag{B.2}$$

where the square matrix A is $(n_{t,c}+1) \times (n_{t,c}+1)$ when applied to the fine-level data and $(n_t+1) \times (n_t+1)$ when applied to the coarse data and follows the form,

$$\begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 & 0 \\ -1 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & -1 & 1 & 0 & \dots & 0 & 0 \\ \vdots & & & \ddots & \ddots & & \vdots \\ 0 & 0 & 0 & 0 & \dots & -1 & 1 \end{bmatrix} . \tag{B.3}$$

This square matrix A creates the spacings that follow a Dirichlet distribution seen in Equation 4.2. The matrix has several properties, one of which is of particular use, is that A^{-1} is a lower triangular matrix of 1's. Since the matrix A, for any dimension is always square and nonsingular, it follows that

$$\mathcal{Y}_t = [A_t^{-1}\mathbf{y}_t]_{1:n_t} \text{ and} \tag{B.4}$$

$$\mathcal{X}_{t,c} = [A_{t,c}^{-1}\mathbf{x}_{t,c}]_{1:n_{t,c}} \ . \tag{B.5}$$

The notation of $A_{t,c}^{-1}$ denotes that it conforms to the dimension of $\mathbf{x}_{t,c}$ and A_t^{-1} conforms to the appropriate dimensions of $\mathbf{y}_t$. The subscript on each of the above vectors is to denote

that the first n_t or $n_{t,c}$ values are pulled, in other words, $\mathcal{Y}_t$ is equal to all but the last value of $[A_t^{-1}\mathbf{y}_t]$.

Appendix C

In Chapter 2 the multiscale model for the order amounts of all products is set forth. The conditional distribution of the fine given the coarse $p(\mathbf{z}_{1,t}|z_{0,t})$ was derived. If the closed-form solution of the joint multiscale model is desired it is derived below.

Recall the use of Jeffrey's Rule and its application towards the revised joint distribution of the fine and coarse scales from Chapter 2,

$$q(z_{0,t}, \mathbf{z}_{1,t}) = q(\mathbf{z}_{1,t}|z_{0,t})q(z_{0,t}) \tag{C.1}$$

$$(\textit{Jeffrey's Rule}) \implies = p(\mathbf{z}_{1,t}|z_{0,t})q(z_{0,t}) \tag{C.2}$$

It was previously shown that

$$p(\mathbf{z}_1|z_0) \propto exp\left\{-\frac{1}{2}(\mathbf{z}_1 - \mathbf{m}_1)^\intercal \Lambda (\mathbf{z}_1 - \mathbf{m}_1)\right\} \ , \tag{C.3}$$

with $\Lambda = (\mathbb{1}\delta_1^{-1}\mathbb{1}^\intercal + v_1^{-1})$ and $m_1 = \Lambda^{-1}(\mathbb{1}\delta_1^{-1}z_0 + v_1^{-1}\boldsymbol{\mu}_1)$. Recognizing the form of the kernal, it can be written $\mathbf{z}_1|z_0 \sim N(m_1, \Lambda^{-1})$. Now let's define $A = \Lambda^{-1}\mathbb{1}\delta_1^{-1}$. I can now rewrite the fine random variable as a linear function of the coarse random variable .

$$\mathbf{z}_1 = Az_0 + \Lambda^{-1}v_1^{-1}\boldsymbol{\mu}_1 \tag{C.4}$$

Recall that $z_0 \sim N(\mu_0, v_0)$, leaving the result for the joint distribution $q(z_0, \mathbf{z}_1)$

$$\begin{pmatrix} z_0 \\ \mathbf{z}_1 \end{pmatrix} \sim N\left(\begin{pmatrix} \mu_0 \\ A\mu_0 + \Lambda^{-1}v_1^{-1}\boldsymbol{\mu}_1 \end{pmatrix}, \Sigma\right) \tag{C.5}$$

$$\Sigma = \begin{bmatrix} v_0 & Av_0A^T \\ Av_0A^T & Av_0A^T + \Lambda^{-1} \end{bmatrix} \ . \tag{C.6}$$

Bibliography

Antoniak, C. E. (1974). Mixtures of dirichlet processes with applications to bayesian nonparametric problems. *The annals of statistics*, 1152–1174.

Bayarri, M. J., J. O. Berger, R. Paulo, J. Sacks, J. A. Cafeo, J. Cavendish, C.-H. Lin, and J. Tu (2007). A framework for validation of computer models. *Technometrics 49*(2), 138–154.

Blanchard, O. J. (1983). The production and inventory behavior of the american automobile industry. *Journal of Political Economy 91*(3), 365–400.

Blei, D. M., A. Y. Ng, and M. I. Jordan (2003). Latent dirichlet allocation. *Journal of machine Learning research 3*(Jan), 993–1022.

Blinder, A. S. (1982). Inventories and sticky prices: More on the microfoundations of macroeconomics. *The American Economic Review 72*(3), 334–348.

Carbonneau, R., K. Laframboise, and R. Vahidov (2008). Application of machine learning techniques for supply chain demand forecasting. *European Journal of Operational Research 184*(3), 1140–1154.

Cattell, R. B. (1952). Factor analysis: an introduction and manual for the psychologist and social scientist.

Celik, N. and Y.-J. Son (2010). State estimation of a supply chain using improved resampling rules for particle filtering. In *Proceedings of the 2010 Winter Simulation Conference*, pp. 1998–2010. Winter Simulation Conference.

Chen, F., Z. Drezner, J. K. Ryan, and D. Simchi-Levi (2000). Quantifying the bullwhip effect in a simple supply chain: The impact of forecasting, lead times, and information. *Management science 46*(3), 436–443.

Chen, F., J. K. Ryan, and D. Simchi-Levi (2000). The impact of exponential smoothing forecasts on the bullwhip effect. *Naval Research Logistics (NRL) 47*(4), 269–286.

Chick, S. E. (2004). Bayesian methods for discrete event simulation. In *Proceedings of the 2004 Winter simulation Conference*, pp. 89–100. Winter Simulation Conference.

Correa, M., C. Bielza, and J. Pamies-Teixeira (2009). Comparison of Bayesian networks and artificial neural networks for quality detection in a machining process. *Expert Systems with Applications 36*(3), 7270–7279.

Cragg, J. G. (1971). Some statistical models for limited dependent variables with application to the demand for durable goods. *Econometrica: Journal of the Econometric Society*, 829–844.

Davis, T. (1993). Effective supply chain management. *Sloan management review 34*(4), 35–46.

Dawid, A., M. DeGroot, J. Mortera, R. Cooke, S. French, C. Genest, M. Schervish, D. Lindley, K. McConway, and R. Winkler (1995). Coherent combination of experts' opinions. *Test 4*(2), 263–313.

De Jong, P. and N. Shephard (1995). The simulation smoother for time series models. *Biometrika 82*(2), 339–350.

Dornier, P.-P., F. Ernst Ricardo, and K. P. Michel (1998). Global operations and logistics: Text and cases.

Doucet, A., S. Godsill, and C. Andrieu (2000). On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing 10*(3), 197–208.

Dubois, D., P. Garbolino, H. Kyburg, H. Prade, and P. Smets (1991). Quantified uncertainty. *J. Applied Non-Classical Logics 1*, 105–197.

Ferguson, T. S. (1973). A bayesian analysis of some nonparametric problems. *The annals of statistics*, 209–230.

Ferguson, T. S. (1983). Bayesian density estimation by mixtures of normal distributions. *Recent advances in statistics 24*(1983), 287–302.

Ferreira, M. A., D. Higdon, H. K. Lee, and M. West (2005). Multi-scale random field models. Technical report, Technical Report, UFRJ-DME.

Ferreira, M. A. and H. K. Lee (2007). *Multiscale modeling: a Bayesian perspective*. Springer Science & Business Media.

Forgy, E. W. (1965). Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *biometrics 21*, 768–769.

Frühwirth-Schnatter, S. (1994). Data augmentation and dynamic linear models. *Journal of time series analysis 15*(2), 183–202.

Galton, F. (1877). *Typical laws of heredity*. publisher not identified.

Garcia, J. I., R. A. G. Morales, and P. E. Miyagi (2008). Supervisory system for hybrid productive systems based on bayesian networks and oo-dpt nets. In *2008 International Conference on Emerging Technologies and Factory Automation*, pp. 1108–1111. IEEE.

Harrison, J. and M. West (1999). *Bayesian forecasting & dynamic models*. Springer New York.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of educational psychology 24*(6), 417.

Jahangirian, M., T. Eldabi, A. Naseer, L. K. Stergioulas, and T. Young (2010). Simulation in manufacturing and business: A review. *European Journal of Operational Research 203*, 1–13.

Jeffrey, R. (1988). *Conditioning, Kinematics, and Exchangeability*, pp. 221–255. Dordrecht: Springer Netherlands.

Jeong, I. J., V. J. Leon, and J. R. Villalobos (2006). Integrated decision-support system for diagnosis, maintenance planning, and scheduling of manufacturing systems. *International Journal of Production Research 45*(2), 267–285.

Jin, S., Y. Liu, and Z. Lin (2012). A Bayesian network approach for fixture fault diagnosis in launch of the assembly process. *International Journal of Production Research 50*(23), 6655–6666.

Kahn, J. A. (1987). Inventories and the volatility of production. *The American Economic Review*, 667–679.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering 82*(1), 35–45.

Langseth, H. and L. Portinale (2007). Bayesian networks in reliability. *Reliability Engineering & System Safety 92*(1), 92–108.

Lee, H. L., V. Padmanabhan, and S. Whang (1997). Information distortion in a supply chain: The bullwhip effect. *Management science 43*(4), 546–558.

Li, M. and W. Q. Meeker (2014). Application of bayesian methods in reliability data analyses. *Journal of Quality Technology 46*(1), 1.

Lloyd, S. (1957). Least square quantization in pcm. bell telephone laboratories paper. published in journal much later: Lloyd, sp: Least squares quantization in pcm. *IEEE Trans. Inform. Theor.(1957/1982) Google Scholar*.

MacCarthy, B. L. and W. Atthirawong (2003). Factors affecting location decisions in international operations–a delphi study. *International Journal of Operations & Production Management 23*(7), 794–818.

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Volume 1, pp. 281–297. Oakland, CA, USA.

McNaught, K. and A. Chan (2011). Bayesian networks in manufacturing. *Journal of Manufacturing Technology Management 22*(6), 734–747.

Meixell, M. J. and V. B. Gargeya (2005). Global supply chain design: A literature review and critique. *Transportation Research Part E: Logistics and Transportation Review 41*(6), 531–550.

Min, H. and G. Zhou (2002). Supply chain modeling: past, present and future. *Computers & industrial engineering 43*(1), 231–249.

Ok, Z. D., J. A. Isaacs, and J. C. Benneyan (2008). Probabilistic and monte carlo risk models for carbon nanomaterial production processes. In *Electronics and the Environment, 2008. ISEE 2008. IEEE International Symposium on*, pp. 1–6. IEEE.

Pearson, K. (1895). Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London 58*, 240–242.

Pearson, K. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 2*(11), 559–572.

Pires, B., J. Goldstein, S. Reese, P. Sabin, G. Korkmax, S. Ba, K. Hamall, A. Koehler, S. Shipp, and S. Keller (2017). A bayesian simulation approach for supply chain synchronization. In W. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page (Eds.), *Proceedings of the 2017 Winter Simulation Conference*. Philadelphia PA: Last Resort Publishers.

Pradhan, S., R. Singh, K. Kachru, and S. Narasimhamurthy (2007). A Bayesian network based approach for root-cause-analysis in manufacturing process. In *2007 International Conference on Computational Intelligence and Security*, pp. 10–14.

Reese, C. S., A. G. Wilson, M. Hamada, H. F. Martz, and K. J. Ryan (2004). Integrated analysis of computer and physical experiments. *Technometrics 46*(2), 153–164.

Robert, C. P. and G. Casella (1999). *Monte Carlo statistical methods*. Springer-Verlag Inc.

Soberanis, I. E. D. (2010). *An extended Bayesian network approach for analyzing supply chain disruptions*. Ph. D. thesis, University of Iowa.

Steinhaus, H. (1956). Sur la division des corp materiels en parties. *Bull. Acad. Polon. Sci 1*(804), 801.

Stern, H. (2005). Baseball decision making by the numbers. In G. C. R. H. D. N. R. S. H. S. R. Peck, G. Casella (Ed.), *Statistics: A Guide to the Unknown* (3rd ed.)., pp. 393–406. Belmont: Thomson Brooke/Cole.

Ward Jr, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American statistical association 58*(301), 236–244.

West, M. and J. Harrison (1997). *Bayesian Forecasting and Dynamic Models (Second Edition)*. New York: Springer-Verlag.

Wright, F. J. (1961). Industrial dynamics.

Xu, D. and Y.-J. Son (2013). An integrated simulation, markov decision processes and game theoretic framework for analysis of supply chain competitions. In *Proceedings of the 2013 Winter Simulation Conference*, pp. 3930–3931. Winter Simulation Conference.