Erasmus University Rotterdam
Erasmus School of Economics

Master Thesis: Econometrics & Management Science
Business Analytics & Quantitative Marketing

A Case Study on the Effects of the COVID-19 Pandemic on Wholesale Iowa Liquor Sales Using A Multi-Scale Dynamic Model Framework

Helmich Beekman
Student Number: 369623

Assessor: prof. dr. Paap, R.
Co-assessor: prof. dr. Franses, P.H.B.F.
Date: April 30, 2021

By combining innovative models from recent literature this paper presents a multi-scale dynamic model framework designed to forecast the sales of a retailers entire product catalogue, from frequently sold products to those with intermittent demand. To test the forecasting performance of the framework a motivating case study is performed, the subject of which is the Iowa wholesale liquor market. The model framework displays favorable forecasting performance in comparison to traditional alternatives, whilst also being scalable and providing interpretable output for managerial insights, it does however also prove to be hard to calibrate which hampers the frameworks applicability. A further line of research is the impact of the COVID-19 pandemic on the wholesale liquor market in Iowa, in which the pandemic is found to have had greatly differing impacts on the sales of the various liquor categories.

Contents

1 Introduction

In today's rapid-paced markets, inventory stock is no longer as simple as it used to be. Where retailers used to base their stocking levels on their routine sales numbers, they now employ highly advanced models and algorithms to forecast their future sales and adjust inventory accordingly in a bid to stay ahead of the competition. Knowing when sales will spike and equally important, when sales will tumble, is therefore expensive information (Aras et al., 2017). In this light, sales forecasting has become an essential element in the operational planning of retailers and wholesalers alike and has thus generated much interest in both the commercial sector and amongst academic researchers over the past decades (Liu et al., 2013).

Particularly for larger, multinational producers, accurately forecasting sales across all their products, markets and customer populations poses a great challenge but thereby also presents the opportunity to find great gains. Such companies, who provide a wide range of products, often find the task of accurately forecasting the sales of their entire catalogue daunting. Aggregating all products into one pool and forecasting at this level is far from ideal. Most products are not substitutes but differ in crucial ways and pooling averages out the variability. On the other hand, isolating each product and forecasting it separately discards the mountains of information on sales drivers which affect all products alike. Furthermore, this approach fails for products with intermittent sales, as these might not provide enough information to conduct meaningful analysis. Attempts to estimate the parameters of all the product time-series in one all-encompassing model often suffer from the well-known 'curse of dimensionality' in addition to leading to complex, undecipherable output for those without a background in statistics. This touches upon an alternative attribute which company managers find desirable in their models, namely interpretability. If forecasting performance were the only interest then retailers would oftentimes resort to 'black box' machine learning algorithms which provide accuracy levels most statistical models can't compete with. However in operational management it is not only relevant to know how much sales will change but also why. In attempts to optimize all aspects of their business strategy, from inventory planning to marketing campaigns, retailers find themselves asking questions as; what are the key drivers facilitating the upcoming shift in demand? How uncertain are these forecasts? What effect will a price increase have? (Wacker & Lummus, 2002; Fisher & Raman, 2010)

Aside from forecasting performance and inferential capabilities other practical matters such as scalability and adaptability are of interest. A model which is unable to scale to a retailers entire product set or incapable of learning from new information as it comes in is of limited practical relevance. To sum, there is need for a scalable, contemporary model framework able of providing relevant managerial insights whilst still being capable of accurately forecasting the sales of all product demand types, from intermittent to frequent.

To meet this need, this thesis employs a framework which combines two recently proposed innovative model classes, the multi-scale Dynamic Count Mixture Model (DCMM) proposed by (Berry & West, 2019) and a Bayesian Predictive Synthesis (BPS) model (McAlinn & West, 2019). The former has been designed to forecast the sales of products with intermittent count-valued demand via a two-

step approach, in which first a Bernoulli Dynamic Generalized Linear Model (DGLM) forecasts the probability of a sale occurring after which a Poisson DGLM forecasts the magnitude of the sale. Whilst the latter proposes a framework capable of incorporating external forecasts, of the dependent variable, in a statistically sound manner into a Dynamic Linear Model (DLM) while "accounting for time-varying biases, miscalibration and inter-dependencies among models". Both employ a multi-scale approach wherein a higher-level, external model generates forecasts which a lower-level model relies on as a predictor. Whilst BPS was devised to harmoniously combine several forecasts of the same dependent variable, DCMM utilizes the external model to forecast latent factors which affect the dependent variable. It is this second approach which will be exploited in this case study, where higher-level DLMs will generate forecasts of relevant common factors which in turn are introduced as regressors in lower-level models. The structure of these lower-level models will depend on the nature of the data, products with frequent, high-valued sales will often be more suited to a DLM and thus the BPS framework will be employed to incorporate the latent factors into the DLM. Products with irregular, lower-valued sales may be more suited to a lower-level DGLM whereas products with intermittent sales will need a DCMM to account for the zero-valued observations. In this manner the model design can be tailored to the nature and needs of each product-specific time series.

The estimation of latent factors common to all product time-series, such as seasonality, is performed at an aggregated level in order to circumvent the noise which often clouds the individual product-level time series. Passing these latent factors to the individual time series allows the model to utilize the power of all the data whilst still producing product specific forecasts. By decoupling the individual time series the number of parameters that need to be estimated is kept at manageable levels ensuring scalability. Furthermore the underlying components of the model such as trend, seasonality and regression effects are conceptually easy to grasp and can provide valuable managerial insights. The models are of a dynamic Bayesian state-space form, meaning the data is modeled on their natural scale on the product level whilst also allowing the parameters to vary over time, whilst the sequential learning and updating of the underlying DGLM/DLM ensures the model remains up-to-date. Lastly, Bayesian estimation of the parameters provides insight into the uncertainty associated with each parameter estimate and thereby each forecast, allowing modellers to account for this uncertainty in their decision-making process.

To test the merits of the proposed model framework a case study was designed in which the forecasting performance for products with varying demand levels and frequencies could be evaluated in a contemporary context. The subject data set contains the wholesale orders placed in the US state Iowa for all liquor products, from vodkas with daily sales running in the thousands to speciality liquors sold once every few days. Given that all the products were of an identical nature, namely alcoholic beverages, the assumption that certain common factors affected each of them was reasonable and therefore the added benefit of the aggregated latent factor estimation could be tested. The contemporary context in this setting refers to the COVID-19 pandemic, which had an immense impact on almost all industries, as such a study on the liquor industry over the past year would feel incomplete without accounting for the effect of the pandemic. Together these lines of research materialize in the following research questions:

1. Does the multi-scale dynamic count mixture model outperform a traditional Generalized Linear Model (GLM) in terms of forecasting performance at the product level for individual stores?
2. Have there been significant shifts in liquor consumption in Iowa since the outbreak of the COVID-19 pandemic?
3. Has the composition of liquor demand changed? Following the enforced closure of local bars & restaurants, did liquor demand shift towards more expensive, high-end brands than before?

Before the conclusions drawn for the aforementioned research questions are presented, this paper ensues with a thorough description of the modelling process, with the traditional literature review, data description and methodology sections outlined. Specifically, Section 2 provides a review of the relevant literature on Bayesian forecasting in dynamic models, with a particular focus on the forecasting of sales of (intermittent) low-valued count series and multi-scale modelling approaches. Section 3 continues with a detailed description of the data set used in this research, including elaborating on the various aggregations possible and the importance of determining the right ones. These modelling choices are consequently detailed in Section 4. The methodology and underlying theory pertaining to Dynamic (Generalized) Linear Models, Bayesian estimation and the employed forecast metrics are presented in Section 5 followed by the results of the case study into Iowa liquor wholesales in Section 6. Thereafter Section 7 discusses possible limitations of this study and opportunities for further research, final conclusions are given in Section 8.

2 Literature Review

To properly present the relevant literature on intermittent sales forecasting, it would be instructive to first lay out the landscape of existing forecasting techniques to place where dynamic count mixture models fit in and how they compare to other models which have been designed for similar purposes. Since the early 1970's three broad categories of sales forecasting models have been defined: qualitative, time series projection and causal models, (Chambers et al., 1971). The former of which relies on qualitative data, e.g. expert opinions, to generate forecasts, the most famous example arguably being the Delphi Method (Helmer, 1994), where a panel of experts is surveyed and their opinions pooled. This approach is often chosen when data is scarce and generally under-performs compared to the other two categories in terms of forecasting power (Mahmoud, 1984). The second approach has seen an explosion in forecasting accuracy in recent years, particularly from Machine-Learning algorithms such as XGBoost (T. Chen & Guestrin, 2016) and variants of Neural Networks. A prime example being (Zhu & Laptev, 2017) who propose a Bayesian deep model based on Long Short Term Memory (Hochreiter & Schmidhuber, 1997) for large-scale time series. Whilst time series projection and causal models are heavily linked, both relying on historical data, with the latter often embedding the results of the former in its process, the crucial distinction between the two lies in their purpose of use. Whereas projection models and algorithms are designed with one key aim, namely the highest possible forecasting accuracy, causal models also intend to map the relations at play between the variable of interest and it's drivers. It is this category to which dynamic count mixture models belong.

2.1 State-Space Time-Series Forecasting

Given the long history of academic research aimed at forecasting time series, a wide range of causal model types have been developed over the years. As the multi-scale approach of (Berry & West, 2019) aims to forecast multivariate time series via decoupling them into univariate DLMs, the literature is presented of univariate and multivariate models alike. DCMMs belong to the state-space class, meaning they are probabilistic graphical models that illustrate the probabilistic dependence between latent state variables and the observed measurements (Koller & Friedman, 2010), a class that has been broadly researched since (Kalman, 1960) published his recursive solution to the discrete-data linear filtering problem. Among this class are a group of more traditional models known as 'ETS' models, which consist of an error, trend and seasonality component aside from the level component. Exponential smoothing (Hyndman et al., 2008) is an example often used, where forecasts are produced by weighing averages of past observations, with weights eroding as observations become older. Most well known are perhaps the Vector Auto-Regressive Integrated Moving-Average (VARIMA) models (Box et al., 2008) and variations thereof. The most appropriate variation of these models for the case study of Iowa liquor wholesales would be the SVARIMAX, which extends the VARIMA by appending a seasonal component and allowing for the inclusion of exogenous regressors. In a case study similar to this paper, (Arunraj & Ahrens, 2015) propose a hybrid SARIMA and Quantile-Regression model to forecast the sales the daily food sales of a retailer and yield promising results.

Whilst these models are versatile and are capable of producing accurate forecasts they often suffer under the vast amounts of parameters that need to be estimated. To illustrate, a VAR model consisting of 10 dependent variables generates a contemporaneous co-variance matrix of error terms consisting of 100 elements. To address the high-dimensionality issue, several methods have recently been proposed. (Wang et al., 2021) restrict the parameter space by rearranging the transition matrices of the model into a tensor and applying tensor decomposition, whereas (Ding et al., 2017) propose a hybridized kernel smoothing and ℓ^1-regularized method to estimate these matrices. However despite the recent advancements, these models still become increasingly complex when faced with high-dimensional data sets, are computationally intensive and require repeated, expensive inference as new data becomes available. Another popular state-space model is the conditionally Gaussian/linear state-space model (West & Harrison, 1997; Prado & West, 2010), which this papers employs at the aggregated sales levels. A development of these models is proposed in (Yelland & Lee, 2003), who recognize that a single DLM is often insufficient to model the entire lifetime of a time-series and thus introduce Dynamic Linear Mixture Models. A DCMM is in a broad sense an extension to this framework wherein two DGLMs are mixed as opposed to DLMs.

2.2 Intermittent Count Time-Series

However, whilst the aforementioned methods are flexible they often fail in the face of time series which exhibit low counts or frequent zero observations along with high volatility (Croston, 1972). To overcome this, a number of novel forecasting approaches have been developed; (McCabe & Martin, 2005) build forecasts from the p-step forward predictive mass functions of distributions nested within the Integer-Valued First-Order Auto-Regressive (INAR(1)) class. An empirical survey of the relevant

forecasting methods is given in (Yelland, 2009), however many new approaches have since been proposed. A popular research direction is that pertaining to adapted Poisson or Negative-Binomial(N-B) models for count series, (C. Chen et al., 2016) focus on an auto-regressive conditional N-B model whereby they also account for time-varying over-dispersion, a common issue with Poisson/N-B models. In an earlier study, (Snyder et al., 2012) compare a Poisson, N-B and hurdle-shifted Poisson model to achieve the same, ultimately favoring the latter. Hurdle-shifting, where the forecasting is split in two parts, one to determine whether there will be a sale and the other to estimate the magnitude of the sale, is an important concept. It debuted in (Croston, 1972), who used a Bernoulli process to distinguish between active and inactive selling periods, DCMMs build on this conceptualization. As do (Jiang et al., 2019), who classify intermittent demand data into zero and non-zero values, to afterwards fit non-zero values into a mixed zero-truncated Poisson model, yielding better results than the Poisson and hurdle-shifted Poisson models for their case study. Another critical research is that of (Kolassa, 2016) who recognized that traditional evaluation metrics as the MAD and MASE are inherently unsuitable for count data and therefore propose new metrics.

2.3 Bayesian Estimation of State-Space Models

Bayesian estimation methods in general have seen a spike in popularity brought about by the recent surge in computing power. Their ability to provide a coherent approach for specifying sophisticated hierarchical models for complex data (Calder & Cressie, 2009) along with the ability to showcase the uncertainty of estimates, has made them popular in both academic and commercial research. This is also the case for state-space models of multivariate time-series, a survey of recent researches in this field is given in (West, 2020). In order to estimate the parameters of the state-space based DCMMs this paper employs Sequential Bayesian analysis. A notable study also based on this analysis with comparable intentions is (Aktekin et al., 2018), who develop a new class of dynamic multivariate Poisson count models, Multivariate Poisson-scaled beta (MPSB) models. Whilst their model shows encouraging results it has an identifiability issue when dealing with uninformative priors and does not have a mechanism to deal with sparsity. Another form of Sequential analysis is Particle Learning (Carvalho et al., 2010), an extension of the Sequential Monte Carlo methods known as Particle Filters (Gordon et al., 1993). The key idea behind this methodology is represent the density of the state vector through a set of random samples, which are consequently updated and propagated via an algorithm. This approach is more elastic than a Kalman-Filter based method, as it does not require restricting assumptions as linearity or a Gaussian distribution of the noise, it is however computationally demanding. Nonetheless, the approach achieves favorable results compared to existing models and has the added benefit of "being an intuitive and easy-to-implement computational scheme" and as such is a promising line of research which warrants further study. A case study which demonstrates the power of the method in the context of daily retailer sales is (Ping et al., 2018). Other applications of Bayesian forecasting for dynamic multivariate state-space models include (Nakajima & West, 2013) who model dynamic sparsity in multivariate time series by thresholding latent processes to adaptively actuate zero values and link these to time-varying parameters, this is done to generate natural variable selection. (Zhou et al., 2014) extend this by including dependencies between the dynamic latent factors. However these models are forms of Time-Varying Auto-Regressions (TV-VAR) (Kitagawa &

Gersch, 1996) and therefore become highly complex as the number of time series grows making them less suitable for our case study.

An alternative family of techniques within the world of Bayesian inference is that of Variational Inference (VI) (Gordon et al., 1993). They are particularly useful as a substitute for Monte Carlo sampling methods for complex distribution that are problematic to sample or evaluate directly. Recent advancements in this field have led to new approximate inference methods which are capable of estimating non-Gaussian models (Archer et al., 2015). Whilst these methods are faster than traditional sampling, the deriving of the equations needed to iteratively update the parameters is complex, laborious work (Zhang et al., 2018). If applied correctly they have still promising outcomes, in a case study similar to this one, (Seeger et al., 2017) apply the methodology when forecasting intermittent demand in linear state-space models. This paper employs VI to forecast and update the individual DGLMs within each DCMM.

2.4 Multi-Scale Modelling

As previously touched upon, this paper employs a multi-scale modelling approach whereby multivariate time series are decoupled into univariate series thereby ensuring scalability whilst still allowing information to be shared across time-series via latent factors. Multi-scale refers to a modelling design, originating from the natural sciences, in which multiple models are used at different scales of resolution to simultaneously describe a system (E & Lu, 2011). The exact approach employed is a decouple-recouple framework which has seen several interesting applications in recent literature. (Zhao et al., 2016) first devised the concept in their Dynamic Dependence Networks Models (DDNM) which define coherent multivariate dynamic models through the coupling of customized univariate DLMs. This concept is extended in Simultaneous Graphical Dynamic Linear Models (SGDLM) (Gruber & West, 2017, 2016) where each series is permitted to be a contemporaneous predictor of any other. Whilst both these models are suitable for sales forecasting of a large catalogue of products, they rely on the assumption of Gaussian distribution of the noise which is not realistic for products with intermittent demand. The approach has also led to innovations in the context of dynamic networks, (X. Chen et al., 2018, 2019) apply it twofold. Firstly to split a conditionally multinomial variable into Poisson DGLMs which model the in-flows into a network, to later map these DGLMS to Dynamic Graphical Model. The primary aim of these models is however not forecasting precision but rather "characterizing normal patterns of stochastic variation in flows and allowing models to monitor and adapt to these changes", rendering them less suitable to sales data. Final mentions should be made of a recent paper which is strongly linked to DCMMs, (Berry et al., 2020) initially employ DCMMs to forecast individual consumer transactions and consequently introduce Dynamic Binary Cascade Models to predict the counts of items per transaction.

3 Data

The ensuing section provides an overview of the variables used, which have been compiled from several publicly available data sets. These sources will be mentioned at the outset of each subsection, in which a summary of each included variable is given, beginning with the dependent variable and

continuing with the predictors. Each subsection is concluded with a report of the data preparation applied, if any was deemed necessary. Before doing so, further explanation of the multi-scale approach is in place to justify certain choices made. Each model framework specification consists of two models, one higher-level DLM whose main purpose is to generate accurate forecasts of the common latent factors and a lower-level DLM, DGLM or DCMM which relies on these latent factors as predictors and whose main purpose is to forecast the dependent variable. The higher and lower-level models therefore rely on different predictor sets. This distinction is crucial as certain aggregations or data treatments were employed on particular variables in the higher-level DLMs, in order to ensure that the corresponding effect of these variables was captured as accurately as possible via the latent factor.

3.1 Iowa Liquor Wholesales

1. Iowa state government website:
   `https://data.iowa.gov/Sales-Distribution/Iowa-Liquor-Sales/m3tr-qhgy/data`

Wholesales of liquor are heavily regulated and catalogued in the midwestern US state of Iowa, for Class E liquor license holders this data is extensive and publicly available. Class E liquor licenses pertain to grocery, convenience and liquor stores and allow for the sale of liquor with an alcohol percentage of 30% or higher for off-premise consumption. As such it contains all the purchases of liquor by Class E liquor license holders from the Alcoholic Beverages Division (ABD), these include grocery & liquor stores but exclude bars & restaurants. It should be noted that the consequent purchases of consumers at these stores is not included. This is an important distinction given the different characteristics of consumer sales data and wholesale data. To illustrate, the former is often characterized by heavy weekly seasonality with increased purchasing on the weekend days whereas the latter sees stocking up on the days leading up to holidays. The data set includes observations starting in January 2014 and ending in January 2021, for each observation 21 variables are included, aside from the date, which can be classified as either geographic, categorical or order-specifics (e.g. six-pack or bottle volume). Unique 4/5 digit *Store, County, Category, Vendor & Item Number Codes* are included for identification purposes as well as the wholesale and retail price of the item in question. These aforementioned variables primarily served for aggregation purposes, to illustrate, if the aim was to forecast the sales of Whiskies in a certain county then all the observations with the corresponding *County Number* and *Category* fields could be filtered out. The data set provided two different dependent variables, *Bottles Sold* and *Sales*. In all bar one model specification y_t represented *Sales* or *logSales*, as ultimately this deemed to be the main dependent variable of interest to the commercial stores and manufacturers. The one exception being DCMMs which aimed to forecast the intermittent orders of a particular product from an individual store. Here *Bottles Sold* served as a directly proportional proxy to *Sales*, given that the price of these products remained unchanged for the given time-frame. This substitution was made as the distribution of *Bottles Sold* was found to be closer to the required Poisson distribution on which the underlying Poisson DGLM component of DCMMs relies.

3.1.1 Data Preparation

Before analysis could be performed, the data set had to be cleaned and certain practical adjustments applied. The original data set included a total of 108 different categories which were deemed to be unnecessarily specific, examples include "Tropical Fruit Schnapps" and "Bottled in Bond Bourbon". It was chosen to aggregate these categories down to 11 broad categories which are listed below:

1. Vodka
2. Brandies
3. Liqueurs
4. Tequila
5. Whiskies
6. Cocktails
7. Schnapps
8. Rum
9. Gin
10. Spirits
11. Specials

This choice was made in consideration of the fact that the category time-series would primarily be estimated to generate latent factors which could consequently be passed down to the individual product-level as predictors. By down-scaling from 108 to 11 categories, each category time-series would have significantly more observations, thereby breeding preciser estimates. Furthermore it was reasoned that the loss of specificity brought about by the aggregation would be of limited-to-no impact. Estimating the effect of any latent factor on "Grape Schnapps" and "Spearmint Schnapps" separately rather than on "Schnapps" seemed excessively detailed.

The ensuing step was to treat the data for missing values. In order to be deemed valuable, an observation needed to include *Date, Sale, Bottles Sold* in addition to geographical and categorical information of any level. Entries lacking the *Date*, *Sale* or *Bottles Sold* fields were dropped altogether, this was however only the case for 10 observations. Observations which lacked item-specific categorical information would still be of value in the category-level DLMs, just as observations which lacked store-specific geographical information could be in county or state-level DLMs. To ensure no observations were needlessly dropped on the basis of missing geographic or categorical information, mappings were made of all the related variable groups, e.g. geographic $\rightarrow$ (*Store Number, Store Name, Address, City, Zip Code, County Number, County Name*). Doing so served a dual purpose, firstly to impute missing values from observations with identical inputs in related fields. To illustrate, if an observation lacked both *County Name, County Number* but included the *City* field, the County information could be imputed from the *City:County* mapping. Secondly these mappings acted as means of cross-referencing to eliminate incorrect entries, if an observation originated from the same store as another but not all geographical variables were identical, it suggested one of the fields was incorrectly entered. After all data points were cross-referenced for false input and imputed accordingly, the observations still lacking necessary information were dropped. This was the case for 79,927 observations accounting for 0.4% of the total set, ultimately resulting in 16,203,716 complete observations. Given the relatively small number of observations with missing fields, their exclusion is not expected to have a significant impact on the estimation of the model.

Since it's inception the ABD has maintained ordering windows of Monday through Friday. However, included within the data set were certain pre-2018 observations which appeared to have taken place on a weekend day (Saturday or Sunday). The website of the Iowa ABD notes that these are the result

of a change in administration implemented in 2018 regarding weekend orders. They state that these 'false' weekend orders were most likely listed by delivery date and not order date, suggesting that the order presumably occurred on the preceding Friday. The dates of these observations were therefore adjusted accordingly.

Further consideration had to be made for the absence of observations on national holidays. This was of particular importance for estimating the seasonality, as the procedure for doing so functioned on an iterative basis and gaps in the time series would therefore distort estimates. To avoid such miscalibration, it had to be assured that entries where included for all weekday dates that took place between the first and last date of the set. Therefore if certain dates were absent in the data set pertaining to the orders of a specific store, it simply suggested the store did not place orders on these dates and a zero-valued observation was appended. However if dates were absent which coincided with national holidays, these were considered false zero-observations as the ABD was closed meaning stores could not possibly place an order. These dates were therefore appended as 'NaN'-observations, which were simply skipped during analysis.

3.2 COVID-19 Cases & Deaths

2. County level COVID-19 data from the New York Times:
   `https://github.com/nytimes/covid-19-data`

The New York Times maintain a publicly available data set on Github which lists the number of new COVID-19 cases and the number of COVID-19 related deaths per day at a county level for the United States. The data set was complete so no imputation had to be performed, it was however presented on a cumulative scale. Portraying the sum of all the preceding COVID-19 cases and deaths within a certain county for a particular date. As these variables were included to reflect the magnitude of the spread of the virus at any given time, it was instead chosen to first-difference the values resulting in the number of new COVID-19 cases and deaths for each given date. It was further reasoned that the nature of the data made it prone to outliers, primarily caused by delayed reporting, a common example being new statistics from weekend days not being disclosed till the following weekday producing inflated numbers for these days. To account for this, the seven-day average was instead taken to which one final transformation was applied, namely a *log*-transformation. The logic behind this decision being that the COVID-19 statistics were included to not only depict the level of spread of the virus but also the fear this spread breeds amongst consumers, who may be more hesitant to travel to grocery and liquor stores in times of elevated spread. It seems unreasonable to model this fear on a linear scale, a doubling of the number of cases from 10 to 20 is unlikely to lead to twice as much fear. The *log*-transformation was therefore applied to remove the assumption of linearity.

3.3 Bar & Restaurant Restrictions

3. COVID-19 related county/state level restrictions:
   `https://iowastartingline.com/iowa-covid-19-timeline/`
   `https://ballotpedia.org/Documenting_Iowa`

In order to properly model the effects of the COVID-19 pandemic on the wholesales of liquor a

measure must be included which reflects the level of restrictions on freedom of movement. The Iowa state government has thus far been relatively reluctant to impose travel restrictions on it's citizens, with the state not yet seeing a total lockdown nor have the grocery stores been forced to closed during certain hours. From the aforementioned sources a list of imposed restrictions was compiled, of all these the ones that would logically have an impact (via a substitution effect) on liquor wholesales in grocery and convenience stores were the restrictions on restaurants and bars. A timeline of these restrictions is presented in Appendix A.1. Upon construction of the timeline it was noted that aside from restrictions during the initial lockdown, which took place when the virus first hit in the Spring of 2020, the majority of the 99 counties saw little-to-no further restrictions. Only a few, predominantly densely populated counties containing major cities, saw returning restrictions throughout the year. To best model the impact of these restrictions on wholesales, it was therefore chosen to discard the smaller counties which only saw one period of restrictions and focus on the six counties which were the most affected: Polk, Johnson, Linn, Story, Dallas & Black Hawk. To incorporate this information a categorical variable was initially included with four levels, whereof the first was omitted from the analysis as a baseline:

0. No restrictions on bars and restaurants.
1. Bars and restaurants can open up to 50% capacity.
2. Bars must be closed but restaurants can open up to 50% capacity.
3. Bars and restaurants must be closed.

Furthermore, it should be mentioned that governor Kim Reynolds intentionally chose not to impose any restrictions on the transport of goods in Iowa in a bid to minimize the impact on the state economy. As such there is no reason to consider that changes in demand were caused by COVID-related stocking problems for the retailers.

3.4 Climatic Data

4. Temperature & precipitation data from the Iowa State University:
   `https://mesonet.agron.iastate.edu/request/coop/fe.phtml`

Recent studies on the relation between temperature and alcohol sales in the United States show a strong correlation, particularly in colder climates such as Iowa (Hirche et al., 2021). For the six aforementioned counties, temperature was thus included as a predictor. Each of the counties housed several weather stations from which daily high and low temperatures were obtained, these were consequently averaged over the stations as well as over both temperature measurements to spawn a mean temperature. Aside from temperature, precipitation levels were also included as a regressor.

3.5 Google Mobility Reports

5. Changes in movement trends drawn from Google mobile phone data:
   `https://www.google.com/covid19/mobility/`

At the request of public health officials worldwide, Google began publicly releasing Community

Mobility Reports to help policymakers make more informed decisions in the fight against COVID-19. The reports aim to present how public response changes to imposed policies via movement trends, generated from the mobile phone data of Google users. The data relay how the number of visitors that travel to six categorized locations change in comparison to baseline days, which have been determined over the 5-week period from January 3rd through to February 6th 2020. The baseline consists of 7 values, one for each day of the week, to avoid comparing changes in movement between weekends and weekdays. The six categories for which trends are charted are: parks, transit stations, workplaces, residential, retail & recreation and groceries & pharmacies, of which the final two are particularly relevant for this case study. Every store found in our data set can evidently be placed in one of these categories, the majority under grocery stores but the liquor stores are classified as retail. For each store, be it grocery or retail, both variables were included to test for a possible substitution effect. On days where Google lacked sufficient data to "confidently and anonymously estimate the change from the baseline", no values were included. Table 1 displays the number of missing values out of 318 for the six chosen counties. These values were imputed from the state-average values for the corresponding days.

County	Retail	Grocery
Black Hawk	0	18
Dallas	18	22
Johnson	0	6
Linn	0	0
Polk	0	0
Story	6	25

Table 1: Missing Values

3.6 Holidays

To account for a holiday effect, seen through increased orders on days around a national holiday, an indicator variable was included. As previously mentioned, for most national holidays the ABD closed it's doors and thus any holiday effect would have to be observed on the days surrounding the holiday. To form an initial prediction of how many days, either before or after the holiday, should be marked as having a holiday effect, the state-wide sales were plotted over time with holiday dates highlighted as can be seen in Appendix A.2. A gradient analysis was subsequently performed, focusing on the number of days around the holidays which saw positive deviations from the current trend. It was estimated that this was strongest on the two days directly after a holiday, suggesting stores generally re-stocked after holidays rather than stocking up beforehand in anticipation of increased sales. This served as a starting point which would later be assessed via model selection.

3.7 Data Exploration

Once the data set had been properly treated and completed, primary data exploration was performed. The principal objective in doing so was to get a grasp of any long-running trends and strong correlations at play which affected all time-series. It was therefore chosen to focus on the state-wide aggregated wholesales to best capture these 'global' relations. A plot of the state-wide aggregated sales for all products over the period 2017-2020, Figure 1, showed a strong seasonal pattern as well as an increasing trend. Each year appears to show an linear rise, from January to December, followed by an acute drop in the following January. Closer inspection of Figure 2, which presents a box-plot of the monthly levels, confirms this and also suggests a slight dip in the summer months of July and August. When evaluated next to a plot of the average temperature over the same period, as seen in

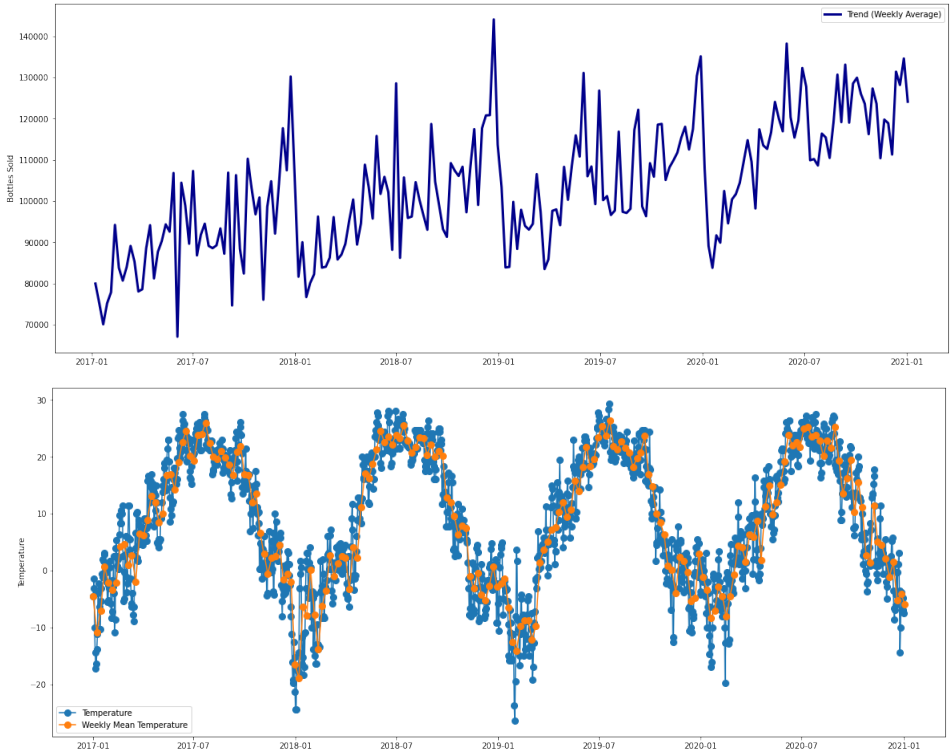


Figure 1: 1A: Total state-wide sales of all liquor products between 2017-2020 & 1B: Temperature and weekly-averaged temperature (orange) between 2017-2020

Figure 1, correlation between the two seems to follow a pattern. For the first six months of the year, temperature and sales seem to grow concurrently, until the summer months arrive when their paths seem to deviate. A remarkable feature of the top graph in Figure 1, which plots the total sales, is that there appears to be no significant change around the time the pandemic first hit Iowa (March 2020). To inspect whether this effect, or lack thereof, was visible at all granularity levels of the data the focus moved to the six counties of interest. The aggregated sales for all products across all six counties for the years 2019 and 2020 are presented in Figure 3. Where the bottom graph, pertaining to 2020, has a colour-coded highlight scheme to reflect the level of restrictions that were in place. Red denotes the

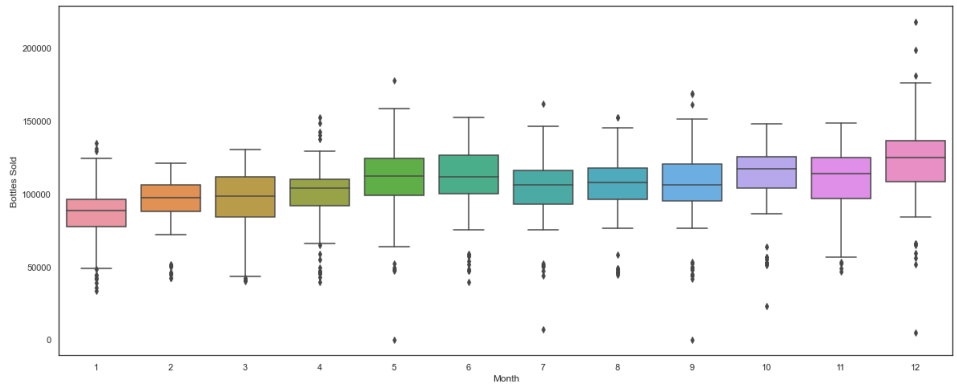


Figure 2: Box-plot of the monthly level of total state-wide sales of all liquor products between 2017-2020

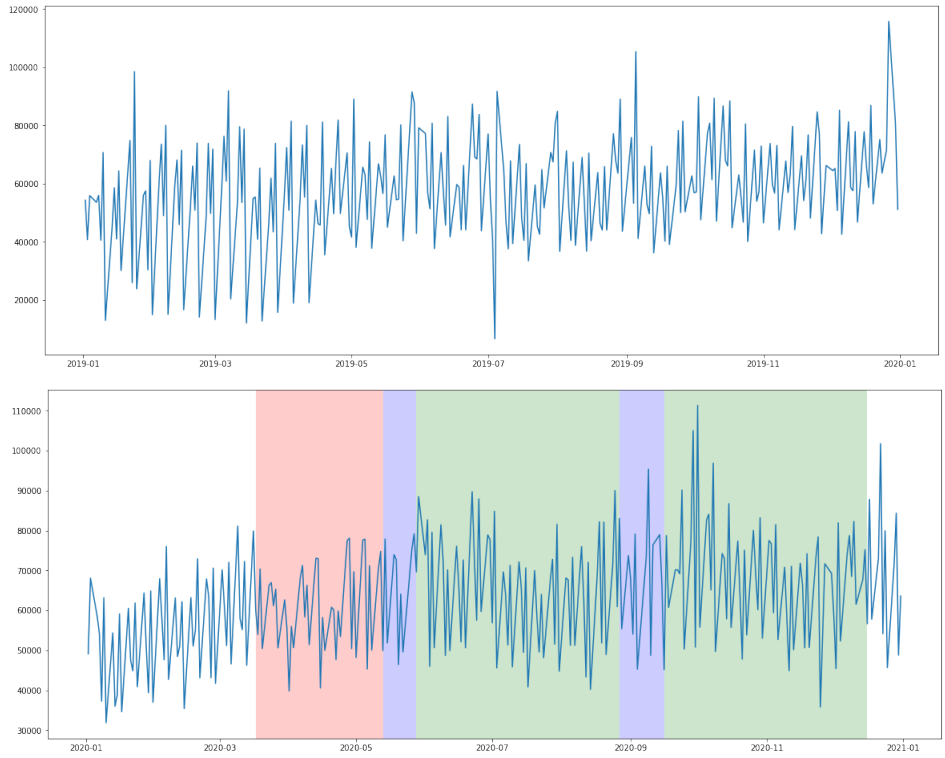


Figure 3: Total sales for all liquor products for the six aforementioned counties in 2019 (top) & 2020 (bottom)

highest level of restrictions, level 3, blue for level 2 and green-shaded areas correspond to times when level 1 restrictions were imposed. Interestingly, aside from an apparent change in volatility during the period of heaviest restrictions, there appears to be little noticeable change in comparison with 2019.

Once the zoom was adjusted to the most granular level, that of individual stores, significant differences did however become visible as portrayed in Figure 4. The two stores depicted belong to the same local chain of liquor stores and are located in the same city less than 7 kilometres apart. Whilst the store displayed on top appears to enjoy an increase in sales during the initial lockdown period, the other store sees their sales almost disappear entirely. This could be for several reasons, firstly it was theorized that the second store may have closed it's doors during this period, however this seems unlikely as both stores belong to the same chain and the store in question still placed orders, be it far smaller ones. More probable was that location played a role, whilst the top store sits at central, downtown location the bottom store lies in the outskirts of the city. These contrasting time-series were seen for stores throughout the set. Although strong conclusions could not be drawn from these plots, one note-able feature was apparent. Whilst the residents of Iowa did not appear to drastically change their level of liquor consumption during these periods, aside from a relatively small increase, the data suggests they did change their means of acquisition. Unfortunately the data set lacked information regarding whether or not stores offered online delivery, otherwise this would seem to be a plausible explanation for the disparities and is therefore an interesting line of further research. A further point of notice which surfaced during the data exploration was the differing impacts amongst

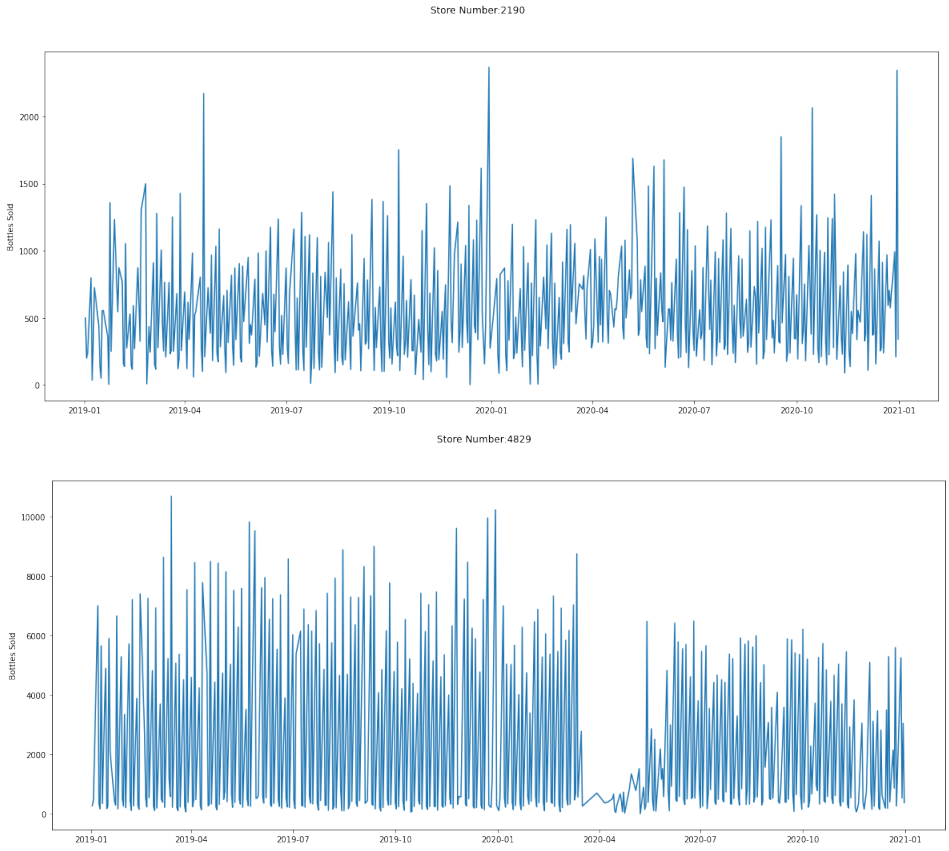


Figure 4: Comparison of the total liquor sales of two different stores in the same city between 2019-2020

the categories. As seen in Appendix A.3, the majority of categories saw relatively little change in their state-wide sales time-series, with a few exceptions. Most notably, Tequila and Cocktails, whose popularity seemed to rise once the pandemic struck. Inspection of the mobility report plots revealed a potential point of concern, namely multicollinearity. Figure 5 plots the two variables, retail and grocery, alongside one another and their movements appear to coincide perfectly which suggests that including both in the analysis may be redundant, this will therefore be tested further on.

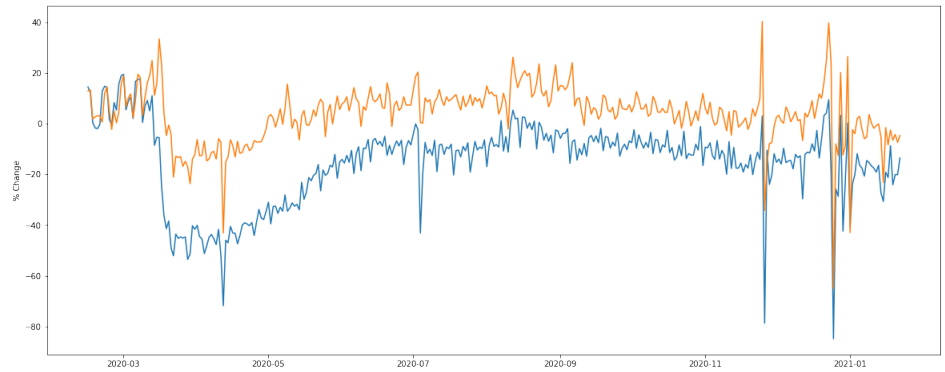


Figure 5: Time-series of the changes in movement trends for Retail (blue) & Grocery (orange) between March-December 2020

4 Modelling Approach

Upon assemblance of the data set, a model design was constructed in which consideration was made for which parties had commercial interests in the Iowa wholesale liquor supply chain. By doing so it could be determined what the dependent variables in the lower-level models, $\mathcal{M}_1$, would be to thereby test the model framework for commercial purposes. It was ultimately chosen to focus on two parties, the ABD themselves as well as individual stores. To establish which aggregation and multi-scale design suited each of the two parties a new look was cast upon the data set. The data could be decomposed along three general dimensions, the first termed 'geographic' which represented the scale of aggregation of retailers, the second, 'categorical', which acted equivalently for products and the third, 'temporal', denoted whether the data was on a daily or weekly basis. A total of 18 various aggregation possibilities were therefore possible, which are presented in Table 2. For the ABD, forecasting the order timeline of all 2387 stores individually would be needlessly exact. From an operational standpoint greater gains could be made if they obtained accurate forecasts of the total demand that was to be expected, for stocking purposes, as well as a decomposition of this demand along category/county lines for logistical purposes which could be modelled on a daily scale. For this specification the higher-level model $\mathcal{M}_0$ would therefore model state-level sales whereas the lower level model $\mathcal{M}_1$ would forecast county-level sales. For individual stores a weekly aggregation proved to be more suitable given systematic differences between weekdays, with stores often implementing fixed order schedules making daily order forecasting redundant. Furthermore, given that individual stores by definition lay on the most granular level of the geographic dimension, the difference between their higher and lower-level models was between product categories and individual products. Once the dependent variables have been determined for both layers of each specification, the appropriate model could be chosen. This was done by examining the distribution of the chosen dependent variable, as well as the distribution of the errors produced by an OLS regression of $Y_t = \boldsymbol{X}'\boldsymbol{\beta}$. For the majority of these a DLM proved to be most suitable, often after a log-transformation, the only exceptions were for the time-series of the sales of one specific product in an individual store, where a DCMM proved more appropriate given that stores generally did not order specific products every week but rather once every few weeks.

Geographic	**Categorical**	**Temporal**
State	All	Week
County	Category	Day
Store	Product	

Table 2: Possible Aggregations

Determining which regressor would be placed on which level was straightforward. The majority of the regressors were county-specific (Climatic, COVID-19, Restrictions and Mobility) and these therefore had to be placed on the county-level of the first specification, the remaining variables were placed at the state-level. For individual stores all regressors were placed within the higher-level model, as the order data for one product for a single store often did not provide enough data points to estimate the effect of all the regressors. For all specifications, the regressors estimated at $\mathcal{M}_0$ would be passed to $\mathcal{M}_1$ as latent factors. To summarize, an overview of the complete model specifications is presented in Table 3.

	Model Layers	Dependent Variable Y_t	Regressors F_t
ABD	$\mathcal{M}_0$: DLM $\mathcal{M}_1$: DLM*	State sales of all products County sales of product category	Holidays & Seasonality (Daily & Monthly) Climatic, COVID-19, Restrictions & Mobility
Individual Stores	$\mathcal{M}_0$: DLM $\mathcal{M}_1$: DCMM*	Store sales of a product category Store sales of one product	All

Table 3: Overview of the 2 model specifications with layers, dependent variables and regressors. * is included for the $\mathcal{M}_1$ level models to indicate these are latent-factor models.

5 Methodology

The three different model types that are placed on the varying levels are thus, Dynamic Linear Models, Dynamic Count Mixture Models and Dynamic Generalized Linear Models within the DCMMs. The ensuing section serves to expand upon the structure, updating and forecasting of each of these, in the aforementioned order. Initially, the working of each of these models will be laid out without the inclusion of latent factors, after which subsection 5.4 will outline how latent factors can be incorporated into DGLMs and DLMs respectively.

Before diving in, mention should be made of the hyper-parameters present in the model, namely the discount factors, $\beta, \boldsymbol{\delta}, \rho$, each denoted by a different symbol as they discount different components. These need to be set prior to analysis, with their importance being paramount as they control the rate at which coefficients are allowed to change, miscalibration can therefore be costly. They determine the change in the coefficients by discounting the old, historical information thereby increasing the contribution of new information, a discount factor set at 0.99 adds $100\% - 99\% = 1\%$ more uncertainty at each time-step. The implementation of these hyper-parameters will be become apparent in the equations below.

5.1 Dynamic Linear Models

Conditionally Gaussian/linear state-space models (West & Harrison, 1997; Prado & West, 2010), also known as DLMs, have become a fundamental tool for modellers since their inception. Their flexible linear structure makes them widely applicable whilst the dynamic nature of the model allows for the modelling of more complex relations than a standard linear model. Their merits become even clearer from a Bayesian perspective, with their parameters defined by known distributions at each time step. To present the working of a DLM, the full equation system for an arbitrary time point $t-1$ will be given first, followed by the updating steps from $t-1$ to t, lastly mention will be made of the parameter initializations in section 5.5 which will complete the full DLM specification.

The structure of a standard univariate DLM is of a state-space form, characterized by an observation equation which relates the dependent variable to the regressors via dynamic regression as shown in (1). Where as per usual, Y_t represents the dependent variable at time t, $\boldsymbol{F}_t$ a vector of regressors and constants, $\boldsymbol{\theta}_t$ the state vector and the observation error is v_t with variance $1/\phi_t$.

Observation: $$Y_t = \boldsymbol{F}_t'\boldsymbol{\theta}_t + v_t \quad v_t \sim \mathcal{N}(0, 1/\phi_t) \tag{1}$$

A system equation (2) describes the structural evolution of the state vector via the state evolution

matrix, $\boldsymbol{G}_t$, and state evolution error, ω_t, which is T-distributed with $n_t - 1$ degrees of freedom and has variance $\boldsymbol{W}_t$.

System: $$\boldsymbol{\theta}_t = \boldsymbol{G}_t\boldsymbol{\theta}_{t-1} + \boldsymbol{\omega_t} \quad \boldsymbol{\omega_t} \sim T_{n_t-1}(\mathbf{0}, \boldsymbol{W}_t) \tag{2}$$

If $\boldsymbol{\theta}_t$ were to be k-dimensional vector, then $\boldsymbol{G}_t$ would therefore be a kxk-dimensional matrix. In the remainder of this paper $\boldsymbol{G}_t$ will be a block-diagonal matrix wherein the diagonal elements each correspond to the coefficients of regressors found in $\boldsymbol{\theta}_t$. This matrix will thus consist of ones along the diagonal and zeros elsewhere, the only exceptions are for model specifications with two trend components which will add a single off-diagonal 1 to the matrix, whilst the block-diagonal elements of $\boldsymbol{G}_t$ which correspond to the seasonality predictors will be filled with harmonic component matrices ((West & Harrison, 1997) Section 8.6). To illustrate the working, an exemplary $\boldsymbol{F}_t$ and $\boldsymbol{G}_t$ are given below for a model which includes a single trend component, a indicator variable for holidays and a Fourier seasonal component of period length 5 to account for weekday seasonality, where $\boldsymbol{H}_1, \boldsymbol{H}_2$ are the harmonic component matrices.

$$\boldsymbol{F}_t' = (1, \text{Holiday}_t, 1, 0, 1, 0) \quad \& \quad \boldsymbol{G}_t = \text{diag}[1, 1, \boldsymbol{H}_1, \boldsymbol{H}_2]$$
$$\text{where} \quad \boldsymbol{H}_1 = \begin{pmatrix} \cos(2\pi/5) & \sin(2\pi/5) \\ -\sin(2\pi/5) & \cos(2\pi/5) \end{pmatrix}, \quad \boldsymbol{H}_2 = \begin{pmatrix} \cos(4\pi/5) & \sin(4\pi/5) \\ -\sin(4\pi/5) & \cos(4\pi/5) \end{pmatrix}$$

A common extension is the allowance of dynamic observational variance, primarily imposed via a Beta-Gamma stochastic volatility model for the observational variance((West & Harrison, 1997), Section 10.8). To impose this, the variance of the v_t is given a structural evolution equation referred to as the precision, presented in (3), in which the reciprocal of the observation variance ϕ_t is multiplied by the Gamma-distributed γ_t and divided by the discount factor β, where n_{t-1} is once more the degrees of freedom parameter whose initialization and evolution will be presented below.

Precision: $$\phi_t = \gamma_t\phi_{t-1}/\beta \tag{3}$$
$$\text{with} \quad \gamma_t \sim \text{Beta}(\beta n_{t-1}/2, (1-\beta)n_{t-1}/2) \tag{4}$$

The parameters of interest are thus $\boldsymbol{\theta}_t$ and $\boldsymbol{\phi}_t$, the information available regarding these parameters at any point in time is summarized by the their respective priors and posteriors. These are displayed below in (5)-(9) in which the symbol $\mathcal{D}_{t-1}$ encompasses all the information, parameters and regressors, available at time t-1. Firstly for $\boldsymbol{\theta}_{t-1}$, the posterior at $t-1$ (5) is T-distributed with n_{t-1} degrees of freedom, mean state vector of $\boldsymbol{m}_{t-1}$ and state covariance matrix $\boldsymbol{C}_{t-1}$. Via the evolution defined in (7) the posterior for $t-1$ becomes the prior for t (6).

Information: $$(\boldsymbol{\theta}_{t-1}|\mathcal{D}_{t-1}) \sim T_{n_t-1}(\boldsymbol{m}_{t-1}, \boldsymbol{C}_{t-1}) \tag{5}$$
$$(\boldsymbol{\theta}_t|\mathcal{D}_{t-1}) \sim T_{n_t-1}(\boldsymbol{a}_t, \boldsymbol{R}_t) \tag{6}$$
$$\text{with} \quad \boldsymbol{a}_t = \boldsymbol{G}_t\boldsymbol{m}_{t-1} \quad \& \quad \boldsymbol{R}_t = \boldsymbol{G}_t\boldsymbol{C}_{t-1}\boldsymbol{G}_t' + \boldsymbol{W}_t \tag{7}$$

In an identical fashion (8) and (9) present the Gamma-distributed posterior for ϕ_t and prior for ϕ_{t-1}, in this notation s_{t-1} denotes the $t-1$ estimate of the observational variance $1/\phi_{t-1}$.

$$(\phi_{t-1}|\mathcal{D}_{t-1}) \sim G(n_{t-1}/2, n_{t-1}s_{t-1}/2) \tag{8}$$

$$(\phi_t|\mathcal{D}_{t-1}) \sim G(\beta n_{t-1}/2, \beta n_{t-1}s_{t-1}/2) \tag{9}$$

Having defined $\boldsymbol{C}_{t-1}$, the setting of $\boldsymbol{W}_t$ can now be clarified, which in accordance with ((West & Harrison, 1997), Section 2.4.2) is set at $\boldsymbol{W}_t = \boldsymbol{C}_{t-1}(1-\delta)/\delta$ to ensure the error is proportionally related to the initial variance, where δ is the discount factor for regressor information. Varying discount factors for different regressors and latent factors are implementable, where $\boldsymbol{\delta}$ would then take a vector form.

5.1.1 Updating & Forecasting

Before the next observation, y_t, becomes available and the updating can commence a forecast distribution is generated based on the state vector prior at t (6). In which the mean and covariance of this distribution are generated via vector multiplication with the regressors $\boldsymbol{F}_t$ while accounting for the observational variance via s_{t-1}.

Forecast:
$$(Y_t|\mathcal{D}_{t-1}) \sim T_{\beta n_t - 1}(f_t, Q_t) \tag{10}$$

$$\text{with} \quad f_t = \boldsymbol{F}_t'\boldsymbol{a}_t \quad \& \quad Q_t = \boldsymbol{F}_t\boldsymbol{R}_t\boldsymbol{F}_t + s_{t-1} \tag{11}$$

From there the parameters are updated according to the following Kalman-filter based equations (12)-(17) in which e_t defines the difference between the forecast and the realized observation and together with $\boldsymbol{A}_t$ forms the 'Kalman filter gain' which is applied to $(\boldsymbol{a}_t, \boldsymbol{R}_t)$ in (14). Furthermore r_t is termed the volatility ratio, which represents the change in the estimate of the observational variance.

Updating:
$$(\boldsymbol{\theta}_t|\mathcal{D}_t) \sim T_{n_t}(\boldsymbol{m}_t, \boldsymbol{C}_t) \tag{12}$$

$$(\phi_t|\mathcal{D}_t) \sim G(n_t/2, n_t s_t/2) \tag{13}$$

$$\text{with} \quad \boldsymbol{m}_t = \boldsymbol{a}_t + \boldsymbol{A}_t e_t, \quad \boldsymbol{C}_t = r_t(\boldsymbol{R}_t - \boldsymbol{A}_t\boldsymbol{A}_t'Q_t), \tag{14}$$

$$n_t = \beta n_{t-1} + 1, \quad r_t = (\beta n_{t-1} + e_t^2/Q_t)/n_t \tag{15}$$

$$s_t = r_t s_{t-1} \tag{16}$$

$$\text{where} \quad e_t = Y_t - f_t \quad \& \quad \boldsymbol{A}_t = \boldsymbol{R}_t\boldsymbol{F}_t/Q_t \tag{17}$$

(10),(12) & (13) provide the primary output of interest, the forecast distribution for the dependent variable and the posterior distributions for the relevant parameters.

The initialization of the four parameters defining the two processes of interest, are generated outside the model and passed through as initial priors: $\boldsymbol{m}_0, \boldsymbol{C}_0, n_0, s_0$. Together with the equations systems above, these initial priors form a complete specification of a DLM.

5.2 Dynamic Generalized Linear Models

Given that a DGLM is a generalized extension of a DLM, the framework of a DGLM will be briefly explained ((West et al., 1985), Chapter 14). A DGLM consists of an observation model (18) with a pdf belonging to the exponential family, defined by the natural parameter η_t and a known scale factor ϕ.

$$p(y_t|\eta_t, \phi) = b(y_t, \phi)exp[\phi\{y_t\eta_t - a(\eta_t)\}] \tag{18}$$

The observation model is consequently linked to a linear predictor, λ_t, via a link function $g(\eta_t) = \lambda_t$, in the predictor the regressors and state vectors take place: $\lambda_t = \boldsymbol{F}_t'\boldsymbol{\theta}_t$. In the three distributions used in this specification, Bernoulli, Poisson and Normal, the link function is the identity function and as such $\eta_t = \lambda_t$. The known scale factor, ϕ is set to 1 in Bernoulli and Poisson models whereas in DLMs (Normal) it becomes time varying and is estimated via a Beta-Gamma stochastic volatility model as explained above. To observe how this generalized notation leads to the observation equation of a DLM (1), input $\eta_t = \lambda_t = \mu_t$, $\phi_t = 1/v_t$, $a(\eta_t) = \eta_t^2/2$ and $b(y_t, \phi_t) = (\phi_t/2\pi_t)^{1/2}\exp(-\phi_t y_t^2/2)$. For further details regarding the parameters and functions of (18) for the other distributions, please refer to Appendix A.4. Given λ_t, a state space form defines the dynamic regression:

$$\lambda_t = \boldsymbol{F}_t'\boldsymbol{\theta}_t \quad \text{where} \quad \boldsymbol{\theta}_t = \boldsymbol{G}_t\boldsymbol{\theta}_{t-1} + \boldsymbol{\omega_t} \quad \boldsymbol{\omega_t} \sim (\boldsymbol{0}, \boldsymbol{W_t}) \tag{19}$$

Where the included parameters are equally defined as above. Aside from the lack of an observational error, (19) mirrors the observation and system equations of the DLM. An extension presented in (Berry & West, 2019) attempts to mimic the effect of the observational error for data which displays over-dispersion relative to a Poisson model, a common issue. By redefining state and regression vectors as follows, $\boldsymbol{F}_t^* = (1, \boldsymbol{F}_t')'$ and $\boldsymbol{\theta}_t^* = (\xi_t, \boldsymbol{\theta}_t')'$, a new model is generated: $\log(\mu_t) = \boldsymbol{F}_t'^*\boldsymbol{\theta}_t^* = \boldsymbol{F}_t'\boldsymbol{\theta}_t + \xi_t$. The new random-effects parameter, ξ_t, is accompanied by it's own discount factor, ρ. This extension allows for time t-specific variation above the baseline level thereby accounting for the over-dispersion. The conditional variance of ξ_t is defined as $V[\xi_t|\mathcal{D}_{t-1}] = Q_t(1-\rho)/\rho$ where $Q_t = \boldsymbol{F}_t'\boldsymbol{R}_t\boldsymbol{F}_t$. Given this setting, the level of ρ determines the impact of the random effect, where $\rho = 1$ results in a standard Poisson DGLM without the extension. For further details please refer to ((Berry & West, 2019), Section 2.3).

5.2.1 Updating & Forecasting

Unlike DLMs the prior and posterior of the state vector is not of a known distribution for DGLMs. The standard Kalman-filter based updating outlined above is therefore infeasible and DGLMs therefore rely on Variational Bayes and Linear Bayes conceptuality to achieve the same. The process is as follows:

1. At any given time, t-1, the mean vector and variance matrix of the posterior of the state vector encapsulate the current information set: $(\boldsymbol{\theta}_{t-1}|\mathcal{D}_{t-1}) \sim (\boldsymbol{m}_{t-1}, \boldsymbol{C}_{t-1})$
2. Standard evolution ensues resulting in the 1-step ahead prior moments of the state vector: $(\boldsymbol{\theta}_t|\mathcal{D}_{t-1}) \sim (\boldsymbol{a}_t, \boldsymbol{R}_t)$ where $\boldsymbol{a}_t, \boldsymbol{R}_t$ are defined identically as for DLMs.

3. Here the Variational Bayes concept determines that a conjugate prior for η_t, $(\eta_t|\mathcal{D}_{t-1}) \sim \text{CP}(\alpha_t, \beta_t)$, must be chosen of the form: $p(\eta_t|\mathcal{D}_{t-1}) = c(\alpha_t, \beta_t)\exp(\alpha_t\eta_t - \beta_t a(\eta_t))$ where $c(.,.)$ is a known function of the hyper-parameters (α_t, β_t) specific to each exponential family model.

4. These hyper-parameters are consequently evaluated as to ensure the conjugate prior meets the prior moment constraints: $\text{E}[\lambda_t|\mathcal{D}_{t-1}] = f_t = \boldsymbol{F}_t'\boldsymbol{a}_t \quad \& \quad \text{V}[\lambda_t|\mathcal{D}_{t-1}] = q_t = \boldsymbol{F}_t'\boldsymbol{R}_t\boldsymbol{F}_t$

5. Through employment of the conjugacy-induced predictive distribution $p(y_t|\mathcal{D}_{t-1}) = b(y_t, \phi)c(\alpha_t, \beta_t)/c(\alpha_t + \phi y_t, \beta_t + \phi)$ a forecast for y_t is generated.

6. Once the true y_t is observed, the posterior for η_t enjoys the conjugate form: $(\eta_t|\mathcal{D}_t) \sim \text{CP}(\alpha_t + \phi y_t, \beta_t + \phi)$

7. Given this posterior, mapping backwards to $\lambda_t = \eta_t$ implies posterior mean and variance: $\text{E}[\lambda_t|\mathcal{D}_t] = g_t \quad \& \quad \text{V}[\lambda_t|\mathcal{D}_t] = p_t$

8. Ultimately, the state vector's posterior mean vector and variance matrix are determined via Linear Bayes updating: $(\boldsymbol{\theta}_t|\mathcal{D}_t) \sim (\boldsymbol{m}_t, \boldsymbol{C}_t)$ where: $\boldsymbol{m}_t = \boldsymbol{a}_t + \boldsymbol{R}_t\boldsymbol{F}_t(g_t - f_t)/q_t \quad \& \quad \boldsymbol{C}_t = \boldsymbol{R}_t - \boldsymbol{R}_t\boldsymbol{F}_t\boldsymbol{F}_t'\boldsymbol{R}_t'(1 - p_t/q_t)/q_t$

5.3 Dynamic Count Mixture Models

A combination of two DGLMs, one binary and one conditionally Poisson, generates a dynamic count mixture model (DCMM). The idea behind the DCMM is that the first component, the binary DGLM, models the probability of a sale occurring and consequently the second component, the conditionally Poisson DGLM, models the magnitude of the sale given that a sale occurs. It is often essential to filter out the zero outcomes as failing to do leads to significant underestimation of the true level of sales. To form a DCMM for a non-negative count time series, y_t defines a parallel binary series $z_t = \mathbb{1}(y_t > 0)$, the corresponding DGLM is characterized by:

$$z_t \sim \text{Ber}(\lambda_t^*) \quad \text{where} \quad \text{logit}(\lambda_t^*) = \boldsymbol{F}_t'^*\boldsymbol{\theta}_t^* \tag{20}$$

The ensuing conditional Poisson DGLM has the following observation model:

$$y_t|z_t = \begin{cases} 0 & \text{if} \quad z_t = 0 \\ 1 + x_t, & x_t \sim \text{Po}(\lambda_t^+) \quad \text{if} \quad z_t = 1 \end{cases} \quad \text{where} \quad \log(\lambda_t^+) = \boldsymbol{F}_t'^+\boldsymbol{\theta}_t^+ \tag{21}$$

Where the differing notation between $\boldsymbol{F}_t'^*\boldsymbol{\theta}_t^*$ and $\boldsymbol{F}_t'^+\boldsymbol{\theta}_t^+$ is used to highlight that both the regressors and latent state factors can differ between the two DGLMs. Certain factors or regressors may be important predictors for determining whether or not a sale may occur (Bernoulli DGLM) but may be fairly irrelevant for determining the size of the purchase (Poisson DGLM), therefore these vectors are allowed to differ.

5.3.1 Forecasting

Given that a DCMM is simply a composition of two DGLMs, it too provides a full predictive distribution for any future point in time. This distribution is mixture of the two independent DGLMs, a shifted Poisson and a Bernoulli. To obtain forecasts for n-days ahead, the forward evolution of the state vectors of both respective DGLMs explained above provides analytic amendability. The form of the probability density function, at time t, for y_{t+n} is as follows:

$$p(y_{t+n}|\mathcal{D}_t, \pi_{t+n}) = (1 - \pi_{t+n})\delta_0(y_{t+n}) + \pi_{t+n}h_{t,t+n}(y_{t+n}) \tag{22}$$

with $(\pi_{t+n}|\mathcal{D}_t) \sim \text{Beta}(\alpha_t^0(n), \beta_t^0(n))$ which represents the forecast probability of a zero-valued observation and consequently $\delta_0(y)$, the Dirac Delta function at zero, imposes this zero-value. Furthermore $h_{t,t+n}(y_{t+n})$ is the forecast density of $y_{t+n} = 1 + x_{t+n}$ where x_{t+n} is the product of the shifted Poisson DGLM and thus has a Negative-Binomial forecast distribution:

$$(x_{t+n}|\mathcal{D}_t) \sim \text{NB}\left(\alpha_t^1(n), \frac{\beta_0^1(n)}{1 + \beta_t^1(n)}\right) \tag{23}$$

For clarity, the 0-superscripted α, β refer to the conjugate parameters of the Bernoulli process whereas the 1-superscripted α, β serve the same purpose for the Poisson process. The aforementioned process describes a marginal forecast distribution for y_{t+n}, what is often of greater practical relevance is a path forecast for $y_{t+1}, y_{t+2}, .., y_{t+n}$. The framework above allows for trivial computation of these path forecasts as follows.

1. Generate the 1-day ahead forecast distribution as laid out above.
2. Simulate an observation y_{t+1}^* from the forecast distribution.
3. Regard this simulated outcome as true and update the respective DGLMs state vectors from prior to posterior accordingly.
4. Progress to day t+2 and repeat the process.
5. Repeat for days t+3,..,t+n,

One run-through of the steps above generates a single path $y_{t+1}^*, y_{t+2}^*, .., y_{t+n}^*$ Monte Carlo sample. Sufficient repetition of the process generates a full Monte Carlo representation of the joint predictive distribution.

For certain product/county model specifications it became apparent that the shifted Poisson DGLM component was inappropriate. After filtering out the zero-valued observations what remained deviated too far from a Poisson distribution, even with correction from the random effects extension. For these cases the non-zero valued observations were log-transformed resulting in an approximately normal distribution, the Poisson component was therefore replaced by a DLM. The derivations above remain applicable, with only the Negative-Binomial forecast distribution replaced by the T-distributed forecast distribution seen in (10).

5.4 Latent Factor Models

The latent factors are the building block of the multi-scale model approach defined above, allowing the models to transfer information to one another from different levels. To illustrate the working, consider the conditionally Poisson model $\mathcal{M}_1$ for a single time series, i, pertaining to the orders of a certain product by one store. The relevant vectors are defined as:

$$\boldsymbol{\theta}_{i,t} = \begin{pmatrix} \boldsymbol{\gamma}_{i,t} \\ \boldsymbol{\alpha}_{i,t} \\ \boldsymbol{\beta}_{i,t} \\ \boldsymbol{\sigma}_{i,t} \\ \boldsymbol{\kappa}_{i,t} \end{pmatrix} \quad \boldsymbol{F}'^{+}_{i,t} = \begin{pmatrix} \boldsymbol{f}_{i,t} \\ \boldsymbol{\phi}_t \\ \boldsymbol{\delta}_t \\ \boldsymbol{\zeta}_t \\ \boldsymbol{\chi}_t \end{pmatrix} \tag{24}$$

where $\boldsymbol{f}_{i,t}$ contains series specific constants and the following three latent factor vectors $\boldsymbol{\phi}_t, \boldsymbol{\delta}_t, \boldsymbol{\zeta}_t$ represent the effects of temperature, holidays, seasonality and COVID-19-related variables which are estimated, in parallel, on the other level of the model $\mathcal{M}_0$. The latent factors thereby transfer from being elements of $\boldsymbol{\theta}_t$ in $\mathcal{M}_0$, to being elements of $\boldsymbol{F}_t$ in $\mathcal{M}_1$. Given the vectors above, the linear predictor is thus:

$$\lambda_{i,t} = \boldsymbol{\gamma}'_{i,t}\boldsymbol{f}_{i,t} + \boldsymbol{\alpha}'_{i,t}\boldsymbol{\phi}_t + \boldsymbol{\beta}'_{i,t}\boldsymbol{\delta}_t + \boldsymbol{\sigma}'_{i,t}\boldsymbol{\zeta}_t + \boldsymbol{\kappa}'_{i,t}\boldsymbol{\chi}_t \tag{25}$$

Note that each series has different state components $\boldsymbol{\gamma}_{i,t}, \boldsymbol{\alpha}_{i,t}, \boldsymbol{\beta}_{i,t}, \boldsymbol{\sigma}_{i,t}, \boldsymbol{\kappa}_{i,t}$ allowing the common factors to impact the individual series differently in addition to being time-varying.

Identically to regular DLMs and DGLMs, the underlying distributional distinctions between latent factor DLMs and latent factor DGLMs cause the parameter estimation to differ. The two estimation approaches are therefore separately presented below.

5.4.1 Updating & Forecasting Latent Factor DLM

When forecasting a latent factor DLM, two distinct groups of parameters have to be estimated and projected into the future. The first are the DLM-related dynamic parameters $\Phi_t = (\boldsymbol{\theta_t}, v_t)$, the second are the latent factors forecasts generated by M_0, denoted by Λ_t. Given that the higher-level M_0 is, in each specification, a DLM in which the latent factors are elements of $\boldsymbol{\theta}_t$, the resulting forecast distribution of Φ_t is thereby T-distributed as laid out in (12). In this context the added benefit of a DLM as opposed to a DGLM, is the known distributions of the DLM parameters as this allows for Markov chain Monte Carlo (MCMC) estimation via a two component Gibbs sampler.

The procedure of the sampler for obtaining posterior results is as follows, MCMC draws from $p(\Phi_t|\Lambda_t, \mathcal{D}_{t-1})$ are taken after conditioning on initial latent agent states Λ_t. The sampling is performed using the well-known forward filtering backward sampling algorithm (FFBS) (Frühwirth-Schnatter, 1994). Subsequently, MCMC draws are taken from $p(\Lambda_t|\Phi_t, \mathcal{D}_{t-1})$ whereby the previously drawn Φ_t is conditioned upon. To achieve this, a standard approach from (Frühwirth-Schnatter, 1994; West & Harrison, 1997) is used in which the forecast T-distributions in the proportionality below are rewritten as a scale mixture of Normal distributions, simplifying the right-hand side to an entirely Gaussian

expression from which draws can be taken.

$$p(\Lambda_t|\Phi_t, \mathcal{D}_{t-1}) \propto \mathcal{N}(Y_t|\boldsymbol{F}_t'\boldsymbol{\theta}_t, v_t)\, T(\Lambda_t|\boldsymbol{m}_t, \boldsymbol{C}_t) \tag{26}$$

For a detailed explanation of the sampler and the exact steps please refer to Appendix A.2 of (McAlinn & West, 2019), where the Gibbs sampler which served as inspiration for the sampler employed in this case study is presented. The code corresponding to the adjusted Gibbs sampler used in this thesis is presented in Appendix B.1. Once posterior MCMC draws have been obtained forecasting is straight-forward. For a 1-step ahead forecast, first generate a draw of $\Phi_{t+1} = (\boldsymbol{\theta_{t+1}}, v_{t+1})$ by drawing v_{t+1} according to (3) for each sampled Φ_t. Given $\boldsymbol{\theta}_t$ and v_{t+1}, $\boldsymbol{\theta}_{t+1}$ follows from (2) after which Λ_{t+1} can be independently drawn from the forecast T-distribution. Together these forecasts provide the necessary components to draw a forecast Y_{t+1} from (1), this process can easily be adjusted to allow for more distant forecasts. To optimally customize multi-step ahead forecasting to a specific k-step ahead horizon, (McAlinn & West, 2019) suggest replacing the 1-step ahead forecast distribution $T(\Lambda_t|\boldsymbol{m}_t, \boldsymbol{C}_t)$ in (26) with the k-step ahead forecast distribution $T(\Lambda_t|\boldsymbol{m}_{t-k}, \boldsymbol{C}_{t-k})$. The reason for doing so is that "this changes the interpretation of the dynamic model parameters Φ_t to be explicitly geared to the k-step horizon. Bayesian model fitting then naturally "tunes" the model to the horizon k of interest."

5.4.2 Updating & Forecasting Latent Factor DGLM

To illustrate how the updating and forecasting works in a DGLM/DCMM which takes an externally generated latent factor as input, the following example is included. Consider two models, $\mathcal{M}_0$ is the higher-level model which generates the latent factor, and $\mathcal{M}_1$ is the model which includes the latent factor ϕ as a regressor. If we wish to forecast $y_{1,t:t+10}$, that is a path forecast of the dependant variable of $\mathcal{M}_1$ 10 days into the future, the following process is run:

1. $\mathcal{M}_0$ independently generates N samples of $\phi_{t:t+10}$, the trajectory of the latent factor till $t + 10$.
2. $\mathcal{M}_1$ consequently runs N analyses, each one conditional on a sample $\phi^n_{t:t+10}$, to forecast $y_{1,t:t+10}$. These trajectories $y^n_{1,t:t+10}$ create a N-sized Monte Carlo sample of the implied predictive distribution thereby accounting for the uncertainties of the latent factor process from $\mathcal{M}_0$.

Once the true value of $y_{1,t}$ has been observed, the state vectors can be updated as follows:

1. Via conditionally conjugate analysis for DGLMs compute the value of 1-day ahead forecast probability density function: $p(y_{1,t:t+10}|\phi^n_{t:t+10}, \mathcal{D}_{t-1})$, for each n. Generate implied posterior probabilities for the n=1:N latent factor samples by using the calculated values as marginal likelihoods and comparing them to uniform (1/N) prior probabilities.
2. Consequently, for each ϕ^n_t, perform DGLM updating to compute the posterior mean vector and variance matrix of $\boldsymbol{\theta}_{1,t}|\phi^n_t, y_{1,t}, \mathcal{D}_{t-1}$. Use the probabilities calculated before as weights to marginalize over the N samples to generate implied Monte Carlo approximations to the posterior mean vector and variance matrix.

5.5 Priors

Before the updating of the parameters can commence, an initial prior must be passed through to the model as a starting point. A subset of the data set was set aside to do so, with the temporal aggregation level determining the size of the subset. For data sets with daily observations the first 60 observations were used, whereas for weekly data the first 20 observations were deemed sufficient. These initial priors, which consisted of the parameters $\boldsymbol{m_0}, \boldsymbol{C}_0$ were generated by running standard OLS and GLS, for DLMs and DGLMs respectively, on the subset using Zellner's objective g-prior (Zellner, 1986), where g was set at 30 for daily-valued data sets and 10 for weekly. The additional parameters n_0, s_0 for DLMs were set as the prior sample size and 1 respectively.

5.6 Hyper-Parameter and Model Selection

To optimize the forecasting performance it was crucial to properly tune the model hyper-parameters, the discount factors, and find the right model specification. To compare various model specifications, with differing variable sets, the forecasting performance of the specifications was compared. Over a period of two weeks, from the test set (October-December 2020), the likelihood of the true observed y under the forecast distribution was calculated. The product of all these likelihoods, $\prod_{h=1:t} p(Y_h|\mathcal{M}, \mathcal{D}_{h-1})$, was compared between model specifications to determine which performed best in terms of forecasting power.

The tuning of the discount factors was equally important, by their nature there is a trade-off to lowering them, whilst the coefficients are allowed to vary more giving the model greater flexibility, it also increases the uncertainty in forecasts and makes the coefficients increasingly vulnerable to noise. A discount factor between [0.9,1] is reasonable as below this range the variance becomes excessively large ((West & Harrison, 1997), Section 6.3). To find the optimal discount levels for the the stochastic volatility, random effects parameter and all the regression components (trend, seasonal, regressors and latent factors) a grid search was performed, testing the forecasting performance at the following levels :[0.9,0.92,0.94,0.95,0.96,0.98,0.985,0.99,0.995,0.999].

5.7 Implementation

The model was implemented using the open-source PyBATS python package (Lavine, 2020), which was developed by Isaac Levine in collaboration with Mike West. Designed to enable both quick analyses as well as flexible customization options for the model form, prior and forecast period. It is capable of running both DCMM models (Berry & West, 2019) and DBCM models (Berry et al., 2020), the only open-source python package at the moment which does so. The package did however lack certain functionalities which were necessary to run the model design laid out above, specifically; the two-step Gibbs sampler for latent factor DLMs as well as sample-based updating and path-forecasting for latent factor DLMs. The option to return the *log*-probability density of observations under the estimated forecast distribution was also lacking. These functionalities were therefore coded outside the package, in consultation with the package developer, and will be appended to the package during the upcoming patch. For completeness, the code pertaining to the aforementioned functionalities has been included in Appendix B.

6 Application & Results

The following section presents and analyzes the obtained results. Given that the multi-layer structure, latent factor working and interpretation of results were the same for both model specifications, it was chosen to focus on one specification to demonstrate the working of the model and particularly the latent factors. The specification chosen was the first of Table 3, the DLMs designed to forecast state-wide sales, the reason for doing so was that this specification could serve a dual purpose. Aside from showcasing the model working it was also deemed the most appropriate to evaluate the effects of the COVID-19 pandemic, through the COVID-statistics, restriction levels and mobility reports. The second specification was less suitable given that it involved weekly observations and thereby aggregated out vast amounts of information, furthermore it revolved around the sales of a single store and therefore store-specific circumstances, e.g. promotions on which we have no data, could influence the outcome. Aside from the detailed analysis of the first specification, the forecasting performance of both specifications are presented and compared to a standard non-multi-scale alternative.

6.1 State-Level Wholesales Dynamic Linear Model

6.1.1 Application

The first point of action was applying a log-transformation to the dependent variables of $\mathcal{M}_0$ and $\mathcal{M}_1$, the aggregated state-wide sales of all products and the aggregated county-wide sales for a particular category, respectively. This was done to to reduce the scale of Y_t and generate multiplicative effects for the regressors, increasing the interpretability.

The following step towards performing any analysis was model selection, in which consideration had to be taken for the wholesale nature of the data. This implied that there was a certain lag between the regressors and the dependent variable at each stage, the logic being that the regressors were primarily predictors of in-store liquor purchases by consumers. Changes in these purchases would subsequently be translated through to the wholesale data, where periods of increased in-store sales would result in greater or more frequent wholesale orders. To determine the optimal length of this lag, the forecasting performance of model specifications with varying lag length were compared as explained in Subsection 5.6. The outcome of these comparisons were however inconclusive and further inspection of store-specific sets revealed the reason. Whilst the model set-up did account for the lag, by aggregating all the stores in a certain county ($\mathcal{M}_1$) or state ($\mathcal{M}_0$) it incorrectly assumed that the lag was equal for each store. Ensuing isolated analysis of stores of varying sizes found that larger supermarkets had lags between 7 to 10 days whereas small-scale liquor stores had lags between 1-3 days. To therefore perform meaningful analysis the stores had to be split by size to ensure an appropriate lag could be imposed. The results presented below pertain to the set of large supermarkets and the optimal lag for this set was found to be 9 days.

The ensuing step was to determine the optimal variable set for each model layer, $\mathcal{M}_0$ and $\mathcal{M}_1$. Initially the set-up was chosen as described in the first row of Table 3, however after subsequent analysis it became apparent that two alterations had to be made. Firstly, given that the climatic data was available on a county level, it was initially placed in $\mathcal{M}_1$ to thereby allow for different temperature

effects between counties. However after comparing variations of the model set-up, a specification with climatic data on the state level proved to generate better forecasts, in which the magnitude of the coefficient was, on average over the test set, smaller than the individual county-level coefficients but also had a far smaller variance. This provided forecasts of greater accuracy whilst also lowering the forecast variance, the climatic data was therefore moved up from $\mathcal{M}_1$ to $\mathcal{M}_0$. The second alteration was in regard to the restriction variable, which was initially included as a ordinal categorical variable with 4 levels based on an assumption of linear effects. This assumption was proven to be misplaced, with particularly the highest level of restrictions having proportionally a much greater effect, the last three levels were therefore included as dummy variables as opposed to a single categorical variable. Two variables, (*Precipitation, log(7-day avg.Deaths)*), were dropped from the regressor vector as they were found to be of little contribution, with coefficients quickly converging to 0 and remaining there, whilst their inclusion did increase the overall variance of the model forecasts. Furthermore, instead of including both *Retail* and *Grocery* only one of the two was always included, for the larger supermarkets this was *Grocery* whilst for the smaller liquor stores this was *Retail* as these were officially classified as retail stores as opposed to grocery or convenience stores. The reason for doing so was that whilst the second variable ,e.g. *Retail* for supermarket specifications, did pick up a small substitution effect with a opposite sign to the first variable, this contributed little to the forecasting accuracy and did increase the forecast uncertainty.

Ultimately this led to the following regression vector $\boldsymbol{F}_t$ and state evolution matrix $\boldsymbol{G}_t$ for $\mathcal{M}_0$ and $\mathcal{M}_1$ (27-28), where the harmonic component matrices $\boldsymbol{H}_1, \boldsymbol{H}_2$ are defined as in Subsection 5.1 and the variables $\text{Feb}_t .., \text{Dec}_t$ are indicators included to capture the monthly seasonality compared to the omitted baseline January.

$$\mathcal{M}_0 : \boldsymbol{F}_{t,0} = (1, \text{Temp}_t, \text{Holiday}_t, 1, 0, 1, 0, \text{Feb}_t .., \text{Dec}_t), \quad \boldsymbol{G}_{t,0} = \text{diag}[1, 1, 1, \boldsymbol{H}_1, \boldsymbol{H}_2, 1, 1.., 1] \tag{27}$$

$$\mathcal{M}_1 : \boldsymbol{F}_{t,1} = (1, \lambda_t, \text{logCases}_t, \text{Grocery}_t, \text{Restr.1}_t, \text{Restr.2}_t, \text{Restr.3}_t), \quad \boldsymbol{G}_{t,1} = \boldsymbol{I}_{12x12} \tag{28}$$

where $\lambda_t = (Y^*_{\mathcal{M}_0,t}, \text{Temp}^*_{\mathcal{M}_0,t}, \text{Holiday}^*_{\mathcal{M}_0,t}, \text{Daily Seas.}^*_{\mathcal{M}_0,t}, \text{Monthly Seas.}^*_{\mathcal{M}_0,t})$ represents all the latent factors generated in $\mathcal{M}_0$ that are passed down to $\mathcal{M}_1$, in which $Y^*_{\mathcal{M}_0,t}$ is the forecast for the dependent variable of $\mathcal{M}_0$. The construction of $\mathcal{M}_0$ laid out above generated 11 latent factors for the monthly seasonality, one for each month included, which was filled with zeros for all time-points outside the month corresponding to the latent factor. To save space and computational time, these were all combined into one latent factor. To illustrate how the seasonality latent factors works, consider the following example of the 4-day ahead forecast of the daily seasonality latent factor:

$$\begin{aligned}
\text{Daily Seas.}^*_{\mathcal{M}_0,t+1} &= [0.05099, 0, 0, 0, 0] \\
\text{Daily Seas.}^*_{\mathcal{M}_0,t+2} &= [0, 0.09767, 0, 0, 0] \\
\text{Daily Seas.}^*_{\mathcal{M}_0,t+3} &= [0, 0, 0.01507, 0, 0] \\
\text{Daily Seas.}^*_{\mathcal{M}_0,t+4} &= [0, 0, 0, -0.19674, 0]
\end{aligned}$$

Where at time-point t the latent factor forecast for $t+1$, a Monday, estimates a multiplication factor of

$e^{0.05099} = 1.0523$ or a 5% increase, whereas the upcoming Thursday is expected to see a 18% decrease ($e^{-0.19674} = 0.8214$) from the baseline level. The final preliminary settings which needed to be tested were the levels of the discount factors, β and δ, for the observational variance and the regressors respectively. A grid search revealed that the forecasting performance was best at [β,δ] = [0.98,0.99], it was also chosen to only discount regressor coefficients at time-points with non-zero regressor values. As otherwise coefficients belonging to regressors with few non-zero values, e.g. Restriction level 2 was only in place for six weeks and therefore had zero values for all other dates, would be pushed to 0 by the consistent discounting.

6.1.2 Results

After having tuned the model to the preferred settings, analysis could be performed. The 1-step ahead forecasts of $\mathcal{M}_0$ generated from there are presented below in Figure 6, where the point forecast line (blue) is the median of the samples produced. The forecasting accuracy proved to be significantly better than a OLS regression with the same regressors, yielding a Mean Absolute Percentage Error (MAPE) of 10.21% for 1-step ahead forecasts which rose to 12.92% for 14-step ahead forecasts compared to 24.12% and 33.82% for the OLS model with the same forecast horizons. The forecasting performance

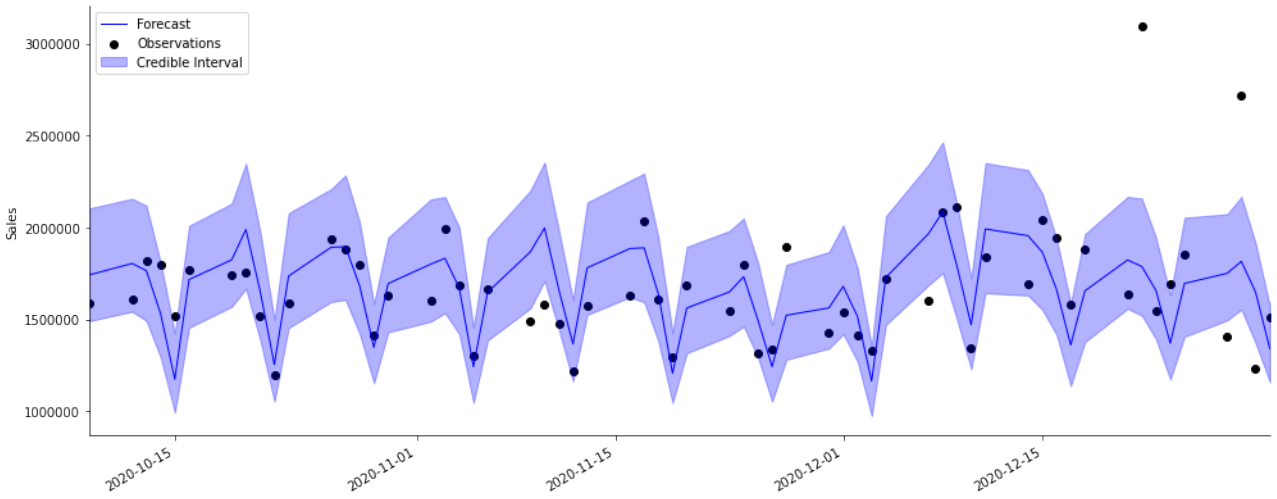


Figure 6: 1-step ahead forecast for $Y_{t,\mathcal{M}_0}$, the aggregated state-wide sales of all liquor products, the shaded blue area represents the 80% credible interval

of $\mathcal{M}_0$ is however not the primary output of interest, but rather the latent-factors generated to aid the forecasting of $\mathcal{M}_1$, which are, as explained in Subsection 5.4, the coefficients of the relevant regressors in $\mathcal{M}_0$. A plot of both seasonality latent factor means is displayed below in Figure 7, the left-hand side pertains to the daily seasonality whereas the right hand side to the monthly. From the daily seasonality plot it becomes apparent that there strong level differences between the various weekdays, the model estimates that in December 2020 the weekday multiplication factor of a Friday leads to, on average, 19% lower sales than a Thursday. Both graphs showcase how the model learns from new information as it comes in, particularly the right-hand side, in which a clear yearly pattern can be seen that is repeated 4 times. As the model learns, the peaks and troughs of the cycle are

deflated resulting in a stable monthly seasonality pattern for 2020, characterized by a dip during the summer months which is followed by a great peak around the end of the year.

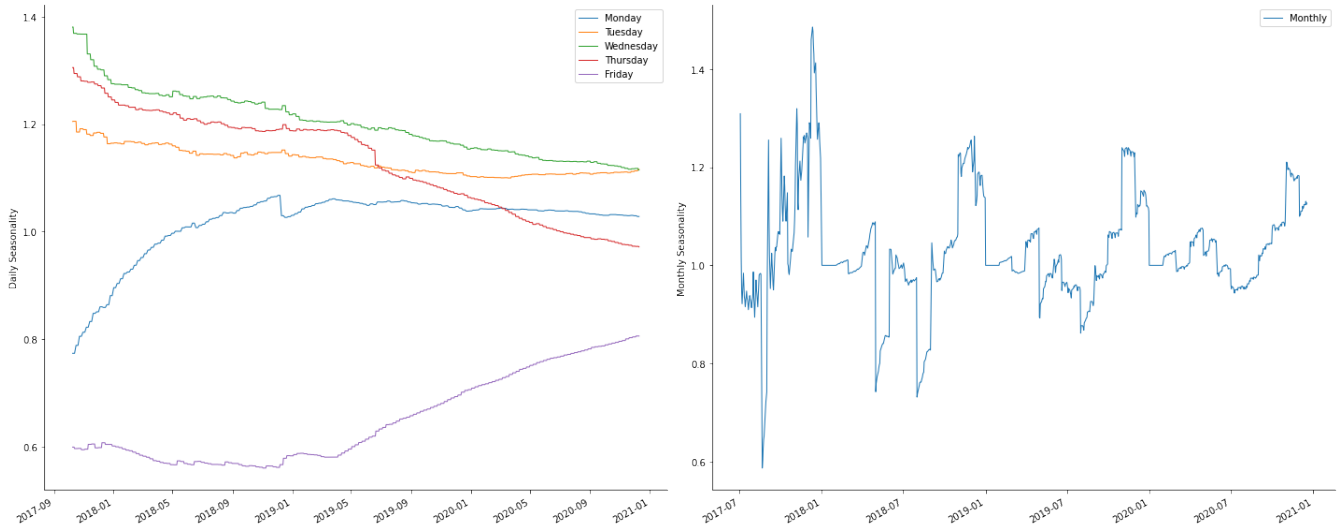


Figure 7: Coefficient plots of the daily (left) and monthly (right) seasonality components

The other two latent factors generated, for the temperature (left) and holiday (right) effect, are presented in Figure 8, where the holiday latent factor also appears to undergo large adaptations over time with the holiday effect in 2017 initially estimated as being more than twice the magnitude than the effect in 2020. The temperature latent-factor, on the left-hand side, appears to accurately capture the correlation originally observed in Figure 1, with the coefficients steadily rising during the first half of each year to consequently fall in the second half.

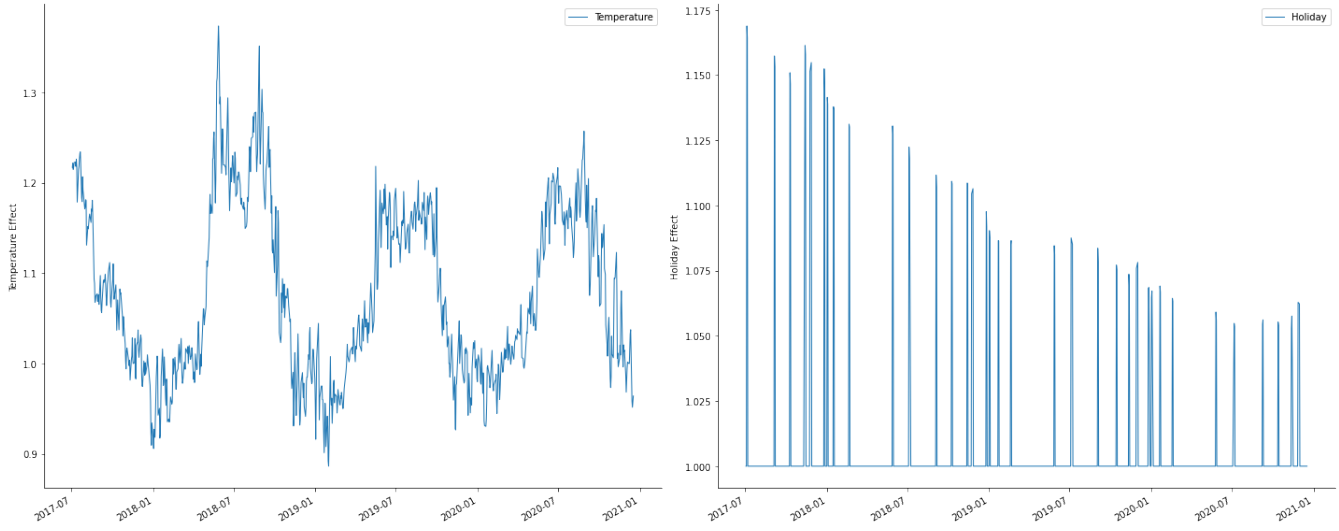


Figure 8: Coefficient plots of the temperature (left) and holiday (right) effect

Including these latent factors in $\mathcal{M}_1$ as shown in (28) and running the two-component Gibbs sampler for 8000 samples (of which 3000 burn-in samples), ultimately led to the 10-step ahead forecasts

presented in Figure 9, where $Y_{t,\mathcal{M}_1}$ denotes the aggregated sales of Whiskies in Polk, the most populous county of Iowa.

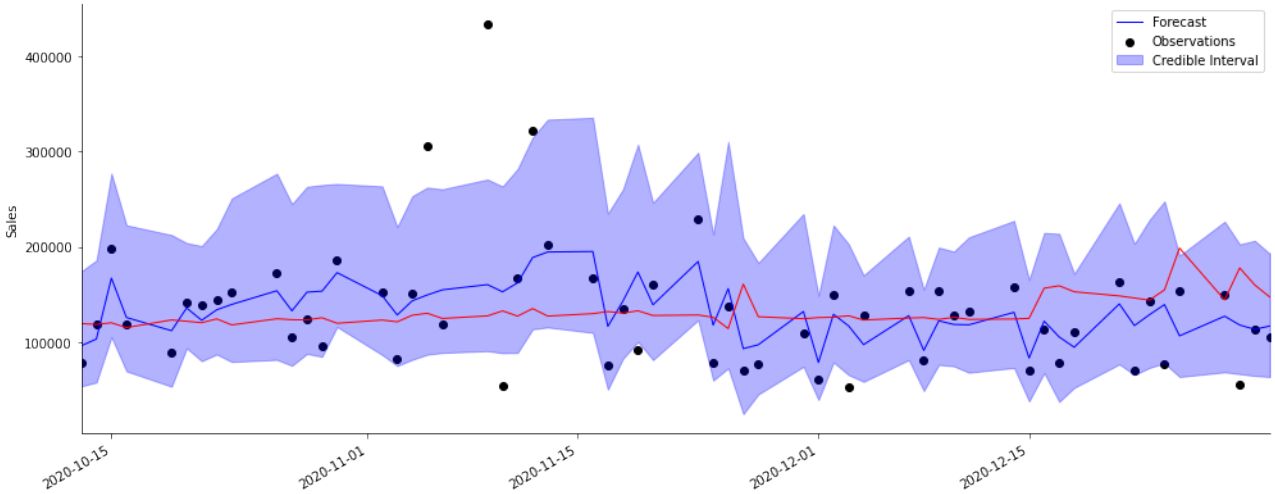


Figure 9: 10-step ahead forecast for $Y_{t,\mathcal{M}_1}$, the aggregated sales for Whiskies in Polk county, where the median of the samples is depicted by the blue line, the shaded blue area represents the 80% credible interval and the red line shows an OLS estimate

The plot appears to better capture the changes in the time-series than an OLS model (red line) with the same regressor set, which is confirmed by Table 4, which displays a comparison of the MAPE for 1 to 14-step ahead forecasts. With the latent factor DLM achieving MAPE scores which are nearly half the size of the OLS model throughout the forecast horizon.

	Forecast Mean Absolute Percentage Error													
Model	1-step	2-step	3-step	4-step	5-step	6-step	7-step	8-step	9-step	10-step	11-step	12-step	13-step	14-step
M_1	19.141	19.719	21.178	20.115	20.580	21.772	20.916	22.884	23.114	23.656	22.018	24.529	24.113	24.916
OLS	35.156	35.892	36.214	39.153	38.944	38.792	39.991	41.256	40.901	41.113	42.361	42.178	44.052	44.452

Table 4: Comparison of the Mean Absolute Percentage Error for 1-14 day ahead forecasts between the latent factor DLM, $\mathcal{M}_1$, and an OLS regression for six most-sold product categories

As previously mentioned, one of the merits of an Bayesian estimation approach is the overview of the uncertainty of estimates which are provided, as opposed to a single prediction for each time-point. This uncertainty is depicted in Figure 9 by the 80% credible interval of the 5000 samples and an example for a single time-point, the 26$^{\text{th}}$ of November, is presented below in Figure 10. The managerial insights provided by these graphs are self-evident, with decision-makers not only given a forecast but also an idea of how likely other forecasts are.

The working of the latent factors in a multi-scale approach are illustrated in Figure 11, which displays the coefficients of the latent factors from $\mathcal{M}_0$ in $\mathcal{M}_1$. These latent factors have been estimated at a higher, aggregated level to generate a clearer picture of the true effect, the lower-level models can however determine to which degree they agree with this effect, by altering the latent factor coefficients. In the example given here, the effects of temperature, holiday and $Y_{\mathcal{M}_0}$ have been estimated for the sales of all liquor categories in Iowa, these effects are however not entirely correct for the sales of

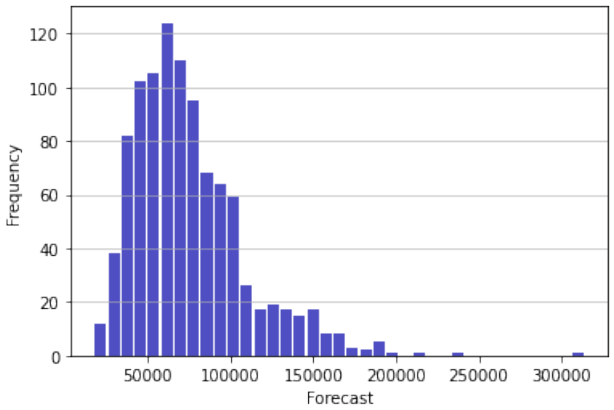


Figure 10: Histogram displaying the samples of the forecast for $Y_{t,\mathcal{M}_1}$ for the 26$^{\text{th}}$ of November

Whiskies in Polk county. Figure 11 shows that whilst the effect of temperature appears to have been estimated correctly, with the coefficient hovering around 1, the holiday effect has been heavily underestimated leading to a doubling of the coefficient. On the other hand, the effect of the level of total aggregated liquor sales in Iowa, $Y_{\mathcal{M}_0}$, appears to have been overestimated with it's coefficient sharply dropping. This beautifully illustrates the power of the multi-scale latent factor approach, namely that all the data is used to generate an estimate of a particular effect, whilst consequently allowing this effect to impact the underlying individual time-series in a bespoke manner.

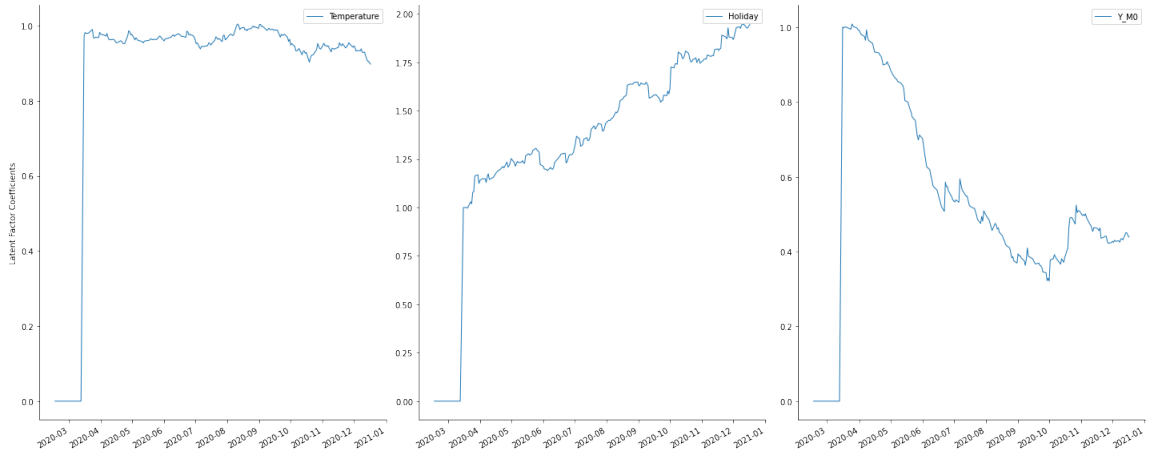


Figure 11: Latent factor mean coefficient plots for Temperature (left), Holidays (center) and $Y_{t,\mathcal{M}_0}$ (right) for $\mathcal{M}_1$

Of all the predictors included in the aforementioned specification of $\mathcal{M}_0$, three proved to be the most powerful. These were the Google mobility reports for grocery stores, the harshest level of restrictions (level 3) and the *log*-transformed number of COVID-19 cases, the plots of their coefficient evolutions are presented in Figure 12, by far the largest effect is contributed to the restrictions, which reaches -1 at certain times whereas the other coefficients peak at a fraction of that.

This raised the question of whether this impact was equally large for all liquor categories and furthermore what the impact was of the other restriction levels and COVID-related variables. To answer

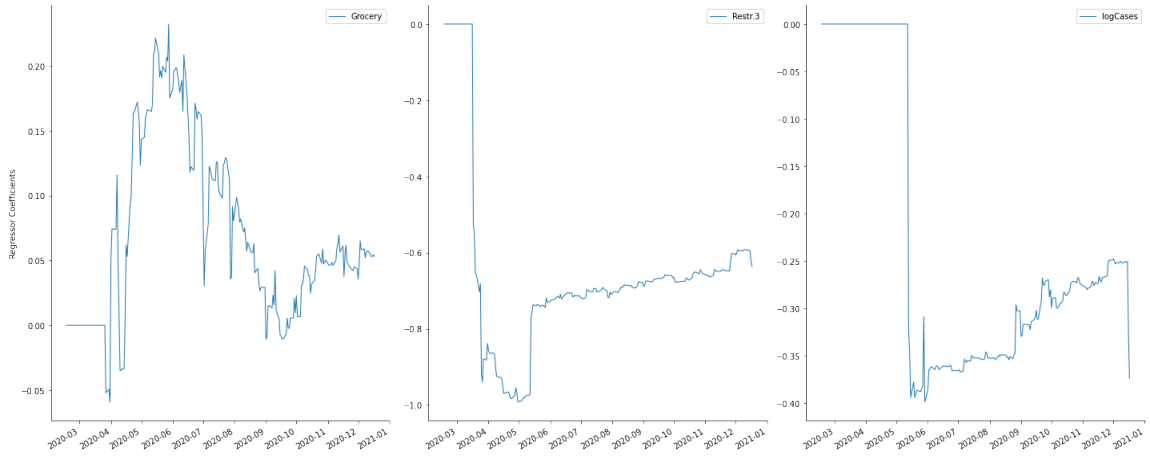


Figure 12: Regressor mean coefficient plots for Grocery (left), Restrictions level 3 (center) and logCases (right) for $\mathcal{M}_1$

these questions, similar analyses were performed for the other liquor categories over the six counties which saw heavy restrictions. An immediately evident result was that the coefficient for restriction level 2 was always small and varied greatly between the specifications, the reality was that there was most likely too little data to accurately forecast the effect. Restriction level 2 was namely first imposed directly after the ending of the harshest restriction level 3 in May, meaning these coefficients were often initially estimated to be extremely large, +4.2 in certain cases, but this was most likely due to the ending of restriction level 3 rather than the imposition of restriction level 2. After this there were too few data points with non-zero observations for restriction level 2 to accurately adjust the coefficient to it's true, unknown value. A comparison of the effects of restriction level 3 between the categories revealed a striking difference. Eight of the eleven categories exhibited a strongly negative effect for the imposition of these restrictions whereas three three showed slightly positive effects, these were Tequila, Spirits and Cocktails. When these restrictions were ultimately uplifted on the 1st of May, these categories showed coefficients of [0.12, 0.06, 0.18] respectively. Overall the Google Mobility reports and the third level of restrictions were shown to have the greatest forecasting contribution over all product categories, whilst the *log*-transformation of the number of COVID-19 cases was also a strong predictor for most categories.

An attempt to test whether consumers had shifted to higher-end, luxury liquors once the restaurants and bars had closed was unfortunately fruitless. Dividing the products into expensive and non-expensive proved to be challenging, with often the expensive products sets unable to generate accurate forecasts due to a lack of sufficient data. These products were logically sold significantly less and therefore provided far fewer observations, often not enough to generate accurate forecasts of all the regressors making a comparison difficult. Lowering the threshold of 'expensive' products to increase the sample size simply led to ambiguous results.

6.2 Forecasting Weekly Store-Product Orders

As previously mentioned, because the multi-layer structure and latent factor implementation work in a similar fashion for DCMMs as for DLMs, this subsection omits latent factor and coefficient plots to

avoid repetition and serves solely to demonstrate the models ability to forecast intermittent sales. To showcase the forecasting performance of the multi-layered model framework described on the second layer of Table 3, the intermittent order history of a store in Des Moines, Polk for a particular whiskey was chosen as the dependent variable in $\mathcal{M}_1$, where each order consisted of six bottles. Aside from the variables mentioned in Table 3, the retail price was also included as this product was one of a handful of products which saw several price changes throughout the data set, thereby presenting an opportunity to measure the price elasticity. Ultimately this led to the following regressor vectors $\boldsymbol{F}_{t,i}$ and state evolution matrices $\boldsymbol{G}_{t,i}$:

$$\mathcal{M}_0 : \boldsymbol{F}_{t,0} = (1, \text{Price}_t, \text{Temp}_t, \text{Holiday}_t, 1, 0, 1, 0, \text{Feb}_t .., \text{Dec}_t, \text{Grocery}_t, \text{Res.1}_t, \text{Res.2}_t, \text{Res.3}_t)$$
$$\boldsymbol{G}_{t,0} = \text{diag}[1, 1, 1, 1, \boldsymbol{H}_1, \boldsymbol{H}_2, 1, 1.., 1, 1, 1, 1, 1]$$
$$\mathcal{M}_1 : \boldsymbol{F}_{t,1} = (1, \lambda_t,), \quad \boldsymbol{G}_{t,1} = \boldsymbol{I}_{10x10}$$

where λ_t once more represents all the latent factors generated in $\mathcal{M}_0$, including $Y^*_{\mathcal{M}_0,t}$, and where $\boldsymbol{H}_1, \boldsymbol{H}_2$ are equally defined as above. The additional hyper-parameter for DCMM specifications, the random effects discount factor ρ, was set at 0.6. Of all the predictors present in this set-up, the price, grocery mobility reports and restrictions levels were found to have the greatest predictive contributions, particularly the mobility reports, which was a trend that was observed for other stores as well. Figure 13 displays the 3-week ahead forecast for the second half of 2020 in comparison to a GLM model with the same variable set, whereas Table 5 also includes the comparison for a 1-week ahead forecast horizon. The latent factor DCMM particularly seems to outperform the standard GLM in forecasting zero observations, where the latter tends to consistently generate forecasts around the baseline level of 1. Notable deviations are for the weeks including Thanksgiving and Christmas, which see spikes in both models.

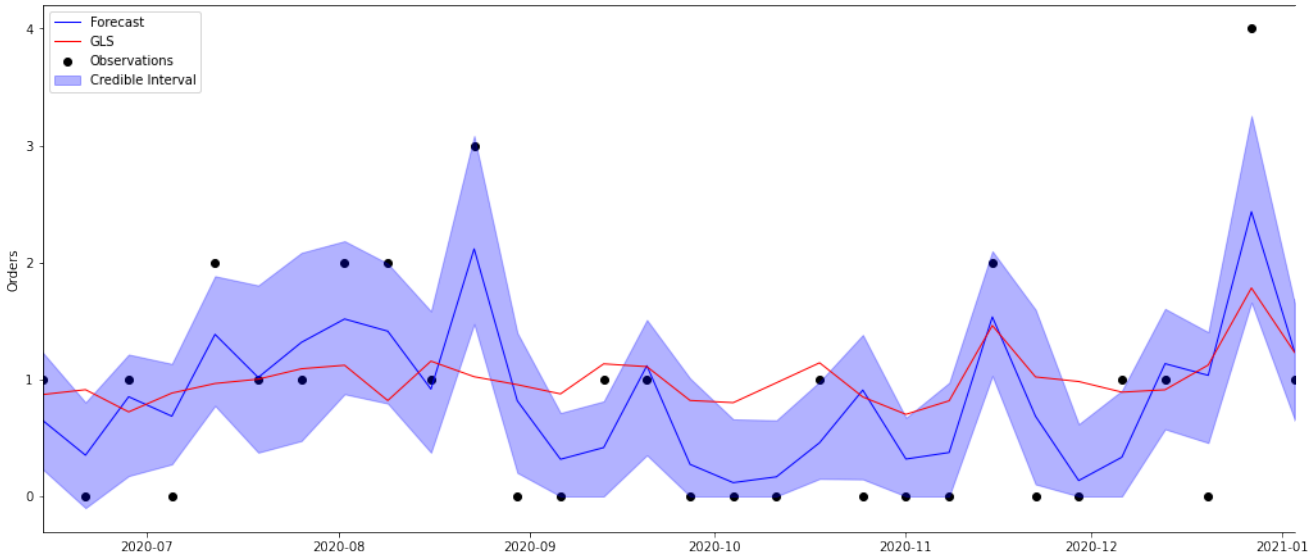


Figure 13: 3-week ahead forecast for $Y_{t,\mathcal{M}_1}$, the orders of one product by an individual store, the shaded blue area represents the 80% credible interval whereas the red line shows an GLS estimate

	Forecast ZAPE	
Model	1-step	3-step
M_1	29.142	34.152
GLS	54.041	59.142

Table 5: Comparison of the Zero-Adjusted Absolute Percentage Error for 1 & 3 week ahead forecasts between the latent factor DCMM, $\mathcal{M}_1$, and a GLS regression

7 Discussion

Whilst the model framework lends itself to the forecasting of product sales of all kinds and provided favorable forecasting performance compared to standard OLS and GLS models, it was also shown to have disadvantages. In a general summation, the model framework was sensitive and hard to calibrate, partly due to the characteristics of the data set used in this case study as will be explained below, but also due to it's inherent structure. This section serves to present the challenges undergone during the modelling process, beginning with the data-specific issues and concluding with possible improvements.

To properly exploit the merits of the multi-scale approach, careful consideration must be made by the modeller in regard to the nature of the data, which common factors are trying to be estimated and what aggregation level is most suited to do so. This became particularly evident after analysis was performed under the initial set-up of the model, which led to ambiguous results. The initial model design was as follows:

1. **State Level**: DLM to model the daily aggregated sales of all observations for all counties in order to capture the effect of temperature, precipitation, holiday and seasonality (daily and monthly).
2. **County/Category Level**: DLM or DGLM depending on the distribution of the county/category level sales data, to model the daily aggregated sales of all product categories per county and capture the effect of the COVID statistics, restrictions and mobility reports.
3. **Store/Product Level**: DCMM to model the daily sales of each individual product in each store where the effects of the aforementioned variables will be incorporated through the latent factors.

The flaws in this design turned out to be two-fold, firstly the set-up failed to account for the wholesale nature of the data thereby making it impossible to determine the optimal lag-length as laid out in Subsection 6.1.1. Whilst the aggregations imposed in the lay-out above were revealed to be inappropriate, particularly for the third level, because the majority of stores were found to not place new wholesale orders on a day-to-day basis but rather have a fixed weekly stocking schedule. Therefore modelling all weekdays within the same model led to inaccurate forecasts, an example is shown in Appendix A.5. Whilst the model correctly captures the day-to-day seasonality, the magnitude of the orders are systematically underestimated. A model specification with a local level (as shown in the appendix) led to forecasts for high-order days which lay between the high and low-order days because the level had to cover each day of the week thereby settling on a midway compromise. A more suitable approach for modelling daily sales of this nature would be to be model

each day separately, with bespoke local levels and volatility parameters. These issues stemmed from the wholesale nature of the data and it should therefore be noted that other applications of the model framework may not encounter these problems. However, a model intended for commercial purposes is only of limited practical relevance if it proves to be needlessly hard to calibrate, the difficulty in modelling the wholesale data of this case study could therefore be mentioned as a downside.

A point of possible improvement for future applications of the framework is hyper-parameter tailoring. In this case study, the level of the hyper-parameters (discount factors (β, δ)) was determined by applying a grid-search, wherein the forecasting performance was compared, to $\mathcal{M}_0$ in the first specification of Table 3. These discount factors were later imposed on all other model specifications due to the computationally demanding nature of the grid search, it may however be that searching for bespoke discount factors for each model specification leads to better results. A further improvement could be the inclusion of informative priors. As explained in Subsection 5.5 the models were initialized with the objective g-prior, however given that certain variables provided relatively few non-zero data points making the influence of the prior greater, the results may improve if a modeller is able to input accurate informative priors. Lastly, given that the higher-level $\mathcal{M}_0$ models served primarily to generate latent factors for $\mathcal{M}_1$, a potential point of further research may be smoothing the time-series of $\mathcal{M}_0$ in a bid to generate better forecasts of the latent factors effect. The time-series of $\mathcal{M}_0$ were often found to be affected by outliers, with single orders of tens of thousands of bottles occasionally occurring. If the forecasts of $\mathcal{M}_0$ are not of importance to the modeller it may therefore be worthwhile to smooth the time-series to reduce the impact of these outliers.

8 Conclusions

In this thesis a model framework is tested which has been designed to forecast the sales of products with all types of demand, from intermittent to steady. A framework of such kind would have great commercial uses, if found to perform adequately. An expansive, publicly available data set was chosen as the subject of the case study which included liquor products with the desired varying levels of demand. The data was of a wholesale nature, which induced several challenges which may have hampered the forecasting performance of the various models found in the layers of the framework. Nonetheless, the framework generated favorable results compared to standard alternative models for both intermittent and frequently sold products. By estimating common effects at an aggregated level, in order to circumvent the noise of individual time-series, and sharing these effects via latent factors with these individual time-series the model framework proved to be scalable. The output generated was also easily interpretable, as is common to linear models, thereby enhancing the practical relevance of the framework. There were however downsides as well, the model framework was sensitive and often required great amounts of work to optimize for a particular specification, the reducing of a discount factor by 0.01 or shortening of the prior sample size by 10 days could completely shift the outcome, making the framework unstable in certain specifications.

A further point of research in this case study was the impact of the COVID-19 pandemic on liquor wholesales in Iowa. Whilst the analysis provided inconclusive results for certain research questions, a general picture of the impact could still be composed. Over the course of 2020 the residents of

Iowa have not appeared to significantly change the amount of liquor they consume, with the total level following along the trend of the past few years namely a slight but steady increase. There were however changes in the composition of their demand, three liquor categories in particular, Spirits, Cocktails and Tequila appeared to suddenly enjoy increased consumption during the harshest restrictions. Leading to the hypothesis that grocery store sales of these 'party' liquors may benefit from the closing of bars and nightclubs as people attempt to move the parties to their own homes. It was also striking that the impact of the COVID restrictions differed immensely between stores, with some stores seeing increased sales whilst others saw their entire demand decimated, this was even the case for stores belonging to the same chain and located within the same city. It would therefore be an interesting line of further research to compare the sales of liquor retailers with and without online delivery, as this could well be the cause of the differing impacts. As far as forecasting power goes for the COVID-related predictors, the Google mobility reports for grocery and retail stores proved to be the strongest in addition to the harshest level of restrictions, which had the greatest impact on almost every time-series observed.

References

Aguilar, O., Prado, R., Huerta, G., & West, M. (1999). Bayesian inference on latent structure in time series. *Bayesian Statistics*, *6*, 3-26.

Aguilar, O., & West, M. (2000). Bayesian dynamic factor models and portfolio allocation. *Journal of Business and Economic Statistics*, *18*, 385-409.

Aktekin, T., Polson, N. G., & Soyer, R. (2018). Sequential bayesian analysis of multivariate count data. *Bayesian Analysis*, *13*, 385-409.

Aras, S., Deveci Kocakoç, İ., & Polat, C. (2017). Comparative study on retail sales forecasting between single and combination methods. *Journal of Business Economics and Management*, *18*(5), 803–832.

Archer, E., Park, I. M., Buesing, L., Cunningham, J., & Paninski, L. (2015). Black box variational inference for state space models. *arXiv preprint arXiv:1511.07367*.

Arunraj, N. S., & Ahrens, D. (2015). A hybrid seasonal autoregressive integrated moving average and quantile regression for daily food sales forecasting. *International Journal of Production Economics*, *170*, 321–335. doi: 10.1016/j.ijpe.2015.09.039

Berry, L. R., Helman, P., & West, M. (2020). Probabilistic forecasting of heterogeneous consumer transaction-sales time series. *International Journal of Forecasting*, *36*, 552–569. Retrieved from `https://doi.org/10.1016/j.ijforecast.2019.07.007` (arXiv:1808.04698. Published online Nov 25 2019) doi: https://doi.org/10.1016/j.ijforecast.2019.07.007

Berry, L. R., & West, M. (2019). Bayesian forecasting of many count-valued time series. *Journal of Business & Economic Statistics*, *38*, 872-887.

Box, G., Jenkins, G., & Reinsel, G. (2008). *Time series analysis: Forecasting and control* (Vol. 4). Wiley.

Calder, C., & Cressie, N. A. (2009). *Kriging and variogram models*.

Carvalho, C. M., Johannes, M. S., Lopes, H. F., & Polson, N. G. (2010). Particle learning and smoothing. *Statistical Science*, *25*(1), 88–106.

Chambers, J. C., Mullick, S. K., & Smith, D. (1971, Aug). How to choose the right forecasting technique. *Harvard Business Review*. Retrieved from `https://hbr.org/1971/07/how-to-choose-the-right-forecasting-technique`

Chen, C., So, M., Li, J. C., & Sriboonchitta, S. (2016). Autoregressive conditional negative binomial model applied to over-dispersed time series of counts. *Statistical Methodology*, *31*, 73–90. doi: 10.1016/j.stamet.2016.02.001

Chen, C. W., & Lee, S. (2016). Bayesian causality test for integer-valued time series models with applications to climate and crime data. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *66*(4), 797–814. doi: 10.1111/rssc.12200

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785–794). New York, NY, USA: ACM. Retrieved from `http://doi.acm.org/10.1145/2939672.2939785` doi: 10.1145/2939672.2939785

Chen, X., Banks, D., & West, M. (2019). Bayesian dynamic modeling and monitoring of network flows. *Network Science*, *7*(3), 292–318.

Chen, X., Irie, K., Banks, D., Haslinger, R., Thomas, J., & West, M. (2018). Scalable bayesian modeling,

monitoring, and analysis of dynamic network flow data. *Journal of the American Statistical Association*, *113*(522), 519–533.

Croston, J. (1972). Forecasting and stock control for intermittent demands. *Operational Research Quarterly*, *23*, 289-303.

Ding, X., Qiu, Z., & Chen, X. (2017). Sparse transition matrix estimation for high-dimensional and locally stationary vector autoregressive models. *Electronic Journal of Statistics*, *11*(2), 3871-39021735–1780. Retrieved from `https://doi.org/10.1214/17-EJS1325`

E, W., & Lu, J. (2011). Multiscale modeling. *Scholarpedia*, *6*(10), 11527. (revision #91540) doi: 10.4249/scholarpedia.11527

Fisher, M., & Raman, A. (2010). *The new science of retailing: how analytics are transforming the supply chain and improving performance*. Harvard Business Review Press.

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

Gordon, N. J., Salmond, D. J., & Smith, A. F. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. In *Iee proceedings f (radar and signal processing)* (Vol. 140, pp. 107–113).

Gruber, L., & West, M. (2016). Gpu-accelerated bayesian learning and forecasting in simultaneous graphical dynamic linear models. *Bayesian Analysis*, *11*(1), 125–149.

Gruber, L., & West, M. (2017). Bayesian forecasting and scalable multivariate volatility analysis using simultaneous graphical dynamic models. *arXiv preprint arXiv:1606.08291*.

Helmer, O. (1994). Adversary delphi. *Futures*, *26*(1), 79–87. doi: 10.1016/0016-3287(94)90091-4

Hirche, M., Haensch, J., & Lockshin, L. (2021). Comparing the day temperature and holiday effects on retail sales of alcoholic beverages–a time-series analysis. *International Journal of Wine Business Research*.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Hyndman, R., Koehler, A. B., Ord, J. K., & Snyder, R. D. (2008). *Forecasting with exponential smoothing: the state space approach*. Springer.

Jiang, A., Tam, K. L., Guo, X., & Zhang, Y. (2019). A new approach to forecasting intermittent demand based on the mixed zero-truncated poisson model. *Journal of Forecasting*, *39*(1), 69–83. doi: 10.1002/for.2614

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, *82*(Series D), 35–45.

Kitagawa, G., & Gersch, W. (1996). Smoothness priors analysis of time series. *Leture Notes in Statistics*.

Kolassa, S. (2016). Evaluating predictive count data distributions in retail sales forecasting. *International Journal of Forecasting*, *32*, 788-803.

Koller, D., & Friedman, N. (2010). *Probabilistic graphical models: principles and techniques*. The MIT Press.

Lavine, I. (2020). *Bayesian computation for variable selection and multivariate forecasting in dynamic models* (Unpublished doctoral dissertation). Duke University.

Liu, N., Ren, S., Choi, T.-M., Hui, C.-L., & Ng, S.-F. (2013). Sales forecasting for fashion retailing

service industry: a review. *Mathematical Problems in Engineering*, *2013*.

Mahmoud, E. (1984). Accuracy in forecasting: a survey. *Journal of Forecasting*, *3*(2), 139-159. Retrieved from https://doi.org/10.1002/for.3980030203

McAlinn, K., & West, M. (2019). Dynamic bayesian predictive synthesis in time series forecasting. *Journal of Econometrics*, *210*(1), 155–169.

McCabe, B. P. M., & Martin, G. M. (2005). Bayesian predictions of low count time series. *International Journal of Forecasting*, *21*, 315-330.

Morlidge, S. (2015). Measuring the quality of intermittent-demand forecasts: It's worse than we've thought! *Foresight: The International Journal of Applied Forecasting*, *37*, 37-42.

Nakajima, J., & West, M. (2013). Bayesian analysis of latent threshold dynamic models. *Journal of Business & Economic Statistics*, *31*(2), 151–164.

Ping, X., Chen, Q., Liu, G., Su, J., & Ma, F. (2018). Particle filter based time series prediction of daily sales of an online retailer. In *2018 11th international congress on image and signal processing, biomedical engineering and informatics (cisp-bmei)* (p. 1-6). doi: 10.1109/CISP-BMEI.2018.8633040

Prado, R., & West, M. (2010). *Time series: Modeling, computation & inference*. Chapman & Hall/CRC Press.

Seeger, M., Rangapuram, S., Wang, Y., Salinas, D., Gasthaus, J., Januschowski, T., & Flunkert, V. (2017). Approximate bayesian inference in linear state space models for intermittent demand forecasting at scale. *arXiv preprint arXiv:1709.07638*.

Snyder, R. D., Ord, J. K., & Beaumont, A. (2012). Forecasting the intermittent demand for slow-moving inventories: A modelling approach. *International Journal of Forecasting*, *28*(2), 485–496. doi: 10.1016/j.ijforecast.2011.03.009

Wacker, J. G., & Lummus, R. R. (2002). Sales forecasting for strategic resource planning. *International Journal of Operations & Production Management*.

Wang, D., Zheng, H., Y. Lian, & Li, G. (2021). High-dimensional vector autoregressive time series modeling via tensor decomposition. *Journal of the American Statistical*. Retrieved from 10.1080/01621459.2020.1855183

West, M. (2013). Bayesian dynamic modelling. *Bayesian theory and applications*, 145-166.

West, M. (2020). Bayesian forecasting of multivariate time series: Scalability, structure uncertainty and decisions (with discussion). *Annals of the Institute of Statistical Mathematics*, *72*, 1–44. Retrieved from https://doi.org/10.1007/s10463-019-00741-3 (arXiv:1911.09656. Published online Dec 9 2019) doi: https://doi.org/10.1007/s10463-019-00741-3

West, M., & Harrison, P. (1997). *Bayesian forecasting and dynamic linear models*. New York: Springer-Verlag.

West, M., Harrison, P. J., & Migon, H. S. (1985). Dynamic generalized linear models and bayesian forecasting. *Journal of the American Statistical Association*, *80*(389), 73–83.

Yanchenko, A., Deng, D., Li, J., Cron, A., & West, M. (2021). Hierarchical dynamic modelling for individualized bayesian forecasting. *Department of Statistical Science, Duke University. Submitted for publication*. (arXiv:2101.03408)

Yelland. (2009). Bayesian forecasting for low-count time series using state-space models: An empirical

evaluation for inventory management. *International Journal of Production Economics*(118), 95-103.

Yelland, & Lee, E. (2003). Forecasting product sales with dynamic linear mixture models. *Sun Microsystems*, 22.

Zellner, A. (1986). On assessing prior distributions and bayesian regression analysis with g-prior distributions. *Bayesian inference and decision techniques*.

Zhang, C., Bütepage, J., Kjellström, H., & Mandt, S. (2018). Advances in variational inference. *IEEE transactions on pattern analysis and machine intelligence*, *41*(8), 2008–2026.

Zhao, Z., Xie, M., & West, M. (2016). Dynamic dependence networks: Financial time series forecasting and portfolio decisions. *Applied Stochastic Models in Business and Industry*, *32*, 311-339.

Zhou, X., Nakajima, J., & West, M. (2014). Bayesian forecasting and portfolio decisions using dynamic dependent sparse factor models. *International Journal of Forecasting*, *30*(4), 963–980.

Zhu, L., & Laptev, N. (2017). *Deep and confident prediction for time series at uber*. doi: 10.1109/ICDMW.2017.19

A Appendices

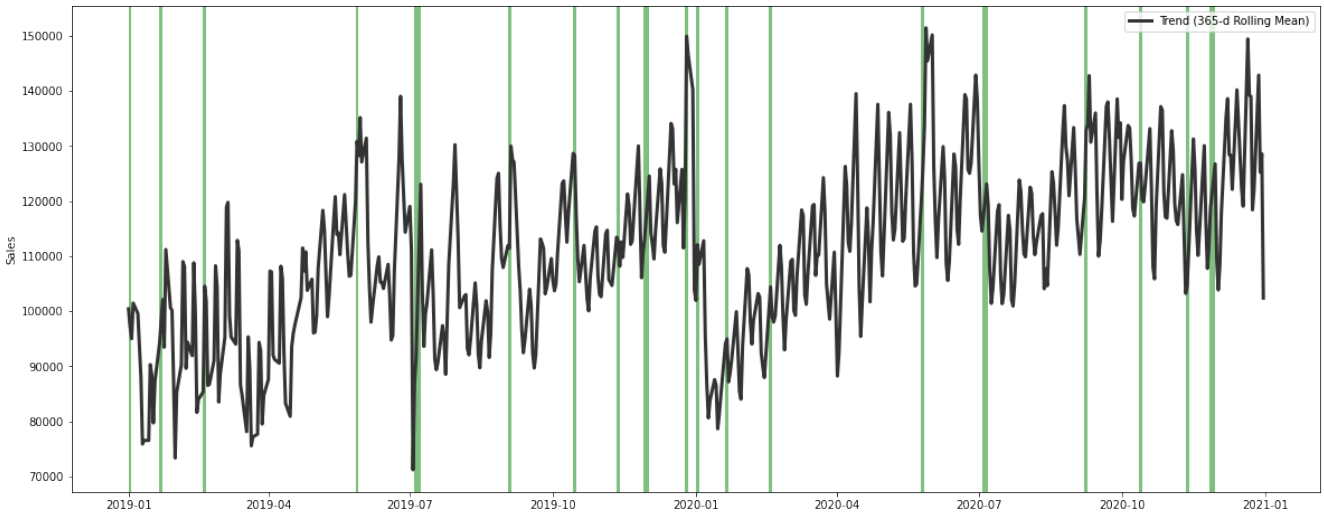
A.1 Timeline of Restrictions

Table 6: Timeline of Restrictions

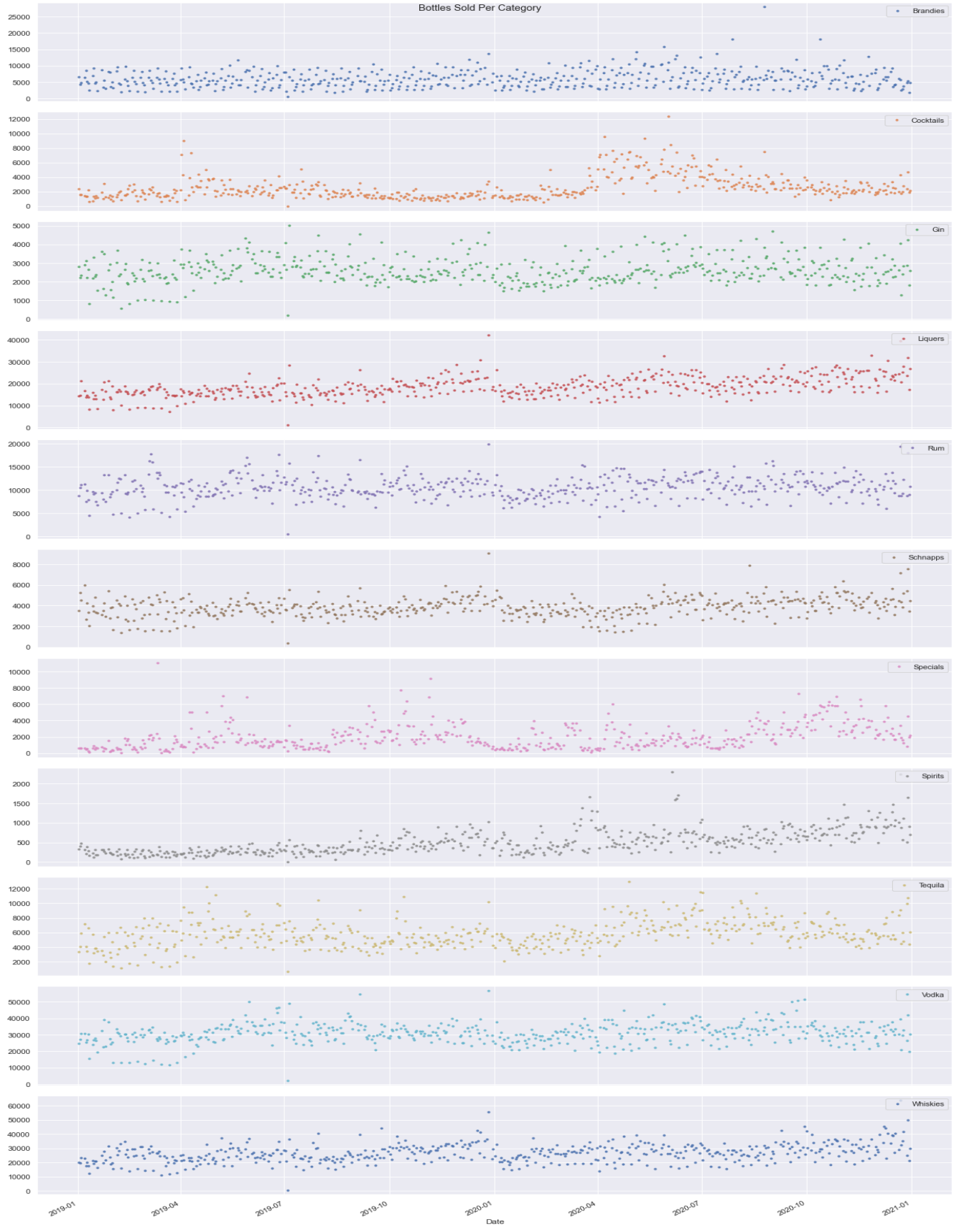
Date	Restrictions	Affected Counties
17th March	All restaurants and bars are closed for dine-in till 30th of March.	All Counties
20th March	All bars are allowed to do carry-out orders of alcohol.	All Counties
26th March	Closing of bars and restaurants extended till 7th April, many retail shops closed.	All Counties
2nd April	Closing of bars and restaurants extended till 30th of April.	All Counties
6th April	Closing of social clubs, arcades, parks, music stores etc.	All Counties
1st May	77 Counties will reopen restaurants, gyms and stores. 22 remained closed.	See Table Caption
13th May	Restaurants and stores can open with restrictions.	All Counties
28th May	Bars, wineries and distilleries can reopen at 50% capacity.	All Counties
27th August	Bars, nightclubs and breweries closed. No alcohol is restaurants after 10pm till 5th September.	Polk, Black Hawk, Johnson, Story, Dallas & Linn
16th September	Bars allowed to re-open.	Polk, Black Hawk, Dallas & Linn
18th September	Bar closing extended till 27th September.	Story & Johnson
5th October	Bars can re-open.	Story & Johnson
17th November	All restaurants and bars must close between 10pm and 6 am, and limit group sizes to 8.	All Counties
16th December	Restrictions on bars and restaurants loosened, normal hours resumed.	All Counties

These 22 are: Allamakee, Benton, Black Hawk, Bremer, Dallas, Des Moines, Dubuque, Fayette, Henry, Iowa, Jasper, Johnson, Linn, Louisa, Marshall, Muscatine, Polk, Poweshiek, Scott, Tama, Washington, Woodbury.

A.2 State-Wide Sales with Highlighted Holidays



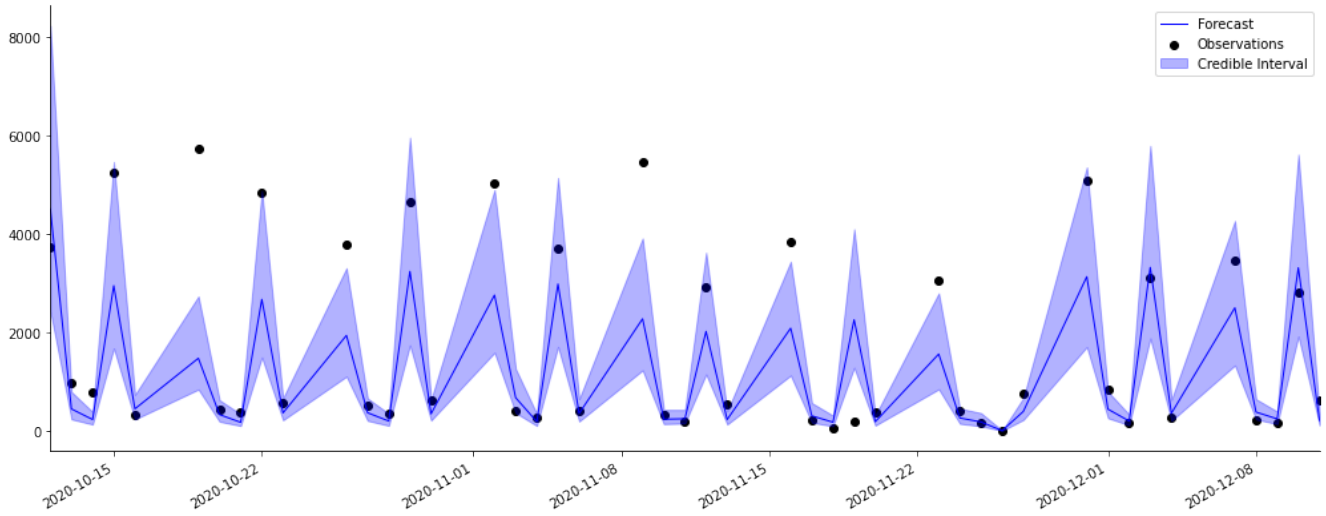
A.3 Category Sales



A.4 Table of DGLM Conjugate Information

	Distribution		
	Bernoulli	Poisson	Normal
$y_t \sim$	Relabeled as $z_t \sim \text{Ber}(\pi_t)$	$\text{Poi}(\mu_t)$	$\mathcal{N}(\mu_t, v_t)$
$\eta_t =$	$\text{logit}(\pi_t)$	$\log(\mu_t)$	μ_t
$\phi =$	1	1	$\phi_t = 1/v_t$
$a(\eta_t) =$	$\log(1 + \exp(\eta_t))$	$\exp(\eta_t)$	$\eta_t^2/2$
$b(y_t, \phi_t) =$	1	$1/y_t!$	$(\phi_t/2\pi_t)^{1/2}\exp(-\phi_t y_t^2/2)$
Conjugate Prior	$\pi_t \sim \text{Beta}(\alpha_t, \beta_t)$	$\mu_t \sim \text{Gamma}(\alpha_t, \beta_t)$	$\mu_t \sim \mathcal{N}(a_t, A_t v_t)$
Forecast Distribution	$\text{BetaBer}(1, \alpha_t, \beta_t)$	$\text{NegBin}(\alpha_t, \beta_t/(1+\beta_t))$	$\text{T}_{\delta n_t - 1}(f_t, Q_t)$

A.5 Daily Store Forecasts



B Code

B.1 Code Gibbs Sampler Latent Factor DLM

```
def Gibbs_LF_DLM(y,a_j,A_j,n_j,delta,m_0,C_0,n_0,s_0,burn_in,mcmc_iter,reg):
    def sv(x):
        return (x+x.T)/2
    mcmc_iter = burn_in+mcmc_iter
    T = len(y)
    p_x = a_j.shape[1]
    p_f = reg.shape[1]
    p = p_x+p_f;
    m_t = np.zeros((T+1,p))
    C_t = np.zeros(((T+1),p,p))
    n_t = np.zeros((T+1,1))
    s_t = np.zeros((T+1,mcmc_iter))
    v_t = np.zeros((T,mcmc_iter))
    a_t = np.zeros((T,p))
    R_t = np.zeros((T,p,p))
    f_t = np.zeros((T,1))
    q_t = np.zeros((T,1))
    phi_t = np.zeros((T,p_x))
    X_t = np.zeros((T,p_x,(mcmc_iter+1)))
    theta_t = np.zeros((T,p,mcmc_iter))
    a_k = np.zeros(((mcmc_iter),p))
    R_k = np.zeros(((mcmc_iter),p,p))
    v_k = np.zeros(((mcmc_iter),1))
    n_k = np.zeros((1,1))
    d = delta[0]
    beta = delta[1]
    m_t[0,:] = m_0
    C_t[0,:,:] = C_0
    n_t[0] = n_0
    s_t[0,:] = s_0
    for t in range(0,T):
        phi_t[t,:] = ((0.5*beta*n_j[t,:])/np.random.gamma(shape=0.5*beta*n_j[t,:],scale
            =1))
        X_t[t,:,0] = a_j[t,:]+np.random.standard_normal(size=(1,len(a_j[t,:])))@np.linalg
            .cholesky(sv(np.diag(phi_t[t,:]*A_j[t,:]))).T
    for i in range(0,mcmc_iter):
        # forward-filter
        for t in range(0,T):
            F_t = np.insert(np.array(X_t[t,:,i]), 0, np.array(reg[t,:]))
            # prior for time t
            a_t[t,:] = m_t[t,:]
            R_t[t,:,:] = C_t[t,:,:]/d
            # predict time t
            f_t[t] = F_t @ a_t[t,:]
            q_t[t] = (F_t @ C_t[t,:,:] @ F_t/d)+s_t[t,i]
            # compute forcast error and adaptive vector
            e_t = y[t]-f_t[t]
```

```
            A_t = R_t[t,:,:] @ F_t/q_t[t]
            # posterior for time t
            n_t[t+1] = beta*n_t[t]+1
            r_t = (beta*n_t[t]+(e_t**2)/q_t[t])/n_t[t+1]
            s_t[t+1,i] = r_t*s_t[t,i]
            m_t[t+1,:] = a_t[t,:]+ A_t*e_t
            C_t[t+1,:,:] = sv(r_t*(R_t[t,:,:]- q_t[t]*(A_t.reshape(len(A_t),1)*A_t)))
        # sample theta at T
        v_t[-1,i] = 1.0/np.random.gamma(shape=n_t[-1]/2, scale= 2/(n_t[-1]*s_t[-1,i]))
        theta_t[T-1,:,i] = m_t[-1,:]+np.random.standard_normal(size=(1,len(m_t[-1,:])))
            @np.linalg.cholesky(sv(C_t[-1,:,:]*(v_t[-1,i]/s_t[-1,i]))).T
        # theta at T+1
        n_k = beta*n_t[-1]+1
        v_k[i,0] = 1.0/np.random.gamma(shape=beta*n_t[-1]/2,scale=2/(beta*n_t[-1]*s_t[-1,
            i]))
        a_k[i,:] = m_t[-1,:]
        R_k[i,:,:] = (C_t[-1,:,:]/d)*(v_k[i,0]/s_t[-1,i])
        # backward-sampler
        for t in range(T-2,-1,-1):
            v_t[t,i] = 1/(1/v_t[t+1,i]*beta + np.random.gamma(shape=(1-beta)*n_t[t+1]/2,
                scale=2/(n_t[t+1]*s_t[t+1,i])))
            m_star_t = m_t[t+1,:]+d*(theta_t[t+1,:,i]-a_t[t+1,:])
            C_star_t = C_t[t+1,:,:]*(1-d)*(v_t[t,i]/s_t[t+1,i])
            theta_t[t,:,i] = m_star_t + np.random.standard_normal(size=(1,len(m_star_t)))
                @np.linalg.cholesky(sv(C_star_t)).T
        #sample X_t
        for t in range(0,T):
            A_st = np.diag(phi_t[t,:]*A_j[t,:])
            a_st = a_j[t,:]
            theta_p = theta_t[t,p_f:,i]
            theta_1 = theta_t[t,0:p_f,i]
            sigma = theta_p@A_st/(v_t[t,i]+theta_p@A_st@theta_p)
            a_star = a_st+sigma*(y[t]-(theta_1@reg[t,:]+theta_p@a_st))
            A_star = sv(A_st-A_st@theta_p.reshape(len(theta_p),1)*sigma)
            X_t[t,:,i+1] = a_star+np.random.standard_normal(size=(1,len(a_star)))@np.
                linalg.cholesky(sv(A_star)).T
            phi_t[t,:] = ((0.5*(n_j[t,:]+1)/np.random.gamma(shape=(n_j[t,:]+(X_t[t,:,i
                +1]-a_st)**2/A_j[t,:])/2, scale=1)))
    # save results
    theta_post = theta_t[:,:,burn_in:]
    X_post = X_t[:,:,burn_in+1:]
    a_k = a_k[burn_in:,:]
    R_k = R_k[burn_in:,:,:]
    v_k = v_k[burn_in:,:]
    return a_k,R_k,v_k,n_k,theta_post,X_post
```

```
E_Gibbs = np.zeros((mcmc_iter,T-1)) # Posterior mean
V_Gibbs = np.zeros((mcmc_iter,T-1)) # Posterior variance
```

```
error = np.zeros((mcmc_iter,T-1))
mlike = np.zeros((mcmc_iter,T-1))
s = set(range(0,T-1))
ak_results = dict.fromkeys(s) #Posterior forecast coefficient mean
Rk_results = dict.fromkeys(s) #Posterior forecast coefficient variance
vt_results = dict.fromkeys(s) #Posterior forecast observation variance
nt_results = dict.fromkeys(s) #Posterior forecast degrees of freedom
nu = np.zeros((T-1,1))
def sv(x):
    return (x+x.T)/2
for t in range(49,T-1):
    y = yI[0:t+1]
    a_j = a[0:t+1,:]
    A_j = A[0:t+1,:]
    n_j = n[0:t+2,:]
    a_k,R_k,v_k,n_k,theta_post,X_post = Gibbs_LF_DLM(y,a_j,A_j,n_j,delta,m_0,C_0,n_0,s_0,
        burn_in,mcmc_iter,reg)
    ak_results[t] = a_k
    Rk_results[t] = R_k
    vt_results[t] = v_k
    nt_results[t] = n_k
    nu[t,0] = n_k
    for i in range(0,mcmc_iter):
        # sample x(t+1)
        lmbda = np.sqrt((0.5*delta[1]*n[t+1]/np.random.gamma(shape=delta[1]*n[t+1]/2,
            scale=1)))
        x_t = np.insert(np.array(a[t+1,:]+lmbda*np.random.standard_normal(size=(1,len(a[t
            +1,:])))@np.linalg.cholesky(sv(np.diag(A[t+1,:]))).T),0,np.array(reg[t+1,:]))
        # compute aggregated mean and variance
        E_Gibbs[i,t] = x_t@a_k[i,:]
        V_Gibbs[i,t] = x_t@R_k[i,:,:]@x_t.reshape(len(x_t),1)+v_k[i,:]
        error[i,t] = yI[t+1]-E_Gibbs[i,t]
        mlike[i,t] = np.exp(np.log(gamma(0.5*(nu[t,0]+1)))-np.log(gamma(0.5*nu[t,0]))
            -0.5*np.log(np.pi*nu[t,0]*V_Gibbs[i,t])-(0.5*(nu[t,0]+1))*np.log(1+1/(nu[t
            ,0]*V_Gibbs[i,t])*(yI[t+1]-E_Gibbs[i,t])**2))
Gibbs_error = np.mean(error[:,-K:],0) # Average error
w = range(1,len(Gibbs_error)+1)
Gibbs_rmse = ((np.cumsum(Gibbs_error[:]**2))/w)**(1/2) # RMSFE
Gibbs_mlike = np.cumsum(np.log(np.mean(mlike[:,-K:],0))) # Marginal likelihood
```

B.2 Code Other Functionalities

```
def forecast_path_lf_sample_dlm(mod, k, X=None, phi_samps = None):
nsamps = len(phi_samps)
samps = np.zeros([nsamps, k])
F = np.copy(mod.F)
for samp in range(nsamps):
    param1 = mod.param1
    param2 = mod.param2
```

```
    a = np.copy(mod.a)
    R = np.copy(mod.R)
    n = np.copy(mod.n)
    s = np.copy(mod.s)
    for i in range(k):
        # Plug in X values
        if mod.nregn > 0:
            F = update_F(mod, X[i, :], F=F)
            # Plug in phi sample
            F = update_F_lf(mod, phi_samps[samp][i], F=F)
            # Get mean and variance
            ft, qt = mod.get_mean_and_var(F, a, R)
            param1 = ft
            param2 = qt
            # Simulate next observation
            samps[samp, i] = mod.simulate(param1, param2, nsamps=1)
            # Update the  parameters:
            et = samps[samp,i] - ft
            # Adaptive coefficient vector
            At = R @ F / qt
            # Volatility estimate ratio
            rt = (n + et**2/qt)/(n + 1)
            # Kalman filter update
            n = n + 1
            s = s * rt
            m = a + At * et
            C = rt * (R - qt * At @ At.T)
            # Get priors a, R for the next time step
            a = mod.G @ m
            R = mod.G @ C @ mod.G.T
            R = (R + R.T) / 2
            # Discount information
            if mod.discount_forecast:
                R = R + mod.W
                n = mod.delVar*n
return samps
```

```
def update_lf_sample_forwardfilt_dlm(mod, y, F, a, R, phi):
F = update_F_lf(mod, phi, F=F)
ft, qt = mod.get_mean_and_var(F, a, R)
param1, param2 = ft, qt
n = mod.n
s = mod.s
# Get the log-likelihood of 'y' under these parameters
loglik = mod.loglik(y, param1, param2)
# Update the  parameters:
et = y - ft
# Adaptive coefficient vector
```

```
At = R @ F / qt
# Volatility estimate ratio
rt = (n + et**2/qt)/(n + 1)
# Kalman filter update
m = a + At * et
C = rt * (R - qt * At @ At.T)
return m, C, np.ravel(loglik)[0], rt
```

```
def update_lf_sample_dlm(mod, y = None, X = None, phi_samps = None, parallel=False):
if y is None or np.isnan(y):
    mod.t += 1
    mod.m = mod.a
    mod.C = mod.R
    # Get priors a, R for time t + 1 from the posteriors m, C
    mod.a = mod.G @ mod.m
    mod.R = mod.G @ mod.C @ mod.G.T
    mod.R = (mod.R + mod.R.T)/2
    mod.W = mod.get_W(X=X)
else:
    update_F(mod, X)
    # Update m, C using a weighted average of the samples
    if parallel:
        f = partial(update_lf_sample_forwardfilt_dlm, mod, y, mod.F, mod.a, mod.R)
        p = multiprocessing.Pool(10)
        output = p.map(f, phi_samps)
        p.close()
    else:
        output = map(lambda p: update_lf_sample_forwardfilt_dlm(mod, y, mod.F, mod.a, mod
            .R, p), phi_samps)
        mlist, Clist, logliklist, rlist = list(map(list, zip(*output)))
        w = (np.exp(logliklist) / np.sum(np.exp(logliklist))).reshape(-1,1,1)
        mlist = np.array(mlist)
        Clist = np.array(Clist)
        rlist = np.array(rlist)
        rt = np.sum(rlist*w, axis=0)
        mod.m = np.sum(mlist*w, axis=0)
        mod.C = np.sum(Clist*w, axis=0) + np.cov((mlist).reshape(-1, mod.m.shape[0]),
            rowvar=False, aweights = w.reshape(-1))
        # Add 1 to the time index
        mod.t += 1
        mod.n = mod.n+1
        mod.s = mod.s * rt
        # Get priors a, R from the posteriors m, C
        mod.a = mod.G @ mod.m
        mod.R = mod.G @ mod.C @ mod.G.T
        mod.R = (mod.R + mod.R.T)/2 # prevent rounding issues
        # Discount information if observation is observed
        mod.W = mod.get_W(X=X)
```

```
        mod.R = mod.R + mod.W
        mod.n = mod.delVar * mod.n
        #print('a:', mod.a)
        #print('R:', mod.R)
        #print('m:', mod.m)
        #print('C:', mod.C)
        #print('w', w)
        #print(mod.n, mod.s)
return rt
```

```
def analysis1(Y, X=None, k=1, forecast_start=0, forecast_end=0,
              nsamps=500, family = 'normal', n = None,
              model_prior = None, prior_length=20, ntrend=1,
              dates = None, holidays = [],
              seasPeriods = [], seasHarmComponents = [],
              latent_factor = None, new_latent_factors = None,
              ret=['model', 'forecast'],
              mean_only = False, forecast_path = False, idx = None,
              dlm_dof = None, **kwargs):
    # Check if it's a latent factor DGLM
    if latent_factor is not None:
        is_lf = True
        nlf = latent_factor.p
    else:
        is_lf = False
        nlf = 0
    if model_prior is None:
        mod = define_dglm(Y, X, family=family, n=n, prior_length=prior_length, ntrend=
            ntrend, nhol=nhol, nlf=nlf,seasPeriods=seasPeriods, seasHarmComponents=
            seasHarmComponents,**kwargs)
    else:
        mod = model_prior
    # Convert dates into row numbers
    if dates is not None:
        dates = pd.Series(dates)
        if type(forecast_start) == type(dates.iloc[0]):
            forecast_start = np.where(dates == forecast_start)[0][0]
        if type(forecast_end) == type(dates.iloc[0]):
            forecast_end = np.where(dates == forecast_end)[0][0]
    # Define the run length
    T = len(Y) + 1
    if ret.__contains__('model_coef'):
        m = np.zeros([T-1, mod.a.shape[0]])
        C = np.zeros([T-1, mod.a.shape[0], mod.a.shape[0]])
        if family == 'normal':
            n = np.zeros(T)
            s = np.zeros(T)
    if new_latent_factors is not None:
```

```
        if not ret.__contains__('new_latent_factors'):
            ret.append('new_latent_factors')
        if not isinstance(new_latent_factors, Iterable):
            new_latent_factors = [new_latent_factors]
        tmp = []
        for lf in new_latent_factors:
            tmp.append(lf.copy())
        new_latent_factors = tmp
    # Initialize updating + forecasting
    horizons = np.arange(1, k + 1)
    if mean_only:
        forecast = np.zeros([1, forecast_end - forecast_start + 1, k])
    else:
        forecast = np.zeros([nsamps, forecast_end - forecast_start + 1, k])
    rt = np.empty(len(Y), dtype='float64')
    for t in range(prior_length, T):
        if forecast_start <= t <= forecast_end:
            if t == forecast_start:
                print('beginning forecasting')
            if ret.__contains__('forecast'):
                if is_lf:
                    if forecast_path:
                        pm, ps = latent_factor.get_lf_forecast(dates.iloc[t])
                        dlm_n = dlm_dof.get_lf(dates.iloc[t])[0]
                        phi_samps = generate_lf_samples(pm, ps,mod.delVar,dlm_n, nsamps)
                        forecast[:, t - forecast_start, :] = forecast_path_lf_sample_dlm(
                            mod,k=k, X=X[t + horizons - 1, :],phi_samps=phi_samps)
                else:
                    if forecast_path:
                        forecast[:, t - forecast_start, :] = forecast_path_dlm(mod,k=k, X
                             = X[t + horizons - 1, :], nsamps=nsamps, approx = False)
                    else:
                        # Get the forecast samples for all the items over the 1:k step
                            ahead marginal forecast distributions
                        forecast[:, t - forecast_start, :] = np.array(list(map(
                            lambda k, x: mod.forecast_marginal(k=k, X=x, nsamps=nsamps,
                                mean_only=mean_only), horizons, X[t + horizons - 1, :])))
                                .squeeze().T.reshape(-1, k)
            if ret.__contains__('new_latent_factors'):
                for lf in new_latent_factors:
                    if(lf == new_latent_factors[0]):
                        lf.generate_lf_forecast(date=dates[t], mod=mod, X=X[t + horizons
                            - 1], k=k, nsamps=nsamps, horizons=horizons, idx = idx)
                    else:
                        lf.generate_lf_forecast(date=dates[t], mod=mod, X=X[t + horizons
                            - 1], k=k, nsamps=nsamps, horizons=horizons)
        # Now observe the true y value, and update:
        if t < len(Y):
            if is_lf:
                pm, ps = latent_factor.get_lf(dates.iloc[t])
```

```
                dlmn = dlm_dof.get_lf(dates.iloc[t])[0]
                phis = generate_get_lf_samples(pm, ps, mod.delVar, dlmn, nsamps)
                rt[t] = update_lf_sample_dlm(mod,y=Y[t], X=X[t], phi_samps=phis, parallel
                    =False)
            if ret.__contains__('model_coef'):
                m[t,:] = mod.m.reshape(-1)
                C[t,:,:] = mod.C
                if family == 'normal':
                    n[t] = mod.n / mod.delVar
                    s[t] = mod.s
            if ret.__contains__('new_latent_factors'):
                for lf in new_latent_factors:
                    if(lf == new_latent_factors[0]):
                        lf.generate_lf(date=dates[t], mod=mod, Y=Y[t], X=X[t], k=k,
                            nsamps=nsamps, idx = idx)
                    else:
                        lf.generate_lf(date=dates[t], mod=mod, Y=Y[t], X=X[t], k=k,
                            nsamps=nsamps)
    out = []
    for obj in ret:
        if obj == 'forecast': out.append(forecast)
        if obj == 'model': out.append(mod)
        if obj == 'model_coef':
            mod_coef = {'m':m, 'C':C}
            if family == 'normal':
                mod_coef.update({'n':n, 's':s})
            out.append(mod_coef)
        if obj == 'new_latent_factors':
            if len(new_latent_factors) == 1:
                out.append(new_latent_factors[0])
            else:
                out.append(new_latent_factors)
        if obj == 'rt': out.append(rt)
    print('adapt_discount:', mod.adapt_discount , 'discount_forecast', mod.
        discount_forecast, 'delregn:', mod.delregn, 'dellf:', mod.dellf, 'deltrend:', mod
        .deltrend)
    if len(out) == 1:
        return out[0]
    else:
        return out
```